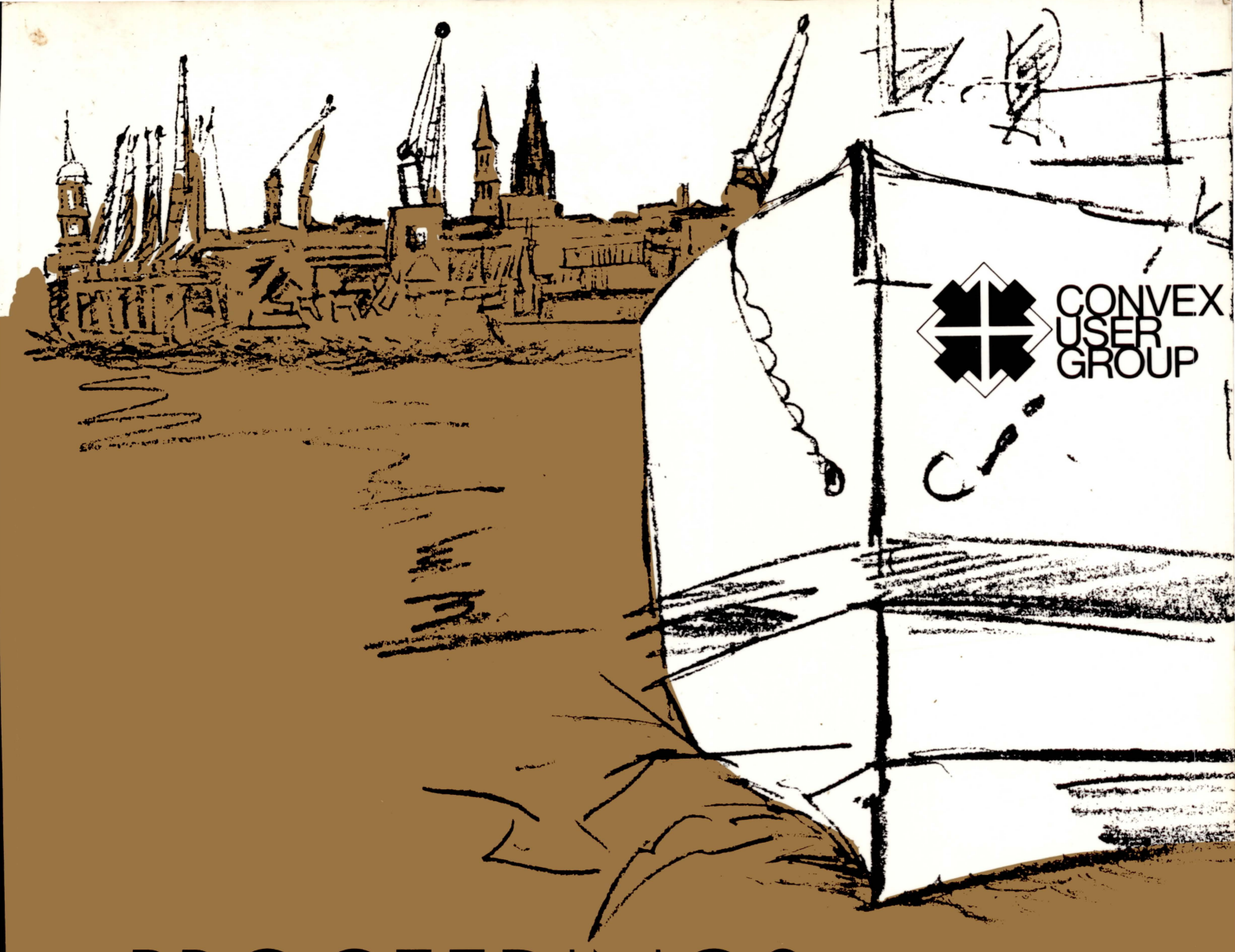


CONVEX USER GROUP

PROCEEDINGS

October 9,10 and 11, 1991

European CONVEX User Conference 1991

Trusthouse Forte Hotel, Hamburg, Germany

November 22, 1991

PROGRAM

The Conference runs from the evening of Wednesday, October 9 until the afternoon of October 11, and possibly a Perl tutorial on Wednesday morning.
Tutorial in room ALTENWERDER.

Perl is an interpreted language that allows to manipulate text, files and processes easily.
It provides a concise and readable way to do many jobs, especially system management tasks, which would otherwise have to be done programming in C, sed, awk or one of the shells, all of which it resembles as well as Pascal and Basic-Plus. Perl, which was written by Larry Wall, is freely available and runs on a variety of systems; it has been distributed with CONVEXOS since version 9.0. The later part of the course will be workshop-oriented approach where existing shell scripts are reworked as Perl scripts. Participants are encouraged to submit in advance existings scripts or ideas for such via email to <tchrist @ convex.com> for consideration.

Wednesday, October 9

16.00 Registration

19.30 Welcome dinner in restaurant MOORWERDER STUBE

Thursday, October 10

in room FRIESLAND

08.00 Registration

09.00 Opening, welcome
Jürgen Kabelitz, Head of German User Group

09.15 CONVEX Overview, Corporate overview,
Jim Balthazar, VP Marketing,
Convex European overview
John Hughes, VP European Operations, CONVEX

10.15 Coffee/Tea

10.45 Product overview
Frank Marshall, Senior VP Engineering, CONVEX
Quality & Reliability
Phil Struve, VP Customer Service, CONVEX Realtime System
Frank Marshall, Senior VP Engineering, CONVEX

12.00 Lunch/Demo

A. SYSTEM MANAGEMENT

in room OSTFRIESLAND

14.00 Security
Jerry Schieffer, General Manager, Systems Software, CONVEX

14.30 CONVEXOS/Secure: UNIX security in perspective?
H.A.M. Luijf, TNO Physics and Electronic Lab., The Netherlands

15.00 Multiple stripe configuration in a CONVEX C240
Maite Sierra, Centro Informatico Cientifico de Andalucia, Spain

15.30 Coffee/Tea

16.00 SysAdmin: A tool for Distributed Management Environment
Martin van Roon, Pink Elephant, The Netherlands

16.30 A help system for the ex/vi editor
Marina Buitrago, Centro Informatico Cientifico de Andalucia, Spain

17.00 Demo

18.00 Social evening

B. VISUALIZATION AND APPLICATIONS

in room NORDFRIESLAND

14.00 Simulation Equipment for High Definition Television, Acquisition and Visualization
M. Crouzet, C.C.E.T.T, France

14.30 FE-Simulation and visualization in groundwater flow and groundwater pollution using high performance systems
W. Hass, M. Resch, R. Branter,

15.00 Visualization, X11R5, PEX
Dave Holt, Manager, Software Systems Engineering, CONVEX

15.30 Coffee/Tea

16.00 A Supercomputing Environment in Climate Research
Hartmut Fichtel, Dt. Klimarechenzentrum, Germany

16.30 Finite element of rubber parts in HUTCHINSON
Daniel Benoualid, B. Ravier, I. Wander, France

17.00 Demo

18.00 Social evening

Friday, October 11

A. COMPILERS AND PROGRAM DEVELOPMENT
in room OSTFRIESLAND

09.00 Parallel Algorithms
Invited speaker:
Wolfgang Gentzsch, GENIAS, Gesellschaft für numerisch-intensive Anwendungen und Supercomputing, Germany

10.00 Parallelization on Shared-Memory, Virtual-Shared-Memory and Local-Memory-Machines - A Comparison
Alfred Geiger ,Universität Stuttgart, Germany

10.30 Coffee/Tea

11.00 CXpa, CXdb, Application Compiler
Dave Holt, Manager, Software Systems Engineering, CONVEX

12.30 Lunch/Demo

B. LARGE SITES
in room NORDFRIESLAND

09.00 CONVEX System Management
Jorgen Olsen, DOU Odense University, Denmark

09.30 CONVEX C210
Robotic Cartridge Loader System and Unattended Processing at KSEPL
S. Verdouw, Koninklijke/Shell Expl. Prod. Lab., The Netherlands

10.00 Virtual Volume Manager (RAID product)
Jerry Schieffer, General Manager, Systems Software, CONVEX

10.30 Coffee/Tea

11.00 Evaluating a CONVEX as a file server
Malcom Read, Natural Environment Research Council

11.30 The File Server Concept at the University of Tübingen
Dr. Dietmar Kaletta, Eberhard Karls Universität, Germany

12.30 Lunch/Demo

14.00 User group business session report
Dick Kaas, ECU Chairman

- Financial report
  Charles Curran, Treasurer
- Questions
- Election of Committee
- Relationship with the World wide User Group
  Michael Padrick ,Pres. CONVEX User Group

15.00 Coffee/Tea

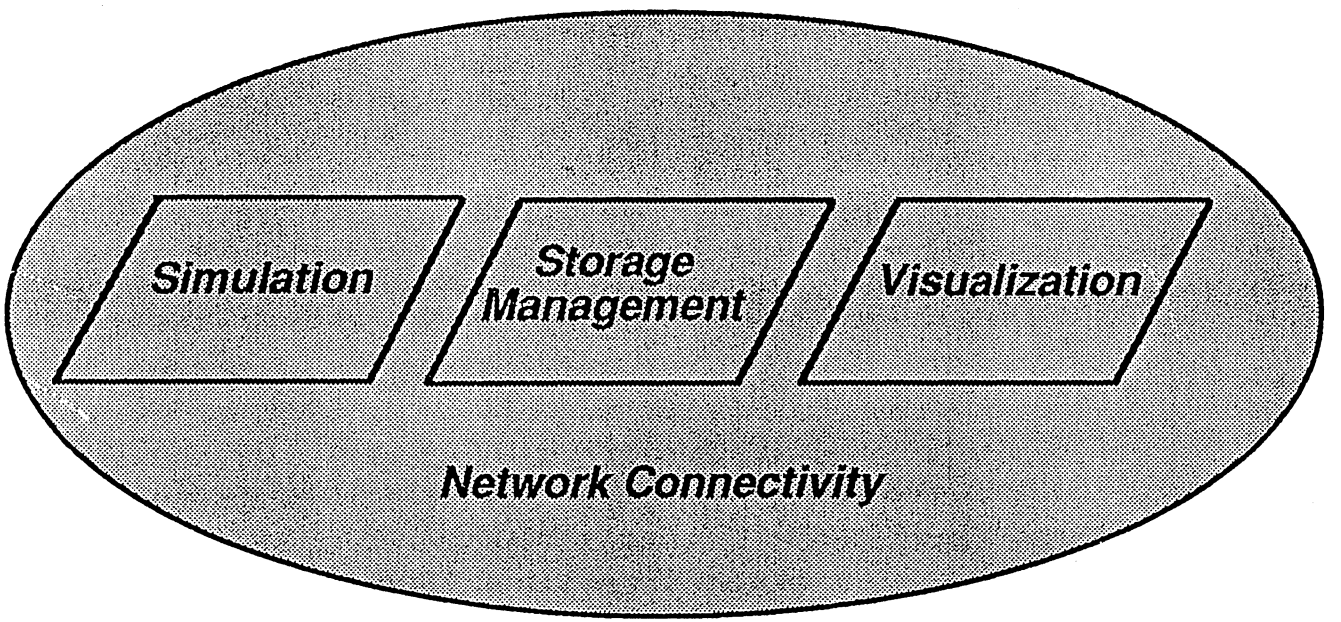
15.30 File-Serving with Unitree
Kent Angell, European Manager Titan

15.50 The EMASS system
Charles Riehm, E-Systems

16.10 Keynote Presentation: Technology direction
Steve Wallach, Senior VP Technology, CONVEX

17.00 Close of conference

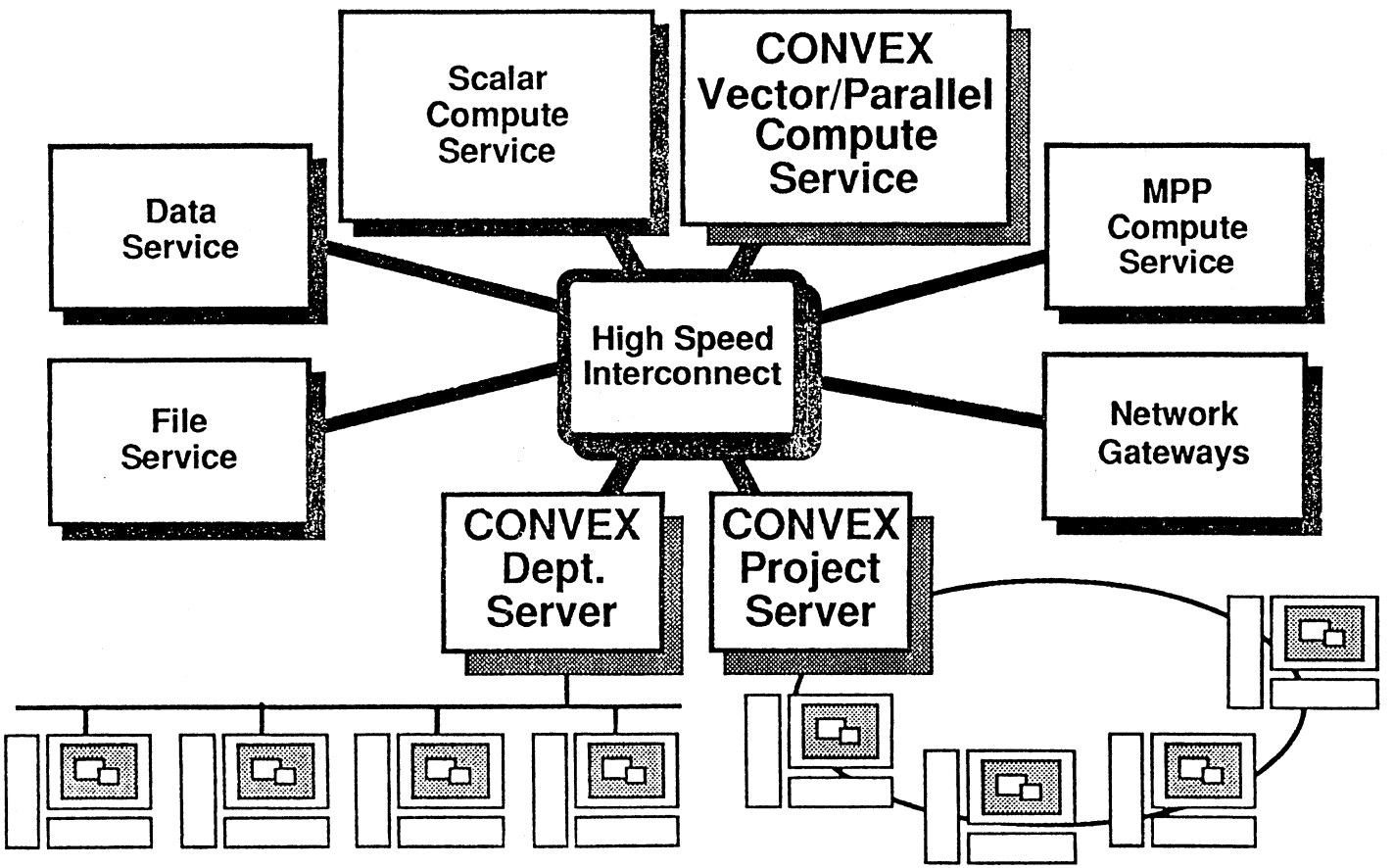
Open Supercomputing from Convex



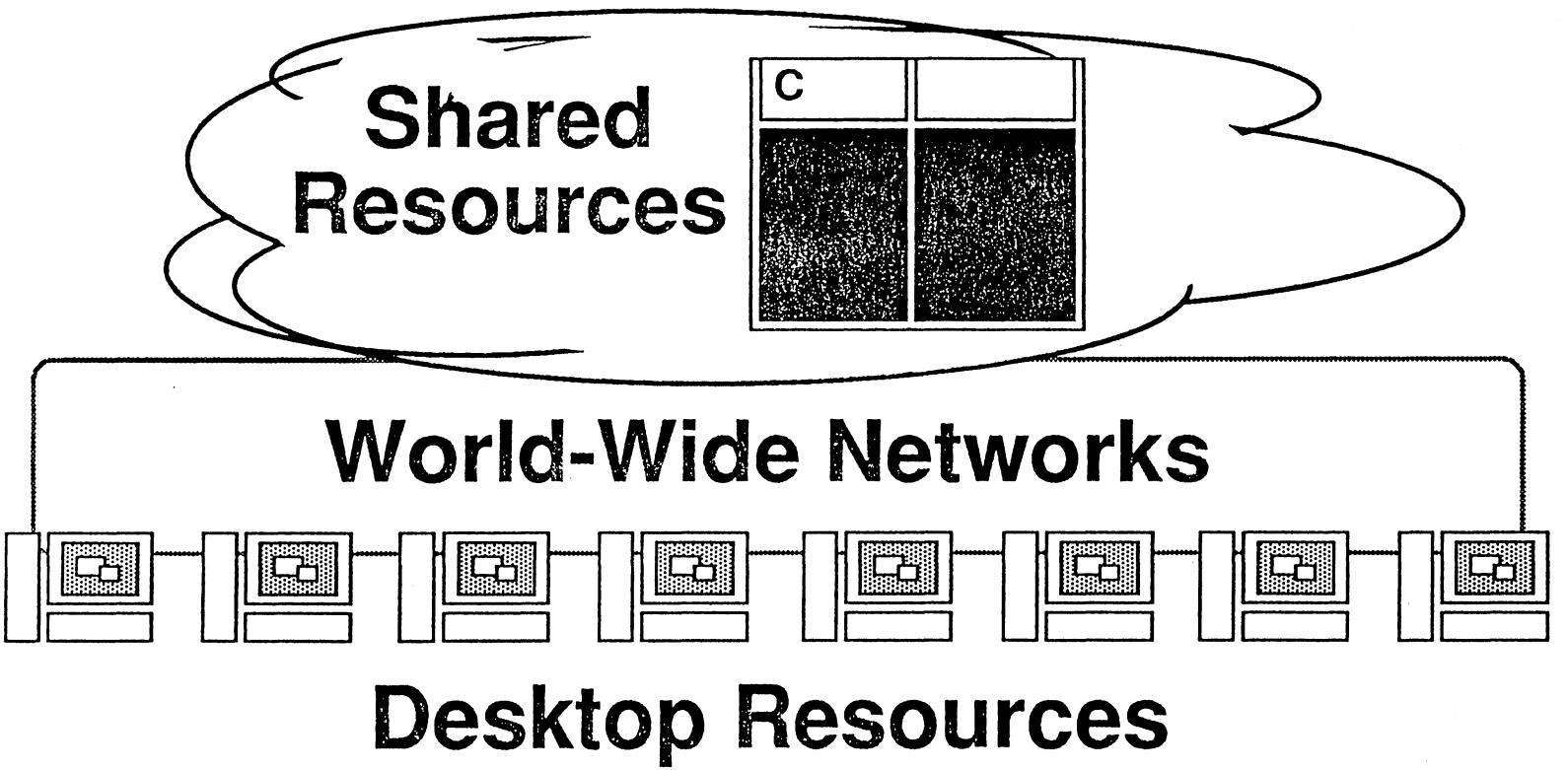
Total Solutions for Scientific Computing

Shared Resources

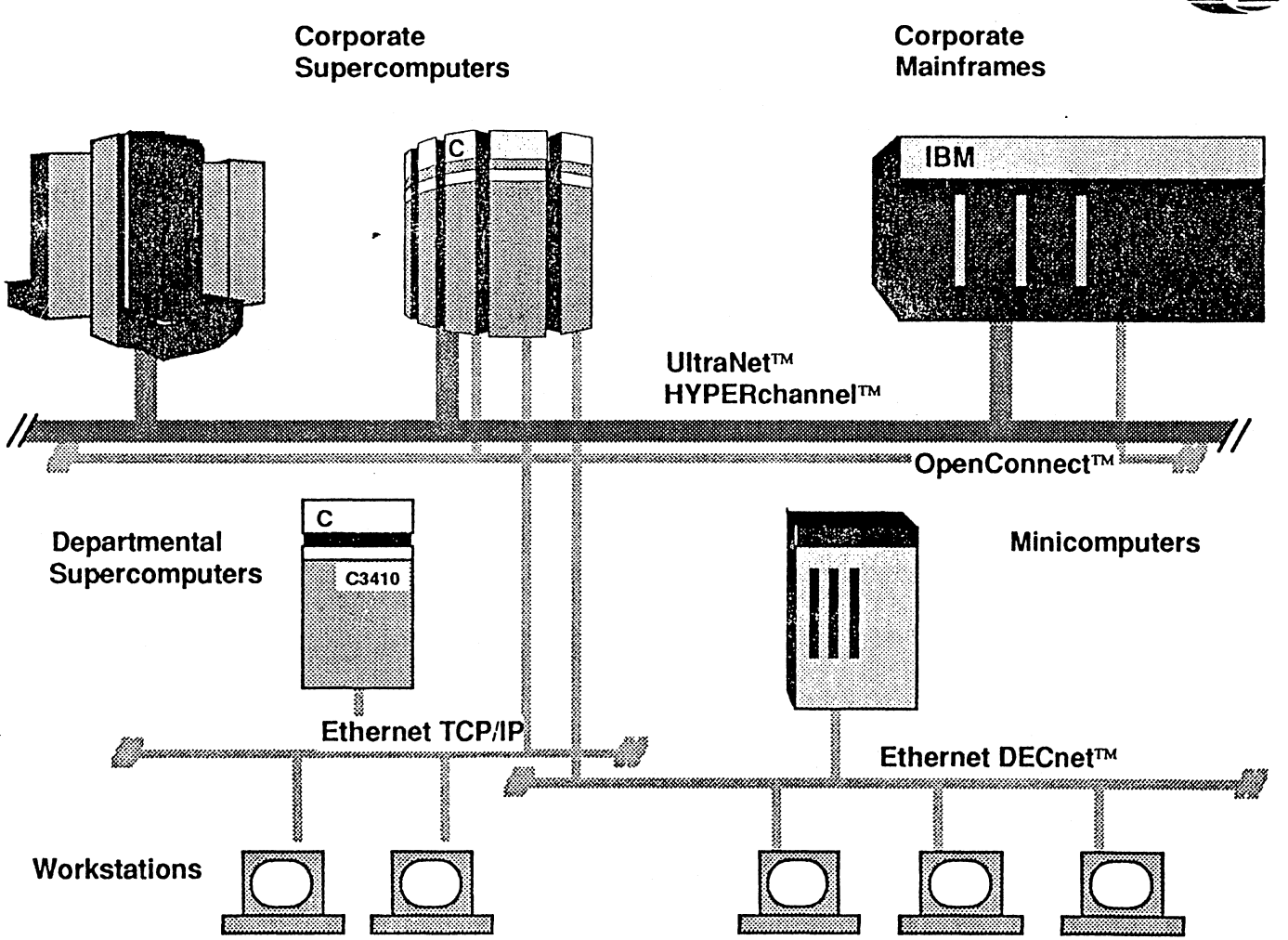
- CONVEX Today



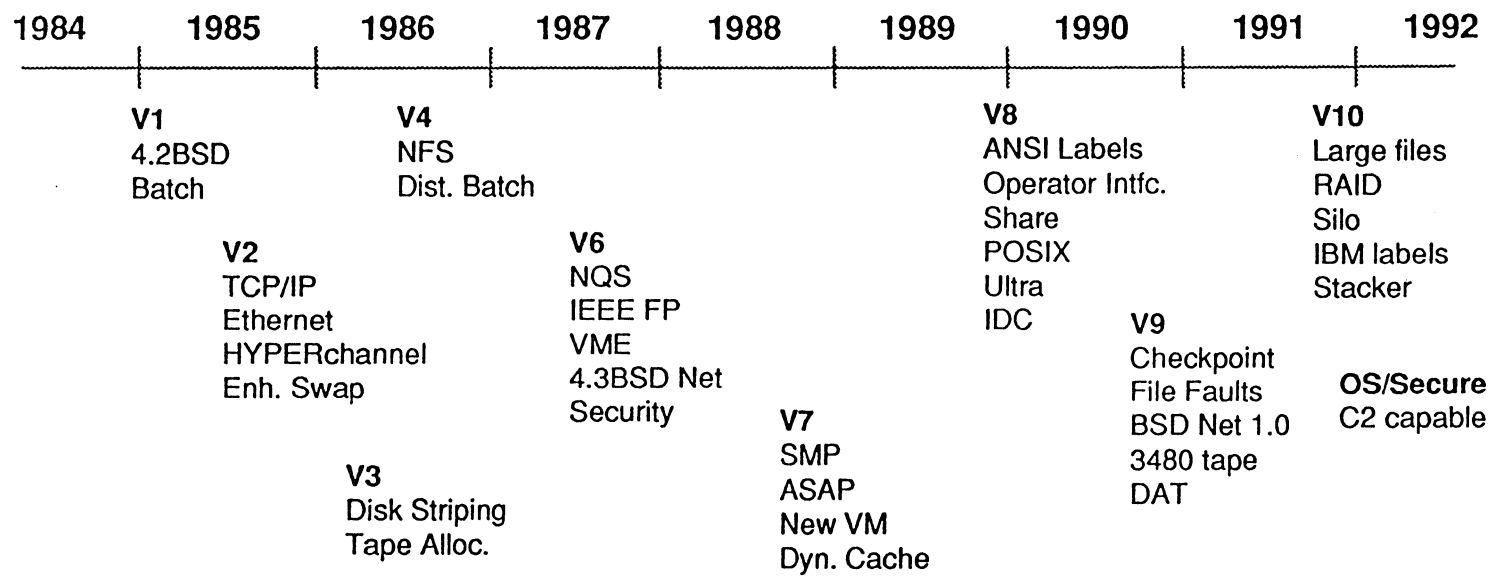
Computing In The '90s



Open Supercomputing



A Brief History of ConvexOS



ConvexOS/Secure

Features – V9.5

- Full C2 security functionality – ACLs, Auditing...
- Supports all ConvexOS layered products
- Supports C2 and C3xx systems

Customer Shipments – Now

Features – V10.0

- Integration of ConvexOS V10.0 features
- Formal evaluation completion

Customer Shipments – 2Q92

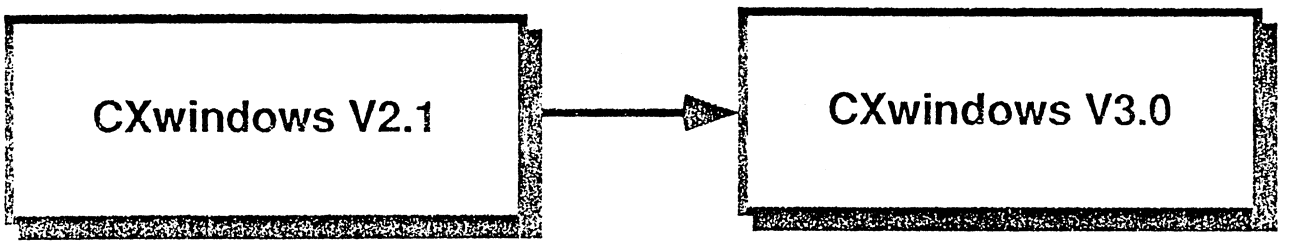
CONVEX OS V10.0

Features

- C3800 and C3400 support
- Virtual Volume Manager
- Large files enhancements
- Faster fsck
- Initial I/O restructuring enhancements
- HYPERchannel enhancements
- Tape enhancements
- Shipped using new GIP installation procedures

Customer Shipment – 1Q92

Software Environments Overview
Interactive Technologies



ANSI C compliance

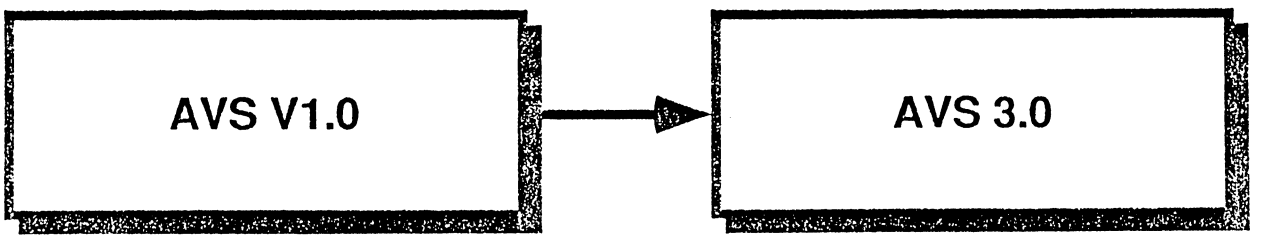
X11R5 toolkit and Xlib library
PEX 5.0 protocol support
Fortran and C PHIGS bindings
POSIX compliance

Customer Shipments – 1Q92

Color X window terminal – available now

PEX interoperability booth at Siggraph '91
PEX interoperability center at CONVEX

Software Environments Overview
Visualization Applications



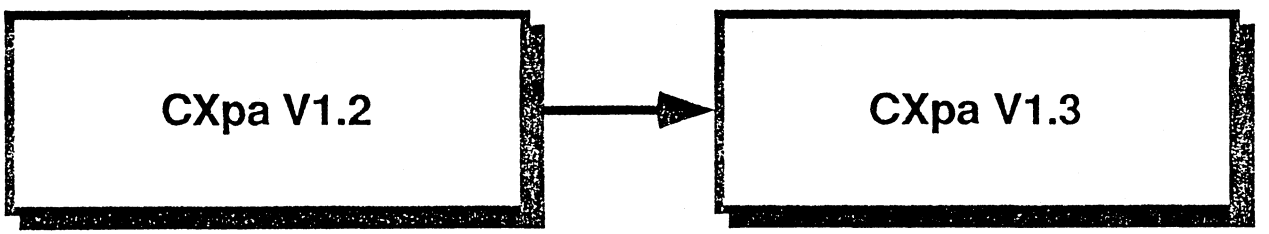
Scientific Visualization
SGI GL renderers

GL, PEX Support
All Stardent AVS 3.0 functionality
Some Stardent AVS 3.5 functionality
Animation facility
Remote module execution

Customer Shipments – 1Q92

International AVS Center Opening
SiGGRAPH '91
Visualization '91 Conference

Software Environments Overview
Development Tools



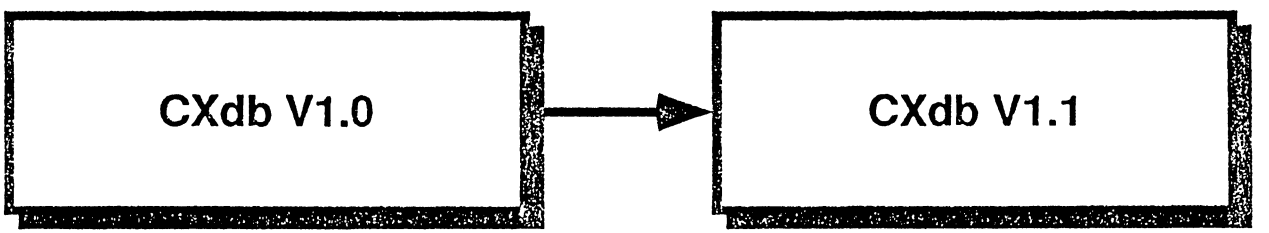
Bug fixes

Handles compiler-generated errors

C3800 and C3400 support

Customer Shipments – Now

Software Environments Overview
Development Tools



Full featured debugger
Level -01 optimization
Motif user interface

Synthesized Variable tracking
Level -02,-03 optimization

Customer Shipments – 4Q91

Los Alamos Debugger workshop
Supercomputing '91

CONVEX Storage Management (CSM) Phase I

Features

- Automatic migration to the STK silo or manual tape mounts
- Transparent to all applications
- Allows online data on file systems with migration enabled to be backed up

Customer Shipment – 2Q92

OSI WAN

Features

- OSI WAN is connection to remote hosts over X.25
- Replaces CONVEX CX.25
- C3200, ConvexOS V9.1 version
- Features
  - Transport class 0/2/4
  - PAD Client side virtual terminal connection
  - UUCP
  - TSL Library (XTI and SUID)
- C3200, Convex OS V9.1

V1.0 Customer Shipment – Now

- C3800 and C3400, ConvexOS V10.0

V1.1 Customer Shipment – 1Q92

I/O Project Status

- HiPPI to Ultra connection
  - Hardware & software design is complete
  - In beta test
  - Production release is scheduled is Q1-1992
- Integrated Tape Channel (ITC)
  - Beta shipments are complete & tests are underway
  - In checkout with the updated R90 tape drive now, this testing will take the remainder of 1991 to complete
  - Production release is scheduled for Q2-1992
- Fiber Distributed Data Interface (FDDI)
  - Currently running VME controller in alpha testing on a 5 node ring
  - On schedule for beta shipment in November & production release Jan. 1992

Recent New Product Releases
Convex I/O

- TLI SILO data path & 4480 tape drive interface, Q4-90
- DAT tape drive, Q4-90
- IDC on-line formatter and verifier software, Q4-90
- Fujitsu 200ips tape drive, Q1-91
- VME Hyperchannel controller, Q1-91
- Seagate Sabre V SMD disk, Q1-91
- TLI to IBM 3480 tape drive interface, Q1-91
- DAT as a system tape drive for software distribution, Q3-91
- Rack Mount 3480 tape stacker, Q3-91
- Low cost ESDI disk drive, Q3-91

Redundancy in Disk Subsystem

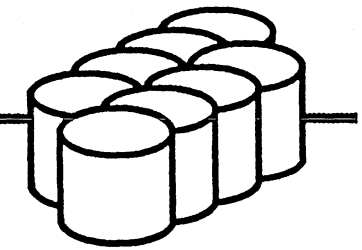
Virtual Volume Manager (VVM) avail with ConvexOS 10.0 !!

Customized Disk Arrays!

- Supports RAID 0 (non-redundant disk striping)
- Supports RAID 1 (mirroring)
- Supports RAID 5 (redundant disk striping)
- "Hot Spares" to automatically rebuild on the fly

CONVEX

Disk Status



New IDC disks now available for Q4!!

Outstanding price/performance

- 50% faster than previous disks !
- 2.5x capacity (formatted) !
- 1/2 the $/MB in cost !
- same form-factor !

CONVEX

Outstanding Performance!

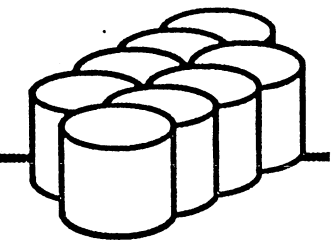
	4-way Previous Stripe Driver	4-way VVM (RAID 0)	4 + 1 VVM (RAID 5)
Read	16.2	18.8	18.7
Write a new file	--	18.4	14.1
Write over existing file	15.1	18.6	16.0

Overhead of adding redundancy still offers better performance than old stripe driver without redundancy!

- Using DKD-502 disks
- Performance in Mbytes/sec

CONVEX

Outstanding Performance!



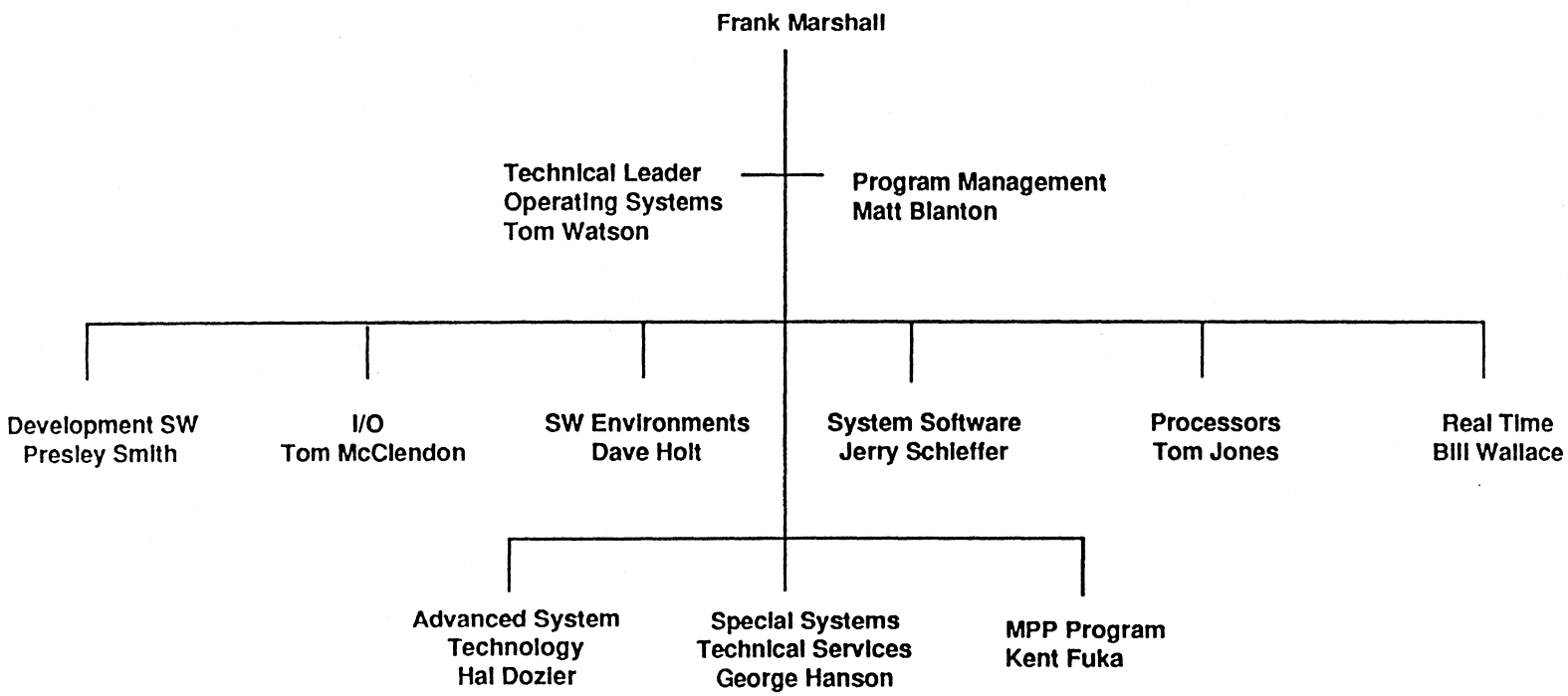
Stripe Width	DKD-504 C3810	DKD-502 C3240
1-way	8.01	5.24
2-way	15.71	10.46
3-way	23.36	15.33
4-way	29.75	16.21
5-way	35.34	
6-way	40.36	

Performance measured is Mbytes/second

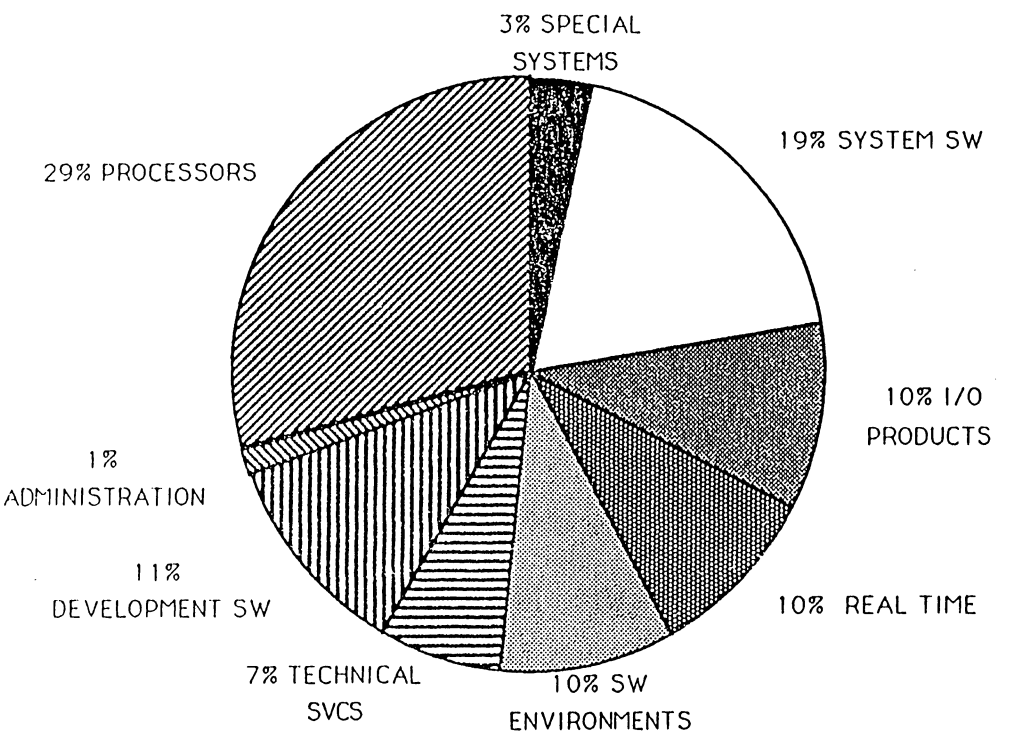
1 IDC can also sustain one DKD-504 per IPI port simultaneously at 8 MB/sec for an aggregate of 32 MB/sec per IDC!!

CONVEX

Organization



DEVELOPMENT HEADCOUNT



Scope of Current Veclib Optimizations

Description	Lines of Assembly Code
Level 1 BLAS	35,400
Level 2 BLAS	29,000
Level 3 BLAS	49,300
FFTs	43,300
Recurrences	19,900
SCILIB	36,600
Total	213,500

Scientific Library Projects

Veclib V7.0

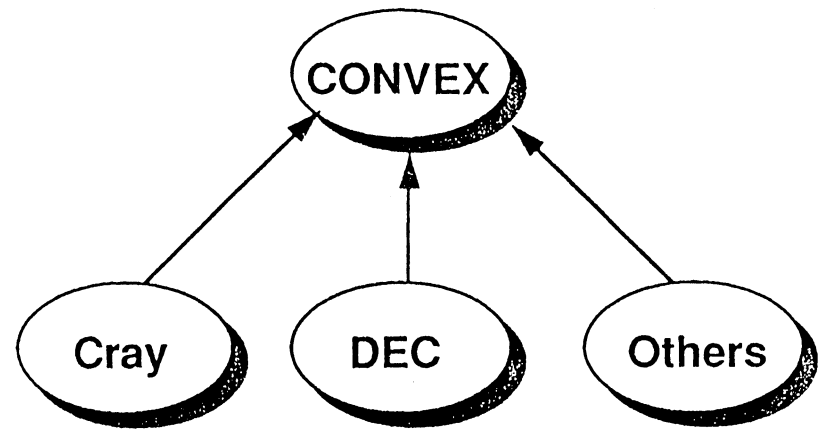
- Adds new Cray compatible SCILIB library to Veclib
  - Compatible with Cray document SR-2081 5.0, March, 1989
  - Same level of vectorization/parallelization as Veclib
  - Works with code produced using the -cfc flag
  - "If it works on a Cray, it works on a CONVEX"
- Adds suite of 1st and 2nd order linear recurrences
- Additional level 2 and 3 BLAS routines parallelized

Customer Shipments – Now

Scientific Libraries - SCILIB

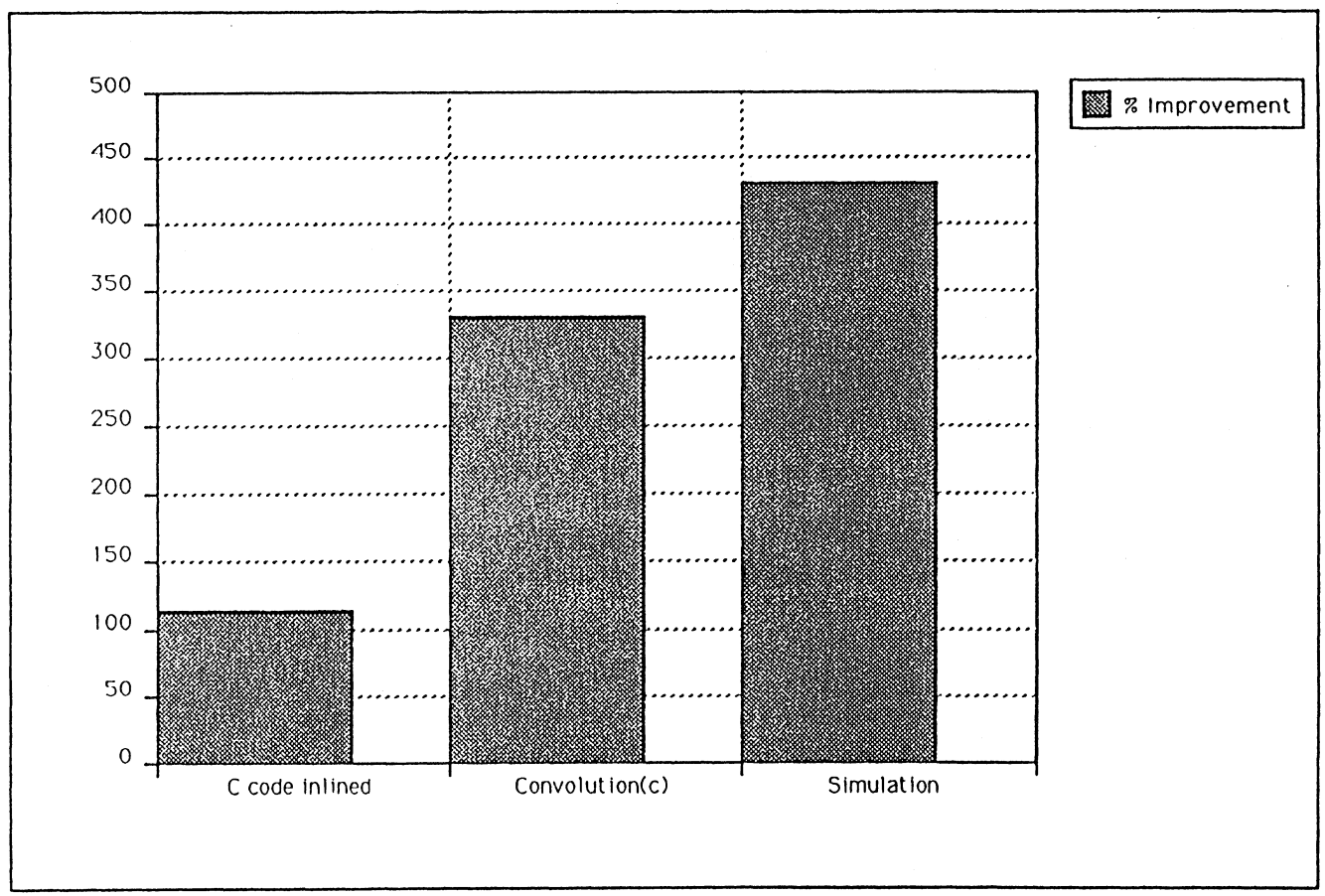
Item	Cray SCILIB	CONVEX VECLIB	CONVEX SCILIB
BLAS 1,2,3	X	X	X
Sparse BLAS	Non standard	X	Cray-Compatible
CONVEX BLAS EXT		X	
Cray BLAS EXT	X		X
LINPACK	X	X	X
EISPACK	X	X	X
Sparse Linear Equ		X	
Sparse Eigenvalues/vec		X	
1-d FFT	X	X	Cray-Compatible
2/3-d FFT		X	
Convolution/correlation	X	X	Cray-Compatible

Compatibility Story

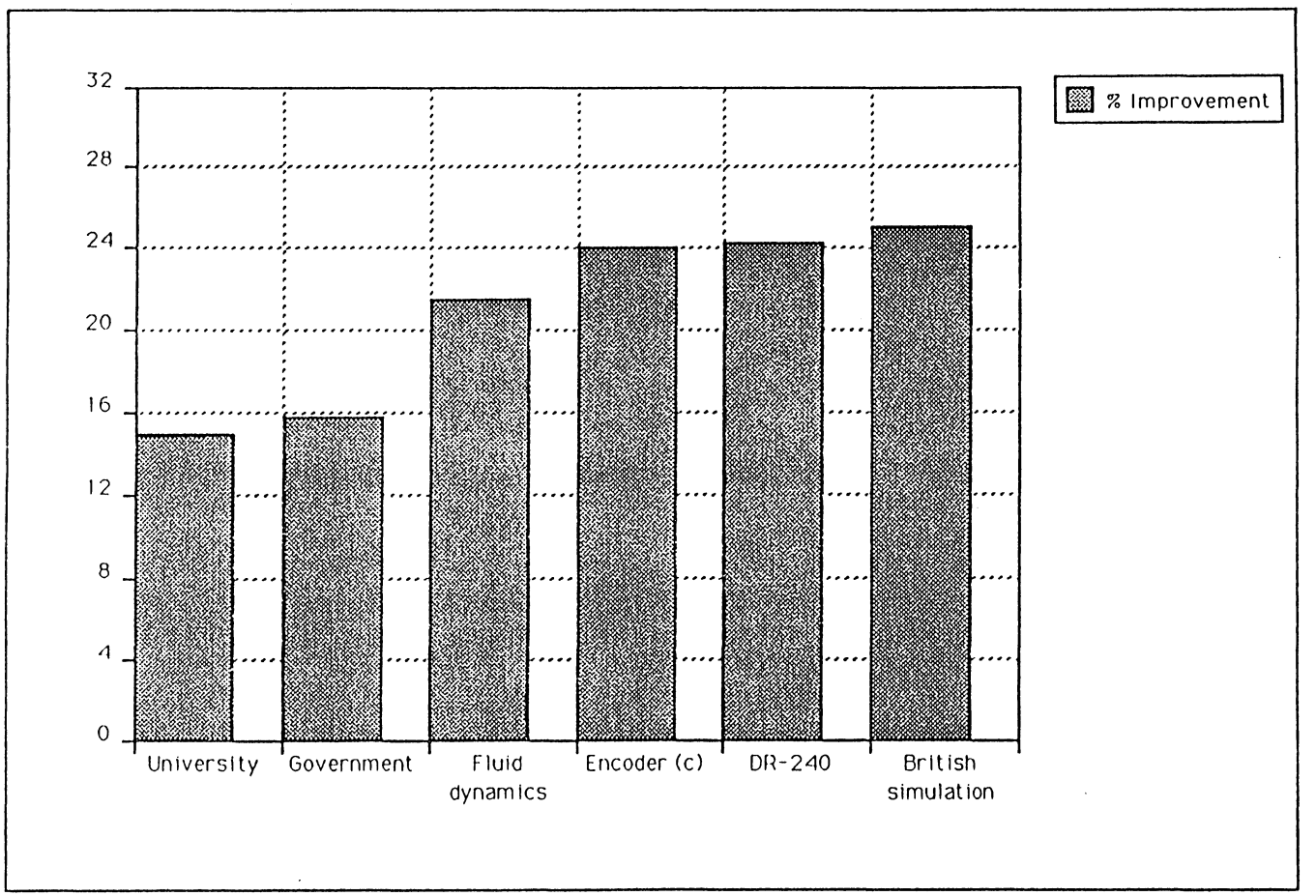


Common Customer			
Source Code	Pointers Bufferin/out	Structures Namelist I/O	Sun IBM
Data	Cray Binary I/O	DEC Binary I/O	Generalized Binary I/O
Libraries	SCILIB	COVUELIB	LINPACK (VECLIB)

Exceptional Benchmark Results



Typical Benchmark Results



Application Compiler Projects

V1.1

- Library archive support
- Use of profile data in optimization
- Dimension extension to reduce memory bank conflicts
- Enhancements to procedure cloning and inlining
- Link line reorganization to improve instruction cache performance
- Faster pointer tracking algorithm to reduce compile time
- Additional directives and options
- New error test capability to help find user errors

Customer Shipments – 4Q91

Ada Projects

- Bug fix releases as required
- Next government certification has been moved to 1993
- Derived validations will be obtained for C3800 and C3400

CONVEX C Projects

Direction is improved application performance.

V4.2 (C3800 and C3400 only release)

- C3800 and C3400 optimizations
- Revised chaining rules
- Improved stack and memory alignment

V4.3

- All cc V4.2 enhancements
- Compiler performance improvements
- IF/FOR interchange
- Enhanced register allocation
- Enhanced scalar code generation

Customer Shipments – 1Q92

FORTRAN Projects – fc V7.0

Direction is "Toward Fortran 90" – fc V7.0

- Fortran 90 array notation
- Array intrinsics
- Automatic arrays
- Allocatable arrays

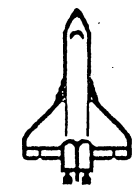
Miscellaneous Enhancements – fc V7.0

- Cray Features
  - TASK COMMON
  - UNBLOCK utility for blocked data files
  - More Cray constant forms supported
  - Adds '*' as hollerith delimiter in formats
- New Cross Reference facility

Customer Shipments – 4Q90

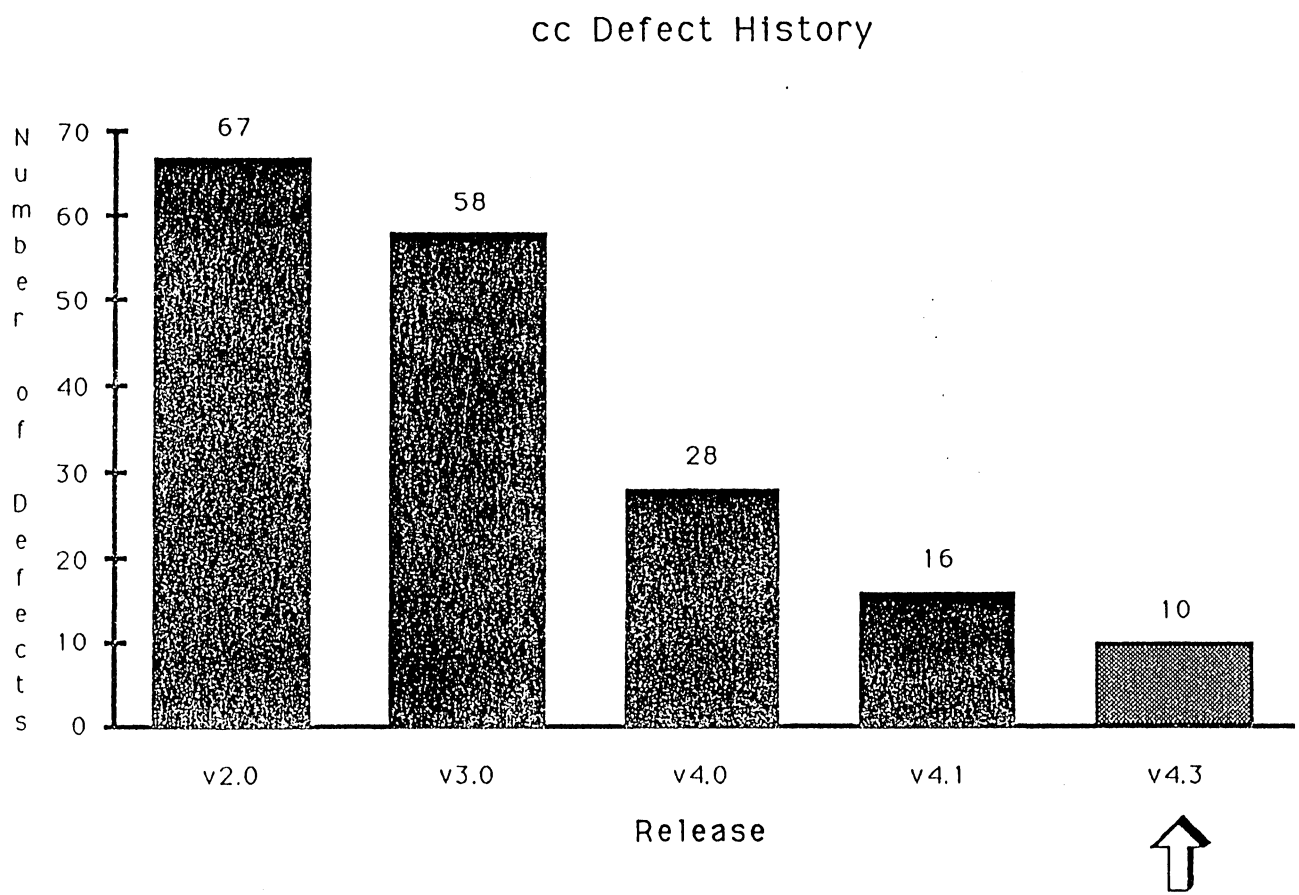
FORTRAN Projects – fc V7.0

Direction is PERFORMANCE!

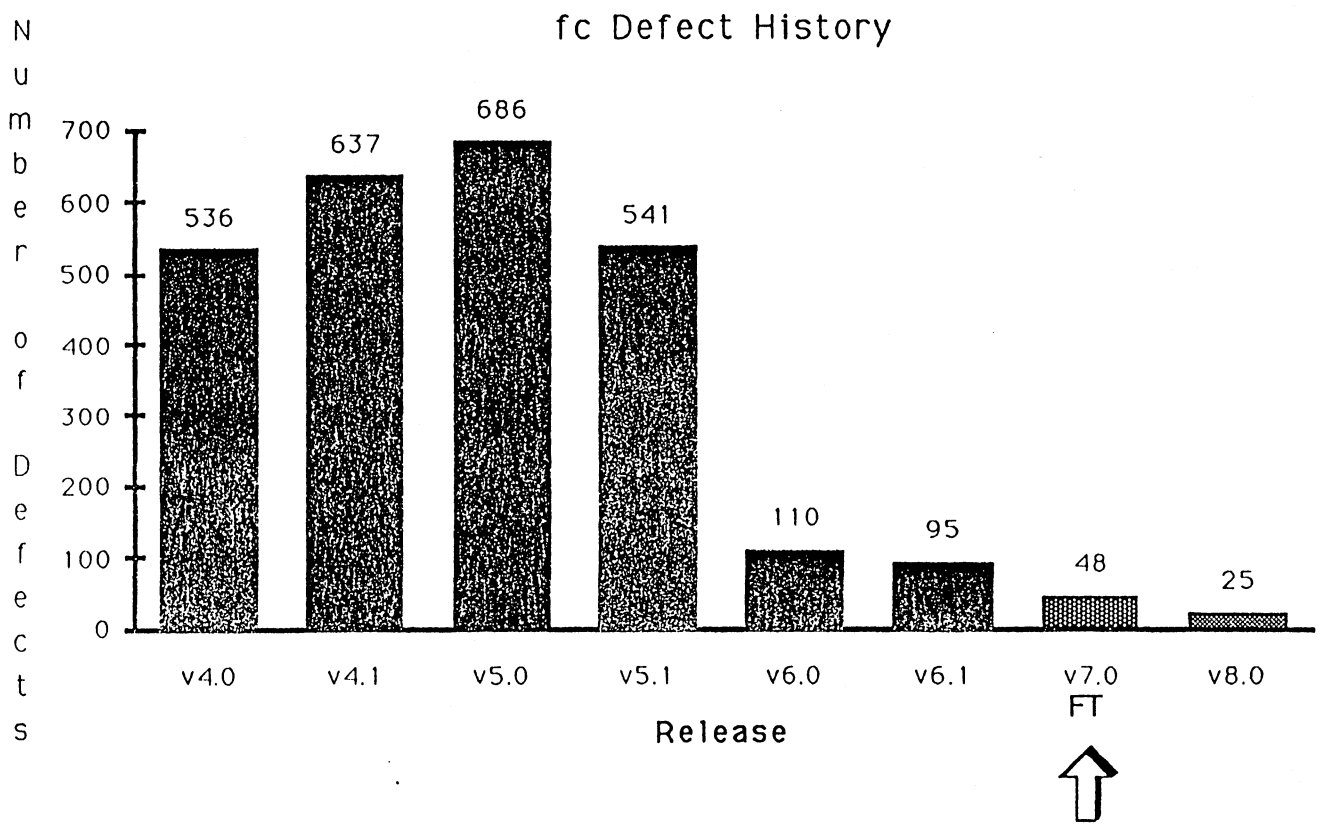


- Improved optimization
  - IF-DO interchange
  - Short circuit IF's
  - Improved dynamic selection
  - Improved loop interchanges
  - Many others
- HOTSHOT - customer driven performance improvements

Enhancement	% Improvement
Vector log function	15
ATAN function	15-50
Double precision power function	220
Single precision power function	280
I/O implied do loops	1000

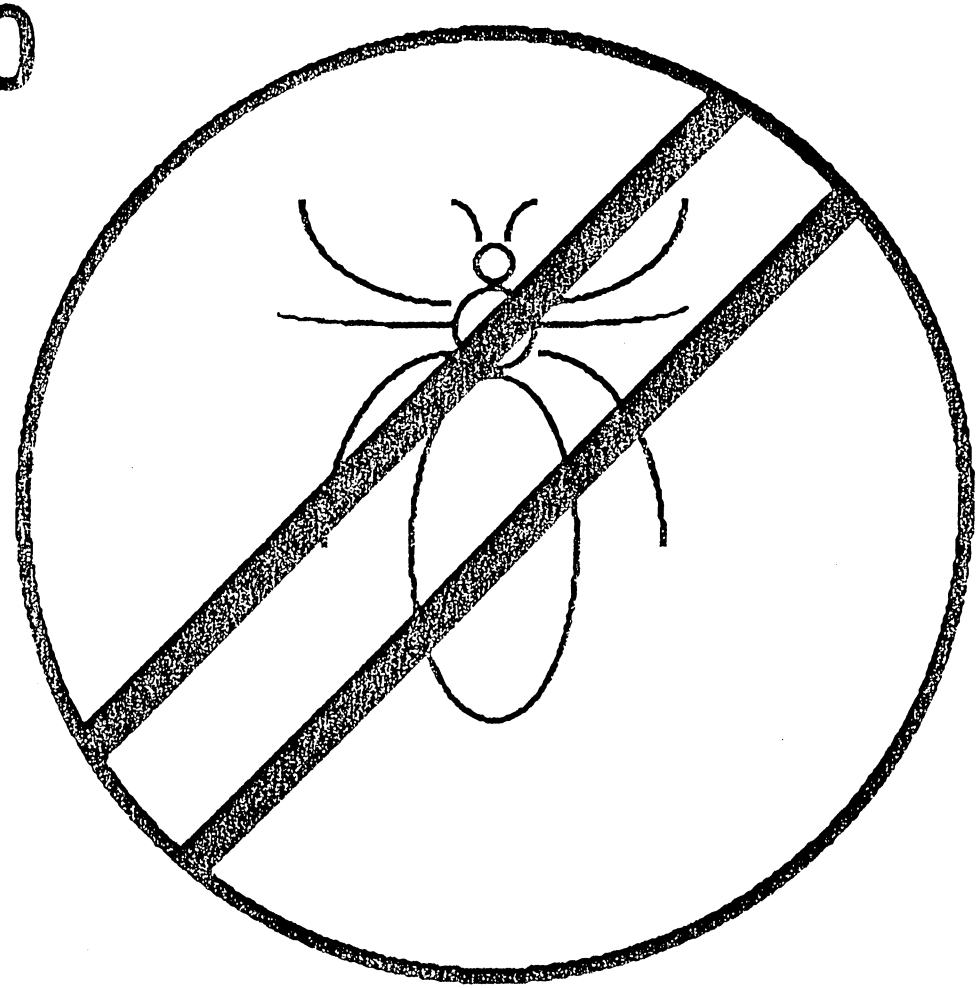

cc Defect History
Number of Defects
70
60
50
40
30
20
10
0
67
58
28
16
10
v2.0
v3.0
v4.0
v4.1
v4.3
Release

CONVEX FORTRAN



DEVELOPMENT SOFTWARE

1990



DEVELOPMENT SOFTWARE

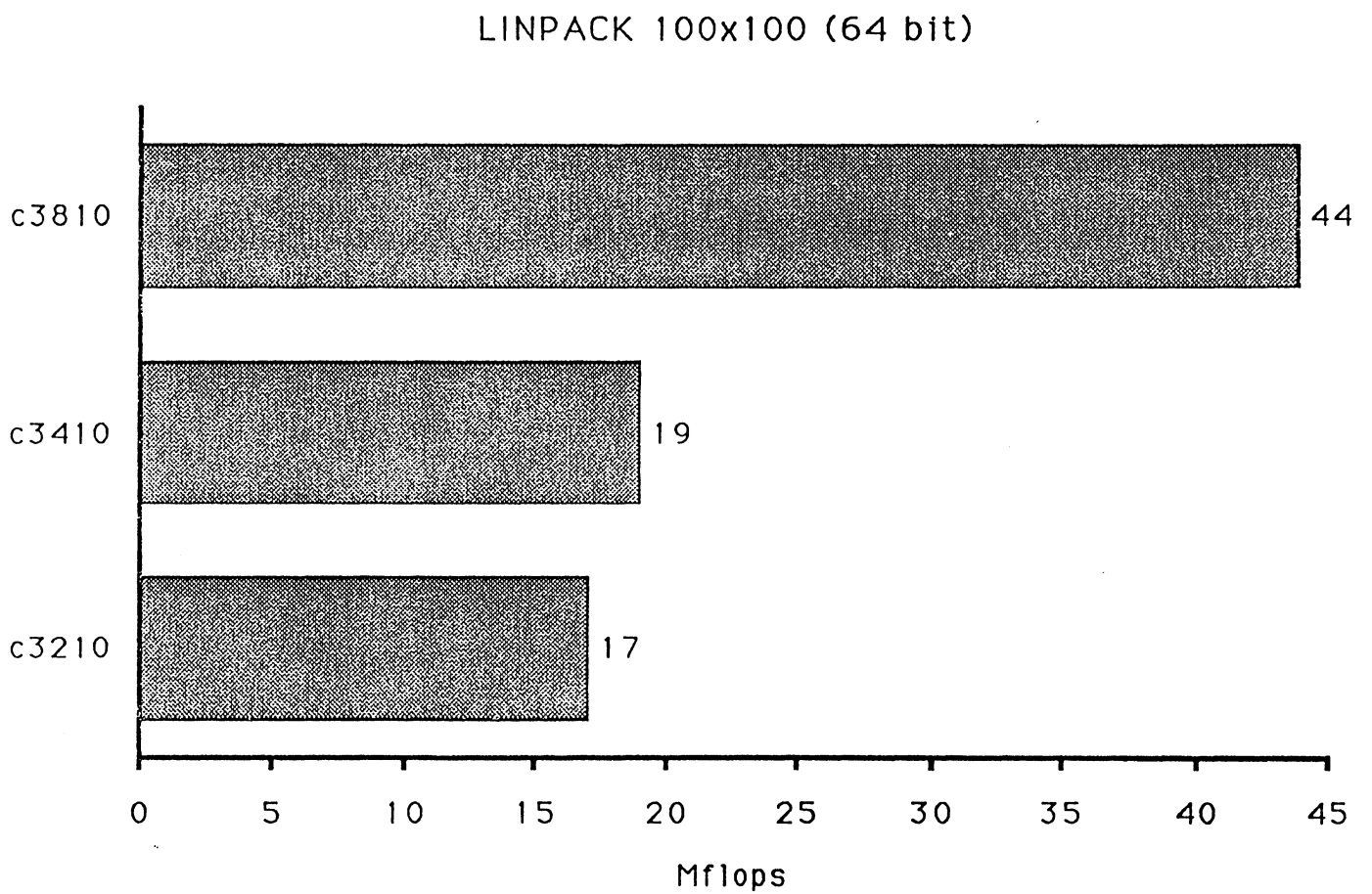
1991

Performance

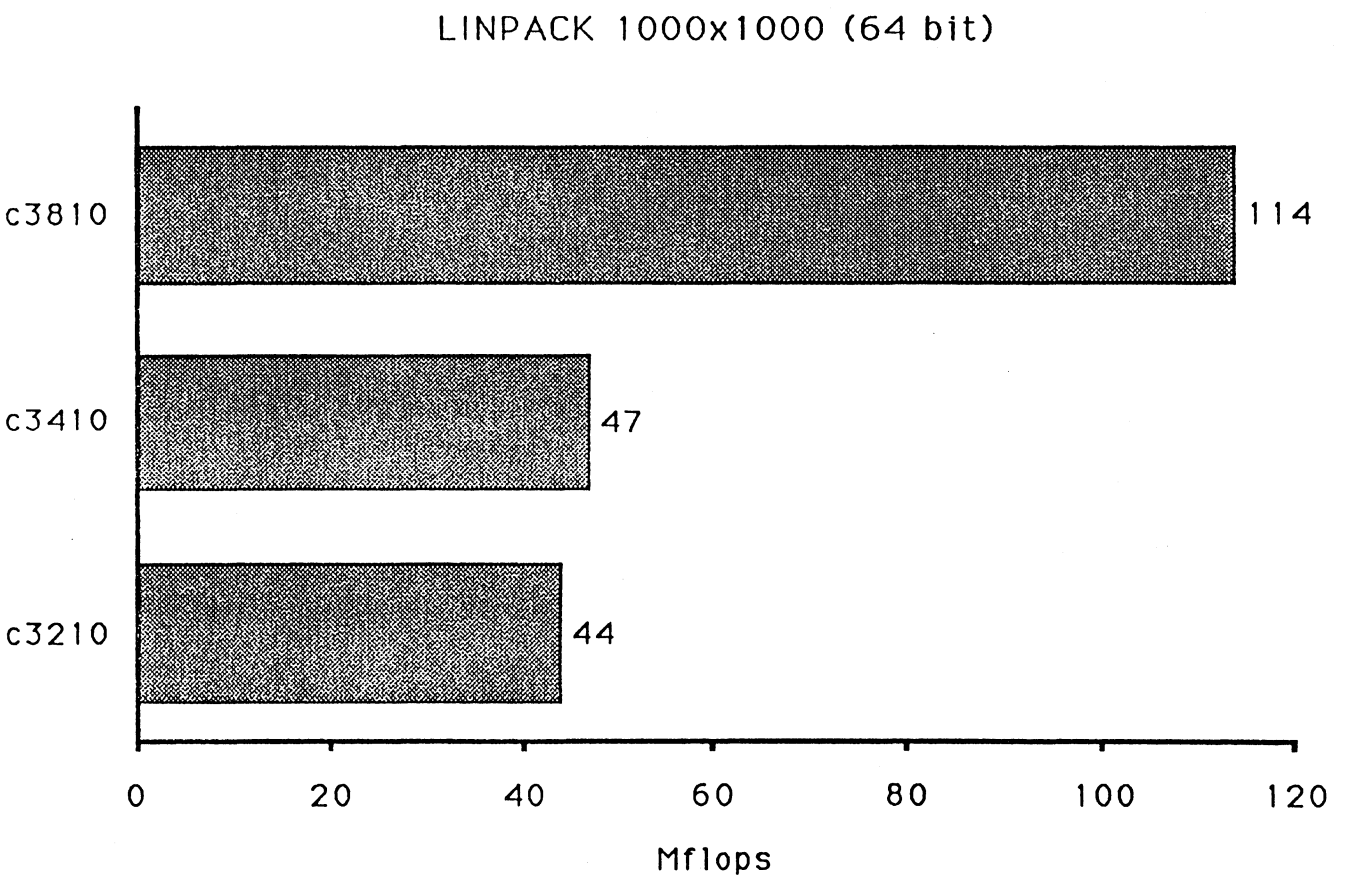
"A"

