LA-12930-C
Conference

UC-905
Issued: March 1995

High Performance Systems

Proceedings from the Conference on High Speed Computing

April 18–21, 1994

Compiled by
Manuel B. Vigil

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Table of Contents

HIGH PERFORMANCE SYSTEMS

PROCEEDINGS FROM THE CONFERENCE ON
HIGH SPEED COMPUTING
APRIL 18–21, 1994

Compiled by
Manuel B. Vigil

ASTRACT

This document provides a written compilation of the presentations and viewgraphs from the 1994 Conference on High Speed Computing given at the High Speed Computing Conference, "High Performance Systems," held at Gleneden Beach, Oregon, on April 18 through 21, 1994.

VIRTUAL ENTERPRISE 2000

Fred Kovac, Goodyear Tire & Rubber Co.

Somewhere in the year 2000, you walk into your activity center at home with your morning cup of coffee. You say good morning to VPS (formerly your TV and computer but now your Virtual Personal System). It recognizes your voice and is prepared to delight you with unique personalized knowledge.

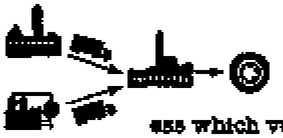
Your VPS says "Good morning. . .while you slept, we have been thinking about the issues we were kicking around yesterday. . .and we have some unique solutions." In your E-mail there is a priority message from Goodyear that your VPS rates as priority 2. Your curiosity aroused, you tell the VPS to display the Goodyear message.

A 3-D holographic image of a Goodyear tire on your car appears. The message states that this tire has been designed specifically to update your car. Goodyear offers to further customize the tires to your lifestyle. . .styling options can be created, such as sidewall design and color, as well as your desired ride and handling requirements. They know that your current tires have considerable miles remaining. . .but in today's world, you don't replace a product when it is worn out but rather when it becomes technologically obsolete. Your old tires will be re-manufactured to the latest technology and recycled.

The voice in the Goodyear message offers to have you experience the product. Putting on your driving gloves, you walk into your "Virtual Reality Area" for a test drive. Sensors in the Virtual Reality system map your particular driving habits to customize for you. As you continue your test drive, the tire/automobile system is modified on the computer to meet your needs.

As soon as you enter the order for your new tires, Goodyear's customer team sets in motion a computer search of data bases for current production and shipping schedules, raw material requirements, currency rates, delivered costs, etc. They simulate an actual, automated, lights-out production in a Virtual Reality environment and inform you of delivery dates anywhere in the world.

You select the best date and place to have the tires mounted on your car. In addition to your local Goodyear dealer, some of the more unique choices include where you work, your home, or in Gleneden Beach where you are currently planning a trip to a conference on "High Speed Computing for the Next Millennium."



Goodyear then activates the product realization process which verifies your test drive information, selects the design and materials for your customized tires, orders the raw materials, schedules the production and shipping, and notifies the local dealer when to schedule a "house call." This completes the logistics.

Before your tires are delivered, VPS reminds you of the appointment, allows you to make any last minute changes so as not to interfere with your golf game, and automatically charges your account after you experience customer delight.

VIRTUAL ENTERPRISE 2000

Corporations are facing a variety of increasingly difficult challenges as the world approaches the millennium:

- Customer/Consumer Satisfaction/Delight/Loyalty
- World Class Quality and Value
- Speed and Responsiveness (Short Cycle Times)
- Changing Demographics/Lifestyles
- Mass Customization (Differentiation)
- Shorter Product Life Cycles
- Innovative New Products
- Unpredictable Technology Changes/Dynamic Markets
- Think Global . . . Customize Local
- Global Rationalization of Investments (Economies of Scale, Scope)
- Intense Global Competition (Products/Knowledge)
- Governmental Regulations
- Environment Sustainability
- Employee Commitment
- Shareholder/Stakeholder Value

To meet these challenges successfully, a new agile business structure called

VIRTUAL ENTERPRISE 2000

based on information systems technology, is emerging (Fig 1).

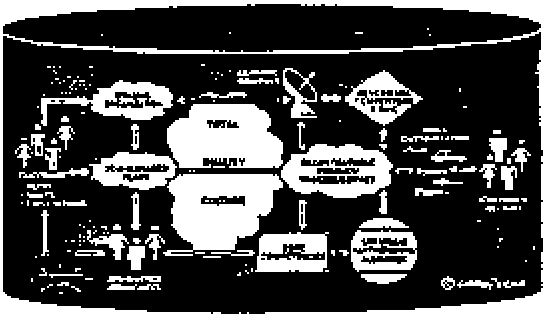


Fig 1 - Virtual Enterprise 2000

If an enterprise is to compete in the year 2000, its associates (employees) must be motivated by a vision. Associates without a vision resort to activity. A vision sets forth the kind of operation the company wants to be. It should be an agile "stretch," not an extension of the past. A vision for virtual enterprise 2000 is presented in Fig 2.

A RECONFIGURABLE ENTERPRISE THAT DELIGHTS CUSTOMERS WITH HIGH VALUE, INNOVATIVE SOLUTIONS (PRODUCTS/SERVICES) CUSTOMIZED TO THEIR NEEDS/WANTS/EXPECTATIONS ANYTIME, ANYWHERE IN THE WORLD

Fig 2 - Enterprise Vision

A vision is a leadership statement. It focuses on the future. It can shape the future.

While a vision points direction, a mission for Virtual Enterprise 2000 states purpose (Fig 3).

THE CREATION AND TRANSFORMATION OF KNOWLEDGE BY EMPOWERED ASSOCIATES INTO PRODUCTS, PROCESSES AND SERVICES THAT RESULT IN CUSTOMER SATISFACTION, SUSTAINABLE COMPETITIVE ADVANTAGE AND SHAREHOLDER/STAKEHOLDER VALUE

Fig 3 - Enterprise Mission

The vision and mission set the stage for the objectives of the enterprise. The example in Fig 4 depicts the objectives outlined by Stanley C. Gault for Goodyear.

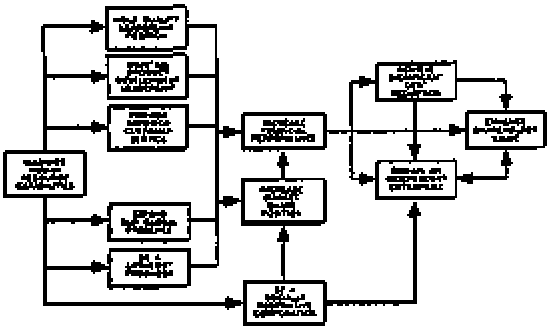


Fig 4 - Stan Gault Objectives

Strategies are then developed to achieve the objectives. These establish the company's strategic focus from which business planning and resource allocation can be formulated (Fig 5).

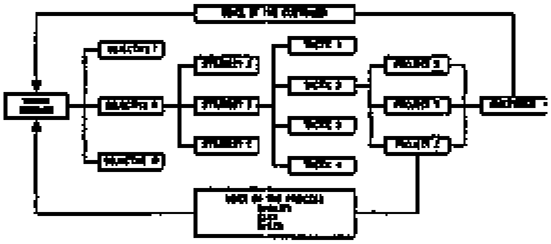


Fig 5 - Strategic Focus

Deployment is facilitated using techniques such as Hax, Hoshin, TQM, etc.

Virtual Enterprise 2000 is basically a reconfigurable, computer-networked, customer solutions delivery system.

Virtual Enterprise 2000 is made possible by sophisticated information systems technology. The emerging international information infrastructure (III) revolutionizes enterprise, education and entertainment (EEE). Information and data are transparent, seamless and easily accessible any time, any place. Systems accommodate digital, voice, text, and imaging as well as differences in languages, customs, currencies, etc. Information augments products (mechatronics and/or cybernetics). Information leveraging is vital to Virtual Enterprise 2000.

Information Age technology permits full spectrum delivery (Fig 6), linking information and enterprise goals.

- Client server - work stations on local area networks for distributed computing
- Compact disks - mini data repositories
- Mainframes - enterprise servers for massive consolidated data repositories (right sizing is business critical)
- Modeling and simulation - system visualizer
- Global network - universal access telecommunications system
- Open systems - plug and play
- Data acquisition at point of origin - warts and all
- Sensing and monitoring real-time metrics with optical scanners
- EDI with customers, suppliers - intercompany boundaries disappear
- Autonomous agent based systems - walk and chew gum
- Lights-out data processing functions - untouched by human hands
- E-mail communications - the great collaborator / equalizer (race, gender, rank are transparent)
- Infotainment - information can be fun
- Computer commuting - millions save on gas!
- System security - high to low

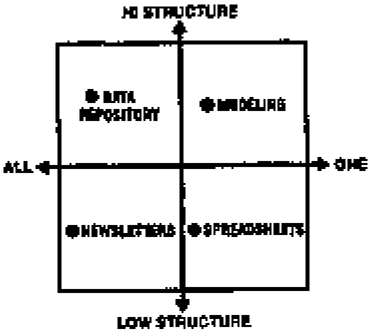


Fig 6 - Information Systems Infrastructure

For Virtual Enterprise 2000, reconfigurable, pervasive, information ergonomics expand the business horizon.

- Object data bases - modular information systems for rapid customization
- Groupware - everyone knows what everyone knows, anywhere
- Scalable power - when you need it, where you need it
- Home shopping and interactive media for database target marketing - time is money
- High bandwidth - volume and/or distance not a constraint
- Flat screen systems - hang a picture on a wall
- Laptops - credit card size
- Complex models - real-time 3-D constructs
- 'Smart' data - live intelligent system links
- Neural network - capture the genius of the human mind

- Voice interactive synthesis - talk to your system synergistically
- Secure satellite systems - global internet
- Virtual mobility - vehicles are information centers linking the asphalt highway and the information highway
- Personal digital assistants and personal communicators - the office is where you are
- Multimedia imaging - Hollywood comes to everyday business with the marriage of TV and the computer
- Fuzzy logic software - not everything is black or white
- Massively parallel processing - the body builder of information systems
- A.I. expert systems - everyone can be the best and the brightest
- Virtual Reality - immersion into what could be; experiencing the unknown
- Knobots - intelligent assistants

In Fig 1, the information flow is represented by the arrows or channels linking all activities. Through this framework, the life blood flows and the system is nourished.

Virtual Enterprise 2000 transparently transforms and integrates a continuing stream of data chaos into usable information images leading to knowledge for business decision-making.

CUSTOMER FOCUS

A Virtual Enterprise has customer - to - customer focus (Fig 1). That is, marketplace success starts with customer needs/wants/expectations and follows through to customer delight (Fig 7). Quality is in the eye of the customer. To understand customer needs, a "voice of the customer" process is required. That is, a customer information system that encompasses everything from point-of-sale input . . . to a telecommunications "hot line" . . . to on-site, in-the-field customer solutions . . . to global information highways. Technology push/market pull can be fully explored for share of market and growth. The objective is the right product for the right customer at the right time.

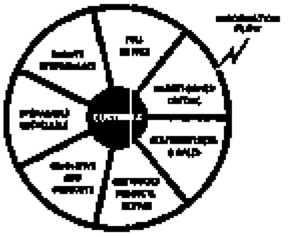


Fig 7 - Customer Delight

For anticipating customer needs and developing innovative new products/services, information systems permit:

- Market mapping for niche market identification (Fig 8) positions products to indicate areas of the marketplace that are without product or service coverage. This matrix facilitates a market-access strategy.

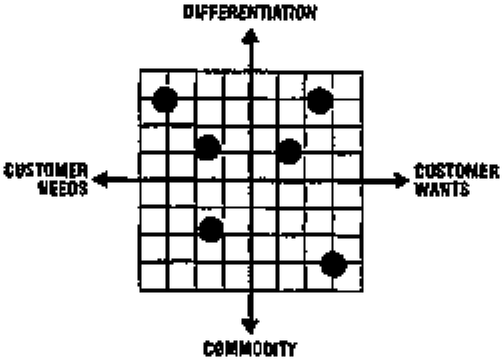


Fig 8 - Market Mapping

- Customer focus groups, panels and workshops utilize electronic QfD, or quality function deployment (Fig 9). This database marketing technique establishes relationships between customer requirements and business decisions, e.g., technical capabilities.

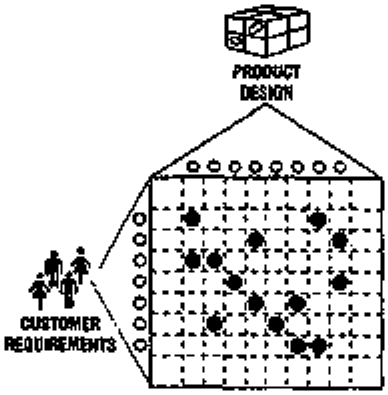


Fig 9 - QfD

- Techno-business research for technological/marketplace forecasting (Fig 10) links emerging technology with emerging wants and ensures that future technology satisfies latent expectations.

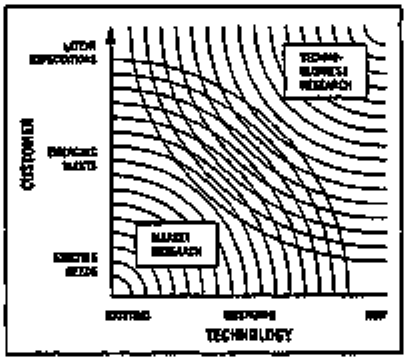


Fig 10 - Techno-Business Research

- Virtual Reality in antenna shops generates breakthrough products/services (Fig 11). Customers do not really know what they want until they experience it. Creating the experience can create the need.

Fig 11 - Virtual Reality

- Analytical Heuristic Protocol (AHP) synthesizes the information obtained from Market Mapping, QfD, Techno-Business Research and Virtual Reality. Combined with associates' expertise, AHP (Fig 12) creates distinctive forms of knowledge that predict future trends, clarify common causes, assess external environments, screen ideas and prioritize tactics to meet strategic objectives.

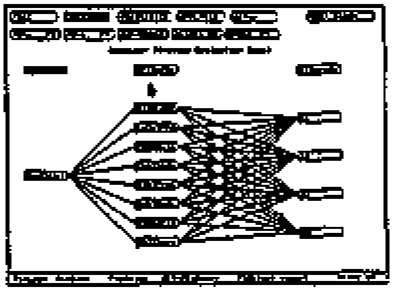


Fig 12 - Analytical Heuristic Protocol

Leveraging the marketplace is a strength of Virtual Enterprise 2000.

DYNAMIC ORGANIZATION

Virtual Enterprise 2000 consists of a flat, flexible organizational structure (Fig 1). The history of organizational structures has been evolutionary . . . from vertical to horizontal . . . with a gradual reduction of levels in the hierarchy resulting in an increased span of control. The span of control has moved over time from a low of one-on-one reporting to 7, then 35 and towards 100. This span will continue to increase. Now, managers are leaders (listen . . . learn . . . and lead) for doing the right things, and associates are empowered to do things right (Fig 13).

DOING THE RIGHT THING
DOING THINGS RIGHT

Fig 13 - Lateral Hierarchy

The organization is everyone. Hi-tech managing provides an environment for associates to perform at their best for maximum organizational excellence.

Just as form follows function, structure follows strategy. Elimination of a non-value-added hierarchy is made possible through advanced information systems, e.g., groupware (everyone knows what everyone knows).

The result is a sense of urgency and quick responsiveness to turn every challenge into an opportunity and supply rapid solutions to customers. Speed is a competitive advantage and directly relates to information systems technology.

LEARNING COMPANY

Corporate strategy for competitive advantage has expanded from the concepts of raw material access and assets utilization to knowledge power. Today, learning is the main source of sustainable competitive advantage. Once the enterprise moves from a rigid hierarchy to a flexible organizational structure, it gains the potential to become a learning company (Fig 1). Information systems technology makes it achievable. The information highway enables associates to gain internal and external perspectives and state-of-the-science knowledge. No longer do communications originate from one's immediate manager but from anyone, anywhere globally. Information-rich systems permit everyone to build infrastructure, identify options, add expertise from other industries, and gain outside viewpoints. Everyone is a global networker (gatekeeper). Unlimited inputs and synergistic collaborations define the learning company (Fig 14).

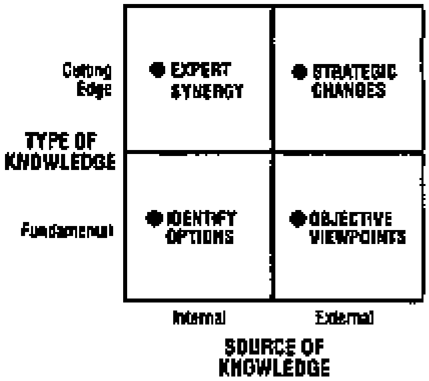


Fig 14 - A Learning Company

Challenges such as these are now tackled routinely:

- Identifying critical success factors, value-added activities and/or cost drivers anywhere in the value chain
- Cluster sharing across portfolios
- Accessing consultants' services
- Conducting technology assessments, especially environmental

NIH (not invented here) is greatly minimized or eliminated through strategic use of information systems.

A learning company facilitates the acquisition of knowledge and continuously transforms itself. A learning company builds data repositories containing technical, marketing, manufacturing and financial information. A learning company captures knowledge in expert systems. These intellectual assets can be as valuable as real properties (plants).

A learning company leverages knowledge and empowers associates. Some managerial functions in the Intelligent Enterprise will be performed by learning specialists or human resources facilitators.

Benchmarking is part of a learning company. Benchmarking broadly covers competitive assessment and best practices, any industry (Fig 1).

Benchmarking is essential for:

- Building sustainable advantage over competition
- Outsourcing for best-in-class
- Early warning system to prevent surprises
- Maintaining an external focus

Benchmarking identifies global competition including:

- Other producers
- Emerging technologies
- Substitute products

Benchmarking involves info-tech including:

- Searching on the information highway with knobots (industry/academia/government)
- Reverse engineering competitive products (by computer)
- Patent mapping to pinpoint future focus and predict future competitive products

Benchmarking should also include internal self-assessment, or enterprise analysis. This includes identifying strengths/weaknesses, core competencies and organizational health.

Key metrics measure productivity and progress, not activity. Metrics for benchmarking agility are grouped as follows:

- External (customer) - customer complaints, product performance, average product age, etc.
- Internal (associates) - sales per associate, profit per associate, degree of networking, percent of profit from new products, cash flow, concept-to-ROI, etc.

Sophisticated benchmarking techniques can include:

- Technological forecasting, e.g., normative
- Modeling of change
- Alternative scenarios of future competitive environments

These techniques can bring future competitive threats and strategic opportunities into focus and provide technology road maps.

SELF-MANAGED TEAMS

Self-managed, cross-functional teams are replacing rigid hierarchies in the Virtual Enterprise (Fig 1). As corporate structures "get horizontal," teams dissolve functional walls. Teams are business units . . . miniature learning companies . . . they have a vision, mission and objectives . . . they establish lateral integration, permitting functions to operate concurrently, not sequentially . . . they have responsibilities and accountabilities . . . they make decisions. They function by means of information systems. Their info-tech tools include:

- Instantaneous telecommunications systems
- Global teleconferencing
- Global data repositories
- Unrestricted information flow
- Statistically designed experiments
- Critical path flow charting
- Value gap analysis
- Performance predictions
- Simulations
- Modeling
- Interactive expert systems

Technology teams, for example, create, engineer, build and evaluate new products on the computer utilizing an integrated knowledge network.

Artificial intelligence permits computer diagnostic systems to effect technology transfer and resolve challenges (problems) anywhere in the process with a sense of urgency. Input symptoms and output solutions (prioritized options) with feedback are illustrated in Fig 15.

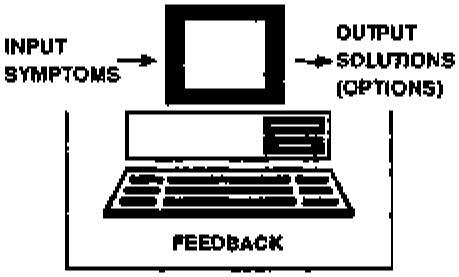


Fig 15 - Diagnostic Systems

Team formation is a science. For example, using Ned Herrmann's left brain/right brain techniques, diversity and team success can be built-in (Fig 16).

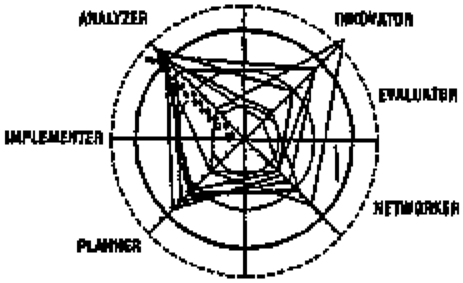


Fig 16 - A "Whole Brain" Team

Teams should be structured for individual excellence and be composed of innovators, evaluators, networkers, planners, implementers, analyzers, etc., in addition to being multidisciplinary and intercultural. This can be called TEAMAGILITY.

To maximize synergy, team building is necessary (Fig 17).

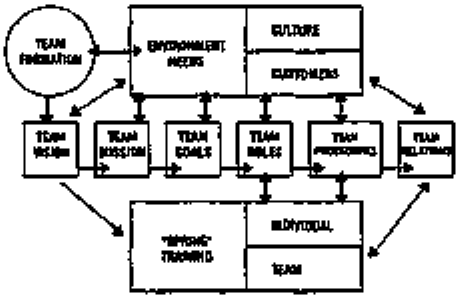


Fig 17 - Team Building Model

Team-based organizations interact more with customers to gain insight into their needs/wants/expectations. In addition to cross-functional teams, the Virtual Enterprise thrives on cross-corporate teams with members from customers and/or suppliers and/or networked companies. LANs and WANs facilitate team operations.

EMPOWERED ASSOCIATES

Self-managed teams with participatory decision-making empower associates (Fig 1). Information flow empowers associates. Empowered associates also result from a flatter hierarchy.

Associates' commitment is strengthened by:

- Trust (everyone wants to do a good job)
- Flatter hierarchy composed of leader-coaches who motivate and energize

Empowerment is the opportunity for associates to decide what needs to be done and discipline themselves to do it. Humanetics catalyzes this process. Humanetics is a synergy of knowledge-associates and computer cognitive skills.

TOTAL QUALITY CULTURE

The philosophy, beliefs, behavior, and shared values of an organization are its corporate culture. The Virtual Enterprise (Fig 1) has a total quality culture (TQC). Whatever the terminology adopted, it is a measure of the organization's health. TQC involves all associates at all facilities worldwide. TQC reflects the quality of work life. TQC means moving from problem-oriented to vision-led . . . from making and selling products to finding and satisfying needs . . . a learning company . . . from solving problems to developing innovative systems to prevent problems . . . from checking quality to building-in quality . . . from a hierarchy to networking . . . from sequential to concurrent . . . from functional silos to self-managed teams . . . from control to empowerment . . . from internally-focused to customer-focused. TQC means re-engineering around processes, not functions . . . flowcharting the job to eliminate non-value-added steps . . . rethinking the job to identify opportunities for improvement . . . and feed forward/feedback from customers.

- Atmosphere of diversity for maximizing associate synergy and innovation
- A creative environment with job assignments that are truly challenging and enriching (high expectations)
- Continuing education and training for continuous personal and organizational improvement (multimedia learning)
- Broad-band job descriptions (if any)
- Clear objectives (linked to corporate and business unit objectives) with performance appraisal (major responsibilities, performance standards)
- Hi-tech tools with high bandwidth
- Budget responsibilities
- Participatory decision-making through groupware
- Opportunity for entrepreneurship (e.g., discretionary funding monies)
- Reward and recognition, including celebrations
- Career-path information and "lateral promotions"
- Succession planning criteria
- Work-and-family programs (flexible work hours, job sharing, work-at-home, work-at-customers, etc.)
- Continuous communications and information flow (fully knowledgeable on the corporate vision, objectives, strategies)

TQC means boundaryless information flow. TQC means diversity for synergy. TQC means trust between associates.

CORE COMPETENCIES

Virtual Enterprise 2000 is built upon its core competencies (Fig 1). Core competencies are the collective learning in the organization. Core competencies enable a company to remain an independent enterprise. Core competencies are what a company does very well . . . preferably better than the competition . . . and possibly better than anyone else in the world.

In addition to TIRES, Goodyear has ten other core competencies. Just three, as examples:

- GLOBAL STRATEGY (includes Think Global . . . Customize Local)
- Domination of HI-TECH RACING (also builds "sense of urgency")
- SYSTEMS MARKETING to sophisticated hi-tech customers (e.g., vehicle manufacturers)

In areas where a company lacks core competencies, it often outsources or collaborates for expertise. The ability to do this is a core competency for a Virtual Enterprise. These activities create strategic partnerships and alliances.

STRATEGIC PARTNERSHIPS & ALLIANCES

Strategic partnerships and alliances are basic to the Virtual Enterprise (Fig 1). These operate on several levels.

- Strategic partnerships are established with customers and suppliers as a basis for long-range planning in a win/win combination. This is often essential for customer loyalty.
- Alliances and/or joint ventures are often formed to penetrate a new geographic market or to enter a new field where additional core competencies are required. These collaborations often involve leveraging technology or core competencies on a global basis. Goodyear joint ventures, for example, include South Pacific Tyres (Australia) and South Asia Tires Limited (India).
- Knowledge relationships with academia and/or governmental entities expand technology and business capabilities.

These collaborations demand electronic partnering through information technology, such as groupware, EDI, etc.

RECONFIGURABLE PRODUCT/ PROCESS/SERVICE

Virtual Enterprise 2000 embodies an agile, reconfigurable product / process / service (Fig 1). Agile and reconfigurable imply the capability to continually restructure everything including resourcing with other companies to achieve customer satisfaction, sustainable competitive advantage and shareholder / stakeholder value in an environment of continuous change. Organizations can be reconfigurable on an annual, monthly, weekly, daily, even hourly basis. It is the ability to thrive on constant change. Open systems are essential in a reconfigurable environment.

Reconfigurable means maximizing quality, cost and speed; knowledge of the marketplace; creative environments; continuous search for new materials; rapid project selection; strategic resource allocation and robust reliability. Reconfigurable requires relentless cost improvement over the product life cycle while retaining quality imperatives. Long-term high-volume products are commodities. There is a continuous stream of innovative new products and flexible manufacturing processes. Global logistics enable a consumer order to instantaneously trigger corporate activity back to raw materials sourcing, low-cost differentiation, value pricing, partnerships with complementary companies, etc.

A reconfigurable Virtual Enterprise is a solution delivery system for customers.

A reconfigurable Virtual Enterprise can delight customers with high value, innovative solutions (products/services) customized to their applications (needs/wants/expectations) anytime, anywhere in the world.

A Virtual Enterprise leverages change, and, benefits from chaos.

MASS CUSTOMIZATION

The result is mass customization . . . ultimate market segmentation... low cost product differentiation with potential deliverables of one (Fig 1).

This strategic agility is the purity of essence for customer delight and loyalty.

CUSTOMER DELIGHT

Customer delight then is real (or perceived) value exceeding expectations with zero customer complaints (Fig 1). Customer delight results when companies realize they are selling value-based solutions, not products (Fig 18).

TIRE CONSUMERS DON'T LIKE TO . . .	
▷ AIR UP TIRES	TUBELESS-HALO TIRES
▷ CHANGE TO WINTER TIRES	ALL-SEASON TIRES
▷ DRIVE IN THE RAIN	AQUATRED TIRES
▷ CHANGE FLAT TIRES	EXTENDED MOBILITY TIRES
	Deliver Solutions... *NOT Products*

Fig 18 - Sell Solutions

VIRTUAL ENTERPRISE 2000

The basic components of Virtual Enterprise 2000:

- Vision/Mission/Objectives
- Customer Focus
- Total Quality Culture
- Agile, Reconfigurable Systems
- Flexible, Flat Organization
- Diverse, Empowered Associates
- Self-Managed Teams
- Learning Company
- Benchmarking
- Core Competencies
- Strategic Partnerships and Alliances
- Mass Customization
- Global Strategy

These components are intertwined. They are unified, energized, and 'wired' for agility by information systems technology. Being agile, a Virtual Enterprise can only be characterized, not defined.

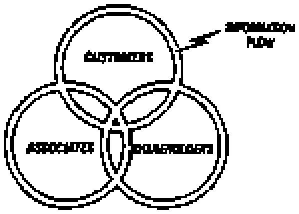


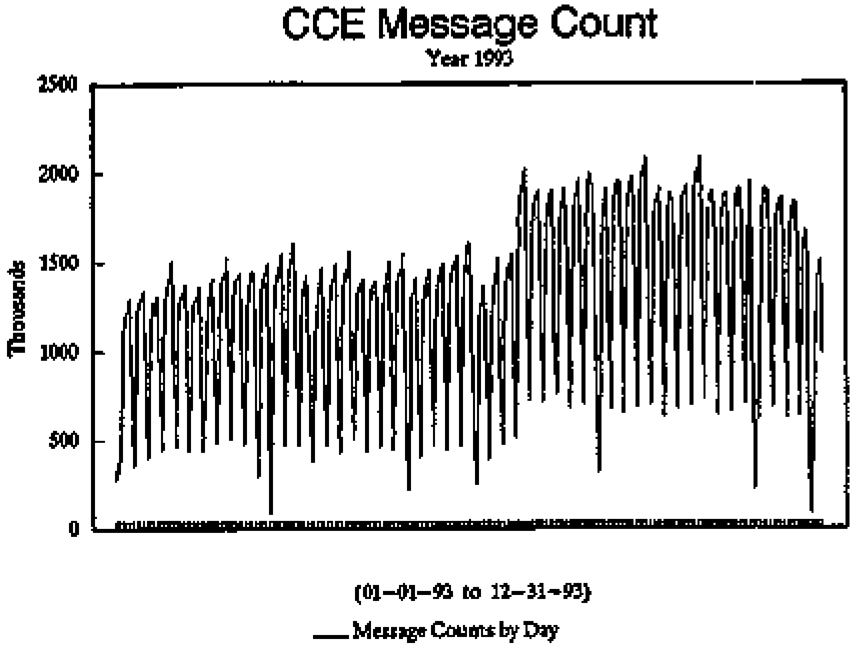
Fig 19 · Corporate Symbiosis

Associates satisfy customers; customers provide shareholders value; shareholders support associates (Fig 19).

The VIRTUAL ENTERPRISE 2000 leverages this corporate symbiosis.

MANAGING USER EXPECTATIONS

Tom Maurer, Summit Information Systems



HIGH PERFORMANCE COMPUTING

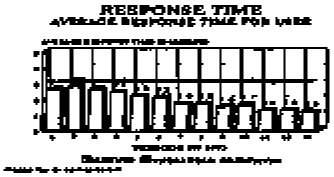
IS DEFINED BY THE CUSTOMER'S

EXPECTATIONS

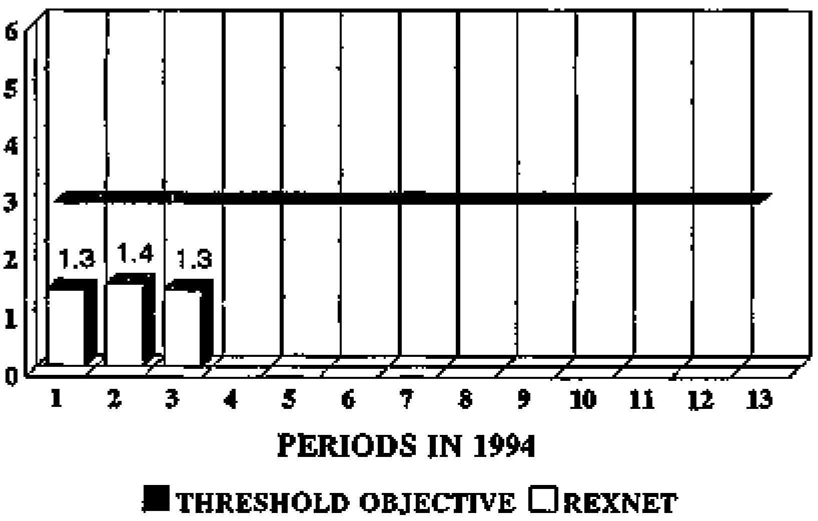
TECHNICAL PARAMETERS

- RESPONSE TIME
- CAPACITY
- AVAILABILITY
- RELIABILITY
- RECOVERABILITY
- SECURITY

RESPONSE TIME
AVERAGE RESPONSE TIME FOR USER



AVERAGE RESPONSE TIME IN SECONDS



Response time for users in the field

BUSINESS PARAMETERS

- PRIMARY NEEDS
- EASE OF CHANGE
- COST TO OPERATE/MAINTAIN
- TOUCH/FEEL
- AD HOC REQUIREMENTS

MANAGING THE EXPECTATION

- CUSTOMER SUPPLIER AGREEMENTS
- IDENTIFY CRITICAL PROCESSES
- SERVICE REPORT CARDS
- KEY MEASURES
- CUSTOMER SURVEYS
- BUSINESS PLANNING PARTNER
- SYSTEMS OWNERS
- MATCHING SUPPLIERS TO THE TASK

THE QUALITY CYCLE

- IDENTIFY IMPROVEMENT OPPORTUNITIES
- IDENTIFY KEY CUSTOMERS/SUPPLIERS
- AGREE ON REQUIREMENTS
- DESCRIBE CURRENT PROCESS
- IDENTIFY THE GAPS
- DETERMINE R00T CAUSES
- DEVELOP/IMPLEMENT SOLUTIONS
- MEASURE AND MONITOR

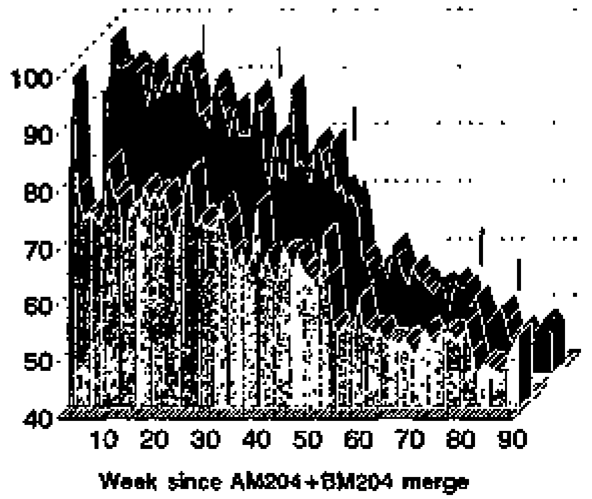
MONITORING KEY MEASURES

DRIVES IMPROVEMENT

Model 204 Resource Utilization

since AM204+BM204 merge

processor cycles and storage reads per transaction



Week 19: rewrote SEIF033

Week 32: installed performance monitor

Week 48: EDI merge w/ TEC; moved TMC319 links to CCE

Week 67: Reseized CJF File

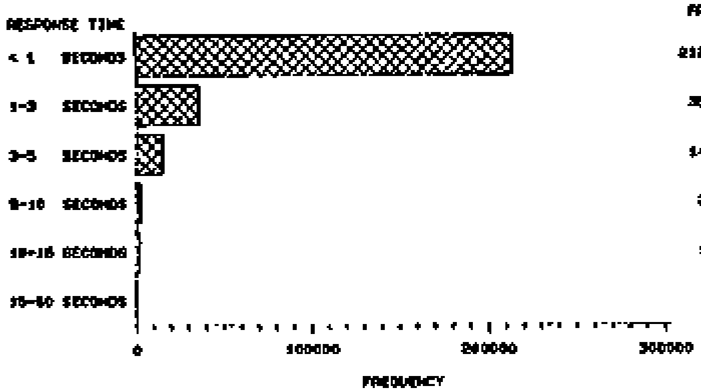
Week 79: Installed solid state disk; ASD deployment begun

█ storage reads

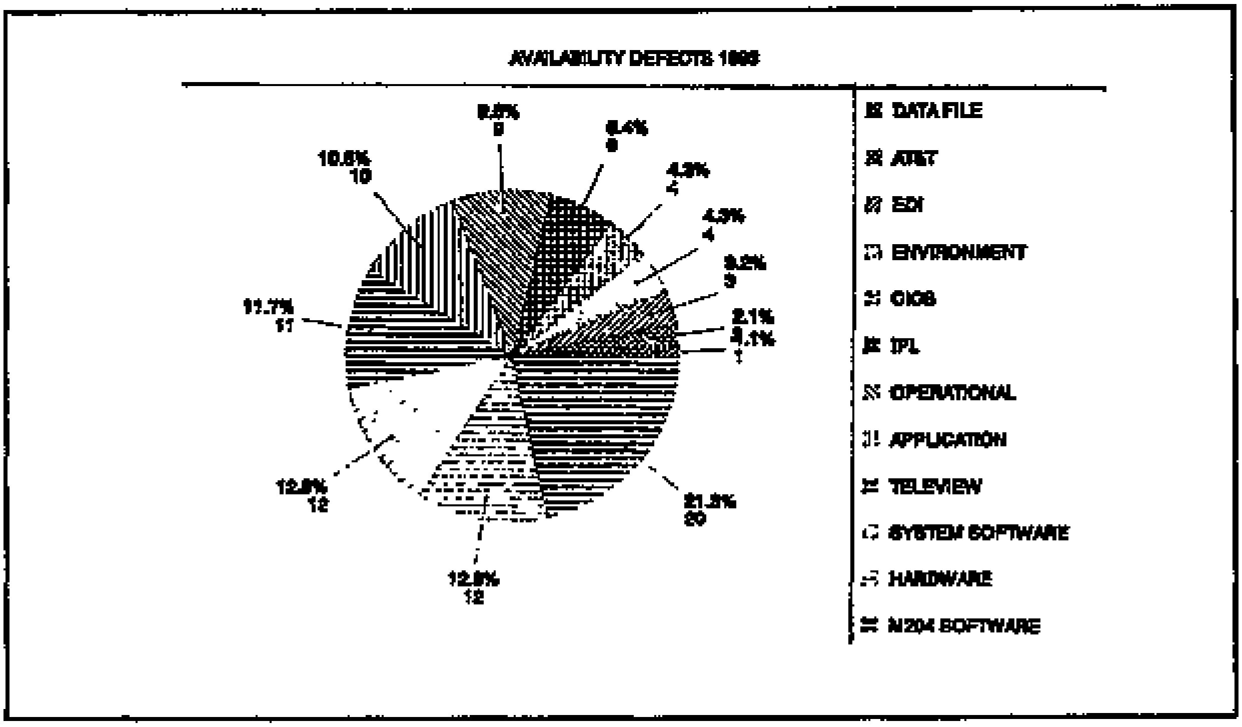
processor cycles

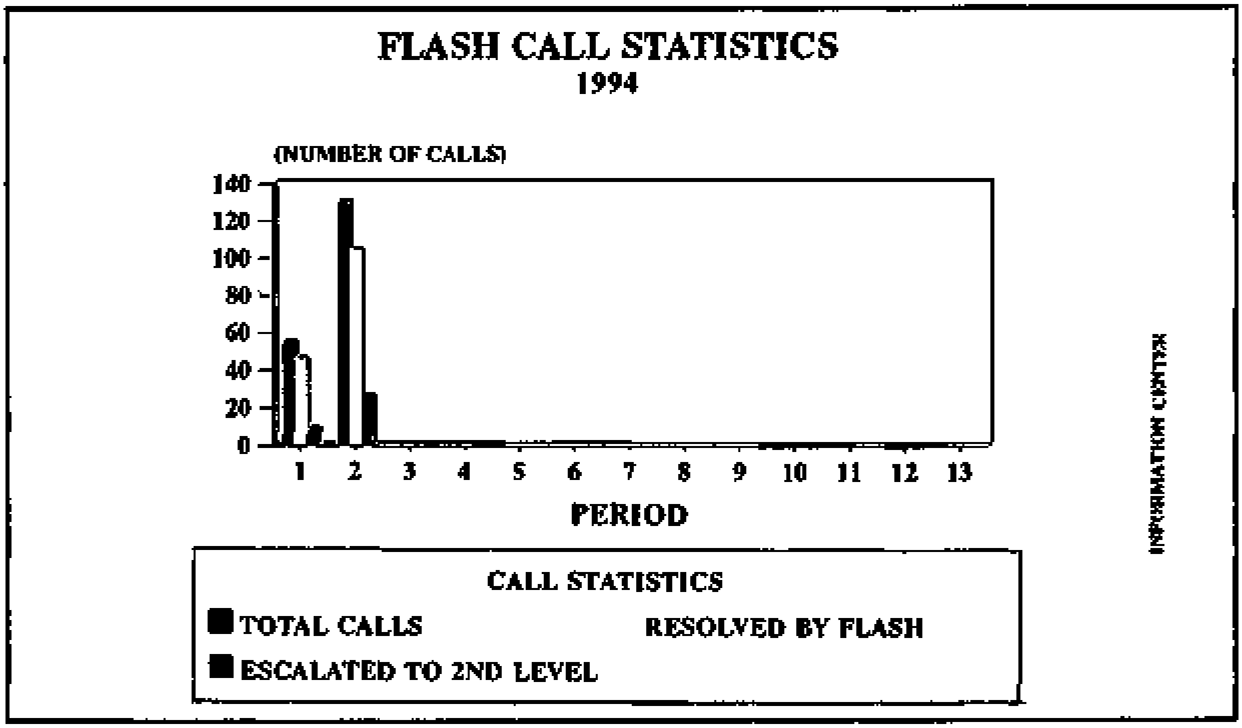
CICS/IMS RESPONSE TIME FOR STORE AND FORWARD SYSTEM

CICS System Identification=PROJ APPLICATION=CCE COMM



RESPONSE TIME	FREQ.	PCT.	CUM. PCT.
< 1 SECONDS	212911	78.62	78.62
1-3 SECONDS	35301	13.03	92.05
3-5 SECONDS	14032	5.18	98.43
5-10 SECONDS	2395	0.90	99.33
10-15 SECONDS	1719	0.64	99.97
15-60 SECONDS	93	0.03	100.00


AVAILABILITY DEFECTS 1993
9.6%
9
6.4%
6
4.3%
4
4.3%
4
3.2%
3
2.1%
2
1.1%
1
21.3%
20
12.8%
12
12.8%
12
11.7%
11
10.6%
10
DATA FILE
AT&T
EDI
ENVIRONMENT
CICS
IPL
OPERATIONAL
APPLICATION
TELEVIEW
SYSTEM SOFTWARE
HARDWARE
M204 SOFTWARE