Us

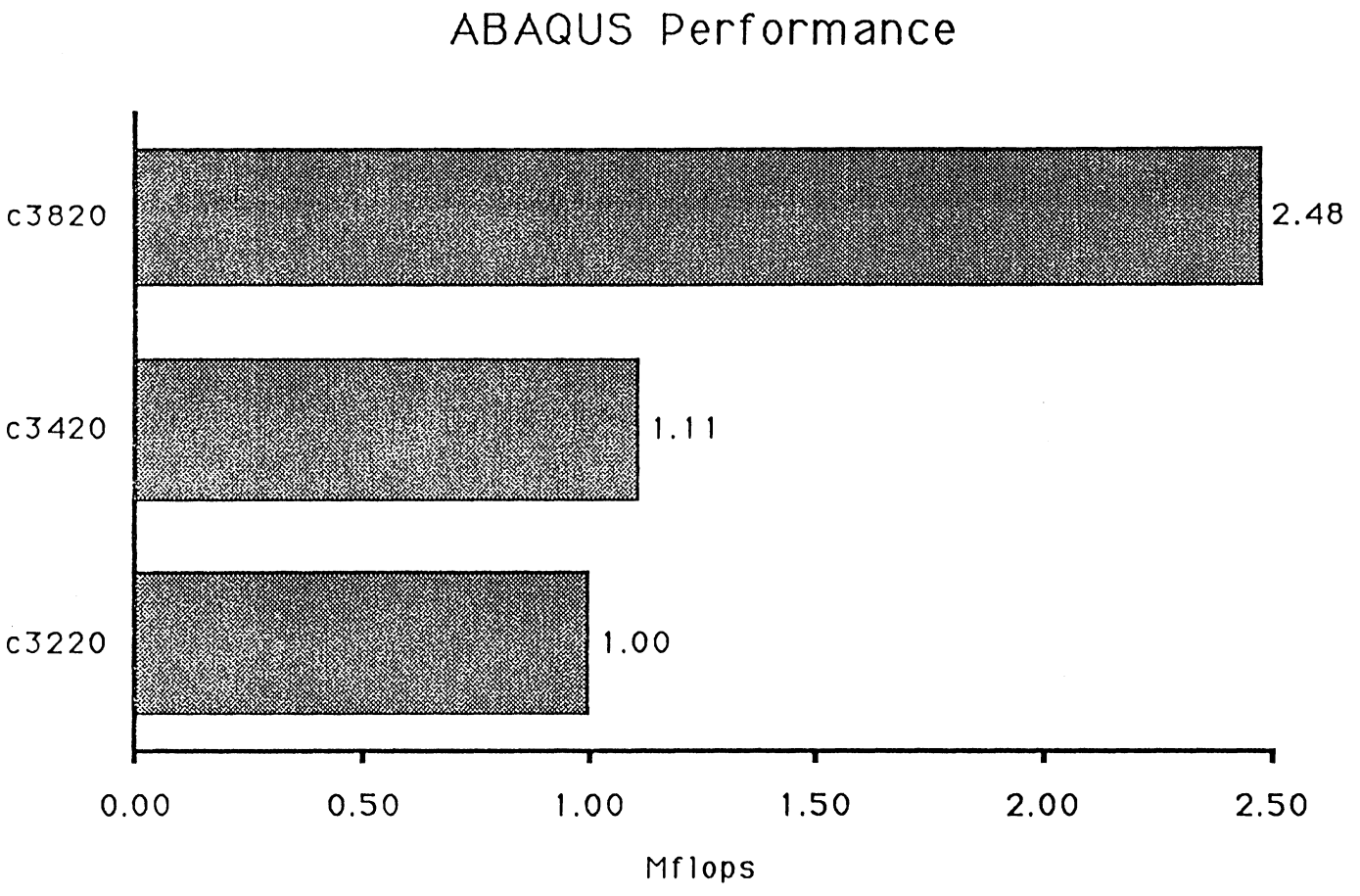
Benchmarks - LINPACK



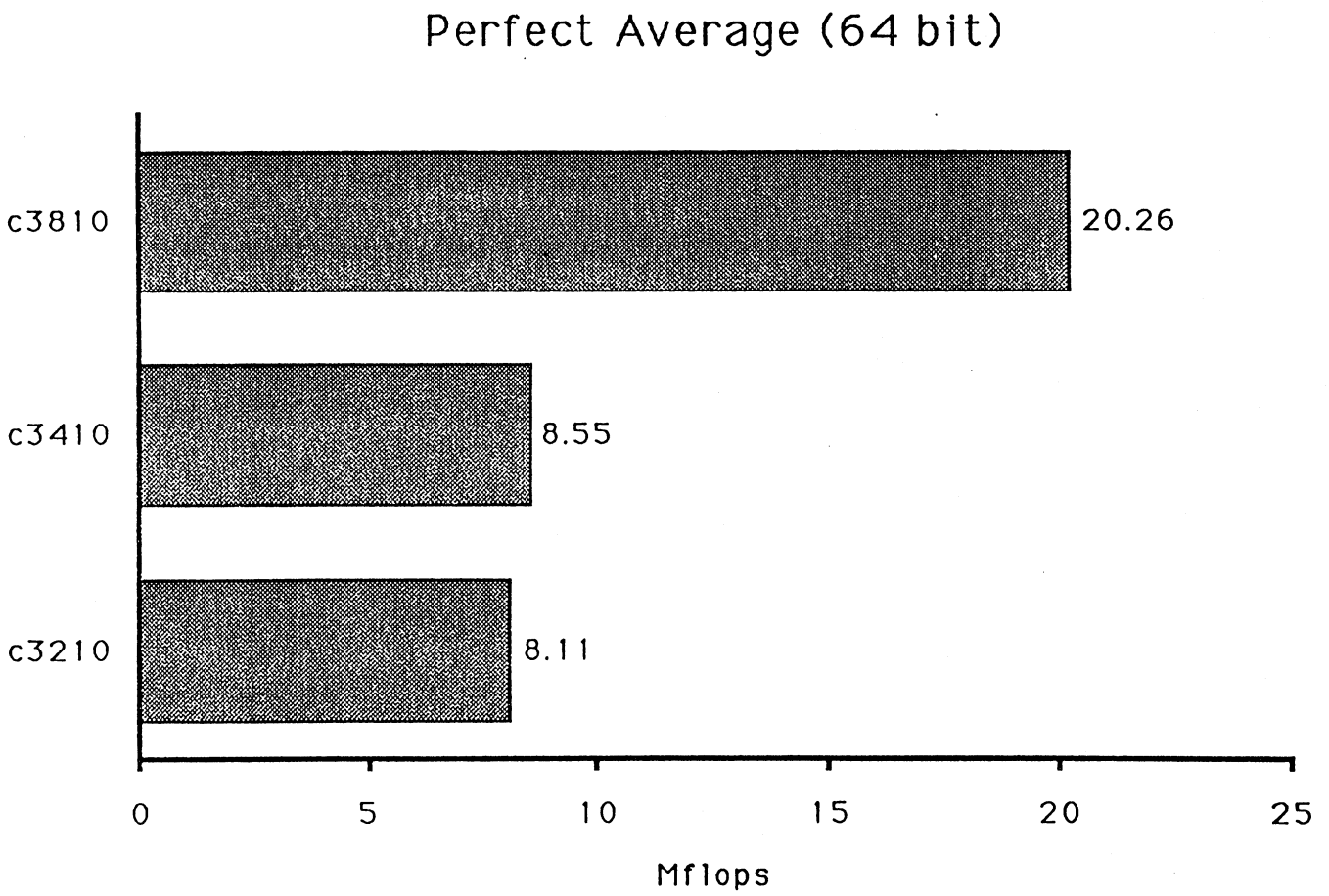
Benchmarks - LINPACK



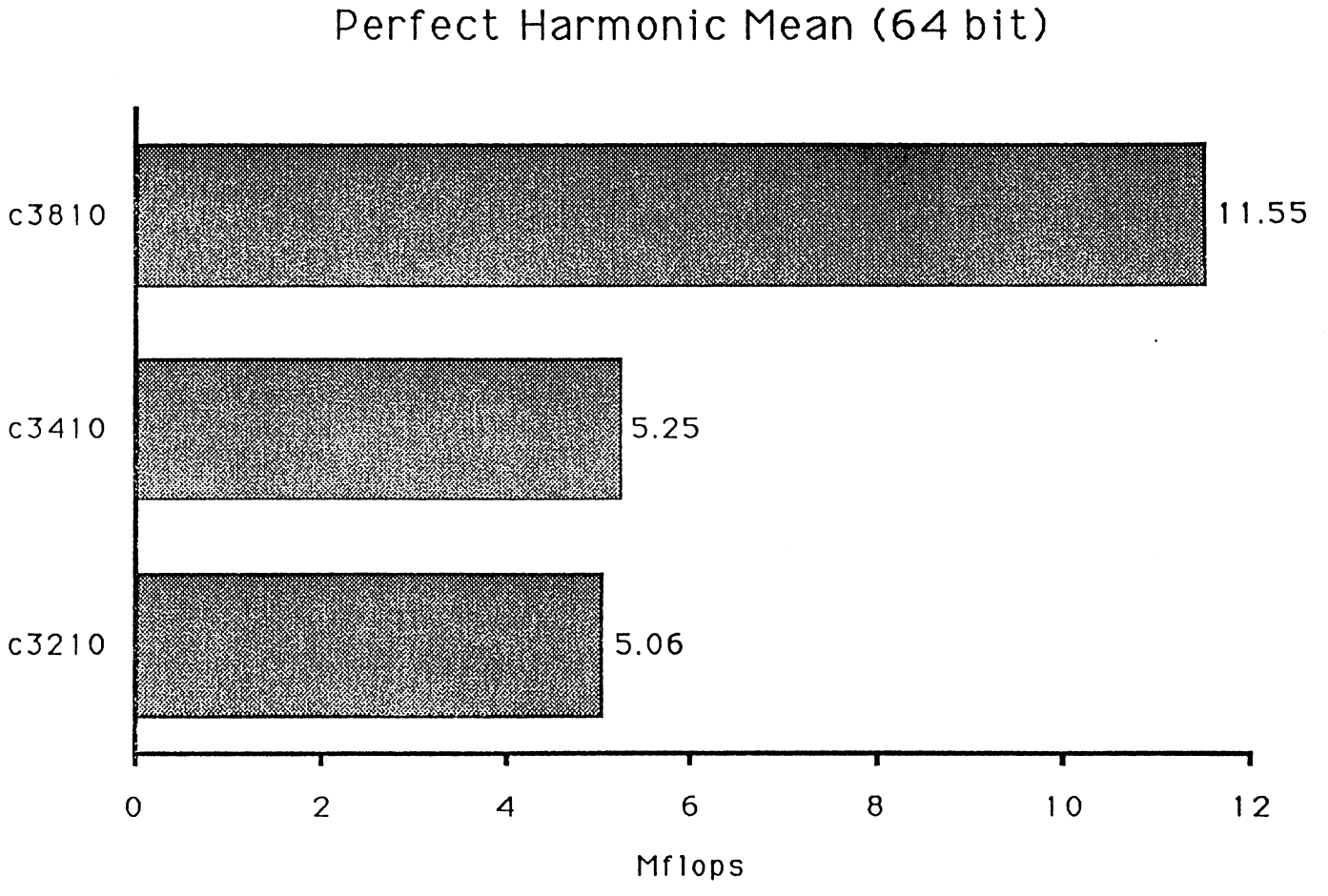
Benchmarks - ABAQUS



Benchmarks - Perfect Club



Benchmarks - Perfect Club



Key New Product Thrust for 1991

Processors - Tom Jones

C3400, C3800

I/O Channels, Peripherals, Controllers - Tom McClendon

HiPPi, TLI, ITC

System Software - Jerry Schieffer

Tape Management, Storage Management,Trusted O/S, Networking

Visualization & Graphical Interfaces -Dave Holt

AVS, PEX, CXdb (new visual debugger)

Development Software -Presley Smith

Fortran, C, Ada, Applications Compiler, Veclib

Realtime - Bill Wallace

New Realtime O/S

Special Systems - George Hanson

All VME systems, HDTV et al.

MPP - Kent Fuka

Business Plan, Funding Effort, Initial Study

CONVEX C3800

- 2 systems shipped in Q3
- Two processor 3820 with 512MB of memory
- Single processor 3810 with 4GB of memory
- 4GB of memory is a new achievement for supercomputer applications
- Four Processor 3840 shipments in Q4 1991
- Eight Processor 3880 shipments in 1H 1992

- The system level design is SOLID !!
- Performance expectations are being meet by the 3800 systems
- Material availability has been our biggest challenge

CONVEX C3400

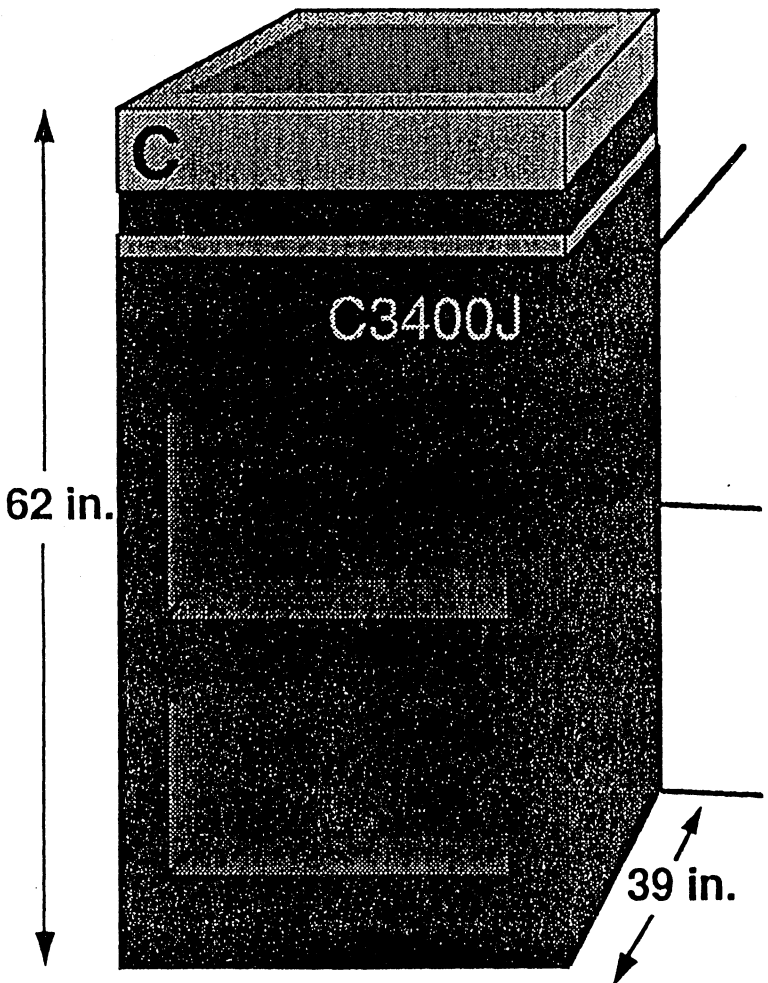
- 8 systems shipped in Q3
- Single and Dual processor configurations shipped
- Shipments of 3440 in Q4 of 1991
- Shipments of 3480 in 1H of 1992
- 20 systems are planned to be shipped in Q4

Introducing the Convex 3400J

The "All-in-One" C3400 !!

- 40% of the floor space of a similar C3400 !
- 100 MFLOPS Peak !
- Up to 512 Mbytes memory !
- Single-phase power !

C3400J Packaging



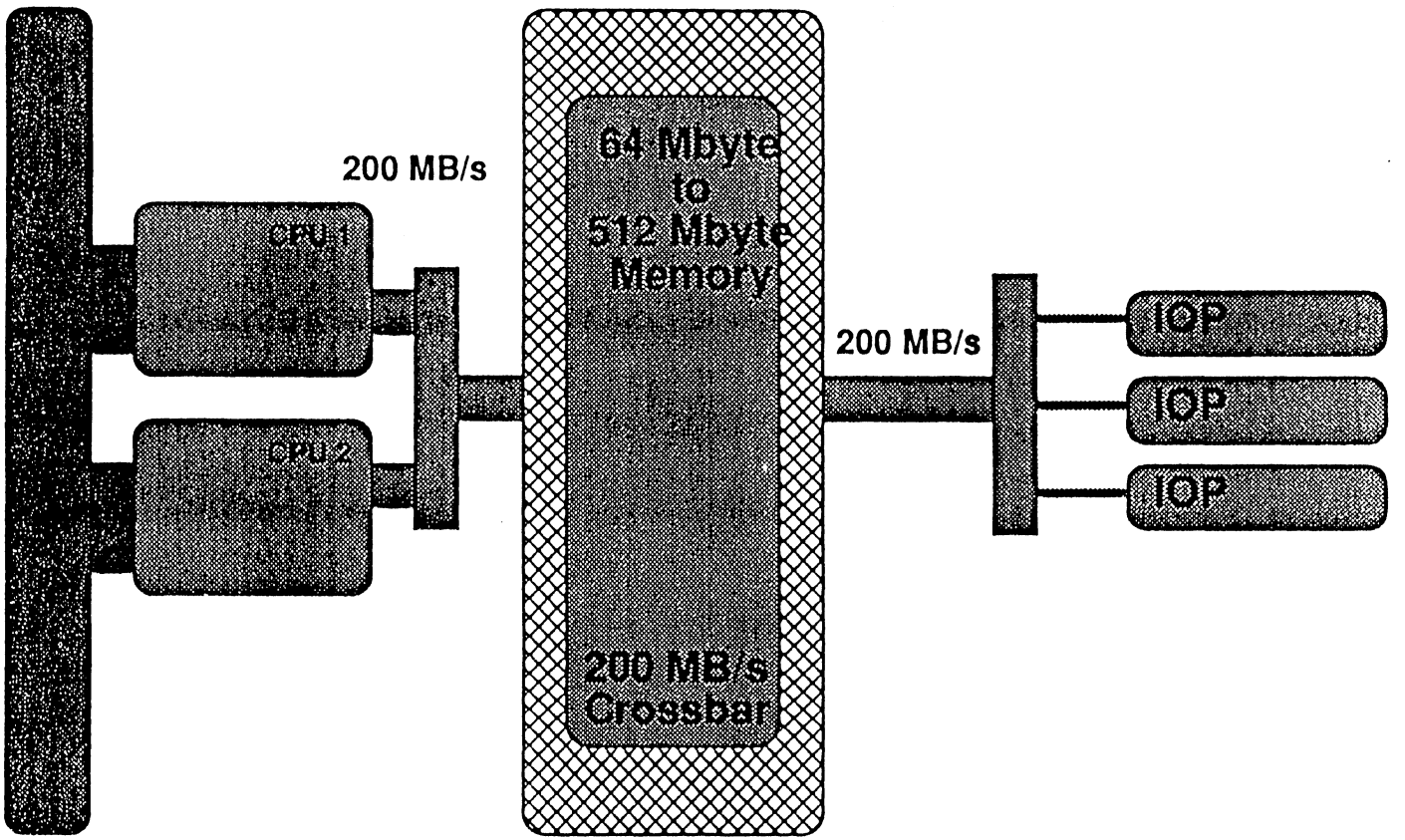
CPU Chassis
- 1 utilities card
- 1 PIA card
- 1 SPU card
- 1-2 CPUs
- 2 MCMs
- 64-512 Mbytes
- 1-3 IOPs

System Chassis
- 0-8 760MB disk drives (Hitachi DK515)
- 1 4mm DAT system tape
- 1 Service processor (includes SPU disk & tape)

VME Chassis
- 1-2 VBCU cards
- 1-8 controller cards

CONVEX

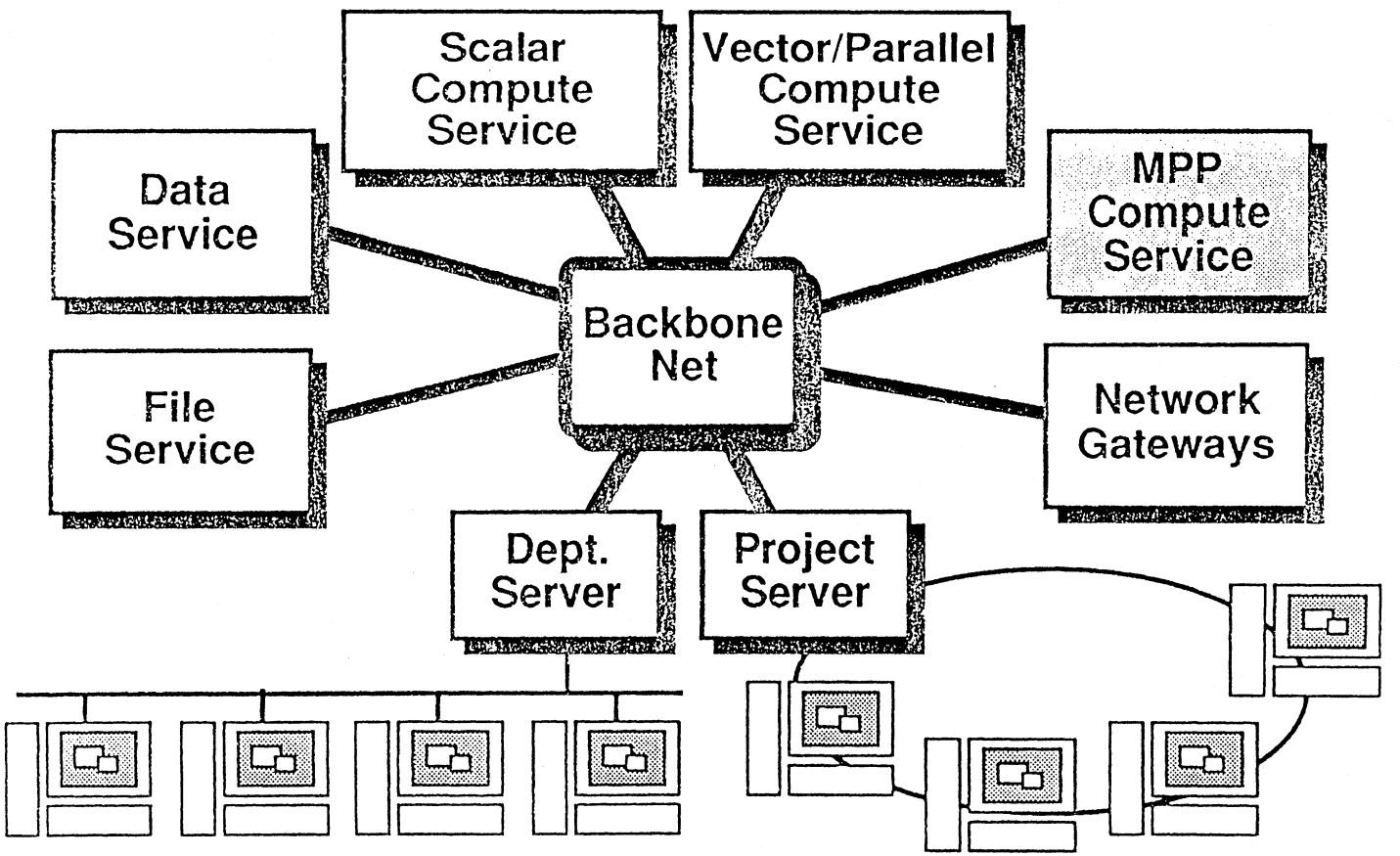
C3400J Architecture Overview



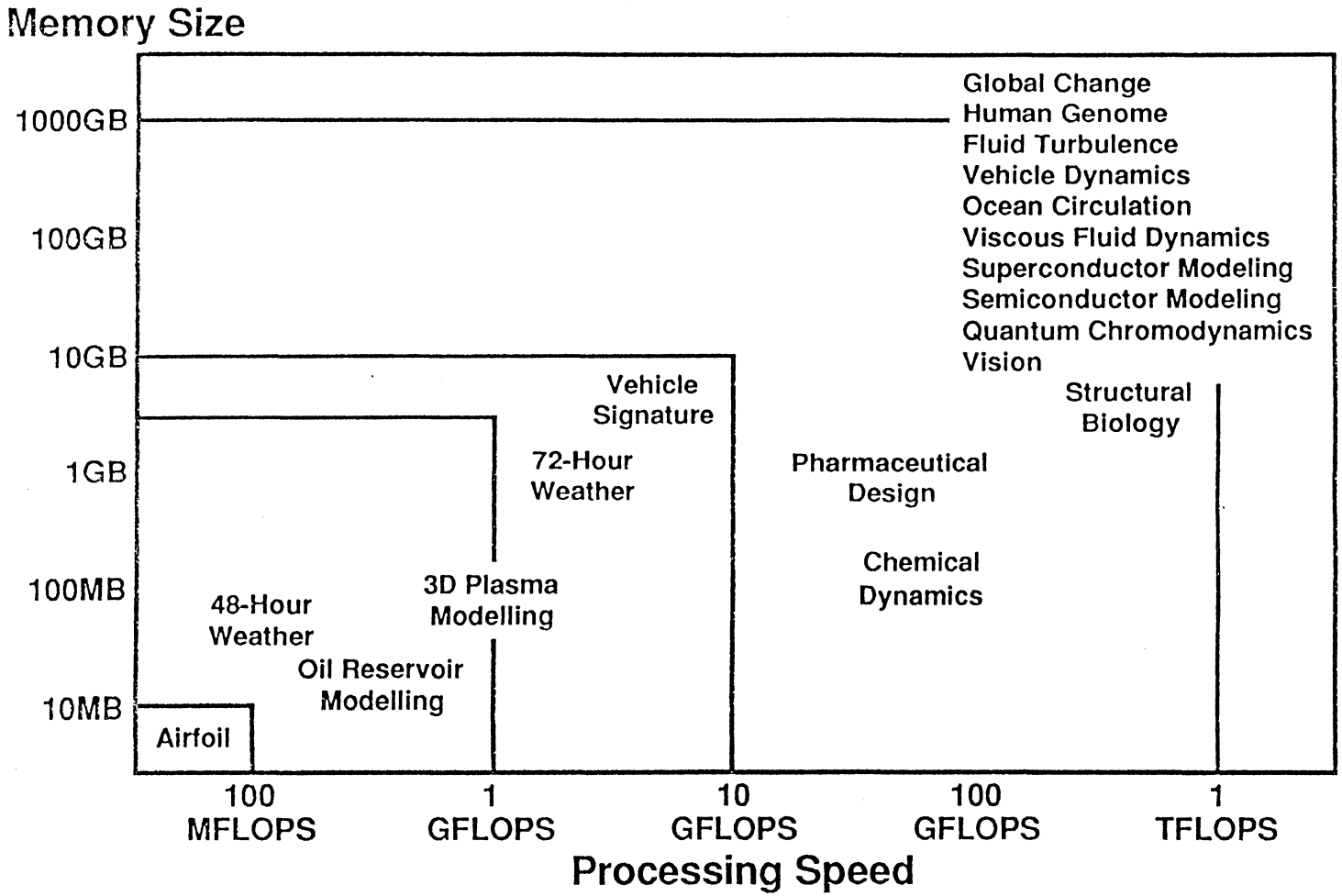
Processors Memory I/O Subsystem

CONVEX

Shared Services



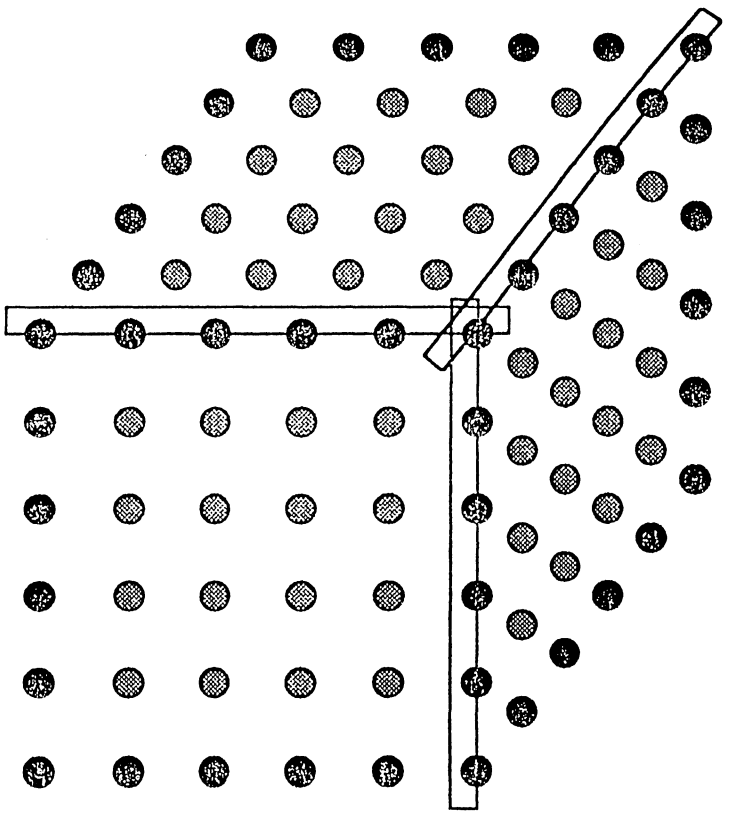
The Future – Grand Challenges



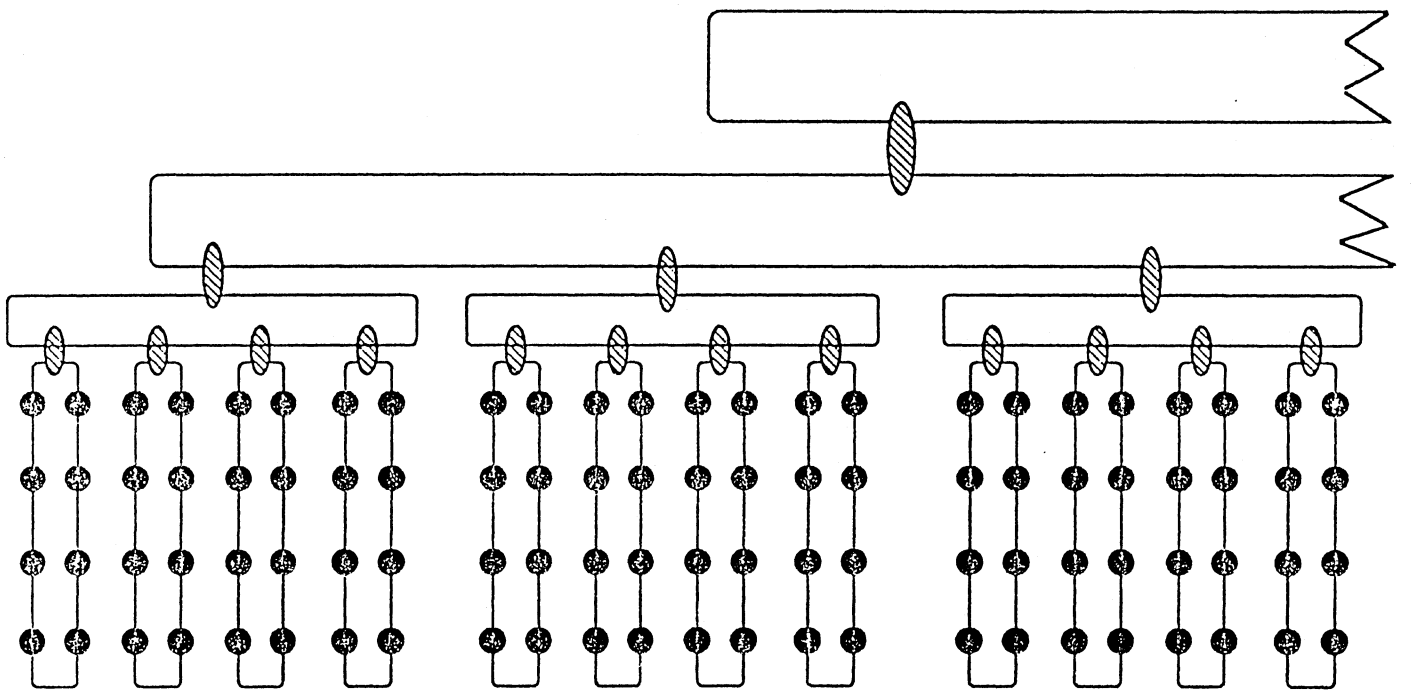
CONVEX MPP Goals

- Close integration to CONVEX system
- Efficiency for existing F77 and C applications
- Scalable architecture for growth to teraflop performance
- Truly easy-to-use software
- Outstanding quality

3D Mesh Interconnect



Tree Hierarchy



A Scalable Operating System

- Must have scalable file system
  - File system must be distributed
  - Disks can be added as CPUs are added
  - Efficiency of routing and access methods is key
- Must support high speed networks
  - Must be reconfigurable as nodes are added
  - Interoperability is key
- Must provide a coherent programming model
- Must provide user friendly tools
  - Performance analysis
  - Debugging

MPP Automatic Compilation

- Interprocedural Analysis is the key compiler technology for MPP
- CONVEX Application Compiler is the basis for MPP compilers of the future
  - Array section analysis, in the current release, is the basis for efficient data distribution
  - Interprocedural analysis is required for safe, efficient, automatic application parallelization
- Automatic and efficient data distribution
  - Key to good application performance
  - Key to ease in porting of applications
- CONVEX is the leader in Interprocedural Technology today

Unique CONVEX Attributes

- Scalable hardware architecture
- Scalable Operating System Attributes
- Automatic compilation and parallel decomposition
- Easy-to-use tools
- Product quality
- Customer service

Problems With Current MPP Systems: 1992

- Difficult to program
- I/O connection is a bottleneck
- Difficult to achieve application performance

MPP Summary

- World's Best at Easy-to-use MPP
- Most User-Friendly and Transparent
- Best Acceleration of Existing Codes
- Best Development Tools
- Best Hardware Integration
- Best Product Quality and Reliability

Real Time Products

Target Market Segments

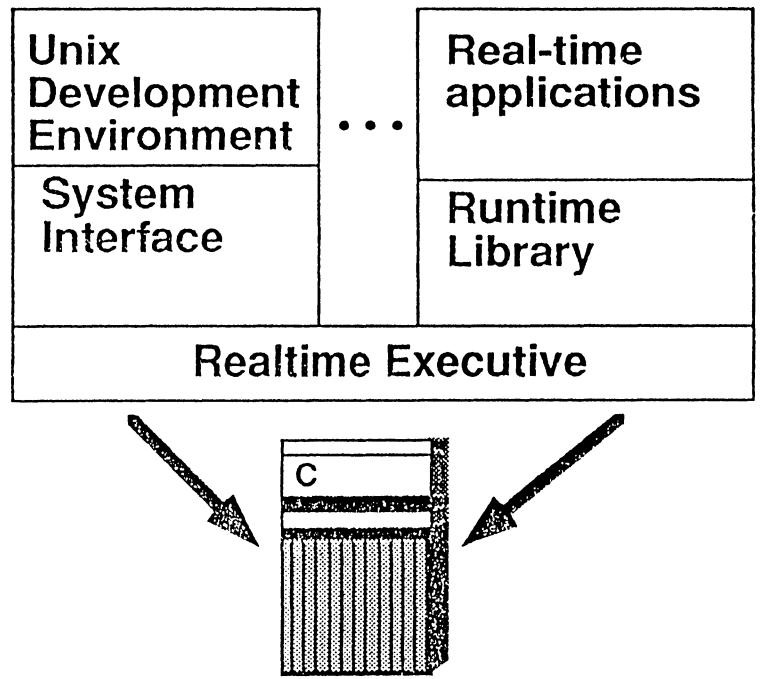
Data Acquisition and Analysis Systems

- High Data Volume in the application
- High Bandwidth required
- Usually needs Large Memory Configurations
- Pre and Post processing desired
- Can use Vector Capabilities & strengths in multiprocessing

Large Realtime Simulators

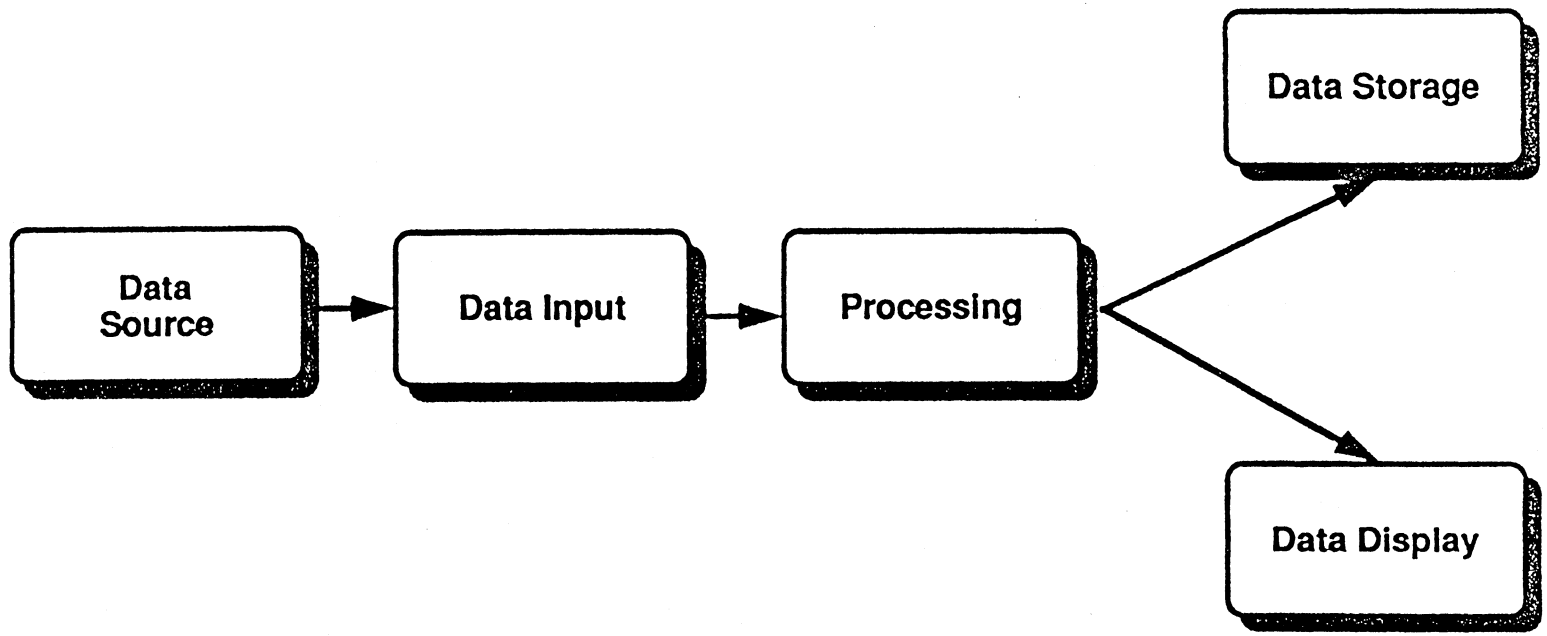
- One of a kind or Limited Volume systems
- Customer is less sensitive to price
- Expansion capabilities are important
- Used for early evaluation and analysis
- Rapid reconfiguration and constant changes in software
- No desire to optimize for smaller systems

Convex Realtime Products



Data Acquisition Model

ConvexRTS™

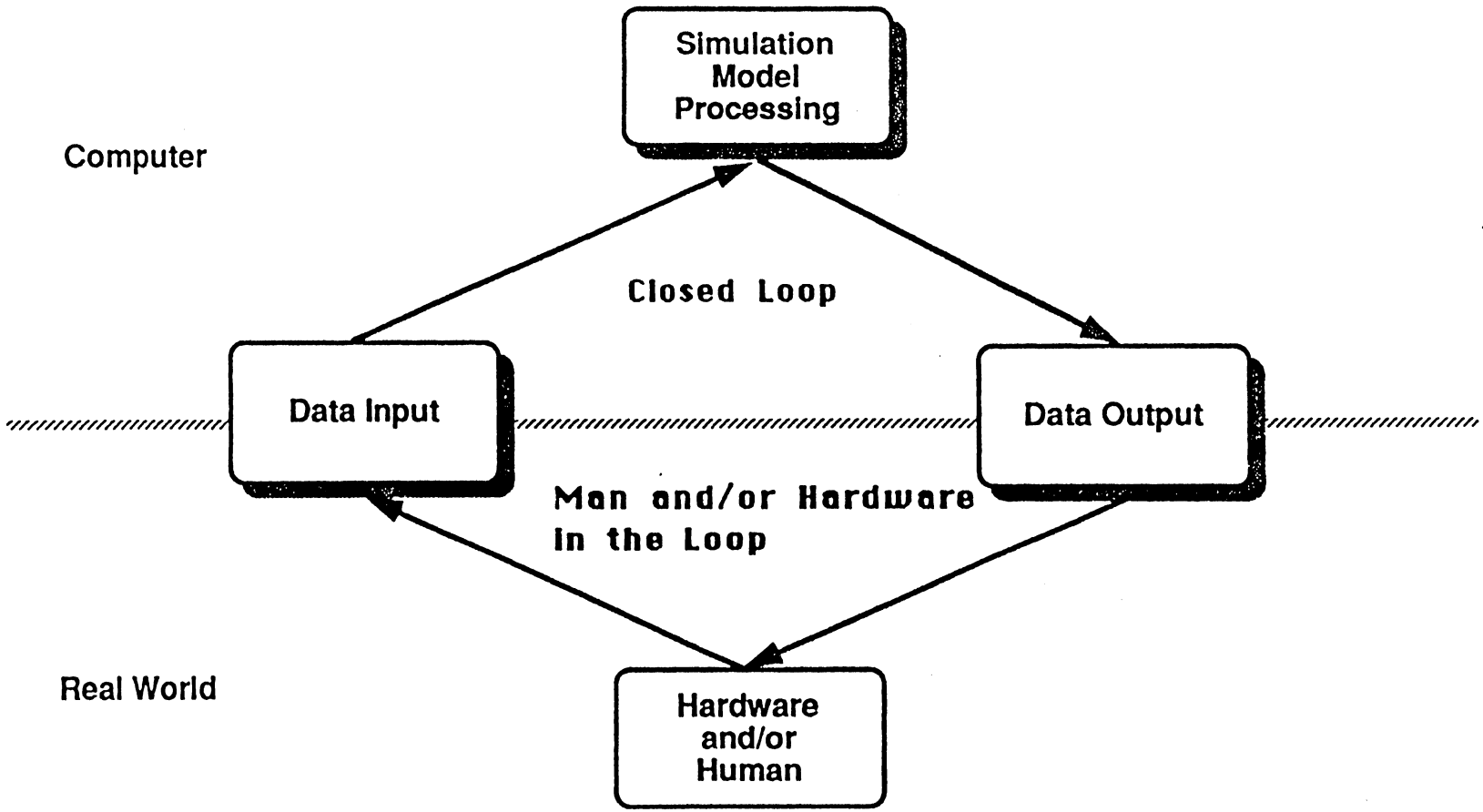


Open Loop

Pipelined Processing of Continuous Data

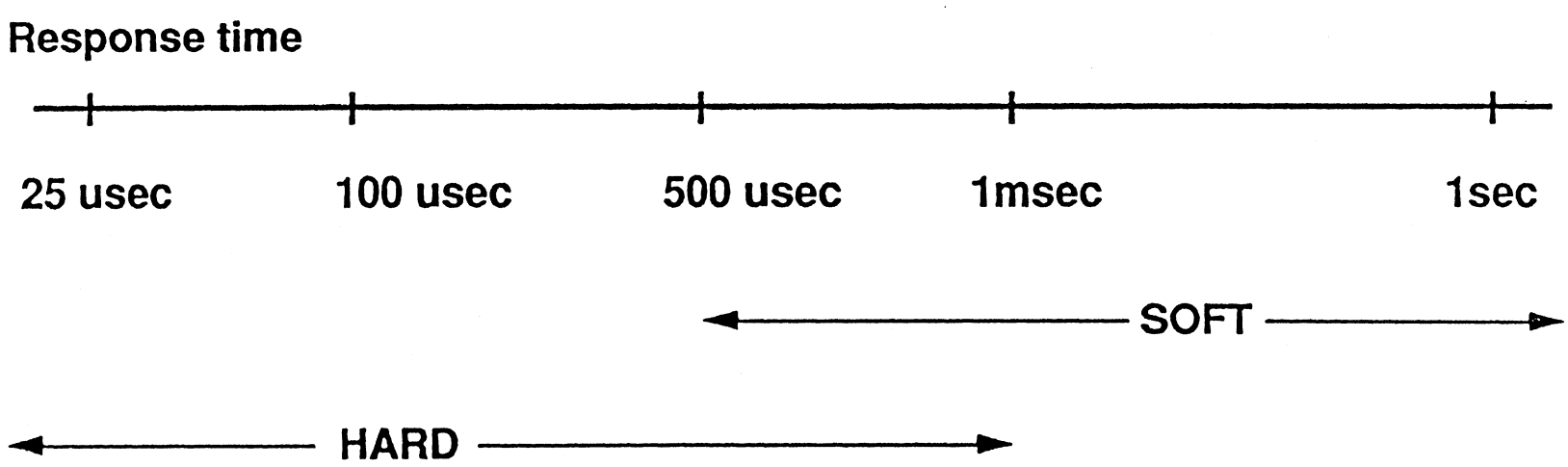
Simulation Model

ConvexRTS™

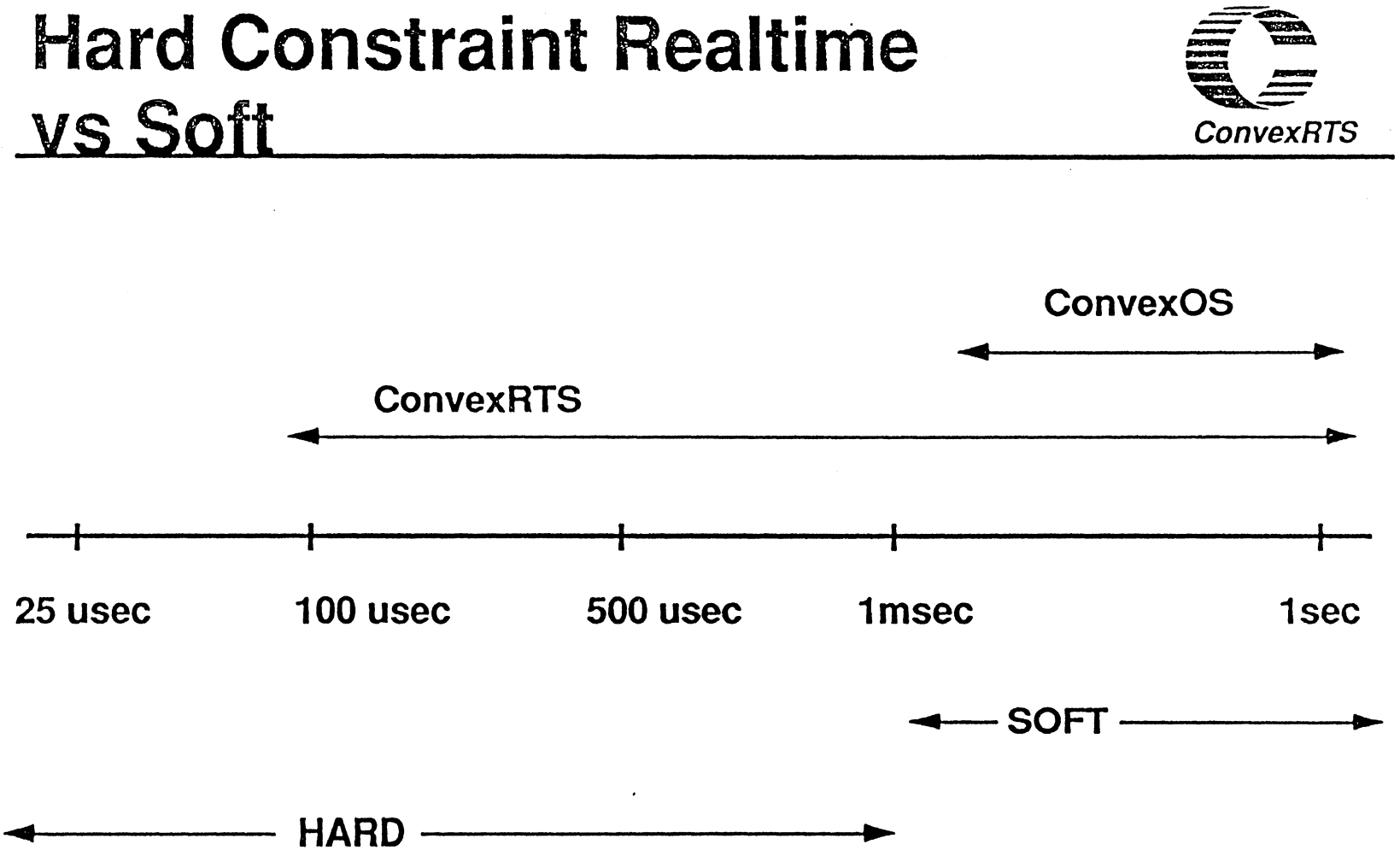


Hard Constraint Realtime vs Soft

ConvexRTS



Hard Constraint Realtime vs Soft



note: ConvexOS is not deterministic even in the SOFT domain

04/30/91 rev 3.0

Preemptor Line Up

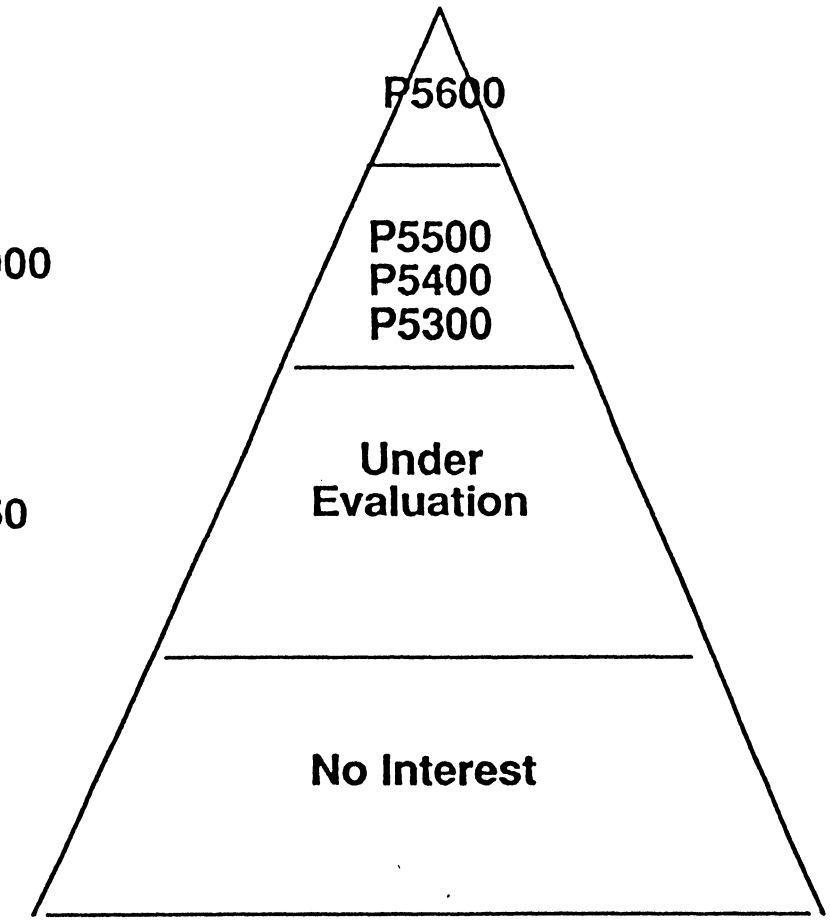
P5500 - 1 to 8 CPUs, derived from C3400 series, Realtime Features

P5300 - 1 to 2 CPUs, small footprint version of P5500, Realtime Features

P5400 - 1 to 4 CPUs, derived from C3200 series, lacks CPU level Realtime features.

P5600 - 1 to 8 CPUs, derived from C3800 series, lacks CPU level Realtime features.

MARKET PICTURE



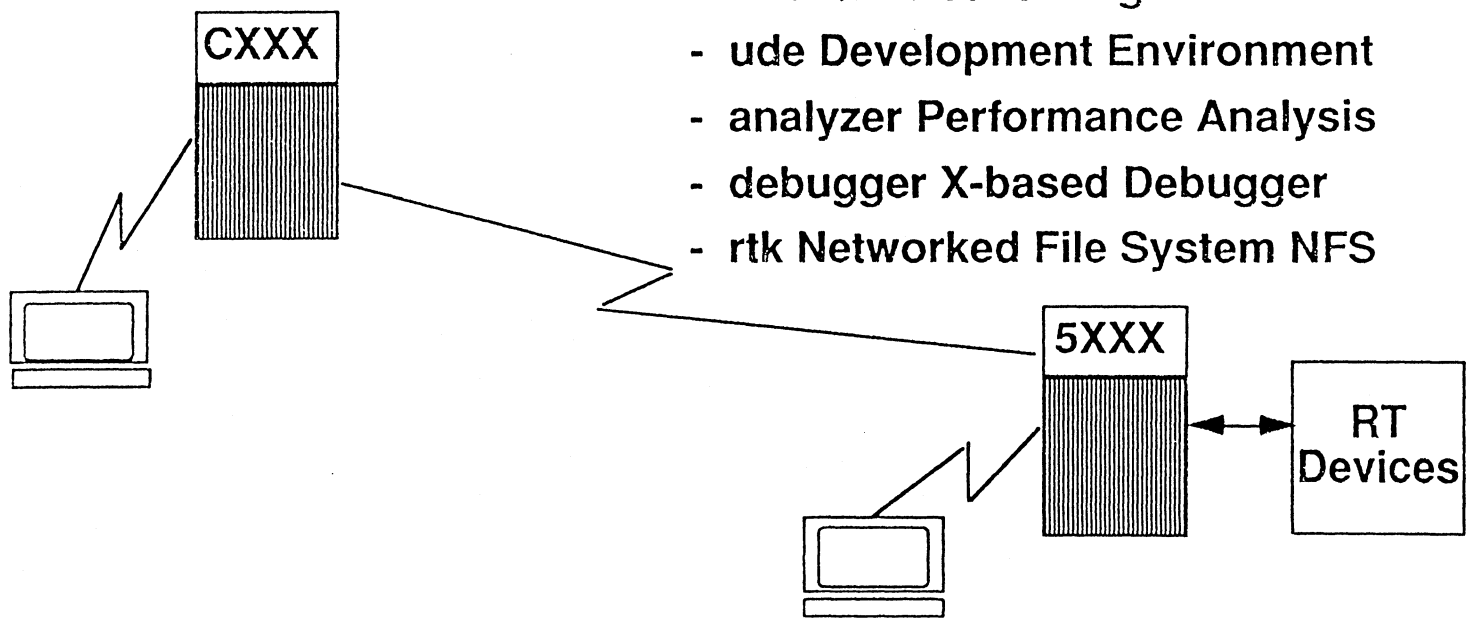
Real-Time Projects

ConvexOS Resident

- Network-based Debugger
- rtk Run-Time Libraries

ConvexRTS Resident

- rtk Real-Time Kernel
  - rdb Debugger
  - Real-Time Filesystem
  - TCP/IP Networking
- ude Development Environment
- analyzer Performance Analysis
- debugger X-based Debugger
- rtk Networked File System NFS

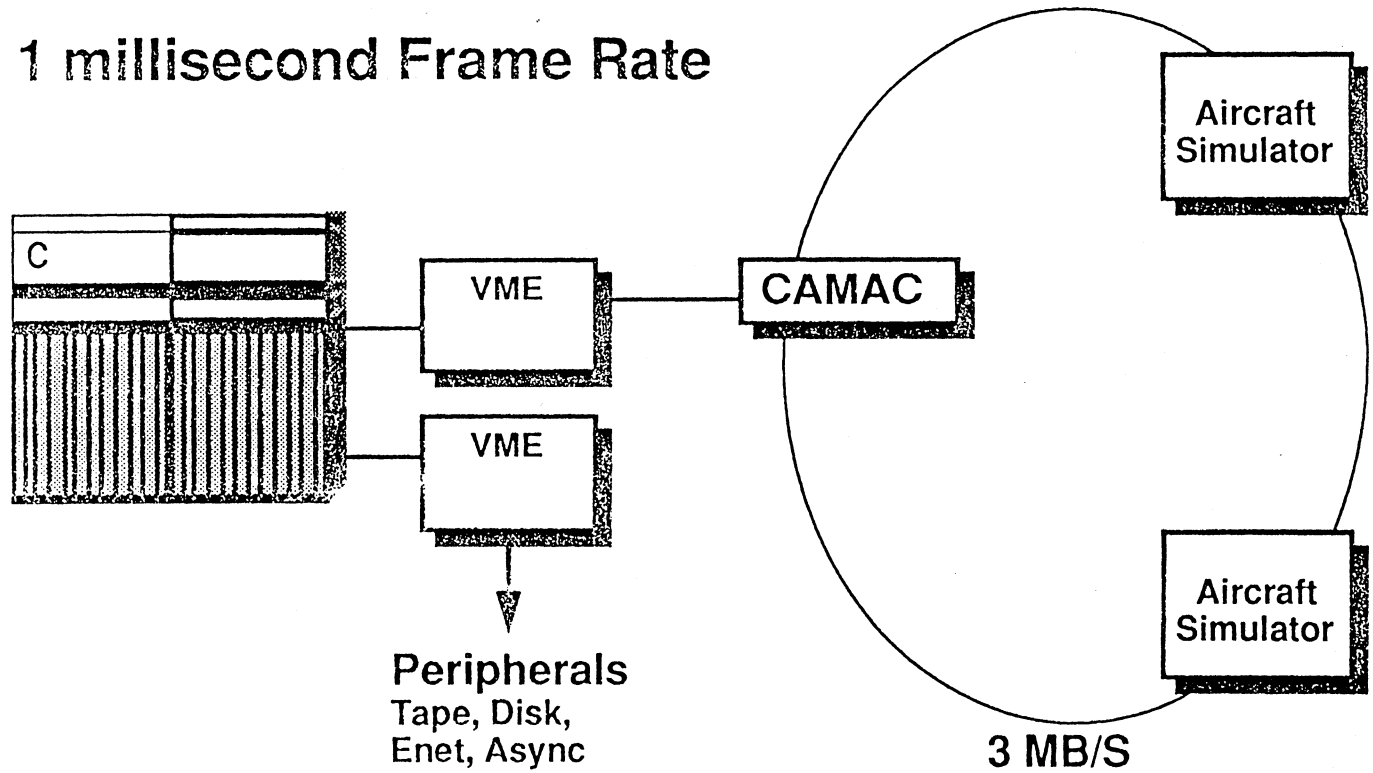


Real-Time I/O Subsystem

Real -Time IOP (RTIOP), reduces latency of real time I/O

- Design modifications are complete to increase RIOP processor speed
- Functional specs are complete to lower interrupt latency to and from real time controllers in the VME bus
- On schedule for Q1-92 production release

Realtime Network



Special Systems Services

Design Consulting

Custom Hardware

Device Drivers

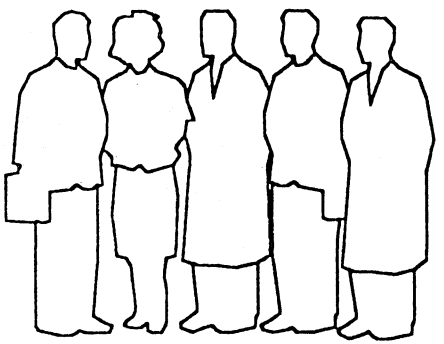
Applications Software

Convex Development Team

I/O Products

Processors

Realtime



Development SW

System Software

S/W Environments

Special Systems

Special Systems

Established to support *unique* customer needs

Team with customer and virtually all Convex organizations

Full fledged development facility

Engineering resources include:

- dedicated hardware, software, & applications engineers
- dedicated laboratory facilities
- state of the art development tools
- strong experience base in many technologies
- flexibility to quickly respond to customer needs

Product maintenance and support

Project Examples

Storage
- Exabyte
- Optical Disk
- STC4980

Data Acquisition
- Solid State Memory
- Aptec Computer I/F

Bus Interfaces
- IEEE-488
- SCSI

Radio Astronomy
- IIS

Image Processing
- HDTV
- Frame Buffers

CPU Modifications
- Additional I/O Ports
- Export

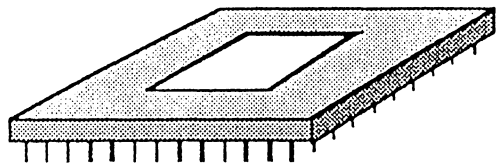
Petroleum
- HSR11B
- Telex Tape
- Versatec

Instrumentation Recorders
- Datatape
- Fairchild
- Ampex

Realtime
- CAMAC

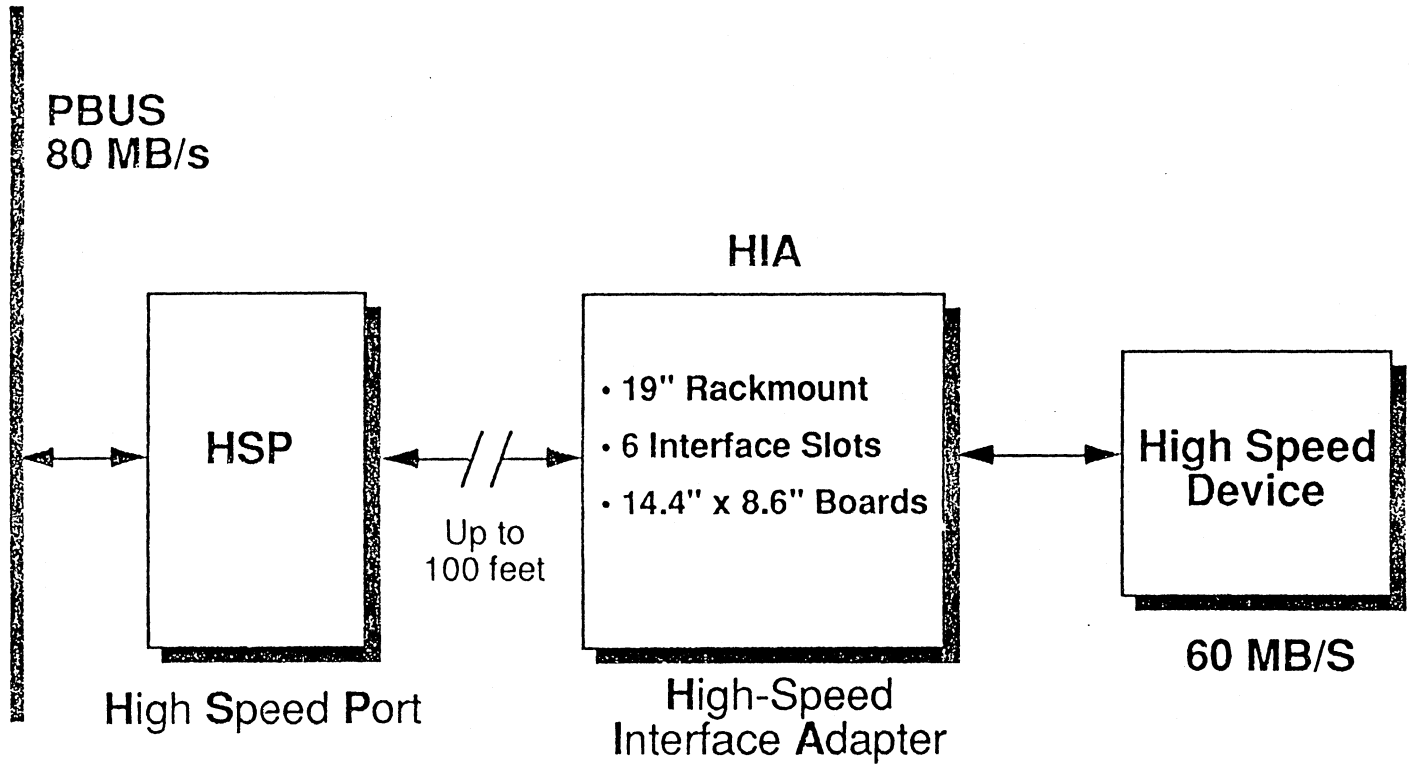
Quick Turn Projects

Reprogramable Logic Cell Arrays (LCA)

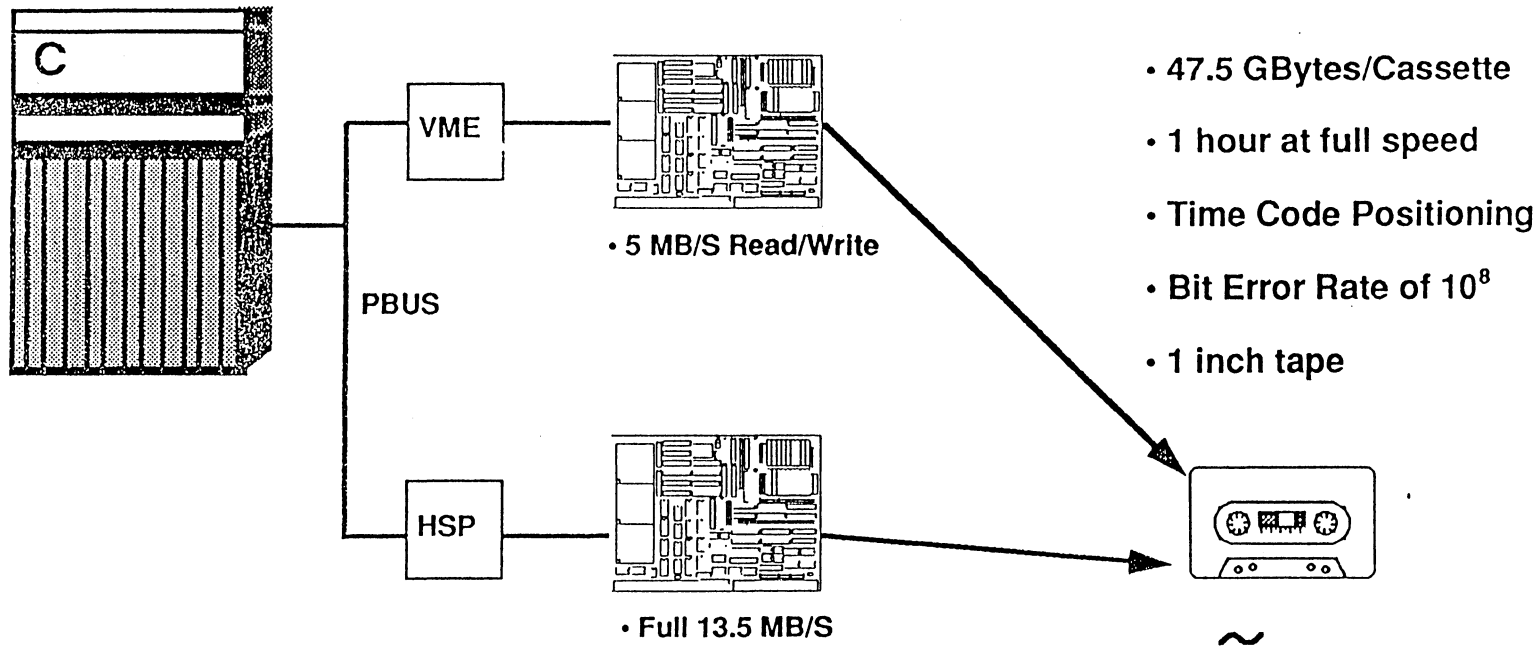


- **Flexible platform for Custom Designs**
- **Quick wireless solution to field integration problems**
- **Faster Time To Market**
- **Reduced cost to the customer**

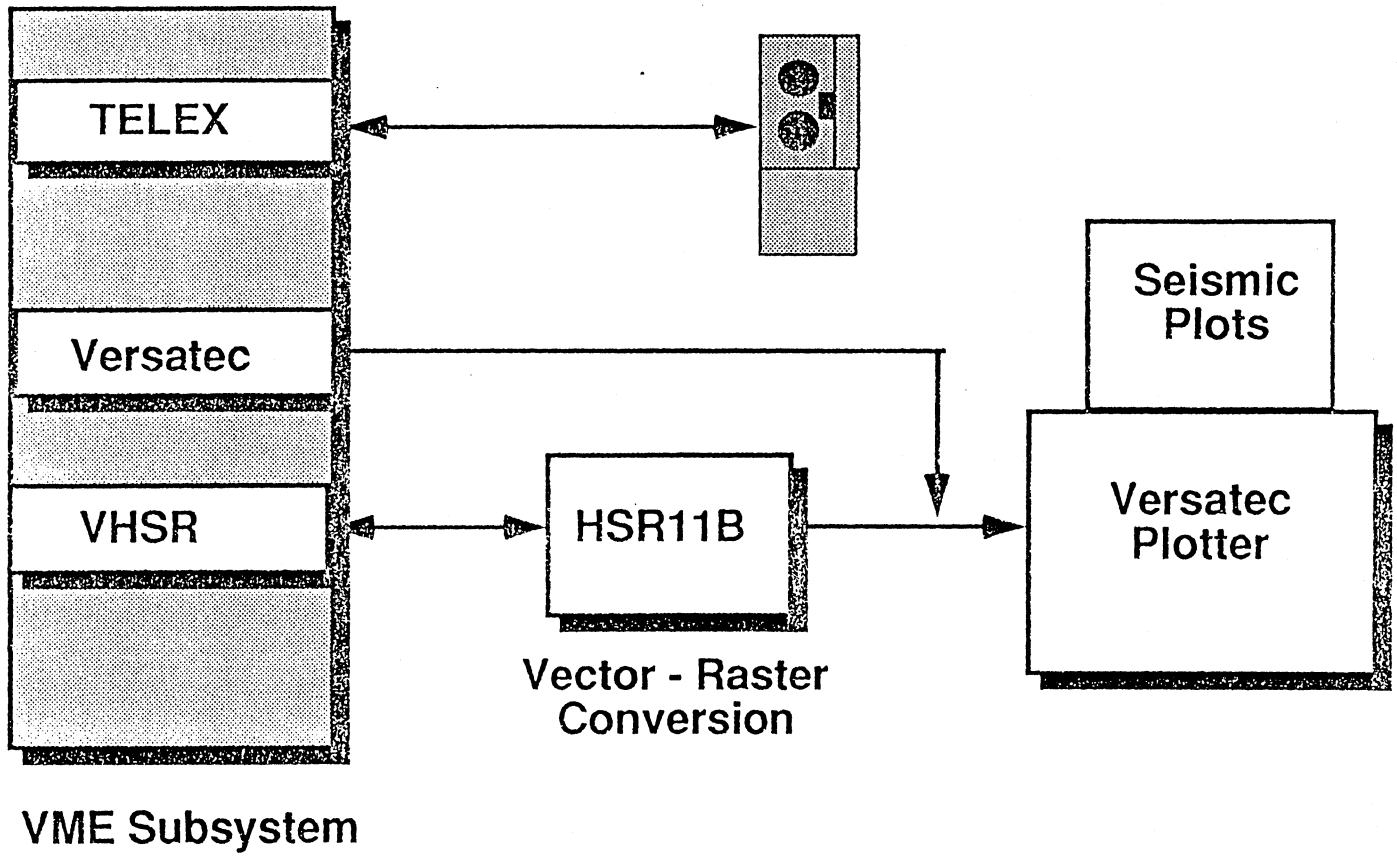
High Speed Interfacing



Ampex DCRSi Tape



VME Products



CONVEX Computer Corporation

ConvexOS/Secure

Development and Evaluation

Jerry Schieffer

Convex Computer Corp

Background for product

– Why develop an *evaluated* OS product?