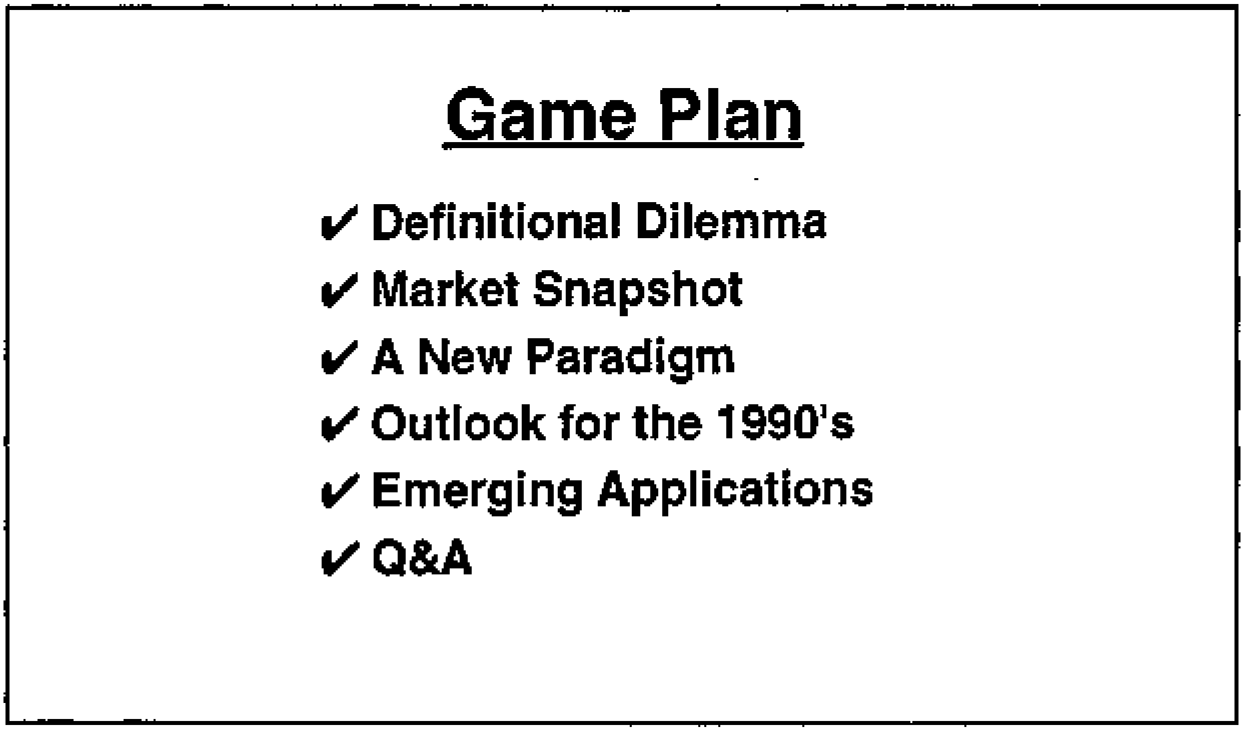

FLASH CALL STATISTICS
1994
(NUMBER OF CALLS)
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
11
12
13
PERIOD
CALL STATISTICS
TOTAL CALLS
RESOLVED BY FLASH
ESCALATED TO 2ND LEVEL
INFORMATION CENTER

FINAL THOUGHTS

- CUSTOMERS PARTICIPATE THROUGHOUT
- NO MAGIC SOLUTIONS
- COSTING MECHANISMS CRITICAL
- PERCEPTION SENSING ONGOING

Survival of the Fittest

Gary Smaby, SMABY GROUPS



Game Plan

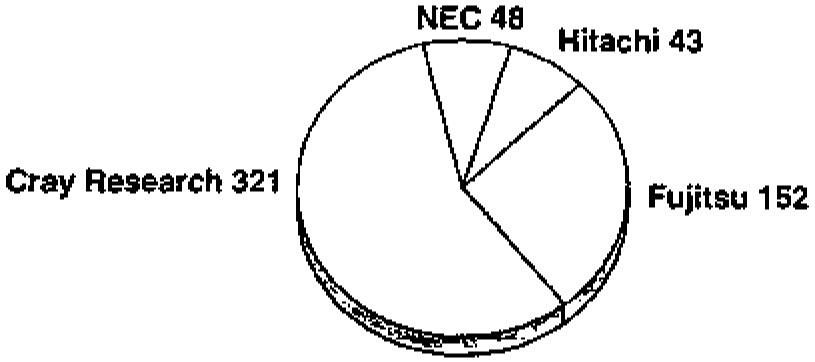
- ✓ Definitional Dilemma
- ✓ Market Snapshot
- ✓ A New Paradigm
- ✓ Outlook for the 1990's
- ✓ Emerging Applications
- ✓ Q&A

ISLA MUJERES - MINISUPER

1993 Supercomputer Market

Leading Enterprise Vendors

564 Systems

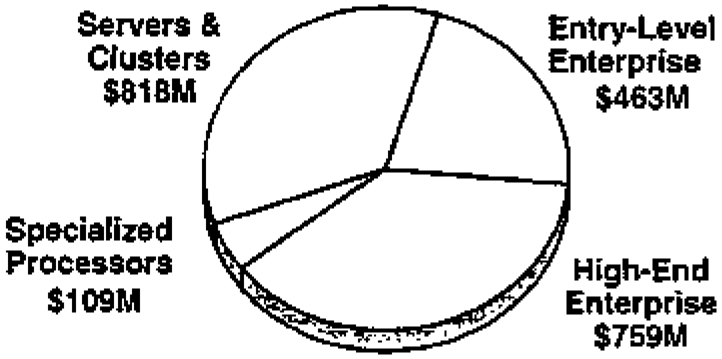


Total System Installations / Worldwide

1993 Supercomputer Market

Scientific/Engineering/Technical

$2.15 Billion

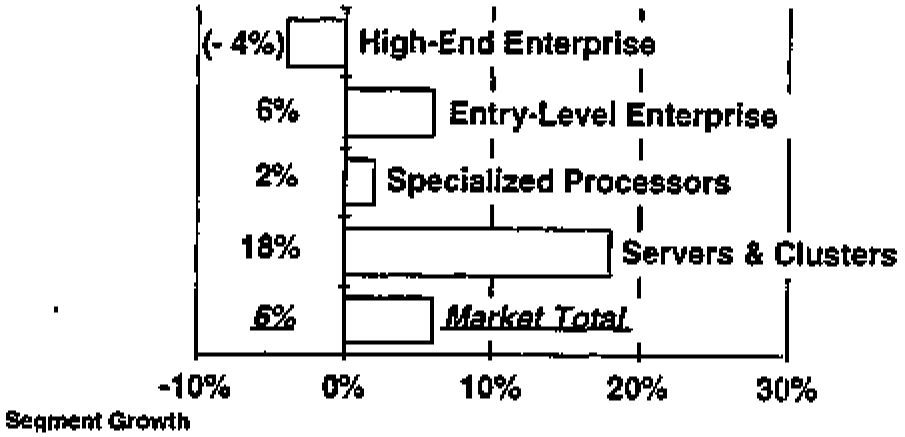


Total Worldwide Factory Revenues - Hardware

Supercomputer Market

Scientific/Engineering/Technical

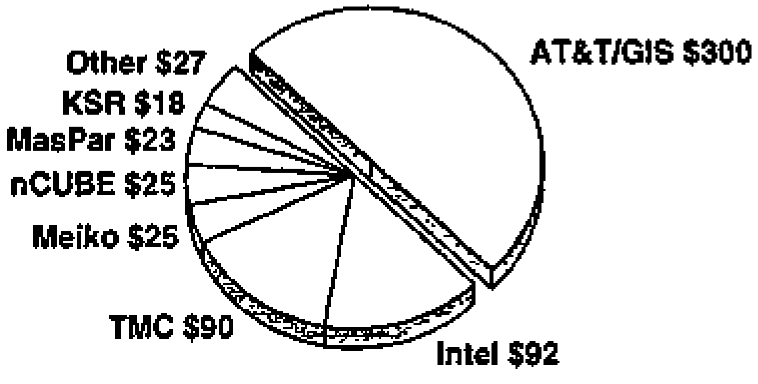
1992 to 1993

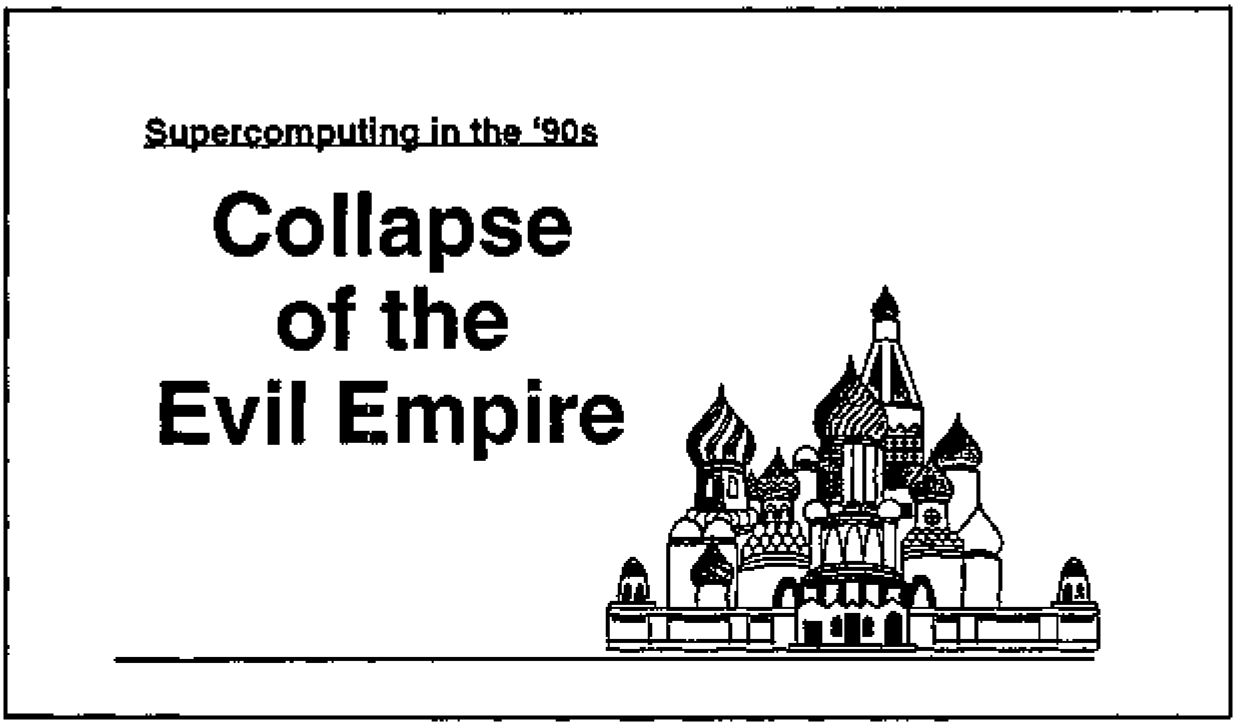


1993 MPP Market Leaders

S/E/T and Commercial Database

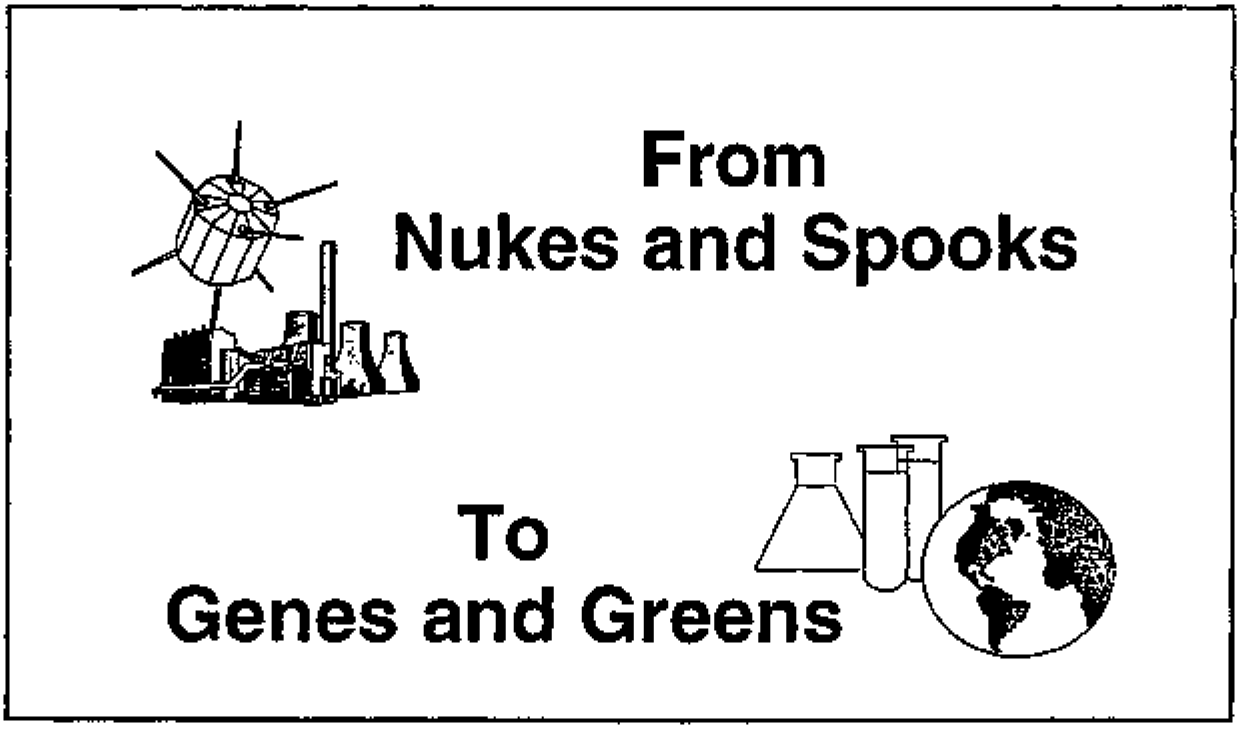
$600 Million





Supercomputing in the '90s

Collapse of the Evil Empire

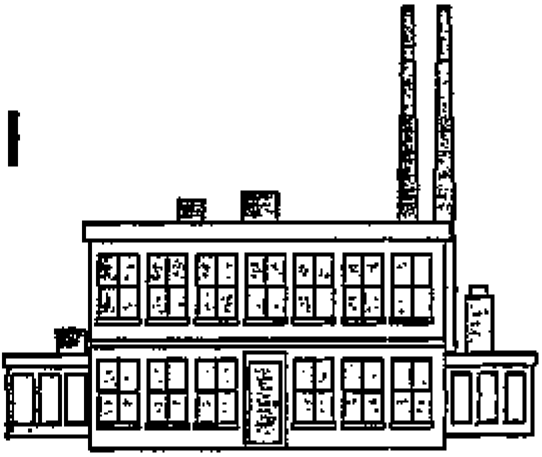


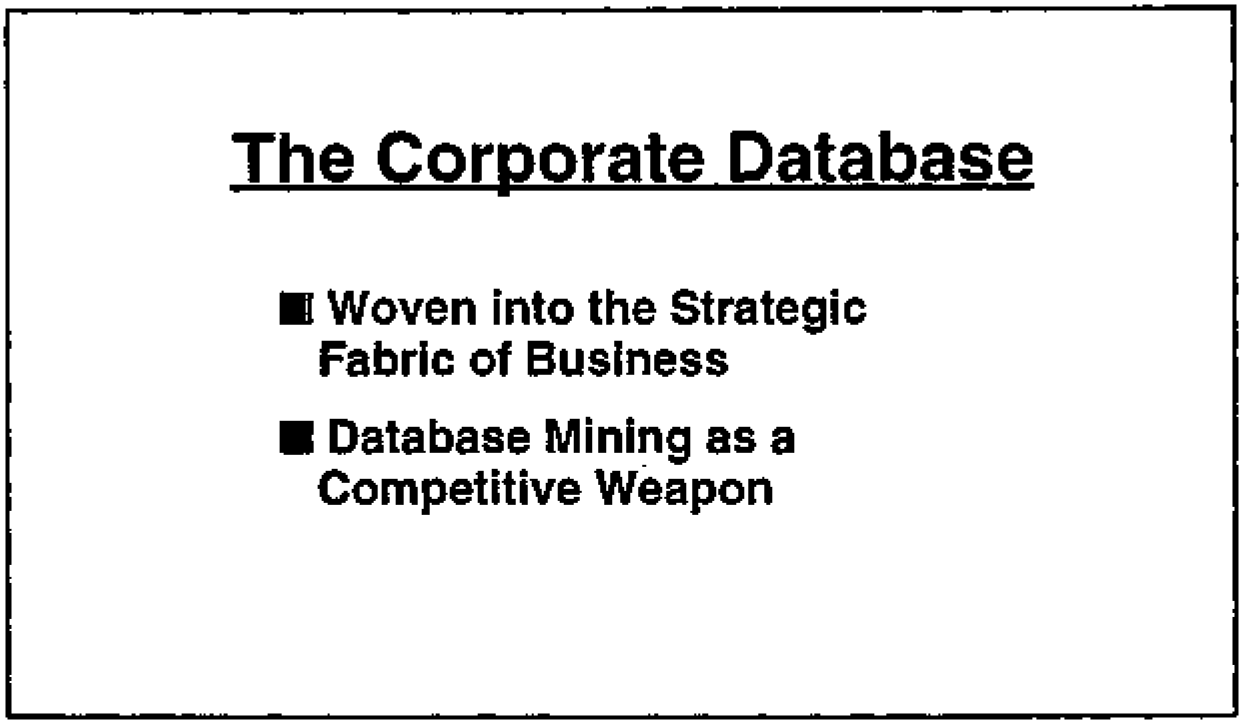
From Nukes and Spooks

To Genes and Greens

Other People's Money

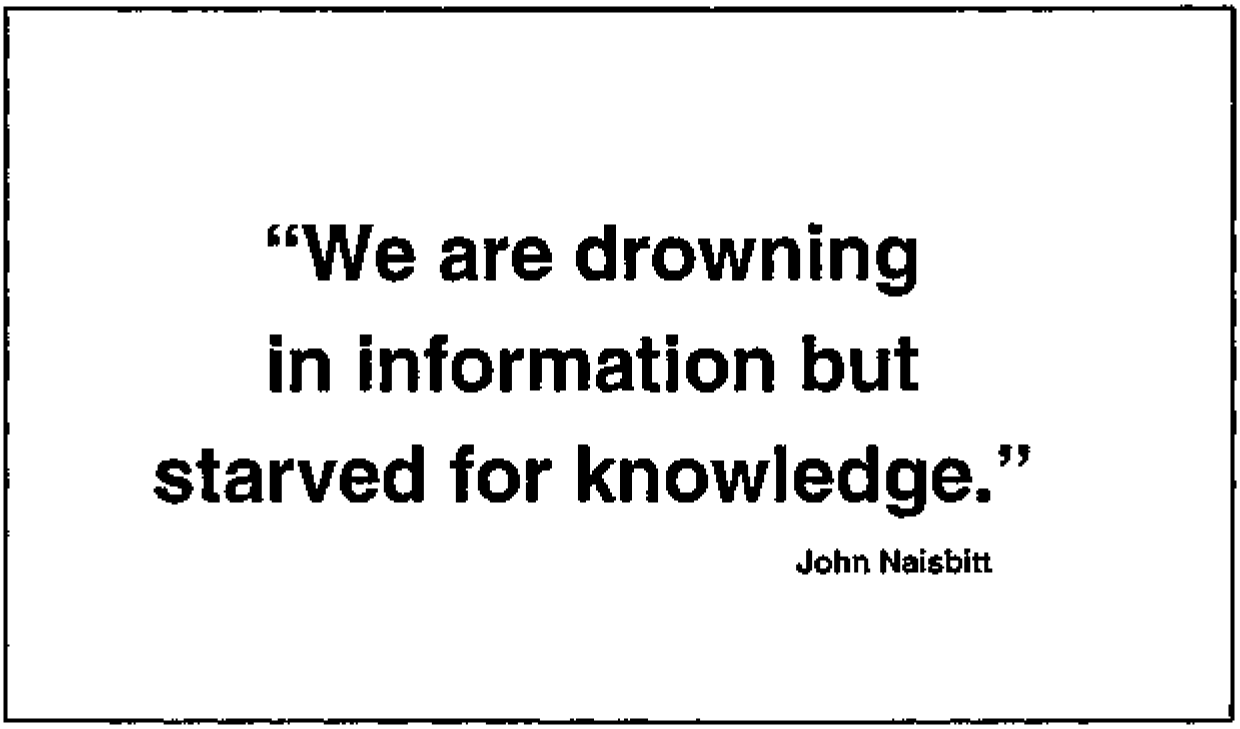
MPP Goes Commercial





The Corporate Database

- Woven into the Strategic Fabric of Business
- Database Mining as a Competitive Weapon



"We are drowning in information but starved for knowledge."

John Naisbitt

The Opportunity:

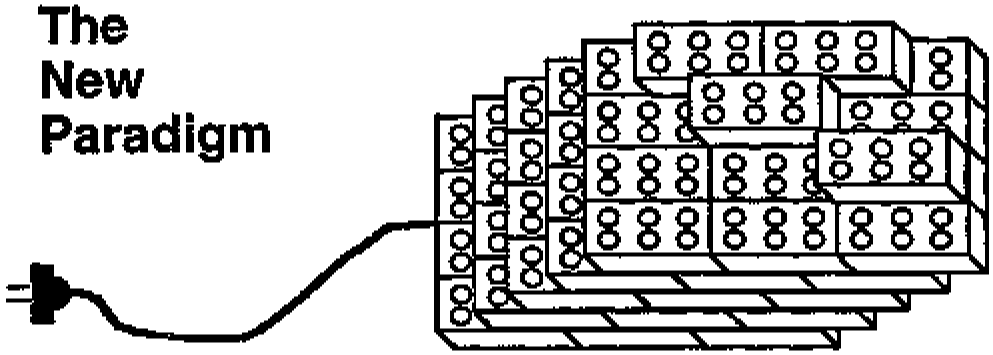
Forward-thinking IS managers are deploying new MPP technologies to competitively leverage their most valuable asset.

Why MPP?

- Compelling Price/performance
- Tremendous Scalability
- Tolerable Entry Price
- Tackle Intractable Problems

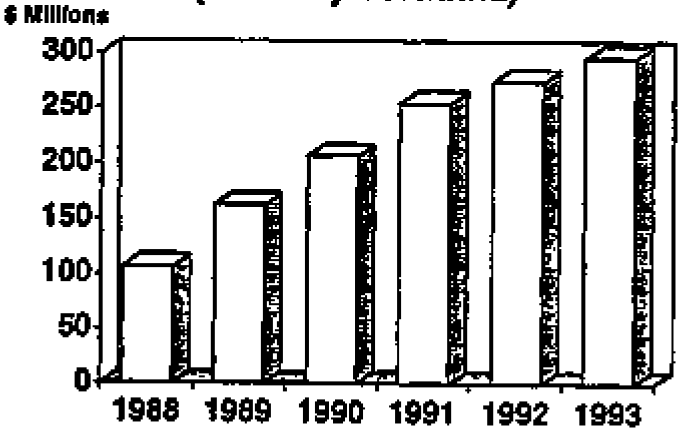
LEGO Computing

The
New
Paradigm



AT&T/GIS

(formerly Teradata)

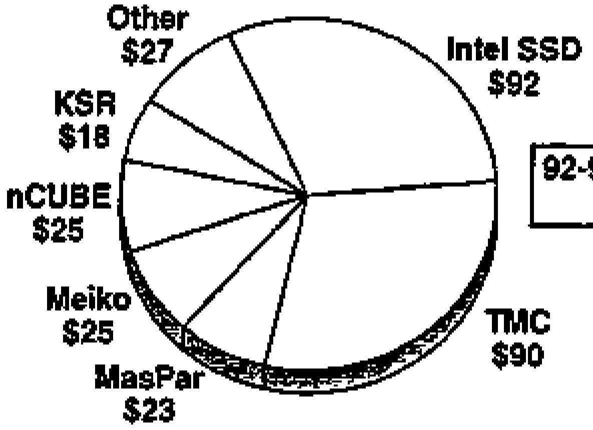


Product Revenues

1993 MPP Market Leaders

Scientific/Engineering/Technical

$300 Million



92-93 Growth Rate
15%

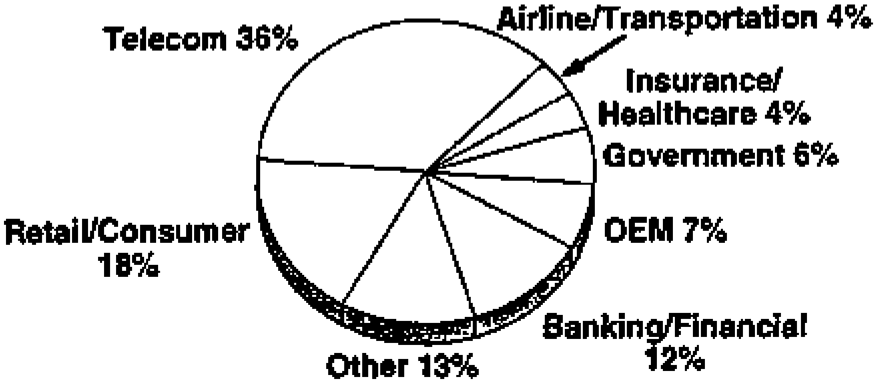
Can Early Adopters Claim Success?

Real customers are tackling real problems across the industrial spectrum:

- Financial
- Banking
- Health Care
- Manufacturing
- Retail
- Transportation

AT&T/GIS

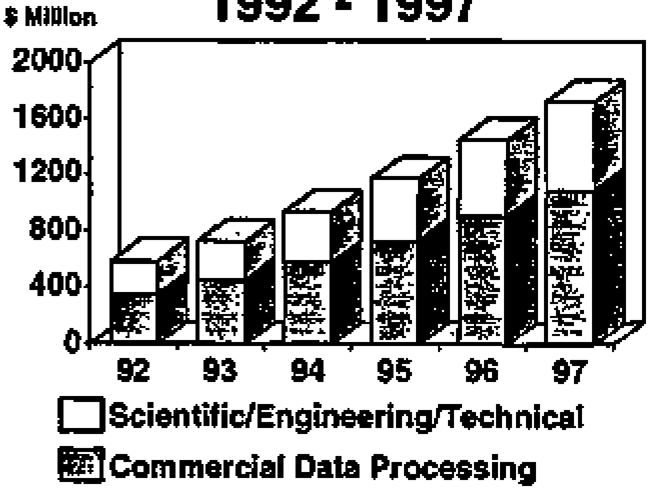
(formerly Teradata)



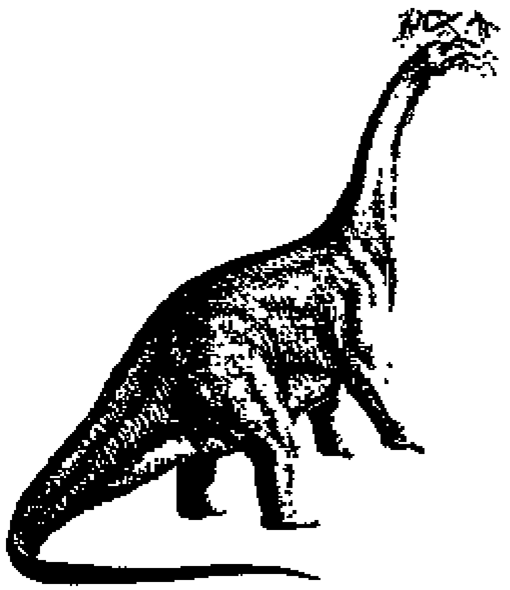
1991 Installed Base by Industry

MPP Segmentation

1992 - 1997



The
Old
Paradigm

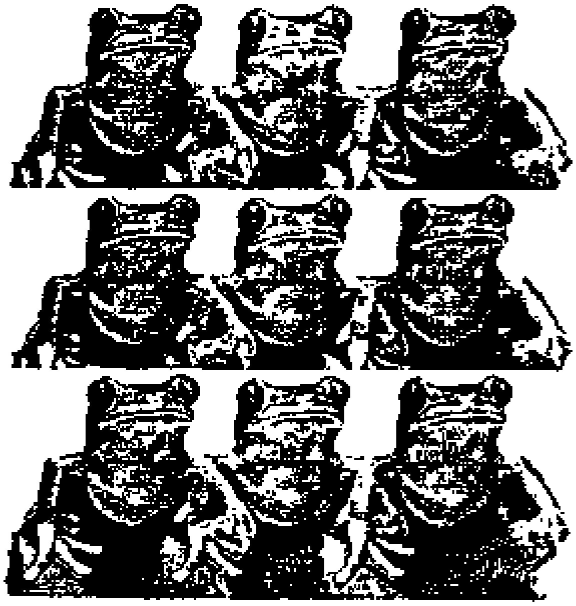


IBM 3090 PHOTO

A New Paradigm

- ✔ Killer Micros
- ✔ Software is King
- ✔ Open Architectures
- ✔ Standards Dominate
- ✔ Proliferating Networks

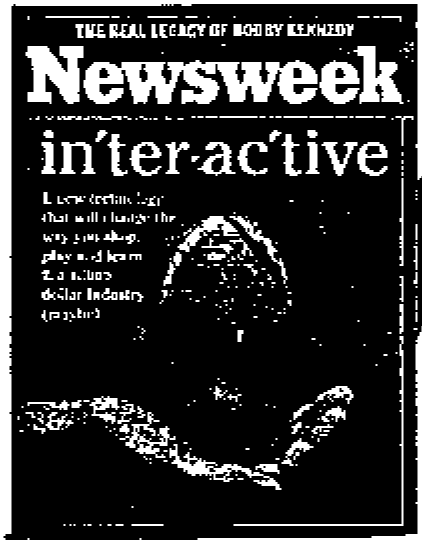
The New Paradigm



Near-Term Outlook

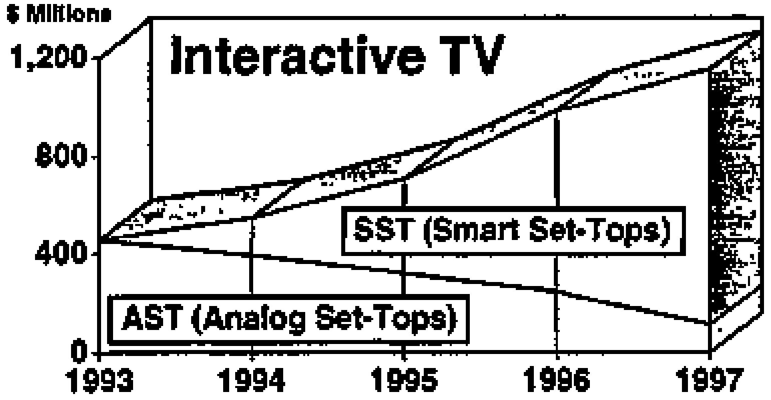
- MPP - Enabling Technology
- New Class of Applications
- One Piece of the Puzzle
- From Testbed to Production
- Transition Will Take Time

Emerging Applications



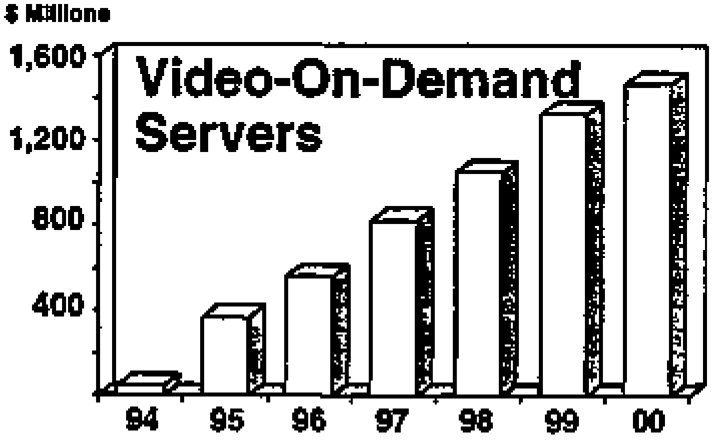
1. new technology that will change the way you shop, play and learn
2. a zillion-dollar industry (maybe)

Emerging Applications



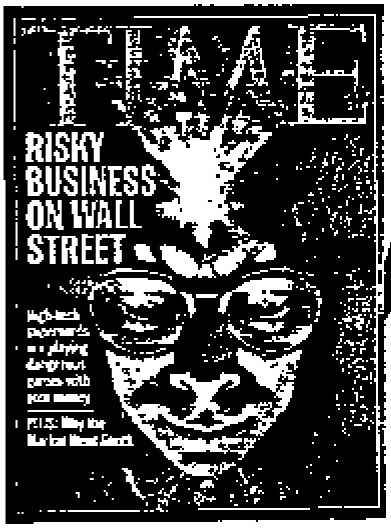
Annual Sales

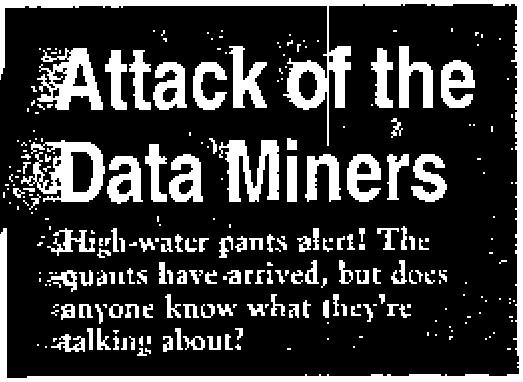
Emerging Applications



Annual Sales / High-End Systems

Emerging Applications




Attack of the Data Miners

High-water pants alert! The quants have arrived, but does anyone know what they're talking about?

"The greatest challenge of the computer industry is to learn how to build *information* bases, not databases."

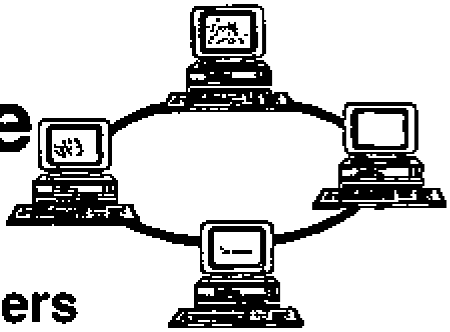
Peter F. Drucker

Is MPP Industrial Strength?

- Connectivity and Transparency are Key to Commercial Users
- Support Tools and Infrastructures Still Immature

Supercomputing in the '90s

Variations on a RISCy Theme

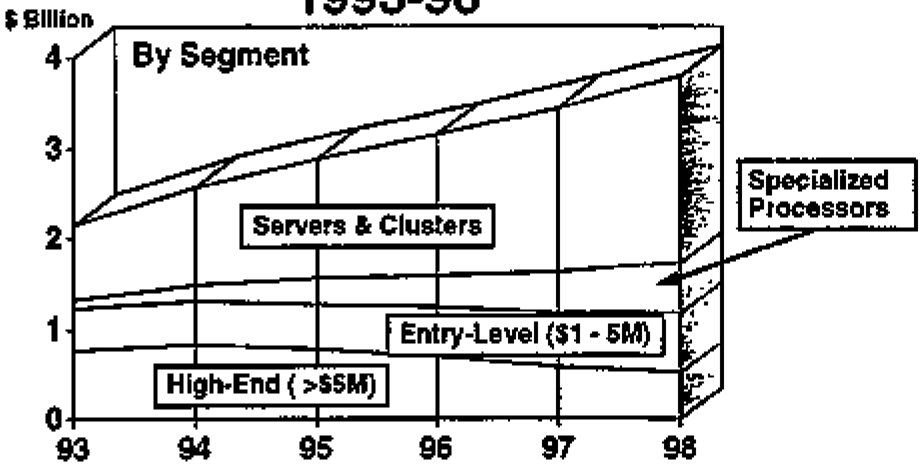


Farms, Clusters, Servers

Supercomputer Market

Scientific/Engineering/Technical

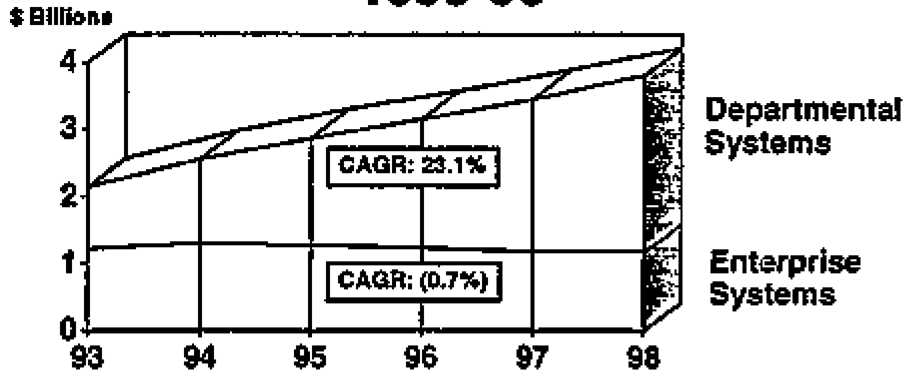
1993-98



Supercomputer Market

Scientific/Engineering/Technical

1993-98

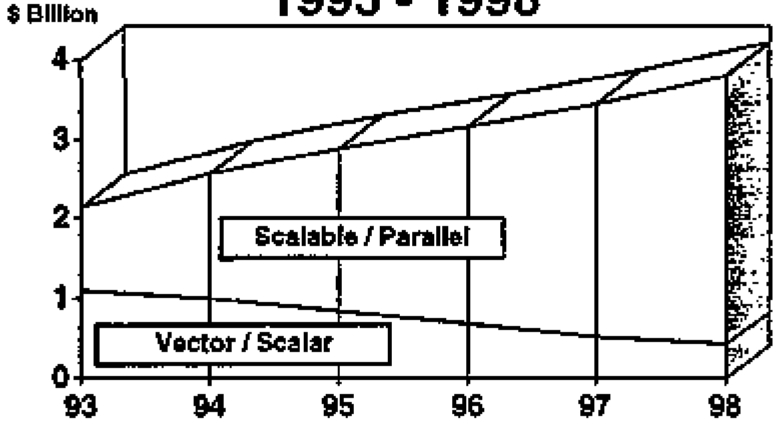


Forecast By Market Segment

Architectural Evolution

Scientific/Engineering/Technical

1993 - 1998



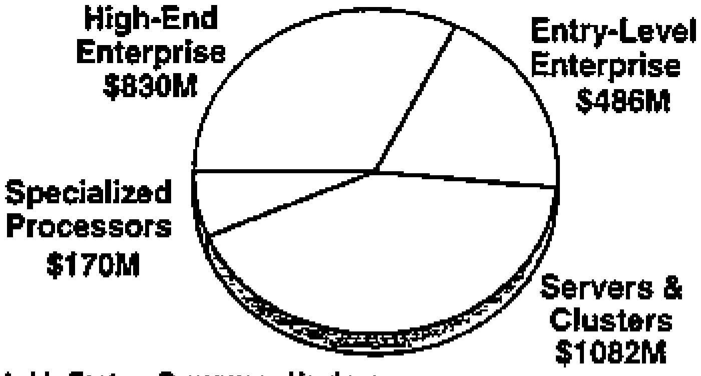
Outlook for the '90s

- ✔ Moderating Growth
- ✔ Continued Globalization
- ✔ Decade of the Database
- ✔ Complex Distributed Computing
- ✔ Digitization of Everything

1994 Supercomputer Market

Scientific/Engineering/Technical

$2.57 Billion

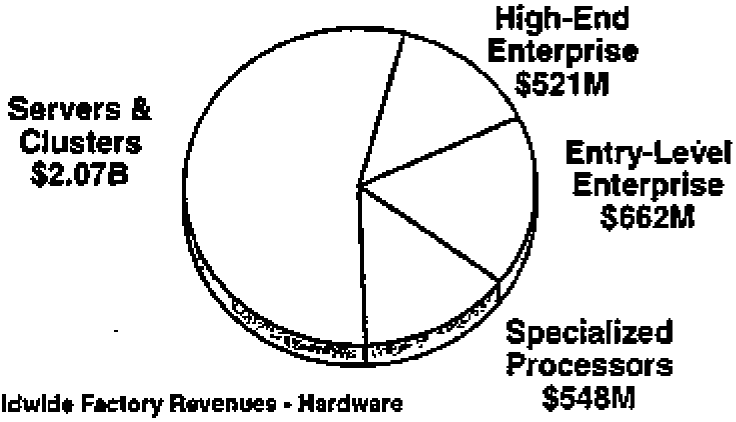


Total Worldwide Factory Revenues - Hardware

1998 Supercomputer Market

Scientific/Engineering/Technical

$3.8 Billion

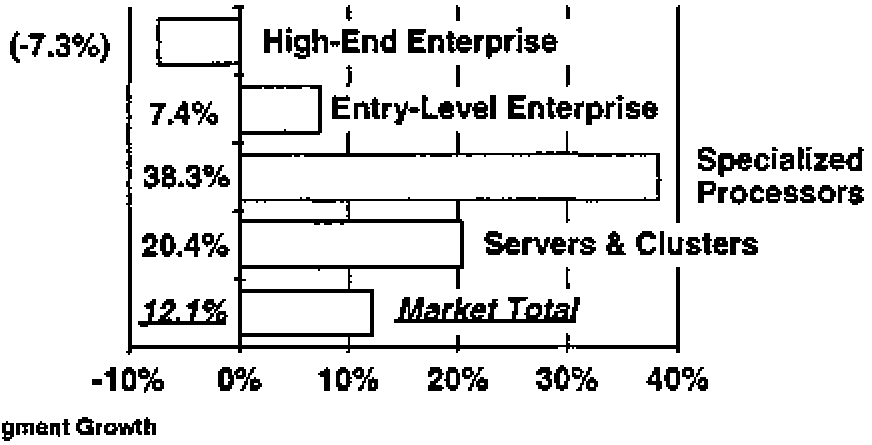


Total Worldwide Factory Revenues - Hardware

Supercomputer Market

Scientific/Engineering/Technical

1993-98



Challenges

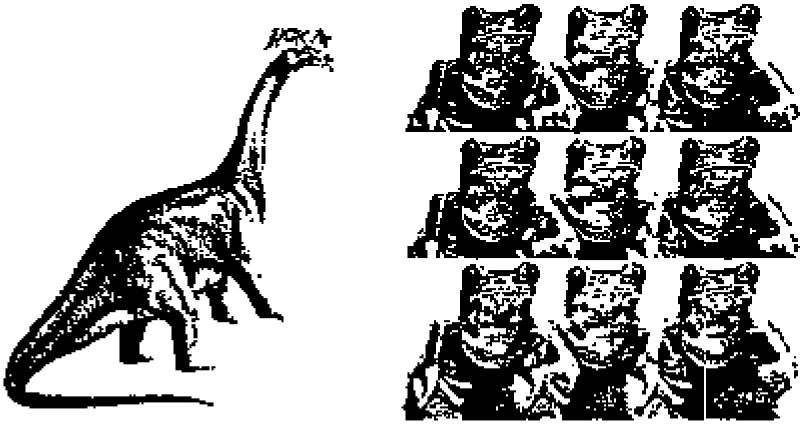
- ✔ Riding Out the Recession
- ✔ Staying Lean and Mean
- ✔ Picking the Right Horse
- ✔ Shrinking Product Cycles
- ✔ Product Differentiation

SGI LOGO

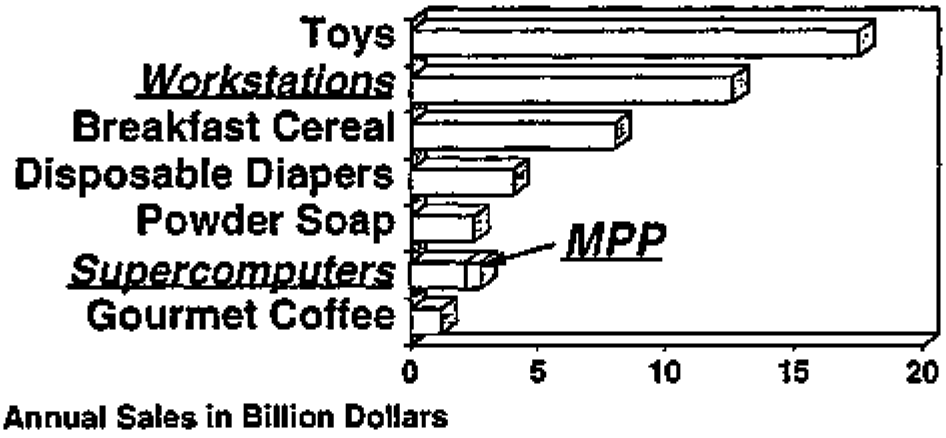
CRAY LOGO

IBM LOGO

Peaceful Coexistence?

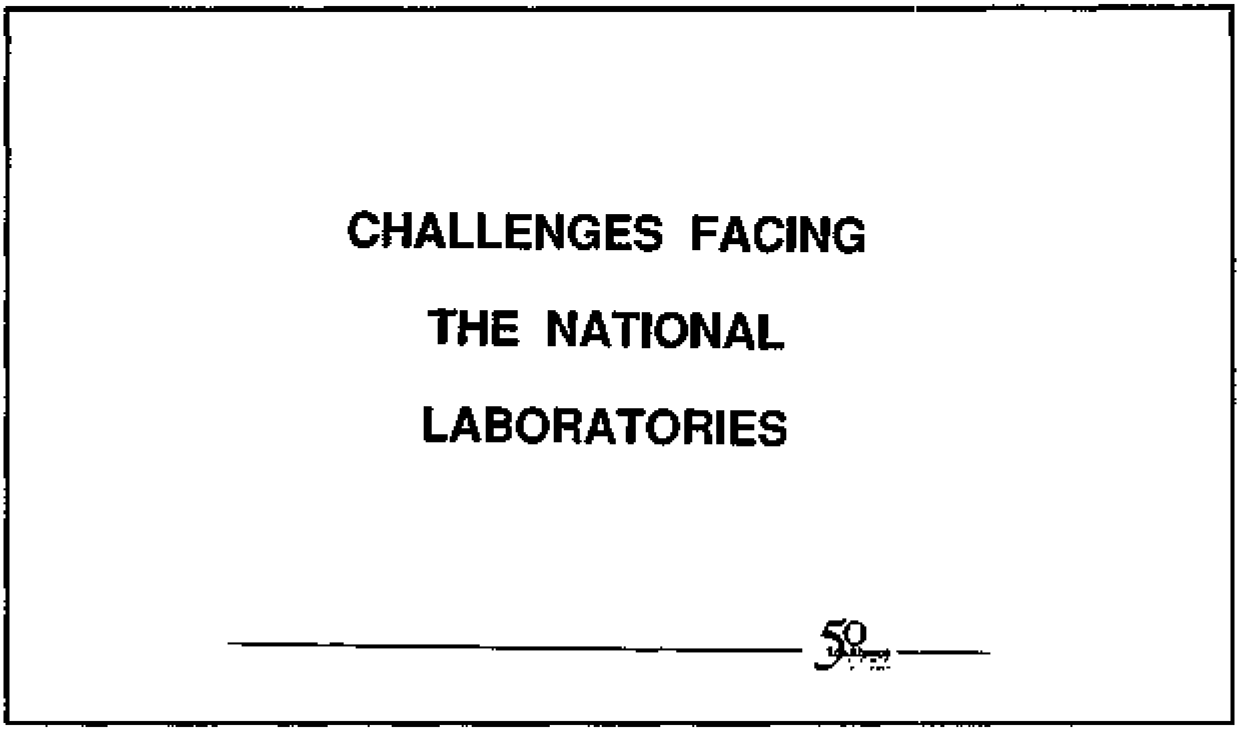


Industry in Context



IS ECONOMIC COMPETITIVENESS A MISSION?

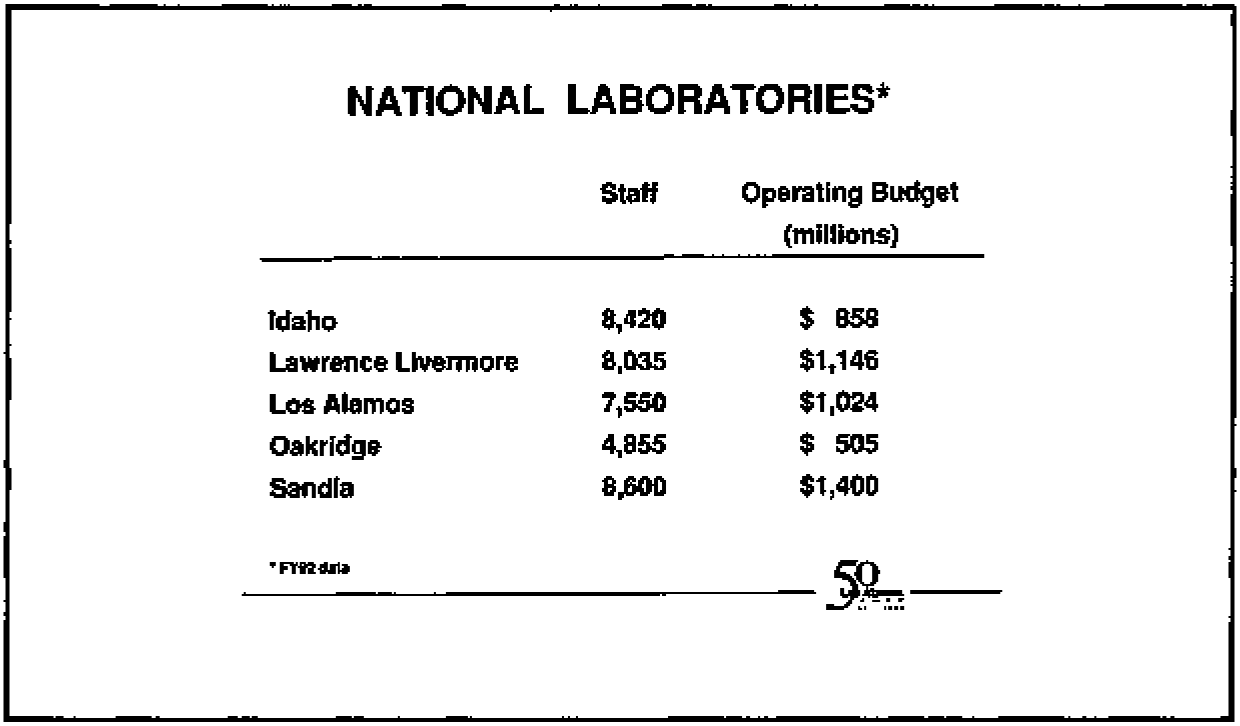
Hassan Dayem, LANL



CHALLENGES FACING

THE NATIONAL

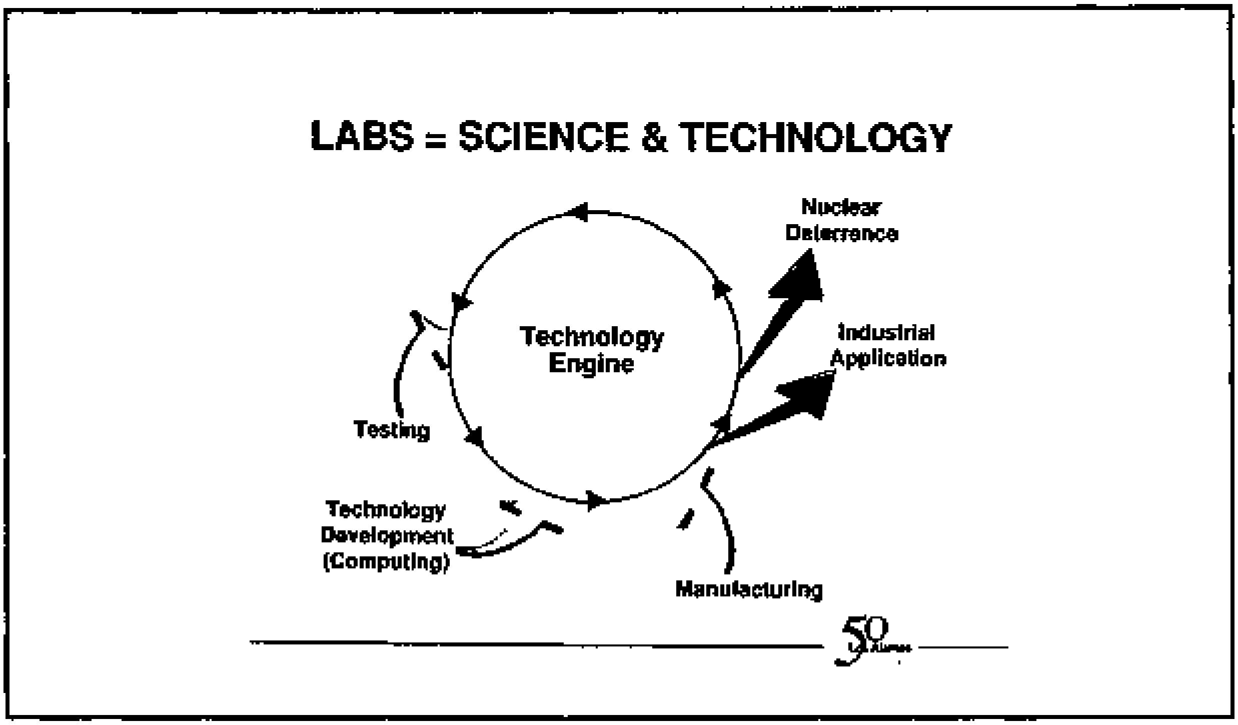
LABORATORIES



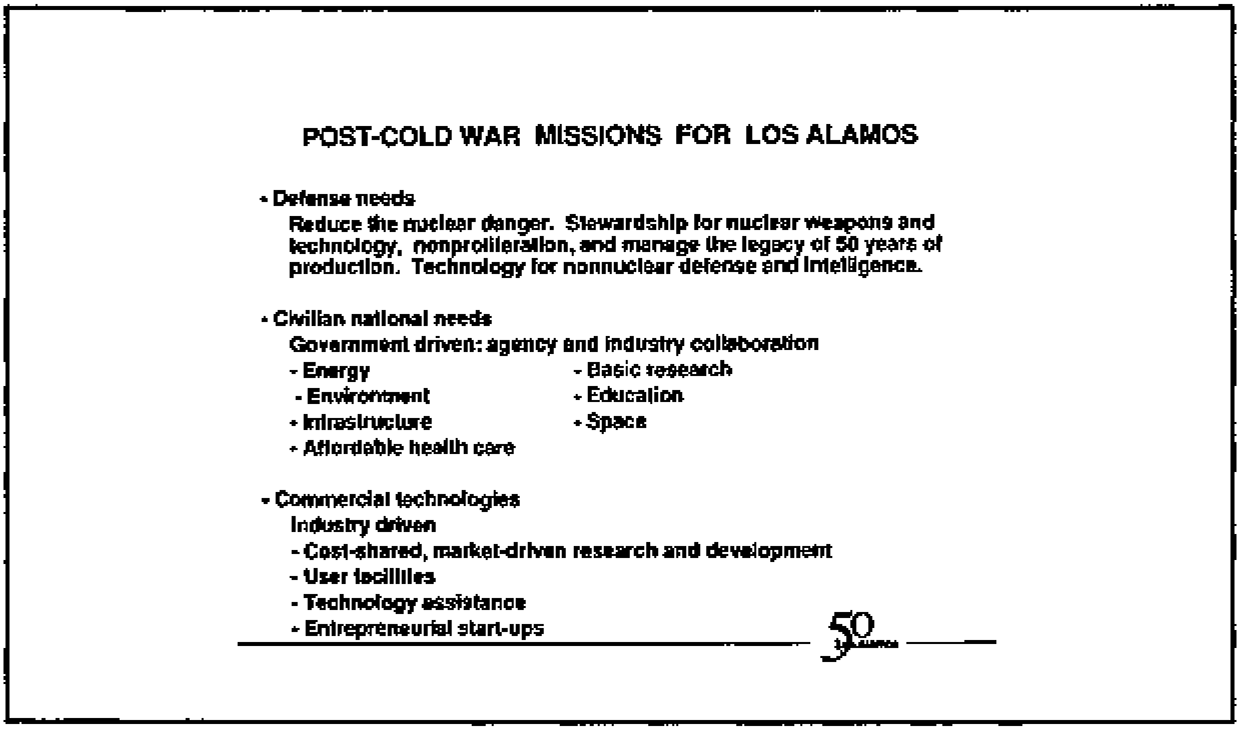
NATIONAL LABORATORIES*

	Staff	Operating Budget (millions)
Idaho	8,420	$ 858
Lawrence Livermore	8,035	$1,146
Los Alamos	7,550	$1,024
Oakridge	4,855	$ 505
Sandia	8,600	$1,400

* FY92 data



LABS = SCIENCE & TECHNOLOGY

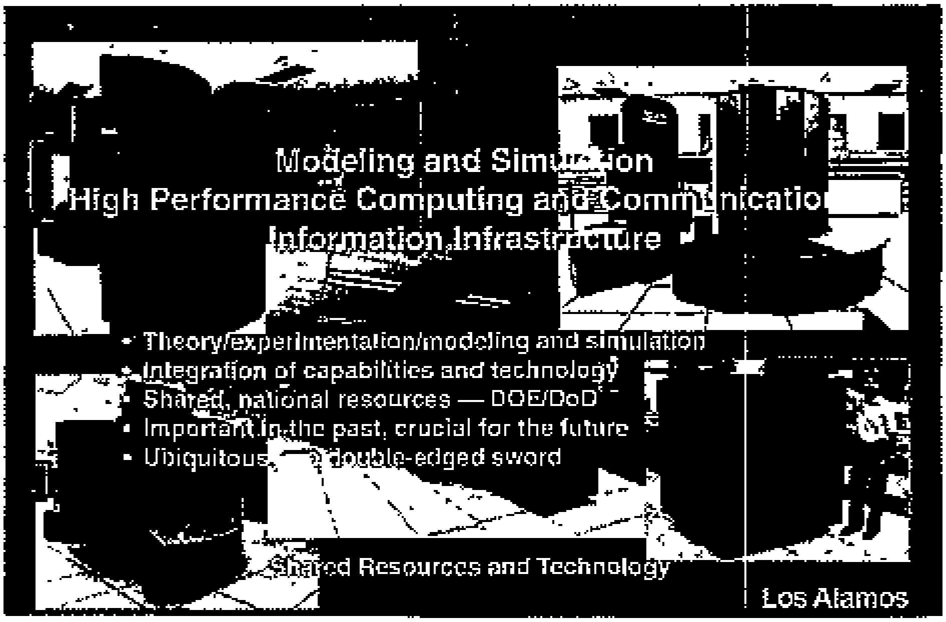


POST-COLD WAR MISSIONS FOR LOS ALAMOS

- Defense needs
  Reduce the nuclear danger. Stewardship for nuclear weapons and technology, nonproliferation, and manage the legacy of 50 years of production. Technology for nonnuclear defense and intelligence.

- Civilian national needs
  Government driven: agency and industry collaboration
  - Energy
  - Environment
  - Infrastructure
  - Affordable health care
  - Basic research
  - Education
  - Space

- Commercial technologies
  Industry driven
  - Cost-shared, market-driven research and development
  - User facilities
  - Technology assistance
  - Entrepreneurial start-ups

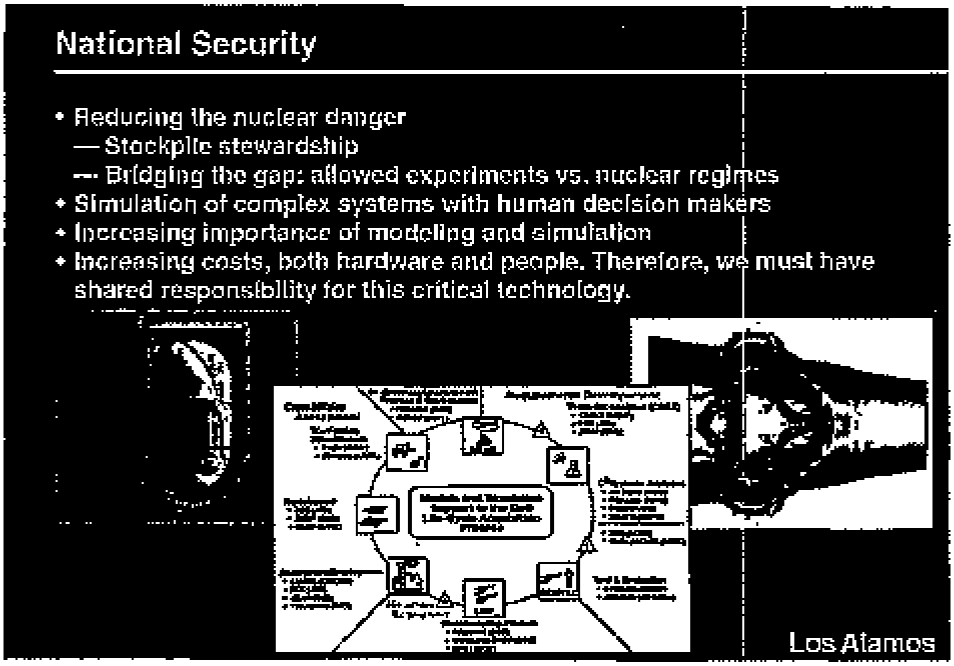


Modeling and Simulation
High Performance Computing and Communications
Information Infrastructure

- Theory/experimentation/modeling and simulation
- Integration of capabilities and technology
- Shared, national resources — DOE/DoD
- Important in the past, crucial for the future
- Ubiquitous, a double-edged sword

Shared Resources and Technology

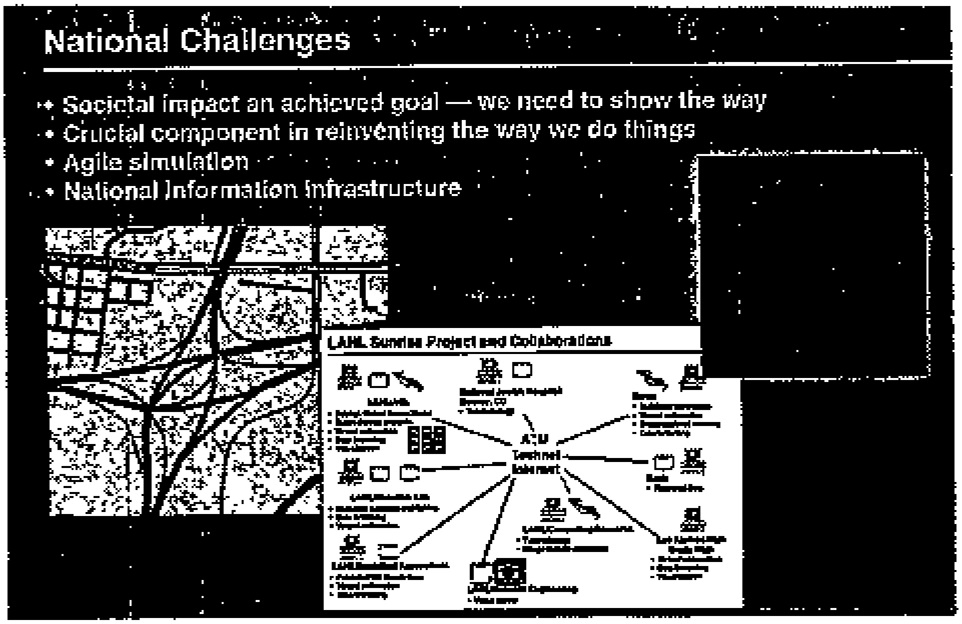
Los Alamos



National Security

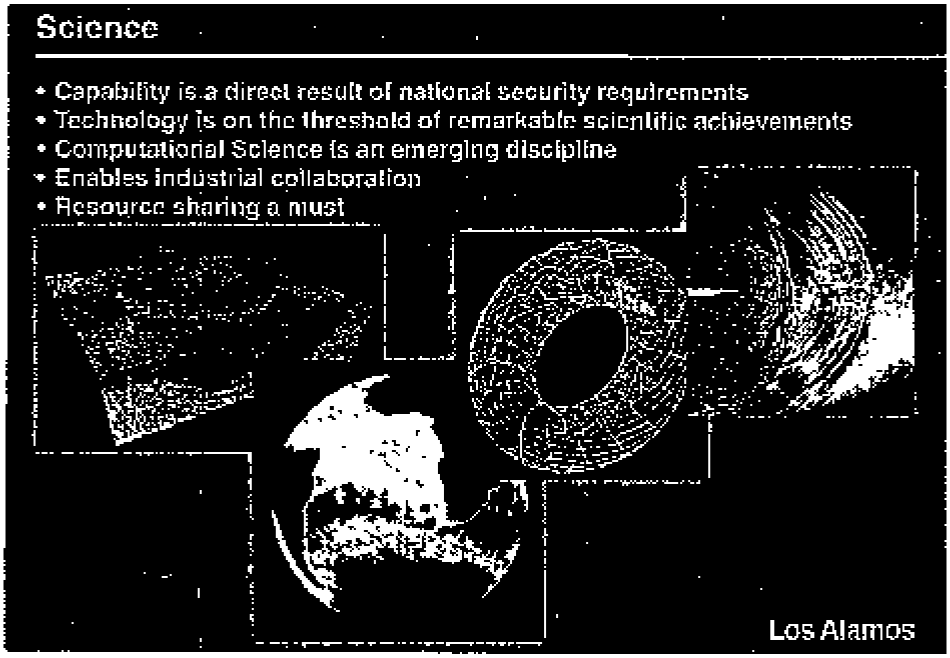
- Reducing the nuclear danger
  - — Stockpile stewardship
  - — Bridging the gap: allowed experiments vs. nuclear regimes
- Simulation of complex systems with human decision makers
- Increasing importance of modeling and simulation
- Increasing costs, both hardware and people. Therefore, we must have shared responsibility for this critical technology.

Los Alamos



National Challenges

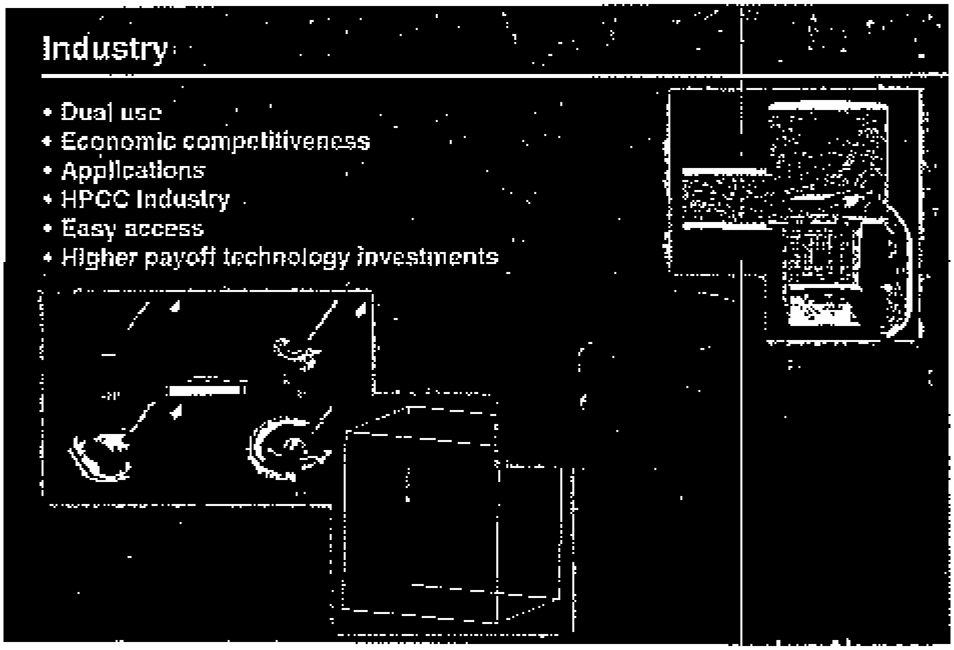
- Societal impact an achieved goal — we need to show the way
- Crucial component in reinventing the way we do things
- Agile simulation
- National Information Infrastructure



Science

- Capability is a direct result of national security requirements
- Technology is on the threshold of remarkable scientific achievements
- Computational Science is an emerging discipline
- Enables industrial collaboration
- Resource sharing a must

Los Alamos

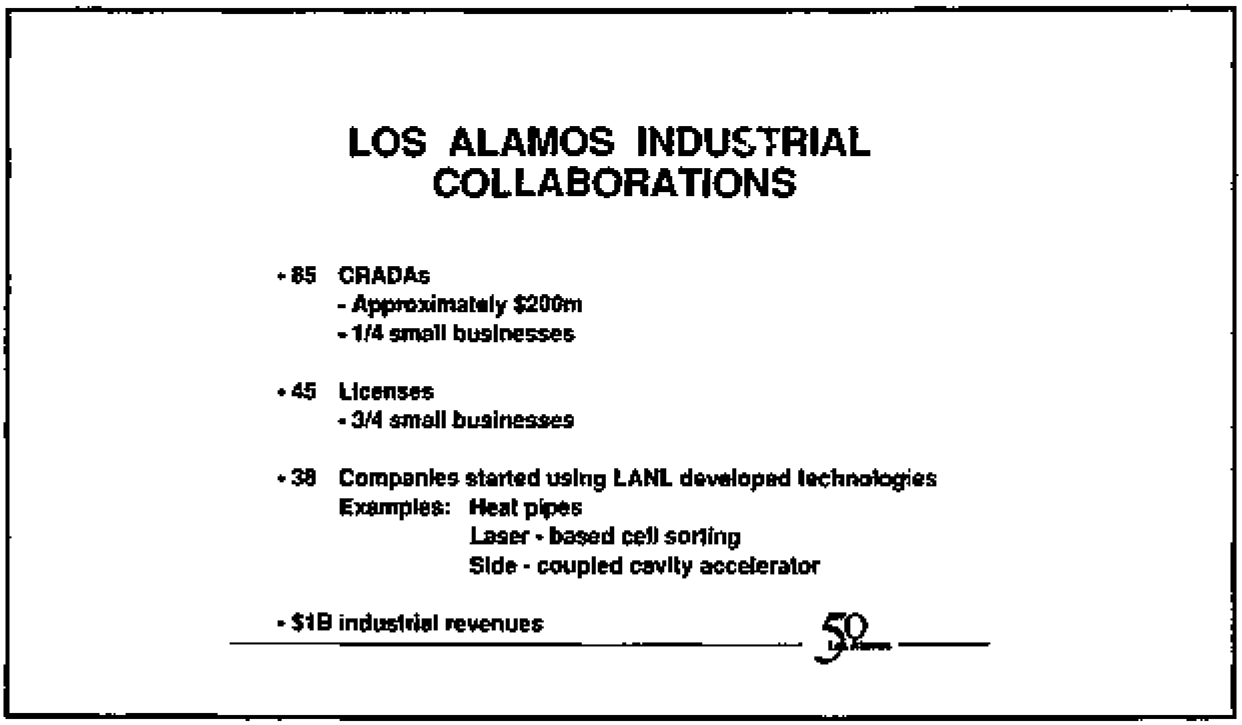


Industry

- Dual use
- Economic competitiveness
- Applications
- HPCC Industry
- Easy access
- Higher payoff technology investments

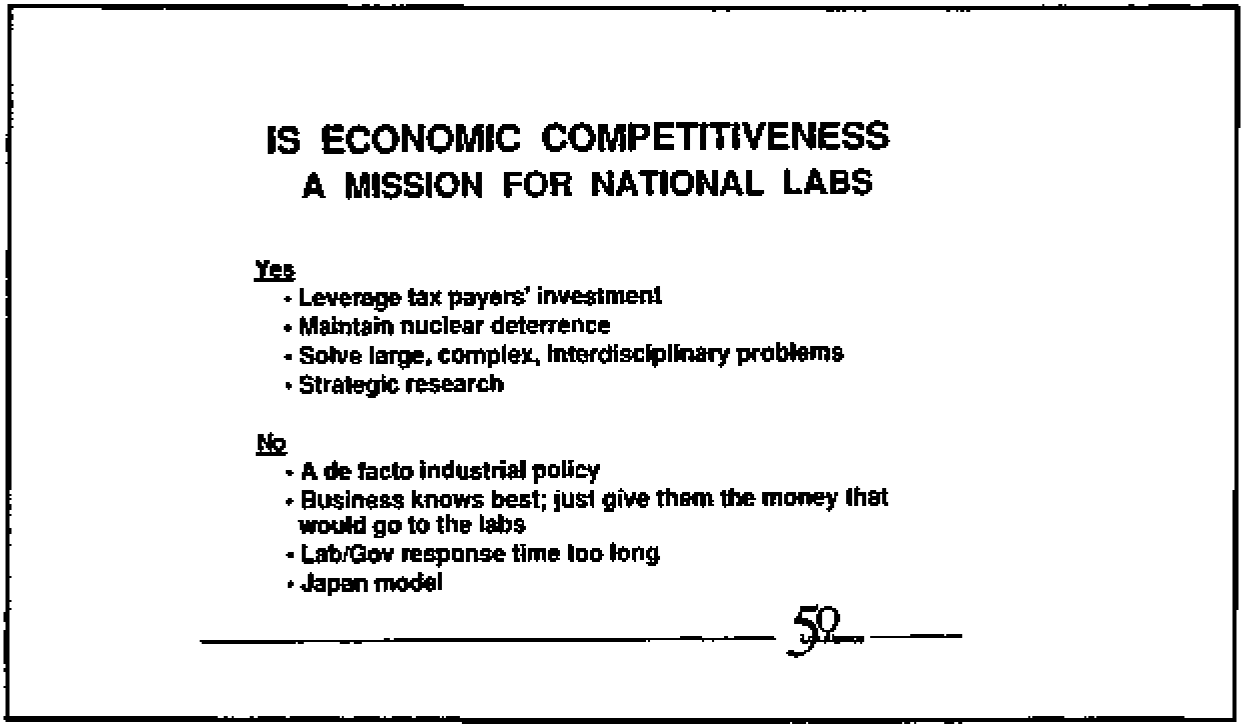
UNDERLYING ISSUES

- Defense conversion
- National economic competitiveness
- Nuclear competence
- Competition within industries
- Industrial policy
- Cost sharing
- Staying power of the Labs
- Identification and selection of areas of cooperation



LOS ALAMOS INDUSTRIAL COLLABORATIONS

- 85 CRADAs
  - Approximately $200m
  - 1/4 small businesses
- 45 Licenses
  - 3/4 small businesses
- 38 Companies started using LANL developed technologies
  Examples: Heat pipes
  Laser - based cell sorting
  Side - coupled cavity accelerator
- $1B industrial revenues



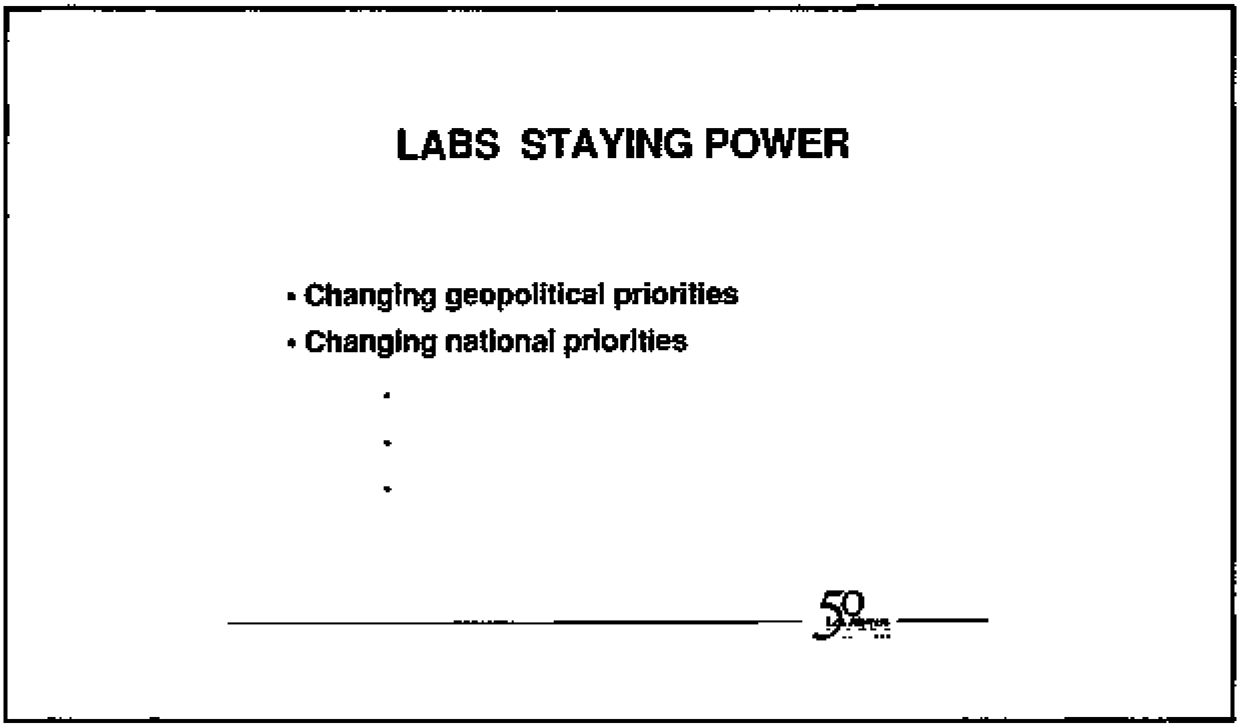
IS ECONOMIC COMPETITIVENESS A MISSION FOR NATIONAL LABS

<u>Yes</u>

- Leverage tax payers' investment
- Maintain nuclear deterrence
- Solve large, complex, interdisciplinary problems
- Strategic research

<u>No</u>

- A de facto industrial policy
- Business knows best; just give them the money that would go to the labs
- Lab/Gov response time too long
- Japan model



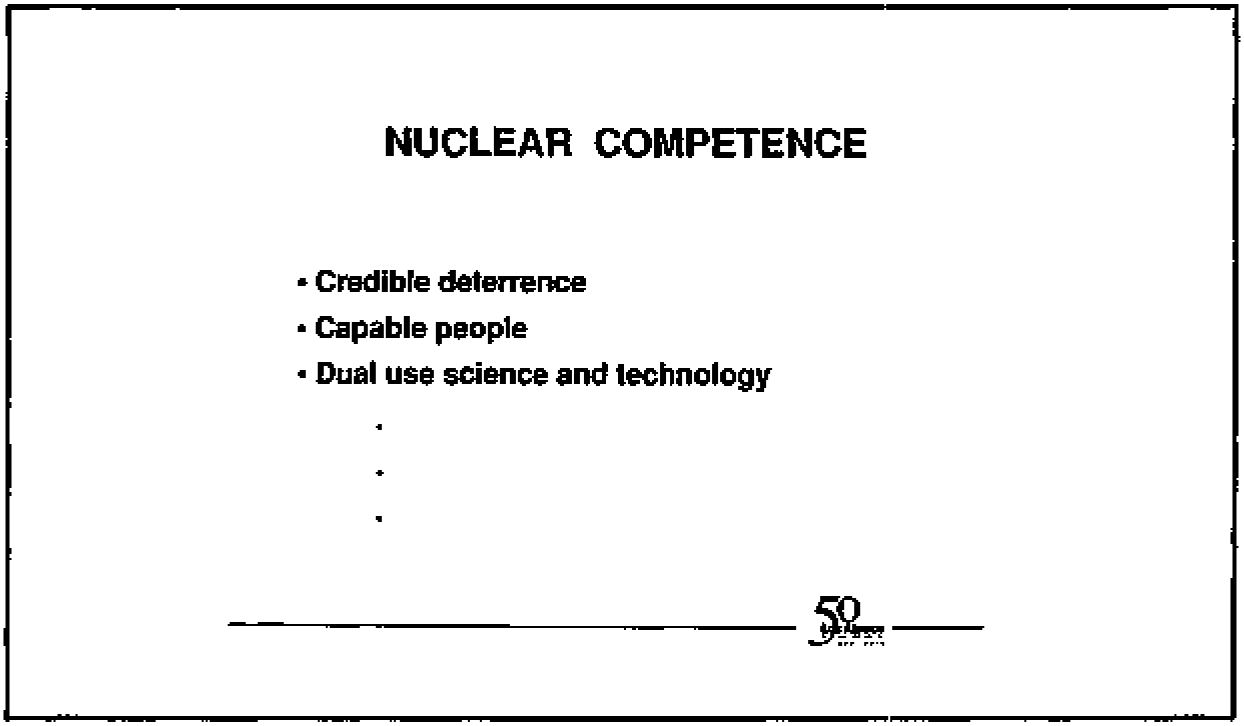
LABS STAYING POWER

- Changing geopolitical priorities
- Changing national priorities

•

•

•



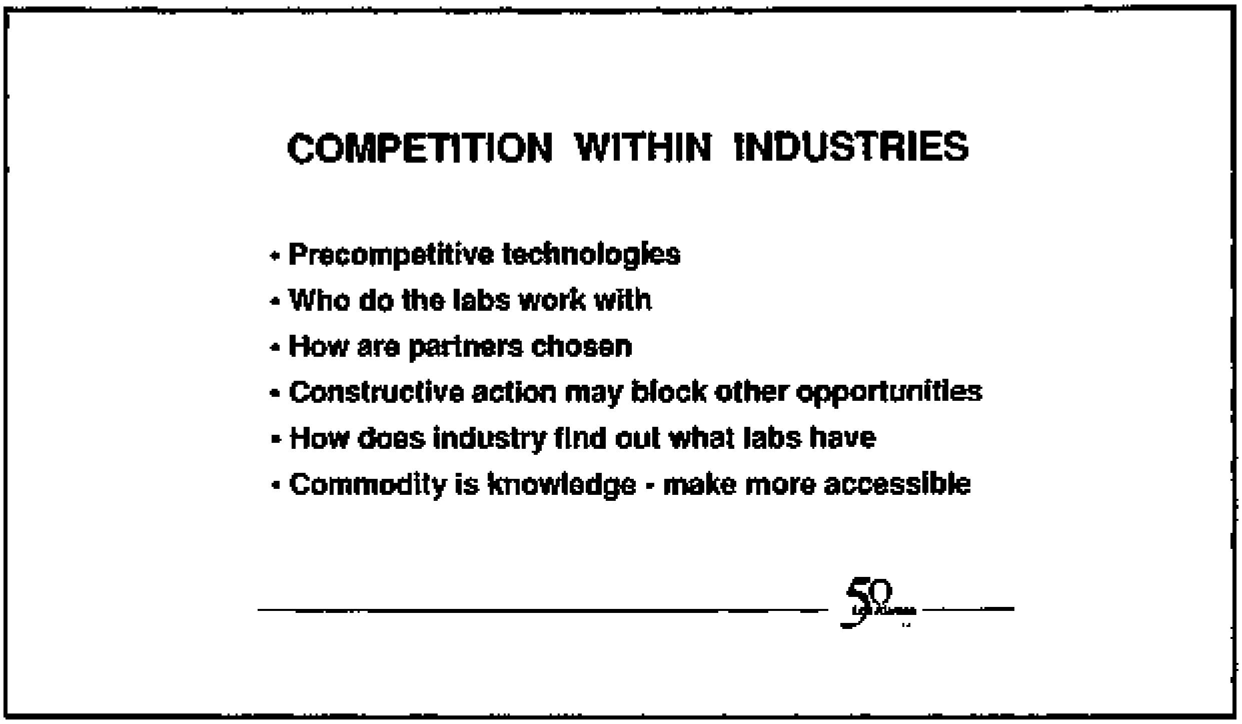
NUCLEAR COMPETENCE

- Credible deterrence
- Capable people
- Dual use science and technology

•

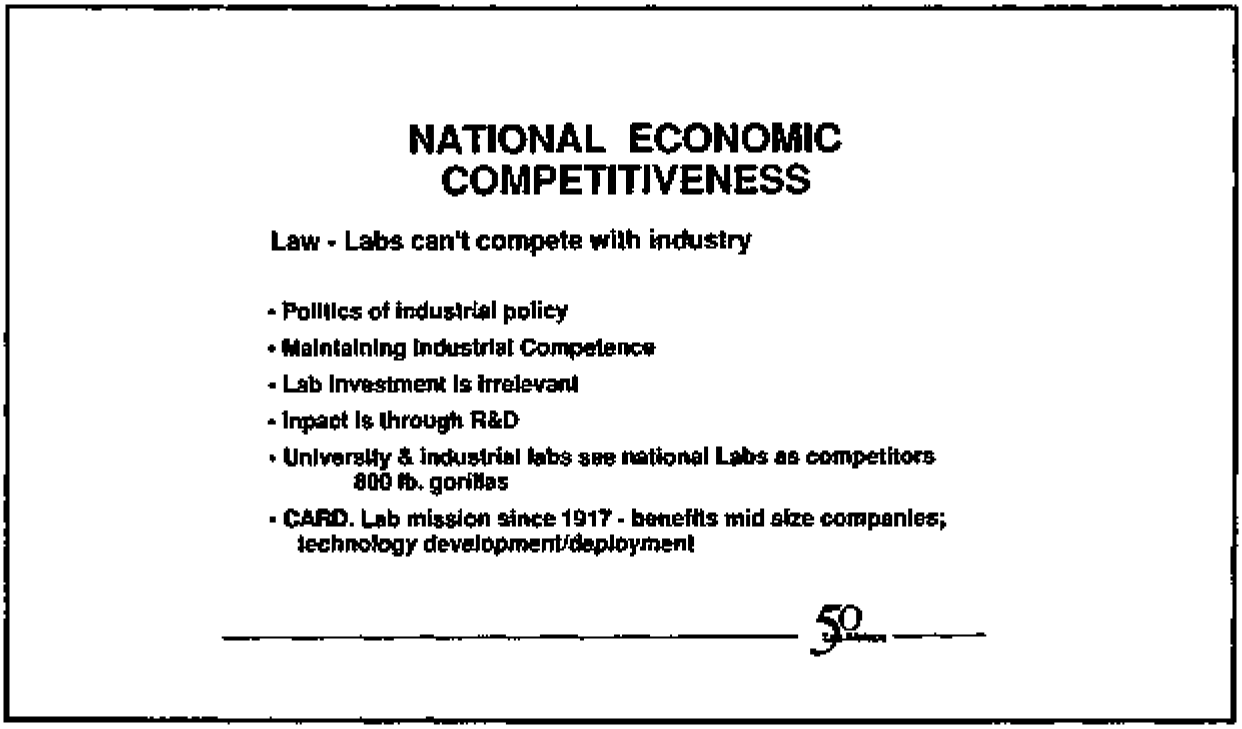
•

•



COMPETITION WITHIN INDUSTRIES

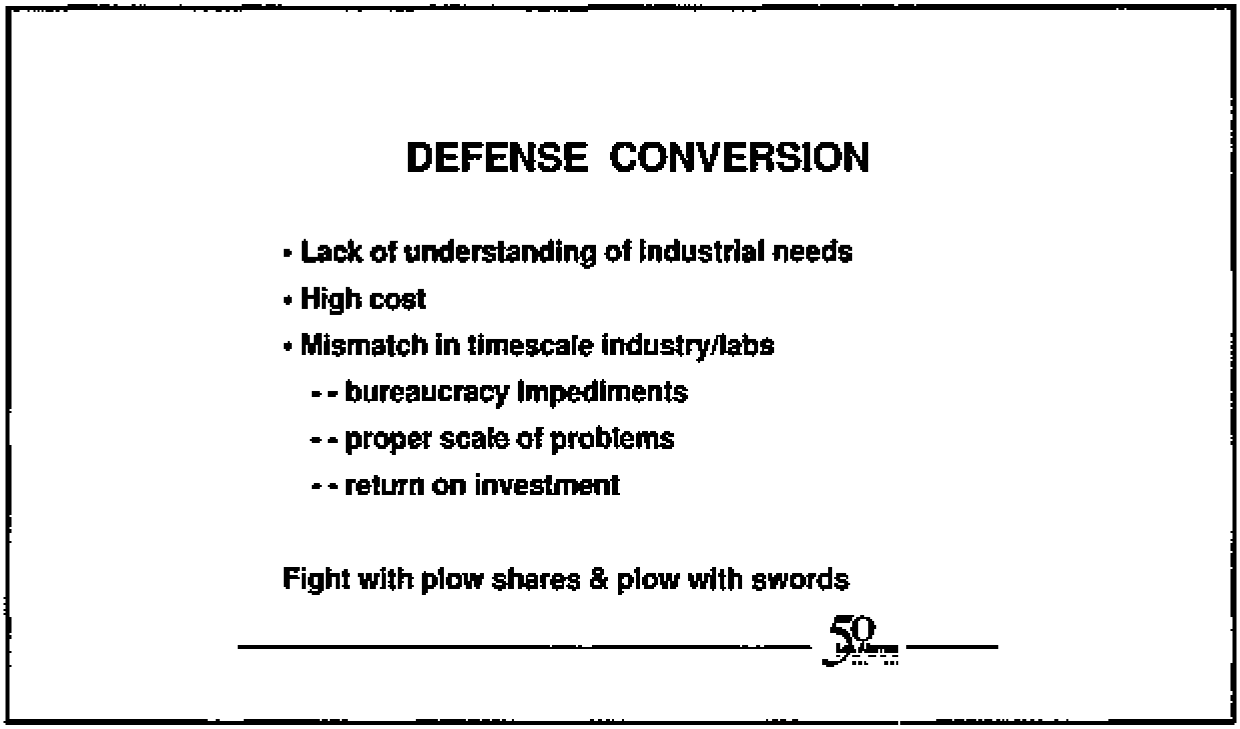
- Precompetitive technologies
- Who do the labs work with
- How are partners chosen
- Constructive action may block other opportunities
- How does industry find out what labs have
- Commodity is knowledge - make more accessible