– Target market : Some customers require evaluated products to meet their procurement rules.

"*Gotta have C2 in 1992*" -Uncle Sam

Product Goals

– Project charter developed: "Deliver an evaluated trusted OS as soon as possible"

– Secondary goal: "Due to duration of project, include as much future functionality of Convex OS development as possible"

Two Themes

There are two parts to delivering a trusted OS product.

1. Develop and package the security technology.

2. Complete the evaluation process including especially documentation, testing, and agency assurance.

TRUSTED PRODUCT EVALUATION PROGRAM

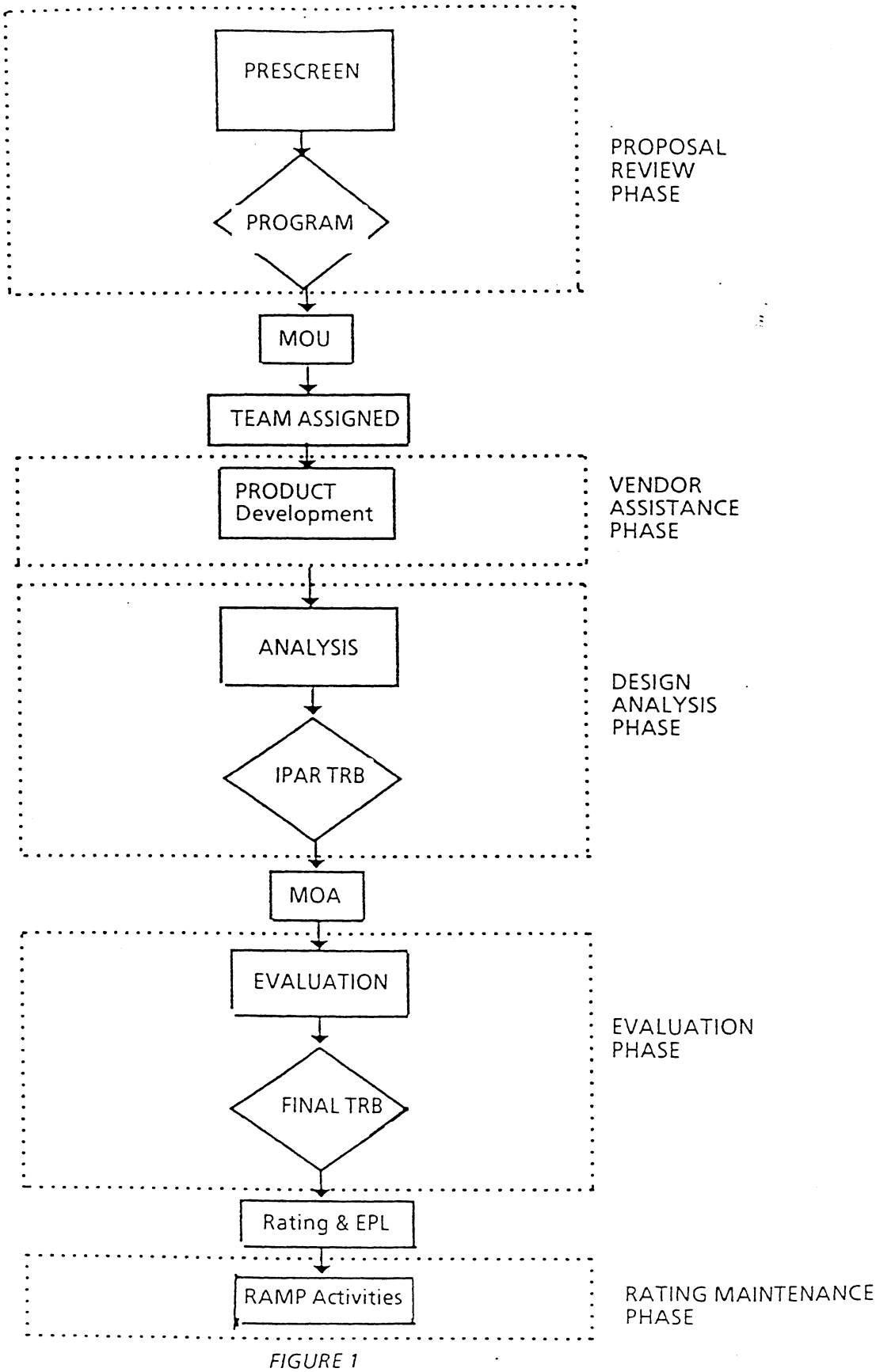
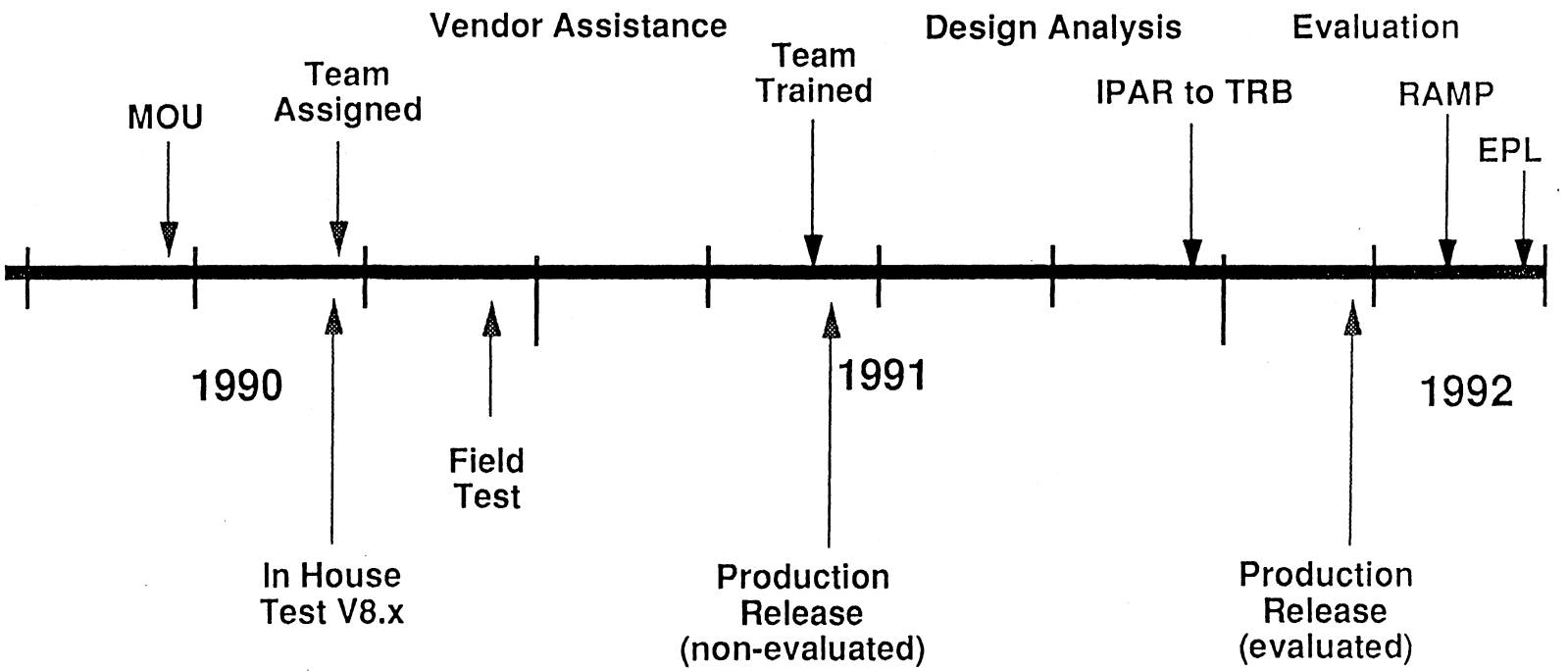


FIGURE 1

Project Milestones

Trusted OS C2 Level Evaluation



ConvexOS/Secure Project History

- Initial staffing in April 1989
- Began hiring development team in Q2 1989
- Project plan published July 1989
- Preliminary Technical Review held August 1989
- Licensed SecureWare technology September 1989
- Development Q3 1989 to present
- Memorandum of Understanding signed June 1990

ConvexOS/Secure Project History (cont.)

- NCSC Evaluation team assigned August 1990
- Beta Release ConvexOS/Secure V9.5 December 1990
- Completed Vendor Assistance Phase March 1991
- Received Initial Product Assessment Report July 1991

ConvexOS/Secure

Overview

C2 Security Requirements

POLICY	Discretionary Access Controls to the granularity of a single user No access to data in reused objects
ACCOUNTABILITY	User identification through passwords Audit trails of all accesses to protected objects
ASSURANCE	Architecture which protects security system Diagnostics to ensure correct operation Testing for weakness
DOCUMENTATION	Documentation for users, administrators, and evaluators

ConvexOS/Secure Features

- Discretionary access control
- Authentication
- Object reuse
- Audit trails
- System integrity
- Documentation

Major Functionality

- Modified /etc/passwd
- Access Control Lists
- Modified microcode
- Extensive Audit Trails
- Documentation
- C2, C3200, C3400, C3800 Evaluations

ConvexOS/Secure

Details

Discretionary Access Control

- Owner, group, other permissions
- Access Control Lists (ACL's) based on Trusix model

Object Reuse

- Disk blocks may be overwritten when freed
- All memory buffers are cleared before being allocated to a process (except for shared memory)
- All processor state will be cleared before a processor is allocated

Authentication

- Authentication via login name and password
- A login user id (LUID) will be inherited by every process descended from the login
- Type restrictions may be enforced on passwords
- Password database will be restructured to protect encrypted passwords from read access

Audit Trails

- Audit trails will be maintained for security related events
- LUID will be associated with each event
- Storage of audit trail will rotate between filesystems
- An audit reduction interface will allow examination of the audit by start and stop time, user and group id, and object name

Audit Events

- **Current system:**
  - Successful/unsuccessful login
  - Successful program instantiation
  - Failed file access
- **Secure system:**
  - Failed process instantiation
  - Password change
  - Database accesses
  - Audit initialization
  - Audit parameter modification
  - Audit report generation
  - Audit file archival
  - Object deletion, modification, and creation
  - Access changes and denials
  - Resource denials

System Integrity

- Hardware protects system through ring structured address space
- Virtual memory protects process data
- File system objects associated with devices or system databases will be protected via ACLs
- SPU verifies correct operation of hardware and microcode

Documentation

- ConvexOS documentation set
- Trusted Facilities Guide
- Security Features User's Guide
- Trusted Programmer's Reference
- ConvexOS Architecture Reference

ConvexOS/Secure V9.5
Current Status

Unevaluated product with C2 trust functionality

Product available now - Released in June

Supports C1, C2, C32XX

ConvexOS/Secure Evaluation Status

- Completed

  Preliminary Technical Review

  Memorandum of Understanding

  Vendor Assistance Phase

- Evaluation is in Design Analysis Phase

  Received first draft of *Initial Product Assessment Report* from evaluation team

- Schedule for entering formal evaluation is Q1 1992
- Convex hopes to be on Evaluated Products List by Mid 1992

ConvexOS/Secure Evaluated Product

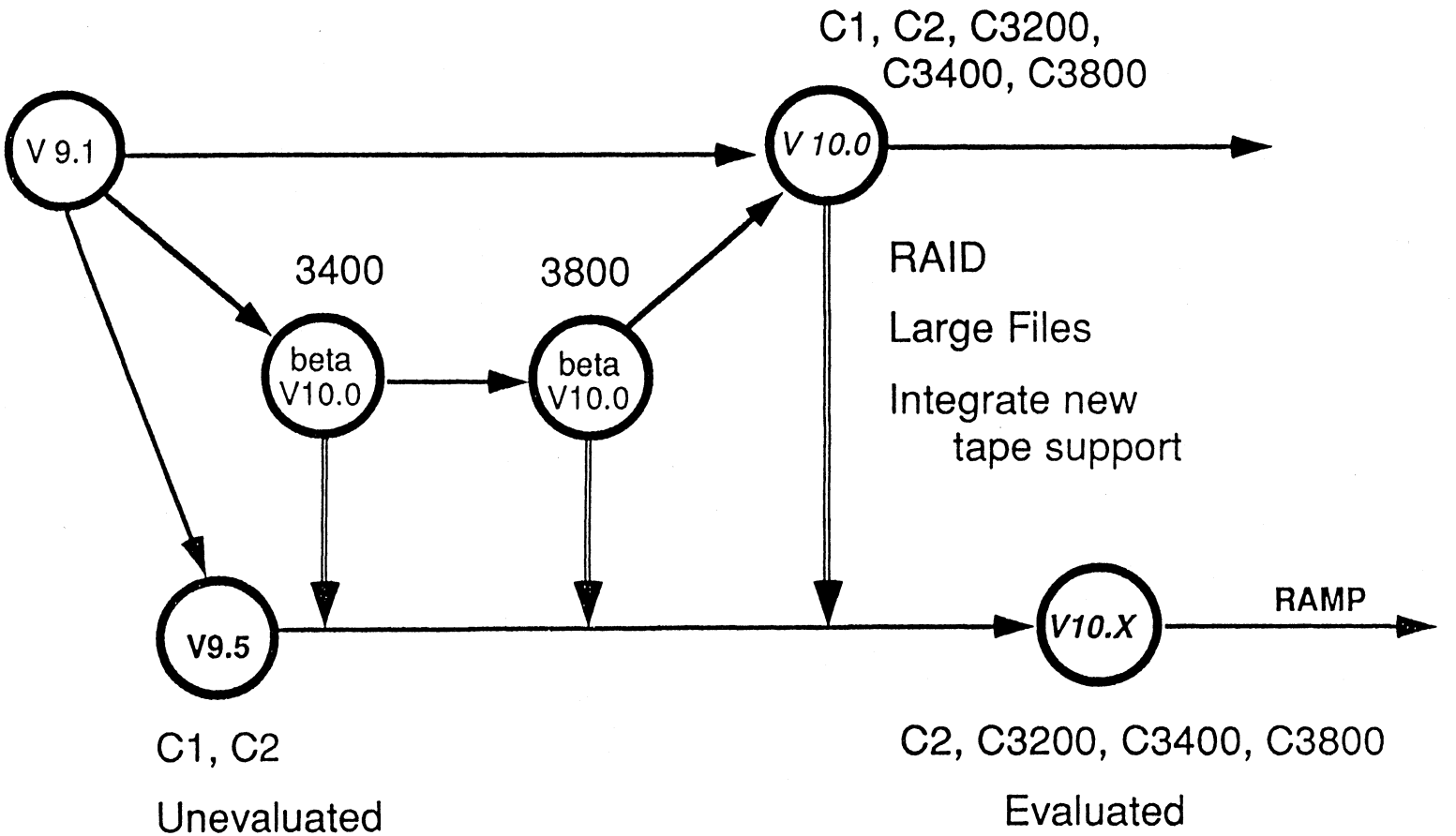
ConvexOS/Secure V10.X

Will be produced from ConvexOS V10.0

- Includes V10.0 functions
  - RAID
  - Large files
  - C3400 support
  - C3800 support

Strict revision control for formal evaluation

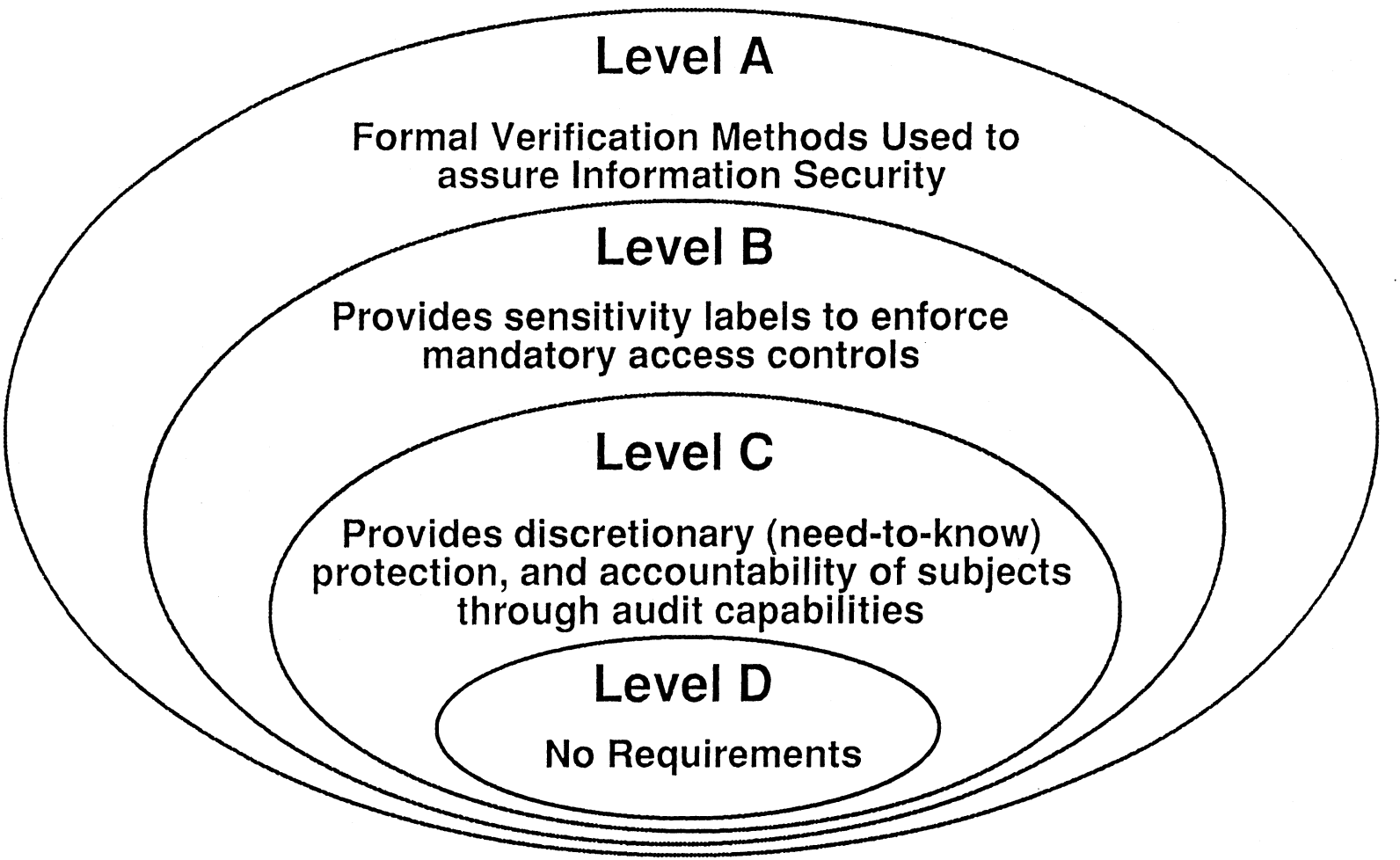
Relation to Other Releases



100

Futures

Orange Book Divisions



Trust Levels

C2

- Basic single level trusted system
- Mandatory in 1992 for Federal procurements

B1

- Provides multi-level security
- Mandatory separation of security level access
- Additional authentication and identification features
- Additional documentation and testing

B2

- Sensitivity labels and additional access controls
- Significant modularity and design requirements
- Significant assurance requirements (NCSC looks over shoulder)

B3

- Export restrictions
- Generally tougher requirements

Summary

Product available today - functionality but not yet evaluated (V9.5)

Pursuing C2 level evaluation (V10.X)

In Design Analysis Phase (DAP) with NCSC

- Have met all of our milestones and expect to stay on track for Formal Evaluation starting in Q1 1992
- Have had no major technology issues - test and assurance issues in work

Commitment to developing a multilevel trust product

- Budgeted to start in 1992
- Expect significant time to evaluation
- Consequently, expect to repeat early release of functionality before evaluation

CONVEXOS/SECURE - UNIX SECURITY IN PERSPECTIVE

Ir. H.A.M. Luijf
TNO Physics and Electronics Laboratory (FEL-TNO)
P.O.Box 96864, 2509 JG The Hague, The Netherlands
Phone: ++31-70-3264221 Email: eric.luijf@fel.tno.nl

ABSTRACT

The need for information security has become obvious over the last decade. This paper covers the following topics:

- Information security, why did it become a hot topic?
- Information security should fit the organization; policies; threat and risk analysis;
- Criteria for trusted systems;
- DoD/NCSC C2 and B1-classes of security, what does that mean?
- UNIX and security, mutually exclusive?
- Security standardization efforts (targeted to UNIX).

This determines the framework for the discussion on how the ConvexOS/Secure product can help sites to partly enhance the security of Convex systems. The security issues left open as well as the initial experience with the ConvexOS/Secure product will be presented shortly.

Keywords: ConvexOS/Secure, information security, UNIX security

BACKGROUND

The TNO Physics and Electronics Laboratory (FEL-TNO) is a research institute of the TNO Division of National Defense Research. TNO is the Netherlands organization for applied scientific research. Current R&D activities concentrate on sensor systems (radar, optics, sonar, infra-red) and information & communication systems, trainers & simulators and policy support.

The R&D activities require adequate and modern information processing and networking facilities. A total of more than 400 networked PC's (Novell), transputer and VME systems, workstations (DEC, SUN), servers (CDC, DEC, SUN), large VAXes and a CONVEX C230 cooperate in a complex network. These facilities should meet quality requirements that include security, flexibility and user friendliness to support (and not hinder) the R&D activities.

One of the R&D fields is 'Secure Information', including both the design of secure information and communication systems and the evaluation of (sub)systems or concepts designed or implemented by 'third parties'.
This paper is based upon earlier and current studies by FEL-TNO.

INFORMATION SECURITY, A HOT TOPIC

Over the last decade information security has become a very hot topic [48]. Many security incidents alerted organizations to pay attention to information security, sometimes in a very hard way.
In the sixties and seventies only a limited number of people had easy physical access to the information processing systems locked up in special rooms. The skills required, the limited access, only a few terminals and social control at the access points offered some security. If any security measures were taken at all, they were extensions of known

security measures: locks, accounting and procedures. Information security was hardly a risk factor at that time. Getting the information processed at all posed enough of a problem each day. As a result, the computer manufacturers paid only limited attention to security aspects in their systems. Features were added ad hoc without a thorough design and integration. As they felt some responsibility, EDP departments applied these security features in an ad hoc way and only if it didn't hinder users.

In the eighties, all the access limitations were taken away. Systems were interconnected, in many cases with the rest of the world. Nowadays it is sometimes even easier to exchange information with one of your antipodes than with someone at your site using another type of system. The PC-boom, both in organizations and at home, raised the skills of many people. On many bulletin boards one can find operating system bugs which can be exploited....

It became more and more apparent to many organizations that their daily activities depend on information processing systems being insecure, open to the world and vulnerable to the first non-cooperative inside, sometimes outside, 'user'. Organizations find themselves trapped in computer and network technical security problems, high costs, organizational problems and a rapidly increasing number of threats causing high risks.
The only way out of these problems is to start at the top of the organization.

INFORMATION SECURITY, AN ORGANIZATIONAL ISSUE

For each organization, the top-level management should have stated the objectives and direction of the organization. The first of four steps to secure information is to deduce from the organization's objectives the organization's information security policy. This policy can only be stated by the top-level management.
The information security policy states for instance:

- the relationship of the security policy with the objectives and direction of the organization;
- the people/department(s) in the organization that are in charge of and responsible for the security of information;
- the relative importance of: information confidentiality, information integrity and availability/continuity for establishing the objectives of the organization;
- the goals for information security;
- the security environment: either active or inactive participation of users; either hostile or closed environment; user security awareness program or not;
- the policy sub-goals like: user-friendly, rational, consistent, coherent, accountability, flexibility, robustness in case of reconfigurations;
- the security evaluation (risk and threat analysis) policy and security maintenance policy.

The security policy might show different priorities for different parts of an organization. Nevertheless, as the objectives of a lower organizational level cannot contradict those of a higher level, the main security objectives should be the same. It should only differ in detail.

The second step is to make an inventory of all threats followed by ordering the relative importance of these threats by assessing the risks. Again, only the organization's own security policies and culture cause a threat to classify as either a high or a low risk.

The third step is to address the highest risk factors, either by technical, physical, procedural or organizational means. The fourth step, the most essential one, is to start all over. Security is not a manner of a one-time effort, but it is a long-term reiterated effort. As the objectives of an organization do not change quickly, security policies will only slightly change. However, the outside world is changing rapidly as well as the use of new technology causing new threats to arise and risks to change in (relative) importance. Note that outside world means society, which also includes the organization's own employees.

Using these steps, technical solutions aren't driving security any more. Rather than doing 'some' information security in a random, unmanageable fashion, these steps cause security to be applied according to the needs of the organization and the users.

SECURITY THREATS AND RISK ANALYSIS

In literature, one finds hardly any information on how to do a threat and risk analysis in a non-commercial environment.
For the threat analysis, we have used the method of horizontal and vertical segmentation, a method developed by TNO-FEL. Each security sensitive environment (total complex, main computer room, special area's, work desks, import/export of data) is identified ('horizontal split') in a top-down approach. Then, for each of those environments the aspects organization, people, applications, operating system, information storage, hardware, network, physical environment are analyzed for threats ('vertical split').

After having identified all threats in this matrix, the risks related to these threats (risk= probability of threat * loss value) can be estimated. In a commercial environment, it is easy to estimate loss values in case a threat becomes reality. In a non-commercial environment, for instance an university, the value of knowledge is less easy to estimate.
An alternative method is to normalize the risk values on the interval [0.0..1.0], zero being no risk and one being the highest risk. Risks can be ordered in this interval using a probability model.
Using the quantitative method, which is promoted by NIST [17], each risk is estimated very precisely. The qualitative method assigns risks to a certain (named) risk interval (distribution clustering). For instance, the interval [0.0..0.15} is a low risk, [0.15..0.6} is a medium scale risk and the interval [0.6..1.0] denotes a high risk. We have found the qualitative method to give suitable results in a technical environment.

CRITERIA FOR TRUSTED SYSTEMS

In 1977 the United States Department of Defense started coordinated efforts to state the (military) computer security requirements and to define general criteria to evaluate special and commercial products against these requirements. The DoD Computer Security Center started in 1978, which in 1981 became the National Computer Security Center (NCSC).
As a result of these efforts, a series of security 'standards' was published (the so called rainbow series, named after the cover colours). The first document, the Trusted Computer System Evaluation Criteria (TCSEC) or orange book, had a large impact [1a,1b]. This because the US government requiring all systems to meet certain minimal security requirements.

Other well-known 'coloured books' are: the Trusted Network Interpretation (TNI) or red book [2] and the Password Management Guideline or green book [5].

Related to these documents are rationales, guides and guide-lines. For further reference, see the literature and reference list [3,4,6-12].

The TCSEC establishes criteria for evaluating the security of computer systems. It should be noted that TCSEC concentrates on information confidentiality and more or less ignores other security aspects such as integrity and availability.

	C1	C2	B1	B2	B3	A1
Security Policy						
Discretionary access control	X	X	e	e	X	e
Object re-use prevention	-	X	e	e	e	e
Labels	-	-	X	X	e	e
Label Integrity	-	-	X	e	e	e
Exportation of labeled data	-	-	X	e	e	e
Exportation to devices	-	-	X	e	e	e
Labelling human-readable output	-	-	X	e	e	e
Mandatory access control	-	-	X	X	e	e
Device labels	-	-	-	X	e	e
Subject sensitivity labels	-	-	-	X	e	e
Accountability						
ID/Authentication	X	X	X	e	e	e
Auditing	-	X	X	X	X	e
Trusted path	-	-	-	X	X	e
Assurance						
System architecture	X	X	X	X	X	e
System integrity	X	e	e	e	e	e
Security testing	X	X	X	X	X	X
Design specification and verification	-	-	X	X	X	X
Covert channel analysis	-	-	-	X	X	X
Trusted facility management	-	-	-	X	X	e
Configuration management	-	-	-	X	e	X
Trusted recovery	-	-	-	-	X	e
Trusted distribution	-	-	-	-	-	X
Documentation						
Security features user's guide	X	e	e	e	e	e
Trusted facility manual	X	X	X	X	X	e
Test documentation	X	e	e	X	e	X
Design documentation	X	e	X	X	X	X

Figure 1: Trusted Computer System Evaluation Criteria (- = no requirements; X = new or enhanced requirements; e = no additional requirements)

The evaluation ratings can be divided into 4 main levels of trustworthiness:

- D - level : minimal/no security.
- C - level : discretionary access (DAC): the owner controls the transfer of authorization rights to access and use of objects (e.g. files, directories).
- B - level : mandatory access (MAC)/multi-level security: the system, using a security policy based upon sensivity labelling and allowed-to-know, controls the transfer of authorization rights to access and use of objects. The security administrator dictates the security policy.
- A - level : verified security and trusted distribution.

Within these levels, currently 6 classes of trust are defined: C1, C2, B1, B2, B3 and A1.

In figure 1, a summary is given of Security Policy, Accountability and Assurance requirements for the security classes. Standard UNIX discretionary access control features (permission bits) fulfil the C1-class requirements. The C2-class requires a more finely grained discretionary access control system. Users should be individually accountable for their actions through login procedures, auditing and resource isolation. Thus access of files and directories should be allowed or denied to individual users, something which is not easily done in case of 'group' and 'other' permissions bits.
The B1-class requirements do not require additional strength above the C2-class requirements. Only mandatory access features are needed. The step from the B1 to the B2-class requires a lot more assurance of correctness, verification, testing and the removal of covert channels.

The last couple of years several European countries developed their own (version of) 'Orange Book' [13,14]. Realizing this, France, Germany, United Kingdom and the Netherlands bundled their efforts to merge their own versions into what might become the European Community IT-Security Evaluation Criteria. Version 1.2 of the Information Technology Security Evaluation Criteria (ITSEC) was published recently [15]. That version will be frozen for the next two years and will be used to evaluate systems and products. Products are evaluated taken their intended usage into account; systems are evaluated in their actual environment. Work has started to develop the Information Technology Security Evaluation Manual (ITSEM). The first public draft is scheduled to appear in January 1992 [16]. A delay is possible as much work has yet to be done.
ITSEC is much more flexible than TCSEC. Beside pre-defined functionality classes, which include the old 'Orange Book' classes, a manufacturer can claim certain security levels which will be evaluated, using methods and processes described in ITSEM, for assurance, correctness and effectiveness. Beside trustworthiness classes (F-C1, F-C2, F-B1, F-B2, F-B3), ITSEC also has pre-defined functionality classes for information integrity and availability (F-IN, F-AV, F-DI and F-DX).

UNIX AND SECURITY

UNIX and security, mutually exclusive? The stream of newspaper articles talking about thousands of UNIX systems being infected by a virus on the Internet and other incidents lead many people to believe that the answer is yes. Is that correct? No, not completely.
UNIX was developed in the early seventies as a simple version of the Multics operating system [50]. UNIX was developed for systems 'around the corner' with a user community consisting of a limited number of cooperative users.

UNIX was never developed with any idea of security in mind [19]. The only security safeguard consist of permission bits, mainly to control inadvertent deletion of files.
The UNIX developments were not coordinated by a single design team, but more or less all universities around the world steered the directions and developments. This very much complicates the separation and shielding of a trusted computer base (kernel).

What are some of the main areas of security concern in 'standard' UNIX? Here follows a list:

1 The superuser has all privileges. There is no distinction of duties to limit privileges. Distribution of privileges amongst several job functions is hardly feasible. As a result, each hacker tries to obtain the superuser password or the root privileges.

2 UNIX vendors deliver the default system setup in a very permissive way. Unlike other operating systems (MULTICS, VAX VMS, NOS/VE), steps should be taken by users to shield their own 'domain'. Most system commands, files and device permissions are set too open, resulting in an eldorado for hackers.

3 Due to the additions of less rigid design and development of commands, features and subsystems, the wide availability of source code and a large, very experienced community of programmers, any operating system bug will be exploited by someone at some time. The only question is: when?

4 Manufacturers of UNIX systems are pressed by the market to deliver new software features. There is hardly any time for proper design and testing of new features, let alone the testing for security implications. If you regularly read the CERT advisories [23] and find which security blunders are made by the vendors, you don't dare to put your bytes on a workstation any more. Nevertheless, system software developed by some of these vendors has been ported to other vendor's systems, including CONVEX systems (e.g. SUN PC-NFS server code).

5 Networking of systems, including connections to the outside world, are made without any careful planning and taking of enough precautions and security measures. A lot of internal information for instance is given away via name servers and commands like *rwho*, *finger* and *sendmail*.
Most of the recent breaches of UNIX system security (intrusions, viruses and worms) were caused by network access openings. Until now, most system administrator's literature on UNIX security concentrate on the operating system, ignoring network security [18,21,28].

6 On the local network, workstations are added giving each owner of such a workstation tools to monitor all packets on the Ethernet. It is fairly easy for a workstation to masquerade itself being a trusted system on the network.

7 The work of a UNIX system administrator is very complicated. It is very difficult to maintain consistency and overview. Just a slight mistake and the system is open for any malicious user.

The last couple of years, UNIX is brought to large systems by many manufacturers. Security requirements for these systems are obvious. Larger user communities, service guarantees and many more factors require more security features in UNIX. Currently, most UNIX 'camps' work on security improvements:

- AT&T UNIX System V release 4.1 Enhanced Security (ES) has many security improvements including B2-class features (labelling, auditing, suppression of covert channels) [20];
- X/OPEN published a very helpful UNIX security guide [22];

- UNIX International has put B1-class (and higher) security on their road map;
- The Open Software Foundation (OSF) included B1-class security features in the OSF/1 portability kit [25];
- NCSC formed, together with manufacturers, the Trusted UNIX working group (TRUSIX) [27];
- IEEE Portable Operating System Interface for Computer Environments (POSIX) established a subcommittee (SC/6) developing the POSIX security standard P1003.6 [26].
  The current draft of POSIX 1003.6 is in our opinion still far from complete and is not unambiguous. It misses any link with the ISO/OSI Security Architecture and other standardization efforts on topics as for instance naming conventions, networking [47];
- The Trusted Systems Interoperability Group (TSIG) facilitate the design and development process of vendors who are working on the interoperability of trusted systems. This group currently focusses on the interoperability of B1/Compartmented Mode Workstations (DEC, SUN, APPLE) [41].
- Internet community: Site security handbook (RFC 1244) [31].

Research efforts as well as commercial efforts indicate that B1-class of trust can be obtained by an adapted version of UNIX and that even B2/B3-class of trust can be attained while maintaining compatibility with 'standard' UNIX [24, 49].

A company, called SecureWare Inc., has developed different packages containing modifications for the UNIX kernel and some programs (for instance login, passwd) and additional auditing and system management programs [25,33]. The Security Module Package (SMP) enhances UNIX to the C2-class. The SMP+ package adds mandatory access features (B1-class) and the Compartmented Security Mode Package (CMP) adds the required features for a compartmented security mode system [40, 41].
A large number of computer manufacturers use one of these packages as a basis to enhance the security of their systems (for instance APPLE, CDC, DEC, OSF, IBM, MIPS). Hence, the SMP(+) code has become a de facto security standard as a basis to start with.
Unfortunately the user commands for dealing with access control lists, as defined in the 7th draft of POSIX P1003.6, are unlike those in the SecureWare product. POSIX proposes the commands *getacl* and *setacl* while SecureWare implementations use the edacl (BSD) and chacl (System V) commands.

OTHER SECURITY (STANDARDISATION) EFFORTS

ISO/OSI SECURITY ARCHITECTURE

From [47]: "Within the ISO OSI/IEC/JTC1 subcommittee SC21 (OSI Architecture, Management and Upper Layers) a unifying view towards security standards is promoted. The aim is to assure coherence of all the work done relative to security in open systems. Until now, SC21 has been concentrating on OSI rather than the broader area of open systems." A direction in which unification could be done has been described in [51,52].

ISO/OSI SECURE OPEN SYSTEMS

The ISO/OSI/IEC/JTC1 subcommittee SC27, which started in 1990, (Secure Opens Systems) will work on standardisation of security techniques and the connection of secure applications. Having seen the scope of the work and the amount of liasons with other groups, it is conceivable that SC27 will become a driving force in the security standardisation field.

KERBEROS

Kerberos, named after the three-headed mythological hell-hound guarding Hades, is a collection of authentication software (700 files, 80.000 lines of code). It is meant to authenticate claimed identities by host's, user's and services in a network of open systems. Some of the goals of Kerberos are: passwords are never sent in plain text over the network; the user should login only once; every user and service has a password.
Kerberos was developed as part of the MIT's Project Athena [35,36] and is part of the OSF technology. Kerberos has been placed by UNIX International on their roadmap as well.

ConvexOS/SECURE

ConvexOS/Secure is a product that is installed on top of the standard ConvexOS operating system. Currently, it delivers C2-class features (not yet evaluated by the NCSC).
ConvexOS/Secure, as most of the current UNIX Security enhancement C2-class type of products, is targeted towards a stand-alone situation. Thus only the trivial networking interfaces (asynchronous terminals) are secured. All CONVEX layered products are supported, although they are not modified. That means, that for instance products and commands like NFS, COVUE, X, *ftp*, and *r*-commands aren't part of the (evaluated) Trusted Computing Base (TCB). For example, they do not report any break-in attempts.
The additionally offered security functionality by ConvexOS/Secure is related to confidentiality in a shielded local environment. Other security aspects as integrity, availability/continuity and networking are not yet covered by the product.

ConvexOS/Secure maintains UNIX compatibility. However, using the offered security features to full extent might impact users, usability and performance. For the additional security overhead one has to pay a price: up to a couple percent of CPU-performance and disk space for storing auditing information. This apart from 27 Mbyte of disk space needed for installation. The actual performance decrease and the disk space required for logging mainly depends on the audit (logging) mask. Logging every security relevant event might result in a couple of Mbyte per several minutes. Tuning towards the site's requirements based upon expected threats significantly reduces the 'price to pay'.
Users regarding open systems as systems 'open for all', will note that the behaviour of the ConvexOS/Secure system is much more restrictive than they were used to (e.g. a restricted umask, login and password restrictions, navigating through directories of other users is inhibited).

A summary of the features of ConvexOS/Secure related to the C2-class requirements (ref. figure 1) follows:

Security Policy:

- Discretionary access control with the granularity of a user is available in the form of ACL's on directories, files, devices and IPC objects (named sockets). Control can be granted or denied for: a user, a group, a wildcard name and none. This in addition to the conventional UNIX permission bits;
- Object re-use is prevented by the *erase_unlink* boot-time parameter, the withdrawal of read access to */dev/mem* and */dev/kmem* for normal users and the clearing of released memory parts (for instance buffers and caches);
- The inadvertent flow of permissions is restricted by clearing the suid and sgid bits in case a file is overwritten.

Accountability:

- Identification and Authentication.

  Using a full screen interface program, called */usr/tcb/bin/authif*, the authentication administrator controls the access to the system as well as defensive measures. The *authif* program superseded the *nu* and *vipw* scripts. System as well as audit administration by means of the *authif* and *auditif* screens respectively is a matter of easy, intuitive navigation. To ease the authentication administration, system wide defaults can be specified. The identification and auditing features are:

  - User control features: memorising the last successful and unsuccessful login as well as the terminal/host from where the login was attempted on a per-user base; locking of the user access after a defined number of login retries and after the expiration of the password;
  - Terminal control features: memorising the last successful and unsuccessful login attempts as well as the terminal/host from where the login was attempted on a per-terminal base; locking of terminal access after a defined number of login retries; adjustable delay of subsequent login attempts;
  - Password control features: either password generation by system or pick by users and additional password restrictions can be selected; expiration time; password lifetime; passwords can be up to 80 (significant) characters long; passwords are moved from */etc/password* to a password database which cannot be read by a normal user; passwords aren't echoed or displayed. The implementation removes some of the password pitfalls described in [43] and helps sites to follow the password security guide-line [5].
  - Privilege control features: the traditional superuser's privileges can be distributed among 'special' users. This results in separation of duties and limitation of powers. The privileges are split into kernel privileges (chown, execsuid, chmodsugid, configaudit, suspendaudit, writeaudit) and subsystem authorizations: auth, audit, fs, lpr, tty and back-up.

- Auditing.

  An additional subsystem consisting of a daemon, kernel changes, kernel interface calls, buffers, control and data reduction programs gives features to audit all security relevant actions.

  Using a full screen interface program, called */usr/tcb/bin/auditif*, the audit administrator controls the actual security logging, collection, reduction and selection processes. These processes can be tuned to become very selective; even security auditing for only one single user can be selected.

  The kernel changes include the book keeping of a login user identification field (luid) which doesn't change when a user runs suid programs or switches to another user's environment via *su*.

Only during system boot, a number of daemons are forked with a login user id (luid) of zero. Only these processes (login, ftp, CXbatch) are allowed to fork and change their luid to a certain user id (non-zero) after authentication of the user. All processes stamped with non-zero luid maintain the luid value during the whole life cycle of the process.

Using the audit selection mask, the system security officer controls the selection of which security relevant actions should be logged. Log entries contain a time/date stamp, luid, real and effective user and group id's, event type, success or failure, calling process as well as further detailed information.

All the auditing features are designed to be compliant with the POSIX P1003.6 logging and auditing interfaces.

- Trusted path.

  Most other vendors support a trusted path login when implementing C2-security, although this feature is only required for B2-class systems and higher. ConvexOS/Secure doesn't support such a feature.

Assurance:

- Architecture.

  The architecture and integrity of the design and implementation of ConvexOS/SECURE will be 'guaranteed' by CONVEX. This is part of the verification and evaluation by the DoD/NCSC of the claimed C2-class security level. This to obtain the intended NCSC C2-rating.

- Integrity and testing.

  Using verification procedures, it is possible to verify the integrity and correct installation of all files and directories required to maintain a secured system according to the released values.

Documentation:

- Additional manuals describing the user security features [39] and the installation and maintenance procedures are available from CONVEX [37,38]. These manuals describe the offered features in an easy to access manner, showing all relevant screens.

EXPERIENCES WITH ConvexOS/Secure

March 1991, TNO-FEL received an early version of the ConvexOS/Secure product which was installed on top of ConvexOS version 9.0. A number of problems were detected, most of which have been corrected in the mean time and/or are documented in the official product's release. The main problem area's are related to another way of system administration as well as some conversion and start-up problems:

1 The conversion of the old password file to the new caused a number of problems. It turned out that users were using passwords longer than 8 characters before the installation of ConvexOS/Secure. Only the first 8 are significant for the old password mechanism. In ConvexOS/Secure all characters are significant, thus user compatibility was broken.

2 The 'finger information field' as supported by *authif* has a limited length compared to 'natural' UNIX. Installing ConvexOS/Secure resulted in some conversion problems.

3 One needs to change SUN Microsystems' PC/NFS server code to adapt to the new password encryption algorithm ('bigcrypt' instead of 'crypt').

4 New passwords are checked against a number of safeguards. As the algorithm was unknown, new passwords were rejected for unknown reasons. The minimum password length depends on the type of characters used.In case of a mixture of uppercase, lowercase and digits, this is 4 characters. For passwords consisting of only uppercase or only lowercase characters, the minimum length is 6. Palindromes and anagrams of user or group names are rejected in case the password restrictions option is enabled. This in accordance of the 'Green Book' requirements [5].

Annoying to Dutch users is the rejection of passwords containing text parts which have no meaning in Dutch, but which are flagged as misspellings by */bin/spell*. Unfortunately, the system doesn't give information about the reason why a new password is rejected. A simple message telling 'too short' or 'known from dictionary' would help to increase the user's security awareness rather than increase user's annoyance.

5 Significant problems occurred when we turned on auditing. Specific to our site, some often used programs do a lot of file I/O operations (for instance open/write/close sequences and a high number of lseeks/sec). This caused serious performance problems as over 450 audit log entries per second were generated. CONVEX has improved the selectivity of auditing in the official release, thus removing the overhead we saw.

6 As only during boot processes can be forked with a login user id of zero, system operation and administration becomes complicated in case of trouble. For instance, when a network daemon or another subsystem (CXbatch) failed unexpectedly under ConvexOS, one was used to (re)start daemons by hand. Under ConvexOS/Secure this might result in unexpected failures after a while as the daemon is not allowed anymore to run under another login user id. We have seen this type of problems when reloading a dump tape after a disk crash; with CXbatch (unable to generate a print out and a mail) and with networking daemons. The only way to solve these problems is to checkpoint jobs and reboot the system.

7 Again, the ConvexOS/Secure product is designed to be NCSC C2-class compliant and only offers security enhancements for stand-alone situations. The way this is done is effective. No diversions of the offered security features have been found by us.

Products and features as NFS, RPC, X, COVUE, r-commands, *ftp* are not part of the TCB to be verified. Sometimes there are valid technical reasons for this. For instance, the access control list (ACL) feature has been added on top of the security of the standard file system. Only the local system calls (and commands) for file access are able to deal with the ACL information [37,38,39]. Exporting and/or importing file systems via, for instance, NFS allow for the by-passing of the access control feature.

In our view, all networking commands (e.g. *ftp*, X, *r*-commands, COVUE) should, as much as possible, be interfaced with ConvexOS/Secure. Currently, this has been done in an inconsistent and incomplete manner. Areas that need to be (re)addressed are: the logging of security relevant actions as for instance break-in attempts; consistent behaviour of the file system regarding files with ACL's; by default a *umask* value of 077 for all the daemons that import files (ftp, Covue, rcp). We have reported to Convex a number of loopholes in this area (e.g. COVUE removing acl's of existing files).

Our wishes for future enhancements of ConvexOS/Secure are:

- DoD/NCSC B1-class compliance (mandatory access), including a configurable Compartmented Security Mode policy behaviour;
- Trusted booting;
- The delivery in Europe of Convex Secure NFS, a product based upon SUN Microsystems' NFS version 4. Convex claims that they are bound by USA export regulations, which disallow the distribution of DES technology. Surprisingly, most other US computer vendors deliver Secure NFS without any problem in Europe (e.g. DEC, MIPS, CDC, SUN).
- A separate network administrator privilege controlling the use of *arp, snmp, netstat, ping, ncp* (COVUEnet), name serving, routing and the read and write access to network configuration tables;
- The *quota* command should be moved from the fs privilege to an operator type of privilege;
- Addition of time period restrictions on interactive user processing (like a feature in SecureWare's SMP package);
- The inclusion of Kerberos software as well as 'kerberized' commands and services, like the rcmd service (*rlogin, rsh, rcp, tftp*);
- The inclusion of (static) security auditing tools, like, for instance, the Computer Oracle and Password System (COPS) package [23,29] and QUEST [30]. COPS is a set of programs that checks: file, directory and device permissions; poor passwords; security of */etc/password* and */etc/group*; programs and files run by *cron*(tab) and during system startup and termination; suid and sgid files that are writeable; system files static integrity (crc); anonymous *ftp* and so on. It includes an expert system which tries to find whether there is a way in which system security can be compromised. Although COPS can be obtained from the Internet by some sites, a version tuned for both the standard and the secured ConvexOS environments might help a lot of sites. This could be done on basis of an 'as-is tool' in the same way as PERL is distributed by Convex. The AT&T QUEST product has the same type of functionality, the expert system exempted.
- A full documentation of the new password rejection algorithm;
- A way to add a database with non-English words (local language) to the new password rejection algorithm;
- The addition of a UNIX security guide, both for ConvexOS and ConvexOS/Secure: The do's and don'ts for system and network administrators. This should include some scripts to verify the static security like the X/OPEN Security Guide.

ACKNOWLEDMENTS

Martin van Roon, who installed the Beta version of ConvexOS/Secure. He worked during the dark hours debugging and solving problems to keep the production going. Jelte Feenstra (Convex B.V.) and Jeffrey Powell (Convex USA) for their assistance in solving problems.

CONCLUSIONS

Security is not a matter of just adding a piece of software to a system. Security is a management issue. Starting at the top, the organization's objectives should be the basis for a security policy (what is important, what needs to be secured). A thorough threat and risk analysis should be undertaken. This is the basis for the organizational, procedural and technical measures to be taken.

In this light, a security enhancement package like ConvexOS/Secure might help sites to secure their CONVEX systems with respect to confidentiality, auditing and tracebility. This in a way that DoD/NCSC C2-level and ITSEC F-C2 class requirements are addressed.

The system security officer (SSO) controls the way ConvexOS/Secure is actually used. The impact on the performance is depending on the selectivity used to log security relevant events.
Layered CONVEX products continue to work on top of ConvexOS/Secure. However, these products are not fully integrated. For instance, break-in attempts via these products are not logged and can not be audited. Accesses of files with ACL's are done in an inconsistent and incomplete manner. We regard this as very serious shortcomings as the ConvexOS/Secure product description suggests to allow auditing of all break-in attempts and to maintain acl security.
In a wish list missing features and products are tabled, including a static security analysis tool as well as Kerberos technology.
ConvexOS/Secure is effective for stand-alone situations. For being effective in networking environments, CONVEX should enhance and extend the layered products a lot.

REFERENCES and LITERATURE LIST

Rainbow series and related documents

1a Department of Defense Trusted Computer System Evaluation Criteria (Orange book)
DoD 5200.28-STD Library No. S225,711
DoD Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000; (December 1985)

1b NATO Trusted Computer Evaluation Criteria
NATO document AC/35-D/102

2 Department of Defense Trusted Network Interpretation (red book)
NCSC-TG-005; DoD -STD Library No. S228,526
DoD Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (July 1987)

3 Computer Security Requirements
(Guidance for applying the DoD Trusted Computer System Evaluation Criteria in specific environments)
NCSC-STD-003-85; DoD-STD Library No. S-226,727
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (June 1985)

4 Technical Rationale behind CSC-STD-003-85: Computer Security Requirements
(Guidance for applying the DoD Trusted Computer System Evaluation Criteria in specific environments)
NCSC-STD-004-85; DoD-STD Library No. S-226,728
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (June 1985)

5 Password Management Guideline (green book)
CSC-STD-002-85 Library No. S-226,994
DoD Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (April 1985)

6 A Guide to Understanding AUDIT in Trusted Systems
NCSC-TG-001 version 2; DoD-STD Library No. S228,470
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (June 1988)

7 A Guide to Understanding Discretionary Access Control in Trusted Systems
3NCSC-TG-003 version 1; DoD-STD Library No. S228,576
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (September 1987)

8 A Guide to Understanding Configuration Control in Trusted Systems
NCSC-TG-006 version 1;
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (March 1988)

9 Computer Security Subsystem - Interpretation
NCSC-TG-009; STD Library No. S230,512
DoD Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (September 1988)

10 Rating Maintenance Phase Program Document(RAMP)
NCSC-TG-013; STD Library No. S232,468
DoD Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (June 1989)

11 COMPUSECese - Computer Security Glossary
NCSC-WA-001 version 1;
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (October 1985)

12 Personal Computer Security Considerations
NCSC-WA-002 version 1;
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (December 1985)

European criteria for trusted IT-systems

13 IT - Security Criteria
Criteria for the evaluation of Thrustworthiness of Information Technology (IT) Systems (The German Green Book)
Zentralstelle für Sicherheit in der Informationstechnik
Bonn, 1st version, 1989; ISBN 3-88784-200-6

14 UK-IT Security Evaluation and Certification Scheme
Publication No 1 - description of the scheme
Department of Trade and Industry (DTI), March, 1991
Room 2/0804, Fiddlers Green Lane, Cheltenham

15 Information Technology Security Evaluation Criteria (ITSEC)
Harmonised criteria of France, Germany, The Netherlands, The United Kingdom
Version 1.2, June 1991

16 Information Technology Security Evaluation Manual (ITSEM)
Restricted draft, July/September 1991
Public draft version 1.0 expected (after) January 1992.

Risk analysis

17 Guideline for Automatic Data Processing Risk Analysis
U.S. Department of Commerce, National Bureau of Standards
FIPS PUB 65, August 1979

UNIX security

18 The UNIX system - UNIX Operating System Security
Grammp, F.T. and Morris, R.H.
AT&T Bell Laboratories Technical Journal
Vol 8 No 63, October 1984, pp 1649 - 1672

19 On the security of UNIX
Ritchie, D.M.
UNIX Programmer's Manual, section 2
AT&T Bell Laboratories

20 Computer System Security: A Closer Look
Amorosso, E.G., Houghton, T.F. and Leighton III, W.J.
(System V release 4.1, System V/MLS)
AT&T Bell Laboratories Technology
Vol 4, No 4, 1989, pp 30-39

21 UNIX System Security
Wood, P.H. and Kochan, S.G.
Hayden Books, Indiana, 1987 ; ISBN 0-8104-6267-2

22 X/OPEN Security Guide
X/OPEN
ISBN 0-13-972142-8
Prentice-Hall, Englewood Cliffs, New Jersey (November 1989)

23 Computer Emergency Response Team (CERT)
Security alerts; security software (COPS package)
Anonymous FTP at: cert.sei.cmu.edu
Short description can be found in ECUforum No 0, May 1991

24 An Overview of Secure UNIX
Datapro Reports on Information Security
Datapro Research, report IS56-001 pp 201-211
McGraw-Hill (January 1990)

25 SecureWare SMP+ Trusted Facility Manual
SecureWare Inc., partno. 010-0006-00, January 1990

26 POSIX Security interface, P1003.6/D7 (7th draft), 1991

27 Rationale for selecting access control list features for the UNIX System
by: TRUSTED UNIX Working Group (TRUSIX)
NCSC-TG-020-A, Library No. S-232,508
National Computer Security Center, Ft.Meade, Maryland (18 August 1989)

28 Improving the Security of Your UNIX System
Curry, D.
SRI International Report ITSTD-721-FR-90-21 (April 1990)
Also at: cert.sei.cmu.edu in /pub/info

29 The COPS Security Checker System
Proceedings of the Summer 1990 USENIX Conference,
Anaheim, California, June 1990, pp 165-170

30 QUEST - A Security Auditing Tool
Kapplow, S.A.
AT&T Technical Journal, May/June 1988, pp 65-71

31 RFC 1244: Site Security Handbook
Holbrook, P. and Reynolds, J.
On-line at ftp.nisc.sri.com or nis.nsf.net as rfc/rfc1244.txt

32 Practical UNIX Security
Garfinkel, S. and Spafford, E.
O'Reilly & Associates, ISBN 0-937175-72-2, May 1991

33 On Incorporating Access Control Lists into the UNIX Operating System
Kramer, S.M. (SecureWare Inc.)
(different ACL models - also see the TRUSIX report [26])
Proceedings USENIX Security Workshop, Portland, OR
August 1988, pp 38-48

34 The Helminthias of the Internet (RFC 1135)
USC/Information Sciences Institute, Marina del Rey, California, December 1989
On-line at ftp.nisc.sri.com as /pub/rfc/rfc1135.txt

35 Kerberos: An Authentication Service for Open Networks
Steiner, J.G. , Neumann, C. and Schiller, J.I.
January and March, 1988
On-line at athena-dist.mit.edu in /pub/kerberos/doc/

36 The Evolution of the Kerberos Authentication Service
Kohl, J.T. (Project Athena, Digital Equipment Corp.)
EurOpen conference proceedings, Tromsoe, Norway; May 1991; pp 295-313
On-line at athena-dist.mit.edu in /pub/kerberos/doc/

CONVEX Documents

37 ConvexOS/Secure V9.5 Release Notice
Document No. 710-013430-001, June 1991

38 Installing ConvexOS/Secure V9.5
Document No. 710-013530-001, June 1991

39 ConvexOS/Secure - Security Features User's Guide
Document No. 710-007030-001, June 1991

Other security related literature

40 Compartmented Mode Workstation: Prototype highlights
Berger, J., Picciotto, J., Woodward, J. and Cummings. P.
IEEE Transactions on Software Engineering
Vol 16, No 6, June 1990, pp 608-618

41 ULTRIX MLS+ Trusted Workstation Software
Digital Equipment Corp. EF-A2192-50; 1991

42 (Tutorial) Computer and Network Security
D.Abrams, M.D. and Podell, H.J.
IEEE Computer Society Order Number 756 (1987)
IEEE Catalog Number EH0255-0, ISBN 0-8186-0756-4
(this book contains a number of interesting reprints)

43 How Crackers Crack Passwords or What Passwords to Avoid
Ana Maria De Alvaré
Lawrence Livermore National Laboratory UCID-21515, Sept 1988

44 A Trusted Network Architecture for AIX Systems
Chii-Ren Tsai et al.
Proceedings USENIX conference, Winter 1989, pp 457-471

45 Guidelines For Audit Log Mechanisms in Secure Computer Systems
Brown, R.L.
Aerospace Report No ATR-88(3770-29)-1, November 1987
The Aerospace Corporation, El Segundo, California

46 Computer (In)security: Infiltrating Open Systems
Witten, I.H.
ABAQUS Vol 4, No 4 - summer 1987, pp 7-25

47 OSI-Security and Relations with other Standards
Overbeek, P.L.
FEL report: FEL-91-B099, March 1991

48 Past, Present and Future
Overbeek, P.L.
FEL report: FEL-91-B100, March 1991 and
Proceedings of Securicom '91, "9e Congrès Mondial de la Protection et de la Sécurité Informatique et des Communications"

49 AT&T System V/MLS R1.1.2 running on UNIX System V R3.1.1. Final Evaluation Report (B1-level rating)
Bielat, K.M et al.
National Computer Security Center, Ft. George G.Meade, Maryland, MD 20755-6000 (October 1989)

Other references:

50 The MULTICS System
Organick, E.I.
MIT Press, 1981; ISBN 0-262-15012-3

51 Uniform Open Systems Model: a network wide view on applications and operating systems
Overbeek, P.L. and Luiijf, H.A.M.
Proceedings of ECODU-47/VIM-50, April 1989; pp.1/112-1/134

52 The FEL-TNO Uniform Open Systems Model: a network wide view on applications and operating systems
Luiijf, H.A.M. and Overbeek, P.L.
Towards an Open World, Proceedings of DECUS Europe Symposium, pp.201-208; The Hague, Holland, September 18-22, 1989

53 Open Systems Security - an Architectural Framework
Karila, A. (Telecom Finland, Business Systems R&D)
Helsinki, June 30, 1991
On-line at ajk.tele.fi in /PublicDocuments
On-line at nic.funet.fi in /pub/doc/security

Abstract of the

Presentation by

MAITE SIERRA

A NON-SCIENTIFIC FIELD TEST OF DISK-STRIPING SPEED

Speed test on computers use to be very different of what users normally can get in their systems. The disk striping system provided by CONVEX seems to be a major advance in speeding up the Input/Output speed, but on the other hand it is a challenge for the disk maintenance: only one failure in any of the disks can produce a stop of the file system(s).

In one system with a large number of disks in stripe we have to consider the advantages offered by that configuration against the possibility of failure on any of the disks. Those advantages have to be measured under the real environment of the file system and the load of the machine. Our main goal was testing the speed offered by different striped disks against single file system disks.

Regardless the results obtained in laboratories with ideal conditions of machine load, I/O specially dedicated to the test, or fragmentation of file systems, we at C.I.C.A. tried to run two different tests: One proposed in the BYTE publication, and a home-made test. The first one allows doing several measures, and among them the I/O speed of one 'disk'; the second one writes and reads the disk and that can be controlled with easy tools (time, etc.).

We run the test on different file systems, with different number of disks in stripe, from one single disk to four, in our CONVEX-240. We also run the test on different CONVEX machines, one in U.L.C.C, and another in O.U.C.S., and with imported file system in a CONVEX from other machines, CONVEX, SUN, VAX.

The results we obtained, and the way we made the test are presented by Maite Sierra.

UNIX – System management

- Distributed environment
- Multi-vendor
- Still no standardization system management
  - POSIX 1003.7 balloting ends Q4 1992
  - OSF DME snapshot 1992 ?
  - No management applications before 1993/1994
- System management remains skill intensive
- System management expertise is non-portable

UNIX – System management

- System management by end-user
  - Takes 10-20 percent of time
- System management by specialist
  - Lot of routine tasks
  - Job satisfaction not very high
- Both options not cost-effective

UNIX – System management

- Backup and Restore
- Software distribution
- User and group management
- Network administration
- User archives
- Single root password
- Security

SysAdmin – System management tool

- Truly multi-vendor
- Uniform menu based interface
- Flexible
- Distributes root password

SysAdmin – System management tool

- Benefits:
  - System management becomes standard
  - Routine tasks by end–user
  - Complex tasks by specialist
- Results in:
  - Improved efficiency
  - End–user needs no specialist skills
  - Specialists handle more complex tasks
  - Reduction in cost

SysAdmin – Functionality Overview

- Administrative Utilities
- Backup Utilities
- Maintenance Menu
- Restore Utilities
- Filesystem Maintenance
- Tape Utilities
- UUCP Maintenance
- Security Menu
- Archive Menu
- Printer Maintenance
- Local Utilities

SysAdmin – Administrative Utilities

- Broadcast a message to all terminals
- Take a telephone message
- Send a mail message
- Read mail
- Kill a process by PID
- Kill processes owned by a user
- Kill processes attached to a TTY
- Display process status
- Copy/move a directory
- System Shutdown
- Start/stop cron daemon
- Modify crontab entries
- Change system date on local machine
- Show who is logged in

SysAdmin – Backup Utilities

- Cleanup backup log files
- Set default tape unit
- Enable terminals for login
- Full backup of all filesystems
- Incremental backup of all filesystems
- Kill users and disable terminals for login
- List backup log files
- Selective backup of a filesystem
- Verify backup tape and create log file
- Filesystem to filesystem backup

SysAdmin – Maintenance Menu

- Add a new user
- Delete a user from the system
- Modify a user's password
- Change user attributes
- Add a group
- Delete a group
- Change group attributes
- Enable a terminal
- Disable a terminal
- Add a directory
- Configure SysAdmin user

SysAdmin – Restore Utilities

- Create a list of files to restore
- Clear the file list
- Edit the file list
- Recover from errors
- Restart file list
- Restore file from tape
- Review file list
- Mail to users
- Select file from file list
- Show backup log
- Show i–node statistics for this restore

SysAdmin – Filesystem Maintenance

- Make an empty filesystem
- Filesystem restoration
- Show filesystem mounting
- Show free disk space
- Physical dump of all partitions
- Physical dump of one partition
- Physical restore of one partition
- Mount/unmount partition
- Fsck a filesystem
- Format floppy
- Duplicate diskette

SysAdmin – Tape Utilities

- Assign a tape drive
- Read from a tape
- Release a tape drive
- Rewind a tape
- Set rewind mode
- Skip a tape file
- Select the default tape unit
- Write to tape
- Write selected files to tape
- Listing of files on tape
- Tape to tape copy
- Add volumes to free pool (optional)
- Modify tape library (optional)

SysAdmin – UUCP Maintenance

- Edit UUCP system configuration
- Edit UUCP device configuration
- Edit dialcodes
- Change debug level
- Snapshot for a terminal used for polling
- Snapshot for a system polled
- Poll a system in debug mode
- Remove lock files for a system
- Snapshot of UUCP queues

SysAdmin – Security Menu

- Audit file protections
- Audit set–uid files
- Audit super user account list
- Create default file protections
- Create set–uid file list
- Create super user account list
- Review security configuration files
- Check obvious user passwords
- Perform general system audit
- User directory check

SysAdmin – Archive Menu

- Archive filesystems
- Check for files to archive
- Create list of files or directories to unarchive
- Create list of files on archive tape
- Display files on archive log
- Review list of file to unarchive
- Unarchive files from list
- Dump the archive partition on tape
- Archive user defined files

SysAdmin – Printer Maintenance

- Create new printer
- Remove printer
- Enable printer
- Disable printer
- Start printer
- Stop printer
- Start scheduler
- Stop scheduler
- Display printer status
- Display printer queue
- Print files
- Test a printer
- Cancel print requests
- Set default lp

SysAdmin – Optional Functionality

- Resource Accounting and Reporting
- Network Maintenance Menu
- Remote System Management
- NFS Maintenance
- NIS Maintenance
- Disk Quota Management

SysAdmin – Accounting and Reporting

- Disk reports per user
- Disk reports per filesystem
- Login accounting
- Command accounting
- Resource user accounting
- Resource charge-back (optional)
- Enable system accounting
- Disable system accounting
- Performance monitoring and reporting (optional)

SysAdmin – Network Maintenance Menu

- Configure hosts
- Configure networks
- Set network permissions
- Set client hosts
- Display time on remote hosts
- Change system time (remote)
- Display network host status

SysAdmin – Remote System Management

- Set client hosts
- Update clients with network databases
- Update clients with password and group databases
- Remote system software update
- Review network configuration
- Syscap maintenance
- Software list maintenance
- Update or report on all hosts and software

SysAdmin – NFS Maintenance

- Set name of host to configure
- Configure exported file systems
- Configure imported filesystems
- Inquire about exported filesystems
- Set client hosts
- Review NFS filesystem configuration

SysAdmin – NIS maintenance

- Set name of server to configure
- Print values in a NIS database
- Print selected values from a NIS map
- Change login passwords in NIS
- Which host is the NIS server
- What version of a NIS map is on a server
- Force propagation of a changed NIS map
- Transfer a NIS map from some NIS server
- Build and install NIS database
- Rebuild NIS database
- Display the map nickname table
- Redefine NIS hosts

SysAdmin – Disk Quota Management

- Perform quota check
- Turn filesystem quotas on
- Turn filesystem quotas off
- Report filesystem quota
- Inquire on quota for user
- Edit quota for user
- Edit quota for a group

SysAdmin – Tape Library

- Tape number
- Media type
- Tape title
- Filesystem
- Device name
- Hostname
- Backup level
- Usage count
- Backup date
- Tape owner
- Volume sequence number
- Maximum sequence number
- Expiration date
- Off–site

Abstract of the

Presentation by

MARINA BUITRAGO

A HELP SYSTEM FOR THE EX/VI EDITOR

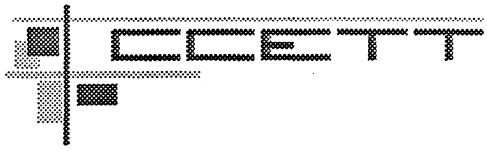
A major problem that anyone arriving UNIX finds is the editing task with the standard tools provided: ex, vi...A big component of this problem is the non-friendly and nonintuitive interface of those editor, designed years ago to satisfy some minimums, but improved up to the point that vi has been chosen as the POSIX standard editor.

Normally, the difficulty in working with vi is avoided picking an alternative, so, the newcomer to UNIX begins his/her work with a system dependent editor.And the problem arises again when the user is facing a different vendor UNIX. This is the case of using vendor dependent editors, or Windows Editor: changing the environment, changes the ability of editing.

This problem was first detected at C.I.C.A. when the first UNIX arrived on 1985, though by then the system was a minor one and the users were not many. But with the arrival of the first CONVEX, intended to be the most used system, the problem reached its maximum.

CONVEX provides a user friendly editor, COVUEedt, that is meant to be a solution for those users who come from the VMS environment, and this helped a little.However the problem still subsisted, and increased as WorkStations were installed in the campus. Then the users wanted a UNIX-standard user-friendly editor.

This presentation shows the interactive help system developed at C.I.C.A. to ease the users the beginning with vi. This help system has been mainly developed and installed in several UNIX system by Marina Buitrago. She will explain its functioning, installation and use, as well as the way it can be ported to different systems.



Alain CROUZET
C.C.E.T.T.
Tel : (33) 99 02 47 33

Email : crouzet@ccett.fr
X400 : G=Alain; S=Crouzet; P=CCETT;
A=atlas; C=FR

Gilles LEROY
INFO'ROP
Tel: (33) 61 39 02 28

Cesson-Sévigné, October, 08 1991

ESTHER

(Équipement pour la simulation en Télevison Haute définition, Enregistrement et Restitution)
Simulation Equipment for High definition Television, Acquisition and Vizualisation.

ABSTRACT:

ESTHER is a set of devices designed for research engineers development in new images coding algorithms. The systems has 3 main parts :

1 – A Convex Computer,
2 – A real-time RAM storage system,
3 – ADC and DAC video converters.

1 - Algorithm software simulation and images sequences files storage are prosseed by the convex Computer. The connection with the real-time device is achieved by an HSP/HIA subsystem.

2 - The real-time RAM based storage device process the image acquisition and vizualisation. It includes :

- A two busses Backbone with 150 MHz clock frequency each;
- Four gigabytes fast RAM (up to 16 Gigabytes expandable);
- A VMEBUS with 2 CPU for adress generation and memory load optimization (8 or 12 bits data paths);
- A computer interface for image sequences backup;
- Four digital video I/O port with 3 x 12 bits connections (up to 150 Mhz clock frequency);
- Spare slots for futures extensions;
- An Ethernet link to a workstation which provides a high level multi-window (X11-OSF/MOTIF) user interface. Two users max in real-time visualisation are allowed.

3 - The Analog to Digital Converter and the two Digital to Analog Converters whith a large amount of TV Standards :

- Sampling frequencies from 13,5 MHz to 148,5 MHz,
- 8 bit conversion in RGB or YUV (12 bits data path),
- 16/9 or 4/3 screen format,
- different sampling raster,
- interlaced of progressive,
- 50, 60, 100 or 120 Hz field frequency
- Adapted filters and time delay compensation.

All the parameters are software controlled by the workstation.

The high definition television is one of the strategic research direction for the CCETT. In particular, bit rate reduction and picture compression are necessary for the HDTV transmission with an acceptable cost. Computer simulation is a performant and unexpensive way for testing a lot of possibilities and algorithm components. This requires digital test pictures and real-time diplay system to compare original and processed sequences.
The CCETT decided to develop a HDTV simulation system, including picture converters, real-time memory unit and a high speed computer. Its name is ESTHER.

1 - INFORMATICS ENVIRONMENT:

Several computers are used by the CCETT for algorithms simulation and pictures display :

A CONVEX C210 computer is used for algorithms simulation and software development. It has a 256 Mbytes memory, 10 Gbytes disk (VME or IPI) and a HSP board for the HDTV memory system connection.Convex special system group has designed HIA-boards for the memory connection, using INFO'ROP's specifications. There are four boards, inserted in the HIA rack.

SUN workstations and X-terminal are used for login on CONVEX, software debugging (CXdb), and so on. Workstations are also used for pictures memory systems like the TRIDYN system from INFO'ROP. It has a 64 Mbytes memory, and is able to store and display in real-time 4:2:2 / 625 lines digital television sequences.

CCETT's general computers (IBM RS600) served for slow data storage through ethernet network (NFS file access) and automatic backup on a DOROTECH optical juke-box.

2 - REAL TIME MEMORY:

ESTHER is designed and produced by INFO'ROP on behalf of the CCETT. This modular system whith e high degree of hardware and software performances enables to have :

- 2 users managing at the same time the pictures,
- sequence acquisition and monitoring in real time with more than 50 video standards, interlaced or non interlaced, from 50 to 120 Hz field rate, rom 625 to 1375 lines and up to 2048 * 2048 pixels screen resolution.
- two synchronized video channels for stereoscopic uses,
- a pixel with several coding (between 8 to 36 bits) without lost of memory capacity.

HDTV video converters can be moved away via optic fibers link on a long distance range (up to 100 meters).

The four Gigabytes memory (16 Gbytes expandable) can store sufficient amount of pictures for current work.

C.C.E.T.T 4,rue du Clos Courtel - B.P. 59 - 35512 CESSON-SÉVIGNÉ Cedex FRANCE - TEL. (33) 99.12.41.11 - Télex 740 284 F - Fax (33) 99.12.40.98
INFO'ROP Buroparc 1 - voie 2 - B.P. 164 - 31676 LABEGE Cedex FRANCE - TEL: (33) 61 39 02 28 - Fax: (33) 61 39 03 17

The block diagram (see ESTHER Harware architecture) shows how the system has been subdivided into functional units interconnected by two data busses.

Since the picture memeory is modular, it enables memory capacities from 1 to 16 Gigabytes (4 Gb standard). This memory is connected to two data busses called HIBUS. Each bus is designed to allow up to 600 Mbytes data rate.

The organization and the management of pictures memory has been set up in order to managed pictures coded with 8, 12, 16, 24, 36 bits by pixel without losses of memory capacity. This allows picture management in RVB, Y-CR/CB or monochrome format.

The memory management block's processor (an high speed DSP) controls the picture memory addressing with an outflow of 1,2 Gbytes/second by block of 4096 pixels and controls the picture transportations onto the both HIBUS as well as the video multi-interface arbitration on each HIBUS. This allows a disk-type memory management.

The video converter interfaces processes speed adaptation between the picture memory and the video converter. Each interface includes :

a format converter changing the picture memory data into a pixel visual,
two video memory of 4 Mbytes x 36 bits,
a look-up table
a graphic processor and a graphic overlay,
a controller of video multi-standards,
a programmable synthesizer of pixel frequency.

The computer interface processor is designed to handle data rate up to 150 Mpixels/second. It is today limited by the HIA/HSP speed to 40 Mbytes/second.

The table, ESTHER:Memory capacaties, shows some possibilities of video standards with corresponding display time of video sequences, regarding the standard and the mode of use (color or monochrome).

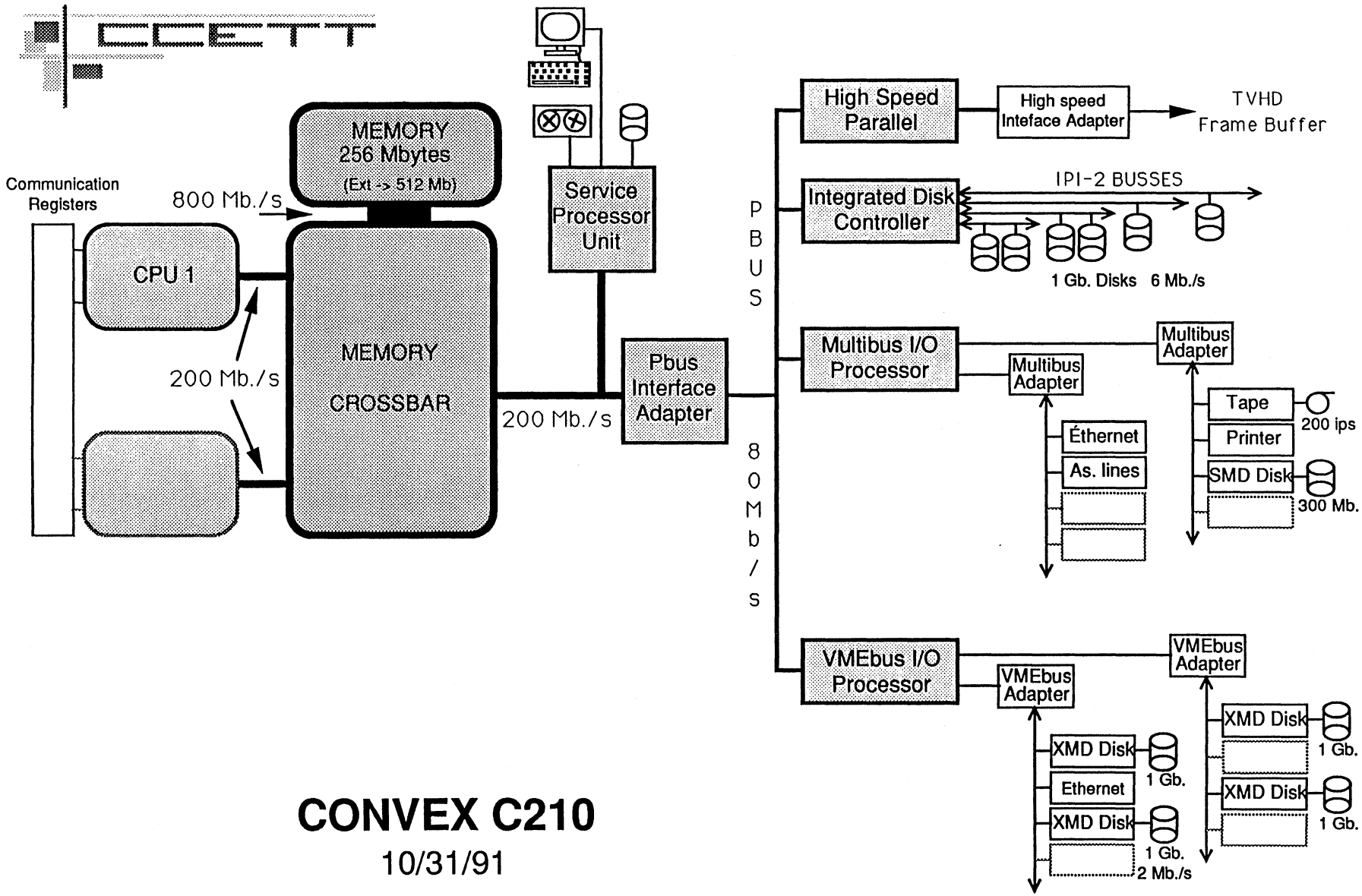
Software components

The software consists of five parts:

- the electronic boards management microprograms,
- the 68000 microprograms to supervise hardware and memory load,
- the users management software, allowing full split-screen mode, pan function controllable down to a single pixel, graphic processor management, picture zooming, ..
- The pictures file transferts software between the memory and the computer,.
- The human-computer interface using MOTIF / X11R4 multi-window software.

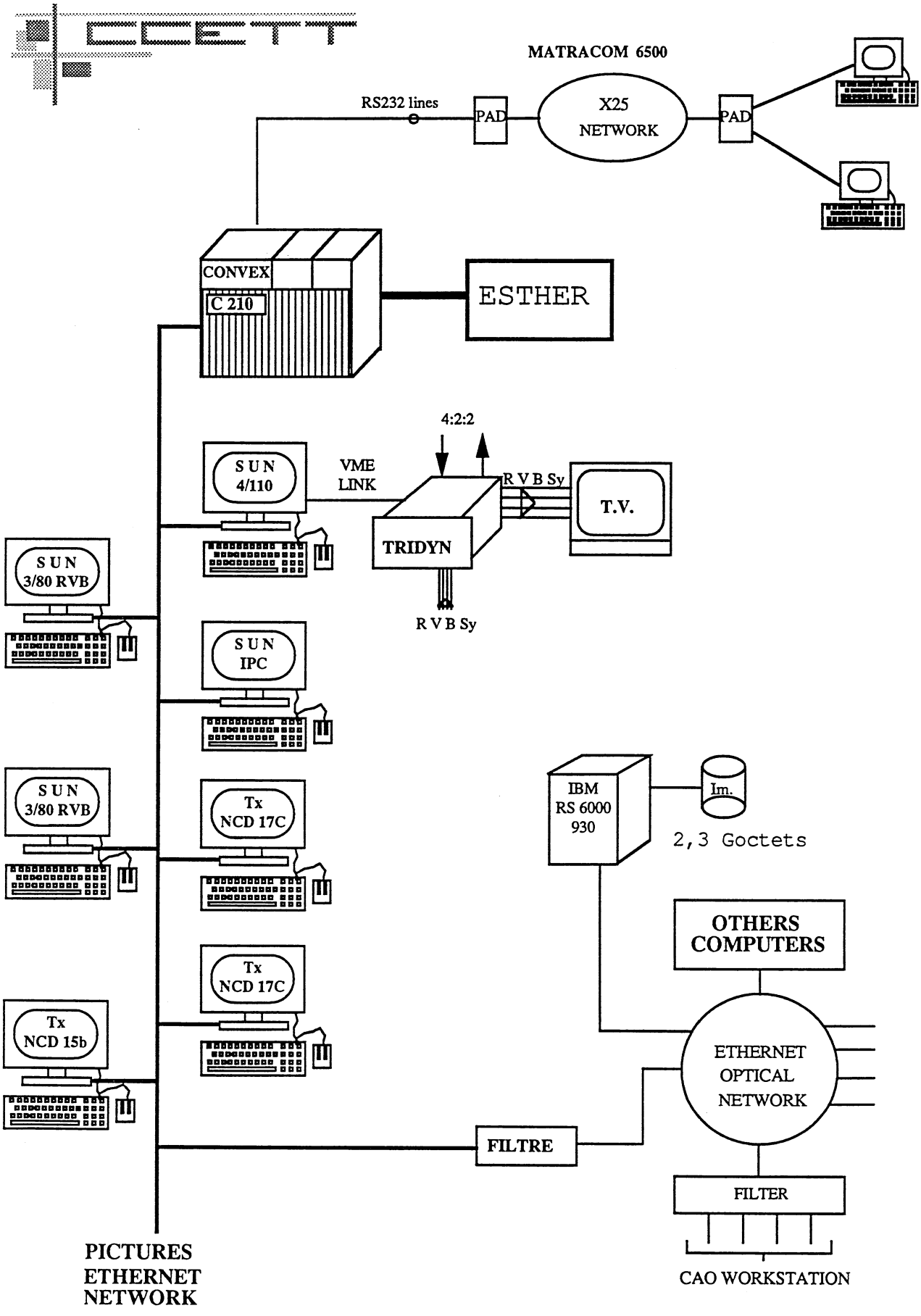
3 - VIDEO CONVERTERS:

The video converters are realized by THOMSON-CSF/LER. They can work with more than 50 video standards; They include data multiplex, video conversion, filter selection and analog matrix; the memory CPU remote controls the standard selection in the converter.
ESTHER includes two digital to analog converters and one analog to digital converter.

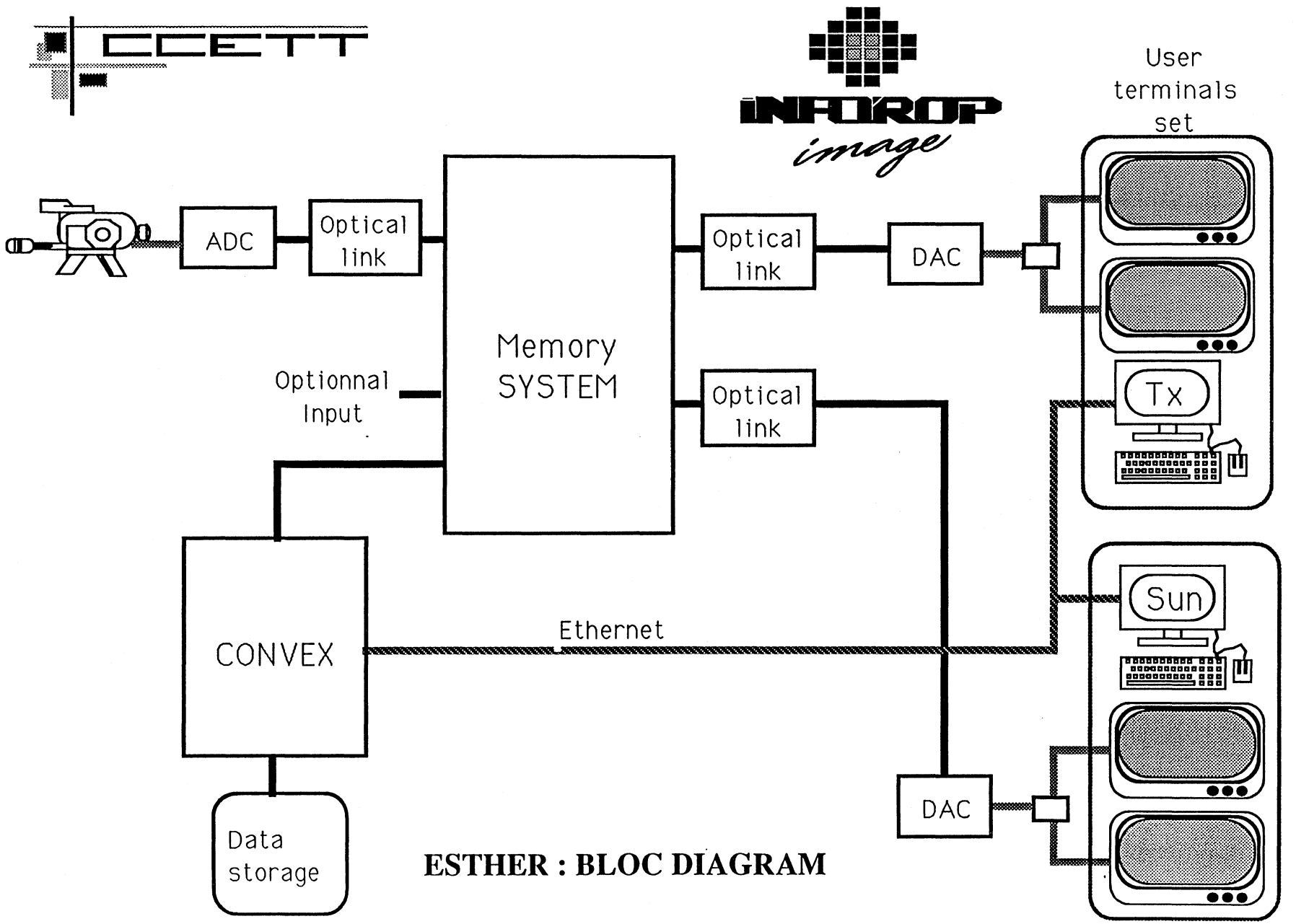


CONVEX C210

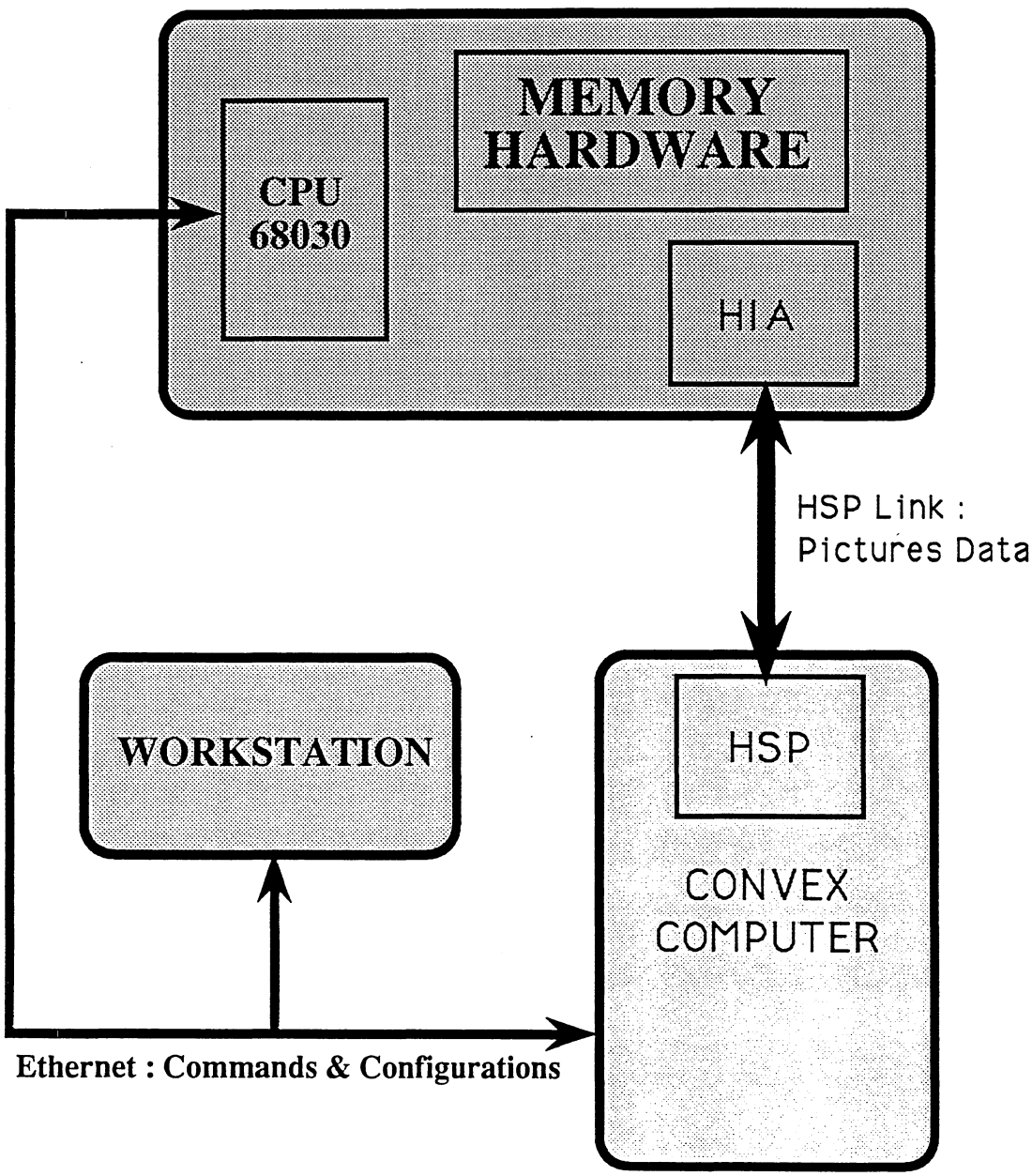
10/31/91


CCETT
MATRACOM 6500
RS232 lines
PAD
X25
NETWORK
PAD
CONVEX
C 210
ESTHER
4:2:2
SUN
4/110
VME
LINK
R V B Sy
T.V.
TRIDYN
R V B Sy
SUN
3/80 RVB
SUN
IPC
SUN
3/80 RVB
Tx
NCD 17C
IBM
RS 6000
930
Im.
2,3 Goctets
OTHERS
COMPUTERS
Tx
NCD 17C
Tx
NCD 15b
ETHERNET
OPTICAL
NETWORK
FILTRE
FILTER
PICTURES
ETHERNET
NETWORK
CAO WORKSTATION

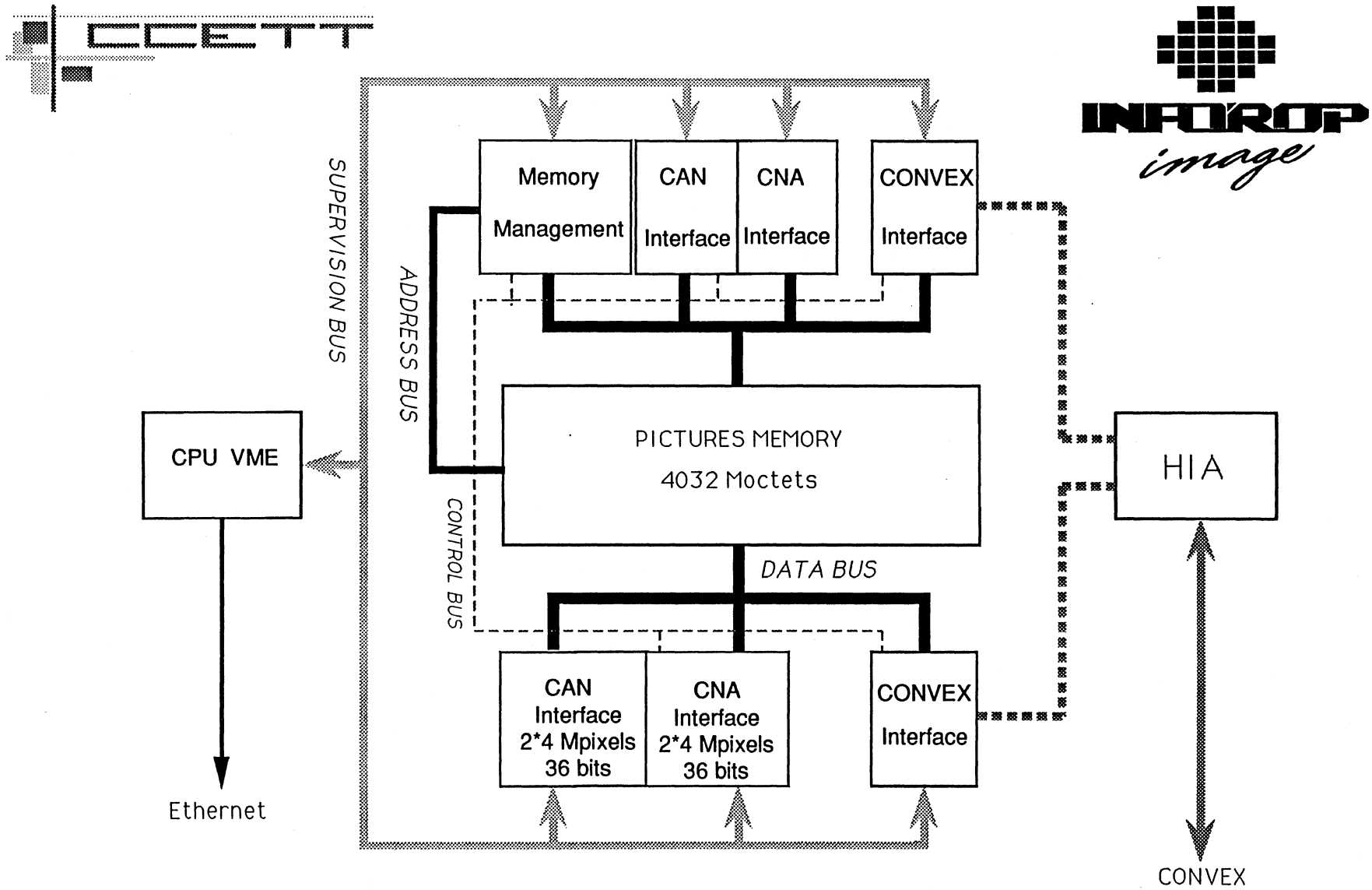
Oct. 1991



ESTHER : BLOC DIAGRAM



ESTHER
COMPUTER LINKS

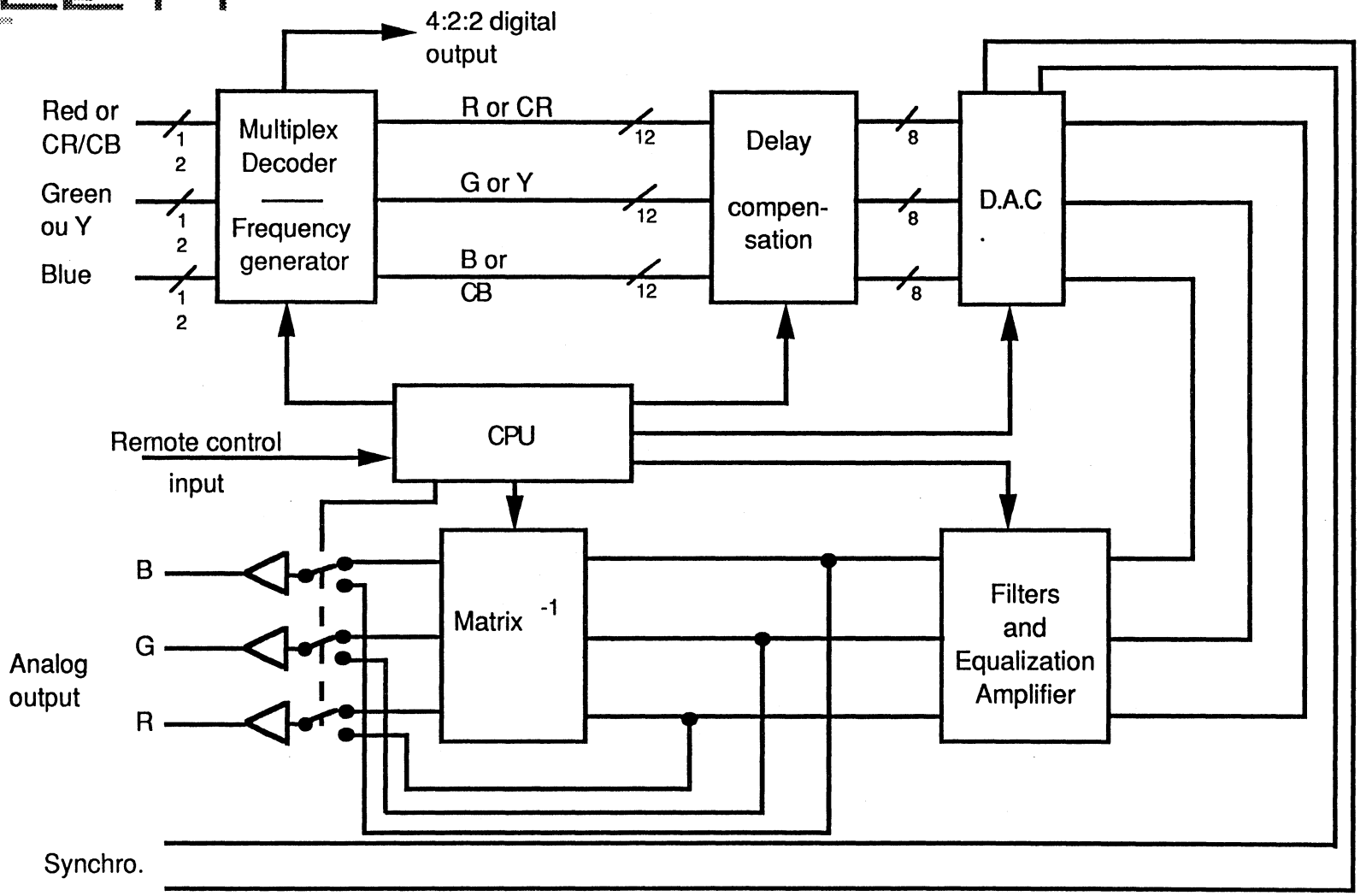


ESTHER : Hardware Architecture

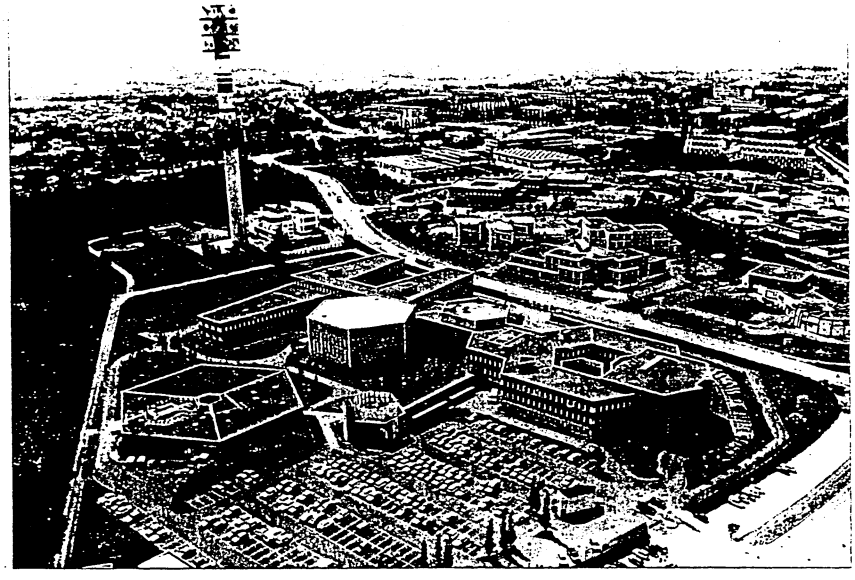
Sampling Frequency Mhz	Interlace ratio	Screen ratio	Stored Picture format (pixels*lines)	Max sequence lenght			
				RGB 12 bits	RGB 8 bits	Y,Dr/Db 8 bits	Y only 8 bits
13,5	50:2	4/3	720*576	1 mn 24	1 mn 56	3 mn	6 mn 11
13,5	60:2	4/3	720*486	1 mn 19	1 mn 49	2 mn 49	5 mn 48
72	50:2	16/9	1920*1152	17 s	23 s	36 s	74 s
72	60:2	16/9	1920*972	16 s	21 s	33 s	69 s
144	50:1	16/9	1920*1152	8 s	11 s	18 s	37 s
144	60:1	16/9	1920*972	8 s	10 s	17 s	35
148,5	120:2	16/9	1920*1035	7 s	10 s	16 s	33

ESTHER : MEMORY CAPACITIES

(4 Gbytes; examples)



ESTHER : DIGITAL TO ANALOG CONVERTER
BLOC DIAGRAM



THE JOINT RESEARCH CENTRE FOR BROADCASTING AND TELECOMMUNICATION TECHNIQUES (CCETT)

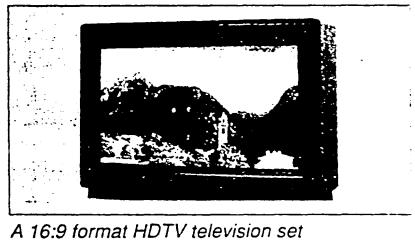
CCETT is a research centre which is taking an active part in the great advances being made in ***audio-visual*** *techniques and* ***telematics*** *both in France and in the rest of the world. Founded in Rennes in 1972 and with a present staff of 400, it has been run since 1983 as a joint venture between* ***the National Telecommunications Research Centre and Télédiffusion de France.***

Located in the heart of the **RENNES ATALANTE Science Park**CCETT works in conjunction with its local partners. Developing these projects with manufacturers from the region remains one of CCETT's main aims and many outside research contracts have been awarded to manufacturers who go on to produce prototypes and production equipment.

CCETT concentrates its research in the following areas:

WIDE BAND SERVICES AND NETWORKS: high definition television, pay-television services (the EUROCRYPT and VISIOPASS standards), D2-MAC packet improved quality television, videotelematic services, processing and bit-rate and/or band reduction of images and digital sound, high quality digital audio broadcasting ...

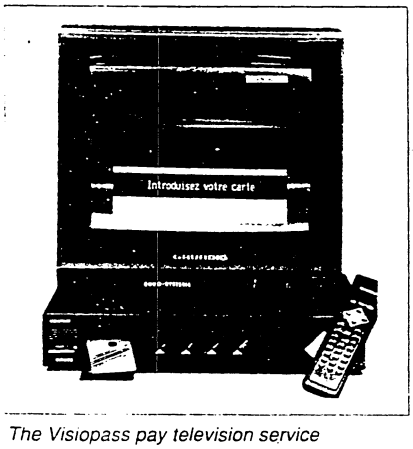
TELEMATICS AND MULTIMEDIA SERVICES (interactive or broadcast): «Minitel», multimedia videographics on ***NUMERIS*** (ISDN), public access terminals, telematics transmission to mobile receivers, remote control surveillance on CCTV.



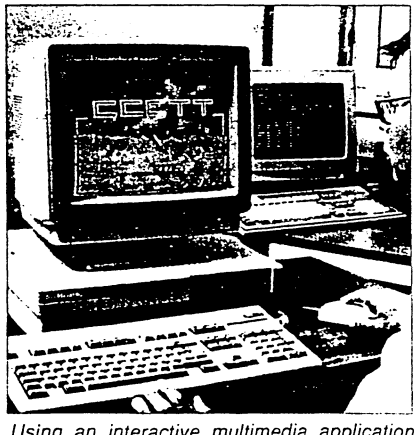
A 16:9 format HDTV television set

In conjunction with universities and schools of engineering, CCETT offers a large number of students the opportunity to carry out work placements in the Centre's laboratories. At the same time, some of the Centre's engineers have external lecture programmes and CCETT organized seminars bring together engineers and researchers from the public and private sector.

In every sphere of research, CCETT can be found actively promoting French innovations in international standard setting organizations as well as in European R & D programmes (***ESPRIT***, ***RACE***, ***EUREKA***..).



The Visiopass pay television service



Using an interactive multimedia application designed for the Albertville Winter Olympics

CCETT

4, rue du Clos Courtel - B.P. 59
35512 CESSON-SÉVIGNÉ Cedex
tél : (33) 99 02 41 11 - fax : (33) 99 02 40 98

INFO'ROP is a French industrial group of about 150 persons, qualified and experimented scientists and engineers, working in the field of :

Computer science, simulation, electronics, project management, digital image processing, artificial intelligence, neural networks.

Our main are customers involved in Aerospace, Defence, Telecom, Energy, etc...

DIGITAL IMAGE PROCESSING

INFO'ROP Image, department founded in 1986, is specialized in the design, manufacturing, integration and distribution of Image Processing Systems (hardware and software).

We offer competence in :

Image Workstations

TRIDYN Image Workstations have a modular memory up to 512 Mb, real colour visualization with a maximum resolution of 1536 x 1152 pixels. Our software includes general image processing toolbox (VINCI) as well as image sequence management (SCENARIO), with dedicated versions (CALYOPE for remote sensing for example).

Claude TORRES,
Toulouse Agency, Tél : (33) 61 39 02 28

Design and development of SPECIFIC SYSTEMS

We are currently developing a more powerful system with a memory of 16 Go dedicated to HD TV within an EUREKA project.

Gilles LEROY,
Toulouse Agency, Tél : (33) 61 39 02 28

RESEARCH

Our research center, INFO'ROP Advanced Technologies, has for mission to be present in scientific circles, and to lead innovative studies for industrial groups.

Neural Networks and Information Processing, Fuzzy Logic

Gérard YAHIAOUI
Paris Agency, Tel : (33) (1) 46 13 01 60

Artificial Intelligence and Image Interpretation, Advanced Databases

Jean-Louis AMAT
Paris Agency, Tel : (33) (1) 46 13 01 60

WE ARE LOOKING FOR

Industrials, universities, research centers with competence in :

- Parallel systems
- Experts systems
- Networks and peripherals
- Thematic applications

Coordination : **Jean-Jacques LEFEBVRE,**
Paris Agency, Tel : (33) (1) 46 13 01 60

THOMSON - CSF, LABORATOIRES ELECTRONIQUES DE RENNES

GENERAL PRESENTATION

THOMSON-CSF, LABORATOIRES ELECTRONIQUES DE RENNES (TCSF-LER) is a corporate research laboratory of THOMSON-SA, specialising in the design and development of advanced electronic imaging systems in line with the basic activities of the THOMSON Group in the areas of defence, telecommunications, professional and consumer television and high definition television.

Since its foundation in 1973, TCSF-LER has developed outstanding expertise in image and signal processing areas, covering all functions of the "image chain" from origination and transmission to display whatever the nature of the image : video, computer synthesized, infrared, radar.

TCSF-LER's main research fields are :

- Image analysis and display systems,
  Analogue processing for TV and HDTV images,
  High performance cameras,
  Analogue to digital and digital to analogue conversion.

- Modelling (image sources, live scenes, etc...) and environment simulation,

- Algorithms for :
  . image processing : motion detection and estimation, bit rate reduction, noise reduction
  . scene understanding, pattern recognition, image interpretation, computer aided vision.

- Real time parallel processing architectures and associated software.

- Video coding and digital processing :
  . base band coding and associated processing equipment
  . encryption functions for conventional television
  . coding and multiplexing of TV and HDTV signals for transmission (MAC, HD MAC standards)
  . standards conversion
  . bit rate reduction functions for digital transmission (codecs) and recording.

- Multirate networks for home distribution,
- Optical fibre transmission (both digital and analogue) and wide band switching systems,
- Human system interface for efficient (realtime) operational requirements,
- Digital modulation and signal processing.

TCSF-LER has extensive picture processing simulation facilities based on high performance computers and high throughput recording and display systems.

In most cases, TCSF-LER's research activity results in prototype implementation and small series of advanced equipment designed with the latest technology through its internal computer aided design and engineering facilities.

As of mid 1991, TCSF-LER is staffed with nearly 300 people, 80 % of them are highly skilled engineers and technicians, and has successfully applied for 200 patents.

TCSF-LER participates strongly in several international standardisation activities as well as European collaborative research programs (e.g. ESPRIT, RACE) and EUREKA industry driven projects (e.g. EU 95 HDTV, EU 147 PROMETHEUS).

-I-I-I-I-I-I-

FE-SIMULATION AND VISUALIZATION IN GROUNDWATER FLOW AND GROUNDWATER POLLUTION USING HIGH PERFORMANCE SYSTEMS

W. Haas, M. Resch and R. Brantner
JOANNEUM RESEARCH
Institute for Information Systems
Steyrergasse 17, A-8010 Graz, Austria

1. ABSTRACT

The numerical simulation of groundwater flow and groundwater pollution using the finite element method is considered - due to the type of mathematical equations that have to be solved and due to the transient nature of the problem - to be among the most time consuming applications of high performance systems.

The technological progress of supercomputers, minisupercomputers and superworkstations, accompanied by a significant decrease in prices, has made these systems available to a wider number of users at universities, research centers and industrial enterprises.

On one hand most of the simulation programs used on these systems are implemented as pure batch systems without any possibilty of interaction for the user. On the other hand it is no question that in the future it will be necessary to concentrate on graphic user interfaces, interactive data manipulation and on-line visualization of the simulation process.

As an example for the realization of a program considering the above mentioned requirements the simulation of groundwater flow and pollution transport in groundwater by means of the FEM is presented. Programming concepts for this application taking advantage of vector architecture and utilizing OSF/Motif and X-Windows for the graphic user interface are presented to show a possible solution for an interactive and user friendly FE-program.

2. INTRODUCTION

During the last few years the employment of numerical modeling techniques has spread to almost every branch of science and has also become an important tool for decisionmakers in the field of managing groundwater systems. Today hydrologists are being caught in the middle between some major advances in science and increasing pressure from legal and regulatory bodies to use models to provide answers to specific questions. The explosion in knowledge during the last 10 years in this field of research has forced them to numerically model and solve increasingly complicated problems requiring more and more computational power.

Due to the technological progress in the field of hard- and software development, accompanied by a significant decrease in prices, new powerful technologies like vector processing and parallel processing have become available to a wider range of users. The development of high level software however, was not yet able to keep pace with this trend. Therefor most of the programs currently in use on these new supercomputers are not able to take full advantage of the underlying hardware because they were designed for Von-Neumann architectures.

As a consequence there is a growing need for newly developed algorithms and programs to fully exploit the power of vector- and parallel technology. Furthermore it is widely recognized that it will be necessary to concentrate on graphic user interfaces, interactive data manipulation and on-line visualization of the simulation process in order to make appli-

cations more user-friendly. The separation of calculation and visualization and their allocation to different hardware systems - each adopted for optimum performance - will increase the number of users dealing with numerical simulation.

As an example for a program taking into account the demands mentioned above the simulation of groundwater flow and pollution transport in groundwater by means of the FEM is presented. For the computational part of the program a totally new code with a new data structure has been written to take full advantage of the vector and parallel architectures of modern supercomputers. Programming concepts for this application utilizing X-Windows for the graphic user interface are presented to show a possible solution for an interactive and user-friendly FE-program.

3. STANDARDS

In software development the use of industry-standards is the only way to keep pace with new hardware improvements. Furthermore it is the only way to establish compatibility between different computer environments. Among a number of international standardization commitees (e.g. ISO, IEEE, CCITT) the X/Open organisation has gained high importance, whose goal it is to adopt and to adapt existing industry and de facto standards into a common applications environment. Main subject of the X/Open portability guide (XPG3, 1987) are operating systems, networks, communication, data management, programming languages and user interfaces.

3.1. Networks

Due to the spreading of distributed data processing, communication has become a key factor for the overall performance of a complex information system. Supercomputer experts have already issued the slogan "The network is the computer !" (Stapleton, 1991).

To enable communication between different hardware platforms, ISO has created the 7 layer OSI model (Stallings, 1988). Based on this model Ethernet (IEEE 802.3) has been developed working at a speed of 10Mbit/s. For visualization the standardized FDDI (Fiber Distributed Data Interface) will offer sufficient speed of 100Mbit/s while for real time visualization and particularly for animation problems the standardization of networks with higher transfer rates (e.g. HIPPI, Ultranet) will be necessary.

3.2. Vector and parallel processing

So far no industy standard has evolved for the hardware of high-speed computers. However several common development trends can be indentified that are briefly described in the following.

During the past decades the performance of computer systems has grown by an order of magnitude every five years. Due to physical limits this evolution cannot continue in the future. (Noor, 1988) The introduction of parallel computing, where several activities are carried out simultaneously, is a possibility to overcome these limitations at least for the next ten years. In the field of parallel computing several possibilities of implementation have been carried out.

One model is the use of multiple functional units in a single cpu. An example for such an approach is vector processing which can be described as the simultaneous processing of several independent data streams on a single processor. It has proved to be highly efficient for a number of scientific problems dealing with vectors and matrices (Gentzsch, 1989)

Another approach is the use of multiple cpus in one system. In this field computers are usually referred to as shared memory or distributed memory machines. Normally, in shared memory architecture a moderate number of cpus is used whereas in distributed memory machines up to 65K processors may be employed.

Besides these two possibilities of parallel processing heterogeneous networks of computers are becoming more and more commonplace in high-performance computing, where systems ranging from workstations to supercomputers are linked by high-speed networks

(Beguelin et al. 1991). This approach has proven to be useful with respect to cost-effectiveness and performance.

3.3. User-interfaces and graphics

To implement distributed visualization and graphic userinterfaces on such systems, the vendor independent X11 standard is an appropriate basis. Among a number of different user interfaces relying on this standard, OSF/Motif seems to be a reasonable choice. Based on X11 it is supported by many workstations and recommended by X/Open. As a high-level programmer interface, GKS (with 3-D extensions) and PHIGS may be considered. However only versions based on X11 should be used for distributed applications.

3.4. Visualization

During the last years visualization and animation of results have become major tasks in the field of supercomputing. Visualization is a method that has emerged from and profited by the combination of the latest advances in visualization software, graphics, networking, high performance systems, and industry standards.

Among other products AVS (Application Visualization System) has gained more and more importance and today has become a de facto standard. It is a full-functionality visualization environment providing many techniques of visualization and is available today or in the near future on most major computing platforms.

3.5. Software development environment

In order to reduce the software development cycle time computer supported methods have become more and more important. The so-called CASE-tools (Computer Aided Software Engineering) usually support all phases of software design and development and often include toolkit-libraries, library managers and other usefull utilities.

The selection of programming languages for scientific applications on high-performance systems is not really a matter of discussion, as only FORTRAN and C are widely accepted. Today most software products dealing with numerical simulation of physical or chemical processes are written in FORTRAN. During the last time the use of C has offered the possibilty of object-oriented programming in this field of research (Forde et al., 1990) and has become more important.

4. MATHEMATICAL MODEL

Mathematical models, used in ground water studies are an attempt to describe processes by mathematical equations. The starting point in modeling is a true understanding of the processes involved.

In order to determine the effects of a pollutant source in a groundwater field, in a first step nonlinear and time dependent saturated and unsaturated flow have to be computed. In a second step the actual motion of the pollutant is calculated. Especially in this step for the finite element method many elements and small time steps must be employed in order to fullfil given stability criteria and to avoid oscillations.

4.1. Groundwater flow

For the description of the flow of ground water one mainly needs to consider the flow in response to hydraulic potential gradients and the loss or gain of water from sinks or sources (GWM, 1990). Considering the fundamental mass balance equation and taking into account Darcy's law the time-dependent flow of ground water can be described by the equation of motion

$$\nabla \cdot K \cdot \nabla\Phi = S_0 \, \partial\Phi / \partial t$$

where K denotes the hydraulic conductivity, Φ denotes potential and S_0 is specific storativity. Due to the time-dependent nature of the problem the numerical solution of this equation

requires high computing effort, which can be further increased by model assumptions that have to be adopted when more complex ground water flows are described. In many cases modeling of realistic situations requires the iterative solution of the problem.

The adequate calculation of the ground water flow is an important precondition to be able to model the transport of contaminants in ground water.

4.2. Groundwater pollution

In the case of contaminant transport, according to literature (Bear, 1987; Kinzelbach, 1987; GWM, 1990), a much larger number of diverse and often complicated processes has to be considered. These processes can mainly be divided into two different groups: (1) those responsible for fluxes and (2) those responsible for sources and sinks for the material. Mass fluxes are prompted by processes like advection, diffusion and mechanical dispersion. Sources and sinks are provided by a number of chemical, nuclear and biological processes. Among them are sorption, ion exchange, oxidation/reduction, radioactive decay, and biodegradation.

For an ideal tracer only the mass flux phenomena have to be taken into account. Mass flux by advection q^c can be described as

$$q^c = c\,u$$

where c is concentration and u is water velocity. The flux by mechanical Dispersion q^m and diffusion q^* can be described by Fick's law.

$$q^m = -D^m \cdot \nabla c\,; \quad q^* = -D^* \cdot \nabla c$$

D^m and D^* denote the coefficients of dispersion and diffusion respectively.

Considering also the second group of phenomena, as has to be done for most known pollutants, yields the advection-dispersion equation.

$$\theta\,\partial c/\partial t = -\theta\,\nabla\cdot(c\,u - D^*\cdot\nabla c - D^m\cdot\nabla c) - f + \theta\rho\Gamma - P\,c + R\,c_R$$

f describes sources or sinks caused by adsorption or desorption (e.g. decay, Freundlich-adsorption or Langmuir-adsorption).

$\theta\rho\Gamma$ describes the influence of chemical processes yielding an increase or decrease of pollutant in the water.

P and R are included to describe artificial sources and sinks respectively.

The discretization of this equations using the finite element method yields a system of non-linear equations that has to be solved numerically by using either direct or iterative methods.

5. INTERACTIVE FE-SIMULATION

Nowadays it is generally accepted that direct interaction between the computer and the user (HCI, Human-Computer Interface) is of superior importance. Therefore many FE-applications do already offer graphic capabilities, but they are mostly restricted to pure postprocessing. In the case of nonlinear and/or time-dependent problems it is often necessary to visualize intermediate results interactively in order to check the status of the computation. Furthermore co-processing offers the possibility to modify parameters interactively in order to achieve better convergence or to perform parameter studies. The program FEJUX (Finite Elements of JOANNEUM Under X-Windows) serves as an example for the implementation of an FE-code offering these possibilities.

5.1. Design goals and adopted standards

In the design process of the program a number of important parameters have been identified for future applicability:

- optimization of the algorithm and data organisation for vector computers and moderat parallel systems; preparation of data organisation for massively parallel systems.
- portability of the application by using industry standards.
- user friendliness by using a standard windows environment.
- preparation of application for processing on heterogeneous networks.

5.2. Performance analysis

From previous studies (Haas and Schweiger, 1989) it is kown that an adaption of older FE-codes to vector computers is a tiresome process ending up in non-optimum performance. Thus it was decided to completely redesign and rewrite the basic FE-library for FEJUX. Only in this way organisation of data and code suited vector and parallel processors best.

For conventional FE-programs the solution of the equation system and the visualization of the results require most computing effort. Today most hardware vendors offer specifically designed solvers that are especially tuned for their machines. And as visualization is running in parallel with computation in FEJUX element based computations (e.g. the value of Φ in Gauss points from nodal values) are predominant and time critical for rapid display of results. Therefore, to get an insight into the behaviour of the program and to test its performance on a vector processor, a list of routines has been chosen, that have to be performed for each element of the domain. A comparison of the runtime to perform these operations for a fixed number of elements was carried out using a new (FEJUX) program and a commercially available one (MISES3) that had not been written with vector processing in mind.

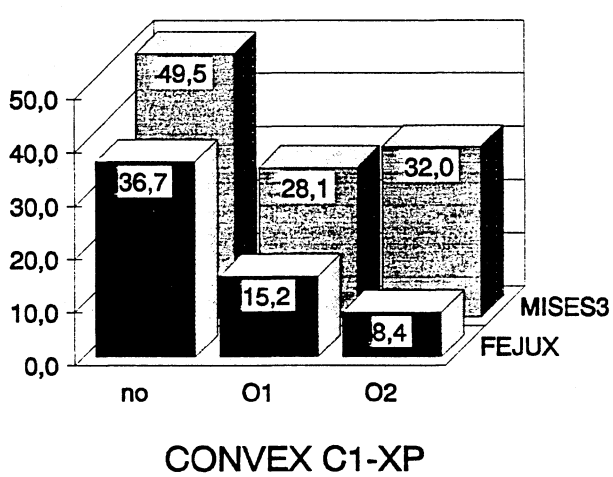


Fig. 1: Comparison of runtime between new (FEJUX) and commercial FE-program (MISES3)

The results shown in Figure 1 were obtained on a CONVEX C1-XP under ConvexOS8.1 and F77 V6.0. The indices no, O1 and O2 refer to the compiler options on CONVEX for scalar unoptimized (no), scalar optimized (O1) and automatically vectorized (O2) code. The graph proves the validity of our assumptions and makes clear that a redesign of the compute intensive part of the program is worth doing. The newly developed code is twice as fast even in scalar mode and four times as fast in vector mode, whereas the performance of the old code even degrades for vector mode. This effect could be caused by the use of short vectors and inappropriate data organisation.

5.3. Portability and distributed processing

Special emphasis was laid on the aspect of software portability. The use of the program netCDF for binary storage of results and information on the status of the calculation enabled the portability of graphic information and results among any kind of hardware platform without need of an additional software interface.

As the development of software packages for use of a heterogeneous network of computers (e.g. PVM, Linda) goes on, much emphasis was layed on enabling the application to fully exploit the capabilities of this kind of supercomputer performance. The program was therefore designed to be distributed among a network configuration enabling distributed computation and visualization. In Figure 2 a typical configuration of a heterogeneous network is given, representing a part of JOANNEUM RESEARCH's network.

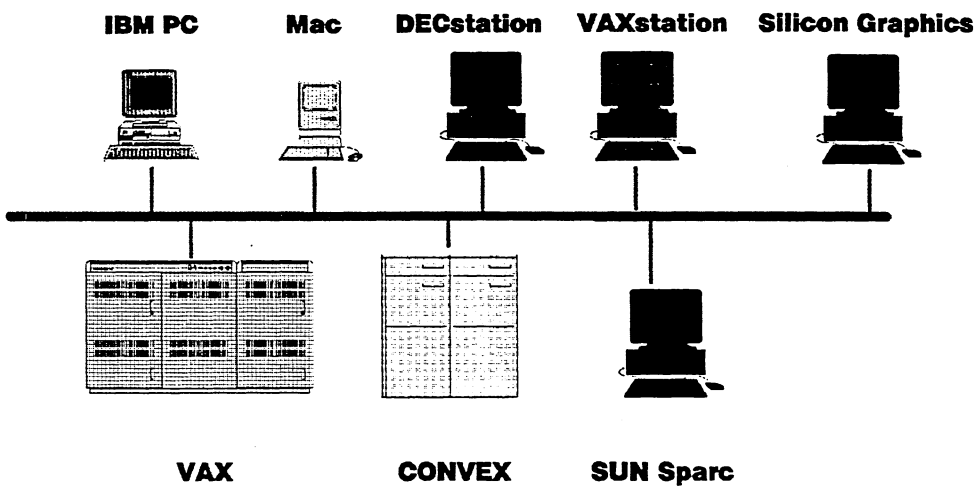


Fig. 2: Part of JOANNEUM RESEARCH's network used for distributed FE-applications

The program may be run on any kind of vector server with OSF/Motif support, while any X11 server from PCs and Macs up to UNIX workstations may be used for graphic input and visualization.

6. CONCLUSION

Due to the technological progress in hardware high-performance systems of different kind like vector computers, parallel computers or heterogeneous networked workstations become at the same time more powerful and affordable for a wider range of users. Computer scientists agree in their opinion that today the crucial point of making high performance power available does not lie in the hardware but in the development of software tools and applications to exploit as soon as possible the performance that is only theoretically provided today.

Furthermore it is widely undoubted that in numerical simulation to achieve acceptance not only the scientific quality of a program but also its userfriendliness - that today can only be achieved by graphical user interfaces - is a crucial point. It is only in this way that specialists of all scientific disciplines, from geologists to environmental researchers, from physicists to biologists will be able to effectively work with up-to-date simulation software.

Finally by adhering to industry standards, programs have to remain portable, and by utilizing advanced CASE tools, cost and time during their development have to be minimized. The methodology developed here is by no means restricted to the FE-simulation of groundwater

flow and pollution. It will in the future be extended to problems of rock and soil mechanics and can also be applied to virtually any other method, based on discretizing partial differential equations by finite elements, boundary elements, finite volumes or finite differences.

7. ACKNOWLEDGEMENTS

The numerical experiments were carried out using the finite element code MISES3. The authors gratefully acknowledge the permission of TDV-Technische DatenVerarbeitung Graz, to utilize the code for the performance tests.

8. REFERENCES

"X/Open Portability Guide", 3rd Edition, Prentice Hall, New Jersey; 1989.

L. Stapleton, "Meet the Metacomputer: Where the Network Is the Computer", Supercomputing, Vol 4, 3; 1991.

W. Stallings, "Handbook of Computer Communications Standards", Vol 1-3, Macmillan Publishing Co., New York; 1988.

A.K. Noor, "Parallel Processing in Finite Element Analysis", Engineering with Computers, 3; 1988.

W. Gentzsch, "Implementation of Algorithms and Programs on Vector and Parallel Computers", Proceedings of the first International Conference on Application of Supercomputers in Engineering, Southampton, September 1989, ELSEVIER , Amsterdam - Oxford - New York - Tokyo; 1989.

A. Beguelin, J. Dongarra, A. Geist, R. Manchek, V. Sunderam, "Opening the door to Heterogeneous Network Supercomputing", Supercomputing, Vol 4, 9; 1991.

B.W.R. Forde, R.O. Foschi, S.F. Stiemer, "Object-Oriented Finite Element Analysis", Computers & Structures, Vol 34, 3; 1990.

"Ground Water Models, Scientific and Regulatory Applications", National Research Council, National Academy Press, Washington D.C.;1990.

J. Bear, A. Verruijt, "Modeling Groundwater Flow and Pollution", D. Reidel Publishing Company, Dordrecht/Boston/Lancaster/Tokyo; 1987.

W. Kinzelbach, "Numerische Methoden zur Modellierung des Transports im Grundwasser", R. Oldenbourg Verlag GmbH, München; 1987.

W.Haas, H. F. Schweiger, "Viscoplasticity and Vectorprocessing", in S. Pietruszczak & G.N. Pande (eds.), Numerical Methods in Geomechanics NUMOG III, ELSEVIER, London; 1989.

Convex Visualization Update

European Convex User Conference
October 10, 1991

CXwindows V3.0

Deliverables:

- **New features of X11R5**
  - New clients
  - Font server
  - Input methods
- **The X extension mechanism**
- **3D graphics extension to X (PEX)**
- **Phigs C and FORTRAN bindings on top of PEX**
- **OSF/Motif V1.1**

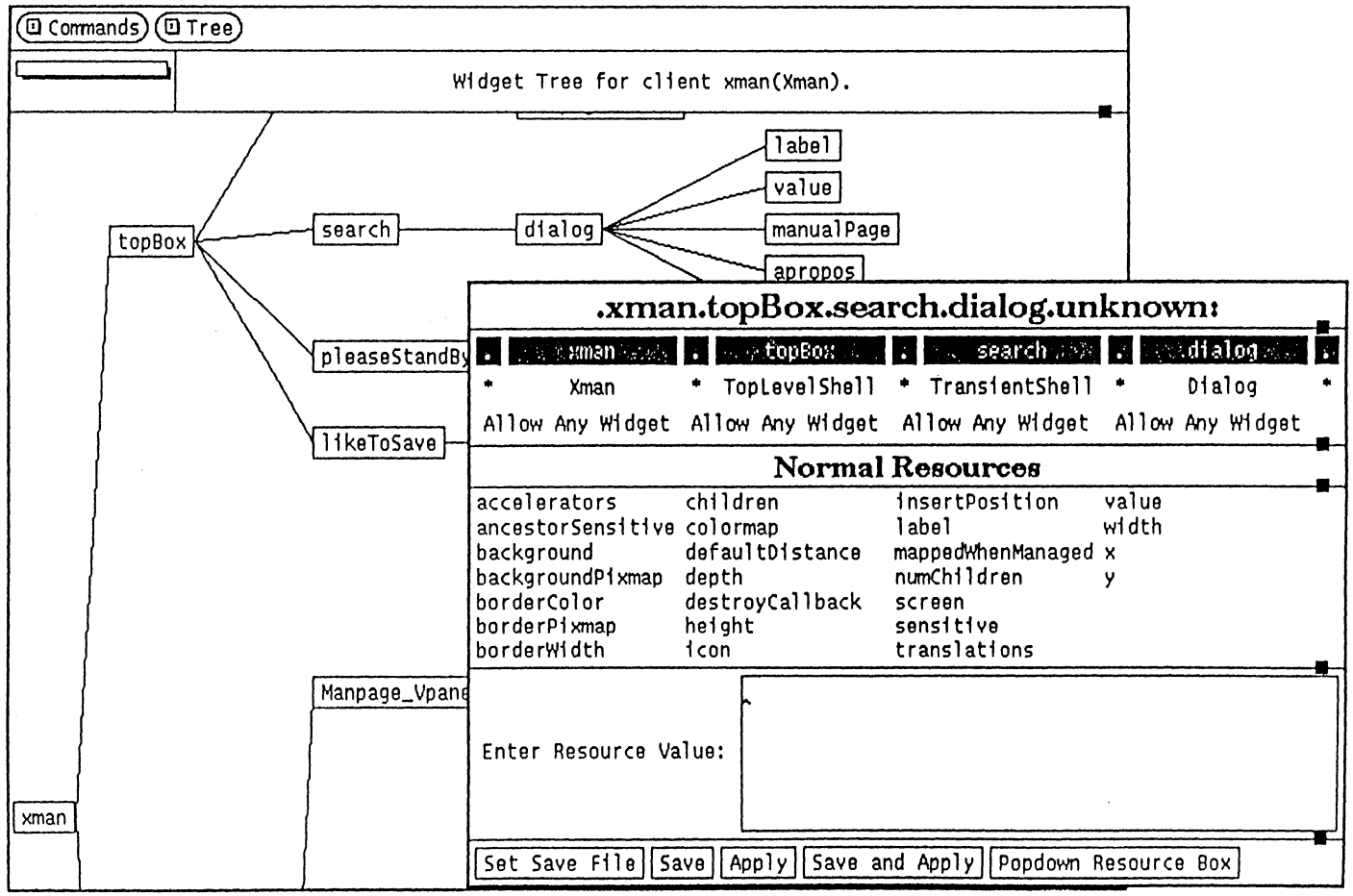
X Version 11 Release 5

New core clients

- Viewres widget viewer
- Editres resource editor
- Tekcms color manager
- Xconsole console monitor

X Version 11 Release 5

editres



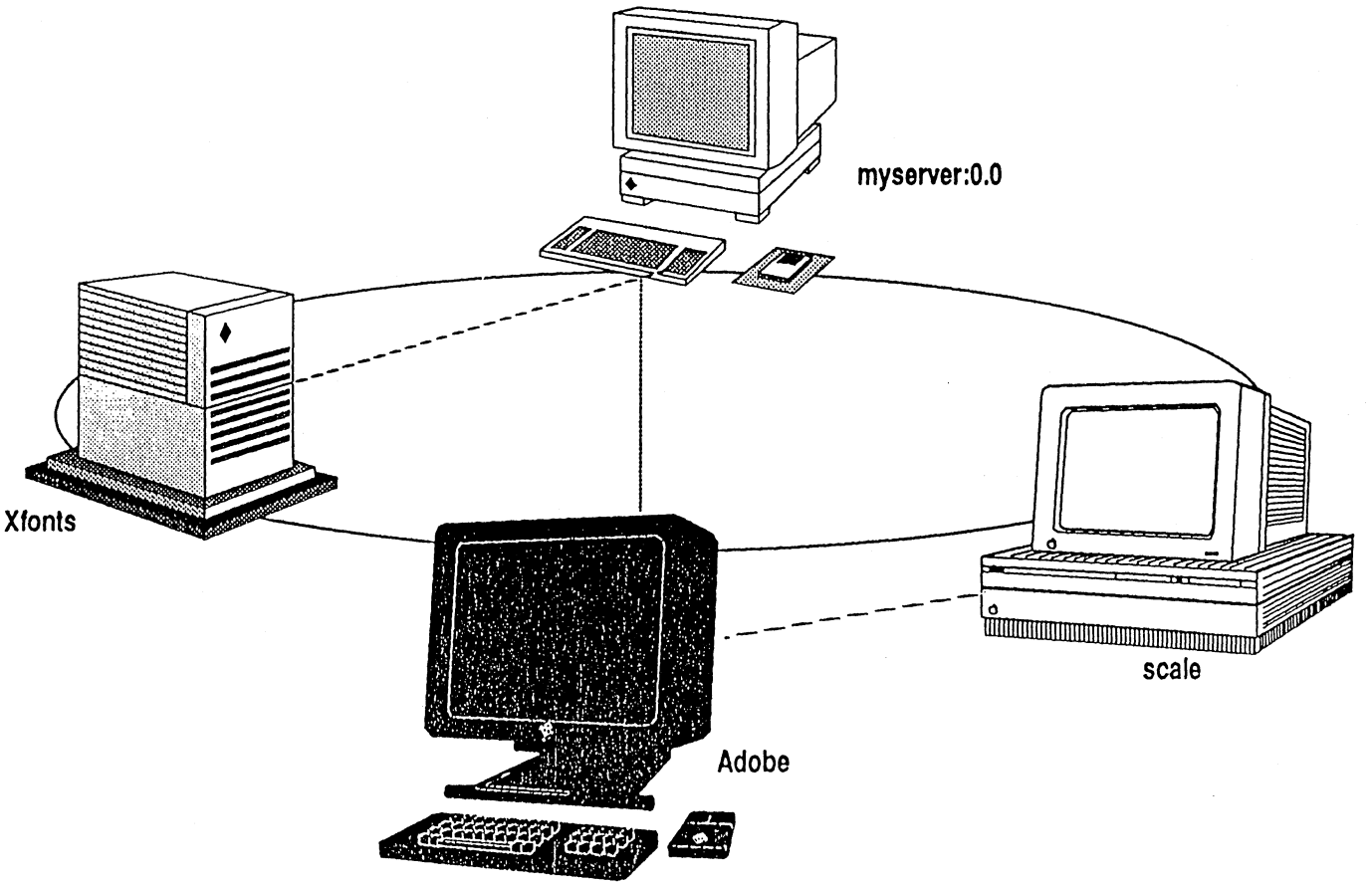
X Version 11 Release 5

Font server

- **Server-side fonts are a problem**
- **No support for other font formats**
- **Font server features:**
  - Can reside on any client machine
  - Can be duplicated on many hosts
  - Can support scalable fonts
  - Can be chained
- **Font server path similar to font path**

X Version 11 Release 5

Font server



X Version 11 Release 5

Input methods

- **Need to input more than English**
  - Support compose input method
  - Support context-sensitive rendering of strings
  - Support ideographic systems
- **Input methods are large systems (dictionaries, filters, input method)...**
- **... Each client cannot embed the input method**
- **Separate input server**

X Version 11 Release 5

Input Method Specification

- **Includes three input method areas**
  - Status area
  - Pre-edit area
  - Auxiliary area
- **Supports four interaction styles**
  - On-the-spot
  - Over-the-spot
  - Off-the-spot
  - Root-window

Distributed 3D Graphics for X

What is PEX?

- **PEX is an extension to the X Window System**
- **PEX is a protocol (like X)**
- **PEX has primitives for 3D graphics**
- **PEX stands for the 3D graphics extension to X**
  - Efficiently support PHIGS/PHIGS+
  - But flexible enough to support other 3D programming APIs, including immediate mode graphics APIs

Distributed 3D Graphics for X

Why PEX?

- Desirable to have portable, distributed 3D graphics
- Desirable to support 2D and 3D graphics in a window system
- Desirable to have an extensible, long-lived 3-D programming interface
- PHIGS/PHIGS+ has wide recognition, stable features

Distributed 3D Graphics for X

PEX Design Criteria

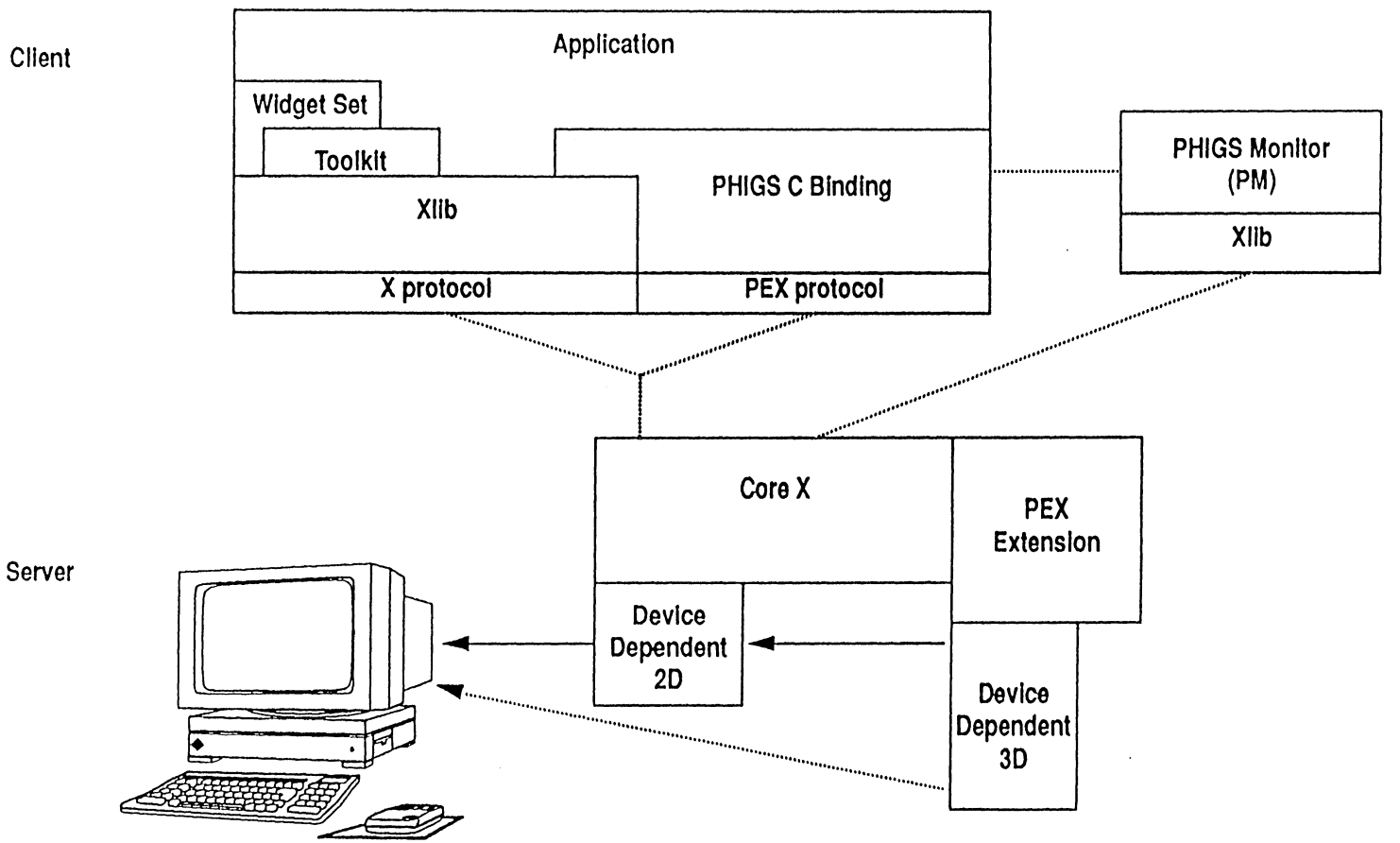
- Each window is a independent PHIGS workstation
- Support either client- or server-side structure storage and traversal
- Registered X extension (introduces no new X events)
- Simple

Distributed 3D Graphics for X

What can PEX do?