NATIONAL ECONOMIC COMPETITIVENESS

Law - Labs can't compete with industry

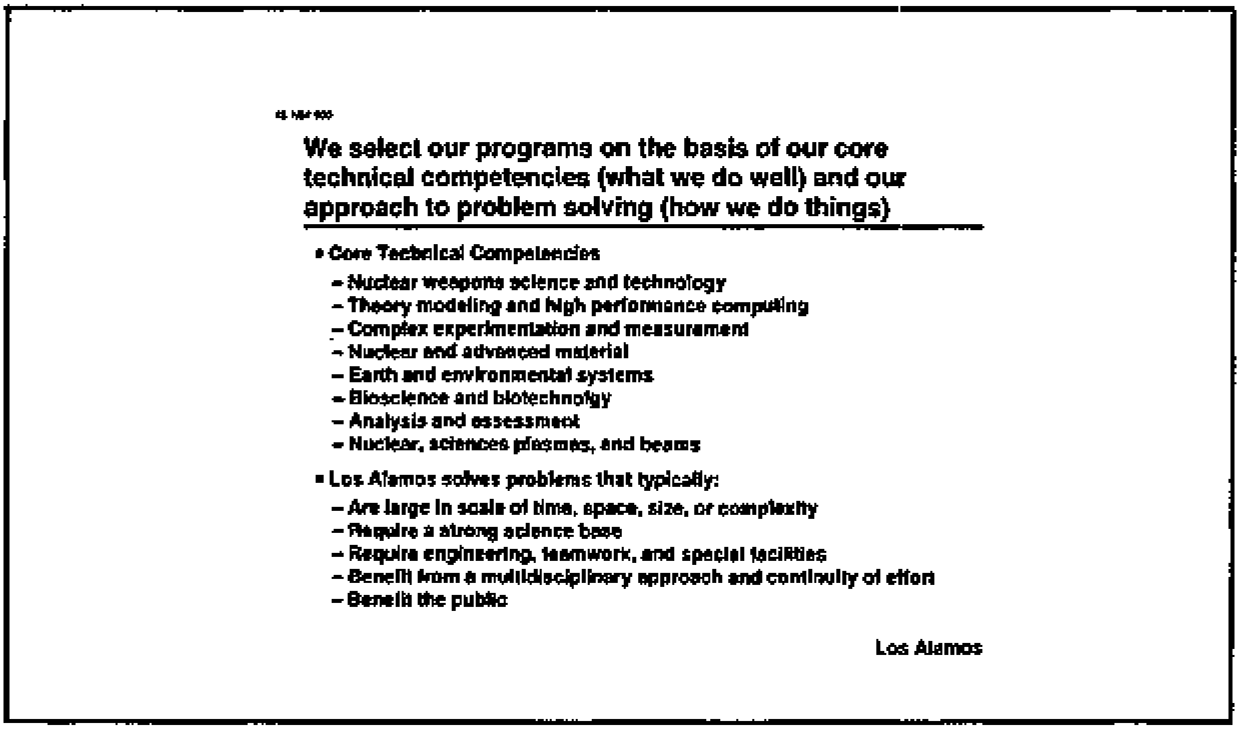
- Politics of industrial policy
- Maintaining Industrial Competence
- Lab Investment Is irrelevant
- Inpact Is through R&D
- University & industrial labs see national Labs as competitors
  800 lb. gorillas
- CARD. Lab mission since 1917 - benefits mid size companies;
  technology development/deployment



DEFENSE CONVERSION

- Lack of understanding of industrial needs
- High cost
- Mismatch in timescale industry/labs
  - -- bureaucracy impediments
  - -- proper scale of problems
  - -- return on investment

Fight with plow shares & plow with swords



We select our programs on the basis of our core technical competencies (what we do well) and our approach to problem solving (how we do things)

- Core Technical Competencies
  - Nuclear weapons science and technology
  - Theory modeling and high performance computing
  - Complex experimentation and measurement
  - Nuclear and advanced material
  - Earth and environmental systems
  - Bioscience and biotechnolgy
  - Analysis and assessment
  - Nuclear, sciences plasmas, and beams
- Los Alamos solves problems that typically:
  - Are large in scale of time, space, size, or complexity
  - Require a strong science base
  - Require engineering, teamwork, and special facilities
  - Benefit from a multidisciplinary approach and continuity of effort
  - Benefit the public

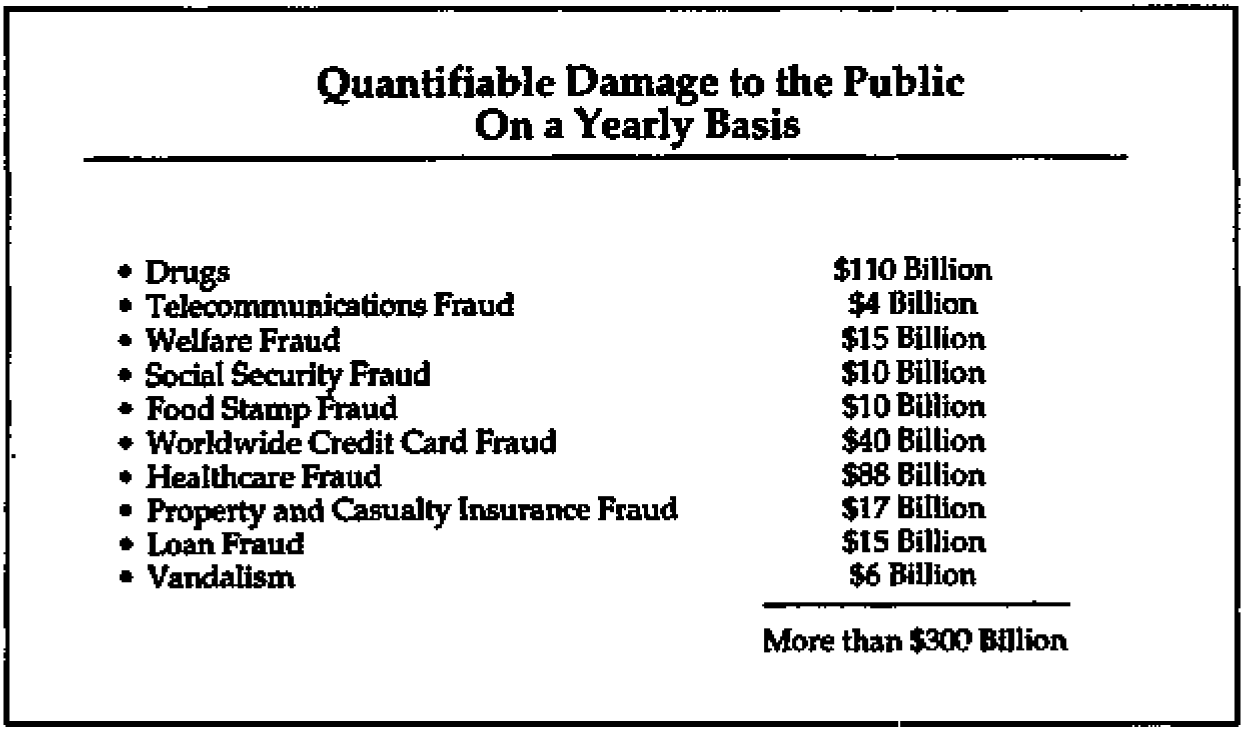
Los Alamos

MISSION

The Los Alamos National Laboratory is dedicated to developing world-class science and technology and applying them to the nation's security and well-being. The Laboratory will continue its special role in defense, particularly in nuclear weapons technology, and will increasingly civilian problems.

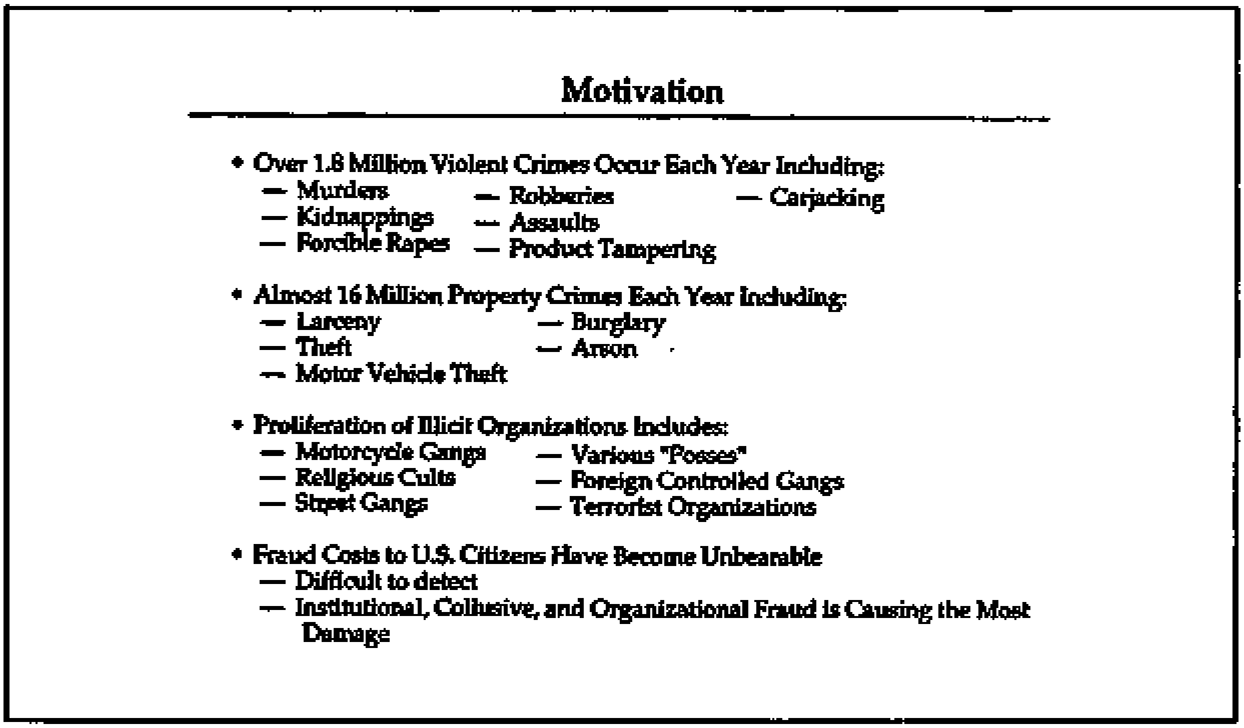
CCC:
CRIMINALS CAUGHT BY COMPUTING

Tom Kraay, Booz, Allen and Hamilton, Inc.



Quantifiable Damage to the Public On a Yearly Basis

• Drugs	$110 Billion
• Telecommunications Fraud	$4 Billion
• Welfare Fraud	$15 Billion
• Social Security Fraud	$10 Billion
• Food Stamp Fraud	$10 Billion
• Worldwide Credit Card Fraud	$40 Billion
• Healthcare Fraud	$88 Billion
• Property and Casualty Insurance Fraud	$17 Billion
• Loan Fraud	$15 Billion
• Vandalism	$6 Billion
	More than $300 Billion



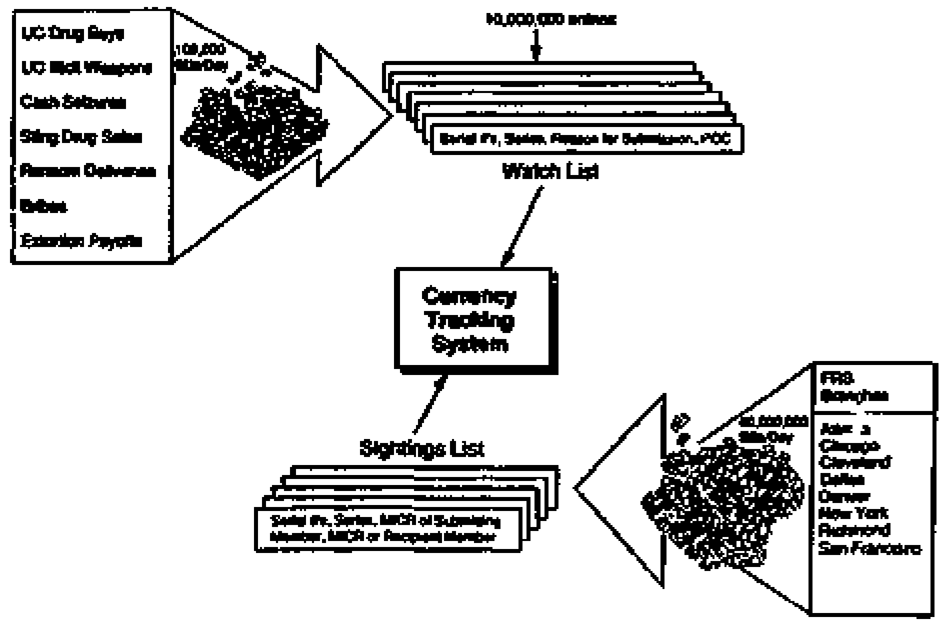
Motivation

- Over 1.8 Million Violent Crimes Occur Each Year Including:
  - Murders
  - Kidnappings
  - Forcible Rapes
  - Robberies
  - Assaults
  - Product Tampering
  - Carjacking
- Almost 16 Million Property Crimes Each Year Including:
  - Larceny
  - Theft
  - Motor Vehicle Theft
  - Burglary
  - Arson
- Proliferation of Illicit Organizations Includes:
  - Motorcycle Gangs
  - Religious Cults
  - Street Gangs
  - Various "Posses"
  - Foreign Controlled Gangs
  - Terrorist Organizations
- Fraud Costs to U.S. Citizens Have Become Unbearable
  - Difficult to detect
  - Institutional, Collusive, and Organizational Fraud is Causing the Most Damage

Unquantifiable Damage to the Public

- Death
- Disability
- Sorrow
- Grief
- Fear of Venturing Out
- Frustration
- Embarassment
- Quality of Life
- International Perception of U.S. Society

MPP-Based Currency Tracking System



Algebraically Speaking ...

$$\sum_{i=0}^{n-1} c_i^2 = 1$$

$$\sum_{i=0}^{\frac{n}{2}-1} c_{2i} - \sum_{i=0}^{\frac{n}{2}-1} c_{2i+1} = 0$$

$$\sum_{i=0}^{n-1-2j} c_i c_{i+2j} = 0 \text{ for } j = 1,2,\ldots,\frac{n}{2}-1$$

Clumsy System to Solve Because There Exist only $\frac{n}{2}+1$ equations in n Unknowns

To Simplify a Solution ...

Introduce Additional Equations:

$$0 \bullet C_{n-1} - 1 \bullet C_{n-2} + 2\, C_{n-3} - \ldots - (n-1)\, C_0 = 0$$

$$0^2 \bullet C_{n-1} - 1^2 \bullet C_{n-2} + 2^2 \bullet C_{n-3} - \ldots - (n-1)^2 C_0 = 0$$

$$\vdots$$

$$0^{\frac{n}{2}-1} \bullet C_{n-1} - 1^{\frac{n}{2}-1} \bullet C_{n-2} + 2^{\frac{n}{2}-1} \bullet C_{n-3} - \ldots - (n-1)^{\frac{n}{2}-1} \bullet C_0 = 0$$

The Four Coefficient Transformation Matrix

$$\begin{bmatrix}
C_0 & C_1 & C_2 & C_3 & & & & & & & \\
C_3 & -C_2 & C_1 & -C_0 & & & & & & & \\
 & & C_0 & C_1 & C_2 & C_3 & & & & & \\
 & & C_3 & -C_2 & C_1 & -C_0 & & & & & \\
\vdots & \vdots & & & & & \ddots & & & & \\
 & & & & & & & C_0 & C_1 & C_2 & C_3 \\
 & & & & & & & C_3 & -C_2 & C_1 & -C_0 \\
C_2 & C_3 & & & & & & & & C_0 & C_1 \\
C_1 & -C_0 & & & & & & & & C_3 & -C_2
\end{bmatrix}$$

Choose the Ci's So That the Transpose of the Matrix is its Inverse:

$$\begin{bmatrix}
C_0 & C_3 & & & \cdots & & & & & C_2 & C_1 \\
C_1 & -C_2 & & & \cdots & & & & & C_3 & -C_0 \\
C_2 & C_1 & C_0 & C_3 & & & & & & & \\
C_3 & -C_0 & C_1 & -C_2 & & & & & & & \\
\vdots & \vdots & & & \ddots & & & & & & \\
 & & & & & C_2 & C_1 & C_0 & C_3 & & \\
 & & & & & C_3 & -C_0 & C_1 & -C_2 & & \\
 & & & & & & & C_2 & C_1 & C_0 & C_3 \\
 & & & & & & & C_3 & -C_0 & C_1 & -C_2
\end{bmatrix}$$

Solutions to This System Lead to the Famous Daubechies Systems

DAUB2: $C_0 = C_1 = \sqrt{\frac{1}{2}}$

DAUB4: $C_0 = \left(1+\sqrt{3}\right)/4\sqrt{2} \quad C_1 = \left(3+\sqrt{3}\right)/4\sqrt{2}$
$C_2 = \left(3-\sqrt{3}\right)/4\sqrt{2} \quad C_3 = \left(1-\sqrt{3}\right)/4\sqrt{2}$

DAUB6: $C_0 = \left(1+\sqrt{10}+\sqrt{5+2\sqrt{10}}\right)/16\sqrt{2} \quad C_1 = \left(5+\sqrt{10}+3\sqrt{5+2\sqrt{10}}\right)/16\sqrt{2}$
$C_2 = \left(10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}}\right)/16\sqrt{2} \quad C_3 = \left(10-2\sqrt{10}-2\sqrt{5+2\sqrt{10}}\right)/16\sqrt{2}$
$C_4 = \left(5+\sqrt{10}-3\sqrt{5+2\sqrt{10}}\right)/16\sqrt{2} \quad C_5 = \left(1+\sqrt{10}\right)-\sqrt{5+2\sqrt{10}}\ /16\sqrt{2}$
$\vdots$

Harnassing More Coefficient Solutions

2 Coefficients: $c_0 = \sqrt{\frac{1}{2}}$

$c_1 = \sqrt{\frac{1}{2}}$

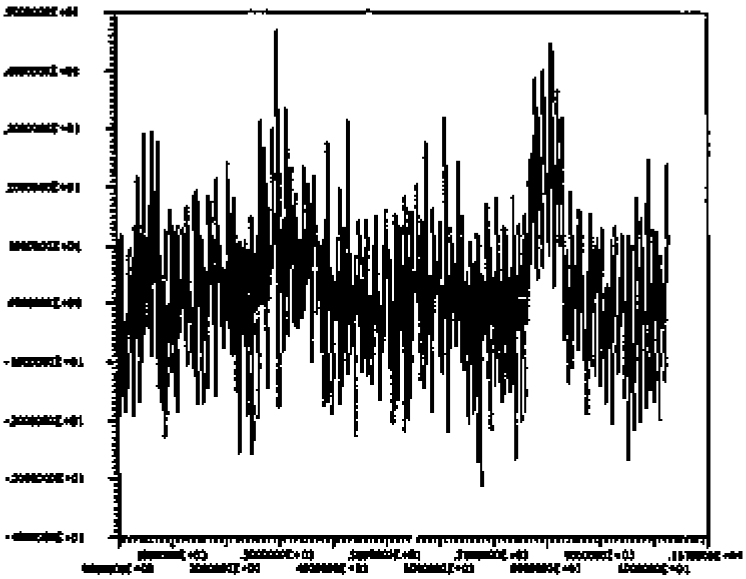
4 Coefficients:

$$\left.\begin{aligned} c_0 &= \frac{1}{2}\sqrt{\frac{1}{2}} + \frac{1}{2}\sin\alpha \\ c_1 &= \frac{1}{2}\sqrt{\frac{1}{2}} + \frac{1}{2}\cos\alpha \\ c_2 &= \frac{1}{2}\sqrt{\frac{1}{2}} - \frac{1}{2}\sin\alpha \\ c_3 &= \frac{1}{2}\sqrt{\frac{1}{2}} - \frac{1}{2}\cos\alpha \end{aligned}\right|\ \alpha = \pi/4 \Rightarrow 2 \text{ Coefficient Wavelet}$$

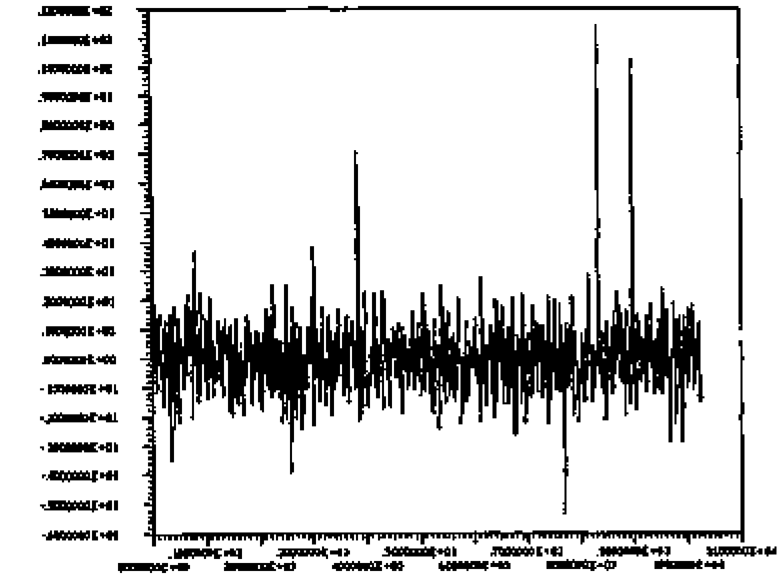
6 Coefficients:

$$\left.\begin{aligned} c_0 &= \frac{1}{4}\sqrt{\frac{1}{2}} - \frac{1}{4}\sin\alpha + \frac{1}{2}\cos\beta\sqrt{\frac{1-\sqrt{\frac{1}{2}}(\sin\alpha+\cos\alpha)}{2}} \\ c_1 &= \frac{1}{4}\sqrt{\frac{1}{2}} - \frac{1}{4}\cos\alpha + \frac{1}{2}\sin\beta\sqrt{\frac{1-\sqrt{\frac{1}{2}}(\sin\alpha+\cos\alpha)}{2}} \\ c_2 &= \frac{1}{2}\sqrt{\frac{1}{2}} + \frac{1}{2}\sin\alpha \\ c_3 &= \frac{1}{2}\sqrt{\frac{1}{2}} + \frac{1}{2}\cos\alpha \\ c_4 &= \frac{1}{4}\sqrt{\frac{1}{2}} - \frac{1}{4}\sin\alpha - \frac{1}{2}\sin\beta\sqrt{\frac{1-\sqrt{\frac{1}{2}}(\sin\alpha+\cos\alpha)}{2}} \\ c_5 &= \frac{1}{4}\sqrt{\frac{1}{2}} - \frac{1}{4}\cos\alpha - \frac{1}{2}\cos\beta\sqrt{\frac{1-\sqrt{\frac{1}{2}}(\sin\alpha+\cos\alpha)}{2}} \end{aligned}\right|\ 2\beta = \alpha + \frac{5\pi}{4} \Rightarrow 4 \text{ Coefficient Wavelets},\ \alpha = \pi/4 \Rightarrow 2 \text{ Coefficient Wavelets}$$

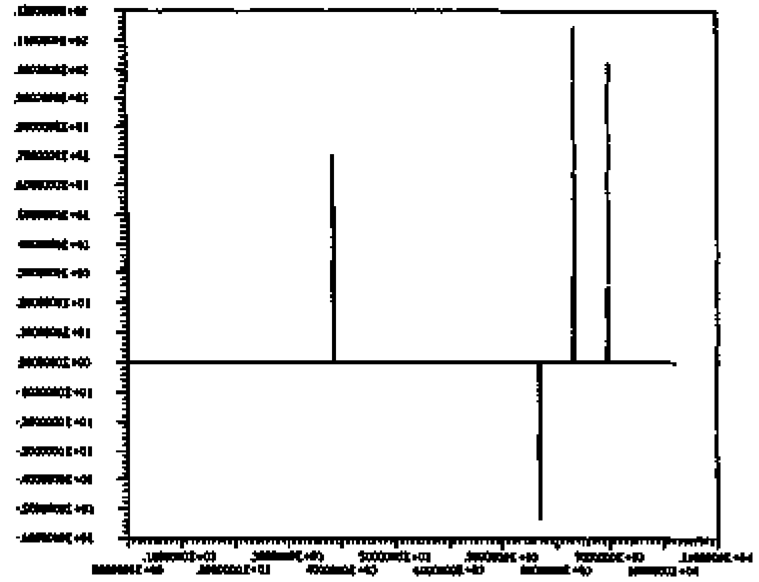
Raw Accoustic Data



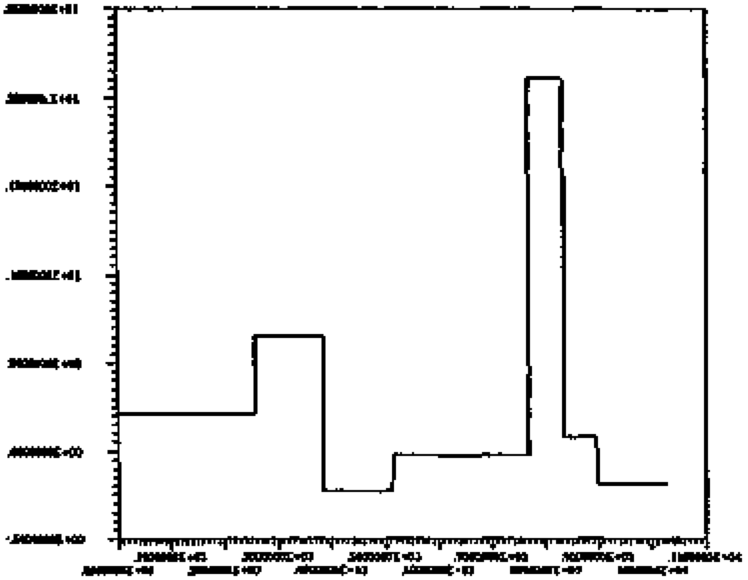
Haar Transform



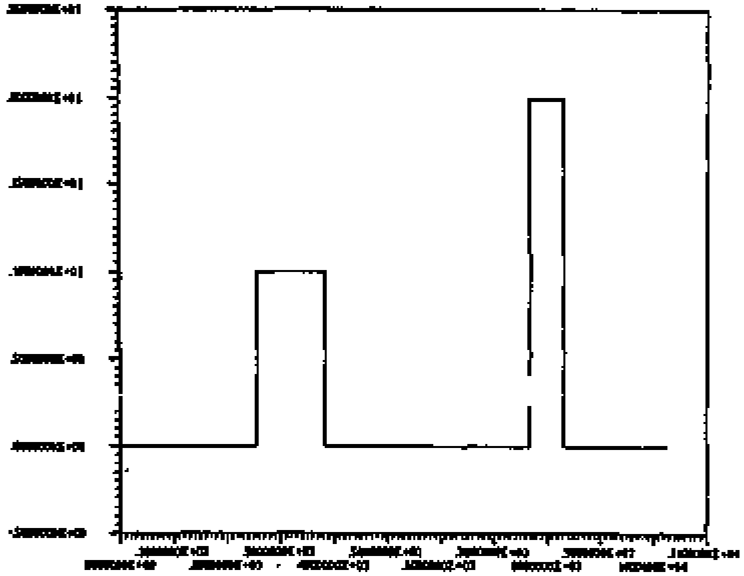
Filtered Haar Transform

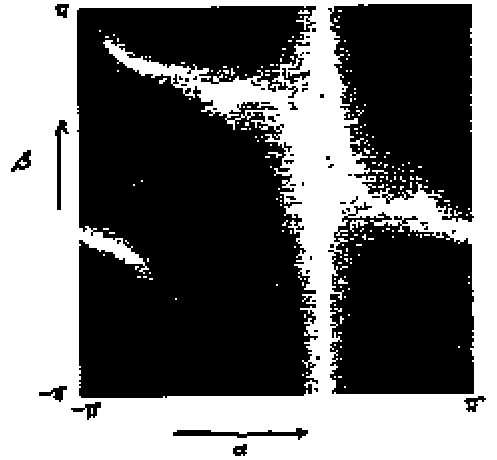


Filtered Pulses in Time Domain



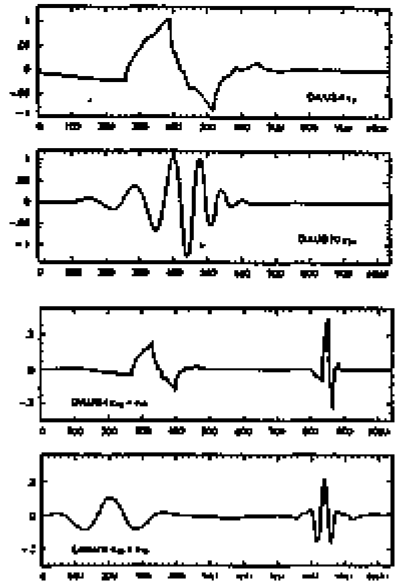
Ground Truth



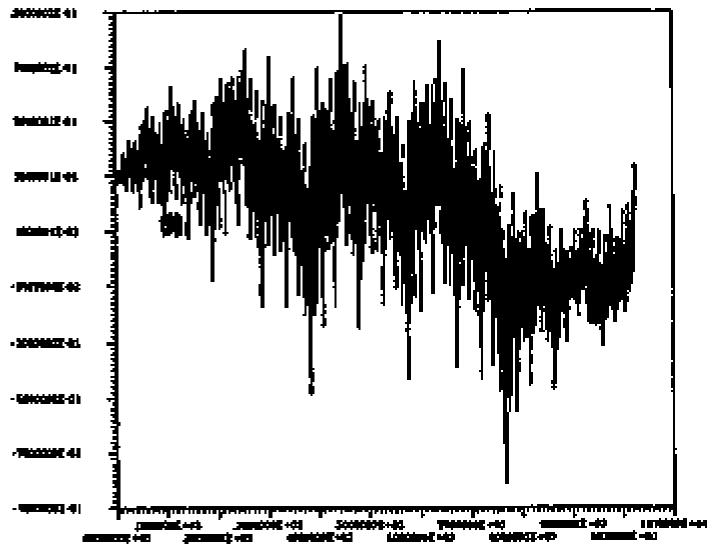


White = Nearly Continuous
⋮
Black = Extremely Discontinuous

Examples of Wavelets



A Discontinuous Wavelet



BEYOND LITHOGRAPHY:
Molecular Manufacturing and the Future of Computing

Ralph Merkle, Xerox PARC

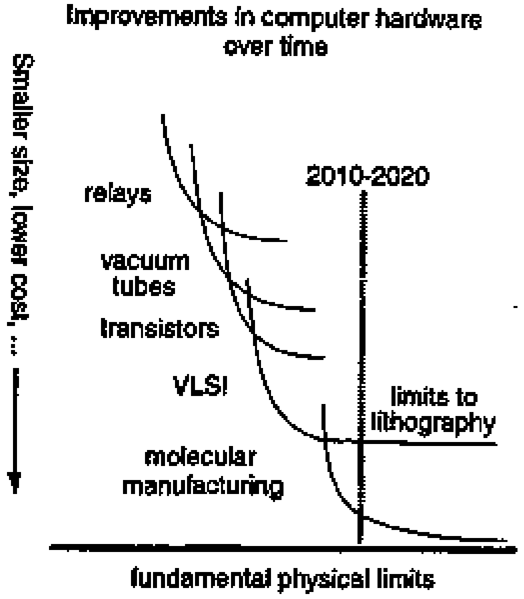
Nanosystems:
Molecular Machinery,
Manufacturing,
and Computation

by

K. Eric Drexler
Wiley 1992

Association of American Publishers Prize for Best Computer Science Book of 1992

Second printing planned



Improvements in computer hardware over time

Nanosystems:
Molecular Machinery,
Manufacturing,
and Computation

by

K. Eric Drexler
Wiley 1992

Association of American Publishers
Prize for Best Computer Science Book of
1992

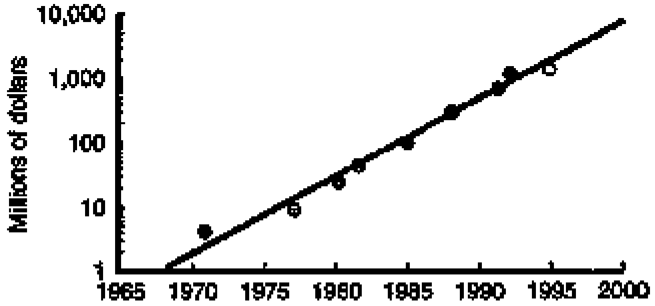
Second printing planned

First printing:
12,000 paperback

Trends in computer hardware suggest that between 2010 and 2020 we will develop:

1. Mass memory that stores one bit per atom
2. Energy dissipation per logic operation of kT for T = 300 Kelvins (thermal noise at room temperature)
3. Logic elements with only a few dopant atoms each
4. Manufacturing resolutions of an atomic diameter

Escalating Cost of Manufacturing Facilities



(From Electronics News, September 27 1993, page 28)

Using molecular-beam epitaxy or lithography for nanoelectronics seems increasingly like making a suspension bridge by carving it out of a large block of steel.

John Hopfield, Caltech, 1992

What would happen if we could arrange the atoms one by one the way we want them (within reason, of course; you can't put them so that they are chemically unstable, for example).

Richard P. Feynman, 1959
Nobel Prize for Physics, 1965

Molecular manufacturing

1. **Almost every atom in the right place**
2. **Manufacturing costs not greatly exceeding the cost of the required raw materials and energy**
3. **Able to make almost any structure consistent with physical and chemical law**

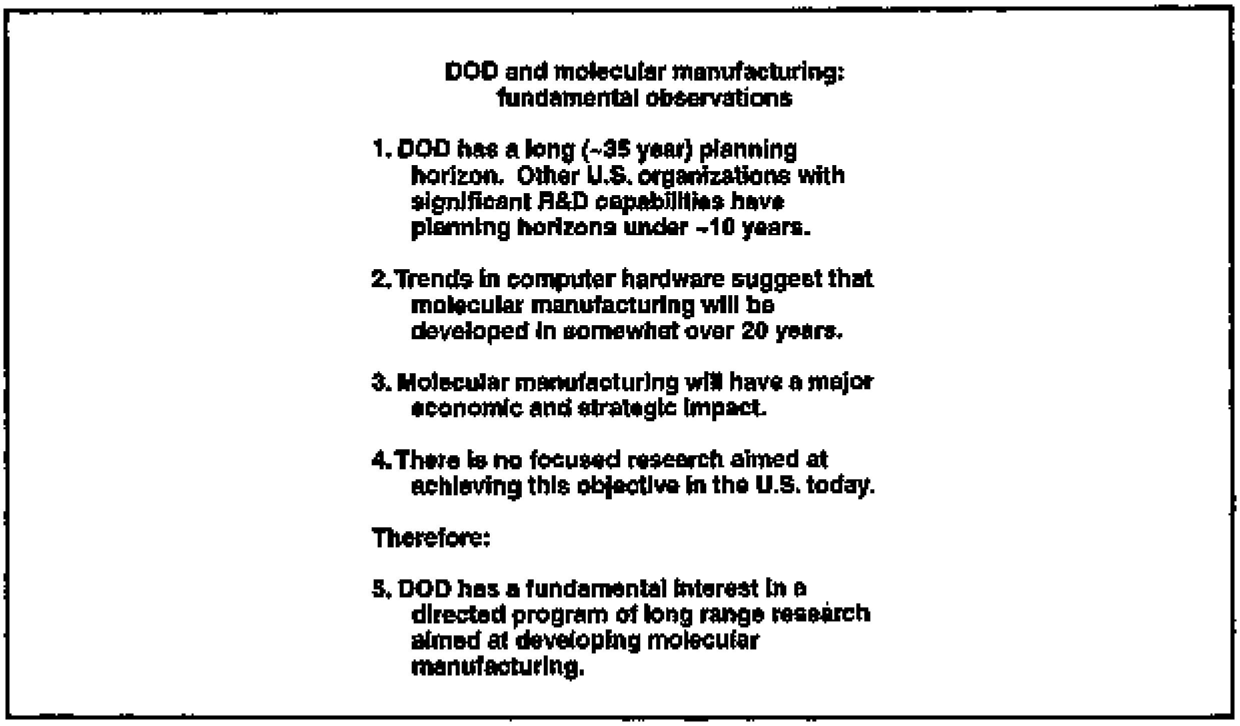
"While speaking to a group of senior naval officers last week, I stressed the need to invest in nanotechnology."

"We want R&D in things like nanotechnology to continue to keep us ahead of potential enemies."

Admiral David E. Jeremiah, USN
Vice Chairman, Joint Chiefs of Staff
February 11, 1992

"More specifically, given severe budget constraints, an emphasis toward long-term research makes most sense. In the short and medium term, fairly modest efforts will suffice to maintain our lead in defense technologies. But over the longer term, a more coherent program is needed."

Project 2025 report
November 6, 1991



DOD and molecular manufacturing: fundamental observations

1. DOD has a long (~35 year) planning horizon. Other U.S. organizations with significant R&D capabilities have planning horizons under ~10 years.