- 2-D and 3-D primitives and attributes (PHIGS)
- Hidden Line and Hidden Surface Removal (HLHSR)
- Advanced curve and surface primitives (PHIGS+)
- Lighting and shading (PHIGS+)
- A variety of clipping mechanisms
- A variety of coordinate spaces, including modeling coordinates

PEX Client/Server Model



Distributed 3D Graphics for X

PEX protocol subsets

- PHIGS Workstation Only
  - —Support for server-side structures
  - —PEX Workstation
- Immediate Rendering Only
  - —Support for client-side structures
  - —PEX renderer
- Mixed Mode

Distributed 3D Graphics for X

PEX protocol status

- **Version 5.0P complete**
  - PEX Sample Implementation complete
  - Server/PHIGS API complete
  - 5.0P publicly available at no cost from the MIT X Consortium as a part of the X11R5 distribution
- **Versions 5.1P and 6.0P currently in X Consortium definition**
- **Convex is active in the**
  - X Consortium PEX-spec subcommittee
  - X Consortium PEX-client subcommittee
  - Multi-vendor PEX Interoperability Committee
  - Multi-vendor PEX Interoperability Center

CXterminal-19C Features

Available now:

- 19" 8-bit color X-terminal
- Unix keyboard
- 3-button mouse
- 1280x1024 resolution
- 72 Hz refresh
- 8MB ram
- Switchable thin/thick ethernet connections

X Window System

Futures

- Imaging extension protocol (XIE)
- Multimedia, video (VEX) extensions
- PEX 5.1P and PEX 6.0P
- PEXlib for 5.1P and 6.0P
- Session management
- Object-oriented toolkit

X Window System

CONVEX Futures

- **CONVEX is a member of X Consortium**
- **Expand CXwindows to include additional clients**
- **PEX is a pivotal technology for CONVEX**
  - Distributed 3-D graphics is real
  - Viable, competitive alternative to GL
  - Widely accepted
  - PEX Interoperability Center at CONVEX

PEX Interoperability Center

Mission Statement:

- To promote the creation and widespread use of PEX-based products

Objectives:

- Increase awareness and understanding of PEX among the press, developer and end-user community
- Provide facilities to develop and verify the interoperability of PEX-based products
- Act as a user group to communicate PEX issues to the appropriate X Consortium committees
- Establish conventions for PEX interoperability among participant companies
- Demonstrate vendor long-term commitment to PEX-based products and technology

ConvexAVS V3.0

- **Robust visualization environment**
- **Provides**
  - Menu-driven visualization applications
  - Prototyping tools for developing visualization techniques
  - Extensible programmer's interface
- **Requires no new code to be developed**
- **Can be used in many scientific fields**
- **International AVS Center begins operations**

ConvexAVS V3.0 Features

- **Compatible with Stardent AVS 3.0**
  - All standard Stardent 3.0 modules and viewers
  - Some Stardent 3.5 features
  - Irregular fields, Unstructured Cell Data (UCD) types
- **Supports three kinds of displays:**
  - Color X Window System servers (8- and 24-bit)
  - PEX servers
  - Silicon Graphics GL servers
- **CONVEX-specific performance improvements**

ConvexAVS V3.0 Features

- **Remote module execution**
- **Graph Viewer**
  - Manipulate linear and contour plots
  - Application or module
- **AVS Animator Toolkit**
  - Produce visualization animation sequences
  - High-quality image renderer
  - Set key frames, interpolate between frames
  - Record/playback sequences
  - Output video to recording device (SGI VideoCreator, Diaquest ImageNode)

ConvexAVS V3.0 Features

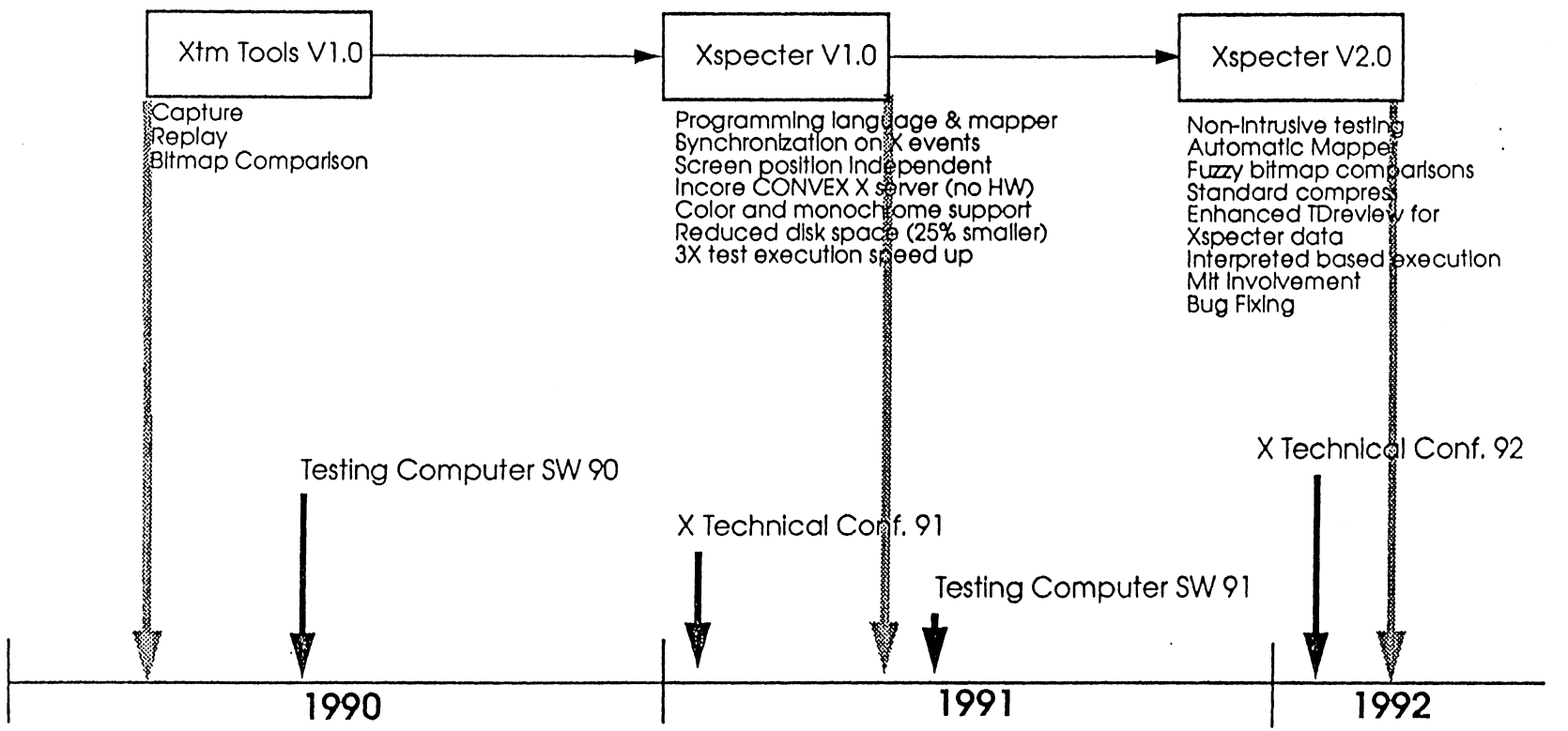
- **50 new modules**
  - antialias
  - tracer/display tracker
  - probe
- **More efficient memory usage**
- **Gaussian post-processor**
- **Extensive online-help and new ConvexAVS online tutorial**
- **User and Reference Documentation**
- **Visualization Concepts Guide**

International AVS Center

Charter:

- **"Free-ware" module repository**
  - Anonymous ftp access to module repository
  - Email ordering of modules via internet
  - Postal mail ordering of modules on magnetic tape at cost
  - Easy-to-follow module submission criteria
- **Hosts AVS Users Group**
- **Edits AVS Network News quarterly magazine**
- **"comp.graphics.avs" newsgroup**
- **AVS training and education**
- **AVS promotion to industry and government**

Interactive Test Technology



XTM Tools

Features:

- Able to capture, replay and verify sessions
- Works on a CONVEX and sun workstation
- Supports keyboard and mouse input
- Records user session exactly as executed
- Easy for anyone to make recordings

Problems:

- Can not adjust recordings when UI changes
- Difficult to read (i.e. uncommented assembler)
- Unreliable playback due to changing machine conditions (i.e. load)
- Records all actions even errors
- Requires SUN HW to execute tests
- Difficult to modify and maintain

Xspecter V1.0

Convex-Internal Deliverables:

- X event driven programming language
- Mapping tool - Separates User Interface from recording (position independent)
- Synchronizes on X events not wall clock time
- Testcases are now programs not recordings
- Testcases are easy to read and modify
- Testcases are compiled into an executable
- Saves disk space
- Speeds up test execution time by 3X
- Incore X server requires no HW (saves $$)

Problems:

- Requires instrumented X server (source code)
- UI changes require a new X client map
- Does not resolve problems caused by new images
- Images are very big and require lots of disk
- No run time decision support(i.e. if X then y)

Xspecter V2.0

Convex-Internal Deliverables:

- Non-intrusive X testing client
- Automatic mapping tool based on editres
- Fuzzy bitmap comparisons
- Standard compress support
- Enhanced TDreview for Xspecter testcase reviewing
- Interpreted execution (support for run time decisions)
- General bug fixing and support

Benefits/features:

- No longer require source code for X servers
- Reduces disk space requirements
- Enhances ability to run when images are similar but different
- Provides tools for easily browsing Xspecter results and updating tests
- Automatic mapper makes updating X client maps easy

A SuperComputing Environment in Climate Research

Hartmut Fichtel

Deutsches Klimarechenzentrum GmbH
Bundesstr. 55
D-2000 Hamburg 13
Federal Republic of Germany

ABSTRACT

DKRZ (Deutsches Klimarechenzentrum - German Climate Computer Centre) is a mainly federally funded institution to provide the climate research community with the necessary computing resources to develop and run large scale numerical climate models. These global circulation models include the relevant physical subsystems like the atmosphere, the world oceans, and the ice shields. The necessary integration times for climate studies range from years over decades to centuries. This is in contrast to operational weather forecasts which typically involve an integration time of some days to a few weeks.

The main characteristics of climate modelling thus are virtually unlimited needs for compute power in terms of currently available machinery. Further needs for computational power will be caused by the desirable increase of grid resolution together with the inclusion of lower scale physical effects instead of parametrization. Strongly correlated to the available computational power is the amount of model data generated which has to be adequately stored and post-processed.

The development of the DKRZ computing systems into the current structure will be described as driven by both the scientific needs and the development of the available technology in computing machinery and mass storage equipment. An outlook into the near future will be attempted.

Introduction

The popular notion of climate involves an average look on current weather variables over a certain interval of time, as e.g. air and sea surface temperature, cloudiness, sunshine and rainfall in particular geographical locations. This is associated with the familiar distribution of climate types, as seen in different geographical areas and over season. It is generally recognized meanwhile that possible changes in climate may have a tremendous effect on both economical and ecological conditions for the human society, as e.g. in the context of average air temperature and amount of precipitation for agriculture, or a possible rise in average sea level caused by either an increase in sea temperatures and/or melting of polar ice caps for coastal regions.

Additionally it has been recognized that changes in climate state may not only occur by the natural climate variability but also caused by anthropogeneous influences e.g. by the substantial burning of fossil fuels for transport and energy production mainly by the industrialized countries of the northern hemisphere and/or the wide area deforestation in tropical areas which lead to the well known greenhouse effect. It is this prospect or anticipation of a changing climate that is largely responsible for the recent increase of both public and governmental interest in the climate question.

The Climate System

Climate is a particularly complex physical problem because many interdependent physical processes create its structure and variability. Intuitively the most prominent subsystem is the atmosphere, but climate is also largely influenced by the average behaviour of the oceans, the conditions of the world's ice masses, and the state of the land surface and its associated vegetation. Each of these components is linked together into a global system, with changes in one subsystem possibly affecting the behaviour of the other components thus creating a chain of events which may either cancel or reenforce the original event in a negative or positive feedback reaction.

The Atmosphere

The central component of the climate system is the atmosphere which displays a spectrum of conditions. The global distribution of climatic zones shows the warm and moist regions of the low latitudes, generally warm and drier climates in the subtropics, the familiar temperate areas in the mid latitudes up to the generally cold and dry climates in the higher latitudes.

The basic driving force of course is the energy flux caused by solar radiation which is depicted in Fig.1.

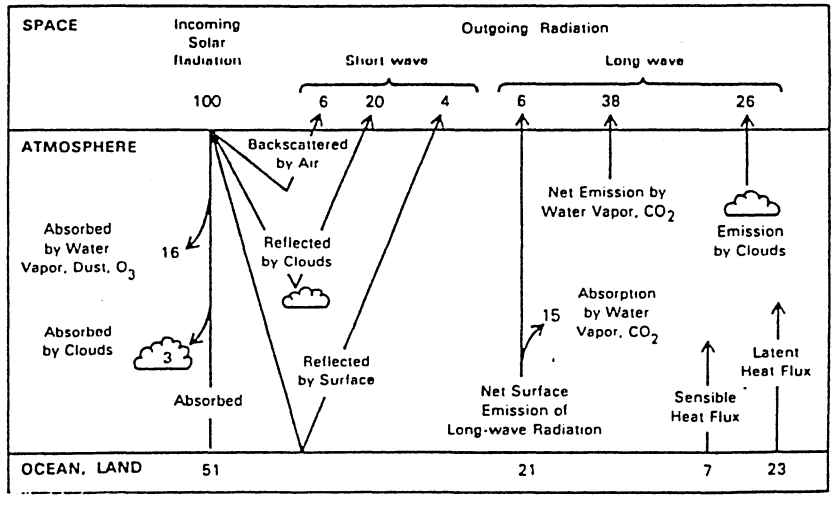


Fig. 1: The average global radiation and heat balance of the climate system (From National Academy of Sciences, 1975)

The global heat balance is maintained by a number of feedback processes involving the transfer of energy between the atmosphere, the clouds, and the surface. On average as much energy has to leave the top of the atmosphere by long wave radiation as enters in the form of solar short wave radiation.

One of the characteristic properties of the atmosphere is the ease with which it can be heated and set into motion. The characteristic time scales of the atmosphere's response to changes in the external forcing are from a few minutes to hours in the case of local convective motions up to a few days in the case of large-scale transient cyclones and anti-cyclones of the mid-latitudes. Other familiar properties of the atmosphere are the afternoon maximum of temperature and, in many locations, of convective clouds and rainfall which are caused by the daily variation of the surface heat budget generated by the earth's rotation, and the characteristical seasonal changes of the mid latitudes. In general, these daily or seasonal phenomina display larger amplitudes in the atmosphere than do their counterparts in other climate subsystems. So the atmosphere is characterized by relatively large synoptic- and seasonal-scale fluctuations around the climatic mean.

The Hydrosphere

Another important component of the climate system is the hydrosphere which is coupled to the atmosphere by various different physical processes as shown in Fig. 2.

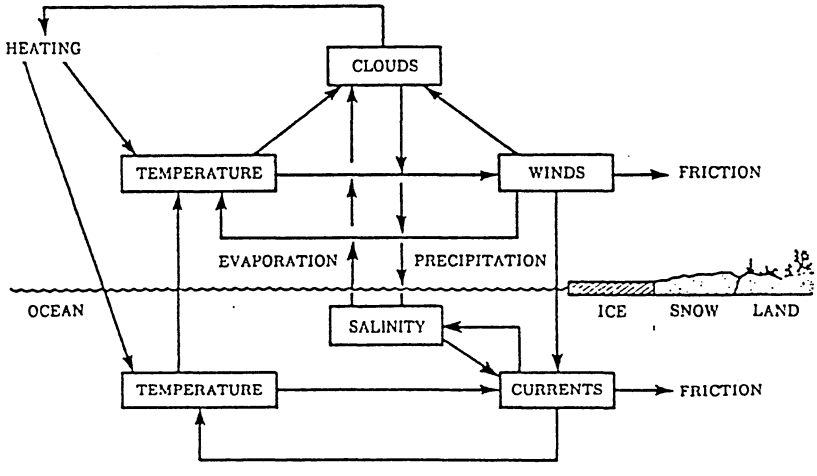


Fig. 2: Major components of atmosphere-ocean feedback processes
(From National Academy of Sciences, 1975)

Since the oceans cover two thirds of the earth's surface, most of the solar radiation falls onto the oceans where it is absorbed by the uppermost few meters of water. Because of the high specific heat of water this results in a relatively small change in the ocean's temperature compared to

which occurs over land. The oceans therefore act as a huge heat reservoir, moderating the dynamics of the atmosphere. The energy is slowly transported by global ocean currents from the equatorial and tropical regions towards the colder mid-latitude and polar regions.

While a portion of the energy stored in the oceans is transferred through conduction and advection into those regions where the sea surface temperature is higher than that of the overlying air, a larger portion is used to evaporate water into the atmosphere. This latent heat is subsequently released upon condensation of the water vapor into clouds and precipitation, and is a major heat source for the atmospheric circulation on both small and large scales.

The time scales which characterize ocean responses vary considerably depending on the depth of the water masses involved. The surface mixed layer in approximately the top fifty meters responds to changes of surface heating in timescales of days to weeks, wheras the deeper water which comprises the bulk of the oceans responds far slower due to its relatively large thermal and mechanical inertia, and reacts to changes of its surface conditions in timescales of decades to centuries.

The Cryosphere

This climatic component consists of a portion closely associated with the oceans (sea ice) and portions associated with the land surface (snow, glaciers, and ice sheets). The importance of the cryosphere to the climate system is mainly caused by its high reflectivity or albedo and its low thermal conductivity. In the northern hemisphere a considerable portion of the surface is covered by snow and ice each winter, while in the southern hemisphere the ice pack surrounding the antarctics undergoes a dramatic wintertime expansion. In addition to these seasonal changes significant variations occur in the cryosphere over much longer periods. In response to gravity a mountain glacier tends to move slowly downwards and outwards and can thus considerably change its shape and extent in timescales of centuries. Generation of ice sheets also occurs in continental dimensions on both the northern and southern hemispheres in timescales of tens of thousands of years or more, depending on whether the climate - which the ice shields themselves influence - is favorable or unfavorable for their maintenance.

The Lithosphere

The surface lithosphere is in contrast to the other subsystems a rather passive component of the climate system in all but geological timescales. Even though the daily and seasonal variations of temperature over land exceed those over water, the physical characteristics of the surface soil and rock are usually taken as fixed in the determination of the climate.

The Biosphere

The biologically active components of the climate system interact with the other components on timescales that are characteristic of their life cycles. Most prominent among these is the seasonal cycle of plant growth which influences the surface albedo and heat flux. Additionally, these plant cycles play a major role in the global carbon dioxide cycle which underlies the greenhouse effect.

Climate models

One of the main goals of climate modelling is to develop realistic models of the coupled climate system, comprised of the 4 principal components atmosphere, hydrosphere, cryosphere, and biosphere, which can be applied for climate prediction and for climate response studies. A necessary requirement for a realistic climate model is that it be capable of reproducing not only the mean climate, but also the observed climate variability as well as the response of the climate system to changing external influences. One of the major problems in this area is the considerable timescale difference of the various climate subsystems involved. A summary of the characteristic timescales and space domain of the 4 main climate subsystems is depicted in Fig. 3.

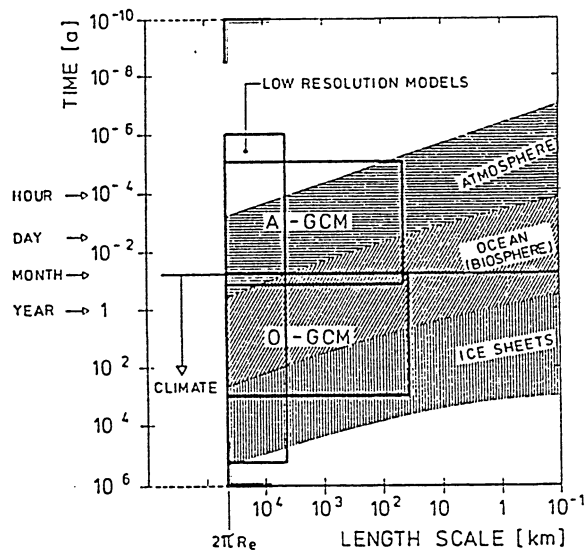


Fig. 3: Characteristic space-time scales of the climate subsystems
(From Hasselmann, 1986)

The boxes in the figure denote the areas in the space-time domain which today can be reasonably modelled using the most powerful computer systems available. The most striking feature is the comparatively small area covered by any model box compared with the space-time region over which the full climate system extends. For any given model box, all physical processes with scales outside the box which are smaller than the smallest scale modelled explicitly must be parameterized. These scales are normally then assumed to be in a statistical quasi-equilibrium state relative to the system in the model box. The larger scales outside the box are treated as frozen boundary conditions.

The computational requirements of climate models clearly depend mainly on the number of grid points (or wavenumbers in the case of spectral models) and thus the resolution. Due to the fact that with decreased grid point distance (possibly in all 3 dimensions) the integration time step has to be decreased accordingly, these computational requirements tend to grow with the fourth power of decreasing grid point distance.

Computing Requirements

The first global modelling activities of climate subsystems began within the Max-Planck-Institute for Meteorology in Hamburg around the years 1984-85 with most efforts centered on global ocean models including the global carbon dioxide cycle. Similar activities regarding the atmosphere have been carried out by the Institute for Meteorology at the University of Hamburg. These were simulations of single climate subsystems.

MPI/Met 1985-1988

In April 1985 the institute computer center installed its first high performance vector computer system to run climate simulations: a CDC Cyber-205 model 412 (2 vector pipes resulting in 200 MFlops peak (64 bit) and 1 MW of central memory initially) that was strictly run in batch mode. Access to the system was possible via frontend systems: a CDC Cyber 860 first running NOS and later NOS/VE and a VAX cluster running VMS operating mainly in timesharing mode leading to the classical 3-layer structure of these times, as depicted in Fig. 4.

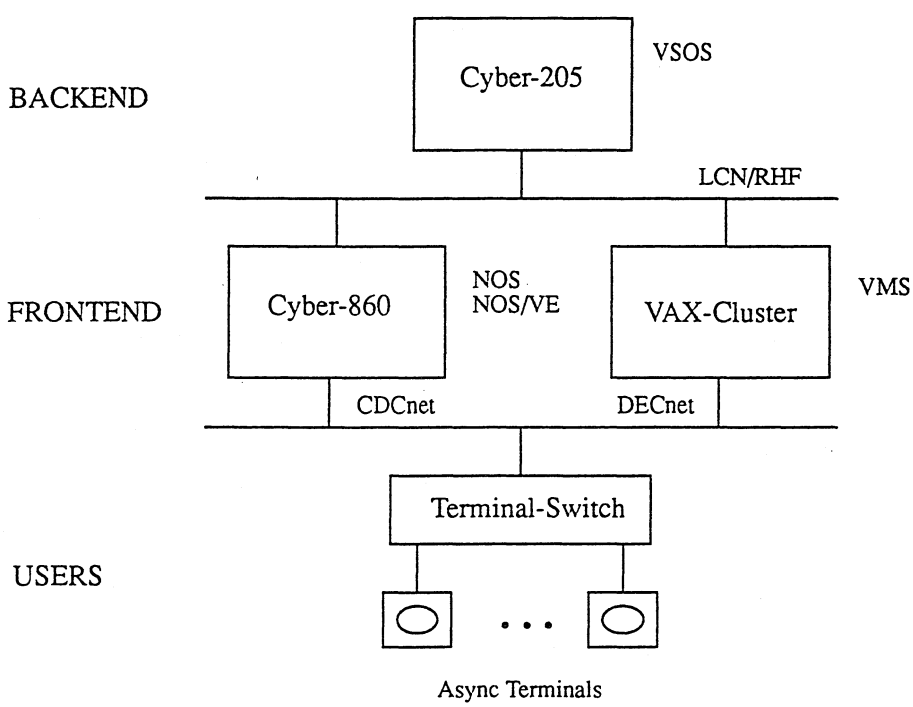


Fig. 3: Frontend/Backend structure of MPI/Met computer center in 1985

The local disk space was a total of 2.4 GBytes and both the backend and frontend computers were equipped with magnetic tape drives being served by human operators. Since there were only 2 operator shifts on 5 days per week serious problems were encountered in keeping the backend systems from idle cycles due to "disk full" situations and/or frontend unavailability. All operating and network software were proprietary and dependent on the hardware vendors. Interactive access to the frontend systems typically occurred via asynchronous terminals. Graphical output devices (plotters) were managed centrally, and graphical preview stations were as rare as expensive.

DKRZ 1988 - 1991

After discussions of some length in the years 1982-1984 between German scientific institutions interested in climate research (in particular the MPI/Met) and the German federal research administration on how to best organize and set up the organizational environment to adequately support and perform numerical climate research, a German Climate Programme had been set up by the federal government. One of the decisions was to organizationally delink the computer center from the Max-Planck-Institute and to form a legally independent institution: DKRZ - the German Climate Computer Center. The location remained the same though, and the existing staff was transferred to the new institution. DKRZ was designed as a national center to serve not only the existing customer base concentrated in the Hamburg area, but to extend its services to the whole German climate research community. These organizational changes were completed in 1987.

At the same time funds were made available to replace the Cyber-205 by a successor system with enough system thruput to start simulation of the climate system, i.e. the relevant climate sub-systems coupled together. Five main arguments directed the selection process of the successor system:

(1) in order to enable the new system to run simulations of the complete climate system in reasonable elapsed times a system thruput of 8-10 times that of the Cyber-205 was considered necessary and sufficient,

(2) the central memory (which meanwhile had been upgraded to 4 MW) was another bottle-neck, so a similar increase ratio of 8-10 was desirable for CM,

(3) the available disk space had been a most serious bottleneck, so the new system was to be equipped with fast disks of capacity between 20-50 GByte,

(4) since the amount of model data generated is strongly correlated to model resolution and integration time, some new solution had to be found with respect to mass storage capacity. The solution could obviously not be based on tape drives and human operators that work in 2 shifts for 5 days per week, so an automatic way of disk data migration without operator intervention had to be realized. So all possible supercomputer bidders were requested to support an automatic tape loader system with tape drives directly connected to the channels of the new system.

(5) the frontend/backend structure with the production system (backend) strictly in batch mode had been identified as one of the major obstacles to increase the human productivity of the scientists involved with computing. To enable both batch access for production model execution and interactive service for short test runs, interactive debugging and housekeeping purposes the new system had to provide both the hardware ability and the software facilities to enable a modest amount of interactive activities.

After intensive benchmarking and system evaluation the choice fell towards a Cray-2S 4/128 with 4 processors, 1 GByte of (static) central memory and equipped with 25 GByte of local disk space initially. A software cooperation contract was initiated in order to support the STK4400

Autoloader system with STK 3480 type transports connected to the BMX channels and the STK control software HSC running on a small dedicated IBM system under VM/SP.

The resulting configuration is depicted in Fig. 5.

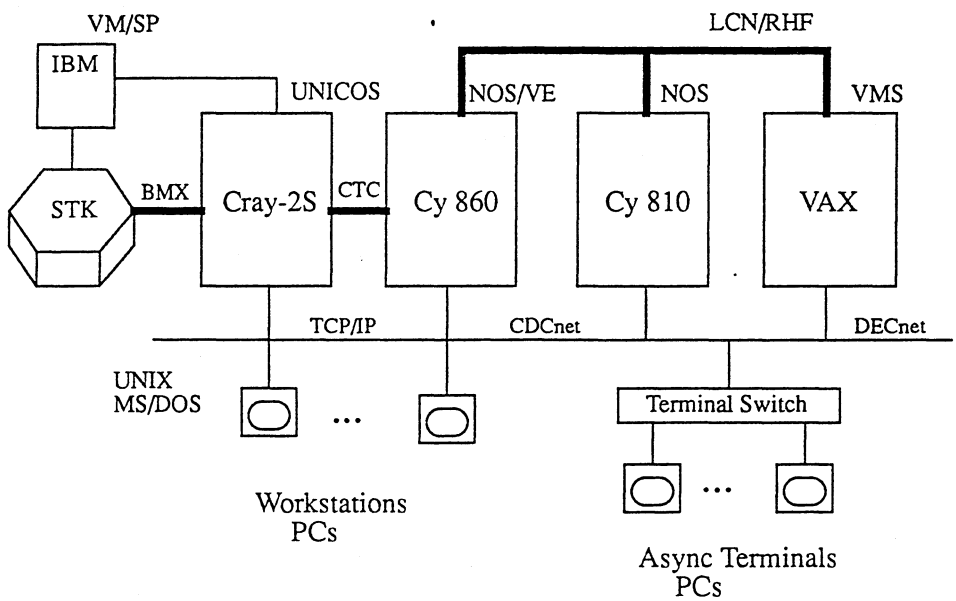


Fig. 5: Structure of DKRZ computer center in 1988

All the main selection criteria mentioned had obviously been met with the selection of the new production system, yet the resulting overall structure was far from satisfactory both for the users and the computer center staff: the existing "zoo" of operating systems and network protocols made it increasingly difficult to adequately use and control the various systems. The solution was rather straightforward and natural though: with the installation of the first major component with an UNIX-derived operating system the strategic decision was made to unify the set of operating systems on the basis of UNIX and to exchange the set of proprietary network protocols for the industry-standard ARPA-protocols, at least for some time.

At the same time the gradual exchange of asynchronous (dumb) terminals by various sorts of PCs has been mainly substituted by a (rather uncontrollable) advent of numerous moderately priced but powerful graphical workstations which also were running the UNIX operating system thus reenforcing the decision to base all central systems on UNIX. The problem of how to structure the central systems consistent with both the still existing set of terminals and PCs and the newly arriving UNIX-workstations quite naturally led to the well known client server structure which ideally is depicted in Fig. 6.

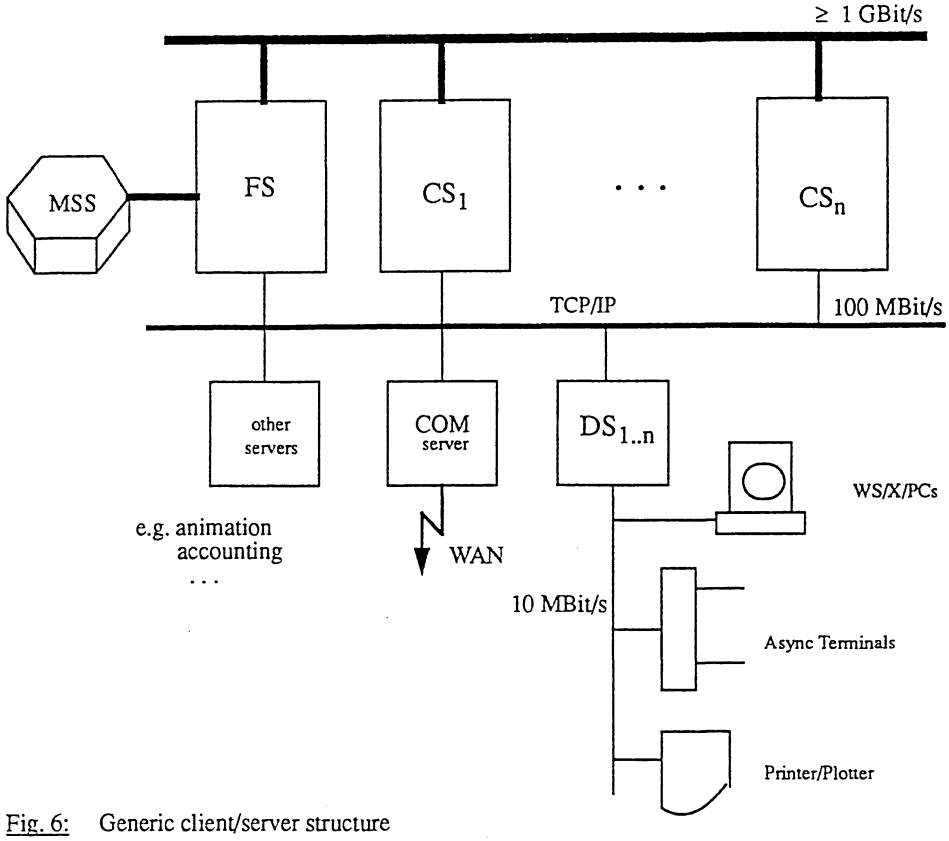


Fig. 6: Generic client/server structure

The first central resource type is a set of compute servers (CS_i) dedicated to the primary task of large-scale numerical processing which - at least in the case of climate modelling applications - leads to a large amount of model data generated. So the second necessary central ressource will be a central file serving facility (FS) with adequate software and some mass storage hardware (MSS) to control and store the data. After appropriate post-processing scientific results typically have to be displayed in some graphical form since the sheer amount of data makes most other types of examination unreasonable. This may happen decentrally for transient preview or hardcopy of single images, wheras the natural display form of time-dependent data - series of images or a movie - or other special graphical oprations may still be made with central resources. So other servers come into play, mainly file and graphical servers with still more types possible. Interconnection between these servers and the clients has to be supported by sufficiently fast transport networks.

The client side is increasingly consisting of graphical workstations, but X-Terminals, PCs, and for some time even asynchronous terminals clustered by terminal servers may stay in use. Local peripherals have come into widespread use. All these devices are naturally clustered on some appropriate departmental level with a dedicated department server (DS_i) for local support.

The function of the departmental server for the terminals, PCs, X-terminals and peripherals is rather obvious: it serves as a local compute server quite similar to the traditional timesharing systems in the backend/frontend structure. An additional advantage for the end user may be the increased redundancy.
A possible disadvantage might be the additional complexity in the user area with many different workstations of possibly different vendors. A rather "easy" administration and support of user workstation clusters would be possible if the status of the workstations is 'dataless' as opposed to 'standalone' which is the recommended workstation mode at DKRZ. All the workstation users home directories reside on the server disks to keep the data in regular backup cycles, but each workstation is recommended to have a small local disk for the root and swap partitions. In this case the departmental server functions effectively as file and software server for the connected workstations, though obviously this workstation mode can hardly be enforced.

In the course of the year 1990 a rather large numerical climate experiment has been conducted at DKRZ at the request of the federal government. To assess possible future climate developments caused by the greenhouse effect the IPCC (Intergovernmental Panel on Climate Change) had proposed to study the effect of 4 different carbon dioxide immission scenarios. The proposed simulation time covered 100 years, starting from 1985 and extending the prediction into the year 2085.
The coupled model used in these experiments was in its atmospheric component based on the medium-range weather forecast model T21 which had been modified in the radiation, the hydrological cycle and other areas to better suit the requirements of climate problems. This model had been developed originally at ECMWF (European Center for Medium Range Weather Forecast) in Reading (England). It is a rather low resolution spectral model with triangular truncation at wavenumber 21, which corresponds to an explicit 64*32 grid in 19 levels. This again corresponds to a 5,625° or 625 km longitudinal grid point distance.
Only 2 of the proposed IPCC scenarios have been finally conducted, each with 2 different global ocean models both being developed at the Max-Planck-Institute for Meteorology in Hamburg. Together with control experiments the total number of experiments resulted in 8 which needed 12000 CPU hours on one Cray-2 Processor or roughly 1.5 Cray-2 processor years. The amount of corresponding model data generated was 450 GByte in compressed format. Obviously both model resolution and integration time had been rather well adapted to the available processing power and mass storage capacity if one is willing to assign nearly half the production capacity to one single series of experiment.

There are a number of immediate conclusions which can be drawn:

- the current global model resolution with its corresponding explicit modelling of physical processes reaches the limits of today's supercomputers in the case of climate timescales, as e.g. with a Cray-2 (2 GFlops peak),

- based on the current model resolution a Cray-2 or equivalent system will produce roughly 1 TByte of compressed model data per year. This can just about be managed with the mass storage technology of today which is generally based on autoloaders equipped with 3480-compatible tape transports,

- the immediate critical factor should be the mass storage capacity as opposed to the transfer rate.

One of the immediate consequences at DKRZ had been the decision to incrementally increase the installed compute power and mass storage capacity. With a Cray Y-MPE4/364 another compute server has been installed in the first half of 1991 which is comparable in system thruput to the Cray-2S as benchmarks with climate applications have shown. This increase in compute power has been accompanied by the upgrade of the mass storage capacity by another STK library storage module with control units and transports.
A new fileserver system is about to be installed shortly: a Convex C3860 which is expected to dedicate about 4 of its 6 processors to compute serving tasks thus incrementally increasing the installed computer power again by a factor of approximately 0.5 Cray-2 equivalents according to benchmarks. The resulting hardware configuration and structure as expected for the end of 1991 is shown in Fig. 7.

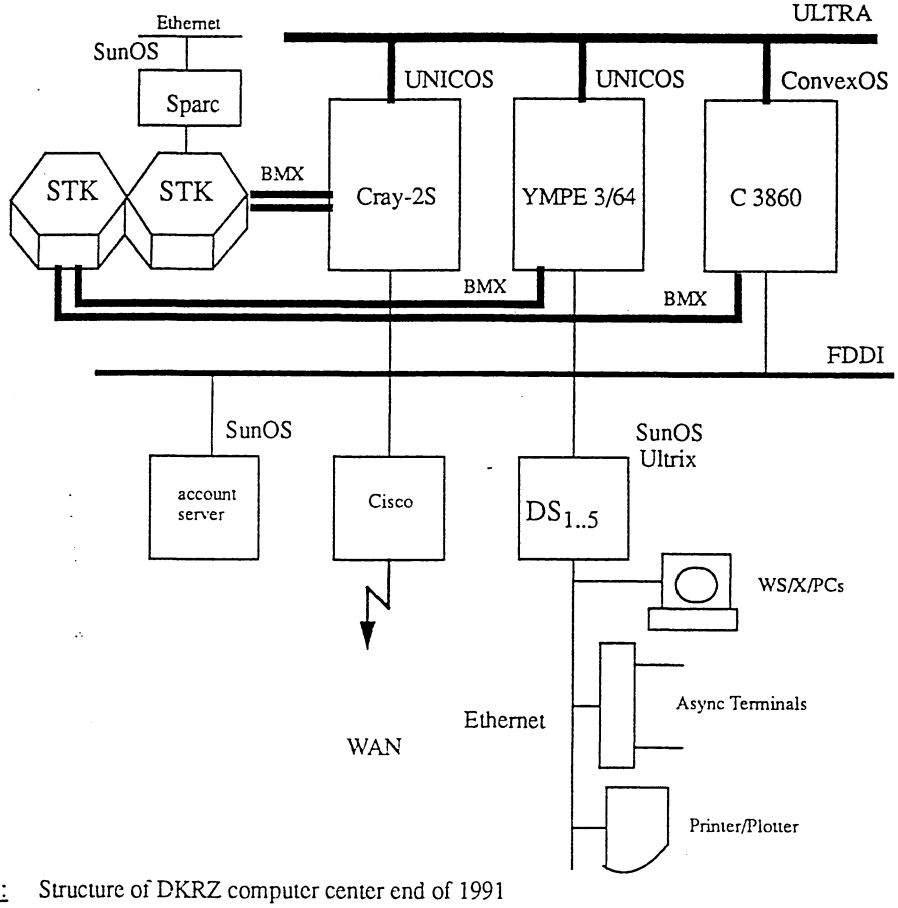


Fig. 7: Structure of DKRZ computer center end of 1991

The migration from the set of proprietary operating systems and network protocols in use in 1988 to an all UNIX-based environment will be completed in 1991. This conversion from an environment which has been in use for many years is by no means an easy process both for the users and the centre staff involved. It requires rather careful planning taking many steps as small as possible, and it implies a lot of burdens in terms of converting programs, data, knowledge and the way of thinking.

Obviously one of the less nice features of the current configuration is the way the nearline mass storage capacity is shared between the 3 central compute servers. The media in the 2 robot silos may be shared between all 3 systems since they all have access to the Sparc station running the STK control software via the Ethernet control path. The data paths and transports though are statically assigned as indicated in Fig. 7 which is not an optimal resource usage and may lead to unnecessary passthru operations between the 2 silos.

Certainly the biggest disadvantage is the logical view of the media in the silos which are assigned to users and groups of users and have to be managed by these users in order to keep track of the data on the tapes. To relieve the users from this organizational burden the fileserver software product UniTree has been installed in the middle of 1991 on a Convex C220 which is used by DKRZ until the C3860 will finally arrive in fall 1991. An evaluation period for UniTree is scheduled until end of 1991 to assess both the functionality and reliability of this product.. If transferred into production the mass storage system will be converted into a private I/O device of the fileserver (both control and data paths) as depicted in Fig. 8.

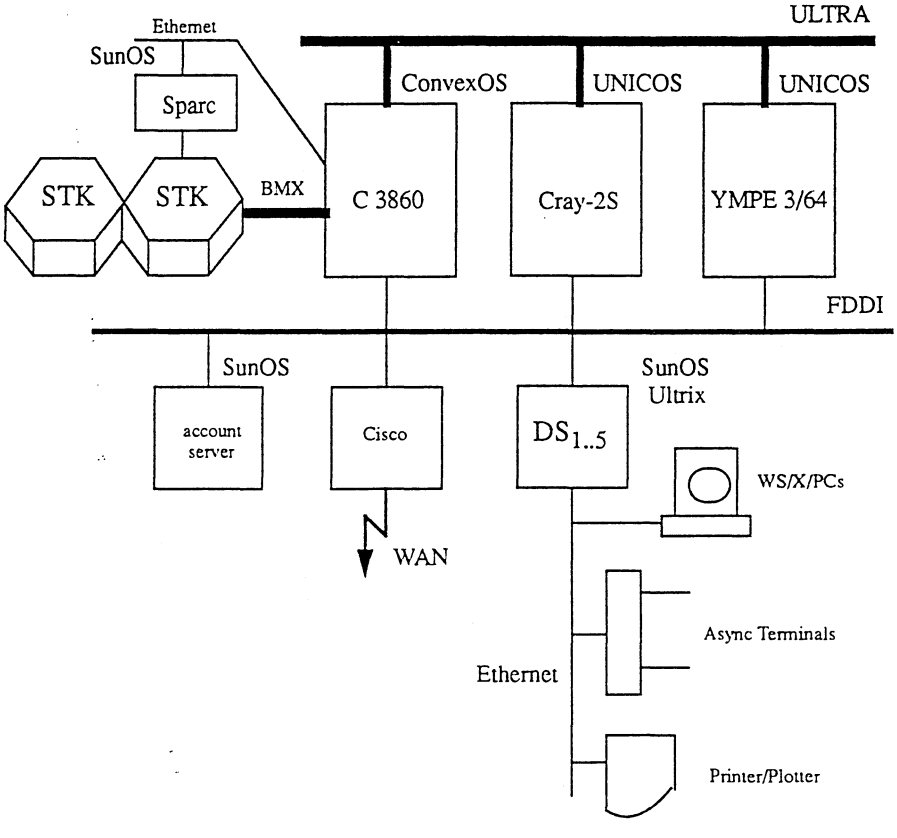


Fig. 8: Structure of DKRZ computer center beginning of 1992

Future Requirements

The next logical and probable step would be to decrease the grid point distance by a factor of 2 which in the atmospheric component would lead to T42 resolution. Together with the additional arithmetic complexity depending on the explicit simulation of physical processes which had been parameterized in the previous resolution models a system with O(10) times the computational power or about 20 GFlops peak would be needed to run these simulations in reasonable time. In order to keep the generated model data in nearline access a mass storage system with a capacity of 10-20 TByte would probably be adequate. Fortunately both the supercomputer systems in the GF20 range and the mass storage systems with tens of TBytes based on helical scan technology are about to be available soon.

As important as the hardware performance is an adequate software environment, again and particular in the case of mass storage systems. It is not reasonable to give users access to thousands of tapes and let them try to organize and keep track of the data on these tapes. It is necessary to have fileserver software which transparently will manage the residency of the data files in several storage hierarchies which to the users look like one UNIX filesystem. The client local disk space should be managed transparently as defined by site-specific configuration threshholds.

Conclusions

Even though the current climate model resolution is rather low and inadequate for the explicit modelling of important lower scale physical processes, the amount of necessary compute power and mass storage capacity requires today's most advanced systems if global coupled models are to be integrated over the full climate time scales. It should be noted that probably not even the T42 resolution with its grid point distance of ~ 2.8° or ~30 km is adequate for detailed regional predictions.

Higher resolution together with the explicit modelling of more physical processes are not the only trend requiring more advanced computing and storage systems. The desire to model a more complete set of physical subsystems as e.g. atmospheric chemistry will put an even higher demand on computational power and storage capabilities necessary for the most advanced environmental applications. So it can be expected that for still some more technology generations the environmental centers will be waiting to upgrade to the most powerful products of the super-computer industry.

1991 USERS CONVEX CONFERENCE

FINITE ELEMENT OF RUBBER PARTS IN HUTCHINSON ON A HIGH PROCESSOR CONVEX C 240

D. BENOUALID - B. RAVIER - I. WANDER - J.Y. LACHARTRE

Thermomechanical responses of rubber components are analysed in HUTCHINSON by means of finite element method. An implicit in house code is used to perform static, quasi-static and dynamic analyses. In this code, the most time consuming parts are matrix decomposition by Cholesky method material routines and wide usage of a scalar product function. Most of the heavy computations are executed on a CONVEX C 240 computer, and some of them on scalar workstations and servers. So, to utilize in the best way the CONVEX C 240 environment, modifications were performed in the code, taking into account the different possibilities of speed up. For example, reprogramming of some routines and optimization for vectorization was done, specific routines belonging to the Convex mathematical library were used, and compilation facilities as inlining option were considered. The modifications, even simple, show an improvement of the performance. In this article, we will discuss the modifications and present the results (execute times) oriented to two main objectives ; speed up of one big computation, and execution of many computations at the same time (throughput). Comparison with scalar processor will also be discussed.

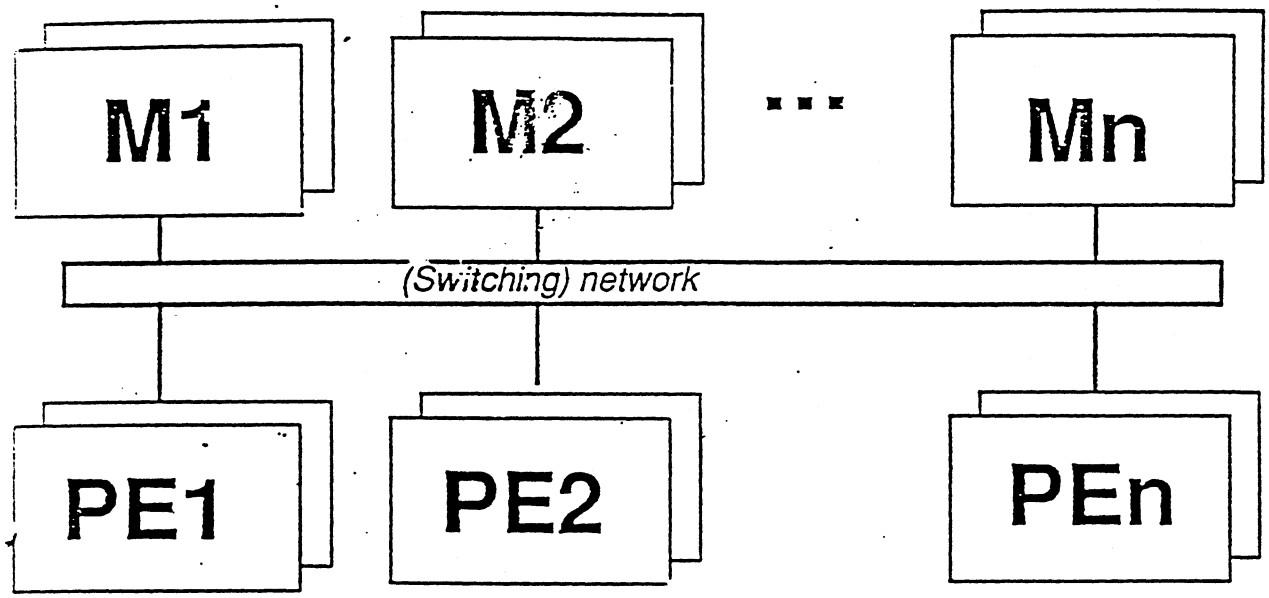
PARALLEL ALGORITHMS

Wolfgang Gentzsch
GENIAS Software GmbH

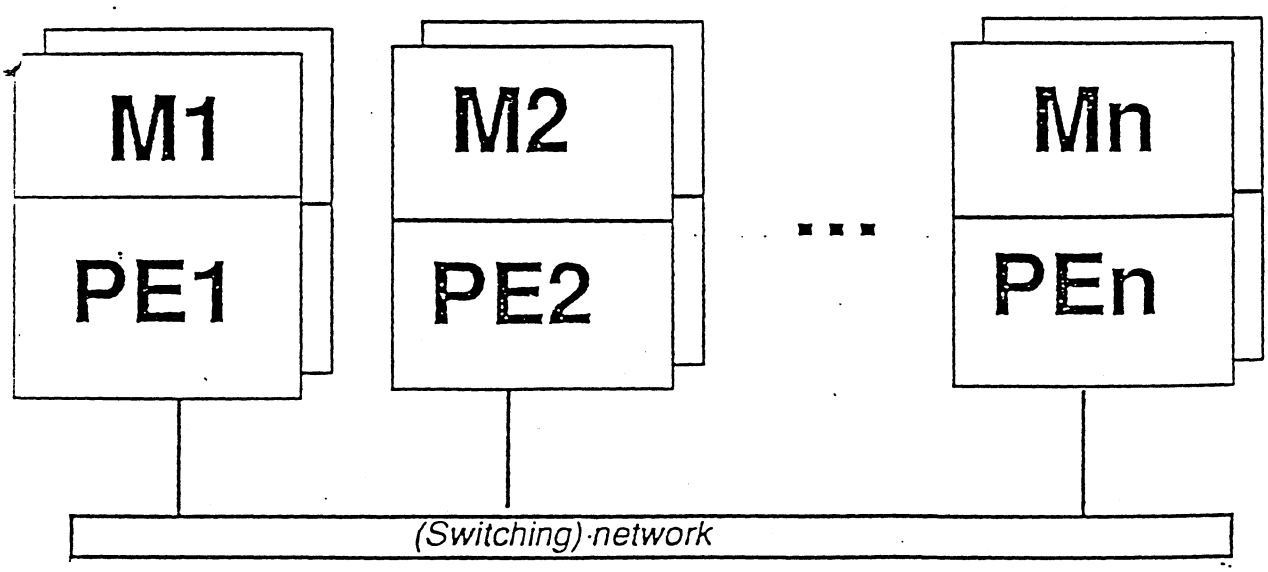
European Convex User Conference
Hamburg, 9 - 11 October 1991

Experience with shared and distributed memory systems
Data parallel vs. function parallel
Domain decomposition vs. algorithm parallelization
Look into the future

Figure. Basic data mechanisms.



Shared memory



Message passing

SPEED-UP FOR PARALLEL FRACTIONS BETWEEN 90% and 100%

$$Sp = \frac{T1}{Ts + Tp} = \frac{1}{1-f + f/p}$$

f [%]	p=1	p=4	p=16	p=64	p=infinity
100	1.00	4.00	16.00	64.00	infinity
99	1.00	3.88	13.91	39.26	100.00
98	1.00	3.77	12.31	28.32	50.00
97	1.00	3.67	11.03	22.14	33.33
96	1.00	3.57	10.00	18.18	25.00
95	1.00	3.48	9.14	15.42	20.00
94	1.00	3.39	8.42	13.39	16.67
93	1.00	3.31	7.80	11.83	14.28
92	1.00	3.23	7.27	10.60	12.50
91	1.00	3.15	6.81	9.59	11.11
90	1.00	3.08	6.40	8.77	10.00

The decrease of speed-up with decreasing parallel fraction f is evident. The conclusion in practice is to use parallel computers with many processors only for highly suitable (parallel) code.

The Magnetohydrodynamics Code

$$\frac{\partial \underline{v}}{\partial t} + (\underline{v} \cdot \text{grad})\underline{v} = -\text{grad}\, p + \underline{j} \times \underline{B} \ ,$$

$$\frac{\partial \underline{B}}{\partial t} = \text{rot}\,(\underline{v} \times \underline{B}) \ ,$$

$$\frac{\partial p}{\partial t} = -\underline{v} \cdot \text{grad}\, p - p \cdot \text{div}\, \underline{v} \ ,$$

$$\text{div}\, \underline{B} = 0$$

where j=rot B and B means magnetic field, v the velocity

$$\underline{j} = \begin{pmatrix} 0 \\ 0 \\ JZ \end{pmatrix} \ , \quad \underline{B} = \begin{pmatrix} BX \\ BY \\ 0 \end{pmatrix} \ , \quad \underline{v} = \begin{pmatrix} VX \\ VY \\ 0 \end{pmatrix}$$

results in

$$JZ = \frac{\partial}{\partial x} BY - \frac{\partial}{\partial y} BX \ ,$$

$$VX = -\frac{\partial p}{\partial x} - BY \cdot JZ \ ,$$

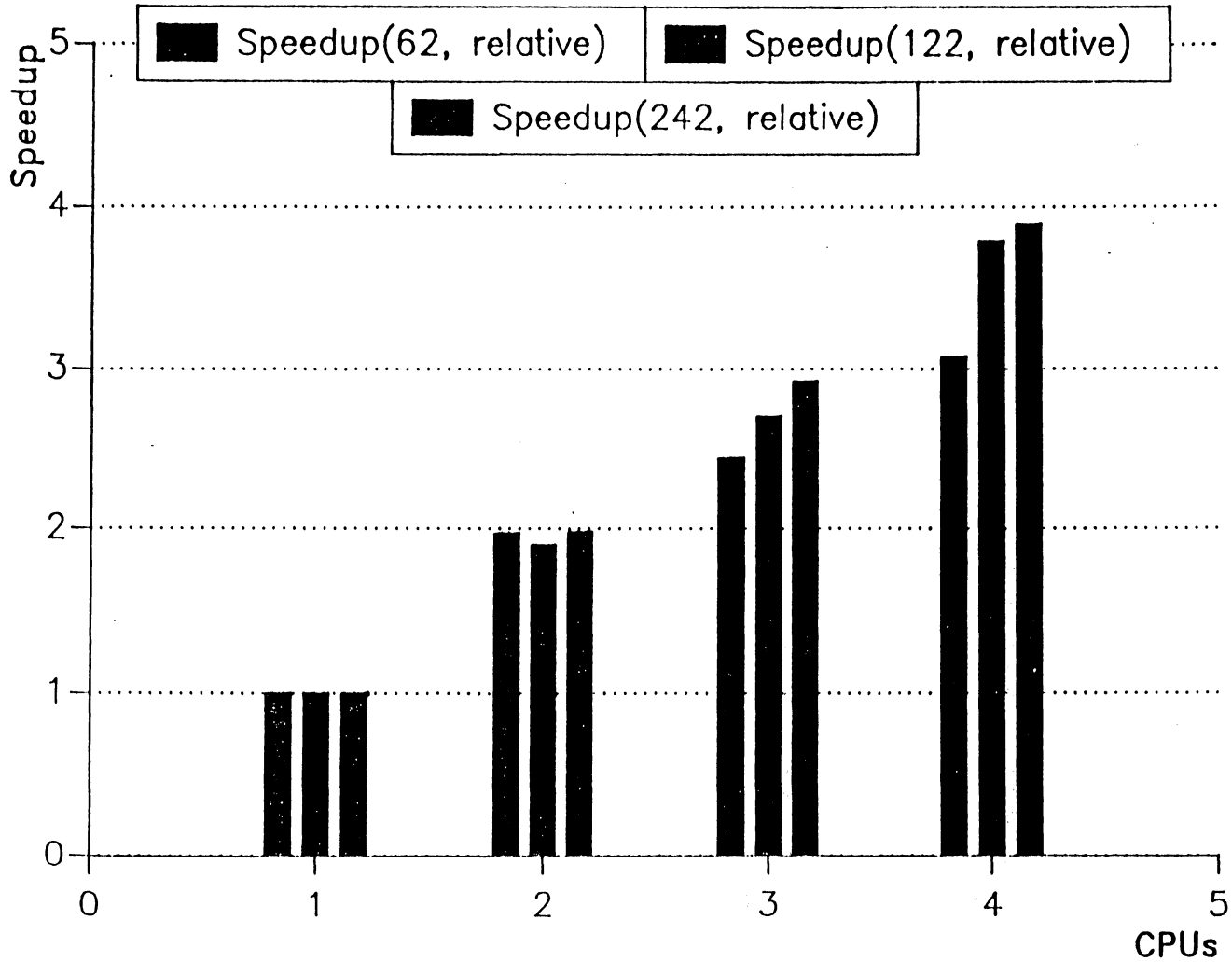
$$VY = -\frac{\partial p}{\partial y} + BX \cdot JZ \ ,$$

$$\frac{\partial}{\partial t} BX = \frac{\partial}{\partial y} (VX \cdot BY - VY \cdot BX) \ ,$$

$$\frac{\partial}{\partial t} BY = -\frac{\partial}{\partial x} (VX \cdot BY - VY \cdot BX) \ ,$$

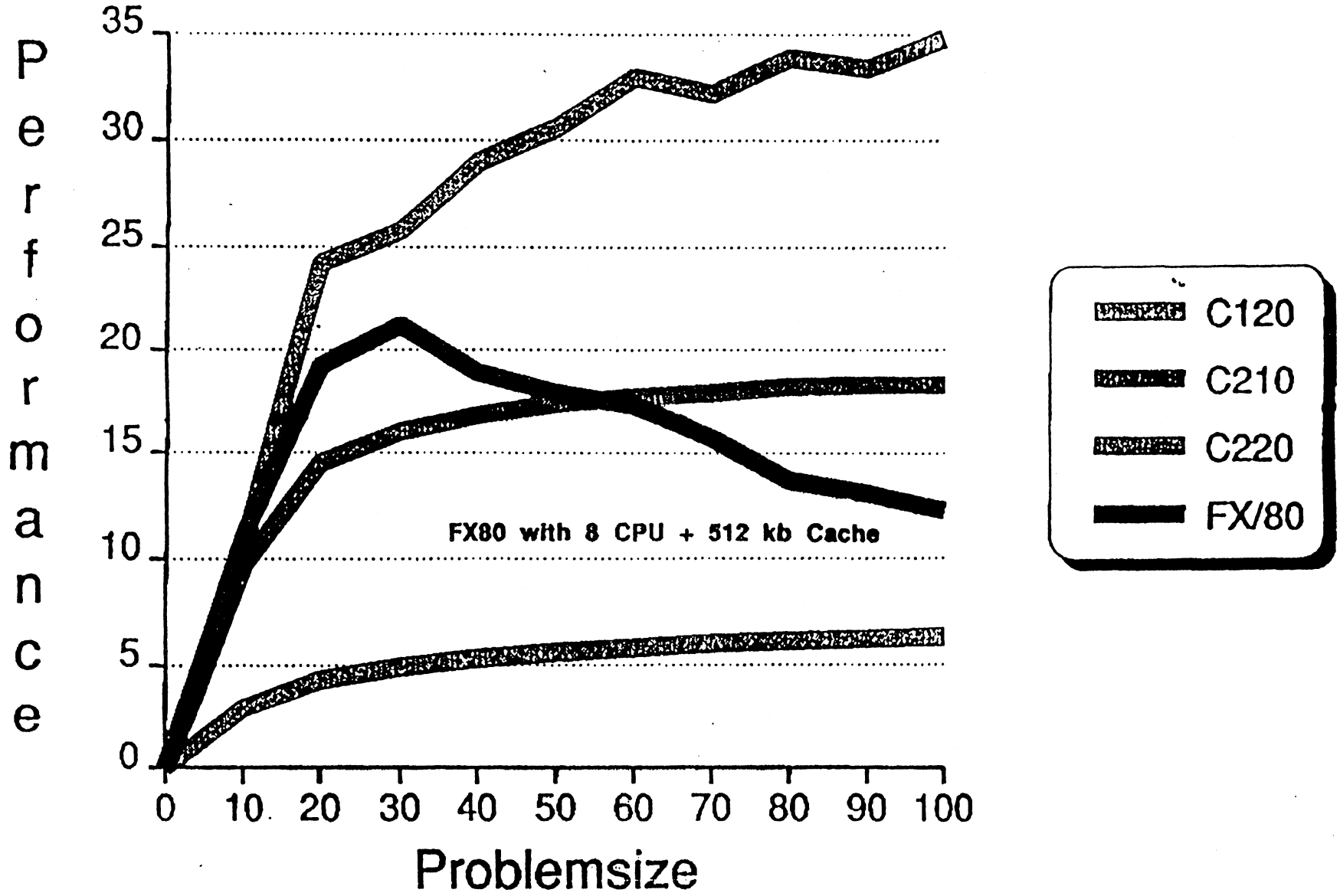
$$\frac{\partial}{\partial t} p = -(VX \frac{\partial p}{\partial x} + VY \cdot \frac{\partial p}{\partial y}) - p\,(\frac{\partial}{\partial x} VX + \frac{\partial}{\partial y} VY) \ .$$

MHD Results (Microtasking)

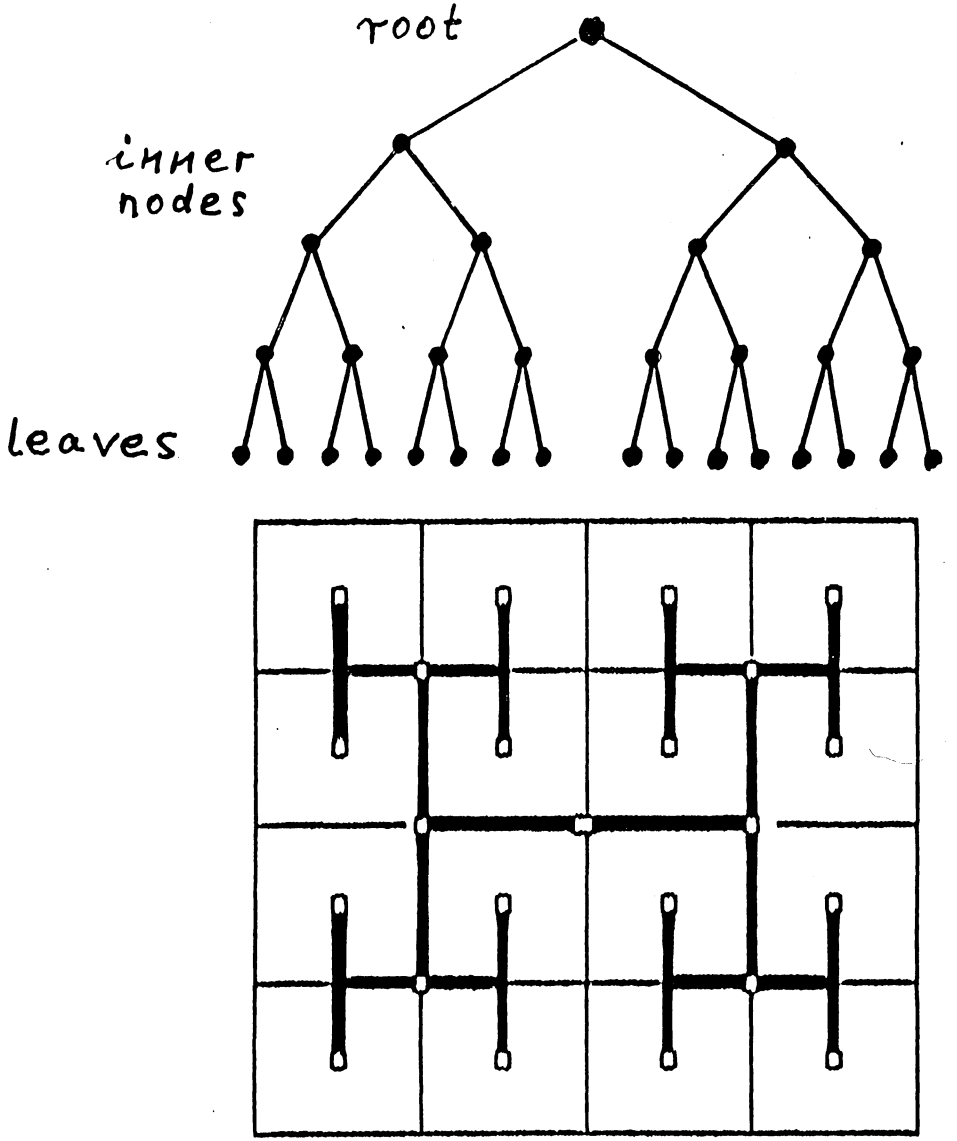


17 NOV 1987 10.12 BY ZDV062

Running a Magneto-Hydrodynamic Code for different Problem sizes (N**2 Gridpoints)



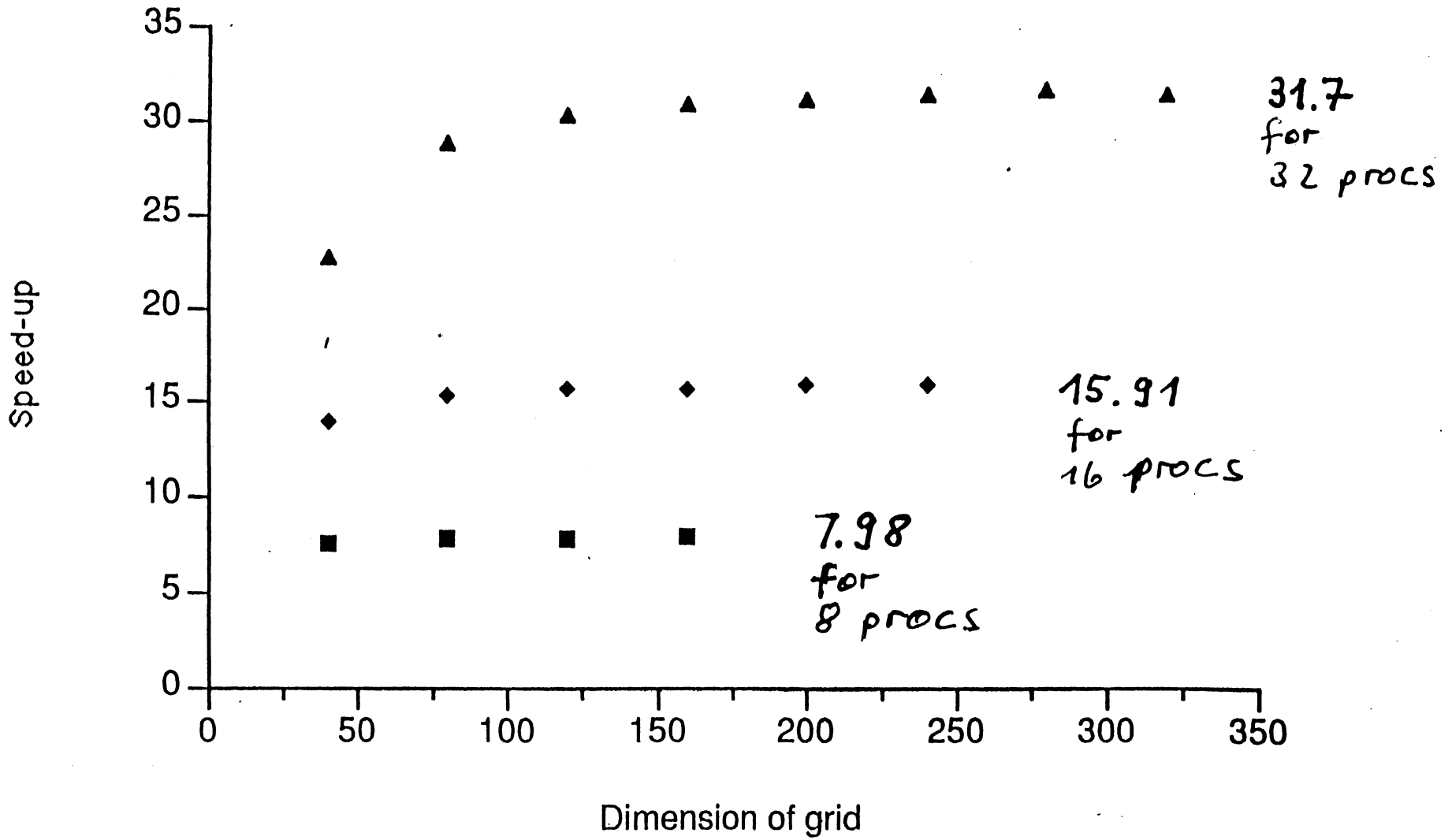
Example:

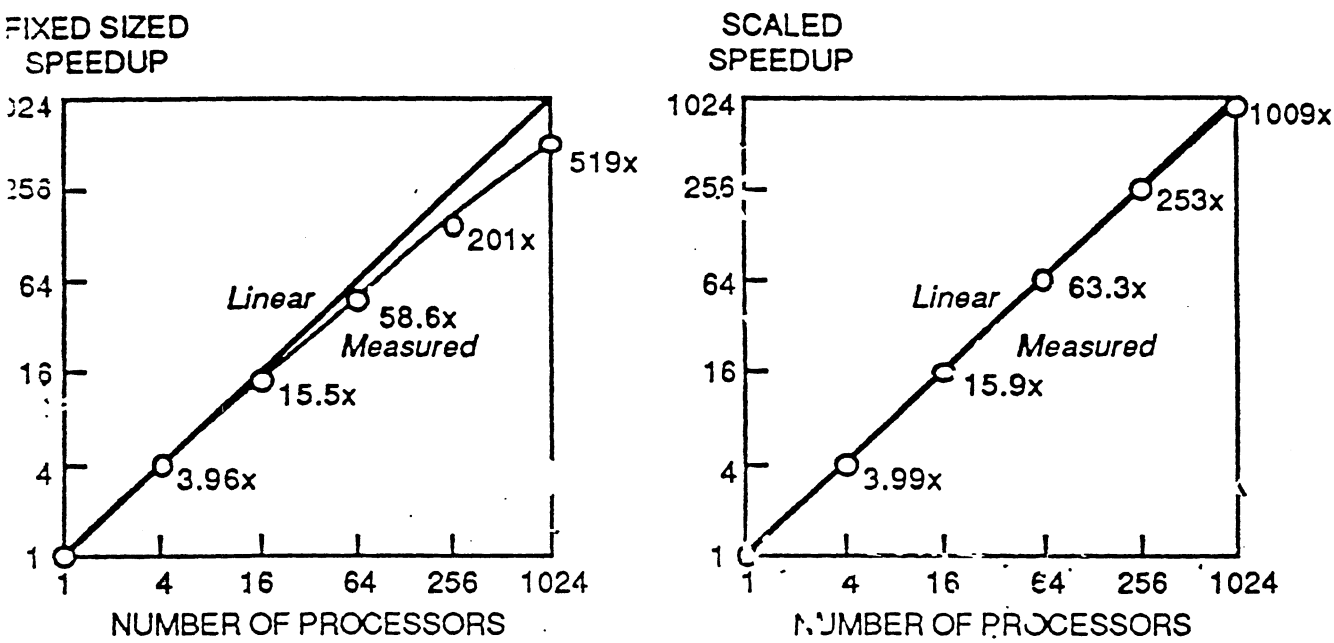


Binary Tree Computer (TX3)

TX3:

iP - Systems
Karlsruhe

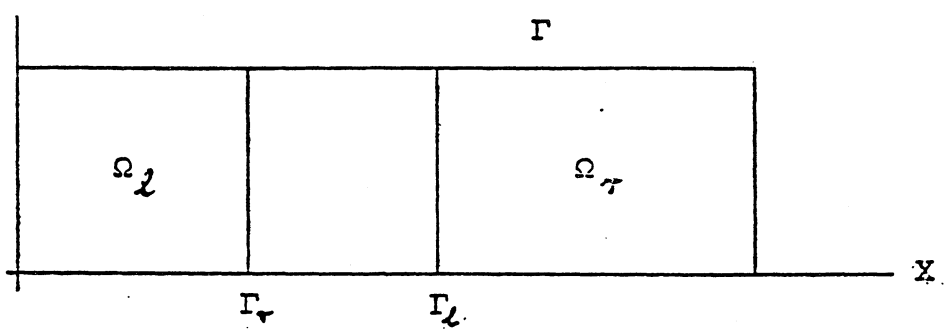




Fluid Dynamics Problem Speedup

D O M A I N D E C O M P O S I T I O N M E T H O D S

* Domain is subdivided into overlapping subdomains
* Involves only local grids
* Local grids can be distributed among processors



Original Problem:

$$L_\Omega u = f_\Omega \text{ in } \Omega$$

$$L_\Gamma u = f_\Gamma \text{ on } \Gamma$$

Decomposition into Local Problems:

$$\Omega_i \subset \Omega, \quad \Omega_j \subset \Omega \quad \wedge \quad \Omega_i \cap \Omega_j \neq 0$$

$$L_\Omega u_i = f_\Omega \text{ in } \Omega_i$$

$$L_\Gamma u_i = f_\Gamma \text{ on } \delta\Omega_i \setminus \Gamma_i, \quad u_i = u_j \text{ on } \Gamma_i$$

$$L_\Omega u_j = f_\Omega \text{ in } \Omega_j$$

$$L_\Gamma u_j = f_\Gamma \text{ on } \delta\Omega_j \setminus \Gamma_j, \quad u_j = u_i \text{ on } \Gamma_j$$

Data Partitioning

```
DO  ISIG=1,NSIG
  CALL CFFT(NSAMP,SIGNAL(1,ISIG))
ENDDO
```

Each call can be executed in parallel on different heads

```
DO K=1,100
  DO J=1,100
    DO I=1,100
      Z(I,J) = Z(I,J) + X(I,K)*Y(K,J)
    ENDDO
  ENDDO
ENDDO
```

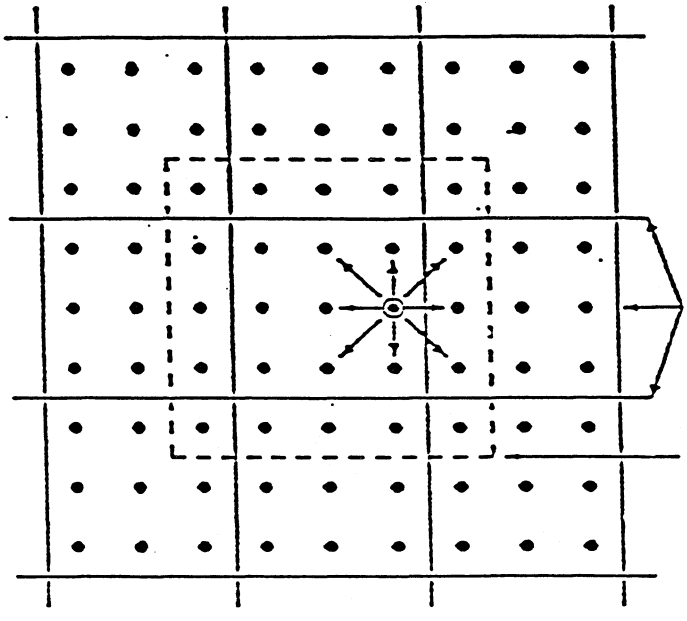
This loop can be executed in parallel

Vectorize this loop on one processor

Dividing the data among the processors

Parallelisierung durch Gitterteilung

Typischer Fall: Gitteroperatoren können simultan auf alle (oder einen Teil) der Gitterpunkte angewandt werden.



jeder Knotenprozeß arbeitet auf einem Teilgitter

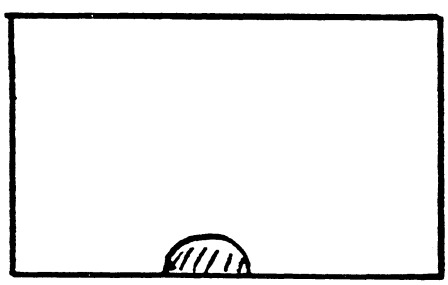
Prozeßgrenzen

Überlappungsbereich

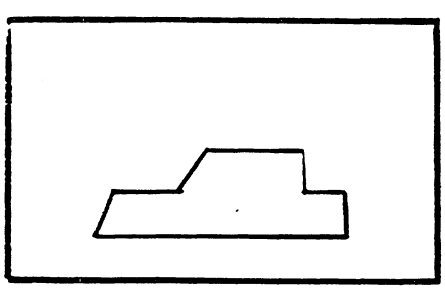
Kommunikation zwischen gleichzeitig ablaufenden Prozessen:
Austausch von Daten im Überlappungsbereich

PARALLELIZATION STRATEGY

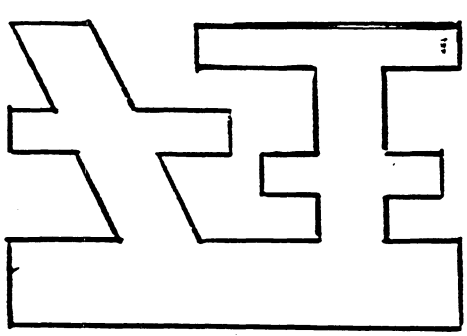
depends on the physical domain and on the mesh generation algorithm



easy

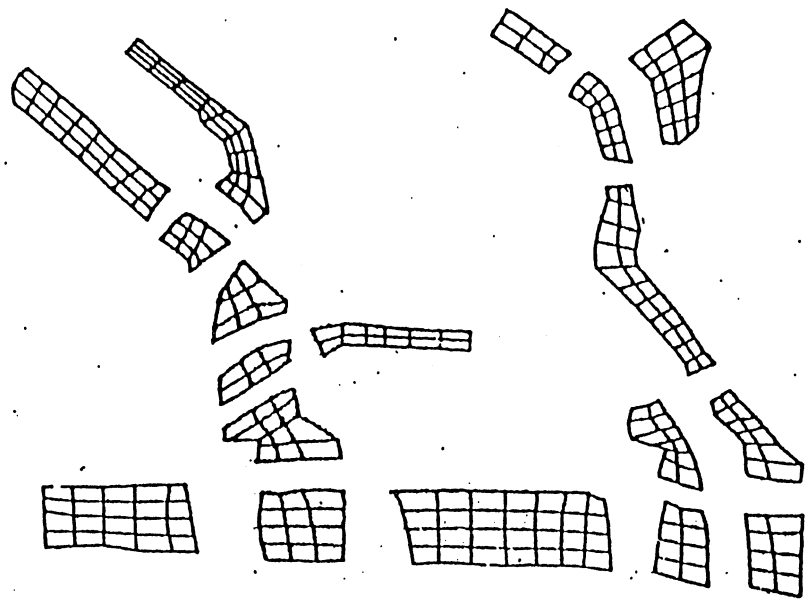


straightforward

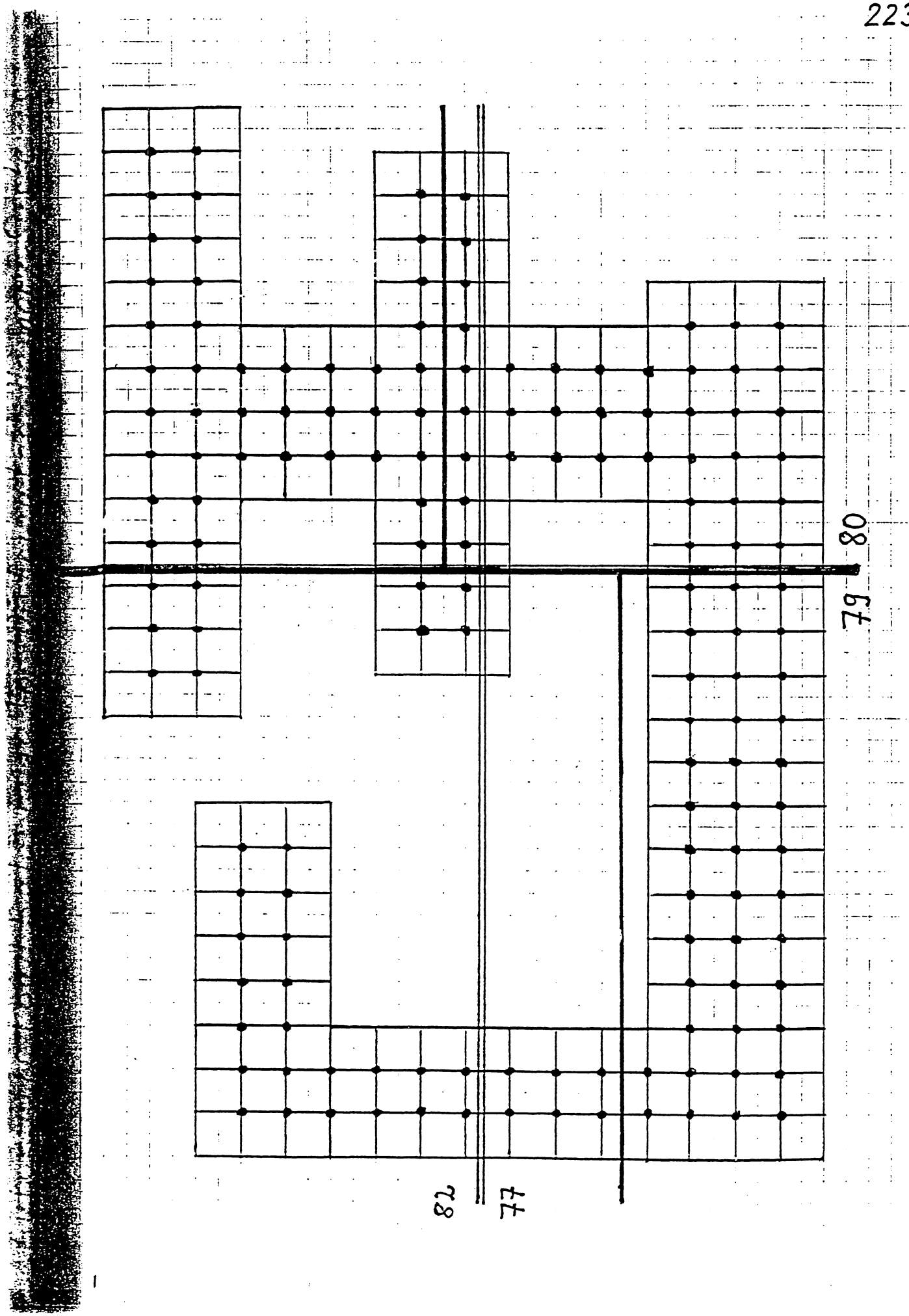


complicated

It might be useful to have different mesh generation procedures corresponding to the problem.



Exploded view of an 18 segment area which is part of the Hamburg harbour region. The shallow water wave equations for constant water depth have been solved on this solution domain


80
79
82
77

The Algorithm (Block Gaussian Elimination)

FACTORIZATION

```
do i = 1, NB
   do j = 1, i-1            (do ith row)
      do k = 1, j-1
         Aij = Aij - Aik * Akj
      Aij = Aij * Ajj       (already inverted)
   do j = 1, i-1            (do ith column)
      do k = 1, j-1
         Aji = Aji - Ajk * Aki
   do k = 1, i-1            (do diagonal)
      Aii = Aii - Aik * Aki
   Aii = inverse of Aii
```

FORWARD ELIMINATION

```
do i = 2, NB
   do j = 1, i-1
      Bi = Bi - Aij * Bj
```

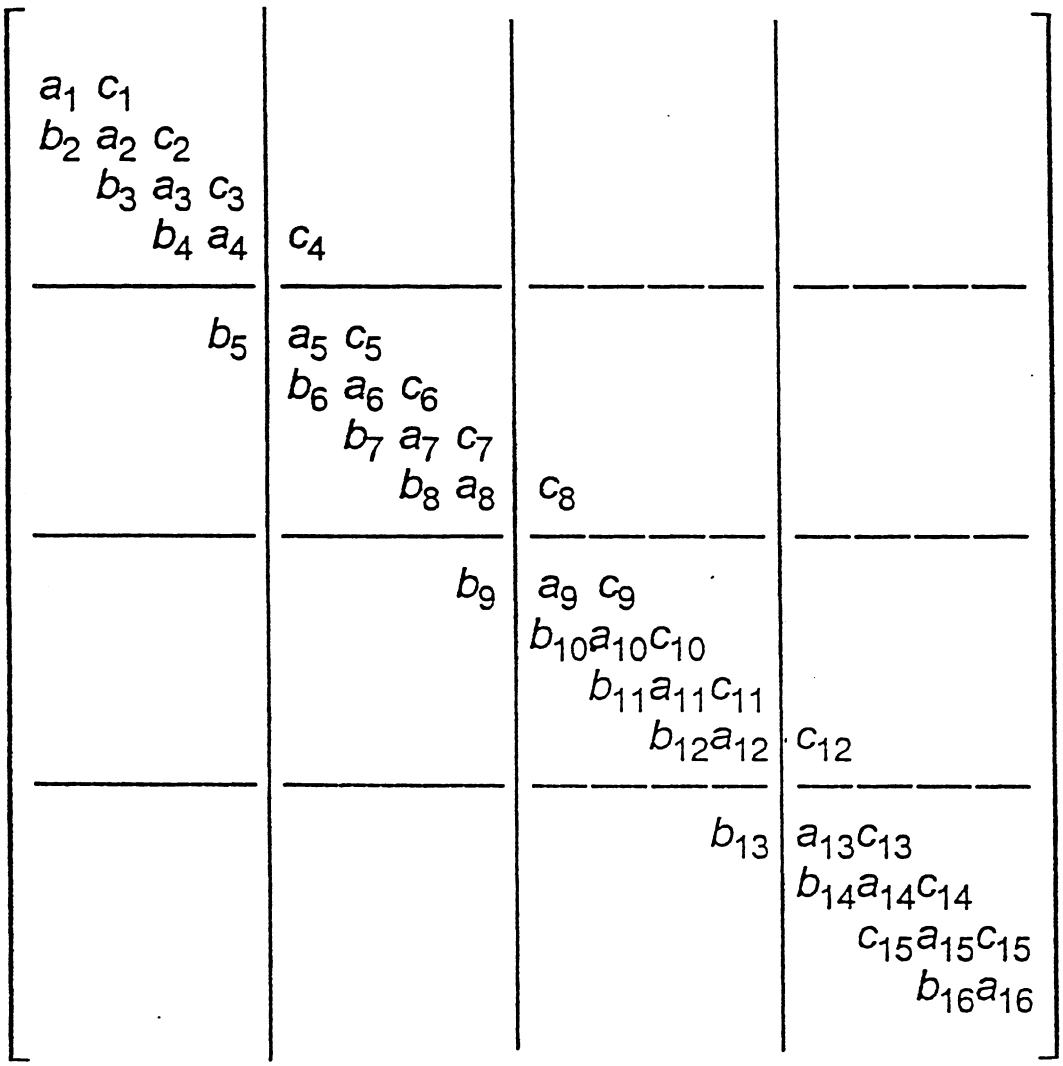
BACK SUBSTITUTION

```
do j = NB, 1, -1
   Bj = Ajj * Bj            (already inverted)
   do i = 1, j-1
      Bj = Bj - Aij * Bi
```

Only 3 Matrix Operations

A = A - BC	~ 85% of consumed cycles
A = BC	~ 12% of consumed cycles
A = A Inverse	~ 3% of consumed cycles

Parallele Algorithmen



In jedem Hauptdiagonalblock (parallel) wird die untere Nebendiagonale eliminiert.Dies führt zu "fill-

Parallele Algorithmen

ins" f's. Analog wird jeweils die obere Nebendiagonale eliminiert ("fill-ins" g's).

$$
\left[\begin{array}{cccc|cccc|cccc|cccc}
a_1 & & & g_1 & & & & & & & & & & & & \\
 & a_2 & & g_2 & & & & & & & & & & & & \\
 & & a_3 & c_3 & & & & & & & & & & & & \\
 & & & a_4 & & & & g_4 & & & & & & & & \\
\hline
 & & & b_5 & a_5 & & & g_5 & & & & & & & & \\
 & & & f_6 & & a_6 & & g_6 & & & & & & & & \\
 & & & f_7 & & & a_7 & c_7 & & & & & & & & \\
 & & & f_8 & & & & a_8 & & & & g_8 & & & & \\
\hline
 & & & & & & & b_9 & a_9 & & & g_9 & & & & \\
 & & & & & & & f_{10} & & a_{10} & & g_{10} & & & & \\
 & & & & & & & f_{11} & & & a_{11} & c_{11} & & & & \\
 & & & & & & & f_{12} & & & & a_{12} & & & & g_{12} \\
\hline
 & & & & & & & & & & & b_{13} & a_{13} & & & g_{13} \\
 & & & & & & & & & & & f_{14} & & a_{14} & & g_{14} \\
 & & & & & & & & & & & f_{15} & & & a_{15} & c_{15} \\
 & & & & & & & & & & & f_{16} & & & & a_{16}
\end{array}\right]
$$

Parallele Algorithmen

2 Möglichkeiten zum weiteren Vorgehen:

- Diagonalisieren der Matrix (Wang [10])
- Lösen eines $p \times p$-System (tridiagonal; skalar; seriell) (Meier [6]):

$$\begin{bmatrix} a_4 g_4 & & & \\ f_8 & a_8 & g_8 & \\ & f_{12} & a_{12} g_{12} & \\ & & f_{16} & a_{16} \end{bmatrix} \begin{bmatrix} x_4 \\ x_8 \\ x_{12} \\ x_{16} \end{bmatrix} = \begin{bmatrix} r_4 \\ r_8 \\ r_{12} \\ r_{16} \end{bmatrix}$$

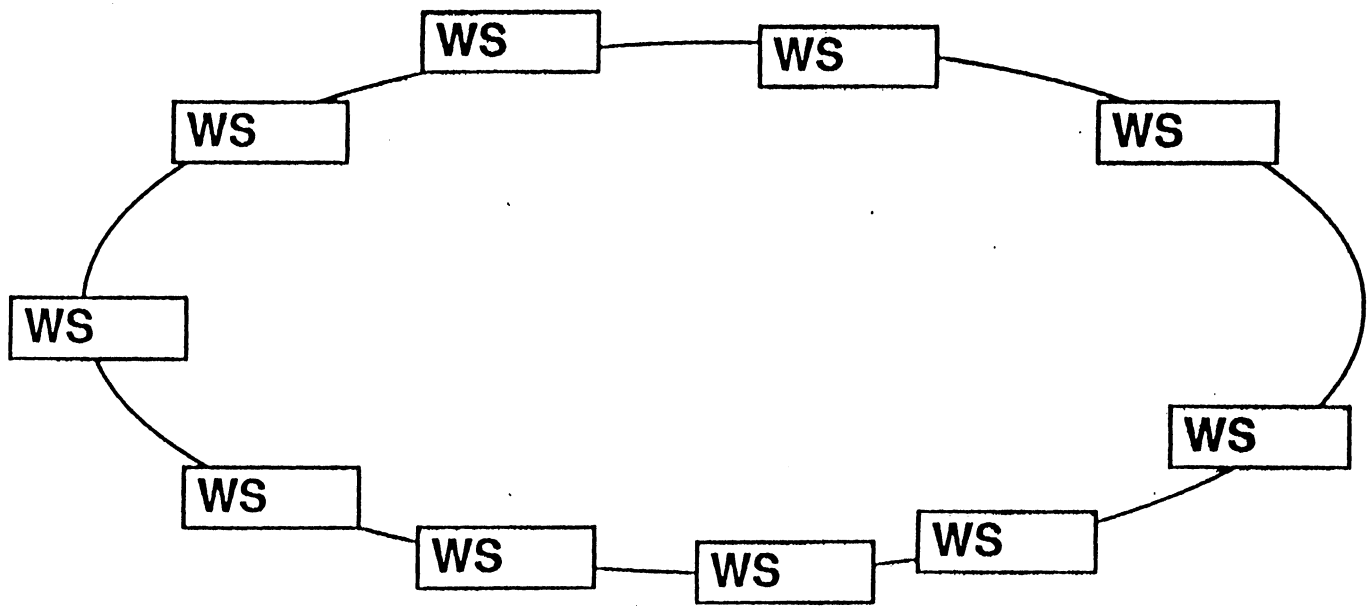
Berechnen der restlichen Lösungskomponenten (parallel).

Parallelität in FE-Codes

- element matrix generation
- matrix multiplication
- matrix decomposition
- forward / backward substitution
- eigenvalue extraction
- data recovery

- element matrix generation and data recov. highly parallel but serial implementation
- matrix multiplication and f/b subst. easy (if multiple load conditions)
- parallel decomposition is a challenge on hierarchical memory systems
- eigenvalue extr. difficult, evtl. Householder tridiagonalization

THE FUTURE?

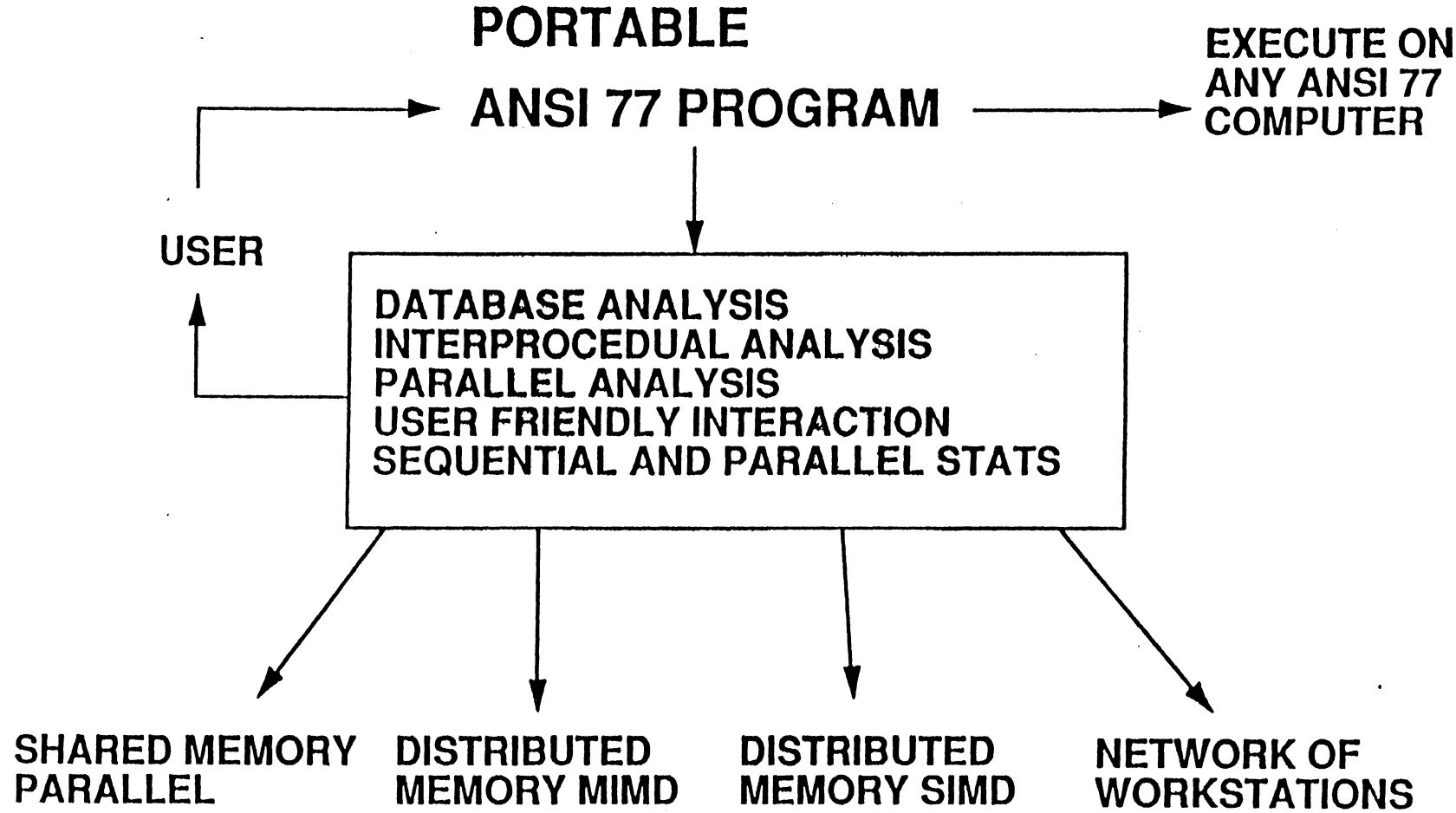


WORKSTATIONS WILL BE WITHIN A FACTOR OF 2-3 OF THE SCALAR PERFORMANCE OF A SUPER - NETWORKED WORKSATIONS COULD BE A VIABLE COMPUTING RESOURCE

What FORGE 90 can do for Parallel Architectures

- Provide a compatible parallel programming environment
- Provide high-level parallelism for each parallel implementation
- Provide intelligent interactivity for improving memory utilization
- Provide intelligent interactivity for partitioning an application across the parallel system
- Provide statistics gathering for sequential and parallel execution
- Provide static cost analysis for each parallel implementation

FORGE 90


PORTABLE
ANSI 77 PROGRAM
EXECUTE ON ANY ANSI 77 COMPUTER
USER
DATABASE ANALYSIS
INTERPROCEDUAL ANALYSIS
PARALLEL ANALYSIS
USER FRIENDLY INTERACTION
SEQUENTIAL AND PARALLEL STATS
SHARED MEMORY PARALLEL
DISTRIBUTED MEMORY MIMD
DISTRIBUTED MEMORY SIMD
NETWORK OF WORKSTATIONS

ParaSoft

The Goals of *Express*

- To provide a *portable* platform on which parallel programs and applications can be built, secure in the knowledge that they will outlast the hardware used during development.
- To provide high quality tools that facilitate the construction, analysis and optimization of parallel programs in a development environment that rivals those currently available on sequential systems.
- To provide both of the above services without sacrificing performance. We believe that the next generation of hardware will not be sufficiently advanced to handle poor programming paradigms and that we should work to make bext use of current available technology.

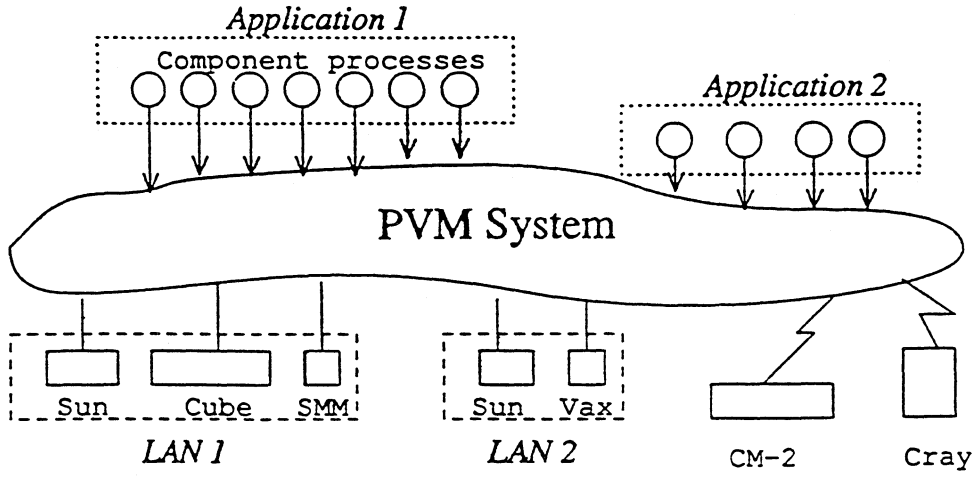
ParaSoft

Versions of *Express*

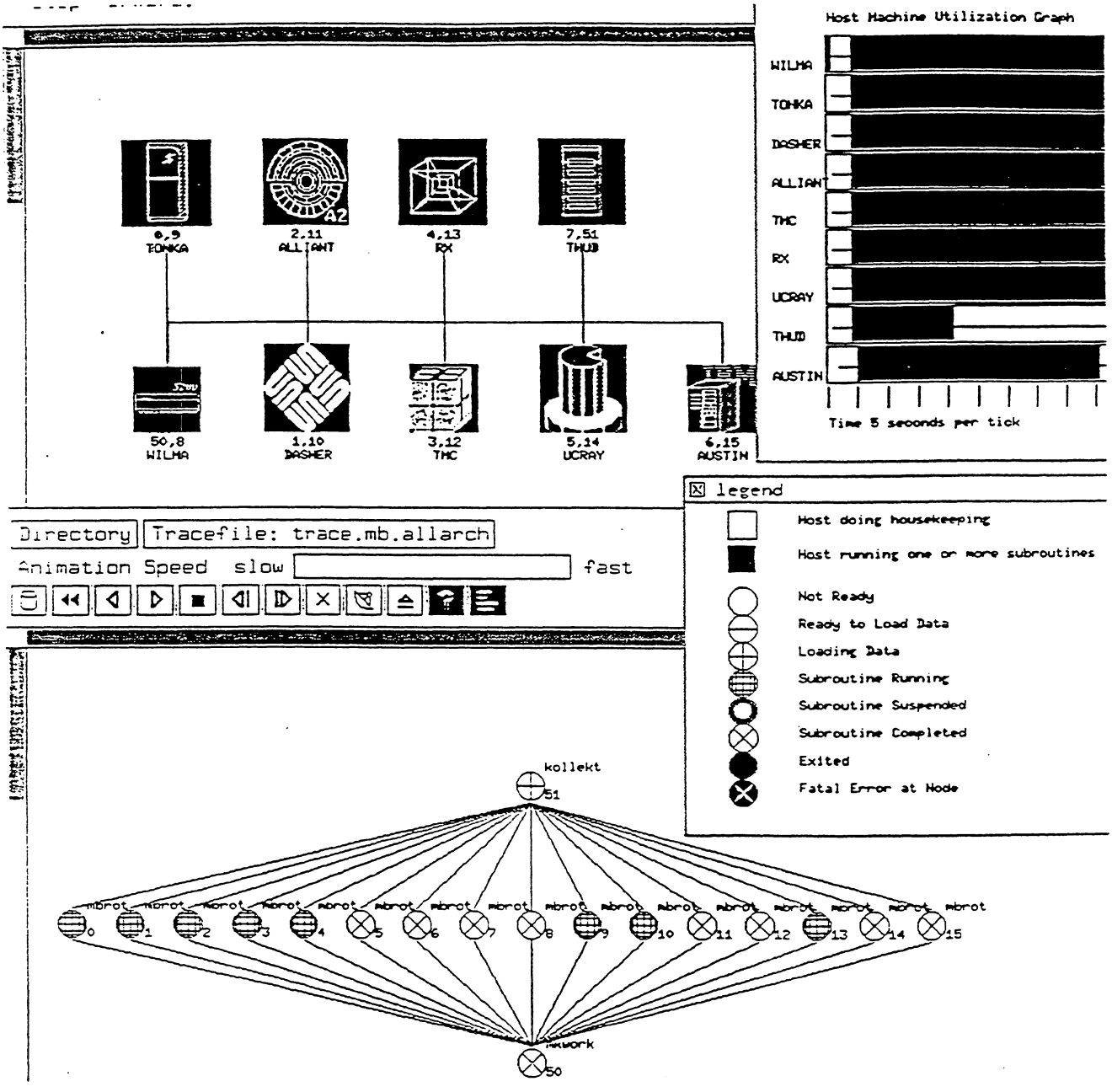
- **Currently Available:**
  - Intel iPSC/2, IPSC/i860, Delta prototype
    Sun networks
    IBM RS/6000 networks
    Silicon Graphics, parallel systems and networks
    Cray (Unicos)
    IBM 3090 (AIX)
    nCUBE/2
    Transputers (PC's, Sun's, Macintosh,...)
    Alacron i860
    PS/2 networks
    Planned for 1991
  - iWarp, Convex, Alliant, HP networks, DecStation Networks

Long range planning difficult because parallel processing systems become obsolete so rapidly.
Suggestions Welcome!

- Architectural model:



- Design Goals
  - Applications:
    + Straightforward, well understood paradigms.
    + Graphical development tools.
    + Debugging and profiling facilities.
  - System:
    + Support for multiple architectures (esp. multiprocessors) and networks.
    + Efficient distributed algorithms, multi-party protocols, failure resilience.
    + Straightforward installation, maintenance, operating and administrative interface.


Host Machine Utilization Graph
WILMA
TONKA
DASHER
ALLIANT
TMC
RX
UCRAY
THUD
AUSTIN
Time 5 seconds per tick
0,9
TONKA
2,11
ALLIANT
4,13
RX
7,51
THUD
50,8
WILMA
1,10
DASHER
3,12
TMC
5,14
UCRAY
6,15
AUSTIN
Directory
Tracefile: trace.mb.allarch
Animation Speed slow
fast
legend
Host doing housekeeping
Host running one or more subroutines
Not Ready
Ready to Load Data
Loading Data
Subroutine Running
Subroutine Suspended
Subroutine Completed
Exited
Fatal Error at Node
kollekt
51
mbrot
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
network
50

Programming-Models for Parallel Computers

Alfred Geiger

University of Stuttgart Computer-Center, Dept. Numerical Methods for Supercomputers, Allmandring 30, D-7000 Stuttgart 80, F.R.G., e-mail: geiger@rus.uni-stuttgart.de

Abstract

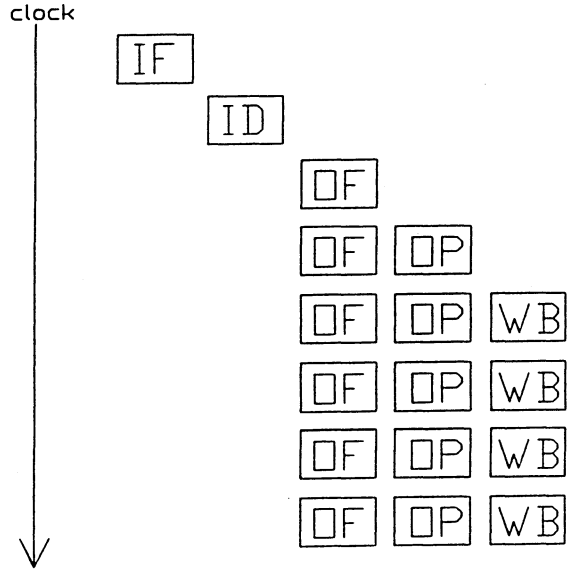
This paper gives an overview about the programming-models as seen by a programmer that are in use on today's parallel supercomputers:

- The vector-model
- The SIMD-model
- The superscalar-model
- The shared-memory MIMD-model
- The virtual-shared-memory-model
- The message-passing-model

Each of these models is analysed and explained, using as examples the same two code-segments.

1 The vector-model (CONVEX)

The vector-model is a data-parallel model based on the assumption, that the same operation should be applied to lots of data of the same type (vector). Processing is done in an assembly-line manner. Therefore one instruction processes a whole vector.



Example: matrix-multiplication (CONVEX)

```
real a(1:n,1:n), b(1:n,1:n), c(1:n,1:n)

do i = 1,n,1
    do j = 1,n,1
C$DIR   FORCE_VECTOR
        do k = 1,n,1
            c(i,j) = c(i,j) + a(i,k)*b(k,j)
        end do
    end do
end do
```

Example: Loop with condition (CONVEX)

```
do i = 1,n,1
    x(i)  = ......
    if (x(i) .gt. 0.0) then
        y(i) = sin(x(i))
    else
        y(i) = cos(x(i))
    end if
end do
```

This loop is processed according to one of the following algorithms:

- Version 1:
  - Compute x(i) for i = 1..n
  - Compute the then-clause for i =1..n and store on dummy-vector y1
  - Compute the else-clause for i =1..n and store on dummy-vector y2
  - Conditional vector-merge gives vector y

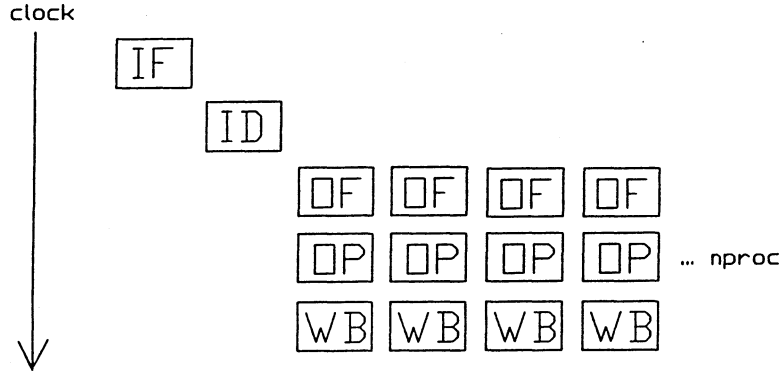
  The overhead is 50%, what limits the maximum efficiency also to 50%.

- Version 2:
  - Compute x(i) for i = 1..n
  - Gather all elements of x that are > 0 into a new vector.
  - Compute y(i).
  - Scatter
  - Gather all elements of x that are ≤ 0 into a new vector.
  - Compute y(i).
  - Scatter

  The overhead depends on the degree gather and scatter are supported by hardware.

2 The SIMD-model (MasPar, TM)

The same operation is executed on many data.
Synchronous parallel processing.
Processing of all data with one single instruction.



Example: matrix-multiplication (MasPar)
This example can only be processed in parallel by means of a library-call. FORTRAN-loops would be processed sequentially on the front-end.

```
real a(1:n,1:n), b(1:n,1:n), c(1:n,1:n)

c = matmul(a,b)
```

Example: Loop with condition (MasPar)

```
x = .......
where (x > 0.0) y = sin(x)
where (x <= 0.0) y = cos(x)
```

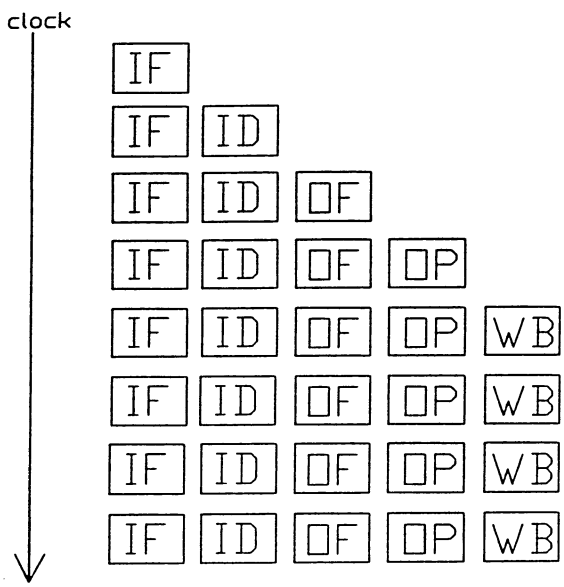
Processing is done according to the following sequence:

- Parallel computation of all elements of x.
- All processors with x $>$ 0.0 compute y, all the others are unused.
- All processors with x $\leq$ 0.0 compute y, all the others are unused.

The overhead in the alternation is 50% and therefore also the maximal efficiency that can be achieved is 50%. As all processors are processing the same instruction at the same time there are a lot of losses if the scope of a loop is much smaller than the number of processors, the same is true if it is slightly higher. Formally SIMD-machines can be looked at as vector-processors with very long vector-registers and a scalar performance tending to zero. These machines are scalable only if the number of processors is regarded, but not relative to the problem-size.

3 The super-scalar-model (i860, IBM-POWER)

Super-scalar processing is very similar to vector-processing as the underlying scheme is again the assembly-line-model. In contrary to the vector-model the operations applied to the data can be different for each set of operands. This is due to an additional instruction-stream that is also processed as a pipeline and therefore principally allows that, after a certain startup-period we can again get one result per clock-cycle.



Processing: One instruction per element.
Example: matrix-multiplication
The sequential program must not be changed, neither by the user nor by the compiler.

```
real a(1:n,1:n), b(1:n,1:n), c(1:n,1:n)

do i = 1,n,1
    do j = 1,n,1
        do k = 1,n,1
            c(i,j) = c(i,j) + a(i,k)*b(k,j)
        end do
    end do
end do
```

Example: Loop with condition

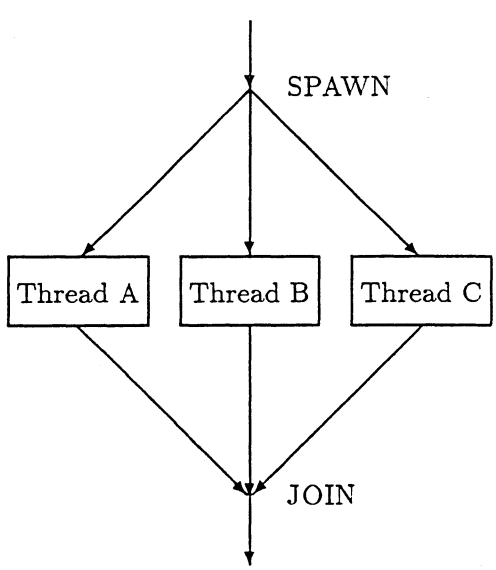
```
do i = 1,n,1
    x(i)  = ......
    if (x(i) .gt. 0.0) then
        y(i) = sin(x(i))
    else
        y(i) = cos(x(i))
    end if
end do
```

The processing-steps are absolutely the same as on a scalar machine, but with the speed-advantages of pipeline-processing. Under the condition that there are enough functional-units, an efficiency of 100% could be achieved. It must be mentioned that this assumption is not realistic with today's superscalar-processors but anyway –the theoretical potential is there and will be used in the future.

4 The MIMD-shared-memory-model (CONVEX)

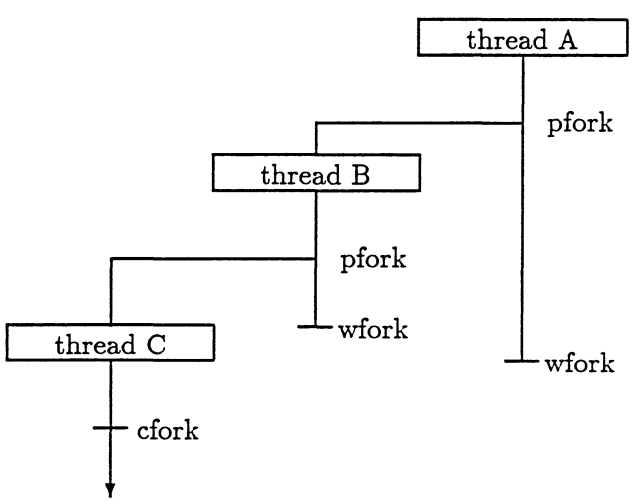
On the CONVEX-machines the parallelisation on statement-level is done by the compiler in a similar way as the vectorization. Thereby there is a principal distinction between two cases: symmetrical and asymmetrical parallelisation.

In symmetrical parallelisation multiple processors execute the same instruction-stream ('asynchronous SIMD'). The parallelism is explicit as e.g. in outer loops (inner loops are vectorized). The compiler generates a spawn-instruction as starting-point for parallel processing and a join-instruction as end-point.



At runtime the threads are processed by the CPUs that are actually idle. If there are fewer CPU's free than threads are possible, the threads are processed sequentially and so the code can be kept independant of the number of CPUs that are in the system. As soon as the last thread has executed its join, the process continues sequentially on the CPU that executed this thread. This dynamism leads to much higher flexibility as with SIMD, especially concerning multi-user environments.

The more general case is asymmetrical parallel processing. This is real MIMD, each CPU executes another instruction-stream what leads to a much more complicated topology of the process-structure as in the symmetrical case.
Everytime when the compiler detects a possibility for parallel execution, a pfork-instruction is generated. At the end of each thread at runtime a wfork is executed if there are still other threads running, a cfork otherwise. The process continues its execution sequentially on the processor that executed the cfork. If actually there are not enough processors, multiple threads are executed sequentially on one CPU.



Data-exchange between the threads is done via the communication-registers.

Example: matrix-multiplication (CONVEX)

```
real a(1:n,1:n), b(1:n,1:n), c(1:n,1:n)

do i = 1,n,1
C$DIR FORCE_PARALLEL
      do j = 1,n,1
C$DIR     FORCE_VECTOR
          do k = 1,n,1
              c(i,j) = c(i,j) + a(i,k)*b(k,j)
          end do
      end do
end do
```

Example: Loop with condition (CONVEX)

```
C$DIR FORCE_PARALLEL
do k = 1,nproc,1
C$DIR FORCE_VECTOR
      do j = 1,n/nproc,1
          i = (k-1)*n/nproc+j
          x(i)  = ......
          if (x(i) .gt. 0.0) then
              y(i) = sin(x(i))
          else
              y(i) = cos(x(i))
          end if
       end do
end do
```

To make use of parallel-processing in this example, it is necessary to introduce an outer loop running over the number of usable processors (!). The scope of the inner loop is therefore reduced. On machines with high pipeline-startup parallelisation therefore is on cost of vectorisation. A loss of efficiency due to the alternation can be caused by the vector-model in the inner. When using modern preprocessors the splitting-up of the loops is done automatically.

5 Virtual-Shared-Memory (BBN, KSR, Cray-MPP)

Virtual-shared-memory means that there is a physically distributed memory forming a logically common address-space, for the user a situation similar to the one described in the last section, but now in a scalable environment. To port codes from a shared-memory to a virtual-shared-memory-machine, the user doesn't have to change anything in his programs in a first step. The programs will run and parallelize in a similar way on task- as well as on statement-level. Depending on the bandwidth of the communication-network, efficiency will be more or less bad. The user can change this situation by segmenting his data according to the structure of the threads, so involving a certain locality. In contrary to the message-passing-model, which will be introduced in the next section, the user is not forced to do the whole communication explicitely and therefore be aware of all data-dependencies; this is work for the operating-system and the hardware.
The VSM-concept is, in the moment, the only visible way leading even people who want to port codes without rewriting them to scalable architectures. It is possible to optimize programs successively.

Example: matrix-multiplication (BBN)

```
real scatter, private a(1:n,1:n), b(1:n,1:n), c(1:n,1:n)

allocate(everywhere) a, b, c

parallel do, depth(2), chunk(equal), coalesce
do i = 1,n,1
    do j = 1,n,1
        do k = 1,n,1
            c(i,j) = c(i,j) + a(i,k)*b(k,j)
        end do
    end do
end do
```

This example is written in PCF-FORTRAN, one parallel FORTRAN-extension (another such proposal is FORTRAN-D) to which several important manufacturers are committed for their future-machines.
PCF-FORTRAN can supply variables with additional attributes e.g.:
scatter: Arrays should be cut into as many pieces of equal size as there are processors available.
private: To achieve faster access, each processor should hold the complete reference-tables for the arrays.
Similar to C or FORTRAN-90, arrays can be created dynamically, therefore the allocate. The attribute everywhere means that on each node the space for a part of the array should be reserved.
The directive **parallel do** means that the following loop(s) should be processed in parallel.
depth(2): Parallel execution of the two outermost loops (i and j). This means that only references to the array b need access to memories of other processors, the rest is local.
chunksize(equal): Each available processor gets the same amount of work.
coalesce: The parallelized loops can be treated as one. This saves administrative overhead.

Array b is needed on each node completely, although it is distributed over the nodes in the same manner as a and c. Non-local data must be accessed over the interconnection-network. On a BBN TC-2000 with a communication-bandwidth of 40MB per link and a startup of $2\mu s$, the access to non-local data is about a factor of three slower than to local data. The user is in no way obligated with explicit communication. The access to non-local memories is the limiting factor on this kind of machines.

Example: Loop with condition (BBN)

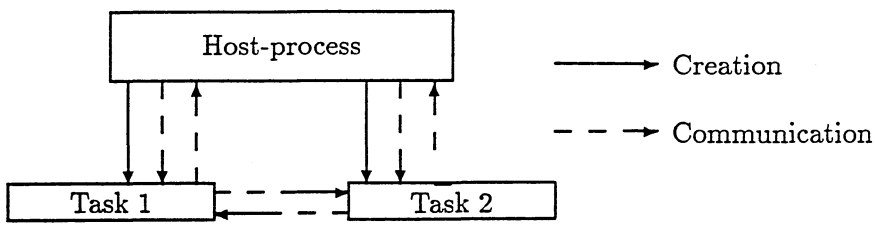
```
real scatter x(1:n), y(1:n)

allocate(everywhere) x, y

parallel do, chunksize(equal)
do i = 1,n,1
    x(i)  = ......
    if (x(i) .gt. 0.0) then
        y(i) = sin(x(i))
    else
        y(i) = cos(x(i))
    end if
end do
```

In this second example all references are only to the local memories, therefore no processor must hold the reference-tables for the complete array (no private-attribute). If we assume superscalar processors as nodes, what is realistic, this loop could run theoretically with an efficiency of 100% and full speedup.

6 Message-Passing

Today, the message-passing programming-model is some kind of a standard for distributed-memory machines.
This model is based on independant processes, running asynchronously and exchanging messages if there are data-dependencies. The message-passing programming-model is only suitable for MPMD-machines, not for MIMD-machines in general.
One of the reasons for the big success of this model is its independency of

the hardware-topology. The user only sees processes and communication and can write portable programs on this base. At least he should. On many machines however this concept is implemented in such a crude way, that at least the number of processors and the local structure is visible (e.g. the number of links per processors). In addition, portability is restricted because there is no standard for message-passing in old programming-languages like C or FORTRAN. Solutions with portable interfaces between the user-program and the manufacturer's proprietary constructs have been developped by several institutions, e.g. by Argonne/GMD (PARMACS).
A complete job within this concept consists of a master-part (having a similar role as main-programs in sequential programming) and tasks called by the master which are asynchronous subroutines. When a task has been started by the master, the master continues execution immediately with the statement after the call. Master and tasks communicate with each other via message-passing on the interconnection-network.



A process can principally be either in the active state or waiting for message-exchange. To work with a message-passing system, the user must be supplied with the following constructs:

- process-creation (task-start).
- send/receive (communication and synchronisation)
- memory-allocation
  Dynamic memory-allocation is necessary to keep variable dimensions in the tasks, similar to subroutines.

For the implementation there are several possibilities:

- The use of languages that are already prepared for tasking and communication, e.g. Ada, Modula-2 or Occam.
- Language-extensions like MIMD-FORTRAN (Suprenum) or Par.C (Transputer).
- Subroutine-libraries(problem: call-overhead)
- Compiler-directives

Interprocess-communication can be implemented either synchronously or asynchronously.

We speak of synchronous communication (rendez-vous), if the sending process must wait for the receiver to be ready. Each communication-event by this synchronizes both partners.

In asynchronous communication (send-no-wait) the sending process deposits the message in the mailbox of the receiver and continues execution. The receiver takes the message as soon as he is ready.

Synchronous communication wastes efficiency by blocking the sending process, whereas asynchronous communication is much more expensive concerning communication-startup. Therefore both concepts should be supplied, to leave it up to the user what is optimal for his application.

Message-passing-machines are best suited for problems that can be split into substructures and that use algorithms that keep a certain locality. Thereby it makes no principal difference whether there are different kinds of tasks running on each node as in plant-simulations, or the same as in domain-decomposition-methods.

Example: matrix-multiplication (Suprenum)

```
integer loc            # logical number of actual task Task
taskid proc(1:nproc)   # array for process-IDs of all tasks
real a(1:n/nproc,1:n), b(1:n,1:n/nproc), c(1:n/nproc,1:n)

receive(tag=1, taskid=master()) a, b

do l = 1,nproc,1                                        # Domains
   do i = 1,n/nproc,1
      do j = 1,n/nproc,1
         do k = 1,n,1
            c(i,(l-1)*nproc+j) = c(i,(l-1)*nproc+j) + a(i,k)*b(k,j)
         end do
      end do
   end do
   if (loc .eq. nproc) then                             # Ring!
          send(tag=2, taskid=proc(1))) b
   else
          send(tag=2, taskid=proc(loc+1)) b
   endif
   if (loc .eq. 1) then
          receive(tag=2, taskid = proc(nproc)) b
   else
          receive(tag=2, taskid = proc(loc-1)) b
   endif
end do
send(tag =3, taskid=master()) c                         # Results
```

Example: Loop with condition (Suprenum)

```
do i = 1,n/nproc,1
    x(i)  = ......
    if (x(i) .gt. 0.0) then
        y(i) = sin(x(i))
    else
        y(i) = cos(x(i))
    end if
end do

send(tag=1, taskid=master())) x, y
```

Software Development Tools Update

CXpa
Convex Performance Analyzer

- Current Release V1.3
- Features and Comparison to Other Profilers
- Future Directions

CXpa V1.3 Features

- **Routine Level Profiling**
  - Exact call counts
  - Inclusive/Exclusive times (min, max, avg)
  - Call graph including call counts

- **Loop Level Profiling**
  - Annotated loop transformation report
  - Inclusive/Exclusive CPU times for loop nests
  - Number of times executed
  - Iteration counts (min, max, avg)

- **Block Level Profiling**
  - Number of executions
  - % of total executions (per routine)
  - Line number and PC
  - List of blocks not executed

CXpa V1.3 Features (cont.)

- **P-Region Level Profiling**
  - Total CPU time, WCT and PVT
  - CPU/PVT ratio (est. parallel speedup)
  - Thread distributions indicating CPU time, WCT, and chore counts per thread
- **Profiler robustness**
  - Enhanced to ignore sections of code that have incorrect compiler-generated instrumentation
  - Doesn't abort when incorrect compiler generated instrumentation is encountered, but allows profiling of the rest of the application

CXpa
Future Directions

- X/Motif User Interface
- Multiple Graphics Analysis Modes
- Statistical Profiling
- Self-paced online tutorial
- Online help
- Updated Users Guide and Reference Material

CXdb
Convex Visual Debugger

- V1.1 Features and comparison to other debuggers
- Future Directions

CXdb V1.1 Features

- **Traditional capabilities**
  - Create a process from an executable file
  - Examine a core file or a checkpoint file
  - Start, stop, and continue process execution
  - Examine and modify program variables, registers, and the stack
  - Set breakpoints to stop execution
  - Step process execution line-by-line
  - Debug at the source level or the assembly level
  - Modify the environment in which your process runs

CXdb V1.1 Features (cont.)

- **Extra capabilities**
  (Really neat things dbx derivatives don't do)
  - Attach to and debug any running process
  - Execute debugger commands while your process is running
  - Fully debug code optimized up to the -O1 level
  - Control execution by source unit
  - Set eventpoints to stop execution when the value of a variable changes, a signal is caught, or an expression becomes true
  - Specify a complete set of actions to take when an eventpoint stops execution
  - Create aliases and macros for commonly used commands
  - Command Composition and Completion

CXdb V1.1 Features (cont.)

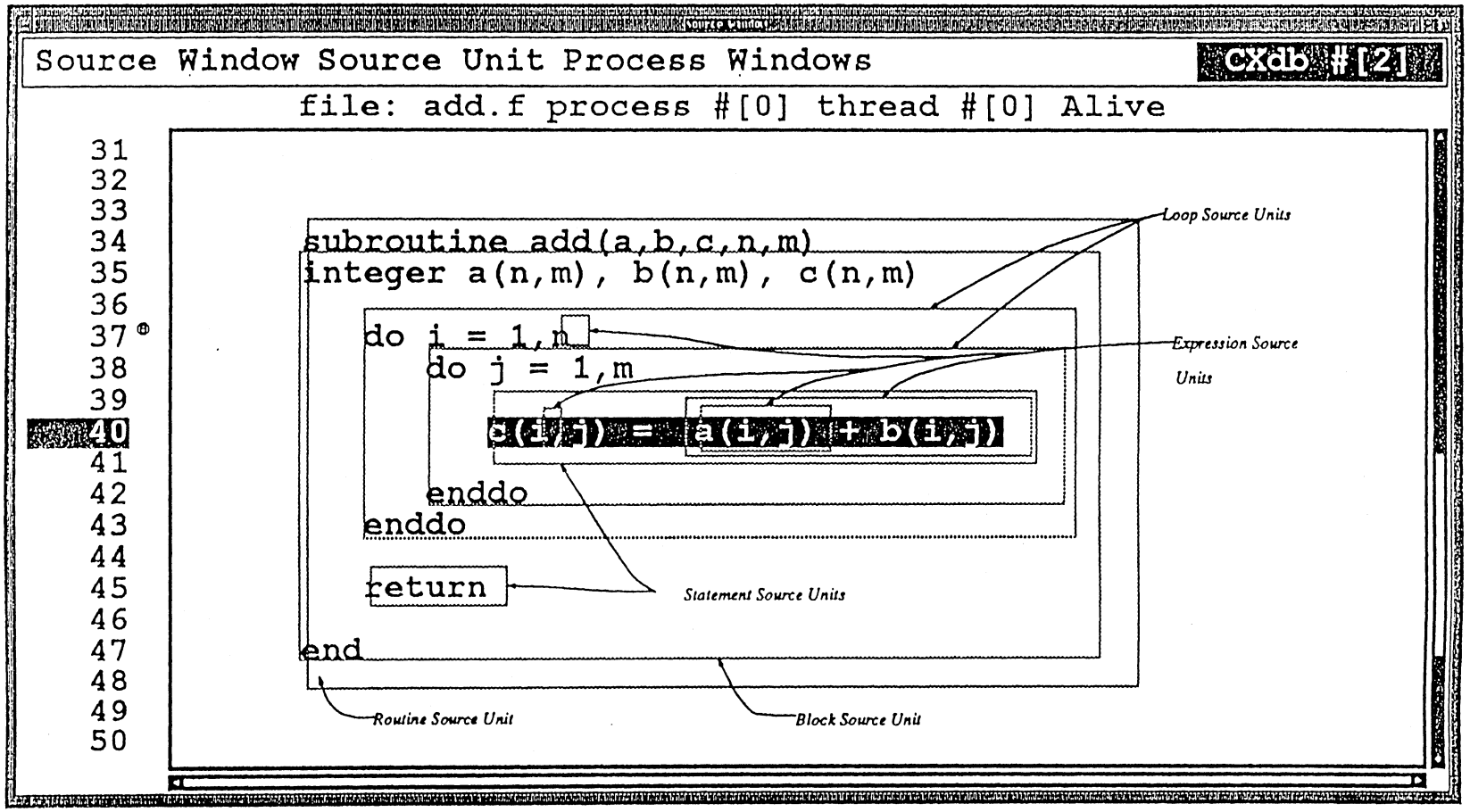
- **Really neat things (cont.)**
  - Debug at the machine instruction level, with complete access to the machine state including scalar, vector, and C2 or C3 communication registers
  - Display 2-dimensional arrays as a table
  - Debug programs written in MIXED Fortran and C
  - Access static program variables not visible from the current point of execution
  - Use debugger variables to store information without affecting program variables
  - Debug, edit, and re-compile your program without leaving the CXdb environment

CXdb V1.1 Features (cont.)

- **Special Features**
  - Multiple windows:
    - ✔ Source
    - ✔ Command
    - ✔ Process
    - ✔ Disassembly
    - ✔ Register
    - ✔ Stack
    - ✔ Memory
    - ✔ Help
  - Optional logging of command input, output and error messages
  - Menu system to access all commands
  - Extensive online help (entire reference guide)
  - Self-paced online tutorial
  - Users, Concepts, and Reference Guides

CXdb V1.1 Concepts

- **CXdb Working Environment**
  - Console working directory
  - Default process settings
  - Command logging settings
- **Program Working Environment**
  - Shell
  - Environmental variables
  - Floating point mode
  - Fixed scheduling
  - Easy control of sequential mode & sequential store (psw bits)
- **Debugger Variables**
- **Source Units**
  - Expression
  - Statement
  - Block
  - Loop
  - Routine


Source Window Source Unit Process Windows
CXdb #[2]
file: add.f process #[0] thread #[0] Alive
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
subroutine add(a,b,c,n,m)
integer a(n,m), b(n,m), c(n,m)
do i = 1,m
do j = 1,m
c(i,j) = a(i,j) + b(i,j)
enddo
enddo
return
end
Loop Source Units
Expression Source Units
Statement Source Units
Routine Source Unit
Block Source Unit

CXdb V1.1 Concepts (cont.)

- **Stepping**
  - Beginning of a source unit
  - End of a source unit
  - Over a source unit
  - Over called routines
  - By machine instruction
- **Eventpoints**
  - Breakpoints
  - Tracepoints
  - Memory Watchpoints
  - Relational Watchpoints
  - User-defined handlers
- **Command and Initialization Files**
- **Signal Handling**
  - Print a message?
  - Pass to process?
  - Stop process?

CXdb V1.1 Concepts (cont.)

- **Optimized Code**
  - Supports all levels of optimization
  - Full support for compiler-synthesized variables
    - ✔ loop induction variables
  - Source Units and optimization
    - ✔ source window and disassembly window together allow user to synthesize information on compiler optimizations
  - Accurate results
    - ✔ understands current variable location - register(s) or memory
  - Graceful handling of eventpoints
    - ✔ at first instruction of all code regions associated with source unit

CXdb V1.1 Concepts (cont.)