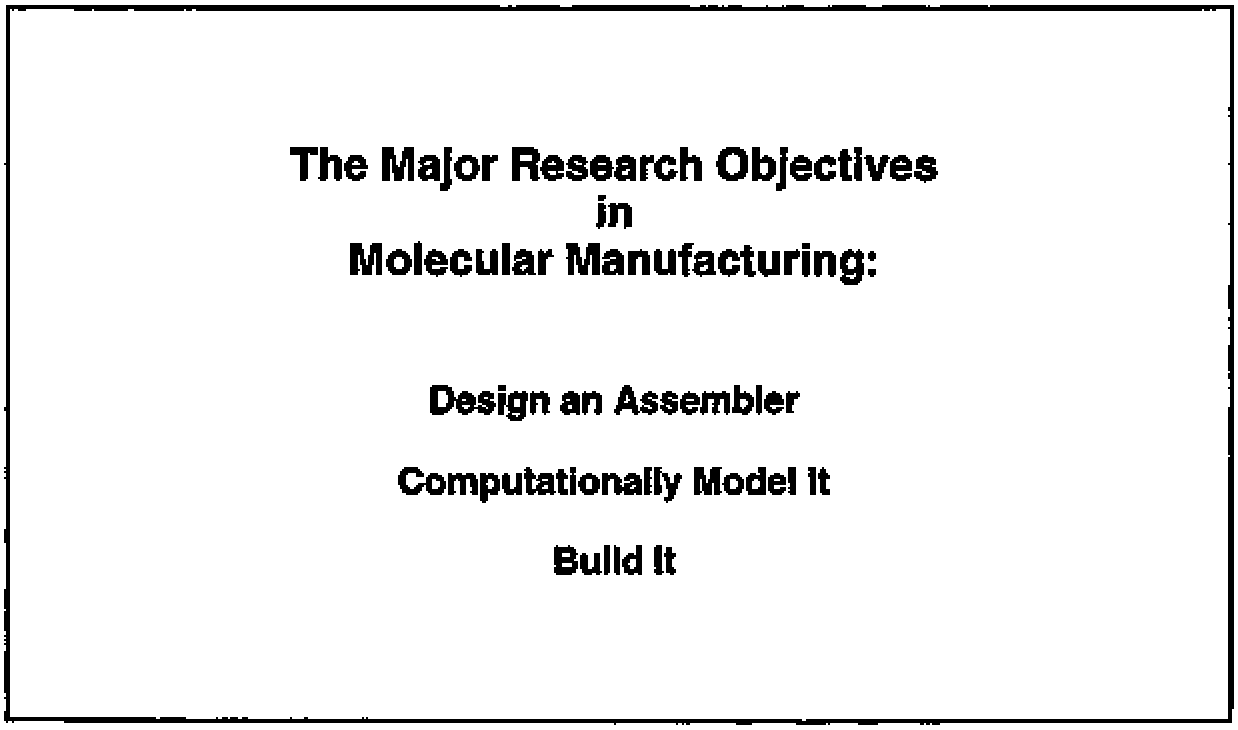
2. Trends in computer hardware suggest that molecular manufacturing will be developed in somewhat over 20 years.

3. Molecular manufacturing will have a major economic and strategic impact.

4. There is no focused research aimed at achieving this objective in the U.S. today.

Therefore:

5. DOD has a fundamental interest in a directed program of long range research aimed at developing molecular manufacturing.



The Major Research Objectives in Molecular Manufacturing:

Design an Assembler

Computationally Model It

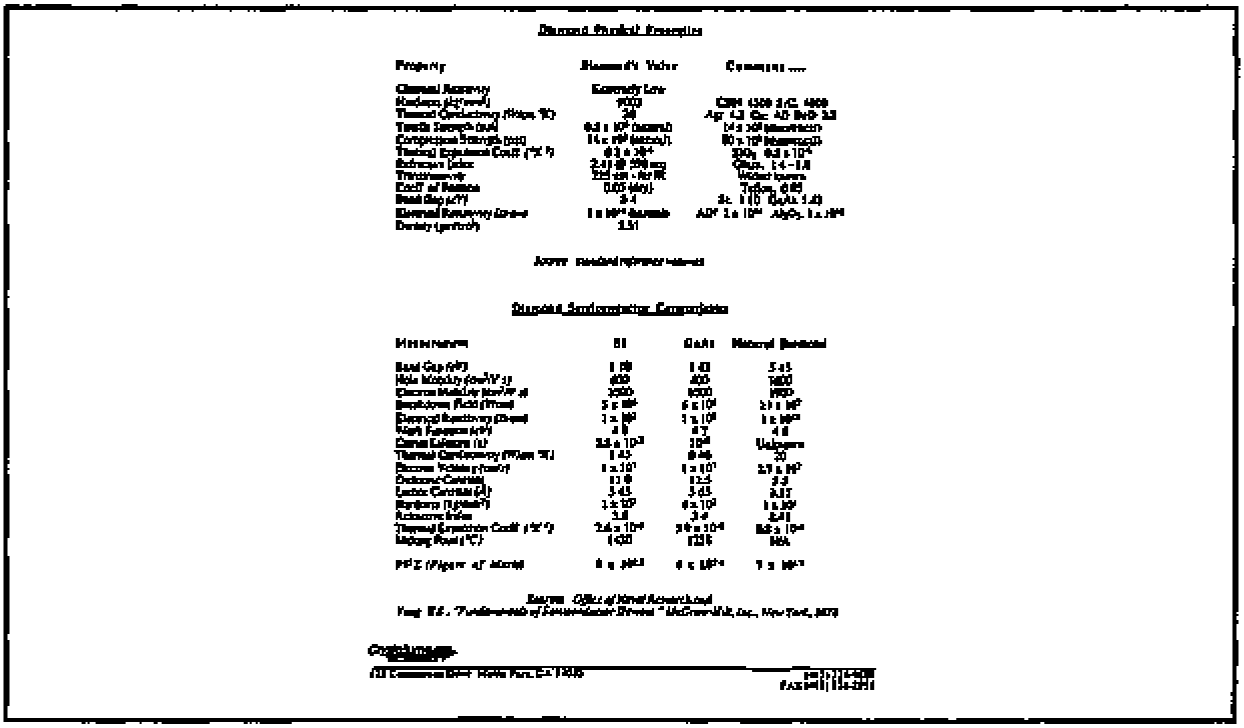
Build It

Computational Experiments can be used to:

Design and Model Long Term Goals (diamondoid systems)

Medium Term Goals (many possibilities)

Short Term Goals (aid existing experimental work)



Diamond Physical Properties

Property	Diamond's Value	Comments ....
Chemical Reactivity	Extremely Low	
Hardness (kg/mm²)	9000	CBN 4500 SiC 2500
Thermal Conductivity (W/cm °K)	20	Ag 4.3 Cu 4.0 BeO 2.2
Tensile Strength (psi)	0.5 x 10^6 (natural)	1.4 x 10^6 (theoretical)
Compressive Strength (psi)	1.5 x 10^6 (natural)	80 x 10^6 (theoretical)
Thermal Expansion Coeff (/°K)	0.8 x 10^{-6}	SiO_2 0.5 x 10^{-6}
Refractive Index	2.41 @ 590 nm	Glass 1.4 - 1.8
Transmissivity	225 nm - far IR	Widest known
Coeff. of Friction	0.05 (dry)	Teflon 0.05
Band Gap (eV)	5.4	Si 1.10 GaAs 1.43
Electrical Resistivity (Ω-cm)	1 x 10^{16} (natural)	AlN 1 x 10^{13} Al_2O_3 1 x 10^{14}
Density (gm/cm³)	3.51	

Bold = standard reference values

Diamond Semiconductor Comparisons

Measurement	Si	GaAs	Natural Diamond
Band Gap (eV)	1.10	1.43	5.45
Hole Mobility (cm²/V-s)	600	400	1600
Electron Mobility (cm²/V-s)	1500	8500	1900
Breakdown Field (V/cm)	5 x 10^5	6 x 10^5	1 x 10^7
Electrical Resistivity (Ω-cm)	1 x 10^3	1 x 10^8	1 x 10^{13}
Work Function (eV)	4.8	4.7	4.8
Carrier Lifetime (s)	2.5 x 10^{-3}	10^{-8}	Unknown
Thermal Conductivity (W/cm °K)	1.45	0.46	20
Electron Velocity (cm/s)	1 x 10^7	1 x 10^7	2.7 x 10^7
Dielectric Constant	11.8	12.5	5.5
Lattice Constant (Å)	5.43	5.65	3.57
Hardness (kg/mm²)	1 x 10^3	6 x 10^2	1 x 10^4
Refractive Index	3.5	3.4	2.41
Thermal Expansion Coeff (/°C)	2.6 x 10^{-6}	5.9 x 10^{-6}	0.8 x 10^{-6}
Melting Point (°C)	1420	1238	NA
JFM (Figure of Merit)	8 x 10^{23}	6 x 10^{24}	7 x 10^{27}

Source: Office of Naval Research and
Yang, E.S., "Fundamentals of Semiconductor Devices," McGraw-Hill, Inc., New York, 1978

Crystallume

125 Constitution Drive Menlo Park, CA 94025

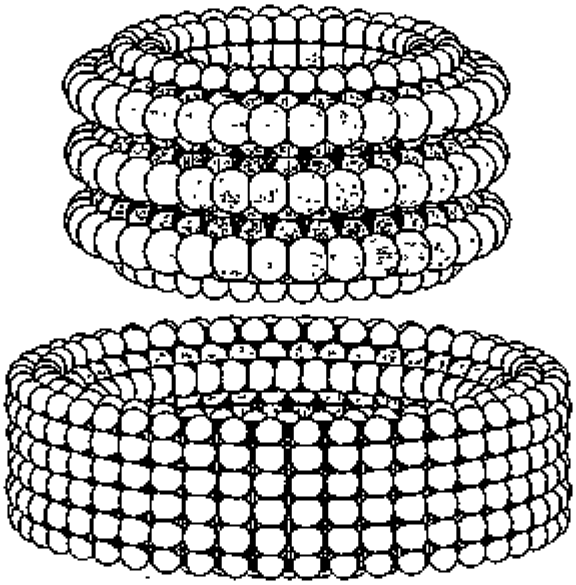
[illegible]
FAX [illegible]

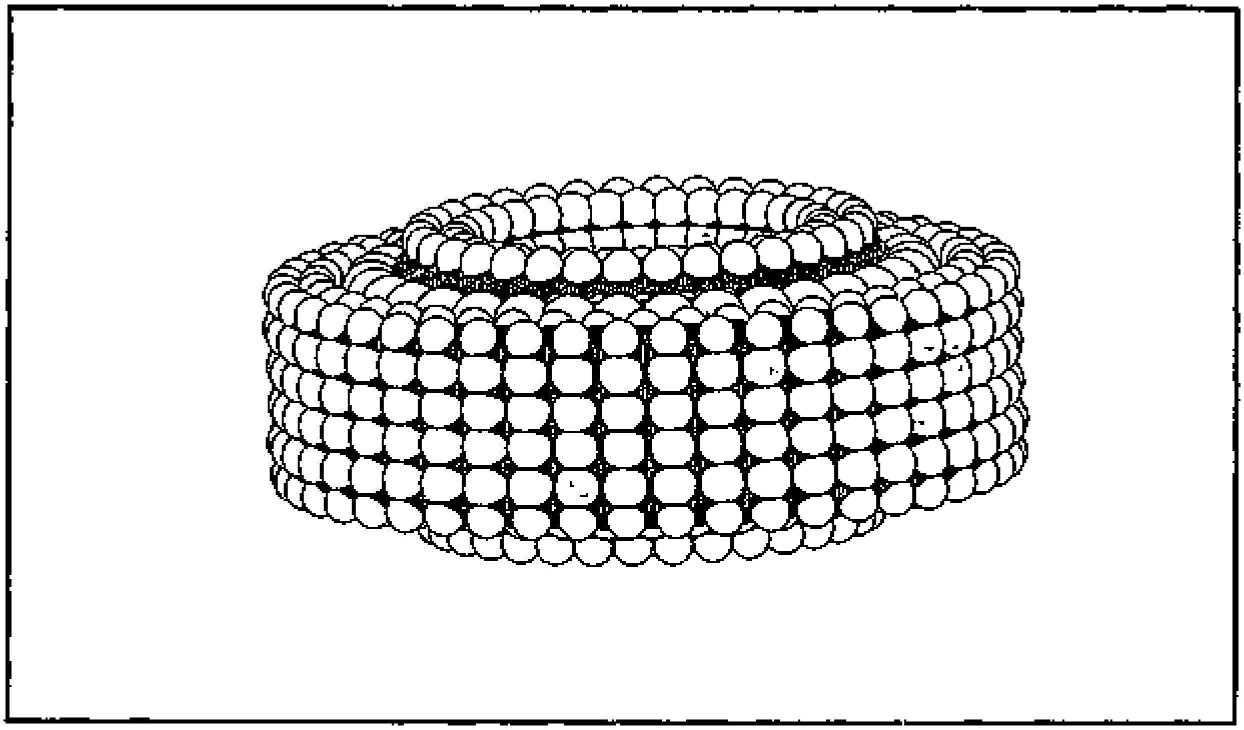
Molecular Manufacturing (slightly simplified)

1.) Inexpensive

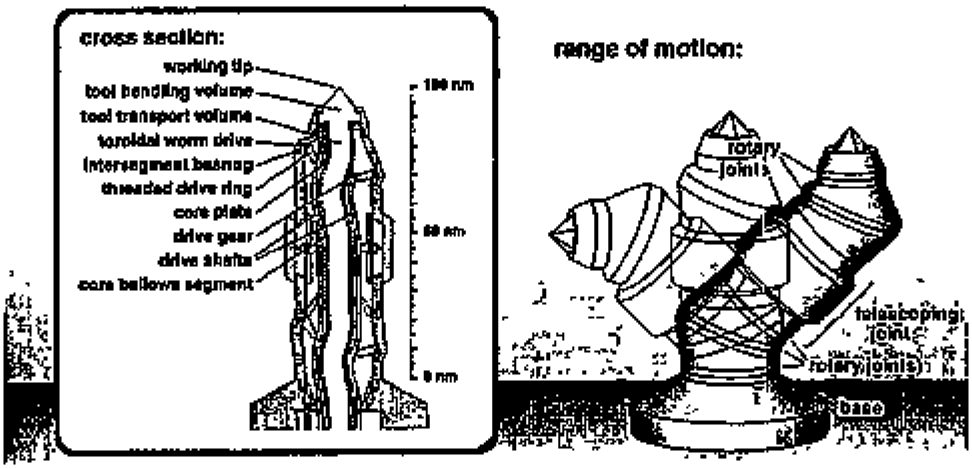
2.) Molecular precision (fewer than one atom in ten billion out of place in properly designed structures)

3.) Make almost any stiff diamondoid structure consistent with the laws of physics and chemistry





Nanomanipulator for molecular assembly:



- tubular diamondoid structure
- six drive shafts
- six-axis control
- ~ 10^{-6} seconds per motion
- > 25 N/m bending stiffness
- ~ 4,000,000 atoms

Synthesis of Diamond Today: Diamond CVD

1.) Carbon: Methane (ethane, acetylene...)

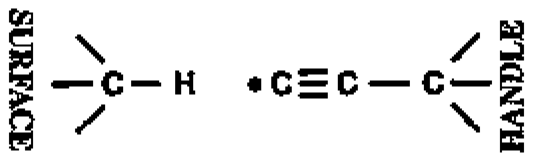
2.) Hydrogen: H_2

3.) Add Energy, producing CH_3, H, etc.

4.) Growth of a diamond film.

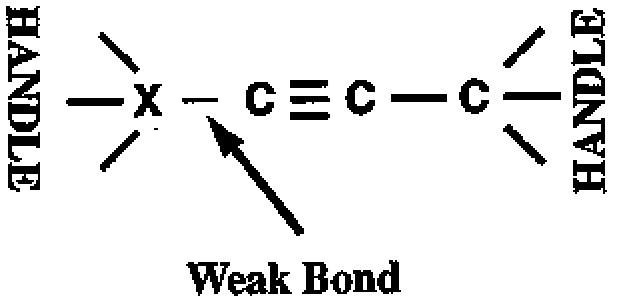
The right chemistry, but little control over the site of reactions or exactly what is synthesized.

A Site Specific Hydrogen Abstraction Tool



Theoretical studies of a hydrogen abstraction tool for nanotechnology, by Charles Musgrave et. al., *Nanotechnology* 2 (1991) pages 187-195.

Precursor to a Hydrogen Abstraction Tool



A Synthetic Strategy For the Synthesis Of Diamondoid Structures

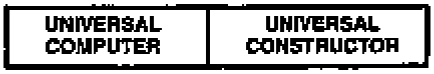
1.) Positional Control
(6 degrees of freedom)

2.) Highly Reactive Compounds
(radicals, carbenes, etc.)

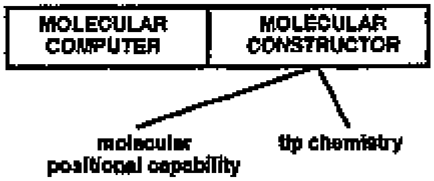
3.) Inert Environment
(vacuum, no side reactions)

A core concept: self replicating universal constructors

Von Neumann's architecture:

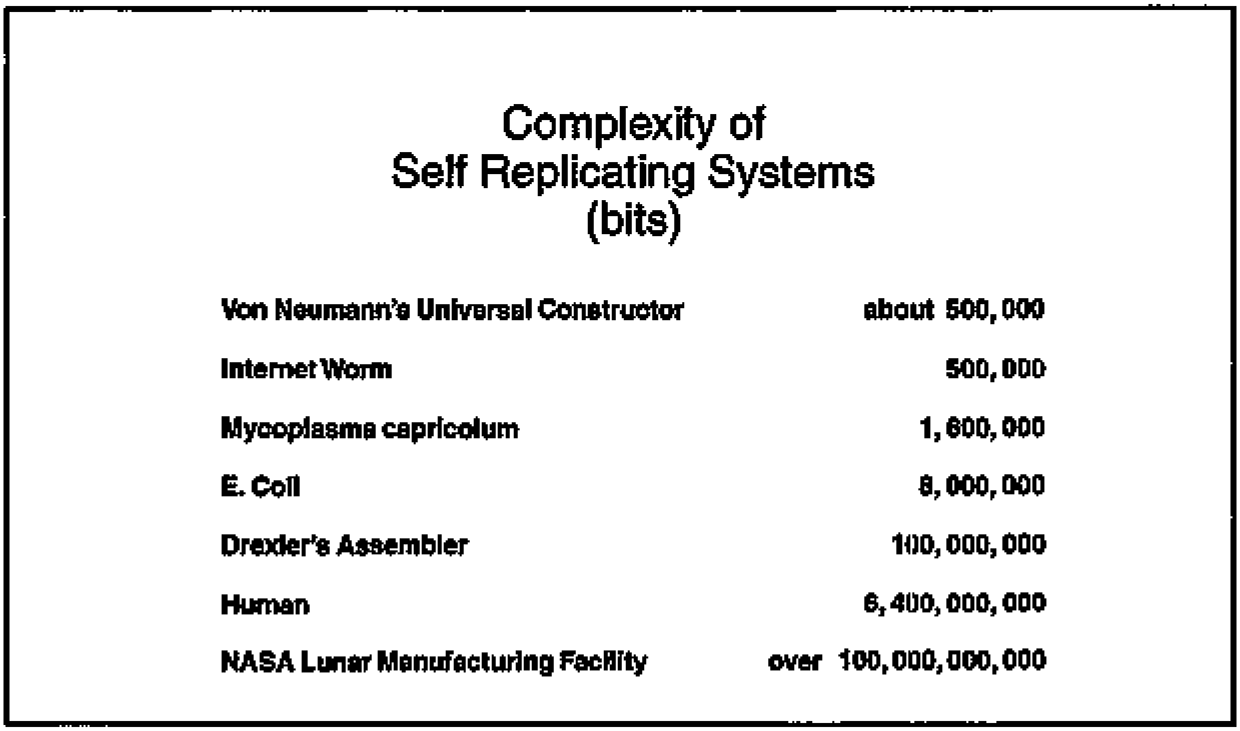


Drexler's architecture:



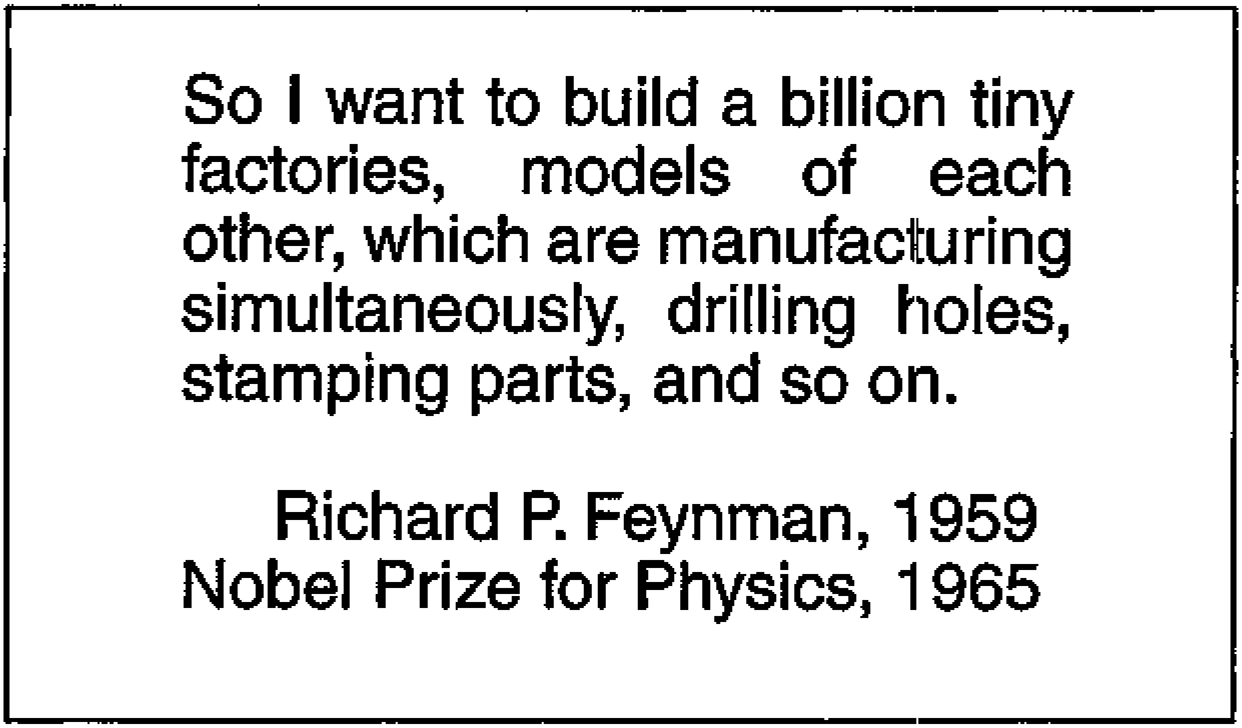
The theoretical concept of machine duplication is well developed. There are several alternative strategies by which machine self-replication can be carried out in a practical engineering setting.

Advanced Automation for Space Missions
Proceedings of the 1980 NASA/ASEE Summer Study



Complexity of Self Replicating Systems (bits)

Von Neumann's Universal Constructor	about 500,000
Internet Worm	500,000
Mycoplasma capricolum	1,600,000
E. Coli	8,000,000
Drexler's Assembler	100,000,000
Human	6,400,000,000
NASA Lunar Manufacturing Facility	over 100,000,000,000



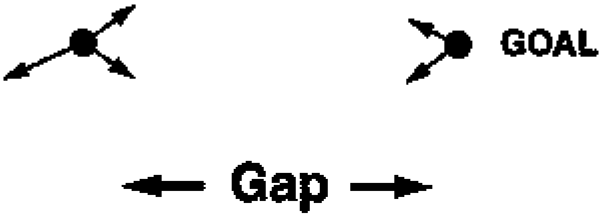
So I want to build a billion tiny factories, models of each other, which are manufacturing simultaneously, drilling holes, stamping parts, and so on.

Richard P. Feynman, 1959
Nobel Prize for Physics, 1965

Molecular Manufacturing Today

What We Can Synthesize Today

What We Think Molecular Manufacturing Looks Like Today

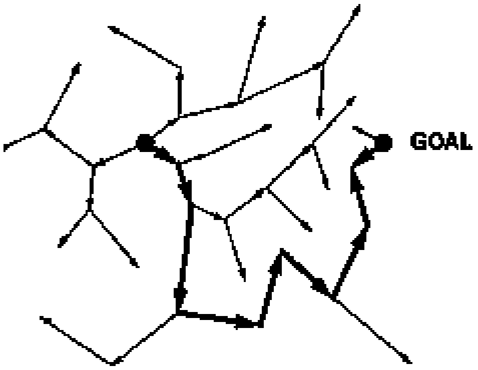


GOAL

Gap

Molecular Manufacturing: A Poor Approach For Closing the Gap

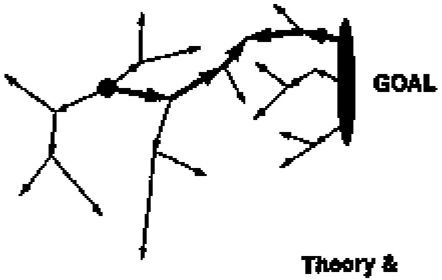
Extend only what we can synthesize: Experiment alone



GOAL

Molecular Manufacturing: A Better Approach For Closing the Gap

Extend *both*
what we can
synthesize
and
what we can model



Problems to Avoid

Finding the lowest energy conformation among the many possible for floppy molecules can be computationally intensive.

Solution: The use of stiff molecules (bricks) can avoid this.

Problems to Avoid

Radiation will cause damage and errors.

Solution: Radiation shielding is difficult. Instead, assume that background radiation is unavoidable and causes a certain error rate. This error rate still permits systems with tens of billions of atoms with a mean time between radiation hits of many decades. Design systems that tolerate this error rate.

Problems to Avoid

Light can cause photochemical reactions and photochemical damage.

Solution: Keep it in the dark. Even a thin layer of metal (a few hundred nanometers) will reduce photochemical damage to below radiation damage.

An alternative approach (somewhat more complex): design the system to be light tolerant (transparent).

Problems to Avoid

Thermal noise can cause damage.

Solution: Design the system so that transitions from a correct state to an incorrect state have barriers that are large compared with *kT*.

To achieve thermal error rates at room temperature comparable with radiation damage, barrier heights should be about 300 to 400 maJ (350 maJ is about 2.2 electron volts, or 50 kcals/ mole).

Problems to Avoid

Computational models can produce the wrong answer if the actual physical system differs from the abstract system (due to contaminants, e.g., unexpected atoms in unexpected places).

Solution: deal with all contaminants at the system boundary. Robust barriers that tolerate external contaminants and keep the internal environment well ordered are easier to design than complex systems that tolerate dirt.

Problems to Avoid

Thermal noise: thermal vibration can cause significant positional errors. This is of particular importance in the design of positional devices.

Solution: if it's not accurate enough, make it bigger. Scaling laws mean bigger objects are stiffer, and hence less subject to thermal noise.

Problems to Avoid

Modeling errors: the system design must work despite the use of an imperfect model.

Solution: Robust designs that work in the face of expected modeling errors must be used. In many cases, this can be viewed as designing for a high-temperature environment. Thermal errors and errors caused by the model can be viewed as abstractly similar.

Many of the questions raised by the design of an assembler can be answered

By experiment
By computational chemistry
By a combination of both

Computational chemistry is a historically unique tool which lets us pose and answer questions inaccessible to present experimental methods. This makes it of unique value in planning the molecular manufacturing systems of the future.

Computational Nanotechnology:

Model future molecular machines using today's computational chemistry software.

Feasible for devices that are difficult or impossible to make with today's methods.

Speeds development of better systems

Rapidly review and discard dead ends

Inexpensive

Informative

DESIGN AND FABRICATION OF THE FIRST MOLECULAR MANUFACTURING SYSTEMS WILL REQUIRE EXPERTISE IN:

Physics
Robotics
Chemistry
Surface Science
Materials Science
Computer Science
Electrical Engineering
Mechanical Engineering
Computational Chemistry

•
•
•

The best way to predict the future is to invent it.

Alan Kay

STUDYING OCCUPATIONAL HAND DISORDERS

Frank R. Wilson, M.D., University of California
George P. Moore, Ph.D., University of California

Humans have established a powerful hold on their environment largely because of the exploitation of novel skill potentials created by the hand-brain marriage. During the latter part of the 20th century, computer-machine technology has been widely exploited in the workplace to achieve accuracy, efficiency, and economy in tasks dominated by rapid, stereotypic movement sequences. Although robotics remains a rapidly growing industrial science, computerized machines designed simply to augment human motor performance have not been the unqualified success designers and users had hoped. The computerized human worker does not always behave as expected.

During the past decade, the incidence of work-related repetitive motion disorders has more than tripled from approximately 6 to 21 per 100,000 workers, making them now responsible for more than one-half of all occupational disorders reported in the United States[1] *Among the most commonly reported of these are tenosynovitis and tendinitis, nerve entrapment syndromes (especially the carpal and cubital tunnel syndromes) the hand-arm vibration syndrome,*[2] *and reflex sympathetic dystrophy.*[3] *Operators of electronic keyboards (especially at computer work stations) comprise an especially fast-growing group of individuals disabled because of hand and arm complaints. Another group increasingly coming to attention are performing artists (especially advanced instrumental music students and orchestra musicians). The study of the latter group has yielded improved understanding into the cause of some forms of occupational hand disorder.*[4]

These unexpected difficulties have contributed to a new generation of research questions in motor control, and may require novel strategies and assessment tools. As ergonomists search for ways to improve the "fit" between humans and machines, and as computers (and employers) drive workers toward higher output, it becomes increasingly apparent how little is actually known about the biomechanical and neurophysiological correlates of human skilled — especially manual — movement.

We will briefly review the history of occupational hand disorders and will introduce the results of preliminary studies of timing among high-level musicians experiencing loss of musical performance skill. Initial experience with these studies suggests the possibility of significant new trends and opportunities in research on biomechanical and neurophysiological processes in human skilled movement.

The following are among the more noteworthy implications:

- *MIDI devices and software developed to control them represent a high level and comparatively inexpensive technology readily available and appropriate for study of the control of hand movement;*

- *individuals with motor impairments (including extremely large populations of patients with congenital and acquired neurologic disabilities) comprise a huge untapped subject population for clinical research and client population for cognitive and motor rehabilitation efforts;*

- *improvements in miniaturized, fast, accurate position sensors and physiologic transducers, as well as supporting software, are needed, as are devices for more detailed study of hand biomechanics;*

- *computer modeling of biomechanics and neuromuscular control will probably be essential for the study of muscle synergies in upper extremity movement.*

1. A brief anthropologic perspective on the "modem hand": troglodytes, Lucy, and beyond — a modern version of the Prometheus myth! An understanding of disturbances of the origins of hand and arm problems presupposes an appreciation of the normal operation of the entire upper extremity, including shoulder and scapular movement, elbow and wrist function, and the special features of prehensile and non-prehensile movements of the human hand. The human hand differs from the ape hand in several important ways: the thumb is longer in relation to the phalanges in humans, permitting greater contact between palmar surfaces of the thumb and finger tips; greater axial rotation occurs at the MP joints of the digits, permitting a "3-jaw chuck" (baseball) grip and "5-jaw chuck" grip; the ulnar side of the hand can be opposed to the thumb, permitting an oblique grasp of a shaft and thereby permitting the long axis of the arm to be greatly extended.

Other modifications in wrist bones (especially the capitate) and ligaments permit dispersion of impact forces delivered through the long axis of the central metacarpals. Most of these changes were present in the hand of Australopithecus afarensis 2.4 million years ago; the hand of Homo sapiens sapiens differs mainly in the increased pronation and opposition of the thumb and greater axial rotation of the 4th and 5th fingers, and their capacity for opposition. The increase in functional capacity of the hand as a result of these "minor" changes. however, probably explains the enormous expansion of manual skill of humans over other primates, the exceptional capacity to make and use refined tools, and possibly to some extent even the three-fold increase in brain size that followed the advances over earlier primate forms found in Lucy's hand.[5,6,7,8,9]

2. Functional anatomy of prehension: In a landmark paper published in 1956, anatomist J.R. Napier established the first anatomical-functional classification of hand movements.[10] First, he separated "prehensile" from "non-prehensile" movements.[a] Prehensile movements are those in which an object is held partly or wholly within the hand; non-prehensile movements are those in which the object is manipulated by the hand or fingers, but not seized or grasped. Combing one's hair is an example of the former; typing or playing the piano is an example of the latter.

Within the prehensile group, Napier distinguished two kinds of grip: the "power grip," in which part of the object is held by fingers and/or thumb against the palm; and "precision grip," in which the object is pinched between the thumb and any of the other fingers, without touching the palm.

Another power grip is a carrying or "porter's" grip, also called a "hook" grip, in which the fingers do most of the work without the thumb — doing chin-ups, and carrying a briefcase use this kind of grip. What is interesting about this grip from an anthropologic point of view is that it does not require supination of the ulnar side of the hand, and therefore control of the object is very crude. As Mary Marzke points out, non-human primates, including A. afarensis, are limited to this kind of power grip.[11]

[a] The term "prehension" comes from the Latin word meaning "to seize," and does not quite do justice to the variety of uses to which the hand can be put in object manipulation and control.

Subsequent classifications have added refinement and modifications to Napier's system. Elliott and Connolly proposed use of the terms "intrinsic" and "extrinsic" to distinguish between movements in which the object is manipulated within the hand (intrinsic) and movements in which the object is "displaced by the hand as a whole, using the upper limb."[12]

Elliott and Connolly proposed a further subdivision of intrinsic hand movements into those involving simple or reciprocal synergies and those using sequential patterns. Simple synergies involve simple flexion of the thumb and fingers, as in handwriting; reciprocal synergies involve separation of the thumb from the movements of the other fingers, as in tightening a nut; sequential step movements involve complex rotary movements, like turning a combination lock or manipulating knitting needles and yarn.

Less attention has been paid by scientific writers to non-prehensile movements, but the increasing prevalence of occupational injuries related to keyboard use will change that. There is in fact a very large body of technical literature on these movements, but the source is largely within the domain of professional teaching of keyboard and musical instrument use. Unfortunately this pedagogy is for the most part entirely empirical. ***The Hand Book,*** written by a concert pianist, has recently been published and suggests that concepts drawn from high level musical instrument study could be have a favorable impact on the present epidemic of computer keyboard injuries.[13]

3. Recent studies of upper extremity biomechanics — what we have learned from musicians with occupational cramp: In a series of English language publications beginning in 1974, Christoph Wagner provided details of the static and dynamic characteristics of movement in the upper extremities of musicians whose ages range from 16 to 72 years, and who play virtually all instruments in common use.[14,15,16] The role of joint hypermobility in musical performance has recently been studied at a major American conservatory.[17] In general, these studies have indicated that unusual biomechanical pre-conditions are associated with recurrent pain syndromes of the upper extremity. Most prominently, stringed instrument players with chronic forearm pain have a higher than expected incidence of low supination range at the elbow; keyboardists with forearm pain, by contrast, have a higher than expected incidence of low pronation range at the forearm. Because of the limited availability of quantitative measures of upper arm biomechanics, the contribution of such abnormalities among office and industrial workers with repetitive stress injuries is unknown.

4. Musical vamps and manual cramps. Even more disabling than repetitive stress injury is the syndrome of occupational cramp. A recent study reported by Wilson, Wagner and Hömberg links biomechanical abnormalities to the establishment of a "learned motor disorder," and to other risk factors, including repetition rates and psychological factors.[18] Moore's studies of timing and muscle activation in performance of trills on the cello and piano establish the feasibility of simultaneous recordings of movement and muscle activation in musical performance.[19,20,21] The recent adaptation of MIDI (musical instrument digital interface) technology creates both new opportunities and new problems

in that regard.[22] We are now seeking ways to use these new methods to examine movements in which there are proven disturbances in the physiologic control of reciprocal inhibition of flexion-extension movements of the fingers, a hallmark of writers' cramp and related disorders.[23,24]

5. Treatment and prevention. The ubiquity and intractability of occupational hand disorders has generated a vigorous response from both industry and the medical community. Some prevention strategies (focusing on work station ergonomics) appear to have been helpful, and modification of work habits (including upper body and limb posture during movement) have been helpful in many cases. Individual biomechanics have not been addressed (as they are beginning to be among musicians), and physiologically rational training for computer-operated keyboards remains a largely unmet challenge in injury prevention.

References

1. Bureau of Labor Statistics Reports on Survey of Occupational Injuries and Illnesses in 1977-1989. Washington, D.C.: Bureau of Labor Statistics, US Department of Labor, 1990.

2. Rempel D, Harrison R, Barnhart S. Work-related cumulative trauma disorders of the upper extremity. JAMA 1992;267:838-842.

3. Lockwood A, Lindsay M. Reflex sympathetic dystrophy after overuse: the possible relationship to focal dystonia. Medical Problems of Performing Artists 1989;4:114-117.

4. Markison R. Hands on fire: how to recognize, treat, and avoid the occupational hazards of keyboard performance. Keyboard, pp. 93-110. April, 1994.

5. Marzke M: Joint functions and grips of the *Australopithecus afarensis* hand, with special reference to the region of the capitate, Journal of Human Evolution 1983;12:197-211.

6. Marzke, M, Wullstein, K, Viegas, S. Evolution of the power ("squeeze") grip and its morphological correlates in hominids. American Journal of Physical Anthropology 1992;89:283-298.

7. Marzke, M. Tool use and the evolution of hominid hands and bipedality. In Primate Evolution: Proceedings of the 10th Congress of the International Primatological Society, Vol. 1. (eds. Else & Lee). Cambridge University Press, Great Britain, 1986.

8. Marzke M: Evolutionary development of the human thumb. Hand Clinics Volume 8, Number 1, 1-8, February 1992.

9. Calvin W. The Throwing Madonna. pp. 3-13. New York. McGraw-Hill, 1983.

10. Napier, J. The prehensile movements of the human hand. The Journal of Bone and Joint Surgery, 38B, No. 4:902-913, 1956.

11. Marzke, M., Wullstein, K., Viegas, S., Evolution of the power ("squeeze") grip and its morphological correlates in hominids. American Journal of Physical Anthropology 89:283-298, 1992.

12. Elliott, J., Connolly, K. A Classification of manipulative hand movements, Developmental Medicine and Child Neurology, 26: 283-296, 1984.

13. Brown S. The Hand Book: Preventing Computer Injury. New York. Ergonome, 1993.

14. Wagner C. Determination of finger flexibility. European Journal of Applied Physiology 1974;32:259-278.

15. Wagner C. Determination of the rotary flexibility of the elbow joint. European Journal of Applied Physiology 1977;37:47-59.

16. Wagner C. The pianist's hand: anthropometry and biomechanics. Ergonomics 1988;31:97-131.

17. Larsson L-G, Baum J, Mudholkar G, Kollia G. Benefits and disadvantages of joint hypermobility among musicians. NEJM 1993;329:1079-1082.

18. Wilson F, Wagner C, Hömberg V. Biomechanical abnormalities in musicians with occupational cramp/focal dystonia. Journal of Hand Therapy; October-December 1993: 298-307.

19. Moore G. A computer-based portable keyboard monitor for studying timing performance in pianists. Annals of the New York Academy of Sciences 1984;423:651-652.

20. Moore G. Piano trills. Music Perception 1992;9:351-360.

21. Moore G. The study of skilled performance in musicians. In Roehmann F, Wilson F (eds). The Biology of Music Making. St. Louis, MMB Music, Inc., 1988.

22. Wilson F. Digitizing digital dexterity: a novel application for MIDI recordings of keyboard performance. Psychomusicology 1993;11:79-95.

23. Nakashima K, Rothwell J, Day B, et al. Reciprocal inhibition between forearm muscles in patients with writer's cramp and other occupational cramps, symptomatic hemidystonia and hemiparesis due to stroke. Brain 1989;112:681-697.

24. Panizza M, Hallett M, Nilsson J. Reciprocal inhibition in patients with hand cramps. Neurology 1989;39:85-89.

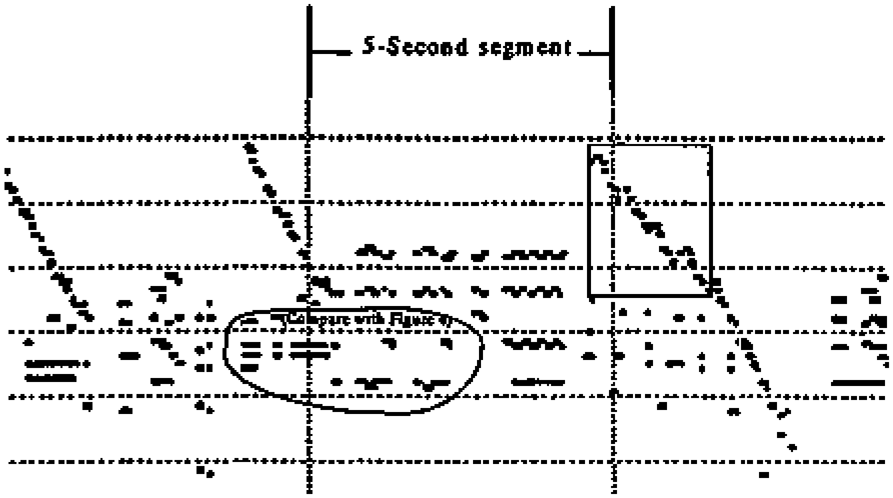


Figure 1. Pianist UD. Unedited 15-second segment Chopin B min Sonata. Note descending runs. 2-second segment of third run (grey box) is 25 notes long and is reproduced in following figures. Reproduced from *Psychomusicology* 1993;11:79-95.

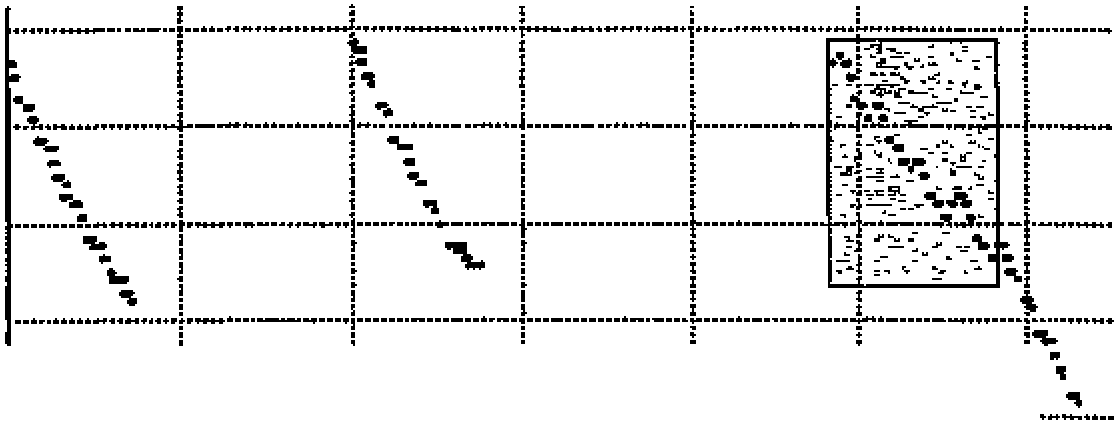


Figure 2a. Pianist UD. Same segment with all left-hand notes removed, original timing unchanged.

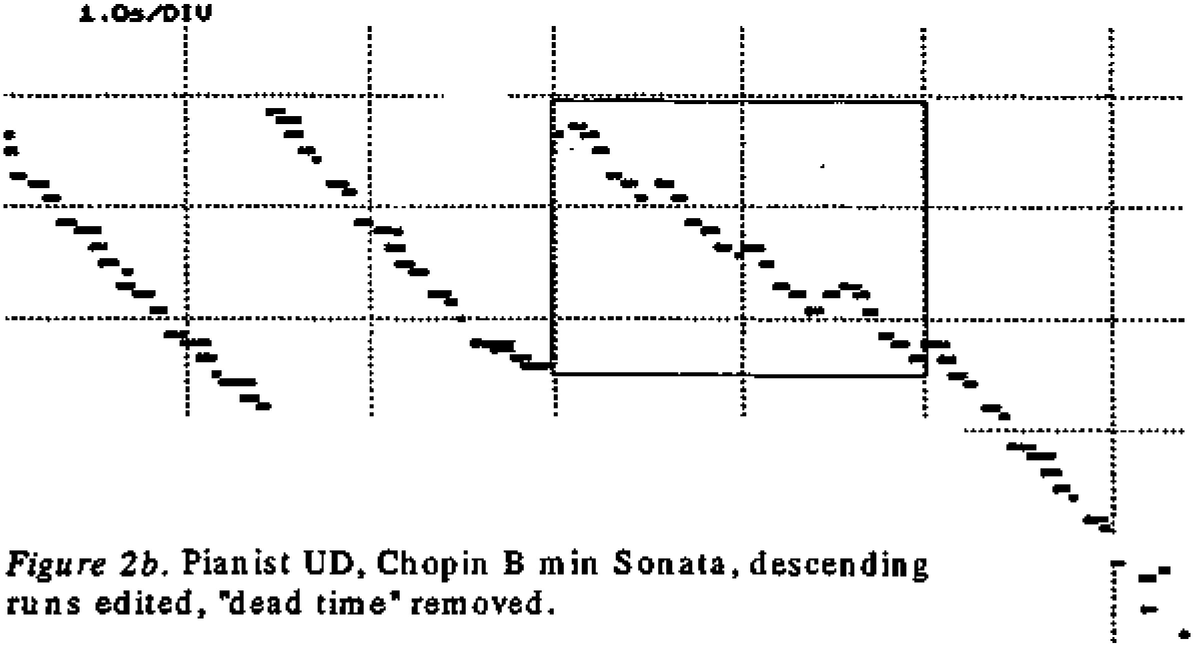


Figure 2b. Pianist UD, Chopin B min Sonata, descending runs edited, "dead time" removed.

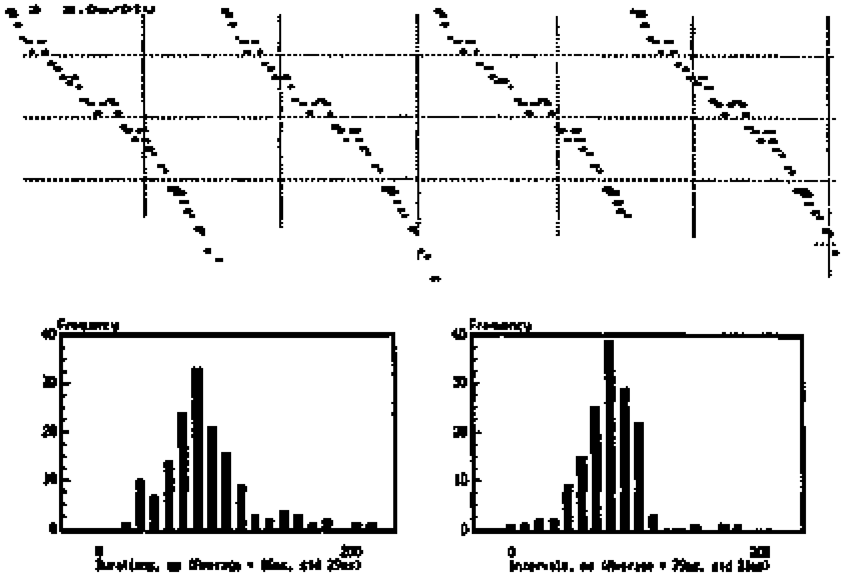


Figure 3. Pianist UD, Chopin B min Sonata. Four successive performances edited to show 38-note descending run with combined note duration and interval histograms.

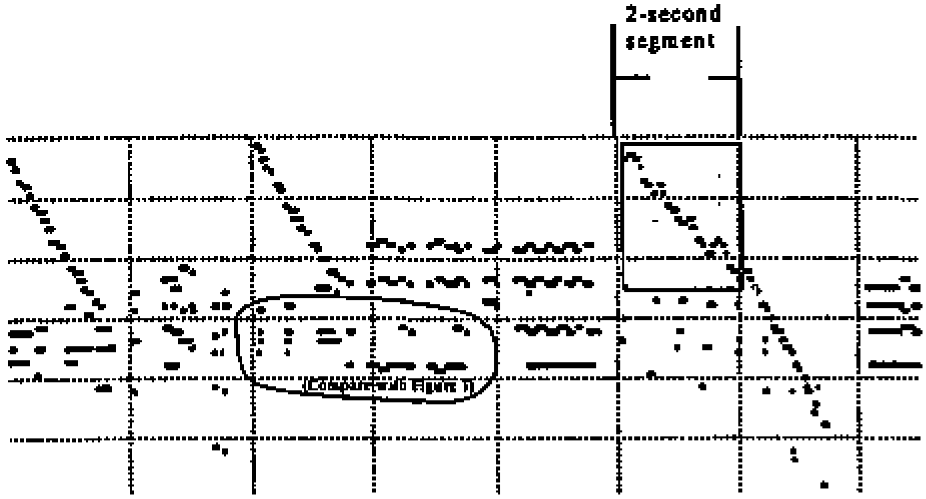


Figure 4. Pianist PO. Unedited 16-second segment Chopin B min Sonata. Note descending runs.

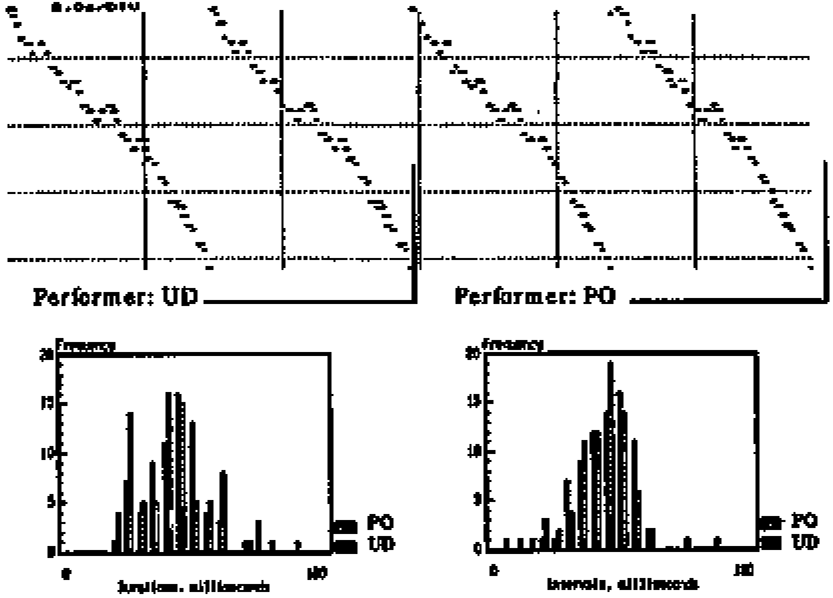


Figure 5. Comparison of pianists UD and PO, two performances by each of same 38-note descending run in Chopin B minor Sonata, with histograms.

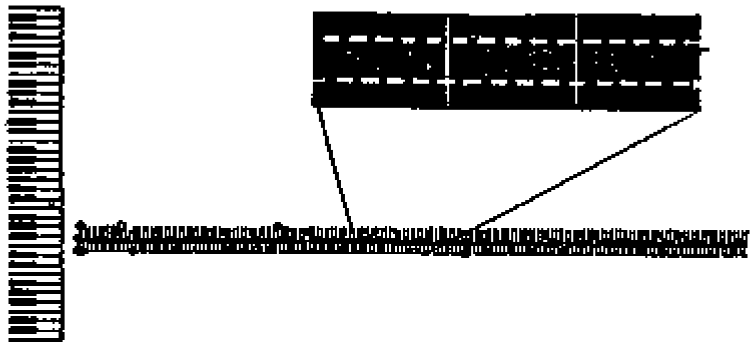


Figure 4a RS, left hand trill, self-paced. Average interval length is 202 ms (std 18 ms); average note duration 197 ms (std 20 ms); average velocity 56.6 units. Vertical bars indicate note velocity; circles indicate new high or low in the series. Insert (color reverse) shows 6-second detail.

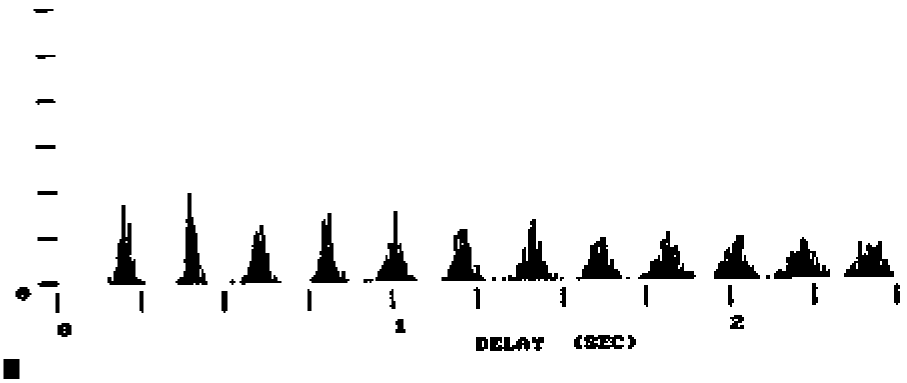


Figure 4c. RS, left hand trill, interval autocorrelation diagram. Each note is plotted as the first of a series to indicate short term and long term timing control. This performance demonstrates extreme stability of note intervals.

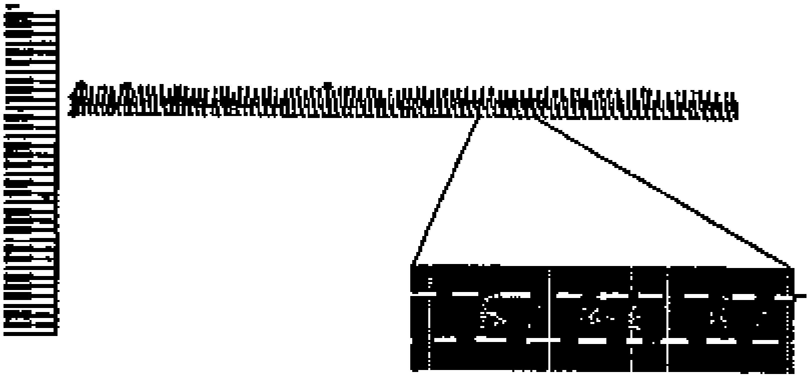


Figure 5a. RS, right hand trill, self-paced. Average interval length is 384 ms (std 84ms); average duration is 385 ms (std 90ms); average velocity is 66.93 units (std 3.51 units). Vertical bars, circles as in Figure 4a. Insert (color reverse) shows 6-second detail.

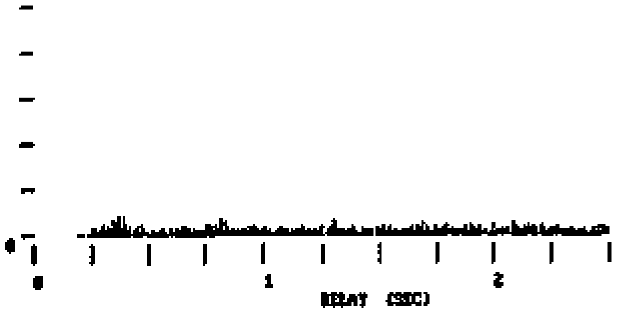
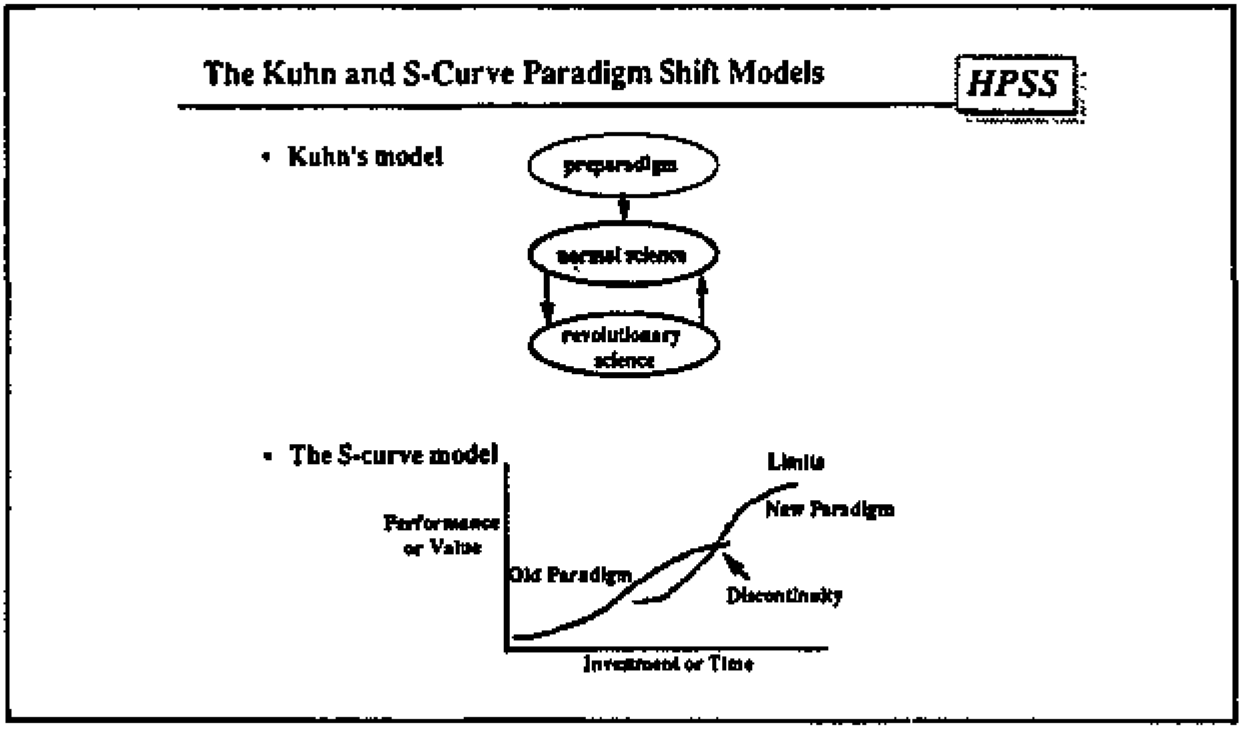


Figure 5c. RS, right hand trill interval autocorrelation diagram. Compare with figures 4c, 6c. Analysis shows no evidence of note-to-note rhythic precision.

THE HIGH PERFORMANCE STORAGE SYSTEM

Dick Watson, LLNL

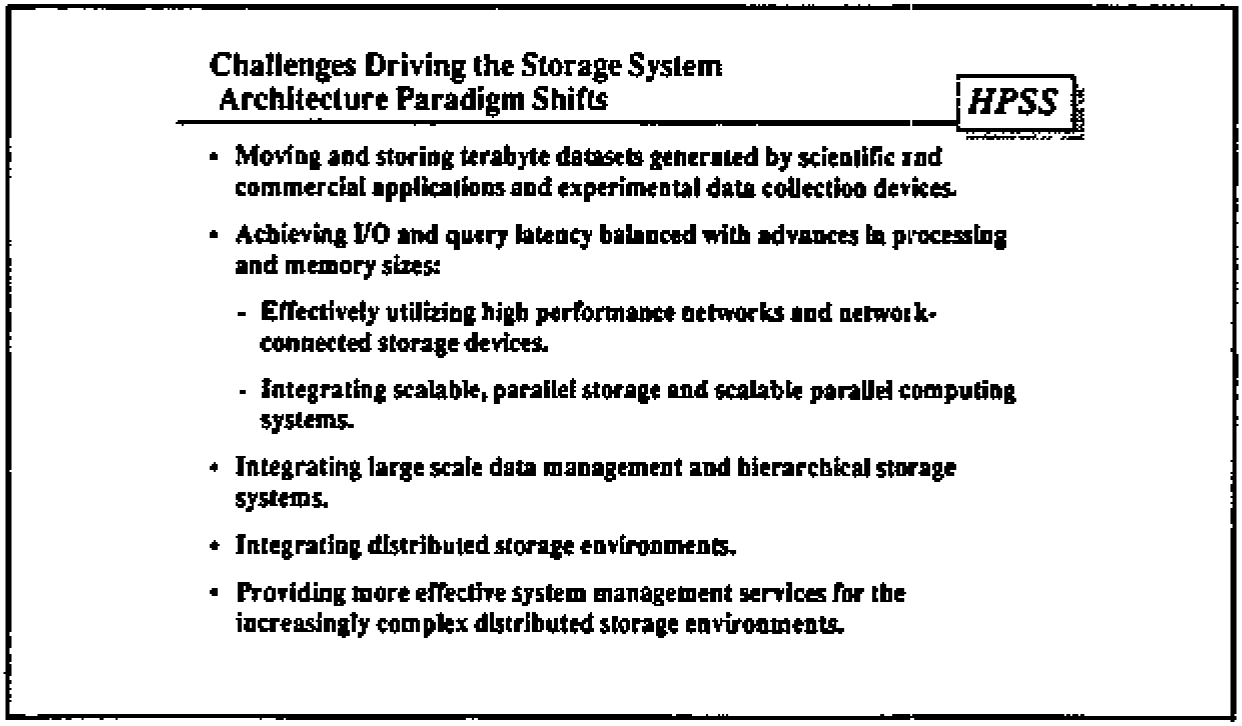


The Kuhn and S-Curve Paradigm Shift Models

HPSS

- Kuhn's model

- The S-curve model



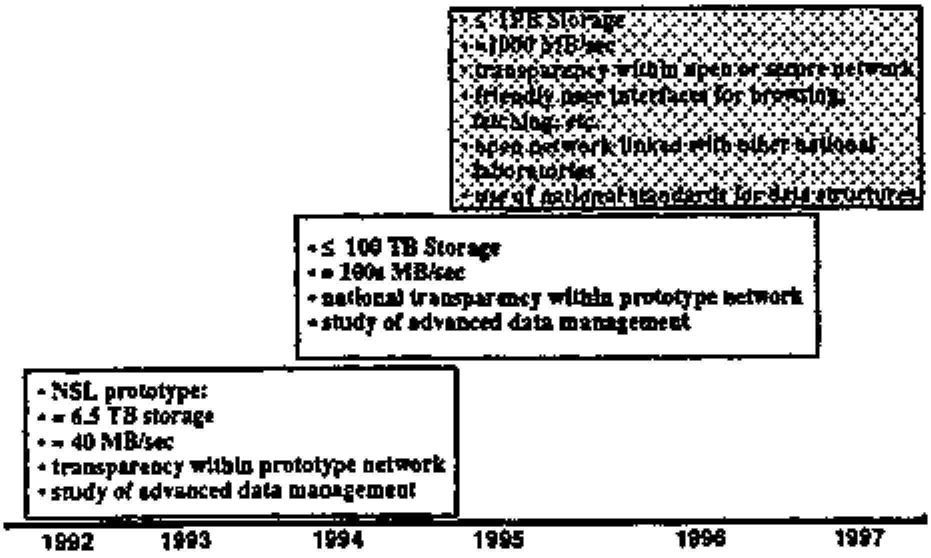
Challenges Driving the Storage System Architecture Paradigm Shifts

HPSS

- Moving and storing terabyte datasets generated by scientific and commercial applications and experimental data collection devices.
- Achieving I/O and query latency balanced with advances in processing and memory sizes:
  - Effectively utilizing high performance networks and network-connected storage devices.
  - Integrating scalable, parallel storage and scalable parallel computing systems.
- Integrating large scale data management and hierarchical storage systems.
- Integrating distributed storage environments.
- Providing more effective system management services for the increasingly complex distributed storage environments.

MPP computing sets pace for growth of archival storage

HPSS



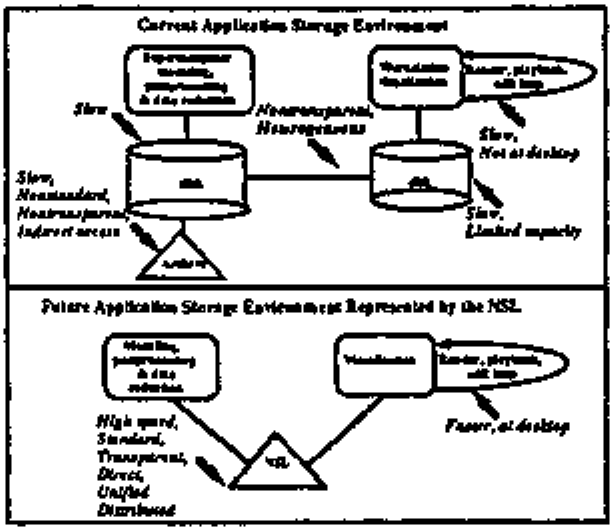
- ≤ 1PB Storage
- ≤ 1000 MB/sec
- transparency within open or secure network
- friendly user interfaces for browsing, searching, etc.
- open network linked with other national laboratories
- use of national standards for data structures

- ≤ 100 TB Storage
- ≈ 100s MB/sec
- national transparency within prototype network
- study of advanced data management

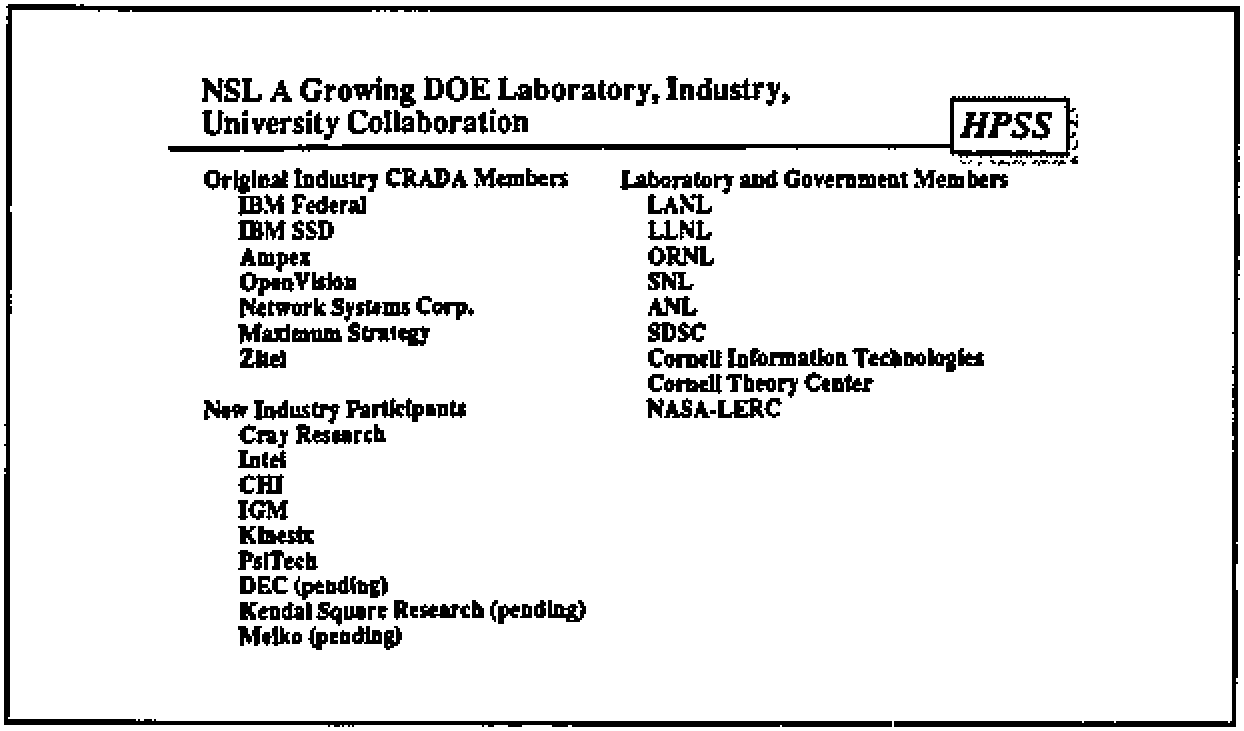
- NSL prototype:
- ≈ 6.5 TB storage
- ≈ 40 MB/sec
- transparency within prototype network
- study of advanced data management

Enabling Applications

HPSS


Current Application Storage Environment

Future Application Storage Environment Represented by the NSL



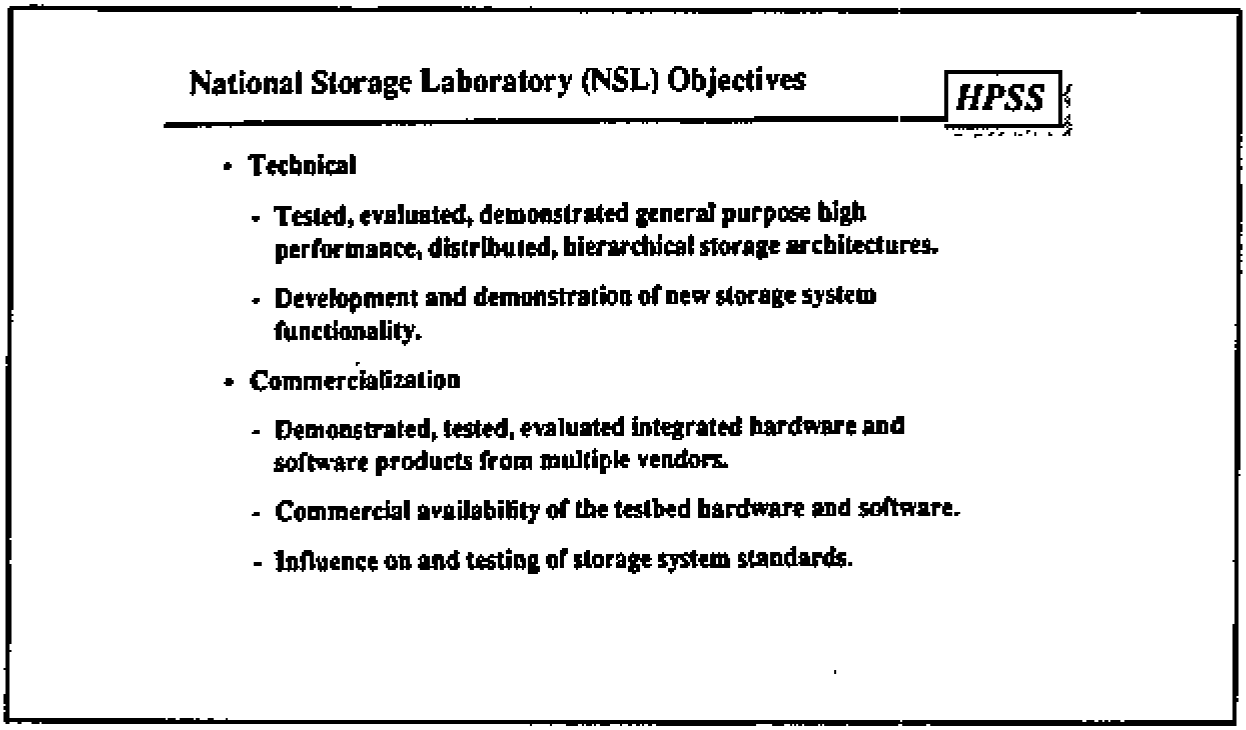
NSL A Growing DOE Laboratory, Industry, University Collaboration

HPSS

Original Industry CRADA Members
- IBM Federal
- IBM SSD
- Ampex
- OpenVision
- Network Systems Corp.
- Maximum Strategy
- Zitel

New Industry Participants
- Cray Research
- Intel
- CHI
- IGM
- Kinesix
- PsiTech
- DEC (pending)
- Kendal Square Research (pending)
- Meiko (pending)

Laboratory and Government Members
- LANL
- LLNL
- ORNL
- SNL
- ANL
- SDSC
- Cornell Information Technologies
- Cornell Theory Center
- NASA-LERC



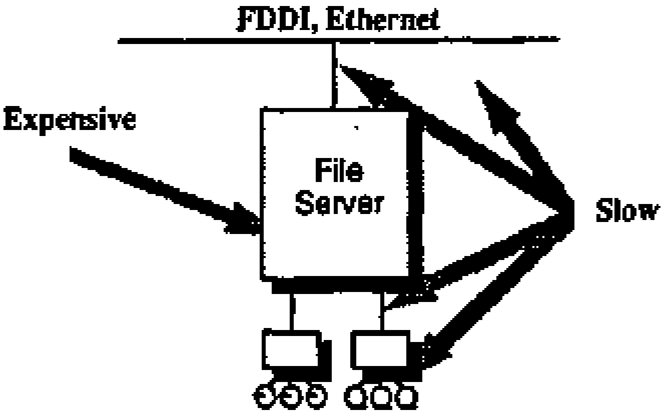
National Storage Laboratory (NSL) Objectives

HPSS

- Technical
  - Tested, evaluated, demonstrated general purpose high performance, distributed, hierarchical storage architectures.
  - Development and demonstration of new storage system functionality.
- Commercialization
  - Demonstrated, tested, evaluated integrated hardware and software products from multiple vendors.
  - Commercial availability of the testbed hardware and software.
  - Influence on and testing of storage system standards.

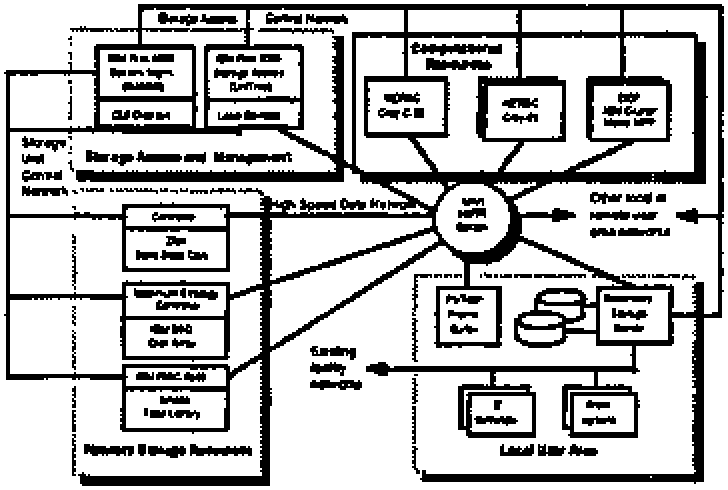
The Challenge of Achieving Balanced I/O Architectures - The Limits of Mainframe Channel Connected Storage

HPSS



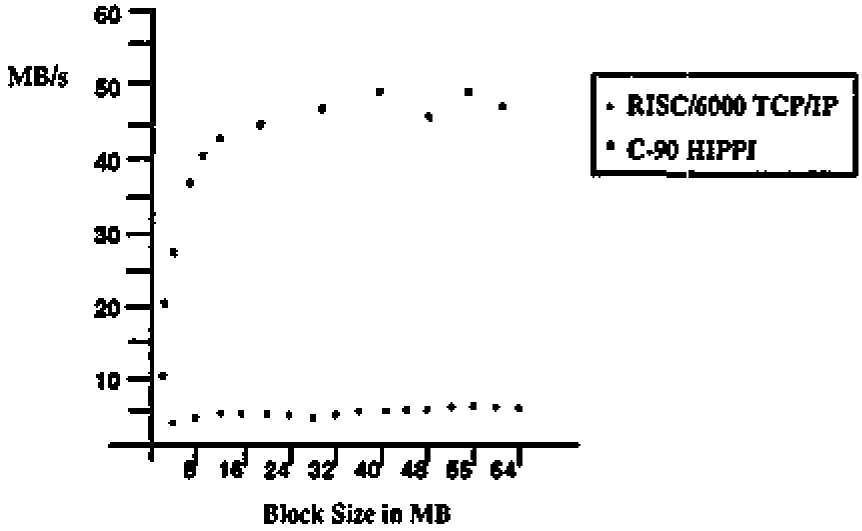
Architectural Overview of the National Storage Laboratory

HPSS


Storage Access
Control Network
Computational Resources
Storage Unit Control Network
Storage Access and Management
High Speed Data Network
Other local or remote user data networks
Network Storage Resources
Local User Area

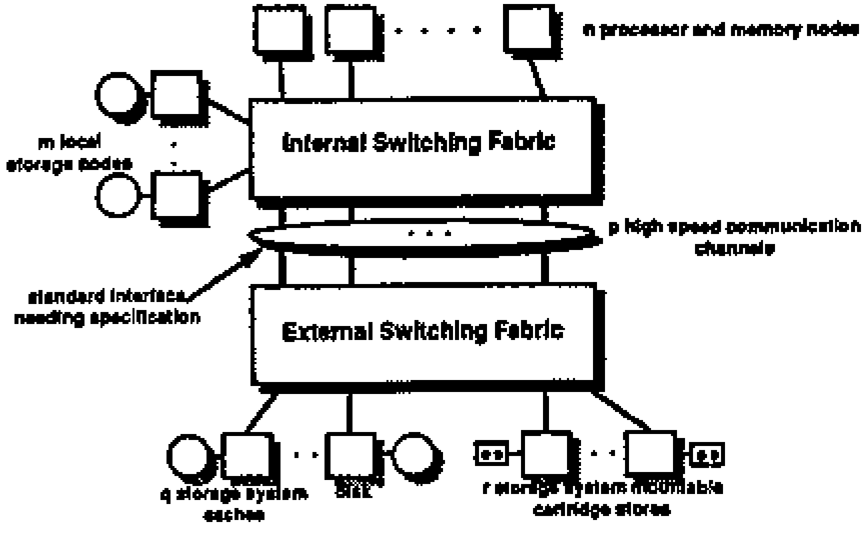
NSL UniTree Performance

HPSS



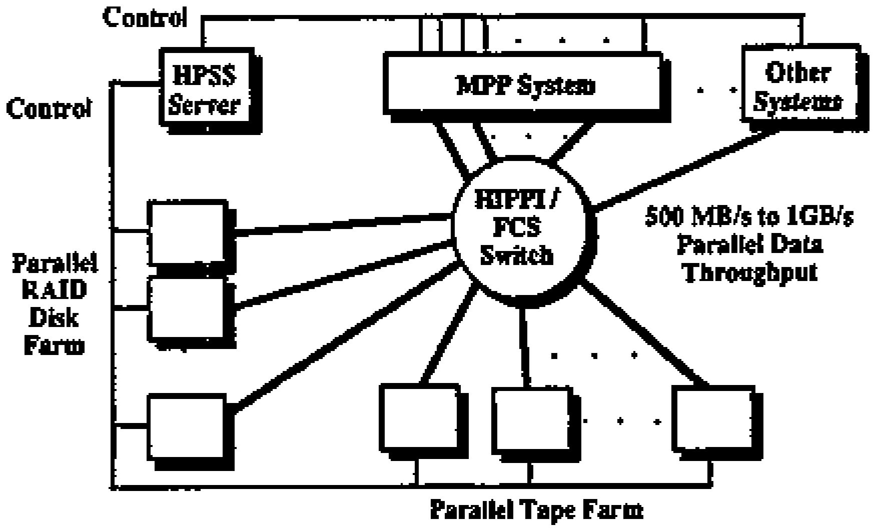
The Challenge of Archiving Scalable Parallel I/O: the Extended Data Layout Problems

HPSS


n processor and memory nodes
m local storage nodes
Internal Switching Fabric
p high speed communication channels
standard interface needing specification
External Switching Fabric
q storage system caches
disk
r storage system mountable cartridge stores

An Example: The Configuration that the NSL High Performance Storage System (HPSS) Will Support

HPSS



Characteristics of HPSS

HPSS

- Focus on scalability and MPP integration
  - Ability to add hardware to support striping across devices and I/O channels.
  - Servers designed for multitasking and multiprocessing.
  - Ability to distribute servers.
  - Petabyte store, containing billions of files, millions of directories.
  - Scalable data transfer to GB/s range.
- Interfaces to support integration with large scale data management systems.
- Support for a wide range of storage devices and multiple storage hierarchies.