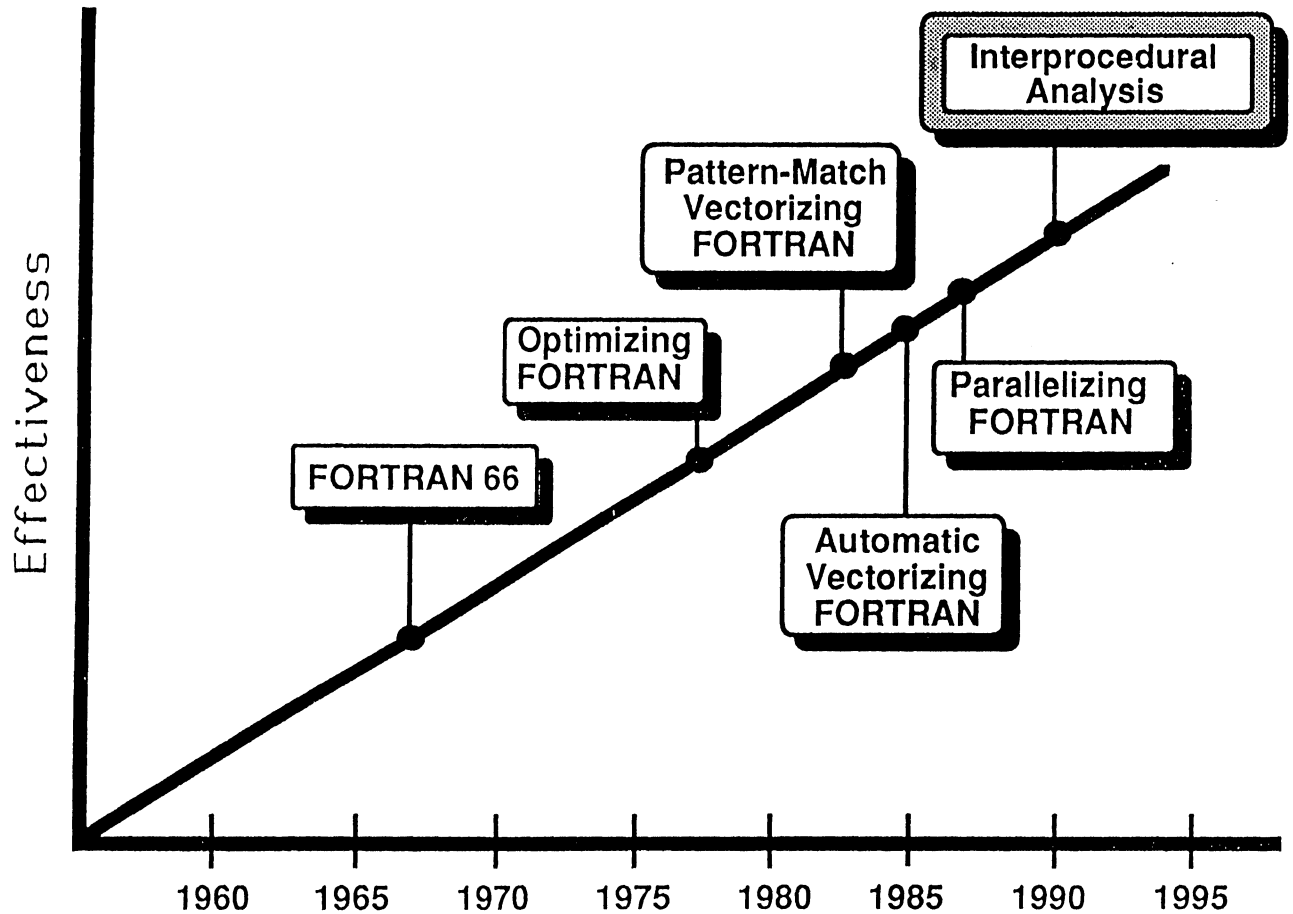
- **Optimized Code (cont.)**
  - Graceful stepping
    - ✔ visually depicts code reorganization by highlighting current source unit(s)
    - ✔ expression source units support stepping through optimized code
  - Parallel debugging
    - ✔ supports both -O3 compiled code and user-created parallelism
    - ✔ eventpoints targeted at specific threads
    - ✔ memory window per thread
    - ✔ machine state per thread
    - ✔ can execute individual or multiple threads concurrently
    - ✔ Spawn & Join eventpoints for threads

CXdb Future Directions

- Remote debugging support
- Multi-process debugging support
- Data browser
- Application Compiler support

COMPILER TECHNOLOGY

Compiler Chronology



COMPILER TECHNOLOGY

Application Compiler Project Goals

Goals

- Increased developer productivity
  - Trade CPU time and disk space for developer time in the application development process
  - Increase automatic error checking capabilities
  - Reduce sophistication required to produce highly optimized executable code
- Increased application code performance
- Increased reliability of the application

COMPILER TECHNOLOGY

Application Compiler V1.0

Features

- Supports both FORTRAN and C
- Automatic inlining of FORTRAN and C
  - Allows manual overrides
  - Enhances current CONVEX FORTRAN inlining
  - Inlined routines must be of the same language
- Interprocedural error checking
  - Has visibility into all routines in the application
  - Can analyze side effects
  - Provides extensive reports

COMPILER TECHNOLOGY

Application Compiler V1.0

Features

- Interprocedural constant propagation
  - Provides more information to the optimizer thus increasing the potential for optimization
  - Can clone procedures when appropriate
- Interprocedural pointer tracking
  - Improves optimization of C program
  - Can distinguish first-level indirections
- Optimization summary provided

Shipped - May 91

COMPILER TECHNOLOGY

Interprocedural Error Checking

Goal: Find potential errors in applications not found by other means

Limitations of other methods

- *lint* doesn't do interprocedural alias or side effect analysis
- Procedural compilers process only one procedure at a time

Application Compiler has visibility of all routines in the application

COMPILER TECHNOLOGY

Errors Detected

Code	Mis-Matched Args	Wrong Number of Args	Mis-Matched Return Type	Invalid Aliases	Scalar Passed To Array	Invalid Subscript	Variables Not Initialized
ADM	23						114
QCD						1	57
MDG	2				2	2	29
TRACK	2						5
BDNA	2						22
OCEAN	119						14
DYFESM	1				4	3	229
MG3D	1					1	49
ARC2D	10						13
TRFD							
FLO52							22
SPEC77	98	1			6		24

COMPILER TECHNOLOGY

Interprocedural Constant Propagation

Goals

- Find interface variables (arguments and globals) which always have the same value on entry to a given procedure
- Modify that procedure by substituting the constant value for the name of the interface variable

COMPILER TECHNOLOGY

Procedure Cloning

Definition of procedure cloning

- Makes a complete copy of a procedure
- Changes the caller to invoke new procedure
- Tailors code of the called procedure to reflect the context of the call site

Procedures are cloned when

- Constants can be propagated into the clone
- Optimization *may* be improved

COMPILER TECHNOLOGY

Pointer Tracking

Issues

- Procedure compilers must assume global and formal pointers can point anywhere
- Any pair of use/assign de-references of such pointers is a potential recurrence
- Potential recurrences prevent vectorization and parallelization
- Current solution is to use directives or command line options

COMPILER TECHNOLOGY

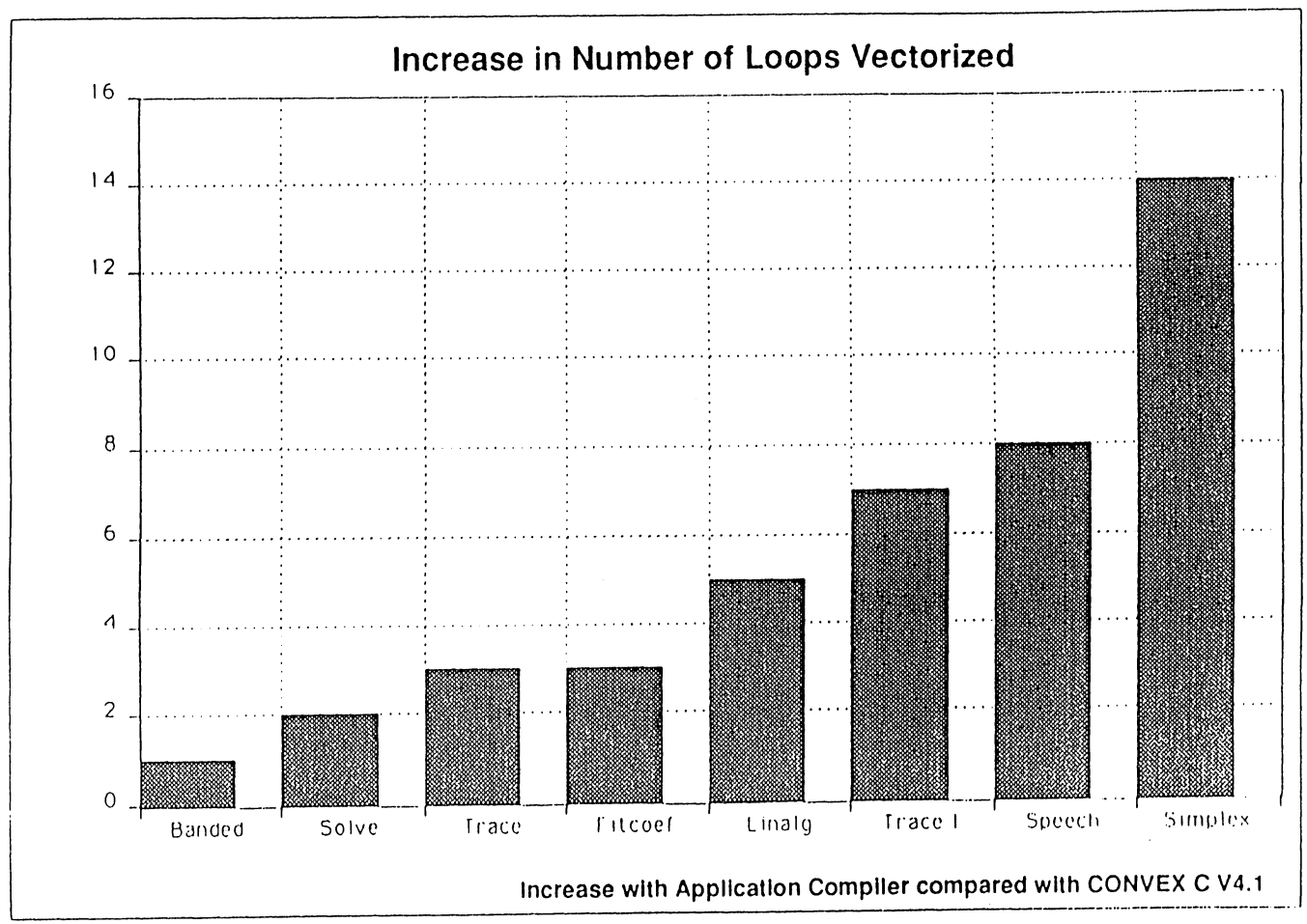
Pointer Tracking

Application Compiler -- compiler solution

- Interprocedural pointer tracking symbolically analyzes the memory locations to which each pointer refers
  - If two pointers never point at the same location, there is no potential recurrence
  - Handles static, automatic, and heap storage
  - Limited to single level of de-reference

COMPILER TECHNOLOGY

Pointer Tracking Results



COMPILER TECHNOLOGY

Inlining Definitions

Procedure Inlining - Replace the invocation of a procedure with the actual body of the called procedure

Automatic Procedure Inlining - Find procedures and procedure calls which, if inlined, make the greatest improvement in application performance

COMPILER TECHNOLOGY

Automatic Inline Substitution

Automatically selects the best calls to inline

- Provides directives for manual override
- Provides command line option to control heuristic

Language independent

- FORTRAN and C in first release
- Inlined routines must be of the same language

Eliminates current FORTRAN (fc) inlining restrictions

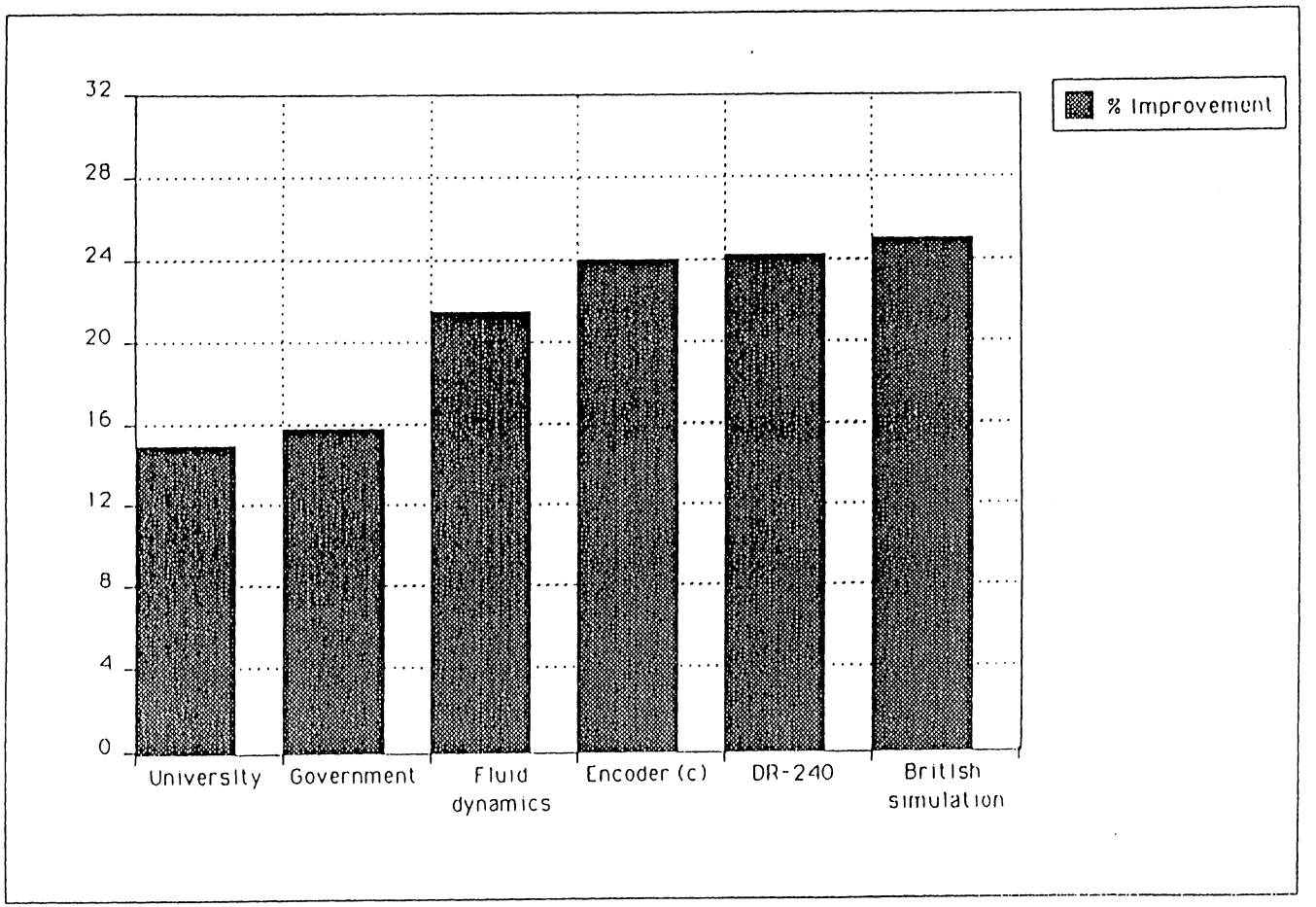
COMPILER TECHNOLOGY

Optimizations Performed

Code	Times Procedure Inlined	Times Procedure Cloned	Propagated Constants Used
ADM	84		11
QCD	39	1	7
MDG	12		139
TRACK	23		30
BDNA	29	1	4
OCEAN	49		17
DYFESM	24	1	55
MG3D	7	9	141
ARC2D		1	205
TRFD	6		
FLO52	36		53
SPEC77	30	6	109

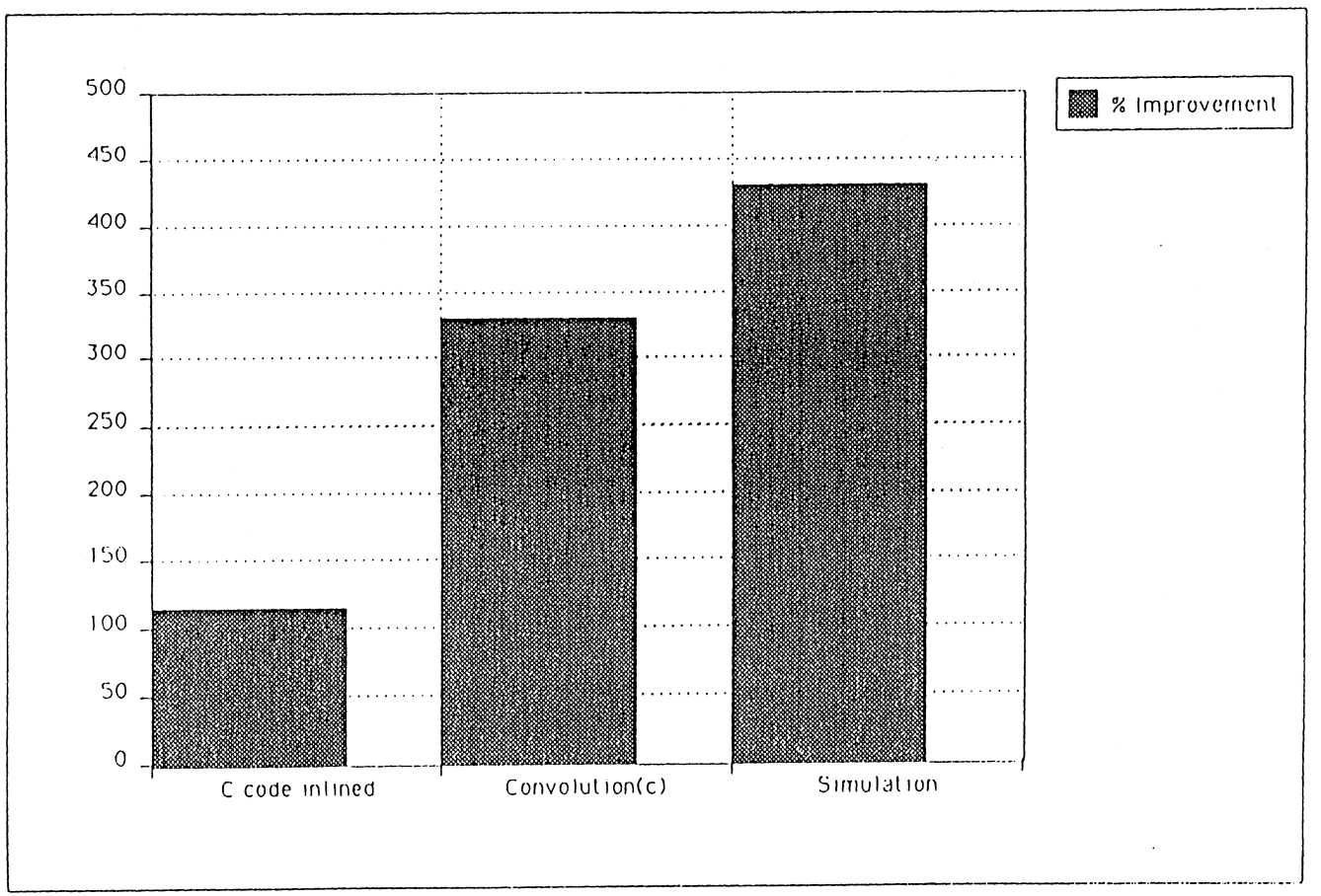
COMPILER TECHNOLOGY

Typical Benchmark Results



COMPILER TECHNOLOGY

Exceptional Benchmark Results



COMPILER TECHNOLOGY

Large Program Example

Computational Fluid Dynamics Code

- 214,500 lines of source code
- 971 source files
- Optimizations performed
  - 1576 calls inlined
  - 221 clones made
  - 1774 constants propagated
- Errors found
  - 1133 mismatched argument types
  - 1041 COMMON/formal aliasing violations
  - 9 array subscripting violations
  - 5701 uninitialized variables

COMPILER TECHNOLOGY

Application Compiler Futures

Enhancements for V1.1

- Use of profile data in optimization
- Improved library support
- Interprocedural storage optimization
- Enhancements to procedure cloning and automatic inlining
- Additional directives and options
- Bug fixes

Ships - 1Q92

STRATIFICATION

A Strategy for Information Processing in an Academic Environment

J Olsen
DOU, 11 Niels Bohrs allé
DK 5230 Odense M
Denmark

Definitions

STRATIFICATION

1) Arrange in strata - e.g. layers [of rock] or classes [in society].
Miscellaneous english dictionaries.

2) Divide an inhomogeneous population [collection of data or elements] into more homogeneous [manageable] subgroups - strata. [Used to simplify the management of the population].
Basic textbook of Business Statistics

The author :
Jørgen Olsen, Director
DOU *(Dep. of Academic Information Systems)*
Niels Bohrs allé 11
DK 5230 Odense M, Denmark
☎ +45 66 13 08 27, Fax: +45 66 12 33 66
E-mail: masjol@dou.dk

Table of Contents **Page no**

Preface

The creation of a *presentation on a specific topic* is somewhat different from the presentation of a *scientific paper.*

In the later case you submit the results of months or years of hard work for peer review and publication, and if you are invited to present it at a conference you come with an abstract based on your paper.

In the first case nothing but an abstract exists before the actual presentation is given for the first time. The creation of the presentation is an ongoing process culminating with the actual presentation. Before that climax only bits and pieces (of which a number will disappear before the presentation takes place) exist. Thus the genuine paper describing that presentation can only be written afterward.

With these remarks I leave you with the bits and pieces in existence a ten days before the presentation.

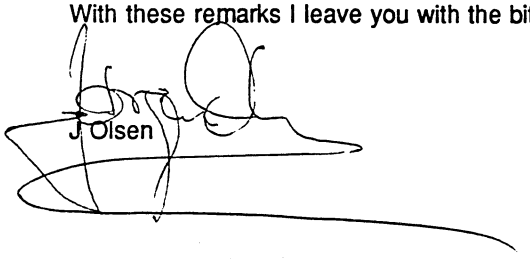


Table of Contents **Page no**

1 A Short Commercial

The Users:

Odense Teknikum
Students: ≈ 1900, Staff: 240

BSc in Civil-, Mechanical-, Shipbuilding-, Chemical-[1], Power- & Electronics-Engineering
MSc in Computer Technology[1]

1. Joint with the University

Odense University
Students: ≈ 8000, Staff: 800 (*500 Academic*)

Faculties of: Humanities
Medical Sciences
Natural Sciences
Social Sciences

Degrees to the highest level of academic achievement

DOU - *Datacentret ved de Odenseanske Uddannelsesinstitutioner* - is the Department of Academic Information Processing serving Odense Teknikum (OT) and Odense University (OU).

Number of persons in Information Processing (IP)[2]	
DOU - Academic IP	10
OU - Academic IP	8
- Administrative IP	5
OT - Academic IP	0[3]
- Administrative IP	0

2. Persons with IP as their job - e.g. neither teachers, students, instructors nor users.

3. Service Bureau oriented applications.

2 Existing Hardware

See Appendix A

3 Supercomputing in Denmark

Many danish researchers have had access to supercomputers all over the world for a good many years - but within the country it is a shorter story:

1985 A recommendation from
The Advisory Board to the Minister of Education on the Use of Computers in Education & Research
to buy a CRAY for Research and Education was killed by the politicians after a group of Computer Scientists from *a danish university* had declared that large computers were a thing of the past, while the future belonged to the PC/Mac/Workstation (*e.g. give us the 7 Mill $ and let us use them to buy these nice and democratic items*).

1986 a) The rest of the research community pointed out - that in several areas - basic research as well as applications - there were - and still are demands for MIPS & MFLOPS.
b) Aarhus University installed an *Alliant (FX8 - 3 CPU's).*

1987 UNI-C (The National Center for Academic Computing) installed an *Amdahl VP1100.*

1988 IBM donated a (used) *3090-180VP* to The technical University of Denmark.

1989 Someone tried to find out why nobody used the *VP1100.*

1990 DOU emptied all pockets in Odense and installed a *CONVEX 201* after an EEC-tender.

1991 a) UNI-C installs a *CONNECTION MACHINE.*
b) As part of a plan to make supercomputing power available to the danish researchers through a decentrallized strategy, DOU receives a sum of money for the expansion of the existing system *(for reasons that we shall not go further into, the upgrade has to be done through another EEC-tender).*
A research group in Aarhus gets a larger sum so that they can replace the existing system.
c) The condition in both places is that part of the capacity of the installed systems must be made available to the Research Councils. They can then distribute it - through a peer review procedure - to researchers - all over the country - with a need for computing power.

1992 UNI-C would like to replace the *VP1100* with a GIGAFLOP computer. The need for such a system is under investigation for the moment.

4 How did we (Odense) get where we are?

1978 - 89 Computing power in Odense was supplied by a mainframe with upgrades in 81 & 85 (Univac/Sperry/Unisys).

The local strategy - that parallels what happened in several other places - can be described as indicated on the following page.

1975 - 77	Let there be a local computer!
1978 - 83	Let there be terminals! And more terminals!
1984 -	Let there be PC's/Mac's/Personal Workstations!
1986 -	Let there be a LAN!?!
1988 -	Let there be a replacement of our mainframe - but why & with what?

The problem with the replacement - nobody wanted to continue with a mainframe where everybody was fighting everybody so that response times and turnaround times were seen as an overwhelming problem.

Our resulting strategy:

STRATIFICATION -	**Future computing power shall be supplied through the LAN from dedicated systems.** **ALL with UNIX as the operating system!!**

Reactions from friendly colleges:

You must be out of your ..... minds!!
Without a mainframe you will be dead!!

The results implemented 87 - 90:

a) A LAN (Bridge-3Com, started as an experimental installation with the Faculty of Social Science in 86)

b) A 64+-user system for students (UNISYS7000/51, basic courses).

c) A 32-user system for miscellaneous applications used by various researchers and graduate students (HP9000/845).

d) A 16/32-user system for the limited group of people with need for number crunching - e.g. the CONVEX 201.

5 Stratification - old wine rebottled?

No! - when you normally talk about decentralizing or distributing your computer resources it often initiated as a reaction to the mainframe and peoples frustration with such a device - resulting in some a policy statement from above.

We did not look at it that way - we were looking at the tasks, we had to accomplice and the alternatives for doing them.

It was recognized, that a major areas of dissatisfaction was where people from different groups - with different needs - had to either fight for resources on the same system or to accept that the best solution to their problems was a bad one.

The *stratification strategy* actually implements itself again and again whether you realize it or not. When all the users of *document processing* on your mainframe started talking microcomputers - they were a homogeneous group - a subgroup of your population of users with specific needs. A group that suddenly discovered that better facilities were available, if they used technology in a different way.

Was it smart to fight them?

As the demands of various groups are very inhomogeneous the standing bottleneck discussion has always focused on whose fault it is - e.g. on who it is that ought to find the money to move the bottleneck.

The *stratification strategy* also reflects the budget structure in our user community, where the autonomous budget structure of the individual faculties makes it difficult to establish cooperation on investments of any kind between faculties unless there is a clear and mutual interest.

In a business - a policy directive from top management would set up a information strategy - but universities do not work like that!

The existing managerial structure in the academic world is somewhere between a *Happening* and *Structured Anarchy*.

Further - we know that *small is beautiful* and that someone has invented the term *downsizing* and a reason for buying their equipment!

But - seriously - there is also some solid reason behind this - TECHNOLOGY!

The technological evolution is

- slowest in the mainframe area - where customers are hooked on megayears of COBOL-code running under a proprietary operating system,
- faster in the Open Systems' environment - especially if you have only unsophisticated needs (such as GIGAFLOPS as opposed to OLTP based on 4GL CASE-oriented MIS),
- fastests in the Personal Workstation area (PC, Mac and similar devices).

Take one gigabyte of mass storage as an example - and compare the prices from a mainframe environment through the open systems environment and down into the workstations (SCSI) area.

6 Where does a CONVEX fit?

Nice and cosy!

It is

- affordable (according to CONVEX's own PR department),
- available, e.g. uncomplicated to run and work with (according to those of our users able to do so),

and - has a wide range of software packages available.

So for an organisation like ours, where the *number crunching community* is a smaller one it fits beautifully.

The easy access has demonstrated its value as The Ministry of Education has offered us the finances for an upgrade - provided that part of resources of the new and larger system will be made available to the rest of the danish research community!

So for the moment we are aiming at replacing our 201

- 35 MFLOPS, 64 MB memory (recently expanded to 256 MB) and 5 GB mass storage (disc),

with at least

- ~ 200 MFLOPS, 512 MB of memory (or more) and a minimum of 15 GB mass storage!!

7 Where do we go from here?

If your strategy is a good one - you will be successful, and able to expand. However - no strategy lasts forever without revision. Revision should be a continuous process - and that is often a problem - when you exist in a participative environment.

In spite of all the tendencies for democracy (a personal computer on every desk) and distribution of resources there is still many services that the users would like someone else to supply.

SAFETY -
Who backups my valuable data?
I just scratched my file by mistake - can I have the latest backup please! Now!?!
SECURITY -
Someone stole my PC and the backup diskettes standing besides it. Is there a more secure way to store confidential data?
LOGISTICS -
It is easier to have one copy of a program available on a server - and then download it to the 100 users connected to that server - than it is to install and maintain one hundred copies all over campus!!
ECONOMICS -
There might be money in having a single data repository (a disk farm or an on-line archive) shared by all your systems.
SERVICES -
The LAN, the Internet connection, the E-mail system, and all the other public available services that people want - if not now then at the same moment you announce that you are not going to provide them in the future!

So just as the *stratification strategy* entices you to unbundle resources into a number of more homogeneous environments - it also induces you to keep things (functions, services) together, if there are advantages in doing so.

8 The future

It is always popular to end some with philosophical reflections about the glorious future one perceive. So let me not disappoint anyone who has stayed on so far! But please take into consideration that almost any vision of the future presented will be a variant of *the zero scenario*, e.g. an extrapolation of existing tendencies.

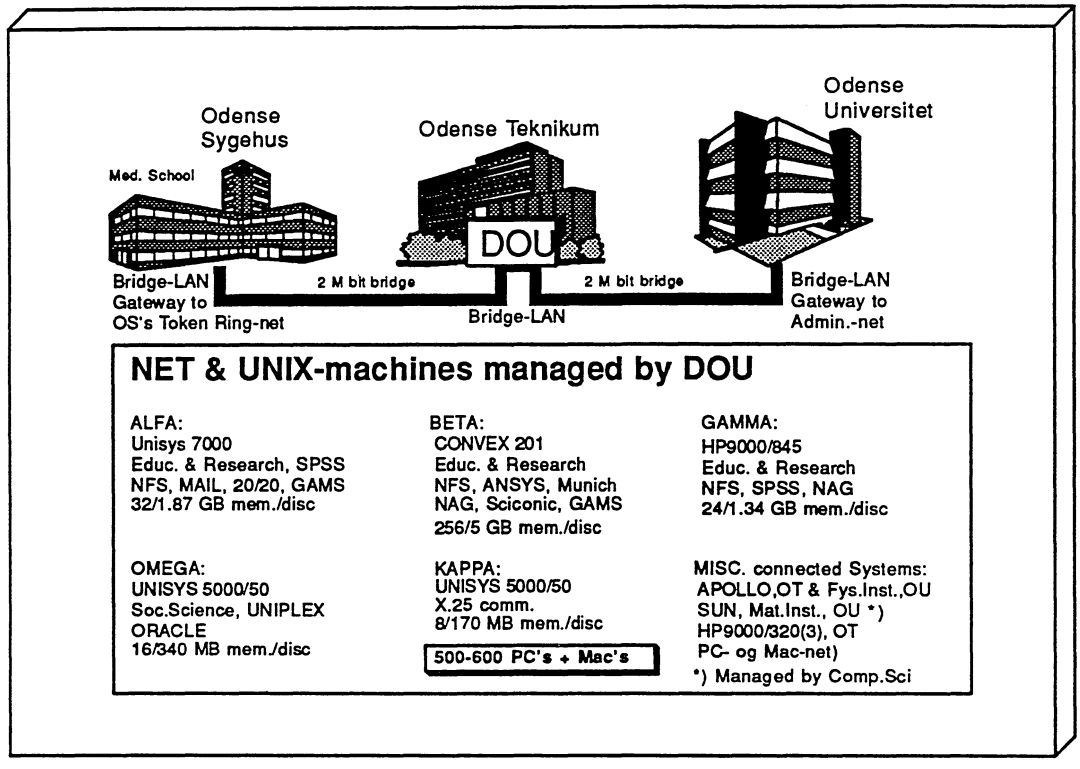
That does not make it worthless. One of the values of creating that scenario is - to be able to consider whether you like it or not! If you like it - work hard to ensure that things continue moving in the right direction. If you do not - take action now - and not in '95!

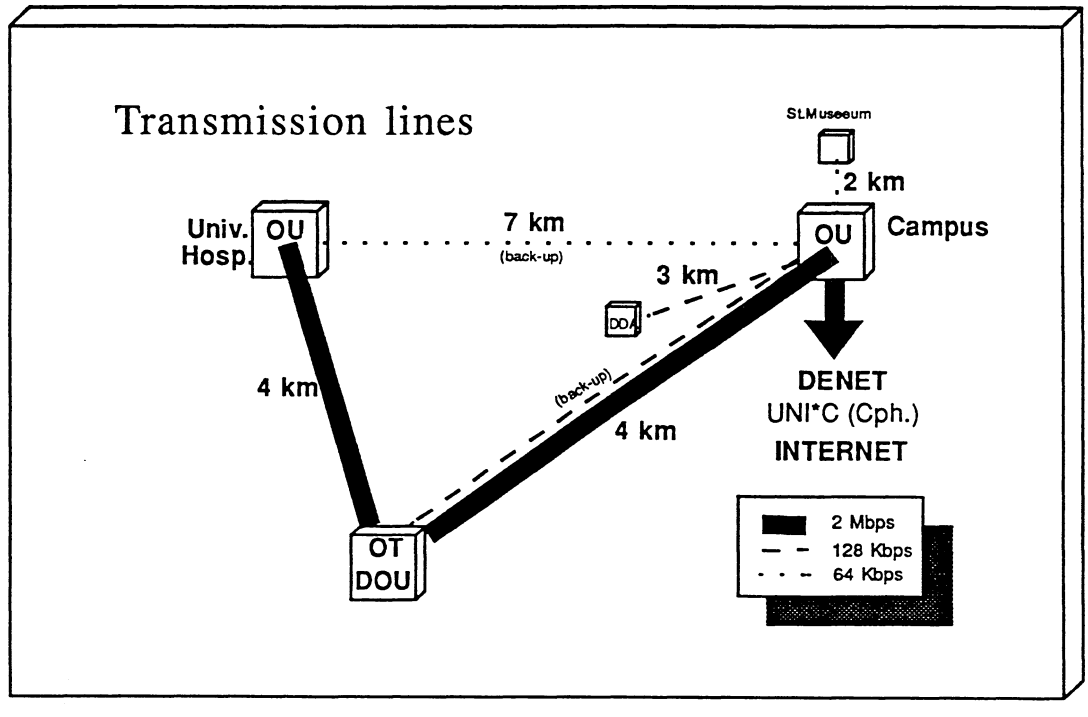
So with these remarks and the reminder that what technology might bring us of new things in the meantime (that we do not know of today) can not be included in the scenario here is my bid on our situation, *when we are in the process of considering a replacement for the system we have not installed yet*!!

Our next super computer will probably have the following specifications :

- ~ 1 GFLOPS processing power, ?? cpu's, but with a interface for a MPP-device (same or other vendor),
- 2-4 GB of memory,
- 40 GB fast mass storage - but with online access to a minimum of 100 GB on a disk farm and/or in an archive with less that 30 sec (?) access.

Time will show what the technology and price will be - and whether it will stand in the operators office or under his desk!


Odense
Sygehus
Med. School
Odense Teknikum
DOU
Odense
Universitet
Bridge-LAN
Gateway to
OS's Token Ring-net
2 M bit bridge
Bridge-LAN
2 M bit bridge
Bridge-LAN
Gateway to
Admin.-net
NET & UNIX-machines managed by DOU
ALFA:
Unisys 7000
Educ. & Research, SPSS
NFS, MAIL, 20/20, GAMS
32/1.87 GB mem./disc
BETA:
CONVEX 201
Educ. & Research
NFS, ANSYS, Munich
NAG, Sciconic, GAMS
256/5 GB mem./disc
GAMMA:
HP9000/845
Educ. & Research
NFS, SPSS, NAG
24/1.34 GB mem./disc
OMEGA:
UNISYS 5000/50
Soc.Science, UNIPLEX
ORACLE
16/340 MB mem./disc
KAPPA:
UNISYS 5000/50
X.25 comm.
8/170 MB mem./disc
500-600 PC's + Mac's
MISC. connected Systems:
APOLLO,OT & Fys.Inst.,OU
SUN, Mat.Inst., OU *)
HP9000/320(3), OT
PC- og Mac-net)
*) Managed by Comp.Sci


Transmission lines
St.Museeum
2 km
Univ.
Hosp.
OU
7 km
(back-up)
OU
Campus
3 km
DDA
4 km
(back-up)
4 km
DENET
UNI*C (Cph.)
INTERNET
OT
DOU
2 Mbps
128 Kbps
64 Kbps

The LAN - your most complicated system

- **Cabling**
  Ethernet - 10 segm. - 5.000 m
  Thin Ethernet - 50 segm. - 10.000 m

- **Bridges, repeaters**
  Local - 6, remotes - 5, repeaters - 12

- **Gateways**
  Terminal servers (50 each with 10 conn.)
  Kinetics boxes (12) - to Appletalk NW.
  ---> Novell NW (2, 140 users)
  ---> 3Com 3+Share (3, 100 users)
  ---> 3Com 3+Open/LAN-mgr 2.0 (1, 15 users)

- **Total no of connections**
  250 PC's - direct
  250 Mac's via Appletalk
  300 PC's via Terminal servers
  + UNIX-systems, HP-Apollo's & Sun's

ROBOTIC CARTRIDGE LOADING AND UNATTENDED PROCESSING ON THE CONVEX C210 AT KSEPL

by
Simon Verdouw

ABSTRACT

This presentation gives an overview of the use of the Convex C210 at KSEPL, with particular emphasis on a project that integrates the Convex with the Storage Tek Automated Cartridge System (ACS).

The project is part of a larger effort to extend unattended processing at KSEPL and other work being done in this area is discussed as well.

SHELL AND RESEARCH

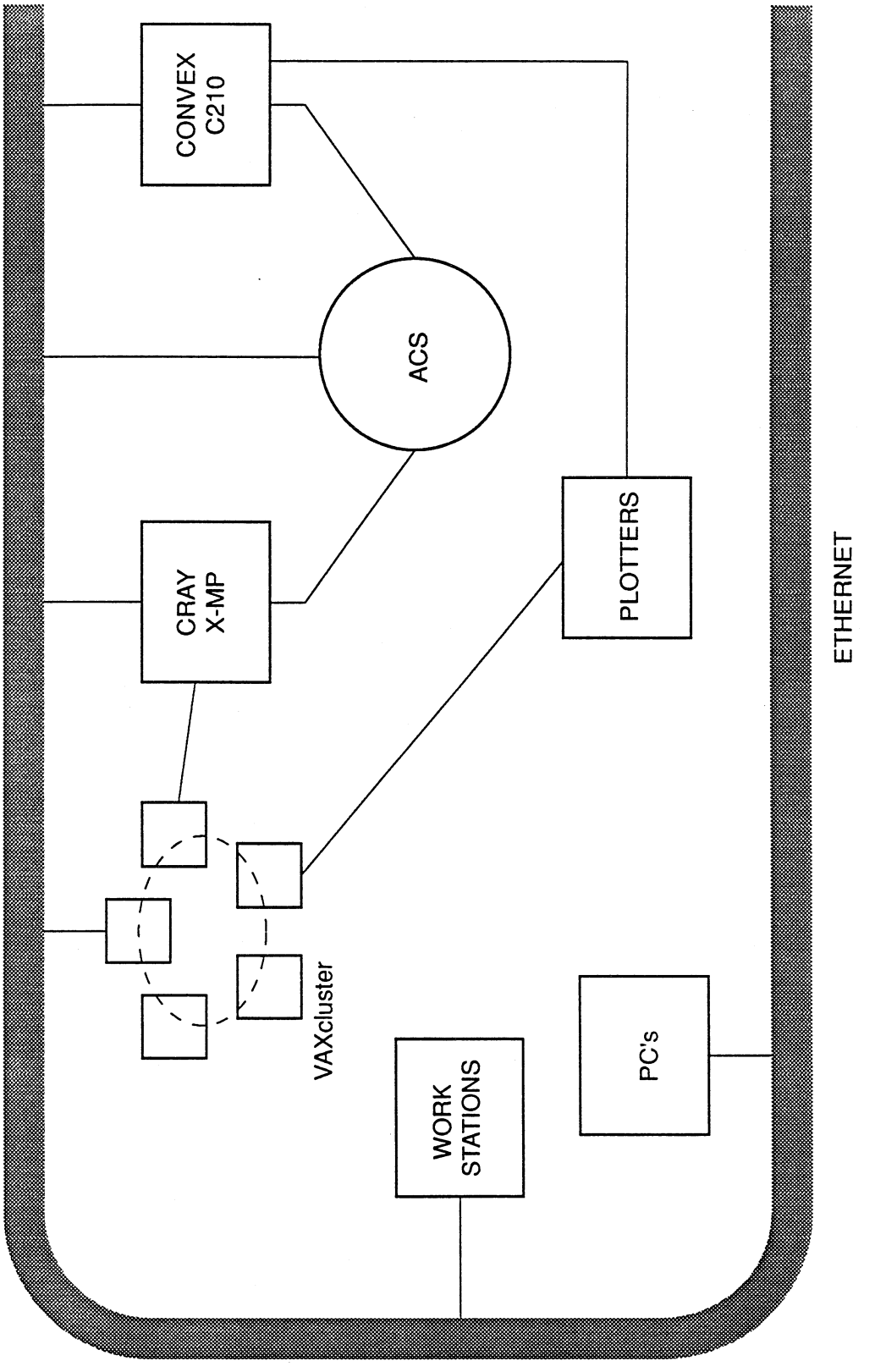
Amongst other activities the Royal Dutch/Shell Group of Companies encompass oil and gas exploration and production, research and development, carried out by approximately 7000 people in 15 laboratories located in eight countries. KSEPL is devoted mainly to research in support of exploration and production activities.

RESEARCH AT KSEPL

The objective of the work of the 650 people employed at KSEPL is to contribute to the goal that the Group acquires the largest possible reserves of oil and gas and that it produces as much as possible of these at the lowest possible cost in a manner that is safe and imposes a minimum burden on the environment. Research at KSEPL involves numerous scientific and engineering disciplines that deal with various types of fundamental and applied research. Because the systems studied at KSEPL usually cannot be observed directly, investigators are obliged in many cases to make intensive use of laboratory experiments. An alternative approach to the problems related to the exploration and production of oil and gas is to develop mathematical equaitons and computer programs for modelling and simulation. This type of research constitutes a large part of the work done at KSEPL. Computers are also used for analysing the multitude of data - in particular seismic data - that are collected in probing the subsurface layers of the earth for oil and gas deposits.

Because KSEPL's expertise is in part acquired through the processing, analysis, visualisation and interpretation of data, the scientists make use of an ardenal of computers and workstations as well as supercomputers and several transputers.

Since the work done at KSEPL is aimed, directly or indirectly, at supporting the exploration and production activities of Shell operating companies, an effective transfer of research results to the operating companies is essential. A good example is the processing of seismic data. At KSEPL, programmers and geophysicists work side by side in support of seismic research in the development of customised software for seismic processing.



KSEPL Computing Environment

FIG. 1

KSEPL COMPUTING ENVIRONMENT

The KSEPL computing environment consists of a Local Area Network connecting a number of computing services. These services, ranging from personal computing to a supercomputing servive, are briefly described below. Figure 1 shows the current hardware configuration.

A Cray X-MP EA (Extended Architecture) with one CPU and 64 Mword main memory, combined with a Convex C210 (128 Mbyte main memory) sharing a Storage Tek Automated Cartridge System (ACS) compose KSEPL's ***supercomputing service.*** The supercomputers are mainly used for the processing of large volumes of seismic data applying a seismic data processing package developed and maintained at KSEPL. Because of the integration through the Automated Cartridge System the Cray and Convex can operate in an unattended mode of operations while processing data contained on cartridge tapes.

Another application of the Cray is developing reservoir simulation models whereas the Convex is also used for plotting purposes and seismic software development.

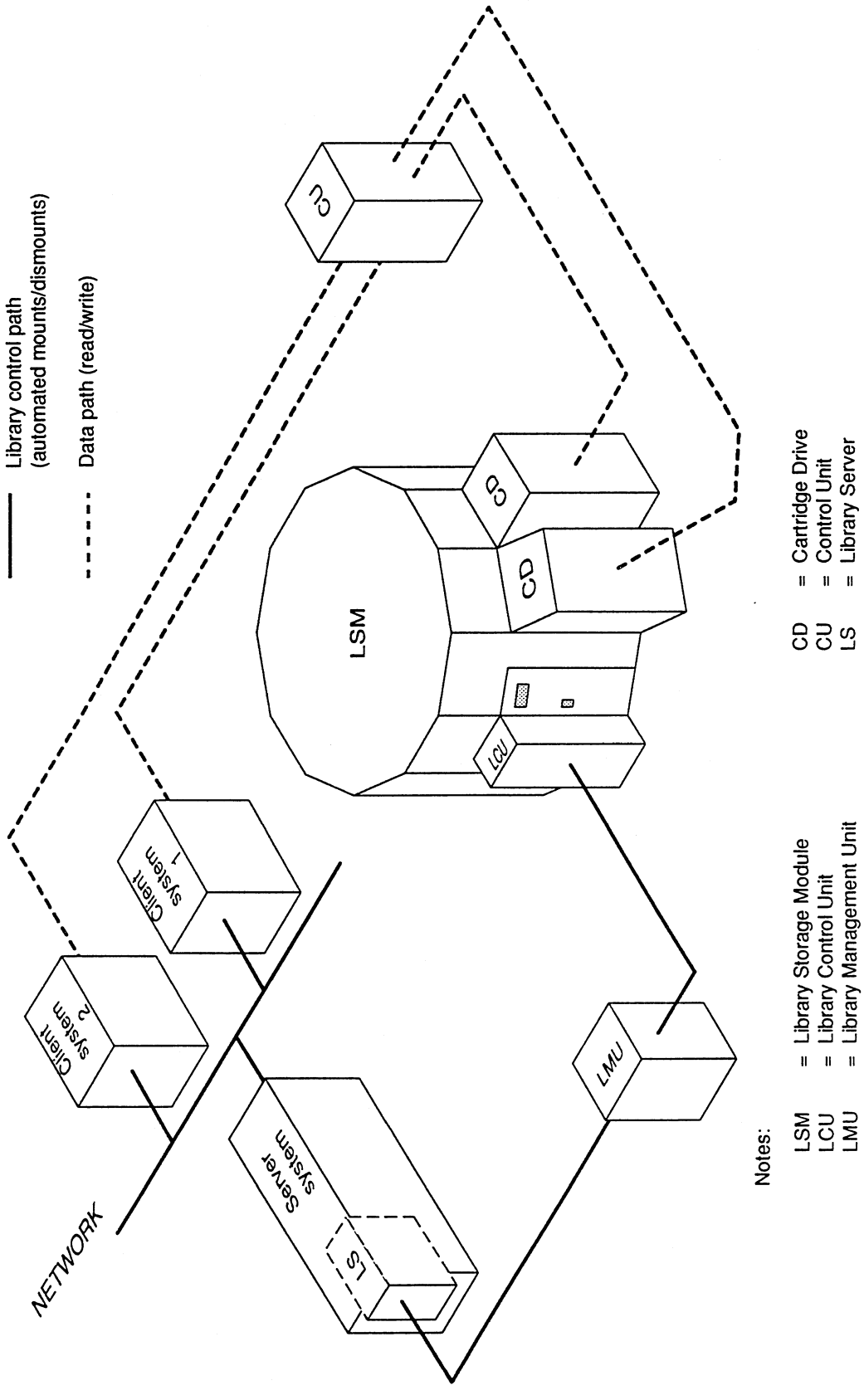
The ***VAXcluster***, using DECnet, TCP/IP or Cray USCP protocol, is used as the front-end system for accessing both Cray and Convex. All work to be processed by the supercomputers is prepared on and submitted from the VAXcluster. Results are received on the VAXcluster after the jobs have been processed. The VAXcluster is also used for the maintenance and development of KSEPL's seismic software package.

Specific tasks that were previously performed on the VAXcluster have been moved to ***workstations*** situated in different departments. These workstations are either Local Area Vax Clusters (LAVC's running VAX/VMS) or Unix workstations organised as NFS clusters. Some of the workstations are also used as front-end systems for Cray and Convex.

Activities such as word processing, spread sheet usage, graphics etc. are performed on ***personal computers***. A NOVELL PC network server system has been implemented for this purpose.

The final product of seismic data processing is a seismic plot. Plotters of different types (electrostatic and film, monochrome and colour) have been connected to the VAX as well as to the Convex to enable bulk and quality plotting. Printing is done on Zerox laser printers connected to the VAX and printing A4 size paper.

The ***Local Area Network*** at KSEPL currently consists of a baseband IEEE 802.3 Ethernet. Computing equipment is connected to the different Ethernet segments and communicates using TCP/IP, DECnet or LAT type protocols.



Automated Cartridge System Overview

FIG. 2

AUTOMATED CARTRIDGE SYSTEM

The Automated Cartridge System (ACS) provided by Storage Tek is a fully automated (standard 3480) 18-track cartridge-based, storage and retrieval system. It provides automated tape cartridge library services to a network of heterogeneous client systems. The client systems may range from workstations to supercomputers. The basic hardware component of the system is a Library Storage Module (LSM), a 12-sided structure containing storage cells for approximately 6000 tape cartridges, a robot that retrieves and moves cartridges, and apertures in the walls of the structure through which cartridges can be passed to load and unload cartridge drives outside the LSM.

The ACS is controlled by the Storage Server Software residing in a server system, works within a UNIX system environment and uses BSD sockets as the interprocess communications mechanism. Each client system should provide an interface (control path) to communicate with the Storage Server in order to request mounting and unmounting of cartridges within the LSM. Figure 2 shows an overview of the ACS.

In May 1990 the Automated Cartridge System was successfully connected to the Cray and integrated in the processing environment. To connect the Convex to the ACS as well, a data path (hardware interface) between the Convex and the Storage Tek 4480 cartridge drives as well as a control path (software interface) to the ACS Storage Server software were required.

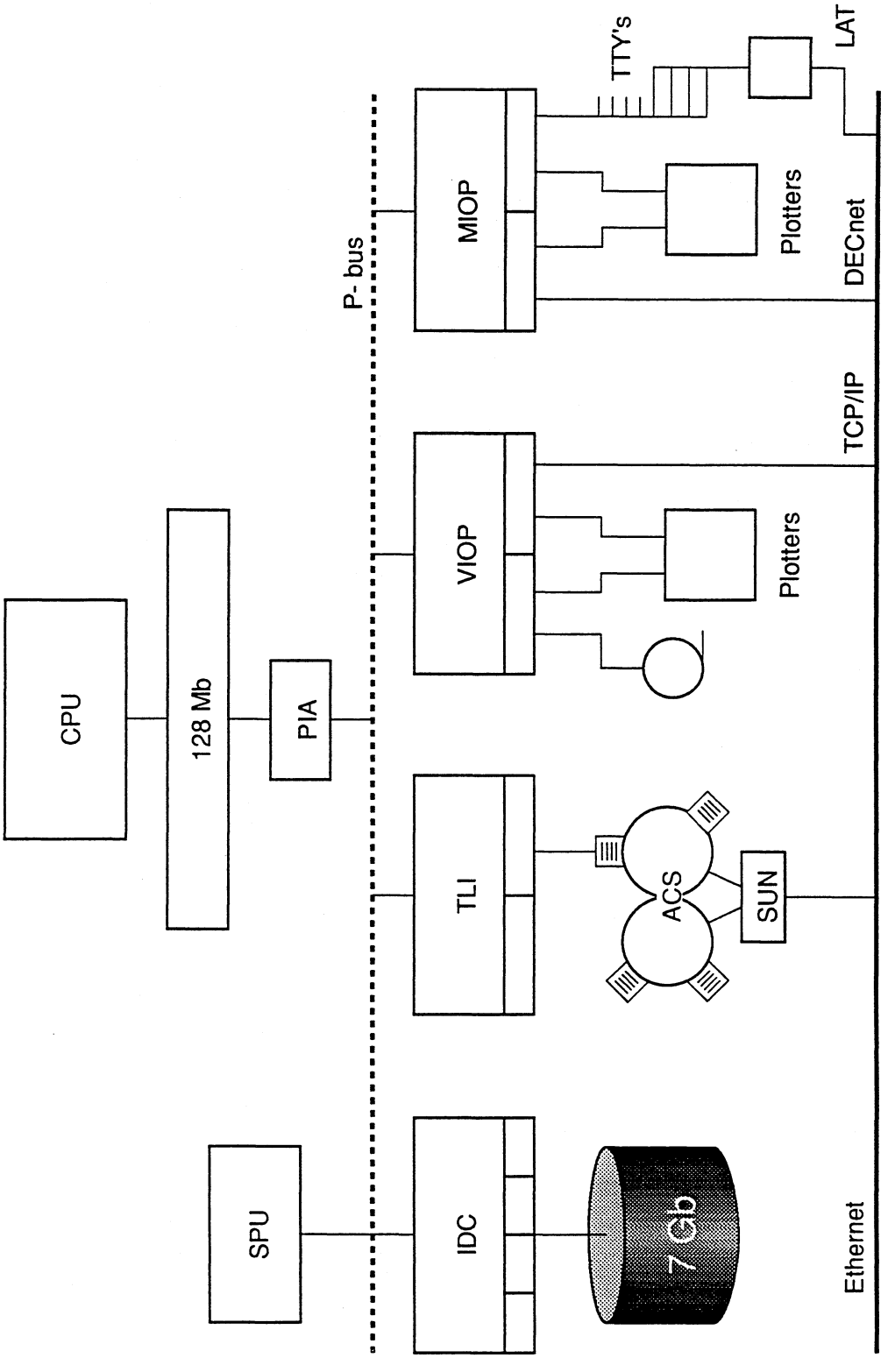

CPU
128 Mb
PIA
P- bus
SPU
IDC
7 Gb
TLI
ACS
SUN
VIOP
Plotters
MIOP
TTY's
Plotters
LAT
DECnet
TCP/IP
Ethernet

FIG. 3

CONVEX C210 AT KSEPL

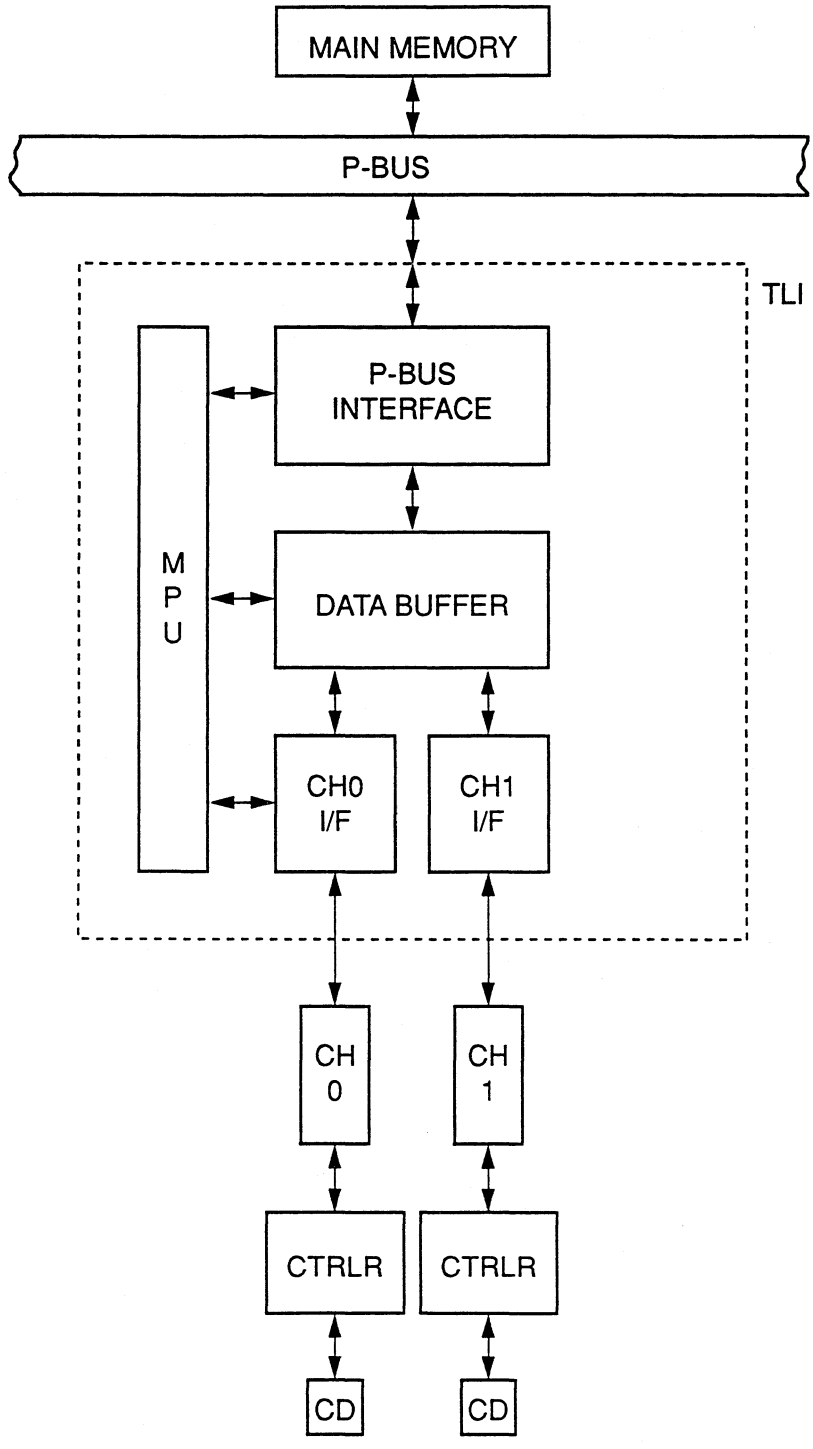
The Convex C210 was installed at KSEPL in August 1989.

After the successful connection of the Cray to the Automated Cartridge System in May 1990, a project was started to connect the Convex to the ACS as well. The main objective was to provide the possibility of periods of unattended production seismic processing. An upgrade was made to the disc capacity and in collaboration with Convex the system and operational support were developed to a level comparable with that available on the Cray and VAXcluster. Figure 3 shows the Convex configuration in use at KSEPL (October 1991).

The C210 system with one processor and 128 Megabyte real memory under control of Convex OS 9.0 has access to 7 IDC controlled discs, 2 tobot-operated cartridge drives, 2 manual tape drives for round tapes, a color plotter and a black and white plotter. Access to the Convex is established via direct terminal lines or via the Local Area Network using DECnet, TCP/IP or reverse LAT protocols. Operators use a terminal on the screen of which multiple windows display different types of information about the current processing status of the system.

In combination with the Cray, the Convex is used for seismic data processing in batch mode only and it is the choice of the user whether a job is submitted to the Cray or Convex.

Although seismic data processing is the main task of the Convex, the system is also used for plotting and seismic program development (in limited interactive mode).



Tape Library Interface (TLI)

FIG. 4

CONVEX CONNECTION TO THE AUTOMATED CARTRIDGE SYSTEM (ACS)

The Convex connection to the ACS consists of two components, the ***data path*** called Tape Library Interface (TLI) and the ***control path*** which is integrated in tpdaemon.

The Convex supplied ***data path*** is called Tape Library Interface (TLI) and is a PBUS Channel Control Unit (CCU) that can connect any of the machines in the Convex C200 series of supercomputers to the Storage Tek 4480 Tape Cartridge Subsystem. The TLI uses two block-multiplexer channel interfaces (FIPS60) for this purpose. Each channel allows data to be transferred between the Convex machine and a Storage Tek device at rates of up to 4.5 megabytes per second.

The TLI communicates with the operating system kernel via the Message Based System (MBS) and interrupts. The channel protocol is implemented with a device driver in Main Processing Unit (MPU) software. Physically, a TLI consists of a PBUS interface to main memory, an onboard MPU, a high-bandwidth multiplexing data buffer and two channels (Fig. 4).

The primary function of the TLI is to transfer data between external storage devices and the main memory. The TLI controls nearly all aspects of data I/O including device selection, media positioning, rate and block size matching, buffer management, and address translation. All elements of the TLI are involved in data transfers to some degree.

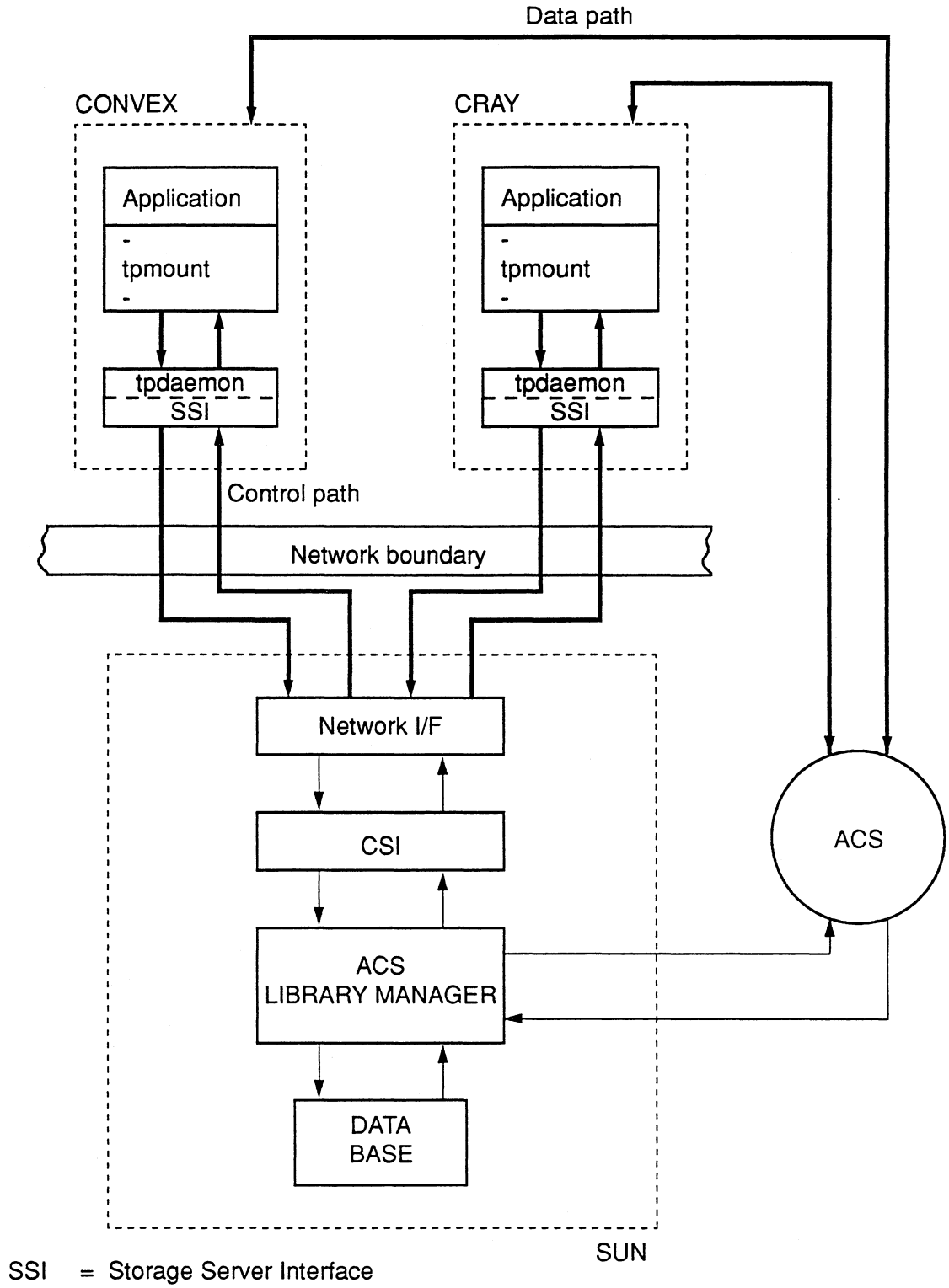
Control path software to communicate with the Storage Server software has been integrated with tpdaemon as a Storage Server Interface (SSI) and provides all the functionality needed to handle robot-controlled mounting and dismounting of cartridges. Figure 5 shows an overview of the ACS Storage Server software as well as the Storage Server Interfaces (SSI) on Cray and Convex.

Whenever a tpmount command is issued by an application running on the Convex, a communication link is set up with the storage server software. To start, a query command is sent to the storage server specifying the requested cartridge. The storage server software replies and informs about the current location of the cartridge and, if the cartridge is present in the ACS, it also provides a list of drives on which the requested cartridge can be mounted. The tpdaemon in turn selects a drive from this list and then issues a mount request to the storage server resulting in the robot mounting the requested cartridge in the specified drive.

In case a cartridge is not present in the ACS, tpdaemon issues a request to the operator to enter the cartridge into the ACS.

Each time reading or writing of a cartridge has been completed the Convex tpunmount command results in a dismount command that is sent to the storage server software which instructs the robot to remove the cartridge from the drive and return it to its original location.

The steps performed in mounting and dismounting are logged within the Convex. Figure 6 shows an extract of the "usr/adm/log/tapelog" file for a mount and dismount sequence for cartridge 441720.



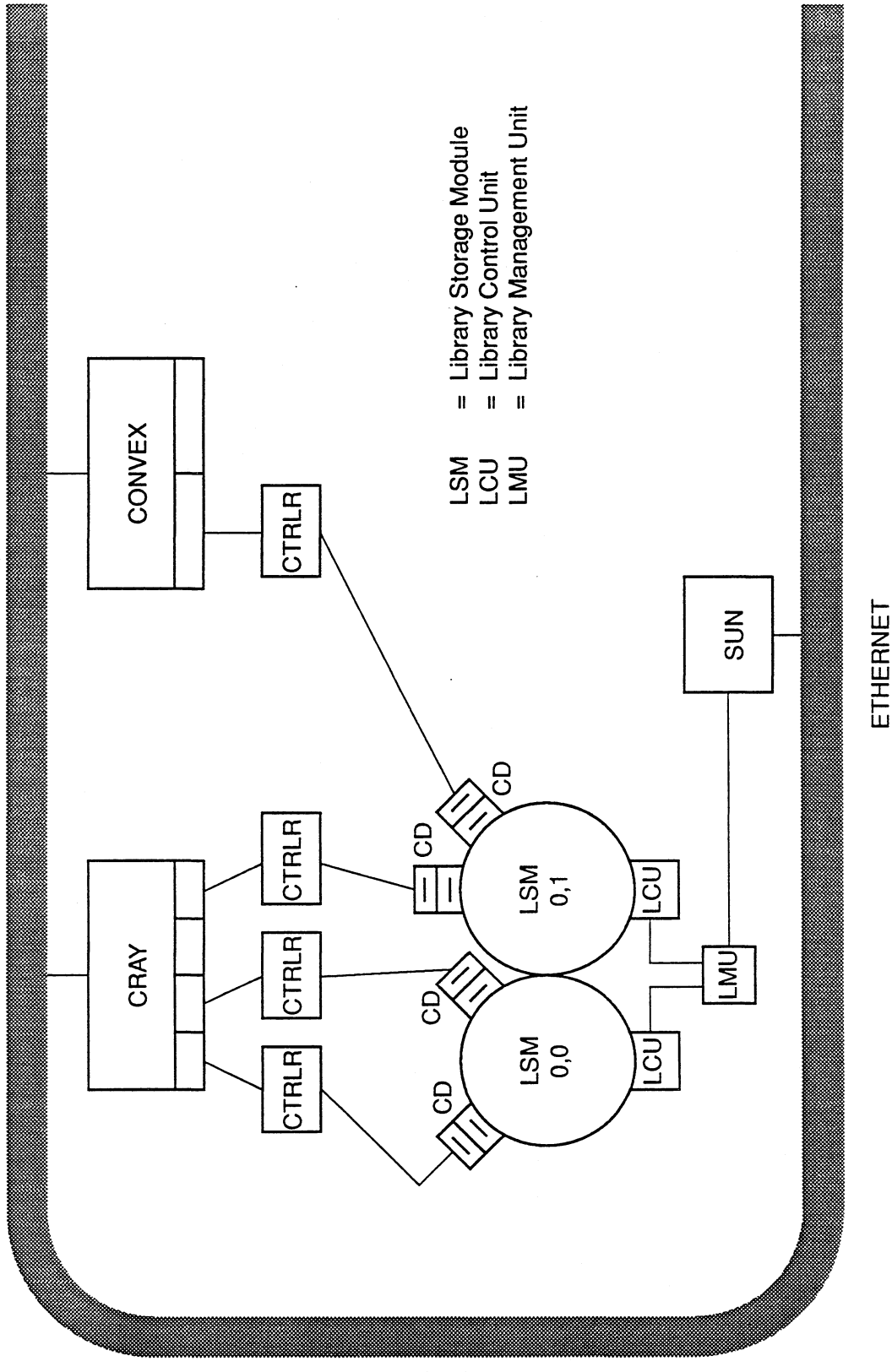
SSI = Storage Server Interface
CSI = Client Server Interface

ACS Software Overview

FIG. 5

```
Sep 25 23:34:33 ksecv1 tpdaemon: [DEBUG] robotQueryMountReq: siloQueryReq(kserb1, 1, M
OUNT, 441720)
Sep 25 23:34:35 ksecv1 tpdaemon: [DEBUG] received ACKNOWLEDGE response (Query) msgID=1
, silo request=9116
Sep 25 23:34:37 ksecv1 tpdaemon: [DEBUG] siloquery mount response received (441720)
Sep 25 23:34:37 ksecv1 tpdaemon: [DEBUG] isVolumeHome: status: Volume home
Sep 25 23:34:37 ksecv1 tpdaemon: [DEBUG] robotMountReq: siloMountReq: host=kserb1 tape
=441720 drive=0,1,10,0, msgID=2
Sep 25 23:34:39 ksecv1 tpdaemon: [DEBUG] received ACKNOWLEDGE response (Mount) msgID=2
, silo request=9117
Sep 25 23:35:00 ksecv1 tpdaemon: [DEBUG] silomount response received
Sep 25 23:35:00 ksecv1 tpdaemon: [DEBUG] Silo mount SUCCESS! (0,1,10,0, 441720)

Sep 25 23:38:34 ksecv1 tpdaemon: [DEBUG] op_unmount 441720
Sep 25 23:38:34 ksecv1 tpdaemon: [DEBUG] robotDismountReq: siloDismountReq: host=kserb
1 tape=441720 drive=0,1,10,0, msgID=7
Sep 25 23:38:36 ksecv1 tpdaemon: [DEBUG] startRobotReq: waiting for Silo Acknowledge
Sep 25 23:38:36 ksecv1 tpdaemon: [DEBUG] received ACKNOWLEDGE response (Dismount) msgI
D=7, silo request=9131
Sep 25 23:38:55 ksecv1 tpdaemon: [DEBUG] Silo dismount SUCCESS! (0,1,10,0, 441720)
```


FIG. 6



KSEPL ACS Configuration

FIG. 7

BETA TEST REPORT

KSEPL has been a beta test site for the Convex Tape Library Interface and associated control path software from October 1990 until May 1991. A prerelease of ConvexOS 9.0 was installed in September 1990 as a prerequisite for the TLI.

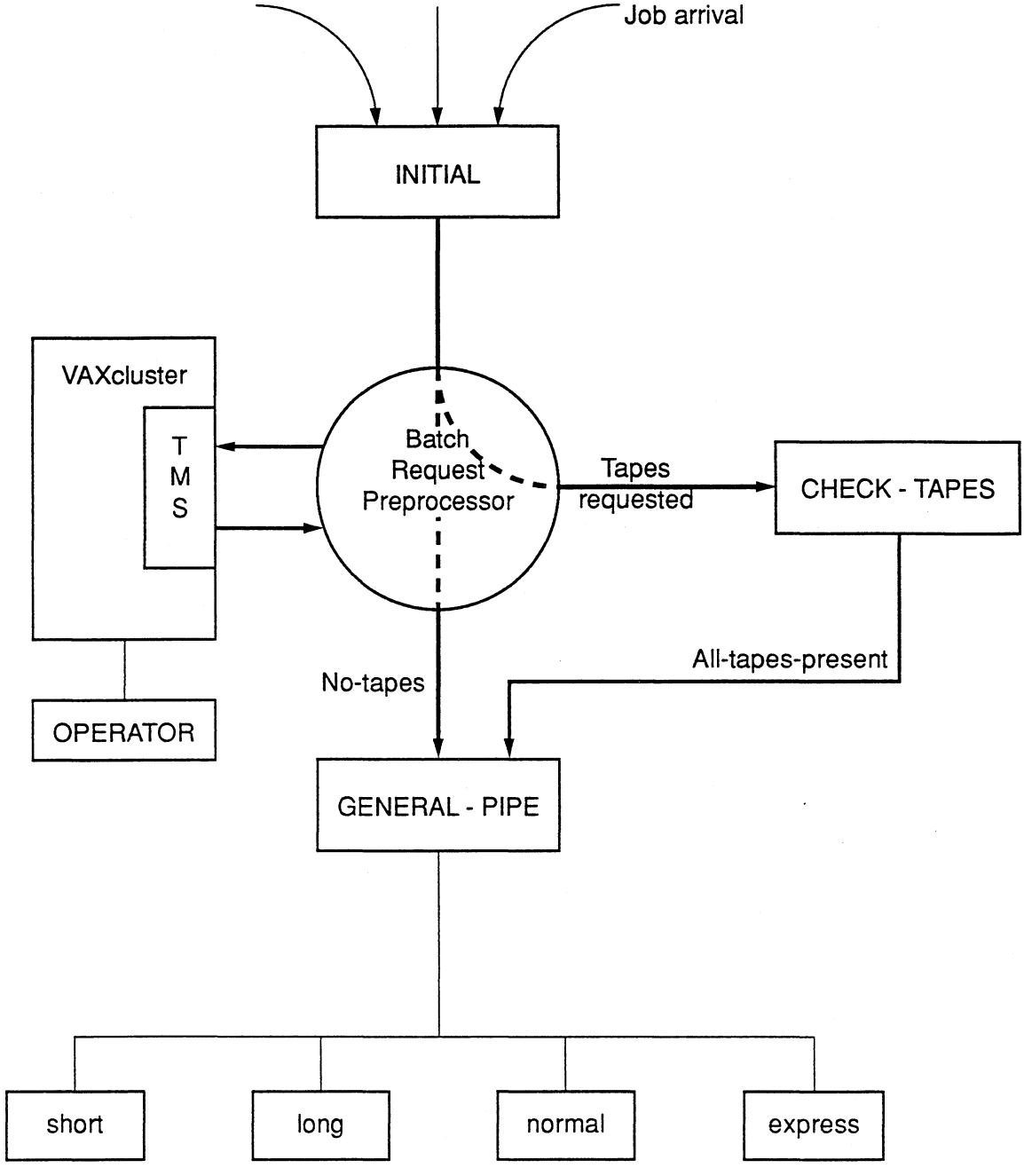
In October 1990 the TLI and a prototype of the control path software, essentially a large C-shell script using basic storage server commands (siloquery, silomount, silodismount) were installed.

The beta test period has been very beneficial for KSEPL. Almost from the beginning it has been possible to make full use of the Convex to ACS connection.

With respect to the TLI, only one serious problem was encountered during the test period. When reading a cartridge under certain conditions, the required number of bytes was transferred but some were put in the wrong order. This design problem was apparently recognised at another beta test site as well. Consequently, a new TLI board was available in Dallas and installed at KSEPL within a few days after the problem occurred.

No major problems were encountered when testing the control path software and in May 1991 the prototype software was fully integrated within tpdaemon.

During the beta test period we received prompt and adequate Convex support. In May 1991 integration of the Convex within the production processing environment was partly realised. Figure 7 shows the ACS configuration at KSEPL.



Job Flow

FIG. 8

UNATTENDED PROCESSING

One of the fundamental issues is how to make optimal use of an ACS once it has been implemented. Because very few sites can afford enough LSMs to hold all available cartridges, a procedure had to be devices to make sure that the cartridges being requested by queued and running jobs are loaded into the ACS.

Most tapes and cartridges used at KSEPL are destined for seismic data processing. Seismic data enter the computing centre as large volumes of round tapes or cartridges, typically 500 - 1500 tapes per seismic survey. During the processing of this data intermediate cartridges are created at several stages maintaining a one to one relationship for the first-stage cartridges. Once the processing is finished, a number of tapes and cartridges is kept in store for possibly re-processing at a later stage, e.g. when enhanced software becomes available or when new geophysical operations are implemented.