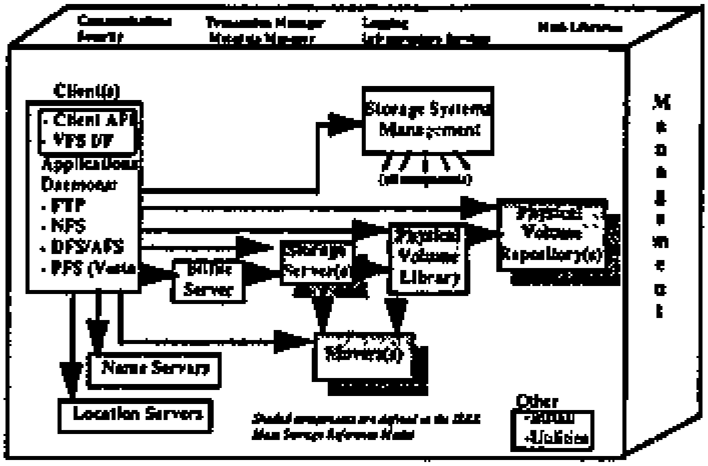
Characteristics of HPSS (continued)

HPSS

- Support for standard interfaces, NFS, AFS, DFS, FTP.
- Extensive GUI based storage system management services.
- Security built in from the initial design.
- Portability to multiple vendors platforms.
  - No kernal modifications
  - Built on OSF's DCE infrastructure, implemented in C POSIX compatible.
  - Layered architecture based on IEEE Mass Storage System Reference Model: Version 5.
  - Built using components from multiple vendors.
- Migration path from NSL-UniTree.
- Projected 3 year development investment > $10M.
- Reliability and recoverability features (e.g., Transaction management, system and File metadata on separate media and multiple copies)

HPSS Architecture

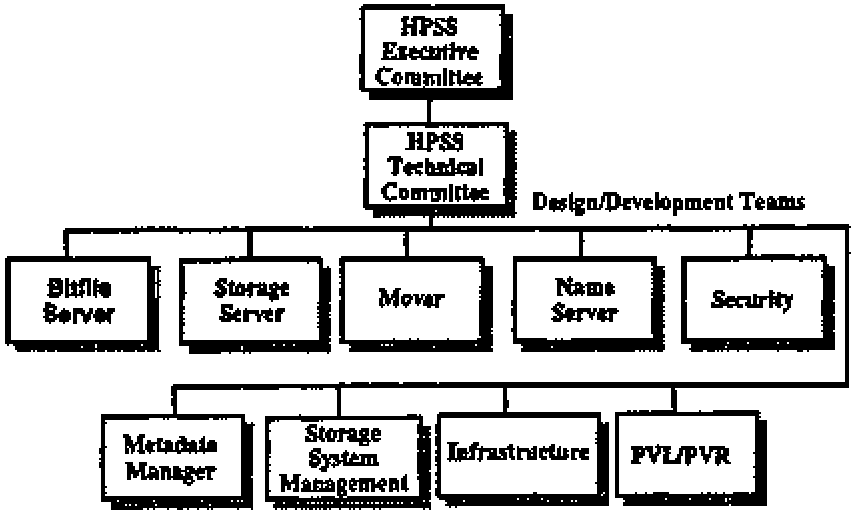
HPSS



Shaded components are defined in the IEEE Mass Storage Reference Model

HPSS Organization

HPSS



HPSS Executive Committee

HPSS Technical Committee

Design/Development Teams

Bitfile Server

Storage Server

Mover

Name Server

Security

Metadata Manager

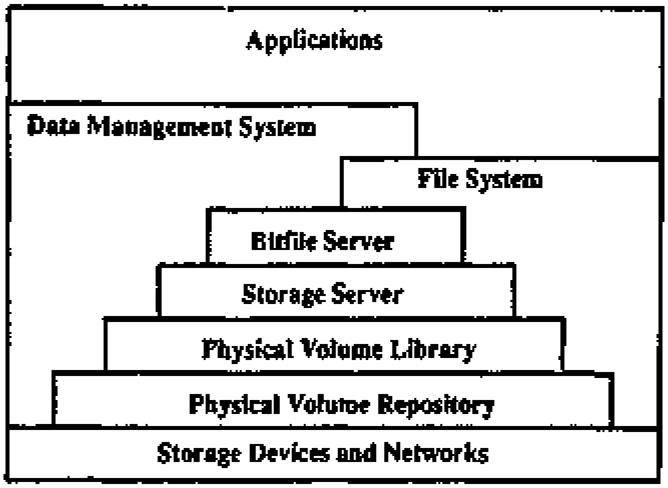
Storage System Management

Infrastructure

PVL/PVR

Layered Access to Storage System Services

HPSS



Applications

Data Management System

File System

Bitfile Server

Storage Server

Physical Volume Library

Physical Volume Repository

Storage Devices and Networks

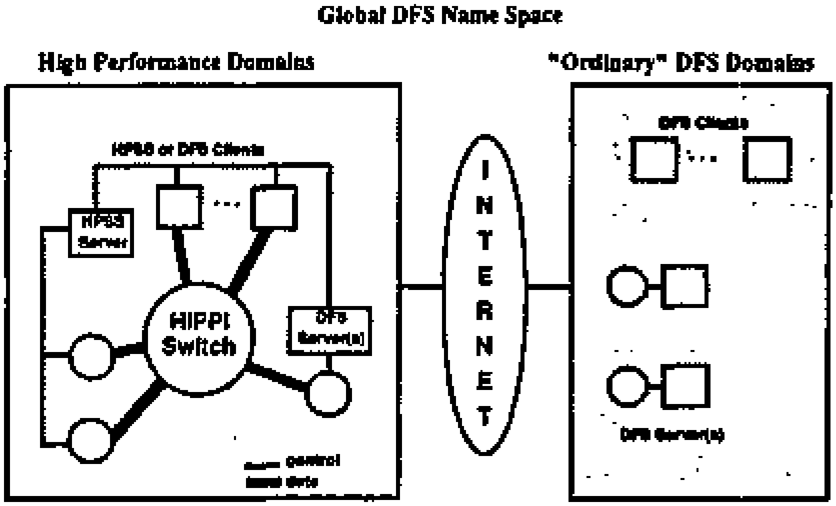
The Challenge of Integrating Large Scale Data Storage and Large Scale Data Management

HPSS

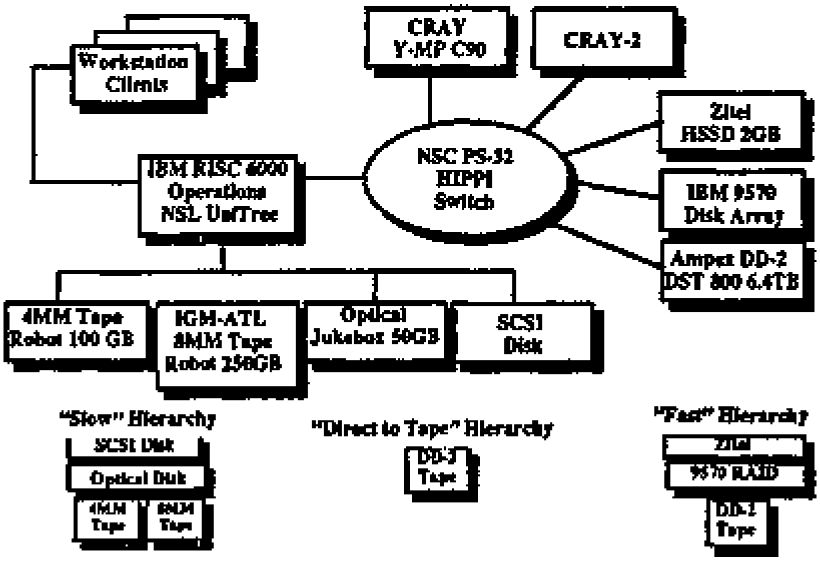
- **Problem:** Users don't think of their data in terms of files or clusters of files, but rather as application oriented abstractions (e.g., climate conditions in spatial and temporal terms).
- Approaches
  - Subsetting of the data based on predicated or observed query behavior and storage system characteristics
  - Abstracts or compression
  - Query time prediction
  - Storing appropriately organized metadata
  - Control by the data management system of clustered data layout (e.g., volume and order)
  - Access by the data management system to the appropriate storage layers and abstractions

Example of Desirable HPSS and DFS Integration

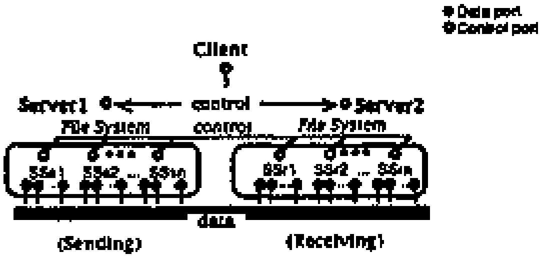
HPSS



Dynamic Storage Hierarchies at the National Storage Laboratory



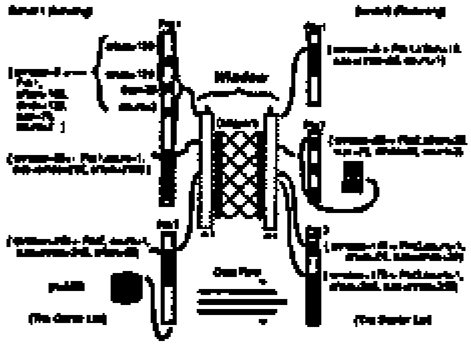
General Physical Model



- A transfer is between two file systems.
  - Data is distributed over a set of subsystems
  - {SSs1,2..} is a set of sending subsystems
  - {SSr1,2..} is a set of receiving subsystems
  - Data is moved between subsystems

Conceptual parallel flow
all bytes flow in parallel from sender to receiver

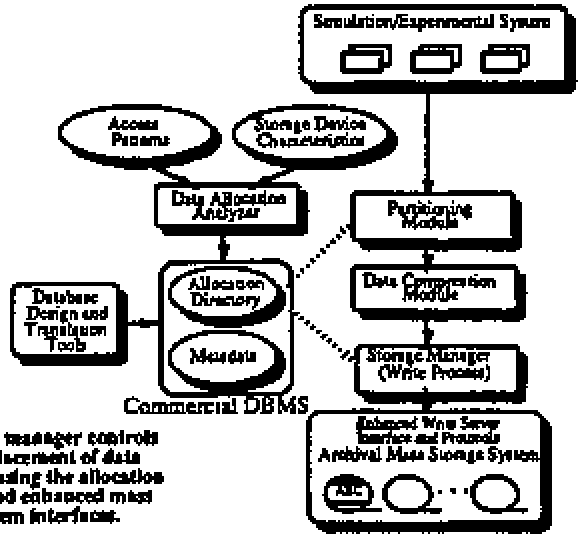
HPSS



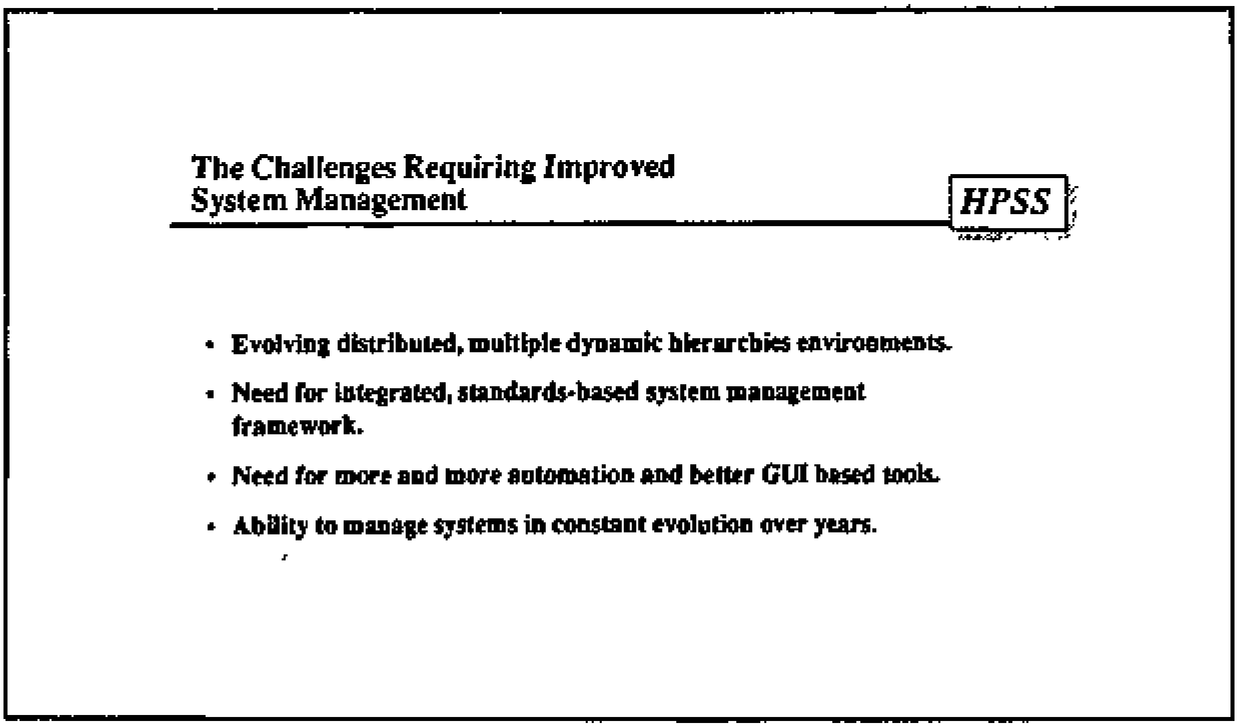
- object in sender sent to objects in receiver
- a gather list for sender, a scatter list for receiver
- list of sender and receiver defines a window
- there will be many mappings involved
- Data flows from sender through window to reciever

Data Allocation and Storage Management Details

HPSS


Simulation/Experimental System
Access Patterns
Storage Device Characteristics
Data Allocation Analyzer
Partitioning Module
Data Compression Module
Database Design and Translation Tools
Allocation Directory
Metadata
Commercial DBMS
Storage Manager (Write Process)
Enhanced Wire Server Interface and Protocols
Archival Mass Storage System
ABC

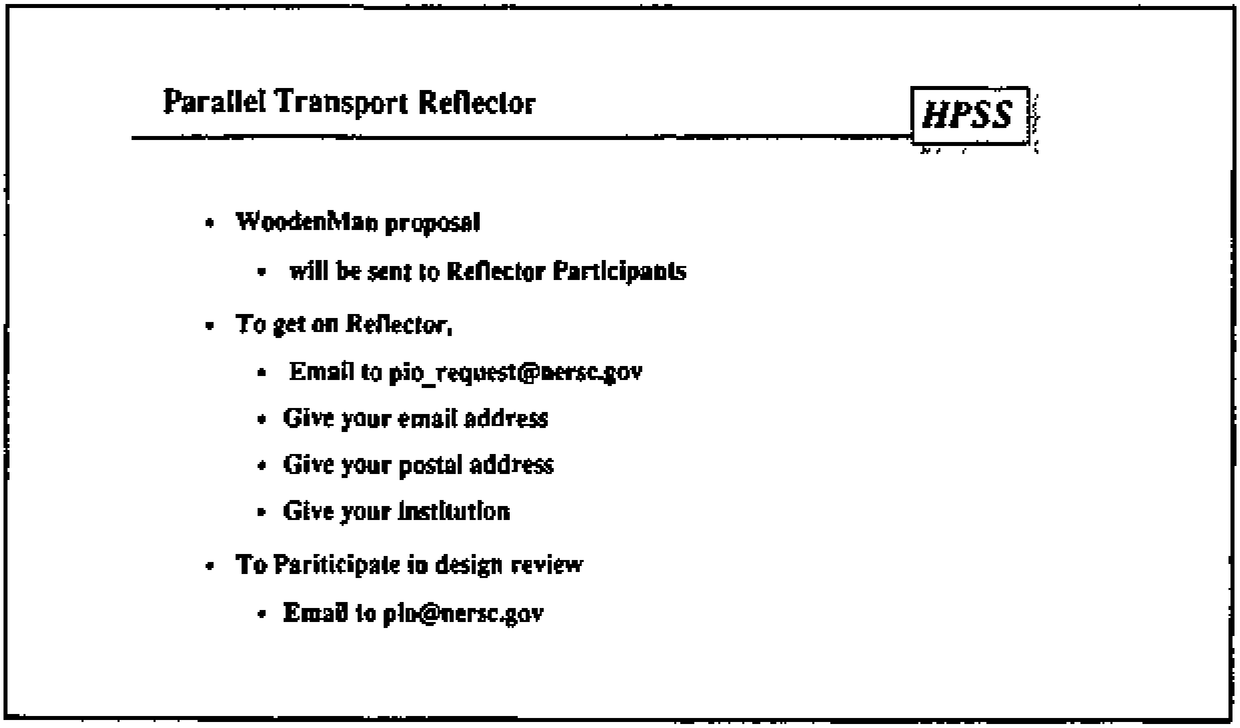
The storage manager controls the initial placement of data "clusters" using the allocation directory and enhanced mass storage system interfaces.



The Challenges Requiring Improved System Management

HPSS

- Evolving distributed, multiple dynamic hierarchies enviroments.
- Need for integrated, standards-based system management framework.
- Need for more and more automation and better GUI based tools.
- Ability to manage systems in constant evolution over years.



Parallel Transport Reflector

HPSS

- WoodenMan proposal
  - will be sent to Reflector Participants
- To get on Reflector,
  - Email to pio_request@nersc.gov
  - Give your email address
  - Give your postal address
  - Give your institution
- To Pariticipate in design review
  - Email to pio@nersc.gov

QUANTUM COMPUTERS AND FREDKIN GATES

Isaac Chuang, Stanford University

1 Introduction

Why do computers dissipate energy? .The amazing fact is that in principle, an ideal computer does not necessarily have to dissipate any energy at all in order to function properly. This conclusion is recent; before, it was widely believed that computation necessarily entailed dissipation of $kT \ln 2$ joules per elementary logic operation, according to the 1949 analysis of von Neumann. But in 1973, Bennett showed that for each logically irreversible Turing machine a reversible one could be constructed, in principle. Around the same time, Fredkin also developed his idea of a reversible logic gate. These demonstrated that in principle, a computer could be built which dissipates negligible energy.

Today, it is believed that a perfectly reversible computer may be constructed in principle, but susceptibility to noise would probably render it useless for practical purposes. Nevertheless, we have learned several fundamental facts about the relationship between energy dissipation and computing.

> First, we now realize that energy dissipation is solely a matter of *convenience*. Dissipating energy allows us to operate a finite sized computer reliably in the presence of noise, and finish our calculation in a finite amount of time.
>
> Second, we have come to understand better what "dissipation" means. Landauer's conjecture is that *dissipation occurs only when information is lost*, and vice versa.
>
> Finally, it is believed that through the study of ideal reversible logic gates, we may come to understand better the physics of *complex quantum systems and quantum measurement*, the subject of much controversy throughout the past half century.

Physics and computing have always had a deep relationship. In particular, the study of reversible computers brings us to what is perhaps the most intimate connection between the two disciplines, as demonstrated by these three observations.

[1] This work was supported by a Fannie and John Hertz Foundation Fellowship.

Today, I would like to present for you a review of the status of this field, and a summary of my on-going research in this area. What I will *not* tell you is that reversible computing is the wave of the future; it is not. What I *will* tell you. is the following message. The study of reversible computing will help to answer these questions:

- How much energy must be dissipated *in practical situations*. to perform arbitrary calculations? What are the fundamental physical limits?
- What issues are important in reducing energy consumption (beyond obvious technological limitations)?
- What technologies may be exploited to better investigate complex (quantum) systems?

This last item may be of particular interest to those who are investigating semiconductor logic devices of length scales smaller than a tenth-micron. because in that regime, quantum-mechanical effects start to become important. It is also a fascinating area for exploration in its own right, from the viewpoint of fundamental physics.

The outline of my presentation is as follows. I will begin with a brief description of the history of reversible computing, starting from Landauer's exorcism of Maxwell's demon. This background will provide a basis for the explanation of my research goals and approach. The ultimate application of my results will be in the area of minimal energy computing, which has already benefited from the technology of reversible logic. Finally, I address some open questions in the field by presenting early results from my research.

2 History of Reversible Computing

The original motivation for the study of the thermodynamics of computing was the desire to develop a thorough understanding of the inability of Maxwell's demon to violate the second law of thermodynamics. As you may recall, Maxwell's demon is an imaginary being that was deliberately constructed by James Clerk Maxwell in 1875 to violate the second law of thermodynamics. He envisioned a miniature demon which could extract energy out of a gas cylinder initially at equilibrium by separating the fast and slow molecules into the two halves of the cylinder.

[Figure showing Maxwell's demon]

Whether such a beast is possible or not eventually boiled down to the question of what information the demon could extract from the system it was observing. It was realized that by performing a measurement on each gas molecule it saw. to determine its velocity, the demon could indeed separate the gas into hot and cold molecules on either side. However. as Landauer noted in 1962, to do this, after each measurement the demon is required to reset its internal state. so as to forget the results of its previous measurement. *This act of erasing information* *increases the demon's entropy by precisely the same amount taken away from the gas molecules in the cylinder.*

This key realization, that information erasure corresponds to an increase in entropy, is known as Landauer's conjecture. More colloquially, we may say that information loss leads inevitably to energy dissipation.

During this period, electronic computers and quantum-mechanics were both introduced. Also, in 1948, Shannon laid the basis for the science of information theory. First to analyze the analyze the energy dissipation of a computer was von Neumann, who estimated that on the average, $kT \ln 2$ Joules must be dissipated "per elementary act of information, that is per elementary decision of a two-way alternative and per elementary transmittal of one unit of information." What is most interesting is that for elementary boolean logic gates, von Neumann's answer coincides exactly with that expected from Landauer's conjecture. This is made obvious by noting, for example, that the AND, OR, and XOR operations are logically irreversible; they correspond to mathematically non-invertible operations.

[Figure showing logic gates, info lost, and energy dissipated]

Until 1973, it was therefore widely believed that any computer would unavoidably dissipate $kT \ln 2$ Joules per gate on average. However, Bennett[1] then realized that nontrivial computation *may* be accomplished without use of logically irreversible operations. He proved that any irreversible Turing machine may be cast into a reversible one by adding the appropriate bookkeeping information. Around the same time, Fredkin[2] also came up with a logically reversible primitive which is boolean complete. This gate, known as the Fredkin gate, has three inputs and three outputs, and the following truth table (Figure 1).

Inputs			Outputs		
A	B	C	A'	B'	C'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	1	1	1

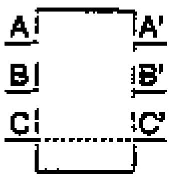


Figure 1: Fredkin gate schematic symbol and truth table.

This gate is particularly interesting, and I shall shortly return to discuss its properties in detail. In 1982, Fredkin and Toffoli introduced a concrete physical model for a reversible computer, based on collisions between billiard balls and appropriately placed mirrors. The fascinating thing about this model is that it is *manifestly* reversible, insofar as its operation is based on the laws of classical mechanics.

[Figure showing Interaction Gate]

The billiard ball model of computation is known as a "ballistic" computer because it depends on perfect operation, and the absence of external perturbations such as those caused by thermal noise. In contrast, Bennett introduced the notion of a "Brownian" computer[3], which depends on the agitation caused by thermal noise to propel a computation forward.

Finally, since 1982, there has been significant progress in extending the realm of reversible computing to the quantum domain: Benioff[4], Deutsch[5], and Feynman[6] have proposed several forms of a quantum-mechanical computer, whose operation depends principally on the Hamiltonian evolution of a state vector. I shall return to this subject later.

The central conclusion reached by researchers in this field thus far is that reversible, dissipationless computers are certainly possible in theory, although in practice, energy dissipation will certainly be required for system stability and noise immunity. We also now understand a key principle – reversibility stems from keeping track of every bit of information in a computer.

1875 Maxwell's demon introduced.

1949 Von Neumann estimated that kT ln2 (3E-21 Joules) must be dissipated "per elementary act of information, that is per elementary decision of a two-way alternative and per elementary transmittal of one unit of information."

1929-1962 Maxwell's demon is exorcised.

1961 Rolf Landuer attempted to prove von Neumann's answer. But he found only that logically irreversible operations generate a local increase in the entropy equal to the information thrown away.

1973 Bennett realized that nontrivial computation *may* be accomplished without use of logically irreversible operations.

1982 Fredkin published his reversible logic gate primitive, and showed that is is boolean complete. Fredkin and Toffoli also devised a model of computation based on the collision of "billiard balls." This was the first model of a ballistic computer.

1982 Bennett introduced his "Brownian computers" which depend on thermal noise to agitate a machine toward completion of a computation. In contrast to the ballistic models, thermal noise is an integral part of the computational process; in fact, the ballistic computer can only function properly in the complete absence of thermal noise.

1982-1986 Study of quantum-mechanical reversible computers by Benioff, Feynman, Zurek, Deutsch, Landauer, etc.

3 Overview of My Research

Now I would like to introduce my own research. I have been personally interested in reversible computers since I was an undergraduate at MIT. My education has actually been in quantum field theory and computer architecture, while semiconductor device physics is something I have been learning only since coming to Stanford two years ago. My discussion will be oriented towards fundamental physics and system issues, but I will be happy to entertain questions or comments from a different perspective as well.

3.1 Goal

The concept which interests me is a relation governing computation which has been hinted at in the literature, but never explored quantitatively – namely, that there exists some tradeoff between time, energy, and reliability. This relation has been expressed as the so-called "Spreng Triangle[7]," shown here.

[Figure showing Spreng Triangle]

The three vertices on this triangle represent the three extremes of zero time, energy, and information, while the three edges correspond to maximal information, time, and energy. Spreng philosophically noted that it is the starving philosopher who seeks the least costly solutions by acquiring maximal information, while the primitive savage is content to expend as much energy as is needed to solve the problem, and modern man is concerned primarily with a quick fix.

I am motivated in this study by our current understanding of some of the possible limits. For example, we know for a fact that given infinite time, we can perform an arbitrary calculation to a specified reliability, with zero energy. Specifically, Bennett has shown that the required energy dissipation per step can approach zero as long as a reversible computer is operated adiabatically. However, this is not so interesting in practice. In reality, the central issue is how to do some useful computation with a finite size machine, at finite temperature, in a finite amount of time. Given these constraints, how much energy must we dissipate to perform the calculation, and how fast may each logical step be performed, at best? The ultimate goal of my study is to answer these questions.

3.2 Approach

My approach towards understanding the *fundamental* physical limits to computation is based on two efforts – first, the construction of a new mathematical theory for describing the quantum-mechanical embodiment of the simplest ideal *physical* logic gate, and second, the actual experimental implementation of a simple cascade of reversible logic gates in a mesoscopic system.

Development of a quantum theory for reversible computing begins with the description of the purest physical model for the elementary building block of the ideal computer. Cascading ideal logic gates into a complex system will then give a model which may be studied to ascertain the performance achievable under specific non-ideal conditions, such as in the presence of thermal noise. Finally, by coupling each ideal logic gate to noise reservoirs, the amount of dissipation required to stabilize the behavior of the system may be determined.

Experimental implementation of an ideal logic gate is an ambitious goal. The principal problem is that an ideal logic gate is a closed system with few degrees of freedom, and that is hard to achieve in practice. This area of research represents work in progress for me. My group at Stanford University specializes in the study of noise in mesoscopic systems. We will be exploring the physics of two-dimensional electron gases in the GaAs/AlGaAs material system at temperatures below 20 millikelvin, using a dilution refrigerator. My thesis advisor and group leader is Professor Yoshihisa Yamamoto, who is well-known for establishing the field of squeezed light semiconductor laser diodes. Our current idea for fabricating a Fredkin gate involves possibly adapting a version of Kouwenhoven's single-electron turnstile device.

[Figure of logic gate cascade and Kouwenhoven's turnstile]

4 Applications of reversible logic

Before continuing with the description of my own research, I will now describe in greater detail what reversible logic is, and how it may be applied in practice.

4.1 Conservative invertible logic

What is a Fredkin gate? There are many interpretations of this device. Consider once again its truth table. The most important characteristics of its transfer function are that 1. the number of "one's" is conserved, 2. the mapping is one-to-one onto, that is, one unique output exists for each input and vice versa. These properties, the existence of an additive conserved quantity, and the invertibility of the transform, are the basis for describing the Fredkin Gate as a conservative invertible logic gate.

Inputs			Outputs		
A	*B*	*C*	*A'*	*B'*	*C'*
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
0	1	1	1	0	1
1	1	0	1	1	0
1	0	1	0	1	1
1	1	1	1	1	1

Figure 2: Truth table for a Fredkin gate classified according to bit class (number of one's).

Why are these properties special? As Fredkin and Toffoli explain in their 1982 paper, these symmetries are motivated by principles fundamental to almost all physical phenomena we understand. The conserved quantity is energy (for example), and invertibility corresponds to microscopic reversibility.

The Fredkin gate is also special in that it is boolean complete and construction universal. Since it may perform either an AND or an OR function when configured appropriately, it can be cascaded to construct any boolean function. Second, the Fredkin gate can be used to perform crossover and straight-through routing, which are the two routing functions necessary to allow construction of arbitrary routes through a regular graph. That this property holds follows from

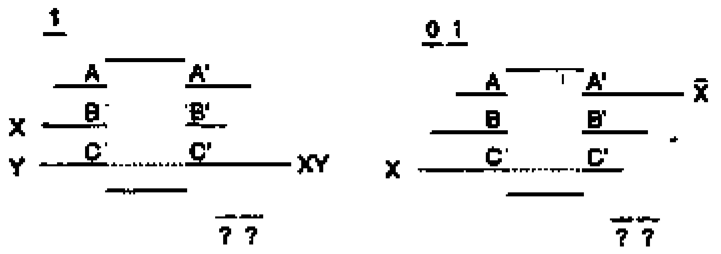


Figure 3: Fredkin gate configured to perform two elementary boolean functions. The left figure shows the AND operation, and the right, the NOT operation.

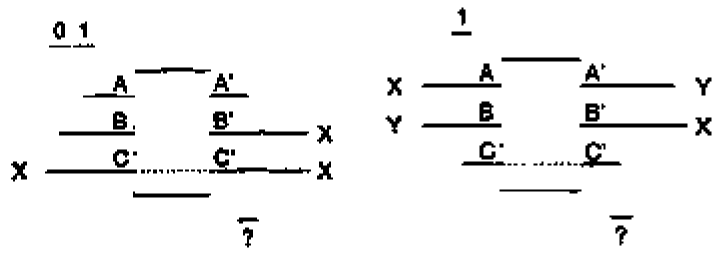


Figure 4: Fredkin gate configured to perform the two operations needed to be construction universal. The left figure shows the fanout operation, and the right, the crossover operation.

yet another fundamental physical principle – that duplication, i.e., fanout, is unnatural, and is an operation that must be accounted for explicitly.

One more interpretation of the Fredkin gate truth table exists. Mathematically, it is useful to view a conservative invertible logic gate as a data-dependent group transformation operation. For the three-port Fredkin gate, by classifying the inputs by number of "one's," we see that the transformation applied to get the proper output is just a simple permutation, with the particular permutation which is performed being a function of the number of "one's." In this case, all bit-classes are transformed identically except for the two-bit case for which these two entries are permuted.

4.2 Applications of CI logic

Conservative invertible logic can be used to reduce energy consumption in two ways. First, energy can be saved by constructing computational systems from CI logic primitives such as the Fredkin gate. This allows unused "one's" to be recycled, thus lowering the amount of energy dissipated. However, achieving logical reversibility will never lower energy dissipation beyond the inherent imperfection of the physical devices used. Thus, the next step is to replace dissipative logic devices (such as MOSFETs) with charge-recover devices, then eventually with ballistic devices, such as Milburn's quantum optical Fredkin gate.

[Figure – "Reducing Energy Dissipation with CI Logic"]

The first step is essentially one of software technology. We know that it is always possible to embed a logically irreversible calculation into a logically reversible one. As Bennett and Fredkin have shown, and as should be obvious from the boolean completeness of the Fredkin gate, an irreversible function may be calculated reversibly and repeatedly by performing the calculation using auxiliary storage, copying the desired result, then performing the reverse calculation to restore the storage to its initial state.

[Figure – Bennett's reversing computer & HPP kernel]

The problem of how much auxiliary storage is required is known as the "garbage collection" problem in reversible computing. However, I believe that this problem is not fundamental. One fundamental attraction of using conservative invertible logic lies in the notion that since physical phenomena are microscopically reversible, it is reasonable that reversible microscopic algorithms for numerically simulating them should also exist. Therefore, CI logic would form natural primitives for implementing, either in hardware or in software, such simulations. The conclusion is that if problems are formulated properly, dealing with garbage should not be a problem.

One immediate technological possibility utilizing CI logic is a minimal-energy programmable logic array. The device itself could be fabricated with CMOS transistors, and operated with a reversible power supply, it could possibly dissipate asymptotically zero power as a function of clock frequency. Of course, the difficulty is in the optimal reduction of the application program to CI logic primitives. This mathematical problem is similar in difficulty to that of normal boolean reduction, but research has showed that certain simplifications may arise from inherent symmetries due to the properties of CI logic. In fact, reduction into CI logic primitives is an interesting problem in its own right; recently, it has been shown that the collision kernel for hydrodynamic lattice gas simulations. which is normally implemented as a lookup table, may be simplified significantly when implemented using CI logic primitives in an application-specific IC.

A fascinating concrete example is Barton's 1978 implementation of a subset of the PDP-10 processor using conservative logic. Shown here is a diagram of his basic machine structure, taken from his paper. There is a random access memory M, a stack memory P, 24-bit instruction I, 24-bit accumulator AC, and garbage stack G. Several interesting things may be noted from his study; first, most instructions may actually be implemented in a reversible manner. The only irreversible functions provided were AND. OR. LOAD. CLRAC. RSH. LSH, and ABS, and it is not clear that those instructions necessarily had to be irreversible (except as a matter of convenience). Other insights gained were that it is necessary for a subroutine to have only a single return point. since the path of execution must be retraceable. Also, each memory operation had to be a read/write: non-destructive reads were not possible.

[Figure: diagram of Barton's PDP]

4.3 Physical implementations of the Fredkin Gate

I now turn to the description of the physical devices used to implement conservative invertible logic gates such as the Fredkin gate. There are two kinds of Fredkin gates, the "quantum" Fredkin gate, and the "demon" Fredkin gate. The difference is that the quantum one is to a very good approximation, a perfectly closed system. while the demon gate is a collection of dissipative gates (such as a MOSFET) *emulating* the logical operation of a Fredkin gate. Behaviorally, they are also distinguished by their noise properties. The demon gate. because it is constructed from irreversible primitives. is manifestly stable. but it may never be dissipationless, no matter how perfect the device is. Examples of some proposed demon Fredkin gates are listed here.

Demon Fredkin Gates

- Likharev: Josephson Junction, *Int. J. Theor. Phys. 21, 311 (1982)*
- Caulfield: Liquid Crystal Modulator. *Applied Optics 28, 2429 (1989)*
- Merkle: Reversible Charge Transfer. *Nanotechnology 4. 21 (1993)*

On the other hand, the quantum Fredkin gate is the real thing. In the limit of perfect physical implementation, the device is expected to operate in a dissipationless manner. Because of this, quantum coherence will persist between devices. and partition noise will arise. Another way to understand this effect is to consider it to be simply due to quantum interference. In fact, coherent effects are key to the operation of such devices. Several potential quantum Fredkin gate proposals are listed here. Note that no quantum Fredkin gate has yet been experimentally demonstrated.

Quantum Fredkin Gates

- Milburn: Quantum Optical, *Phys. Rev. Let. 62, 2124 (1989)*
- Islam, Soccolich: Billiard-ball solitons, *Optics Let. 16, 1490 (1991)*
- Huang: Fiber Logic Sagnac, *Applied Optics*, to appear *(1994)*
- Lloyd: Pulsed Arrays, *Science 261, 1569 (1993)*

5 Open Questions

The quantum Fredkin gate is perhaps the most interesting device to have emerged from the study of reversible computing. Investigation into the inherent nature of this device has connected the disciplines of computing, physics, and information theory, motivating a wide variety of open questions. In this last part of my talk, I will describe some of the preliminary results from my research on the quantum Fredkin gate.

To recapitulate, my goal is to establish a quantitative relation between fundamentally inevitable dissipation and the reliability and speed of a computing machine. My approach is to devise a physical description of the elementary building blocks of a dissipationless machine, then to study the limits on its operation as external effects such as thermal noise are introduced. To begin with, I note that the Fredkin gate is an ideal quantum logic gate. In fact, I postulate that the proper quantum description of any logic gate consists of two components, one being the Fredkin transform, and the other, a coupling to an external reservoir (corresponding to dissipation). By studying this separation, I hope to understand why dissipation seems to be essential for system stability, even at the quantum level. Related to this is the understanding of quantum measurement, as a simple quantum Fredkin gate gedankenexperiment shows.

5.1 The ideal quantum logic gate

I begin with the following observations. The ideal computer is a reversible one, and it may be constructed from Fredkin gates. In fact, the simplest of the "ideal" logic gates is the Fredkin gate, (or any of the equivalent conservative invertible three-port cousins), since all higher order conservative logic gates may be constructed by cascading Fredkin gates. That no simpler reversible logic gate exists is proved by Fredkin.

To capture the physical performance limits of an ideal logic gate, we must embody its logical behavior in a physical system. One of the simplest systems we may choose is the interaction of three harmonic oscillators described by the usual quantum-mechanical picture with linear coupling. Although this theoretical model is quite simple, I will show that it indeed properly describes the expected performance of a real physical device, that of Milburn.

Let me begin by explaining the operation of Milburn's quantum optical Fredkin gate, shown here. The basic structure of the device is a Mach-Zehnder interferometer, constructed from two 50/50 beamsplitters and two perfect reflectors. Because the path lengths of both arms are identical, the two incoming beams travel the same distance, then recombine; in the absence of any control signal, the outgoing beams are exactly the same as the incoming ones. When a control signal is present, however, the path length of the upper arm is increased, and the interferometer becomes unbalanced. The device is adjusted to operate in a manner such that when a control signal is present, the arms become unbalanced by 180°, so that the input signals are perfectly switched to give the outputs.

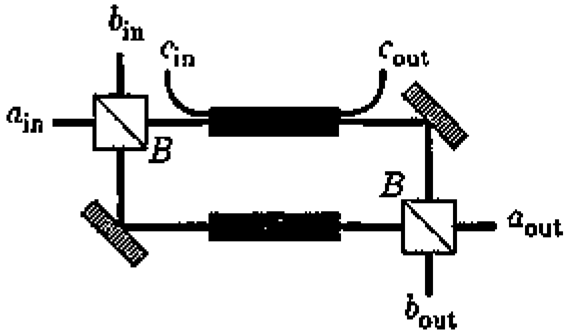


Figure 5: A prototypical Mach-Zehnder based Kerr effect Fredkin Logic Gate.

The un-balancing occurs because of the physics of the "Kerr" medium; this is a χ^3 nonlinear crystal whose index of refraction is proportional to the total electric field intensity in the medium. In other words, the more light going through a Kerr medium, the faster it travels. Mathematically, we say that the input states undergo self and cross-phase modulation; the phase of the light is changed by a function of the total number of photons present in the crystal when the light beam passes through it.

[Figure: kerr medium]

Coming back to Milburn's gate once more, we see that if we have Schrödinger picture operators for the beamsplitter and the Kerr media, then we can write down a unitary transformation operator which describes the logic gate. The logarithm of this transformation will then give us the Fredkin gate Hamiltonian. We proceed with this program by using the following operator description of a beamsplitter, as described by Yurke and others

$$B = \exp\left[i\theta\left(a^\dagger b + ab^\dagger\right)\right] . \quad (1)$$

The key to this description is that a beamsplitter can be seen as a SU(2) group transformation operator, which performs a rotation in the space of its inputs, a and b.

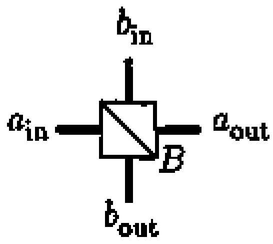


Figure 6: Quantum-mechanical Beamsplitter

Finally, putting all our mathematics together gives us an operator description of Milburn's quantum optical Fredkin gate:

$$\begin{aligned} |\text{out}\rangle &= \dot{\phi}_b\dot{\phi}_c B_1 X B_2 K B_2 B_1|\text{in}\rangle && (2)\\ &= \exp\left[-i\chi\left(n_a(n_a-1)+n_b(n_b-1)+n_c(n_c-1)\right)\right] \\ &\quad \times\exp\left[-i\chi\left(2n_a n_b+n_c(n_a+n_b)\right)\right]\exp\left[i\chi n_c\left(a^\dagger b+ab^\dagger\right)\right]|m\rangle_a|n\rangle_b|p\rangle_c . && (3)\end{aligned}$$

Here, the operator K is the Kerr medium transformation, while B_1 and B_2 are the beamsplitter transforms. For number-eigenstate inputs, the output is found to be this expression. The first two exponentials are simply irrelevant phase transforms, corresponding to self-phase modulation and cross-phase modulation. The third exponential is the real heart of the logic operation. It is what we identify as the quantum Fredkin gate operator.

What does this expression mean? It is a *controlled beamsplitter*. The effect of this operator is to perform a rotation in the SU(2) space of a and b, by the angle χn_c. That is, the rotation angle is determined by the field strength of the control input. The constant χ is an engineering parameter determined by the strength of the Kerr media, and can be chosen so that when the control reaches the appropriate strength, the logic gate is switched *on*; when no control signal is present, the gate is naturally switched off.

5.2 The Fredkin gate operator

This operator expression for the quantum Fredkin gate is a significant result. The logarithm of the Fredkin gate transform immediately gives us the Hamiltonian

$$H = \chi c^\dagger c(a^\dagger b + b^\dagger a). \qquad (4)$$

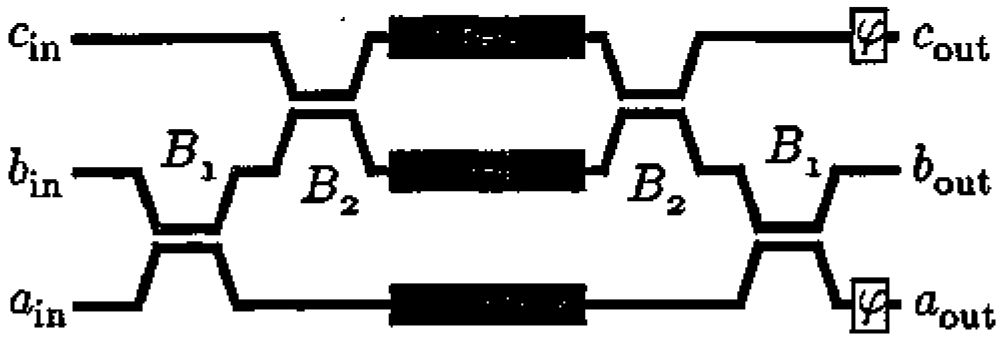


Figure 7: Three-input Kerr medium extension of the non-linear Mach-Zehnder interferometer which works as a quantum-optical logic gate.

This is the Fredkin gate Hamiltonian, a beautiful concrete example of Deutsch's Hamiltonian[5]. It is, to my understanding, the first complete Hamiltonian description of an actually physically realizable quantum Fredkin gate. The implication of this work is that since all conservative invertible logic gates are related through an equivalence transform, therefore in fact *all* quantum logic gates may be described by the Fredkin gate Hamiltonian. The proof is the same as that for the universality of the Fredkin gate.

Note that although the formalism presented here was developed with boson operators, it generalizes immediately to fermions. In fact, it has been shown by Kitagawa that closely located electron waveguides allow electron cross-phase modulation to occur; thus, the nonlinear Mach-Zehnder interferometer structure used for the quantum optical Fredkin gate could equally well be used for a ballistic electron Fredkin gate, in principle (unfortunately, it is difficult to fabricate with present technology).

The Hamiltonian for non-ideal logic gates will have the same form of Eq.(4), but the coupling may be different. For example, a transistor with dissipative source and drain, and a ballistic gate, might be described by the Hamiltonian

$$H = \sum_n \chi_n c^\dagger c \left(a^\dagger b_n + b_n^\dagger a \right). \tag{5}$$

The coupling here takes on the form of a Caldeira-Leggett dissipative coupling to a reservoir with operators b_n. This expression, however, is still tentative, and continues to be the subject of active investigation.

5.3 Quantum measurement and the Fredkin gate

My last subject deals with the relation between quantum measurement theory and the quantum Fredkin gate. The interesting question to ask is, what is the simplest level at which one may ask an "if-then" question? That is, for example, "if particle A has momentum P then eject an electron." Such questions are naturally part of a logic gate's operation; physically, such questions correspond to performing a measurement, then acting on the result.

The key realization is that the "if-then" experiment is *not* possible if only two states are correlated; a three-body interaction is essential. It is simple to see that a two-body interaction is insufficient; for example, when the polarizations of two photons are correlated with each other then sent in opposite directions (this is the photon twin experiment), measurement of one photon conveys *no information* about the other photon, even through it collapses the superposition state arbitrarily. No information is encoded into the original polarizations, and therefore superluminal communication is impossible with photon twins.

On the other hand, if three photons are allowed to interact at the origin, then nontrivial information transfer seems to be possible. The following example is due to J. Jacobson[2]. We prepare the input states $A = |\phi\rangle_a$, $B = |0\rangle_b$, and $C = (|0\rangle_c + |1\rangle_c)/\sqrt{2}$ and feed them into a Fredkin gate, with Hamiltonian

$$H = \frac{\pi}{2} c^\dagger c \left[a^\dagger b + b^\dagger a\right] . \tag{6}$$

The output state is

$$|\mathrm{out}\rangle = \frac{1}{\sqrt{2}} \Big[|011\rangle + |100\rangle\Big] , \tag{7}$$

a macroscopic superposition state; measurement of the photon in one of the three modes collapses the wavefunction, leaving the other two modes in a mixed state. Say A' and B' (primes denote output variables) are sent away to Antares, while the C' output is kept locally. If C' is left unmeasured, the detector at Antares finds A' and B' to be in a superposition state, while on the other hand, if C' is measured, A' and B' are found to be in a mixed state. Thus, measurement of C' would seem to change the statistics of A' and B', *faster than the speed of light.*

At present, the paradox of this gedankenexperiemnt has not been satisfactorily resolved. Superluminal communication should not be possible, and yet this example would seem to show that it is. The prevalent belief is that an answer lies in the definition of a measurement process. Measurement, and the consequential von Neumann reduction of the wavepacket, occurs only after a contract is signed between two interacting systems that stipulates they will never interact again. That is, their mutual information is discarded. Thus, the resolution of Jacobson's paradox lies in viewing the separation of the two signals as an implicit measurement,

[2] Private communication, 1993

which automatically collapses the three-photon superposition state, and destroys their mutual information.

6 Conclusion

The study of reversible computers probes a fertile new juxtaposition of the worlds of quantum physics, computing, and information theory. Fundamental insights from this field promise to bring new understanding to the definitions of measurement, computation, and dissipation. Hopefully, through the creation of theories to explain quantum logic gates, and experiments to test the avoidability of dissipation, we will eventually develop not only new computational machines and paradigms, but also quantitative limits for the computational performance of the physical world.

References

[1] C. H. Bennett. Logical reversibility of computation. *IBM J. Res. Dev.*, 17:525, 1973.

[2] Edward Fredkin and Tommaso Toffoli. Conservative Logic. *International Journal of Theoretical Physics*, 21(3/4):219, 1982.

[3] C. H. Bennett. The thermodynamics of computation – a review. *Int. J. Theor. Phys.*, 21:905, 1982.

[4] Paul A. Benioff. Quantum Mechanical Hamiltonian Models of Discrete Processes That Erase Their Own Histories: Application to Turing Machines. *ijtp*, 21(3/4):177, 1982.

[5] D. Deutsch. Quantum computational networks. *Proc. R. Soc. Lond.*, A425:73–90, 1989.

[6] R. P. Feynman. Quantum Mechanical Computers. *Optics News*, (February):11, 1985.

[7] A. M. Weinberg. On the relation between information and energy systems: A family of Maxwell's demons. *Interdisciplinary Sci. Rev.*, 7:47–52, 1982.