Only a fraction of the total number of round tapes and cartridges is available in the computer room and it was decided to develop procedures to ensure that batch jobs will be activated only when all cartridges that have to be used are present in the ACS and all round tapes that have to be used are on the computer floor. Implementation of these procedures takes place at different levels:

a. At the beginning of a batch job script, the user specifies the numbers of the cartridges and round tapes required for the execution of the job.

b. Software has been designed to analyse batch job scripts and to communicate with the KSEPL Tape Management System software (running on the VAXcluster) in verifying the locations of requested tapes and cartridges and to inform the operator to initiate tape or cartridge moves if necessary.

c. A procedure has been developed to hold a batch job until all tapes and cartridges are available and only then release the job.

Figure 8 shows a diagram of the installed mechanism.

Batch requests (or batch jobs) enter the system in the queue "initial" which is a stopped pipe queue.

A batch request preprocessor analyses all scripts entered in the queue "initial" to check whether any cartridges or round tapes are required.

If no cartridges or round tapes are requested, the preprocessor initiates a move to put the job in the "general-pipe" queue from where the job will be placed in its destination queue depending on group-id, CPU time requested and some other parameters.

If round tapes or cartridges are needed for a job, the preprocessor moves the job to the "check-tapes" queue, which is another stopped-pipe queue. A package containing information about tape and cartridge requirements of that job is built and transmitted to the Tape Management System (TMS) running on the VAXcluster.

The Tape Management System software verifies the current location of all requested tapes and cartridges. In case round tapes are not on the computer floor or cartridges are not present in the ACS, a print-out requests the operator to take appropriate action.

Once the Tape Management System has been informed that all requested actions are completed, appropriate jobs waiting in the "check-tapes" queue are moved to the "general-pipe" queue and from there to their final destination queues.

This mechanism performs an important function in ensuring that jobs become active only when all round tapes are present on the computer floor and all cartridges are loaded in the ACS. This has made it possible to create a backlog of jobs ready to be processed in an unattended mode of operation.

MONITORING TOOLS FOR BATCH PROCESSING

A Bourne shell script has been developed, which is activated every 15 minutes and performs a monitoring and control function within the attended and unattended modes of operations.

The basic ideas behind this monitoring script are to check the proper operation of the system (as much as is "definable" in scripts), report anomalies and take action in critical situations, e.g. if convuenet daemon has disappeared it tries to restart it.

Moreover, this script implements an autoscheduling mechanism that enables monitoring of the performance of the system over a five minute period and on the basis of the measured CPU utilisation and disc-space availability, increases queue limits (according to predefined rules) so that more batch requests may be started and idle time is avoided. Especially during unattended mode of operations this autoscheduling mechanism is of use to ensure an optimum utilisation of the system. All relevant information "seen" by this script as well as all actions taken are logged to enable subsequent analysis of what has or what should have happened. This monitoring script is still being expanded and further enhanced, establishing an ever more automated operational environment.

CONCLUSIONS

The integration of the Automated Cartridge System in the computing environment of KSEPL has expanded the scope of unattended seismic data processing considerably, enabling both Cray and Convex to operate in unattended mode during nights and weekends.

Operational requirements have changed: the number of staff is reduced and the type of work is moving towards a monitoring and controlling type of task. Effective 1 March 1991 the three-shift service (29 people) was reduced to a two-shift service (18 people).

More enhancements in the area of unattended processing, e.g. unattended plotting and further development of automated operation tools will make it possible to implement a one-shift service (12 people) in 1992.

The Virtual Volume Manager in ConvexOS V10.0

Jerry Schieffer

Convex Computer Corp

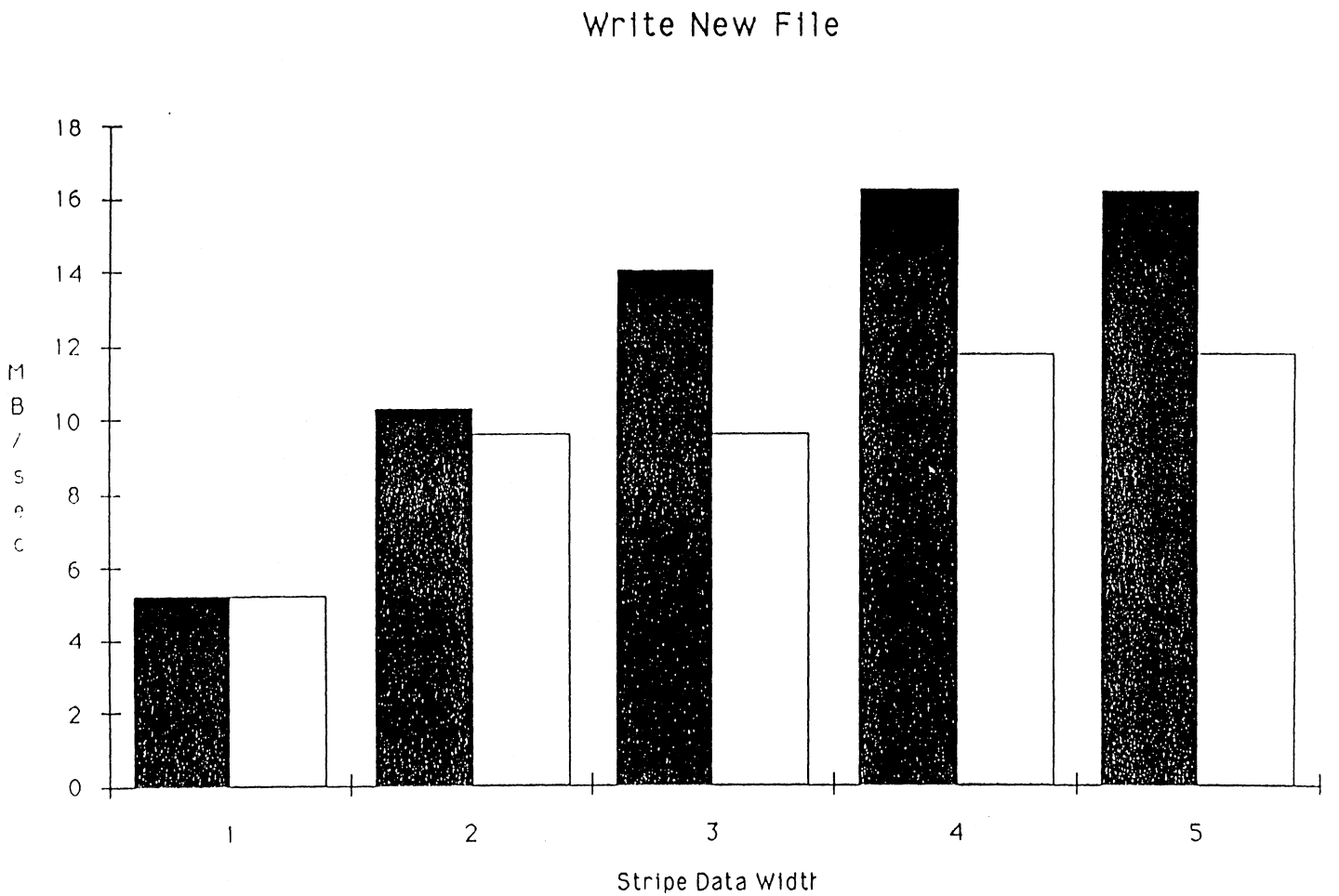
VVM Performance

Parameters used for Performance Experiment

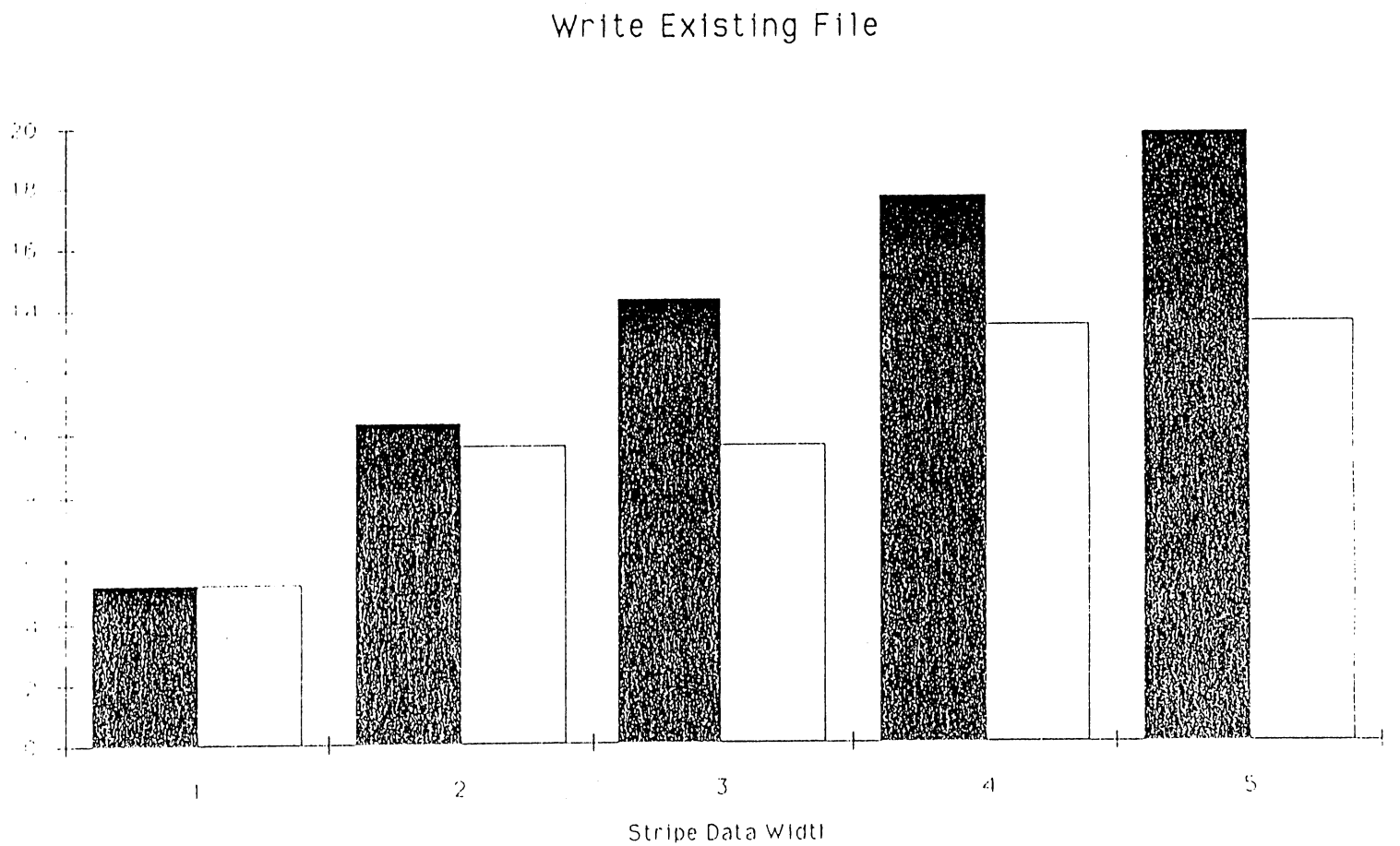
- Hardware configuration
  - √ Convex C220
  - √ DKD 502 disks (1GB capacity, 6 MB/sec nominal transfer rate)
  - √ Integrated Disk Channels

- Software configuration
  - √ ConvexOS V10.0 beta test version
  - √ File size is 128 MBytes
  - √ Filesystem blocksize is 64K Bytes

VVM Performance



VVM Performance

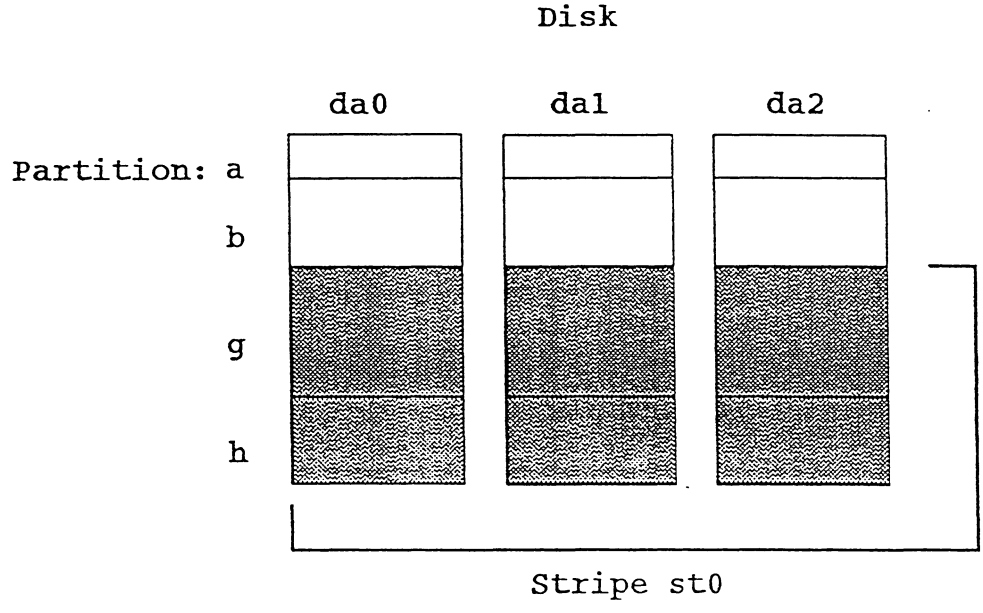


Virtual Volume Manager

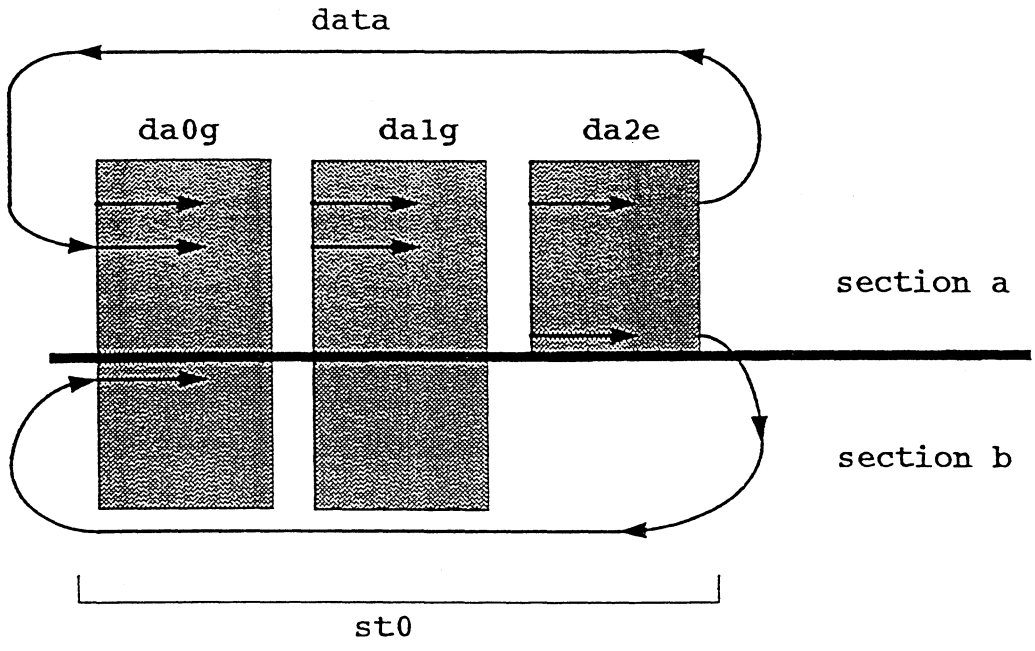
- Increases reliability of striped file systems through data redundancy
  - mirroring
  - parity
- File systems can be reconstructed while system is up
- If 'hot spare' disk space is available, data may be reconstructed automatically
- Several new utilities are useful with non-redundant stripes as well

Disk Striping

- Combining multiple physical disk partitions into one logical partition
- Should span multiple disks

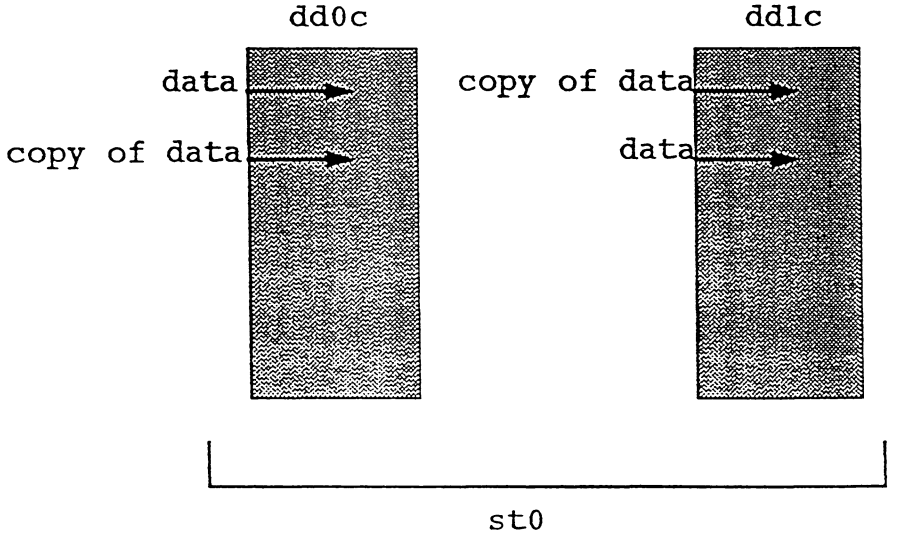


Stripe Sections



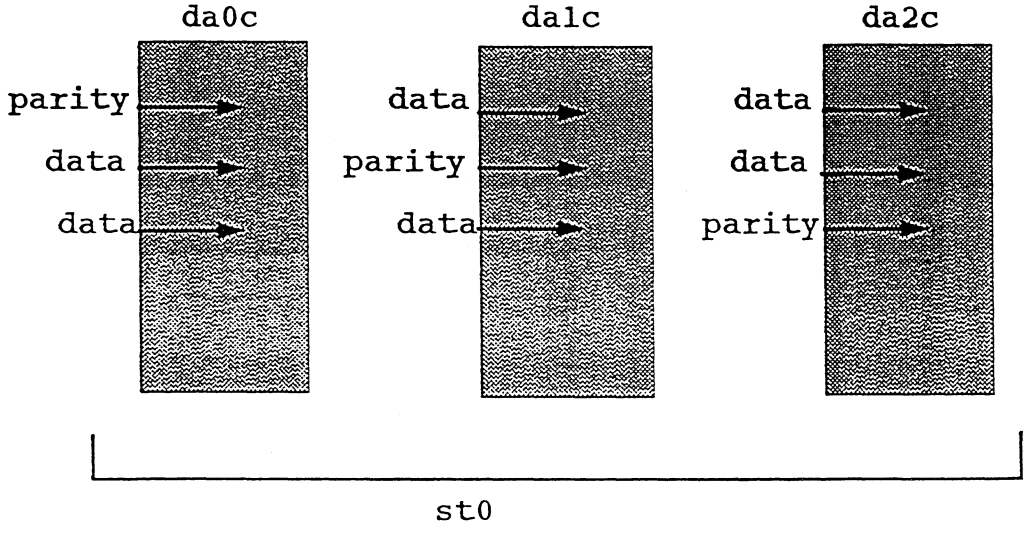
Mirroring

- Automatic for stripes with only two partitions
- Uses half the space in the stripe



Parity

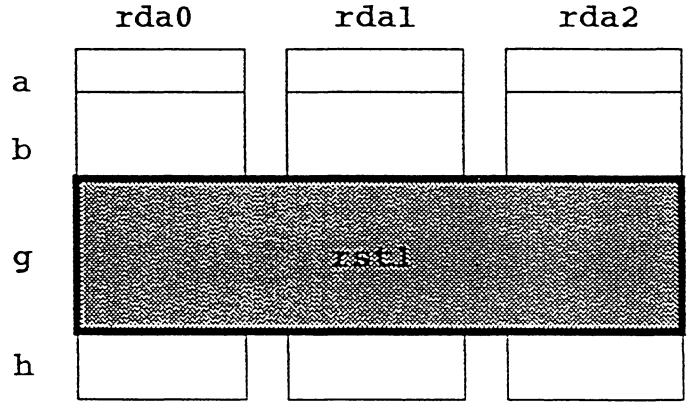
- Parity information is spread across all disks in stripe



Creating a Redundant Filesystem

- `-R` option to `newst`:

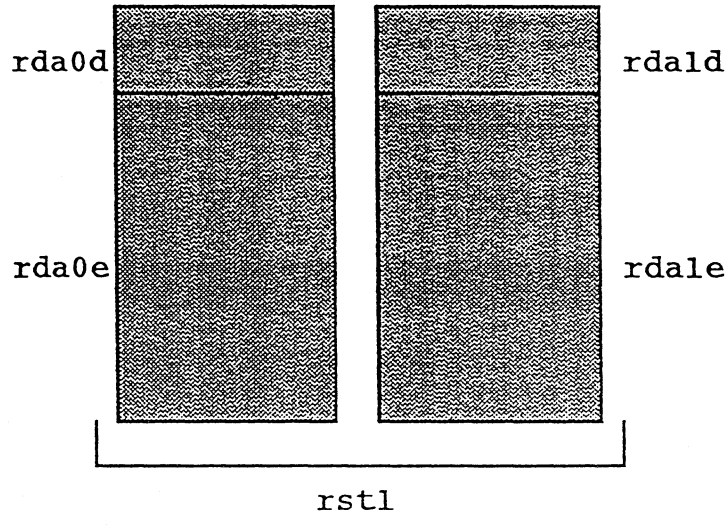
```
# newst -R /dev/rst1 /dev/rda0g dkd-001 /dev/rda1g \
  dkd-001 /dev/rda2g dkd-001
```



- This example will use parity
- `newst -R` will attempt to create stripe sections that are $2^n + 1$ wide

- Specify width of stripe section with `newst -RP#`

```
# newst -RP2  /dev/rst1 /dev/rda0d dkd-001 /dev/rda1d \
  dkd-001 /dev/rda0e dkd-001 /dev/rda1e
```



- `-P2` forces mirroring

The Hot Spare List

- List of partitions that can substitute for a stripe section on a failed disk
- Partitions are added to the hot spare list with **`newst -H`**:

```
newst -H [stripedev] diskdev type
```

- For example:

```
# newst -H du3c dkd-001
```

- ***`stripedev`*** argument is used to specify stripe devices for which this partition has an *affinity:*

```
# newst -H /dev/rst1 du3c dkd-001
```

Affinity

- A hot spare with an affinity for a particular stripe will have a higher priority to serve as a replacement for a failed disk in that stripe.
- Hot spares with affinities will not be used to replace other partitions.
- A hot spare with affinity should:
  - have sector size less than or equal to stripe device
  - equal or greater in size than the largest partition in the stripe
  - be of same disk type (preferable)

Disk Failure

- Console messages indicate disk failure. Reconstruction is attempted automatically if
  - a suitable hot spare is available
  - vvmdaemon is running
- Console messages indicate if reconstruction must be done by hand

Manual Reconstruction

- Use `qst` to determine affected stripes.
- Check for suitable hot spare using `getst -H`
- Or, find an available disk partition that is suitable and add it to the hot spare list using `newst -H`
- Reconstruct data using `mvst -H`
- Restart vvmdaemon if necessary

Reclaiming Hot Spares

- Moving data off a hot spare after reconstruction/replacement does not automatically reclaim space
- Use `rmst -H` and `newst -H` to clean a hot spare.
- Be careful! A single hot spare partition may support more than one failed stripe. Use `qst` to check.

Performance Issues

- Same rules apply for redundant and non-redundant stripes:
  - span controllers
  - MUST span 2 disks
  - avoid striping different disk types
  - NEVER stripe root or swap
- Mirroring uses half available disk space
- Creating large parity stripes may take several minutes and will generate vector context switches

EVALUATING A CONVEX AS A FILE SERVER

Dr Malcolm Read, NERC, Polaris House, Swindon, UK.

ABSTRACT

NERC spends many millions a year on collecting and buying data relevant to the environmental sciences. A number of data centres within NERC act as guardians for much of this data; the rest remaining the responsibility of individual users. To date this data has been stored on various mainframes and minis with consequent migration problems especialy when sites change hardware. A Corporate Digital Data Archive has been proposed to ensure the long term integrity of high quality scientific data.

This data archive fits into a wider model of distributed computing hich is described.

Two approaches to the archive were considered: (i) IBM MVS/DFSMS and (ii) UNITREE on a central UNIX machine. The IBM approach is safe but expensive. The UNITREE approach is untested but by utilising NERC's Convex would be fairly economical. Both these solutions provide software which migrates aging files and keeps track of archived datasets initiating automatic recalls. Various storage media were considered: principally 3480 cartridge and optical disk. 3480s are economical for archive storage and an established standard. Optical disk juke boxes offer much quicker access but are expensive and lack an industry standard.

The Natural Environment Research Council (NERC) is responsible for funding high level research in the environmental sciences through Universities and its own research institutes. Areas of research include geology, oceanography, hydrology and ecology and is carried out through component bodies such as the British Geological Survey (BGS) and British Antarctic Survey. Computing within NERC is managed centrally by the NERC Computing Service (NCS). NCS manages a Convex C210 at the BGS headquarters in Keyworth near Nottingham as well as a variety of IBM mainframes and VAX machines.

The emphasis on computer provision in NERC is changing from centrally provided minis and mainframes to a more distributed service based on PCs and UNIX workstations and servers. This requires the provision of a number of different server functions at sites based on a network infrastructure using TCP/IP and OSI. The servers are divided into two categories:

i) Executive servers which provide a function directly to end users (e.g. file and database servers).

ii) Support servers which usually provide a function to the executive servers (e.g. directory and authentication servers).

Figure 1 illustrates the main services provided to terminal, PC and workstation based users.

The executive servers identified perform the following functions:

1. Mail server. Coupled with a name and directory service this provides routing for all ingoing and outgoing electronic mail and other Office Automation facilities.

2. Print/Plot server offers the use of expensive peripherals to all users.

3. A file server holds large amounts of data to be shared by a body of users. It also provides an automated archive and backup facility.

4. A database server handles central DBMS.

5. A compute server is the traditional central computing service; it may be a parallel processor. Big models would typically run here.

The support servers ensure the integrity of the executive servers. They are:

1. A name server which returns the address of the required server (i.e. a table of which servers are available and where they are - not necessarily local).

2. A directory server performs a similar function but for users.

3. The authentication server handles security. This ensures a user is authorised to request a particular service. That service may itself impose a further layer of security.

4. A monitoring function is also required for LAN and distributed computing management and project accounting etc.

5. In addition an access gateway service is required to control internal and external routing to other LANs, JANET (the UK academic wide area X.25 network) and remote dial up users. This also performs file routing.

These are logical servers; they do not necessarily require separate processors for each server. There may be more than one server of the same type on a site. Ideally the use of a server is transparent to the user; he does not need to logon to each server machine.

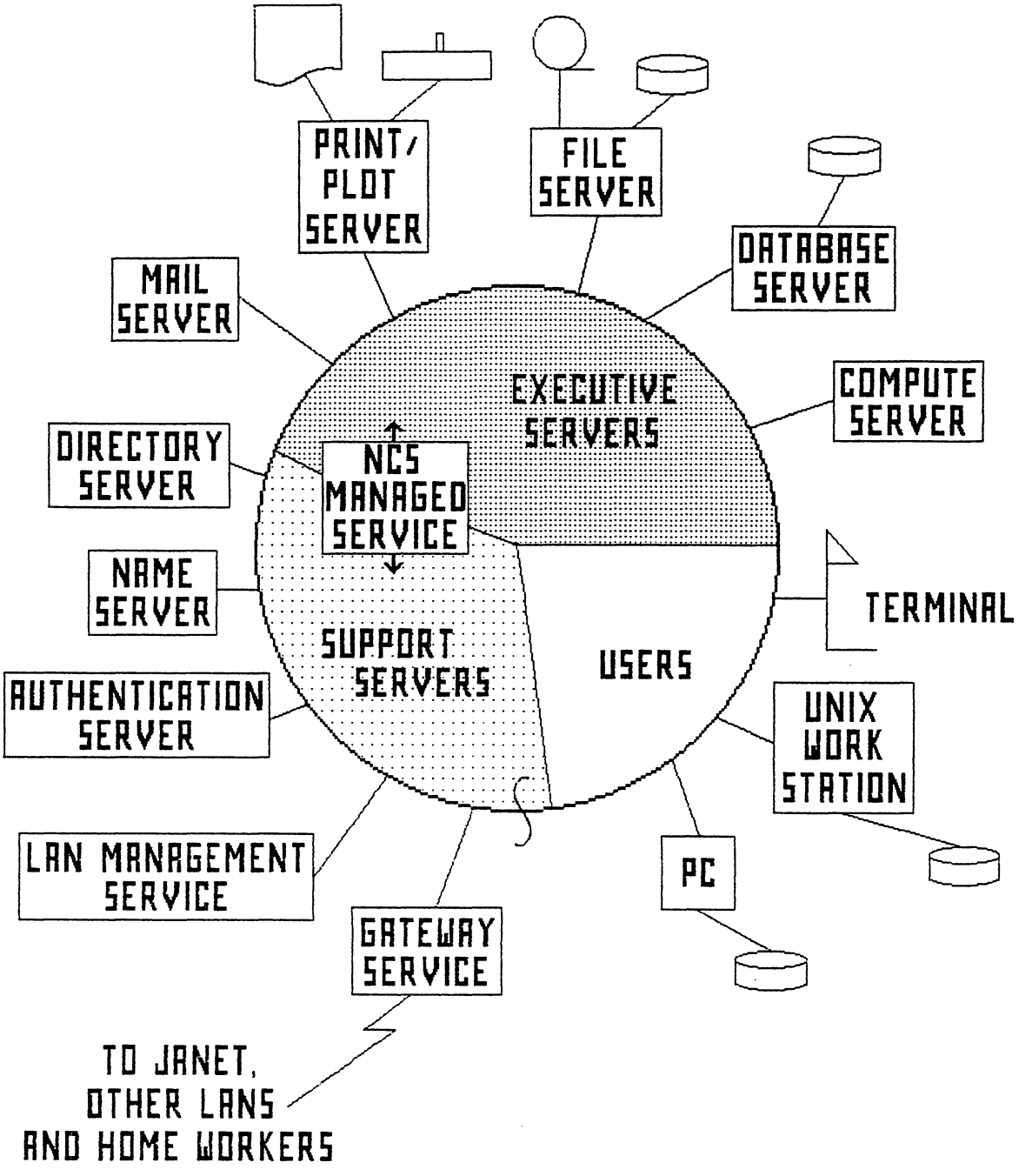


FIG 1

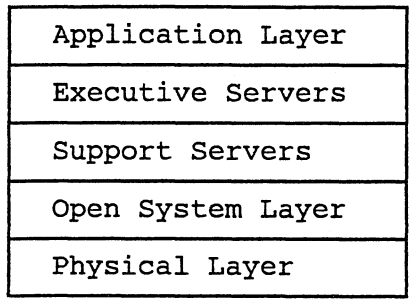
A DISTRIBUTED COMPUTING MODEL

Other services will probably be provided. E.g:

i) A time server to ensure time integrity across all servers at a site.

ii) An architecture independent application and system software servers to down load new versions of software.

It is possible to consider this model as a hierarchy of levels:



The bottom two layers of this "AESOP" model can be considered as corresponding to the lower six layers of the ISO/OSI seven layer model.

This paper will concentrate on the file, and particularly data archive, service and identify ways in which the NERC Convex 210 can form the basis of a distributed computing environment at Keyworth.

A file server manages file systems for users of a LAN. It enables a PC or workstation user to migrate files off his workstation to a centrally managed server; this makes them readily available to other users and ensures his files are backed up and kept secure. Ideally a user's files on a file server are available to him in a location transparent manner; i.e. it does not matter to the end user whether the data is on the workstation disk or on the file server. In practice it may be advisable to copy files from the central file server to the workstation before use for performance reasons. In any event it should not be necessary to logon to the server. It follows therefore that all data files on a LAN, if they are to be readily available to other users, need to be in the same format; the two standards to aim for are the IEEE standard for Binary Floating-Point Arithmetic (ANSI/IEEE 754-1985) and ASCII (8 or 7 bit). UNICODE (a 16 bit character code) may become an acceptable character code standard for the future.

NFS is a high level tool that offers a transparent file access service using lower level Remote Procedure Calls (RPC). The Andrew File System is an alternative to NFS. It is assumed that these are the tools that will be used in a distributed computing environment. NFS is widely available on all UNIX workstations, IBM PCs, Apple-Macs, VMS, VM and a number of other operating systems. AFS is less widely available however.

When a user (client) requests a file (or part of a file) from the file server AFS will always copy the entire file (although "paging" of large parts of a file is possible) to the workstation, whereas NFS copies the file (or part of a file) block by block. Under some circumstances AFS is to be preferred to NFS as it places less load on the LAN and the user can enjoy a more consistent response. Kerberos authentication is an integral part of AFS and AFS is therefore a more secure file system than NFS.

Facilities exist, such as UNITREE, which offer hierarchical storage management such that data migrates off the user's workstation to the file server and eventually to tape, or other backup medium, as it ages. This would provide the archive service.

The NERC Convex has been traditionally used as a compute server for running modelling software to users throughout NERC. We are now interested in exploring additional functions that it could perform at our Keyworth site. Options being considered are:

i) Database server. The alternatives here are finer grained parallel processors such as Sequent and Pyramid.

ii) File server. Juke box architecture optical disk servers such as Epoch offer cost effective alternatives where large amounts (i.e. 100 Gb +) of active data are needed "on line".

iii) Corporate data archive utilising the UNITREE software and robot media handling.

It is this last function that NERC is investigating in detail.

Figure 2 illustrates the features that a corporate archive machine must have. Data once validated can be sent to the archive over the network or on tape. This data could be in machine dependant form or in a standard data format. The conversion to a standard format (IEEE floating point and ASCII) can be performed either locally or on the archive machine. In the latter case information from the user about data type is required so if the data is not in standard format the user will need to logon to the archive machine to perform the conversion.

Once the data is received it needs to be catalogued in terms meaningful to NERC. This is the first level catalogue and will hold information about the data set designed to aid retrieval on the basis of such factors as geographical location, type of data (e.g rainfall, wave heights etc.), author etc. The catalogue also describes the internal structure of the data set. The first level catalogue can also provide an interface to more sophisticated retrieval systems such as Oracle and Geographical Information System applications.

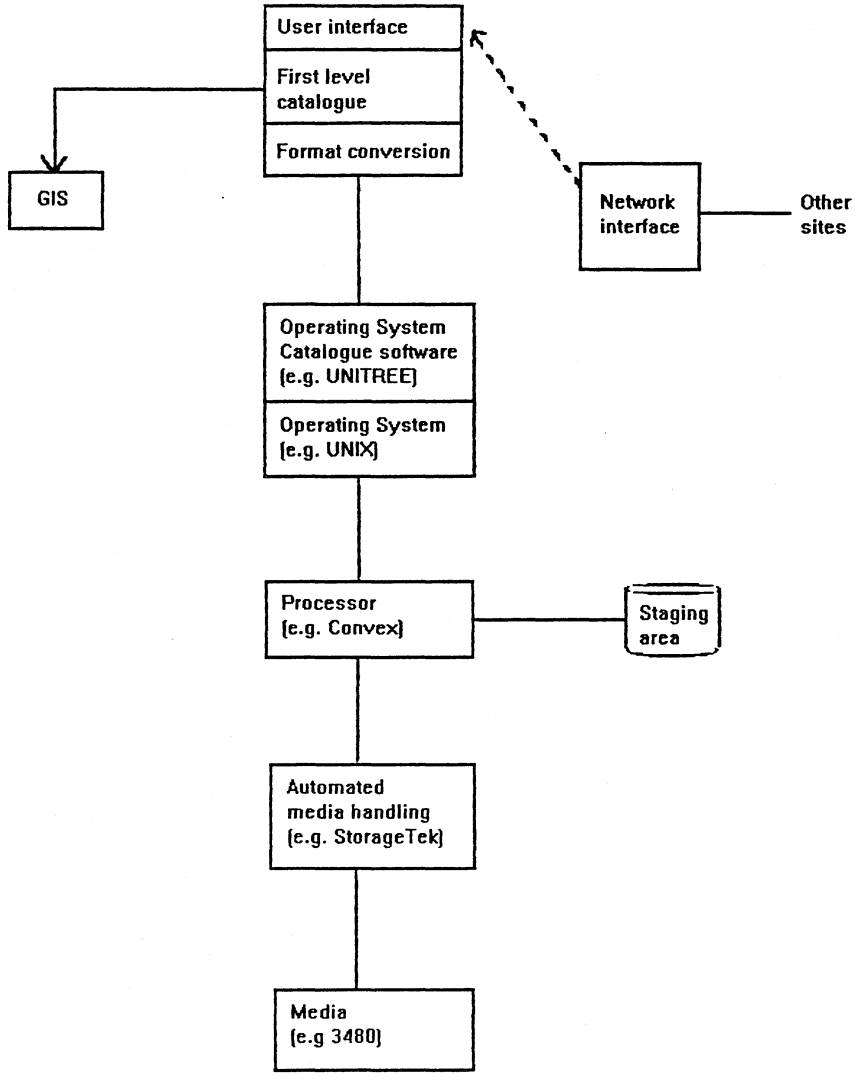

User interface
First level catalogue
Format conversion
GIS
Network interface
Other sites
Operating System Catalogue software (e.g. UNITREE)
Operating System (e.g. UNIX)
Processor (e.g. Convex)
Staging area
Automated media handling (e.g. StorageTek)
Media (e.g 3480)

Figure 2

The data file gets passed to the operating system file system which must have the facilities to migrate data, through a staging area, to some archive format. The file system must then be able to keep track of the location of the data and all data files must be available from a single point of interrogation (i.e. it is not acceptable to have to search for the data by interrogating the staging area and archive databases separately). It is also necessary that the chosen operating system (although not necessarily the hardware platform on which it runs) will enjoy a long, stable life. IBM's MVS is the safest option because of the huge investment industry has made in the system. UNIX can also be expected to provide a stable platform for many years and the UNITREE product offers the necessary data migration software. There are other proprietary options (Cray and Epoch) which can also be considered but the stability and long term life of the operating system and associated software is less certain.

The choice of suitable operating system and data migration software is critical. It is the one level of the model which could not be easily changed. The hardware platform, application and communications software and even the archive media (because it is automated) could all be changed without serious disruption to the service. An operating system (or associated migration software) change would entail a serious conversion exercise which is the kind of problem the archive is designed to overcome. It is important therefore to base the archive on a well established de facto, or international standard.

MVS is the standard operating system in the commercial world and has been around for over ten years (and in turn is only a development of a very similar operating system, MVT, with an even longer history). Virtually all large companies, throughout the world, base their administrative computing on MVS. The investment in MVS is enormous and it is very hard to envisage any changes to the operating system within the next two decades or so. It is a proprietary standard of long standing and therefore represents a very safe platform to base an archive on. The cost however is high.

An IEEE standard is being developed for data storage (the Mass Storage System Reference Model). This is still a draft standard and currently represents a model rather than a rigorous standard. The UNITREE software follows the model. In theory choosing one solution conforming to the model would enable NERC to migrate to another conformant platform; in practice the standard is insufficiently well developed to ensure this. It is important therefore that any solution based on this model is expected to be fully supported, over a long time period, by the supplier. UNITREE is available on a number of hardware platforms (Convex and Amdahl in particular) with many manufacturers, including IBM, planning implementations in the near future. Clearly many manufacturers see UNITREE as the way forward for mass storage in a UNIX environment.

It is expected that the archive will need to cope with about 1 Tbyte of data within the first few years. An economical media is needed with an acceptable access time. If operator overheads, and consequent scope for human error, is to be avoided some form of automation of the media is essential; this ensures access to all data within a reasonable time period (a few minutes) at all times.

A storage medium with a long and stable life is required. Optical disk suffers from a lack of any widely accepted international standard but could form the basis of an archive where very large amounts of data need to be stored; especially if floor space is limited. The IBM 3480 cartridge is an industry de facto standard that has been around for over five years and probably has another 10 to 15 years before the technology is superseded. Although capacity at 150 Mb in a 130 mm x 100 m x 20 mm cartridge is low (although the newer 3490E version offers up to four times the capacity) it is adequate for modest volume (1 Tb) archives. It can be automated by a robot cartridge handling system such as the StorageTek Nearline (up to 6000 cartridges per site).

NERC is therefore considering using its Convex at Keyworth (suitable upgraded) as the basis for a file server of "archival" data probably stored on 3480 cartridge in an automated silo. Processing and I/O demands are modest and well within the capabilities of a Convex processor without adversely impacting its traditional role as a compute server. The Convex is a suitable platform, with UNITREE, to act as a server of active files but the exact usefulness would depend on the data storage peripherals chosen.

The File Server Concept at the University of Tübingen

Dr. Dietmar Kaletta, Zantrum für Datenverarbeitung,
University of Tübingen, D-7400 Tübingen, Brunnenstraße 27, F. R. Germany

Mass storage is one of the challenges of this decade for data processing sites responsible for the central maintenance, processing and archiving of information. The term information is used in the narrow sense of computer science and means information which can be digitized in an appropriate manner. Requirements from the users of such computer centres using their own data terminal equipment define both a short-term and a long-term response to their activities. Their needs can be classified by three actions:

(1) mounting
(2) backup
(3) archiving

The demand for "unlimited" disk space is one of the oldest requirements of users as well as of system administrators and it yielded and yields a continuous increase of disk farms at the user's site as well as at the system's site year by year. The answer to that problem is NFS allowing the use of disk space at local and remote disks. The local part can be considered as the smaller one for temporary activities, whereas the larger remote part under the administration of the computer centre serves for permanent file systems.

But today, the disk volume, however, even of local disks is of the order of hundreds of Megabytes with increasing tendency. Thus the saving of these temporary data sets is more a business for professionals than for the sometimes unexperienced or less-interested-in user. The answer to that second problem is (a) to extend the standard backup procedures for the remote or central disks to the local or peripheral disks and (b) to supervise these activites by the regular operating staff of the computing centre.

The archiving of data is traditionally a standard task of each computing centre and should be done there for continuity-reasons. The expected data volume increase to be archived for a mid-size computing centre such as Tübingen is about 200-300 Gbytes per year for the next five years and requires a silo-technique. This estimation is only valid for the storage of data yielded by computation or word processing. The lifetime of these data conventionally is of about ten years. Libraries and archives normally need data storage over decades of years which provides additional requirements to the management and data carrier hardware of the silos mentioned above.

In order to realise the three classified tasks we have to provide two things substantially:

(1) to establish a high-speed local area network for providing fast file access and transfer
(2) to take care for a uniform data management software operating in a distributed environment consisting of different data terminal equipments at the user's site and of different hosts and servers at the computing centre.

The presentation will discuss several solutions which are offered by different computer companies and their applicability under the conditions of the University of Tübingen and its computing centre.

USING THE FAIR SHARE SCHEDULER

AT

MICHIGAN STATE UNIVERSITY

Charles Severance
301 Computer Center
Michigan State University
East Lansing, MI 48824 U.S.A

crs@convex.cl.msu.edu

11Oct91

Presented to European Users Meeting
Hamburg, Germany

OUTLINE

General Environment

User Base on Convex

Policies and Share Setup

What works well with Share

What could be better about Share

Tools we have written to cope with Share

Conclusion

Michigan State University Environment

Convex C240

256 MB Real Memory

100% Utilization for the last 2.5 years

Load Average Typically 5-7

200-400 MB Virtual Memory Typically

USER BASE

Non-Paying

- Physics (Astronomy, Condensed Matter)
- Computer Center

On Campus Pay

- Chemistry
- Mathematics
- Computer Science
- Engineering

Off Campus Pay

- State Government
- Otrher Universities

POLICIES

Non-pay users have lowest priority

Non-pay users cannot run >10 min. interactive

On campus pay users have unlimited interactive

Off campus pay users have unlimited interactive

SHARE SETUP

Non-pay interactive	75 *
Batch	150
On-Campus Pay	250
Off Campus Pay	250

* 10 Minute Maximum

CHARGE SETUP

```
/usr/convex/charge -D10,1000 -Y0.5 -X1.1
  -k3600s  -F06
```

LIMSHARE	On
Half Life	1 Hour
ADJGROUPS	Off

GOOD NEWS WITH SHARE

Can enforce policies without keeping track manually.

Eliminates user temptation for a user to run many processes to get extra CPU time.

Makes paying users think they have the whole machine even when it is 100% utilized.

ANNOYING SHORTCOMINGS

- What is the point of slxqt anyways?

- Undefined user shares become

  sgroup=root,shares=0

  Misconfigured people cannot even log in with negative priority

- sharecf is almost a useful tool. It needs to know about the group field in the passwd file - not just the groups defined in the /etc/group file

- Batch flaw with per-process CPU limit not per-job CPU limit

TOOLS WE HAVE WRITTEN TO COPE WITH SHORTCOMINGS IN SHARE

- msupreshare - extends sharecf to allow groups from passwd file instead of group file. Also allows group name of *default*. (C)

- Tune - enforces the 10 CPU minute limit for non-paying users interactive. (Perl)

Batch

- Balance - allows lots of jobs to be submitted to the queue by one user but jobs will be run round-robin. (C)

- qpolice - Will enforce job CPU limits in batch queues. (under development) (perl)

EXAMPLE MSUPRESHARE INPUT

```
 lname=root shares=1 notshared
  lname=online shares=100 notshared
   pwgrp=chem shares=1;
   lname=crs shares=1;
    ;
  lname=nopay share=10 notshared
   pwgrp=staff shares=1;
   pwgrp=physics shares=1;
   pwgrp=*DEFAULT* shares=1;
    ;
  ;
```

TO RUN

```
msupreshare < input > tmpfile
sharecf -f tmpfile
```

Typically run with cron.

CONCLUSION

- SHARE IS GREAT (For users and my boss)
- SHARE takes some getting used to for system managers.
- SHARE will not solve policy problems but
- SHARE can enforce the policy you come up with.
- Tools we have developed make life bearable.
- I started telling these problems to Convex within hours of the 8.0 Beta Test. I continue to tell anyone in Convex who will listen about these problems.
- I think than several problems might be addressed soon (not in 10.0 though)

1990–1991 European Users Committee report

1. On Friday 12 October 1990 the following members were elected in the European User committee '90–'91: Félix Sánchez (Es), Charles Curran and Jerry Hopkinson (UK), Dick Kaas (NL), Jürgen Kabelitz (D), Denis Mars (F), Andrea Mattasoglio (It), Isabella Weger (Aus). Jørgen Olsen was co–opted onto the Committee in July.

2. Early in 1990 the decision was made to organize the 1991 conference in Germany. The German User Group took the responsibility of choosing a place. They decided to have the conference in Hamburg (Forte hotel)

3. It was decided that the Committee should meet at the UK meeting (8/9 January 1991). The members available were Dick, Charles, Félix, Jürgen, and Jerry. Convex was represented by Lyla O'Driscoll (USA) and Ilana Ron (UK). Some decisions on the conference planning, committee structure and finances were made. It was decided that a treasurer was needed. The next face–to–face meeting was planned for the inaugural meeting of the Spanish User Group which was to take place at the end of May.

4. At the beginning of May the newsletter—ECUforum—was launched and sent to various (known) sites as well as being distributed electronically in PostScript form. The next edition is expected to be published in November. However, for it to be regular and worthwhile contributions are needed from you, please.

5. On 24/25 May the Spanish User Group held their inaugural meeting. The European Committee meeting, which was held to do the final planning for the Hamburg conference, was somewhat less of a success: only Charles, Dick and Félix showed up; despite telephone calls and faxes to Germany no–one was able to attend. Rossend Llurba, who was a speaker at the Spanish meeting, also attended. Because of the absence of anyone from the German Planning Committee, it was decided that the Planning meeting for ECUC'91 be held at Hamburg a month later and since Dick was available then that he should attend. It was decided to contact sites to get more papers for the conference.

   Other Committee business was dealt with: discussing what formal articles of association the group should have; a treasurer was chosen—Charles pulled the short straw (a match actually). It was also decided that it would be helpful for the Committee formal chairman to act as a focal point in discussions with Convex; Dick was proposed. Both these positions were ratified after consulting the rest of the Committee. Félix has also been acting as secretary since then. There was also a meeting with Jim Balthazar about the relation between the European user group and Convex.

6. The Hamburg planning meeting was held on 20 June. At this meeting were some German Convex people and Mary Kay Havens. All of the conference points were looked after and decisions were made on who should do what in the following weeks. The preliminary agenda was fixed and a planning for the demonstrations was filled in. There was also a proposal for a PERL tutorial.

7. In the weekend of 14 September Dick, Félix, Charles, and Jürgen got together in Hamburg. The financial planning (Convex contribution, conference cost, etc. and also the number of registrations) were discussed in great detail. We also made remarks on the first draft of the articles of association. We decided that a meeting between the committee and Convex (Jim, Mary Kay) should be planned just before the start of the conference. A meeting with the representatives of the User Group from the USA to discuss inter–group relations was also planned.

8. On Wednesday 9 October the committee had a meeting to discuss the financial reports, financial planning for next year, future conference planning and relation Europe and USA were discussed. Meetings were also held with representatives from Convex, and with the representatives of the User Group from the USA.

9. Throughout the year most of the Committee kept in regular contact by e–mail.

10. Following a suggestion at the UK January meeting and in ECUforum a Job Swap scheme was started.

Financial state:

The Rotterdam Conference made a profit of approximately 14500 NLG (7000US$).
The Hamburg conference should break even.

European CONVEX User Conference
October 9-11, 1991
Hamburg, Germany

CONFERENCE QUESTIONNAIRE
The answers

order to present you again a high level of quality and a broad selection of topics at the next User Conference, we need your assistance.
Would you please fill in the questionnaire below and return it to the conference desk. Please mark those answers that you find most appropriate.By some questions you can give any number in the range from 0 to 10. An overview will be given in the proceedings.

Number returns 29

A. CONVEX Presentations

How interesting were the topics for you ?
(0 = unacceptable,10 = very interesting)

Corporate Overview [ ]
2*1 1*3 1*4 3*5 3*6 6*7 7*8 2*9 4*10

European Overview [ ]
3*1 2*3 5*5 4*6 7*7 4*8 3*10

Product Overview [ ]
2*1 1*4 1+6 6*7 9*8 7*9 3*10

Convex Quality & Reliability [ ]
1*1 2*4 4*5 4*6 2*7 9*8 4*9 3*10

Realtime System [ ]
3*2 2*3 1*4 9*5 5*6 1*7 4*8 1*9 1*10

Comments
:.....
.....
.....

Please indicate which of the parallele sessions you attended:

System Management	[ ]	22
Visualization and Applications	[ ]	17
Compiler and programmdevelopment	[ ]	16
Large sites	[ ]	21

European CONVEX User Conference
October 9-11, 1991
Hamburg, Germany

How attractive were the topics on the parallel sessions for you ?
(1 = non attractive 10 =very attractive)

Security [ ]
1*2 1*4 2*5 2*6 6*7 4*8 5+10
ConvexOS/Secure: Unix security in perspective ? [ ]
1*2 2*3 1*4 3*6 3*7 5*8 2*9 4*10
Multiple stripe configuration with a Convex C240 [ ]
1*1 1*3 2*4 4*5 5*6 2*7 1*9 2*10
SysAdmin: A tool for Distributed Managment Environment [ ]
1*2 2*5 6*6 5*7 1*8 3*9 2*10
A help system for the ex/vi editor [ ]
4*1 1*3 1*4 1*5 4*6 2*7 2*8 1*9
Simulation Equipment For High Definition Television,
Acquisition and Visualization [ ]
1*1 3*2 2*3 1*4 5*5 1*6
FE-simulation and visualization in ground-water flow
and ground-water pollution using high performance systems [ ]
1*1 1*4 3*5 1*7 4*8 1*10
Visualization,X11R5, Pex [ ]
1*1 1*4 1*5 3*6 1*7 2*8 2*9 4*10
A Supercomputing Environment in Climate Research [ ]
2*4 2*5 2*6 4*7 2*8 3*10
Finite element of rubber parts in Hutchinson [ ]
3*1 1*2 2*4 2*5 1*6 1*7

Parallel Algorithms [ ]
1*1 1*4 2*5 2*6 1*7 3*8 2*9
Parallelization on Shared-Memory, Virtual Shared-Memory
and Local-Memory Machines -- A Comparison [ ]
1*1 1*4 2*5 2*6 3*7 2*8 2*9 1*10
Cxpa,Cxdb, Application Compiler [ ]
1*1 1*5 1*6 8*8 1*9 3*10
Convex Systems Managment [ ]
1*4 1*5 4*7 7*8 1*9 4*10
Convex C210 Robotic Cartridge Loader System and Unattended
Processing at KSEPL [ ]
1*1 3*6 6*7 5*8 1*9 1*10
Virtual Volume Manager (RAID product) [ ]
1*3 1*4 4*6 4*7 5*8 1*9 2*10
Evaluating a Convex as a file server [ ]
3*3 1*5 5*6 2*7 3*8 3*10
The File Server Concept at the University of Tuebingen. [ ]
1*2 1*3 2*4 1*5 4*6 3*7 2*8 1*9 2*10
FileServing with Unitree [ ]
1*2 2*4 5*5 4*6 5*7 4*8 1*9 5*10
EMASS from E-Systems [ ]
1*2 2*4 3*5 5*6 2*7 6*8 2*9 5*10
Technology Direction [ ]
1*4 4*5 3*6 1*7 10*8 5*9 3*10

European CONVEX User Conference
October 9-11, 1991
Hamburg, Germany

Would you contribute (next year) to one of the following sessions ?

System Managment 5 YES / 8 NO

Visualization 6 YES / 9 NO

Applications 6 YES / 8 NO

Compiler and Programdevelopment 5 YES / 8 NO
Large Sites 7 YES / 6 NO

Networking 6 YES / 8 NO

Which other session do you will see next year ?

.....

.....

Did you visit the demo's ? YES / NO
What is your opinion about the number of demo's
(0 = unacceptable, 10 = too much) [ ]
3*4 11*5 6*6 3*7 2*8 1*0
What is your opinion about the selection of supplies [ ]
(0 = unacceptable, 10 = very good)
2*4 4*5 5*6 5*7 5*8 1*10
Other topics to be involved in next year's conference :

1.....
2.....
3.....

Conference Location [ ]
(0 = unacceptable, 10 = very good)
1*1 3*2 2*3 1*4 3*5 7*6 3*7 3*8 1*9 4*10
Conference Room [ ]
(0 = unacceptable, 10 = very good)
2*4 2*5 5*6 6*7 9*8 2*9 3*10
Parallele Session Rooms [ ]
(0 = unacceptable, 10 = very good)
1*3 2*4 4*5 3*6 6*7 9*8 1*9 2*10
Demo Room [ ]
(0 = unacceptable, 10 = very good)
1*1 1*4 4*5 6*6 8*7 6*8 1*9 2*10
Help Desk [ ]
(0 = unacceptable, 10 = very good)
1*0 5*5 2*6 8*7 7*8 2*9 3*10

European CONVEX User Conference
October 9-11, 1991
Hamburg, Germany

Lunch [ ]
(0 = unacceptable, 10 = very good)
1*3 1*5 2*6 6*7 7*8 4*9 4*10
Hotel [ ]
(0 = unacceptable, 10 = very good)
1*5 2*6 3*7 11*8 2*9 4*10
Welcome buffet [ ]
(0 = unacceptable, 10 = very good)
1*3 2*4 2*6 8*7 5*8 2*9 5*10
Social evening [ ]
(0 = unacceptable, 10 = very good)
1*4 1*5 1*6 3*7 9*8 3*9 7*10
Number of days for the next conference [ ]
2 days 8 3 days 8 4 days 3 5 days 2 6 days 1
Conference price [ ]
(1 = to high,2 = acceptable,3= too low)
4*1 20*2 1*3
Parallel sessions 15 YES / 3 NO

Contribution by Convex [ ]
(0 = should be less ,10 = should be more)
1*0 1*3 2*4 10*5 3*6 4*7 2*8 2*10
Contribution by users [ ]
(0= should be less 10 = should be more)
10*5 5*6 2*7 5*8 2*10
What type of contributions do you will see in the next demo's

.....

.....

Any other comments :

Algeria

A. Agoudjil
Sonatrach
2 Rue Capitaine Azzoug
Algiers

T. Boliterbiat
Sonatrach-Exploration
2, Rue Capitaine Azzoug-H/Dey
Algiers

A. Mecheraoul
Sonatrach
2, Rue Cap. Azzoug
Algiers

Austria

Reinhard Brantner
Inst.f.Information Systems
Joanneum Research
Steyrergasse 17
8010 Graz

Michael Fink
EDV-Zentrum der Universität

Technikerstr. 13
6020 Innsbruck

Peter Maier
Control DATA
Barichg. 40-42

1030 Wien

Jaroslav Sadousky
EDV-Zentrum der TU Wien

Wiedner Hauptstraße 8-10
1040 Wien

Belgium

Jean-Pierre Malisse
UGMM
Gulledelle, 100
7700 Mouscron

Anthony McClure
E-Systems, Inc.
Chaussee Teh Hulpe 164
1170 Brussels

Denmark

Torben Moller Christensen
DOU Odense University

Niels Bohrs Alle 11
5230 Odense

Jorgen Olsen
DOU, Odense University
Niels Bohrs Alle 11

5230 Odense

Finland

Kirsti Lounamaa
CSC
P.O.Box 40
02101 Espoo

France

Christophe Auger
Total
Cedex 47

92069 Paris La Defense

Agnes Bai
CNET
PAA/TIM/SSI
38 rue du Leclerc
92131 Issy les M

Alain Crouzet
C.C.E.T.T.
4 rue du Clas Courtel
35512 Cesson-Sevigne

Jean-Sibashen Cruz
Aerospatiale
Centre des Gatines
91370 Verrienes Le Buisson

Philippe Destuynder
Convex S.A.
9, Avenue Ampere
78180 Montigny -le-Bretonneux

Christian Favre
Info'Rop
BP 164
31675 Labege Cedex

Pierre Herchuelz
Cerfacs
42, Ave. G. Coriolis

31057 Toulose Cedex

Jean-Yves Lachartre
Hutchinson S.A.
Centre de Recherche
P.O. Box 31
45120 Chalette Sur Loing

Gilles Leroy
Info'Rop Image
BP 164
31676 Labege Cedex

Jean-Marc Murello
Cisi Ingenierie
9 Rue Corneille
83000 Toulon

Farzin Parsai
L'Air Liquide-CRCD
BP 126
78350 Les Loges en Josas

Philippe Pedriau
Graphael
1/3 rue Stephenson
78182 Montigny Le Bretonneux

Sylvaine Roy
C.E.A. Grenoble
DSV/LIP/LCCP, C.E.N.G. BP 58 x

38041 Grenoble

Gerard Sabelete
CNET
PAA/TIM/SSI
38 rue du Leclerc
92131 Issy les M

Marc Till
L'Air Liquide-CRCD
BP 126
78350 Les Loges en Josas

Germany

Kent Angell
Titan Corp.
Lyoner Str. 14

6000 Frankfurt 71

Michaela Binder
Max-Planck-Institut
Heisenbergstr. 1
7000 Stuttgart 80

Dr. Peter Bolte
Convex Computer GmbH

Winterhuder Weg 29
2000 Hamburg 76

Maria Helena Bredehöft
Technische Universität
Hamburg-Harburg
Denickestr. 17
2000 Hamburg 90

Johannes Diemes
Hahn-Meitner-Institut Berlin

Glienicker Str. 100
1000 Berlin 39

Joachim Faulhaber
DMT-Institut für angewandte
Geophysik
Herner Str. 45
4630 Bochum

Tadeusz Frencel
Universität Tübingen
Brunnenstr. 27

7400 Tübingen

Helmut Biesenbach
Inst.f.angewandte Mathematik
Universität Bonn
Wegelerstr.6
5300 Bonn 1

Kurt Boehm
DFKZ
Im Neuenheimer Feld 280
6900 Heidelberg

Brandt
Universität Bonn

5300 Bonn

Manfred Buchholz
Uniras GmbH
Niederkasseler Lohweg 8

4000 Düsseldorf 11

Elisabeth Dregger-Cappel
RZ Heinrich-Heine-Universität

Universitätsstr. 1
4000 Düsseldorf 1

Hartmut Fichtel
Deutsches Kuma-RZ
Bundesstraße 55

2000 Hamburg 13

Wolfgang Friebel
Institut für Hochenergiephysik

Platanenallee 6
1615 Zeuthen

Heike Frisch
Universität Rostock
RZ
Albert-Einstein-Str. 21
O-2500 Rostock

Alfred Geiger
Universität Stuttgart
Computer-Center
Allmandring 30
7000 Stuttgart 80

Prof. Wolfgang Gentzsch
Genias Software GmbH

Röntgenstr. 13
8402 Neutraubling

Rolf Goos
Dornier GmbH
Abt. TMB
Postfach 1420
7990 Friedrichshafen

Peter Junglas
TU Hamburg-Harburg
Denickestr. 17
2100 Hamburg 90

Jürgen Kabelitz
TU Hamburg-Harburg
Denickestr. 17
2100 Hamburg 90

Dr. Dietmar Kaletta
Zentrum für Datenverarbeitung
Universität Tübingen
Brunnenstr. 27
7400 Tübingen

Karlheinz Kandler
Debis Systemhaus GmbH

Postfach 1340
7990 Friedrichshafen 1

Rainer Kleinrensing
Inst.f. angewandte Mathematik
Universität Bonn
Wegelerstr. 6
5300 Bonn 1

Dr. Rolf Knocke
TU Magdeburg
Postfach 4120

O-3024 Magdeburg

Axel Koch
Rechenzentrum
Universität Marburg
Hans-Meerwein-Str.
3550 Marburg

Simone Korzer
Convex Computer GmbH

Lyoner Str. 14
6000 Frankfurt 71

Dr. Manfred Kunicke
ZFK Rossendorf
Postfach 19
O-8051 Dresden

Dr. Cebel Kücükkacara
Universität Kiel
Olshaussenstr. 40-60
2300 Kiel 1

Dr. Harry Meier-Fritsch
Convex Computer GmbH

Lyoner Str. 14
6000 Frankfurt 71

Rainer Muchow
Universität der Bundeswehr HH
Rechenzentrum
Holstenhofweg 85
2000 Hamburg 70

Detlev Müller
BGR
Stilleweg 2
3000 Hannover 51

Klaus Nigemeier
Uniras GmbH
Niederkasseler Lohweg 8
4000 Düsseldorf 11

Christa Radloff
Universität Rostock
RZ
Albert-Einstein-Str. 21
0-2500 Rostock

Dr.
Telefunken Systemtechnik

Schnackenburgallee 114
2000 Hamburg 54

Jürgen Schmidt
Max-Plank-Institut für
Radioastronomie
Auf dem Hügel 69
5300 Bonn 1

Wolfgang Schneefeld
UltraNet
Max-Volmer Str. 1

4010 Hilden

Manfred Schoessler
TU Hamburg-Harburg
Denickestr. 17
2100 Hamburg 90

Roland Sieger
Diehl GmbH
Fischbachstr. 16
8505 Röthenbach

Heinz-Joachim Staerke
Max-Planck-Institut
Heisenbergstr. 1

7000 Stuttgart 80

Manfred Stolle
Freie Universität Berlin

Fabeckstr. 32
1000 Berlin 33

Fatima Streit
Institut für Hochenergiephysik

Platanenallee 6
1615 Zeuthen

Patricia Taggert
Convex Computer GmbH
Winterhuder Weg 29

2000 Hamburg 76

Paul Walter
Convex Computer GmbH

Winterhuder Weg 29
2000 Hamburg 76

Peter Wick
Rechenzentrum der Universität

Olshausenstraße 40
2300 Kiel

Ann Wildt
Convex Computer GmbH

Winterhuder Weg 29
2000 Hamburg 76

Italy

Ugo Belliani
Astrophysical Observatory

Viale A. Doria 6
95125 Catania

Luca Centili
University of Modena
Via Campi 213/B

41100 Modena

Carmelo Lo Presti
Astrophysical Observatory

Viale A. Doria 6
95125 Catania

Pietro Massimino
Astrophysical Observatory
Viale A. Doria 6

95125 Catania

Antonio di Paolo
CNR-Istituto Ricerche
Sozza Combustione
Piazza Le Tecchio, 40
80100 Neapel

Fabrizio Maguliani
Convex Computer S.P.A.

C. Colleoni-Orione 3
20041 Agrate B. ZA

Aldo De Micheli
Ansaldo Componeti Srl.

Via N Lorenzi 8
16152 Genova

Japan

Kokichi Hashimoto
NKK Corporation
Hieikudankita Bldg.
4-1-3- Kudankita, Chiyoda-Ku
102 Tokyo

Noriyuki Kitamura
NKK Corporation
Hieikudankita Bld.
4-1-3 Kudankita, Chiyoda-Ku
102 Tokyo

Netherlands

Jep de Ble
Convex Computer B.V.

P.O. Box 3267
3502 GG Utrecht

Rob de Bruin
Computercenter University
of Groningen
Postbus 800
9700 AV Groningen

Hinderikus R. Eekhof
University of Twente
P.O. box 217
7500 AE Enschede

Adrian v. Bloois
ACCU
Budapestlaan 6

3584 CD Utrecht

W. Drinkwahrd
Erasmus University
Postfach 1738

3000 DR Rotterdam

Robbert van Etten
FEL-TNO
Oude Waaldorperweg 63
2597 AK Den Haag

Dick Kaas
ACCU
Budapestlaan 6

3584 CD Utrecht

Jan von Kats
Convex Computer BV

Europlaan 514
3526 KS Utrecht

Rossend Llurba
Delft University of Technology

P.O. Box 354
2000 AJ Delft

H.A.M. Luiijf
FEL-TNO
Postbus 96864

2509 JG Den Haag

Marc Petit
Computercenter
Universität Groningen
Postbus 800
9700 AV Groningen

H.M.M Raven
DSM Research
P.O.Box 10

6160 MD Geleen

Martin van Roon
Pink Elephant
P.O. Box 106
2270 AC Voorburg

Arie Unlandt
KNMI
P.O.Box 201
3730 AE De Bilt

Simon Verdouw
Koninklyke Shell Exploratie
en Productie Laboratorium
Volmerlaang
2288 GD Ryswyk

Jan Wagemaker
ECN
P.O. Box 1

1755 ZG Petten

Norway

Sigurd Ruud
Geoteam
P.O. Box 52
0311 Oslo 3

Poland

Dr. Marek Ksiezyk
Academic Computer Service
Cyfronet-Krakow
Reymonta 4a
30059 Krakow

Jacek Moscinski
Inst. Computer Science

Mickiewicza 30
Krakow

Prof.Dr. Marian Noga
Academic Computer Centre
Cyfronet-Krakow
Reymonta 4a
Krakow

Janusz Starzyk
Academic Computer Centre
Cyfronet-Krakow
Reymonta 4a
30059 Krakow

Portugal

Heitor Pina
FCCN
Av. do Brasil 101
1799 Lisboa

Spain

Marina Buitrago
TARFIA S/N

41012 Sevilla

Maria L. De La Cruz
Consejo de Seguridad Nuclear

c/Justo Dorado 11
28040 Madrid

Felix Diez Sacrisyan
Convex Spain
C/ Velazques, 94

28006 Madrid

Sebastian Espinaa Cerrejon
Puplic Work Administration
PCM
C/ Vallehermoso 78.I
28015 Madrid

Felipe Navarro
Lagein
Cuesta de Olabeaga 16
48015 Bilbao

Felix Sanchez
CICA
Tarfia Sin
41012 Sevilla

Maite Sierra
CICA
Reina Mercedes, S/N
41012 Sevilla

Miguel Angel Vences Benito
Convex Spain
C/ Velazquez, 94
28006 Madrid

Sweden

Björn Brunnberg
Saar-Scania AB
Dep. BLTMD

15187 Södertälje

Lars Winberg
Convex Computer

Klarabergsviadukten 70
11164 Stockholm

Switzerland

Sreekumaran Padiyath
Paul Scherrer Institute
Computing Division
5232 Villigen

USA

Jim Balthazar
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Tom Christiansen
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Paul Gregory
Convex Computer Corp.

3000 Waterview Parkway
Richardson Texas, 75080

Mary Kay Havens
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Dave Holt
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Dale Lancaster
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Frank Marshall
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Michael Padrick
University of North Carolina

CB 3460, 308 Wilson Library
Chapel Hill, North Carol.27514

Bob Paluck
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Charles Riehm
E-Systems
Garland Division
P.O. box 660023
Dallas, Texas 75266-0023

Jerry Schieffer
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Charles Severance
Michigan State University

301 Computer Center, MSU
East Lansing, Michigan 48824

Phil Struve
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

Steve Wallach
Convex Computer Corporation

3000 Waterview Parkway
Richardson, Texas 75080

United Kingdom

Kevin Ashley
ULCC
20, Guilford Street
London WC1N 1DZ

Peter George Bowler
NNC
Booth Hall, Cherford Road
Knutsford WA16 (QZ

Charles Curran
Oxford University
Computing Services
13 Banbury Road
Oxford OX2 6NN

David Brian Hawkes
NNC Ltd
Booth Hall, Chelford Road

Knutsford WA16 8Q2

M.J. Hewitt
Atomic Weapons Establishment

Aldermaston, Reading RG7 4PR

Jerry Hopkinson
Science & Engineering
Research Council
Daresbury Laboratory
Warrington WA4 4AD

Malcolm Keech
ULCC
University of London
20 Guilford Street
London WCIN IDZ

Stuart Merrylees
Simon Horizon Ltd.

Azalea Drive
Swanley Kent BR8 85R

Steve Nutt
Convex Computer Ltd.
Randalls Research
Randalls Way
Leatherhead KT 22 TT5

John Parish
Convex Computer Ltd.
Randalls Research
Randalls Way
Leatherhead KT 22 TT5

Tschu Ralph
ULCC
20 Guilford Street
London WC1N 1DZ

Malcolm Read
NERC
Polaris House, North Star Av.
Swindom SN2 1EN

Philip Standing
Catalytic Information
Storage
P.O. Box 11
Alton, Nants GW34 5HF

Robert Vickers
ULCC
20 Gilford Street
London WC1n 1DZ

Yugoslavia

Mark Martinec
Jozef Stefan Insitute
Jamova 39
61000 Ljubljana

Matej Wedam
Jozef Stefan Institute
Jamova 39
61000 Ljubljana