DYNAMIC TASK MIGRATION FROM SPMD TO SIMD VIRTUAL MACHINES

James Armstrong, Purdue University

James B. Armstrong[†], Howard Jay Siegel[‡], William E. Cohen[‡], Min Tan[‡],
Henry G. Dietz[‡], and Jose A. B. Fortes[‡]

[†]Sarnoff Real Time Corporation
Princeton, NJ 08543-5300 USA

[‡]Parallel Processing Laboratory
E. E. School, Purdue University
West Lafayette, IN 47907-1285 USA

Abstract -- *A method to migrate a task dynamically from a virtual SPMD machine to a virtual SIMD machine is proposed. It is assumed that the SIMD and SPMD virtual machine models only differ to support the different modes of parallelism, and that the program was coded in a mode-independent programming language. The migration procedure does not require the SPMD PEs to be at the same location in the SPMD program at the time of the migration. This work is directly applicable to mixed-mode hybrid SIMD/SPMD systems and part of the general problem of task migration in SIMD/SPMD mixed-machine heterogeneous systems.*

1. INTRODUCTION

In a heterogeneous system [11, 30], different types of parallel machines are interconnected by high-speed links. Task migration in a such an environment may be necessary for fault-tolerance, load balancing, administrative reasons, or improving execution time of a single task. The migration procedure "captures" a program's execution state on one type of machine and then maps it to a viable state on a different type of machine. When task migration is performed in the context of fault-tolerance, the "capturing" of the state is done periodically at checkpoints, and the mapping is done at the time of the fault.

One possible approach to migrating a task dynamically between a synchronous SIMD machine and an asynchronous single program – multiple data stream (SPMD) machine is illustrated in Fig. 1. In general, SPMD mode is the use of a MIMD machine when all PEs execute the same program, but asynchronously with respect to one another. A task is assumed to be coded in a hypothetical mode-independent programming language, referred to here as the VPL (virtual programming language) (e.g., ELP [20], HPF [15], Paralation Lisp [5], XPC [21]). The hypothetical VPL compiler generates object code for each machine or a subset of machines on a network, as well as produces information necessary for the task migration procedure. The dotted arrows in Fig. 1 indicate that from a single VPL source program each machine's executable program is generated.

To move a task from some physical machine (A, B, ... F) executing in one mode of parallelism to another physical machine executing in another mode of parallelism, two types of transformations are performed that rely on information generated by the VPL compiler. The first type of transformation (dashed arrows in Fig. 1) maps the execution state of the task on a particular machine to/from an execution state on a virtual machine model, which represents all machines with the same mode of parallelism. The different shapes of the machines depict their differing physical architectures. Mechanisms for migrating tasks between different single-processor computers, which can be applied to specifying this first type transformation, have been proposed (e.g., [10], [14], [23], [26], [27], [29]).

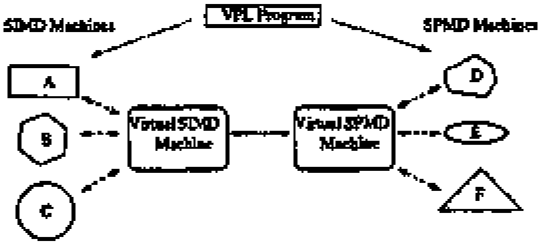


Fig. 1: Graphical depiction of task migration between SIMD and SPMD machines.

The second type of transformation (solid arrow in Fig. 1) conceptually migrates the task between a state on the virtual SIMD machine and a state on a virtual SPMD machine. The similarity of shapes between the virtual SIMD machine and the virtual SPMD machine represents that the conceptual architectures only differ to support the different modes of parallelism. The work described here addresses the SPMD to SIMD portion of the second type of transformation. It proposes a method by which a multiple instruction stream SPMD program can be mapped to a single instruction stream SIMD program. The approach taken is to characterize a single instruction stream program and a multiple instruction stream program as programs that execute on a virtual SIMD machine (Subsection 2.2.2) and virtual SPMD machine (Subsection 2.2.3), respectively.

A general approach to implementing the second type transformation was proposed in [8]. It discusses a way to transform any MIMD program into pure SIMD code. It

This work was done when James B. Armstrong was at Purdue University and funded by a NASA fellowship under grant number NGT-50961. This research was also supported by Rome Laboratory under contract number F30602-92-C-0150, by NRaD under contract number 91-D-1799, and by the National Science Foundation under grant number CDA-9015696.

does this by having the SIMD code (running on an SIMD machine) emulate the MIMD program. One of the goals here is to have the VPL compiler generate efficient code for each of the source and destination machines (i.e., the generated code is specifically targeted for each machine). The task migration procedure maps a point in the efficient code for one machine to a point in the efficient code for another machine.

This research proposes a method by which a point in an SPMD program can be mapped to a point in an SIMD program, assuming that the machine models only differ to support the different modes of parallelism. This assumption is directly applicable to a mixed-mode system [7], in which the processors of a single machine are capable of operating in either the SIMD or SPMD (or full MIMD) mode of parallelism and can dynamically switch between modes at instruction-level granularity with relatively little overhead (e.g., OPSILA [9], PASM [4,24,25], TRAC [18], Triton/1 [22]). However, this research is primarily targeted for solving part of the general problem of task migration for heterogeneous SIMD/SPMD mixed-machine systems [30], where a suite of SIMD and SPMD systems are interconnected by a high-speed network. Although parts of the migration procedure were implemented on the mixed-mode PASM prototype as a proof-of-concept, a full implementation is beyond the scope of this paper. Section 2 states more specifically the assumptions about the programming language, operating system, and machine models. The task migration problem is stated formally (i.e., mathematically) in Section 3. The procedure to migrate a task between an SPMD and an SIMD virtual machine is presented in Section 4.

2. THE HETEROGENEOUS ENVIRONMENT

2.1 Overview

This section describes the conceptual model of SIMD, SPMD, and mixed-mode computation that is assumed here, and mentions some of the language features that are expected to be part of a mode-independent programming language (i.e., VPL). It also briefly overviews aspects of an operating system that are relevant to this study.

2.2 Virtual Machine Models

2.2.1 Overview

It is assumed that the virtual SIMD and SPMD machines are as similar as possible, differing only in the mechanism needed to support the different modes of parallelism. This implies such things as: all processors store data in the same format (e.g., byte order, number of bits), memory addresses are consistent across the machines' memory modules (i.e., valid addresses on one machine are not invalid on the other), the number of processors across machines are the same, and the inter-processor networks used in both machines are the same.

Furthermore, both the SIMD and SPMD machines are assumed to have a physically distributed memory organization. In such a system, each processor is paired with a memory module to form a PE (processing element). Most parallel systems currently in use are physically implemented as distributed memory organizations (e.g., CM-5 [13], KSR1 [16], MasPar MP-1 [6], and nCUBE2 [12]).

2.2.2 The Virtual SIMD Machine

The virtual SIMD machine is composed of a CU (control unit), P PEs, and an interconnection network. The PEs are activated if they can be used by the executing program (as explained in Subsection 4.2.3). When a PE is active, it is said to be enabled if it executes instructions. The enabled PEs receive and synchronously execute common instructions that are broadcast from the CU. The PEs fetch data from their individual memory modules. The CU has the ability to enable selectively PEs for the execution of instructions. Those PEs that are not enabled for the particular instruction being broadcast by the CU are disabled, i.e., remain idle and do not execute the instruction. The interconnection network allows PEs to communicate among themselves and exchange data. Furthermore, the CU processor is assumed to have *Creg* general purpose registers available for storing data and each PE's processor is assumed to have *Sreg* general purpose registers. Examples of existing SIMD machines with a similar structure include CM-2 [28] and MasPar MP-1.

2.2.3 The Virtual SPMD Machine

The virtual SPMD machine consists of P PEs and an interconnection network. Each PE's instructions and data are stored in its memory module. Because there are multiple threads of control, the PEs execute asynchronously with respect to one another. As with the SIMD model, the interconnection network provides communication links among the PEs. Also, each PE's processor is assumed to have *Creg + Sreg* general purpose registers available for use. The registers in the SIMD and SPMD machine models are assumed to have the same size. Examples of constructed systems with a similar structure that are capable of SPMD execution include the CM-5, KSR1, and nCUBE 2.

2.3 Virtual Programming Language Features

2.3.1 Overview

Many of the aspects of the hypothetical mode-independent language, VPL, are based on the existing ELP language [20]. The overriding concern with VPL is to insure that the language definition is mode independent. Language constructs must be translatable to both the SIMD and SPMD models of execution. Thus, constructs that do not have a translation to both models of execution are illegal. VPL is explained here using a C syntax with extensions.

Like C, VPL has pointers. However, pointers to local variables are illegal because the migration process changes the location of variables in the stacks and the translation of the pointers to the new addresses would be an expensive operation to implement. The other semantic differences from C are owing to VPL having more than one flow of control in the program. The features unique to VPL will be discussed in greater detail in the following subsections.

2.3.2 Variable Attributes

In addition to the standard types in the C language such as int, float, and double, variables in VPL also have attributes that describe the location of the variables and the types of operations possible on the variables. These attributes are mono and poly.

A poly attribute specifies that a local copy of the variable resides in each PE in the machine. When a poly expression is being evaluated, each PE in the machine is operating on an independent copy of the poly variable that is located in its local memory.

A variable declared as mono effectively has one copy across the entire machine. In particular, on SIMD machines, a mono variable would have a single copy stored on the CU. Operations involving only mono variables and constants are executed on the CU. This may lead to better SIMD performance than if the variables were present on each of the PEs [2]. In contrast, on an SPMD machine, a local copy of the mono variable is stored on each PE.

In SIMD mode, for any operations that involve both mono and poly variables, the mono variables are broadcast to each PE and the operation is then performed in parallel on the PEs.

2.3.3 Flow of Control

If a conditional statement consists of at least one poly variable, the conditional is considered to be a poly conditional statement, otherwise it is a mono conditional statement. During execution, each PE on an SPMD machine is able to test independently local mono or poly values and branch around code that should not be executed. In contrast, only the CU on the SIMD machine can execute jump instructions and branch around sections of code. A mono conditional expression is evaluated on the CU in SIMD mode, and thus the execution of an if, for, while, or do statement is similar to the way a sequential machine would execute it. Because each PE may have different results from evaluating a poly conditional expression, each PE may iterate through the body of a for, while, or do statement a different number of times. In SIMD mode, when a PE fails a poly conditional test in a for, while, or do statement, it is disabled until all PEs have failed the poly conditional. Similarly, if the conditional statement of a if-then-else statement is a poly conditional expression, only those PEs that evaluate the condition as "true" are enabled and execute the then clause. Those PEs that evaluated the condition as "false" are disabled until the then clause has been executed by the enabled PEs. Only those PEs that evaluated the condition to be "false" are enabled to execute the else clause, and the other PEs are disabled until the else clause has been executed by the enabled PEs. It is assumed in this paper that each PE in an SIMD machine has an enable stack that stores the PE's enable status for various depth nestings of poly conditional expressions (as in, for example, the MasPar MP-1 and the CM-2).

To unify the representation of SIMD and SPMD conditional execution, the VPL compiler imposes several restrictions. One of the restrictions that was proposed for the XPC language [21] is illustrated by the poly conditional below.

```
if (PEname == 0){A} else {B}
```

The difference between SIMD and SPMD execution is the ordering of the execution of statements A and B by the PEs. On an SIMD machine, statement A would execute before statement B, but on an SPMD machine, statements A and B may execute concurrently. In VPL, unless it can be guaranteed that the execution order will not affect the results of the if-then-else statement, the SIMD semantics are enforced.

2.3.4 Inter-PE Communication and Synchronization

The operations that can affect ordering of statement execution across PEs are inter-PE communications and synchronization operations. In SIMD mode, typically when one PE sends data to another PE, all enabled PEs send data to other distinct PEs. Therefore, the "send" and "receive" commands are implicitly synchronized. Because all enabled PEs are following the same single instruction stream, each PE knows from which PE the message has been received and for what use the message is intended. Thus, no buffering of messages or explicit message identification is needed. Conversely, an SPMD mode program is executed asynchronously among all PEs. As a result, the PEs must execute explicit synchronization and identification protocols for each inter-PE transfer. In addition, because a specific ordering of messages cannot always be guaranteed, messages need to be buffered. It is assumed that library routines implement the various communication protocols. For the migration process discussed, it will be necessary for an intermediate SIMD program to make use of the explicit synchronization and identification protocol, as well as the buffering mechanism, which is normally associated with SPMD transfers.

The approach used for updating mono variables in VPL, which is the same as in ELP, is to disallow assignments to mono variables within poly conditional statements because the coherence of the mono variables cannot be guaranteed. This implies that at any point in time, a mono variable may have different values on different PEs in an SPMD machine; however, it will have the same value across PEs at the same location in the SPMD program.

2.4 Operating System

2.4.1 Overview

Many parts and details of an operating system must be considered for the general case of migrating a task between two machines [23]. However, this subsection focuses on the parts of the operating system that uniquely impact the migration of a program between two machines that have different modes of parallelism.

2.4.2 Memory Layout

One aspect of the operating system that is pertinent to this study is the virtual address space. It is assumed that mono and poly global variables share the same virtual address space in each PE, but are grouped into separate memory segments. (In SIMD machines, the mono variable virtual addresses map to CU memory locations.) Separate mono and poly "heap" data segments also exist for dynamic memory allocation. By making a distinction between mono and poly data segments, the modification of the virtual address tables is simplified.

A VPL subroutine can have both mono and poly parameters and local variables. In SIMD mode, upon the call of a subroutine, stack space for mono parameters, mono local variables, and subroutines' return addresses is allocated on the CU. The memory space for poly parameters and poly local variables is allocated on the PE stack, as in, for example, the MasPar MPL programming language [19]. In SIMD mode, it is assumed the CU and each PE has a frame pointer that points to the locally stored stack (e.g., PASM prototype). In SPMD mode, mono and poly parameters and local variables are pushed onto the PE stack. Stack and frame pointers are kept in each PE.

Ideally, within the user stack, mono and poly local variables and parameters should have separate segments as well. However, because local variables and parameters are stored on the run-time stack, separate stacks would be required. This implies that the SIMD and SPMD machines have separate stack pointer registers available to be used for a poly variable stack and a mono variable stack. While some SIMD machines (e.g., MasPar MP-1) may have this feature, in general SPMD machines do not. Thus, mono and poly local variables and parameters are assumed to occupy the same memory segment in this discussion.

2.4.3 Program Migration

When a signal to migrate the program is received by the SPMD machine, the operating system must save the state of the program so the program can be restarted at the appropriate point on the SIMD machine. In addition to the memory image, the operating system stores other information, such as: messages to processes on the same PE and different PEs that have not been sent, messages received by a process but not read, and inter-process and inter-PE communication paths established. The operating system is also assumed to have the capability of "flushing" messages from the inter-PE network (e.g., CM-5 operating system [13]). If the network is a packet-switched multistage network, some packets may be blocked within the network at the time a process is interrupted to be mapped to another machine. In this case, the operating system "flushes" the network of the messages, so that all messages are saved as part of the "per process" information mentioned above.

3. MATHEMATICAL MODEL

3.1 Goal

Let a VPL program, $\underline{P}$, be compiled to produce an object code program, $\underline{S}$, for a virtual SIMD machine and an object code program, $\underline{M}$, for a virtual SPMD machine. Either in SIMD mode (due to enabling/disabling) or SPMD mode (due to branching), not all PEs will necessarily execute the same sequence of instructions. $\hat{S}_{1:\sigma}(x)$ denotes the sequence of states representing the collective actions of all PEs that occur during the execution of the entire SIMD program with input $\underline{x}$, starting with state 1 and ending with state σ. $\hat{M}_{1:\mu}(x)$ is defined similarly for the SPMD program. This paper describes a transformation $\underline{H_M}$, implemented in SIMD mode, SPMD mode, or mixed-mode, such that $\hat{S}_{i:\sigma}(H_M(\hat{M}_{1:j}(x))) \equiv \hat{S}_{1:\sigma}(x)$, for $1 \le i \le \sigma$ and $1 \le j \le \mu$, and for all x. The equivalence statement means that if $\hat{M}_{1:\mu}$ is interrupted at some point having executed the sequence of states $\hat{M}_{1:j}$, then H_M can transform the results computed by $\hat{M}_{1:j}(x)$ to a form that can be processed by $\hat{S}_{i:\sigma}$ (executed in SIMD mode), so that the result is the same as that found by $\hat{S}_{1:\sigma}$. In other words, H_M transforms the results of $\hat{M}_{1:j}$ to yield a valid state of $\hat{S}_{1:\sigma}$. No particular mode of parallelism is specified for implementing H_M, because it can be performed totally in SIMD or SPMD mode, as well as partially in either mode (i.e., mixed-mode). More details about the mathematical model are in [3].

3.2 VPL Program Characteristics

Because the two machines support two different modes of parallelism, there may be points in the execution of the program on one machine that do not correspond to points on the other. For example, in SIMD mode, a mono variable may need to be broadcast to each PE to be added to a poly variable. The "broadcast and addition" operation may require a sequence of instructions in the SIMD code, but only a single addition instruction in the SPMD code. The state of the SIMD machine during the execution of these instructions may not correspond to a state in the SPMD machine. Another example is that the SPMD program may need to identify transferred information explicitly, which is not necessary for SIMD transfers. The SPMD machine state while performing the identification protocol may not be equivalent to any SIMD machine state.

To make all possible interruptible points in both programs "equivalent," the VPL compiler divides the object code programs S and M into uninterruptible blocks of instructions with the following properties: (1) there are an equal number of blocks, B, generated from S and M, (2) the function implemented by block j from S and block j from M are equivalent for $1 \le j \le B$, and (3) no block in either S and M can be further divided into blocks such that properties (1) and (2) are true for the resulting blocks when $B > 1$. By constructing programs S and M with uninterruptible blocks of instructions in this way, for any interruptible point in M there is an equivalent point in S. An algorithm that performs this function is given in [3].

4. SPMD TO SIMD

4.1 Overview

This section discusses the transformation of the suspended state of an interrupted SPMD program, *M*, to a state in an equivalent SIMD program, *S*. (Recall that programs can be interrupted only at block boundaries, as defined in Subsection 3.2). Mathematically, a function H_M was presented in Subsection 3.1, such that $\hat{S}_{i:\sigma}(H_M(\hat{M}_{1:j}(x))) = \hat{S}_{1:\sigma}(x)$, for $1 \le i \le \sigma$ and $1 \le j \le \mu$, and for all x. Here, one possible implementation of H_M is described. An estimated worst-case asymptotic time complexity for each part of the migration procedure is given. The actual time complexity of the implementation presented here is application dependent.

An important design requirement of H_M is that when *M* is interrupted, execution of the program to be migrated on the SPMD machine must end "quickly." This is desirable because it allows the prompt migration of tasks by load balancing routines. In the context of fault-tolerance, interrupts can be used by the operating system to checkpoint the memory image of *M*. *M* can then be checkpointed periodically in time instead of at specific locations in the program.

There are several data structures that are assumed to be produced by the VPL compiler for each program that provide mapping information between the *S* and *M*. The mappings between the memory addresses of subroutine call instructions and between interruptible points in the SIMD and SPMD programs are kept. This information is used by H_M to map return addresses in *M* to those in *S*, and is stored in a table called the ART (address resolution table). This table, therefore, provides a mapping from an instruction address in one program to the equivalent instruction address in another (for all those addresses discussed above).

Information about which location on the PE stack is occupied by a mono variable is also kept. The VPL compiler produces this information for each subroutine. This is done efficiently if the VPL compiler assures an ordering on the stack of parameters and local variables. For example, the VPL compiler can push all mono parameters on the stack before all poly parameters; likewise, for mono and poly local variables. Then, for each subroutine, the VPL compiler associates a location on the stack that separates mono and poly parameters and mono and poly local variables. This information is stored in a SST (subroutine stack table). Also, in addition to specifying uninterruptible blocks and return address mappings, the ART maps all uninterruptible blocks to the entry in the SST for the subroutine in which the blocks appear.

Finally, each ART entry contains a pointer to the entry in the ART of the block that represents the conditional statement in whose scope it appears. If a block is not within a conditional statement the pointer is null. The pointer fields for blocks representing subroutine entry points are also null. These pointers are used to create the enable stack on each PE of the SIMD machine when the SPMD program is migrated (Subsection 4.2.3). At the time of the interrupt, the pointers are used in conjunction with the return address values on the run-time stack of each PE (showing the sequence of subroutine calls) to determine the nesting of conditionals at the time the PE was interrupted. This information can then be used to create the PE enable stack on the SIMD machine.

The discussion is divided into two parts. The first part determines the starting instruction address in *S* given an interrupted *M*. The second part presents how a viable starting state for $\hat{S}_{i:\sigma}$ is established from the state produced by $\hat{M}_{1:j}$.

4.2 Determining $\hat{S}_{i:\sigma}$ from $\hat{M}_{1:j}$

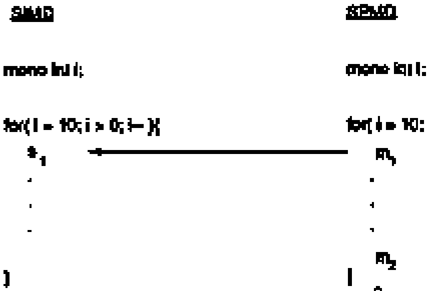
4.2.1 Overview

At the time of an interrupt of $\hat{M}_{1:\mu}$, because the PEs are executing asynchronously with respect to one another, each PE may be at a different point in the execution of the program. Due to the synchronous nature of *S*, the multiple individual PE states of $\hat{M}_{1:j}$ must be mapped to a single state in $\hat{S}_{i:\sigma}$. This is done by using a temporary intermediate SIMD program *S′*. It is assumed that the entire sequence of states of *S′* is represented by $\hat{S}'_{1:\eta}$. The SPMD machine is not used for this because a goal of the migration method is to terminate execution on the SPMD machine as "quickly" as possible. Once all individual PE states have effectively reached the same point in the *S′* program, the state of $\hat{S}'_{k:\sigma'}$ (σ' is the state of *S′* at the time of migration) is mapped to a state in the efficient SIMD program $\hat{S}_{i:\sigma}$. Thus, a two step mapping is implemented by H_M.

4.2.2 Finding the Starting State in $\hat{S}'_{k:\sigma'}$ and $\hat{S}_{i:\sigma}$

The difficulty in finding the starting state encountered in $\hat{S}_{i:\sigma}$ from the interrupted individual PE states that comprise the machine state $\hat{M}_{1:j}$ is that mono conditional loop statements are part of the VPL language. For example, if two PE states are within a VPL mono conditional loop statement, then the mono loop control variable and the end-of-loop mono conditional test need to be checked to determine which value for the mono loop control variable from the two PEs should be used for the starting state. In fact, the way the loop variable is modified by each iteration must also be known.

Fig. 2 illustrates the type of decision H_M must make. Suppose at the time of an interrupt, PE 0 was at m_1 when $i=6$ and PE 1 was at m_2, when $i=8$ (assume $P=2$). Because, the loop modifies a mono variable i (the loop control variable), the value of the mono variable at m_1 and at m_2, and the way it is modified (decremented), needs to be considered. If H_M evaluates the PC (program counter) values of PE 0 and PE 1, $\hat{S}_{i:\sigma}$ would be started at s_1 with the state of PE 0. The I value would be 6. From this starting state, $\hat{S}_{1:\sigma}$ would never generate PE 1's state (i.e., PE 1 was interrupted at m_2 with $i=8$), because *i* is decremented at each iteration. Because there is a single storage location for mono variables in SIMD mode, it is necessary to choose the correct first state in $\hat{S}_{1:\sigma}$. From this state, all the values of the mono variables in the interrupted PEs' states from $\hat{M}_{1:j}$ will be encountered in $\hat{S}_{i:\sigma}$.

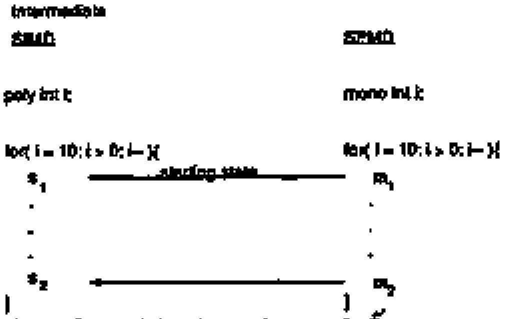


```
SIMD                              SPMD

mono int i;                       mono int i;

for( i = 10; i > 0; i-- ){        for( i = 10; i > 0; i-- ){
    s1   <--------------------      m1
    .                               .
    .                               .
    .                               .
                                    m2
}                                 }
```

Fig. 2: Determining the starting state in $\hat{S}_{i:\sigma}$.

To avoid having to check the values of the mono variables and the way they are updated, an intermediate program, S', is generated by the VPL compiler. This intermediate program is an equivalent SIMD program that has no mono variables. H_M maps the various states of $\hat{M}_{1:j}$ to a state in $\hat{S}'_{k:\sigma'}$. The advantage that S' has over S is that the number of times a loop is executed is determined by the PE and not the CU. Therefore, to map the states of the interrupted PEs in $\hat{M}_{1:j}$ to $\hat{S}'_{k:\sigma'}$, the mono variables do not have to be evaluated to determine at which iteration of the loop in the SIMD program processing begins. One disadvantage of S' is that CU/PE overlap may be reduced, which could lead to poorer SIMD performance [2]. S' is only used temporarily (i.e., it is an intermediate program). Once all the PEs have been activated and they are not executing a loop statement that corresponds to mono conditional loop statement in S, the state of $\hat{S}'_{k:\sigma'}$ is mapped to a state in $\hat{S}_{i:\sigma}$. It is possible that no point in $\hat{S}'_{k:\sigma'}$ meets this requirement given the interrupted states of the PEs, in which case S is never invoked.

Fig. 3 shows the same code segment as that in Fig. 4. Again, it is assumed that $P = 2$, and PE 0 and PE 1 were interrupted at m_1 and m_2, respectively. In this case, H_M simply starts S' at s_1 with PE 0 activated, regardless of the value of i. PE 1 is activated when $\hat{S}'_{k:\sigma'}$ reaches s_2. The loop's poly conditional expression is evaluated on each PE. At the end of the loop, assuming all PEs are activated, the state of $\hat{S}'_{k:\sigma'}$ is mapped to a valid state in $\hat{S}_{i:\sigma}$. Thus, in effect, $\hat{S}'_{k:\sigma'}$ synchronizes the interrupted states of the PEs in $\hat{M}_{1:j}$ to yield a valid state in $\hat{S}_{i:\sigma}$. The synchronization is not done on the SPMD machine because it is assumed that when an interrupt occurs, execution on the SPMD machine must end "quickly." Because synchronizing the PEs takes an indefinitely long period of time, it is done on the SIMD machine by S'. Clearly, if the original SPMD loop used a poly conditional the same technique would apply. If the loop body contains inter-PE data transfers, it is not a problem for $\hat{S}'_{k:\sigma'}$ because SPMD-like protocols are used, as mentioned in Subsection 2.3.3 and further discussed in Subsection 4.2.3.



```
intermediate
SIMD                              SPMD

poly int i;                       mono int i;

for( i = 10; i > 0; i-- ){        for( i = 10; i > 0; i-- ){
    s1   <-----starting state-----  m1
    .                               .
    .                               .
    .                               .
    s2   <------------------------  m2
}                                 }
```

Fig. 3: Determining the starting state in $\hat{S}'_{k:\sigma'}$.

The PC values of the PEs at the time of the interrupt of $\hat{M}_{1:j}$ cannot be used alone to determine where in S' computation should begin. This is because a single subroutine can be called from more than one place in a program. Consequently, two PEs can have a PC value in the same subroutine, but the subroutine may have been called from two vastly different program locations. A unique dynamic program position can be computed by evaluating the return addresses stored on each PE's stack. These values together with the current PC value specify a unique program location. Thus, to find the PE state from which $\hat{S}'_{k:\sigma'}$ should be started, the return values on each PE's stack are compared in the order that they were pushed on the stack. This can be done by a sequence of recursive doubling operations. The program location in S' that is "closest to the beginning" is chosen as the starting state.

Now, consider the time complexity of finding the starting location in $\hat{S}'_{k:\sigma'}$. Assume that the nesting depth of subroutines in $\hat{M}_{1:j}$ is d and the number of PEs is P. Then the worst case time complexity to compare by recursive doubling the return address values on each PE's stack and their PC value is $O(d \cdot \log P)$.

4.2.3 Determining When to Activate a PE

Once $\hat{S}'_{k:\sigma'}$ has been started, some PEs may be disabled because the PEs in M to which they correspond were interrupted at a point in $\hat{M}_{1:j}$ that appears further along in $\hat{S}'_{k:\sigma'}$ than the starting state. A PE is activated when $\hat{S}'_{k:\sigma'}$ reaches a program location that is equivalent to the location in $\hat{M}_{1:j}$ at which it was interrupted. However, because a PE may have been interrupted after any uninterruptible block of instructions, it would be costly to have $\hat{S}'_{k:\sigma'}$ check, after each uninterruptible block, if any PEs should be activated.

For this reason, M can only be halted after specific blocks. The VPL compiler groups blocks together making the set of blocks uninterruptible. The compiler groups as many blocks together as possible without reducing interrupt response time to unacceptable levels. For example, if the response time is set at .01 seconds, then the VPL compiler would only group blocks together that would not exceed .01 seconds to execute. If the compiler does not know how many times a loop will iterate, then at least one point in the body of the loop must be interruptible no matter how short the loop is. Subsequently, the S' program only needs to check if any PEs need to be activated at the locations that are equivalent to the interruptible points in M.

To accomplish this, the VPL compiler inserts the following code segment at equivalent points in S' as the interruptible points in M:

```
        goto SKIP;
        activate_PEs(remap, list);
    SKIP:
```

As shown, the subroutine activate_PEs() is not reached because of the goto instruction. Although this code segment introduces overhead into S', the goto corresponds to a jump instruction in a processor's object code and is expected to have nominal overhead. The amount of overhead depends upon how frequently the code segment is encountered, which is a function of the interrupt response time in M and how short some loops are.

When $\hat{M}_{1:j}$ is interrupted, H_M makes a list of all locations at which PEs were interrupted as well as which PEs were interrupted at each location. Then for each location in the list, H_M overwrites the goto instruction with a nop instruction. It also writes a value for list that is passed to activate_PEs(), which is a pointer to a linked list (other data structures can be used) of PEs that have been halted in $\hat{M}_{1:j}$ at the equivalent location. Then, whenever $\hat{S}_{k:q'}$ reaches a point when a PE may be activated it invokes the subroutine activate_PEs() passing it the list of PEs that may be activated. Not all the PEs in the list passed to it are necessarily going to be activated at each invocation of activate_PEs(). This is because the interrupted location may be within a subroutine that has been called from multiple places in the program. To activate the appropriate PEs, activate_PEs() compares the return values on the CU stack and the current PC value to that of those PEs on the list passed to it. Those PEs whose return values and PC value at the time $\hat{M}_{1:j}$ was halted correspond to the return values and PC value of the CU in $\hat{S}_{k:q'}$ are activated. When all the PEs in the list are activated by this process, the activate_PEs() routine overwrites the nop instruction with the goto SKIP for the location in $\hat{S}_{k:q'}$ from which it was invoked. This prevents $\hat{S}_{k:q'}$ from checking whether any PEs need to be activated at this location, because all have been.

As mentioned in Subsection 2.3.3, each PE has an enable stack by which it determines if it is enabled or disabled for an SIMD instruction. In SPMD mode, the PEs are never disabled and thus do not have an enable stack. Thus, activate_PEs() must establish an enable stack for the PEs that are activated. This is done by using the information in the ART in conjunction with the return address values on the run-time stack. For the PC and return address values, the ART has a pointer to the conditional statement in whose scope those instruction addresses appear. For each address, the pointers can be traced until a null pointer is found. By adding the number of these non-null pointers for each address, the conditional nesting depth of the interrupted PE can be determined. From this number (depth), the enable stack can be found [17]. If the conditional nesting depth in $\hat{M}_{1:j}$ is c, then the PEs can compute their enable stacks in parallel in worst-case time complexity of $O(c + d)$.

Given the method of activating PEs, a problem may arise when $\hat{S}_{k:q'}$ executes a poly conditional statement. Actually, all conditionals in S' are poly conditionals because there are no mono variables. Assume that the poly conditional statement disables all active PEs. This would cause $\hat{S}_{k:q'}$ to "skip" the then clause of the poly conditional. It is possible, however, that some deactivated PEs would be activated inside the then clause. By "skipping" the then clause, $\hat{S}_{k:q'}$ may leave those PEs deactivated permanently. Thus, for each poly conditional statement that evaluates condition, i.e., if(condition){A} else {B}, the following conditional test would be performed instead by $\hat{S}_{k:q'}$:

```
if(condition || (if_none() && PE_activated())){A}
if(!condition || (if_none() && PE_activated())){B}
```

The if_none() routine determines if none of the activated PEs are enabled after condition is evaluated and PE_activated() determines whether any PEs are activated within A or B. The then clause, A, is taken if condition is "true" for some PE or if no PEs evaluate condition to be "true" but some PEs will be activated in A. The else clause, B, is executed when any PEs evaluated condition as "false" or if no PEs evaluated condition as "false" but at least one PE will be activated in B.

To determine if any of the PEs will be activated in A or B, the CU routine PE_activated() uses the ART to find the address of the current if conditional and the return addresses on the run-time stack to determine the current dynamic program location. The address of the if conditional and the dynamic program location are then broadcast to the inactive PEs (the operating system can can activate them temporarily for this operation), which use the information in their suspended state to see if they were interrupted during execution of A or B in $\hat{M}_{1:j}$. If so, PE_activated() returns "true"; otherwise, it returns "false."

The added computation for each conditional is another source of inefficiency for S'. The impact of this added overhead is dependent upon how many conditional statements disable all active PEs. This number is application dependent and cannot be determined statically.

Inter-PE communication in $\hat{S}_{k:q'}$ also has added overhead. Because the PEs are not necessarily activated at the same point in the program, inter-PE messages must be buffered. Some PEs may not have reached the point in $\hat{M}_{1:j}$ where they read messages sent to them. These messages must be buffered in $\hat{S}_{k:q'}$ until the PEs read them. Furthermore, the order of the messages in the buffer is not known and thus a message identification protocol is necessary. The overhead associated with SPMD inter-PE transfers are therefore retained in SIMD mode while $\hat{S}_{k:\sigma'}$ executes.

Because of the overhead in $\hat{S}_{k:\sigma'}$, it is desirable to move to S as quickly as possible. However, the PEs must be synchronized before this happens. Synchronization is guaranteed when all the PEs are active, and $\hat{S}_{k:\sigma'}$ is not in the scope of a mono conditional statement. At the time of the interrupt, H_M computes where $\hat{S}_{k:\sigma'}$ will be halted and its state mapped to a state in $\hat{S}_{i:\sigma}$. This is determined by finding the interrupted PE state in $\hat{M}_{1:j}$ that corresponds to the state in $\hat{S}_{k:\sigma'}$ that is "farthest from the beginning" of $\hat{S}_{k:\sigma'}$. This is done the same way the "closest" state was found. Once the "farthest" state is found, H_M determines which of the interruptible locations is not within a mono conditional loop in S that is at or past the "farthest" location. For this location, a nop is written over the goto SKIP instruction, the list parameter is specified if need be (*i.e.*, if any PEs will be activated here), and the remap flag, remap, is set. When activate_PEs() notices that all the PEs are activated and the remap flag is set, it invokes H_M to map $\hat{S}_{k:\sigma'}$ to $\hat{S}_{i:\sigma}$. The mapping from a location in S' to a location in S is just a matter of referencing the ART. S has no added overhead owing to the remapping procedure and can be as efficient as possible.

Consider the time complexity of synchronizing all the PEs. H_M must construct a list of all interrupted locations in M, which would take in the worst case $O(P)$ time. Then, whenever activate_PEs() is called, it checks which PEs on the list should be activated. This is done by comparing the return address values and PC value of the CU to those of the PEs' individual interrupted states in $\hat{M}_{1:j}$. Thus if the maximum subroutine nesting depth in $\hat{M}_{1:j}$ is d, this takes a worst case $O(d)$ time (checked by all nonactive PEs simultaneously). Also, for each conditional statement in $\hat{S}_{k:\sigma'}$ where all active PEs are disabled, $O(d)$ comparisons take place. Suppose the number of such conditionals executed is <u>*Ncond*</u>, then an overhead of $O(d \cdot Ncond)$ is incurred. Finally, to determine the point at which $\hat{S}_{k:\sigma'}$ should be mapped to $\hat{S}_{i:\sigma}$, the "farthest state" from the beginning of $\hat{S}_{k:\sigma'}$ needs to be found. This would take worst case $O(d \cdot \log P)$ time (same as finding the starting state). Thus, the total worst case time complexity to synchronize the PEs is $O(P + d \cdot \log P + d \cdot Ncond)$.

4.3 Specifying H_M

4.3.1 Overview

This subsection specifies how H_M maps the state of an interrupted SPMD program, $\hat{M}_{1:j}$, to a valid starting state of an intermediate SIMD program, $\hat{S}_{k:\sigma'}$, and then how a state in $\hat{S}_{k:\sigma'}$ can be mapped to a state in $\hat{S}_{i:\sigma}$, where $\hat{S}_{1:\sigma} = \hat{S}_{1:\eta} = \hat{M}_{1:\mu}$ (recall η is the final state of S'). The discussion is divided into three parts: remapping the stack, moving data, and reallocating the registers. A worst-case time complexity to do each is given.

4.3.2 Remapping the Stack

To remap the stack from the SPMD PEs to both the CU and PEs in the SIMD machine, the SPMD machine's PEs' stack is divided among the SIMD machine's CU and PEs. The portions of the SPMD PE stack that gets copied to the SIMD CU stack are the mono temporary and local variables, mono parameters, and the subroutine return addresses. The SIMD PE stack will receive the poly temporary and local variables, and poly parameters. Information about which location on the PE stack is occupied by a mono variable is kept in the SST (Subsection 4.1). The SPMD code subroutine return addresses are mapped to corresponding SIMD code addresses using the ART. Frame pointers for both the SIMD CU and PE stacks can be derived from the SPMD PE frame pointer. An example of remapping the stack is shown in Fig. 4 (recall in S' all mono variables are made poly variables). If the PE stack has <u>*csize*</u> mono stack variables and <u>*psize*</u> poly stack variables, the worst case time complexity required to remap the stacks from a state in $\hat{M}_{1:j}$ to that in $\hat{S}_{i:\sigma}$ is $O(P \cdot (csize + psize))$.

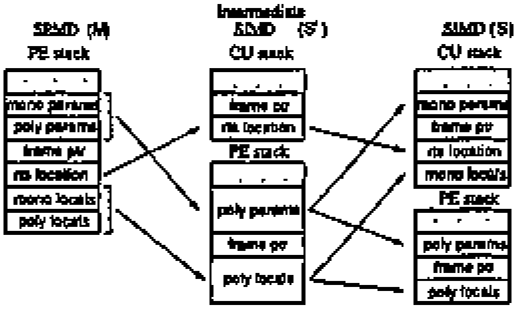


Fig. 4: Remapping the stack from M to S' and from S to S. rts location is the return address and frame ptr stands for frame pointer.

4.3.3 Moving the Data

Because H_M is a two step process using an intermediate SIMD program, S', which has no mono variables, all mono variables in M must be treated as poly variables in S'. Consider a pointer to a global or dynamically allocated mono variable in M. If the mono variable has a different virtual address in S', the pointer would need to be remapped accordingly. To avoid this added overhead, H_M maps the mono variables of the SPMD program to the same virtual address locations in S', even though S' has no mono variables. This is shown pictorially in Fig. 5. Storing poly data in what is normally a mono data segment may present a problem if the virtual address of the operands determines if an instruction operates on poly variables or mono variables. It is assumed, however, for the SIMD virtual machine used here that something other than the virtual address is used to specify operations on poly or mono data (e.g., the opcode for MasPar MP-1 and PASM).

The second step of the mapping occurs when a state in

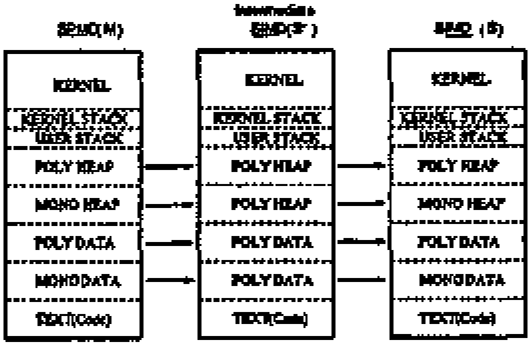


Fig. 5: Data mapping from M to S' and from S' to S.

$\hat{\hat{S}}_{t,x_0'}$ is mapped to a state in $\hat{S}_{t,x_0}$. Both S' and S exist on the same SIMD machine. Thus, the same virtual address space can be shared by both programs. They can also use the same page tables. Only mono data needs to be moved from the PEs to the CU and their page table entries remapped during this step.

If the number of mono and poly variable data bytes is Cdata and Sdata, respectively, then the time complexity to move the data from M to S' to S is $O(P \cdot (Cdata + Sdata))$.

4.3.4 Reallocating Temporary Data

Register usage in S, S', and M may differ significantly from each other. The reason for this is that on an SIMD machine, some registers exist on the CU and others on the PEs. The registers on the CU are not used for poly variable operations. On the SPMD machine, all registers are on the PEs and thus any register can be used for poly or mono variable operations. It is possible to have the VPL compiler store, in the ART, temporary memory location and register mappings (temporary memory locations are assumed to exist in global memory and not in the stack). A compiler keeps track of the information stored in registers and temporary memory locations as it generates and optimizes code [1], and could encode the relevant parts of this information in the ART of all the programs. This information could then be used by H_M to map the values stored in registers and temporary memory locations at an interrupted point in $\hat{M}_{t,f}$ to a valid state in $\hat{S}'_{t,x_0'}$ and from a state in $\hat{S}'_{t,x_0'}$ to a state in $\hat{S}_{t,x_0}$, where S, S', and M efficiently use the registers on the SIMD machine and SPMD machine. However, if some degree of register usage emulation is done in either S and M, simple mappings may exist.

As stated in Section 2, the SIMD machine has $Creg$ registers on the CU and $Sreg$ registers on the PEs. Furthermore, the SPMD machine has $Creg + Sreg$ registers on its PEs. If $Ctemp$ and $Stemp$ temporary memory locations are used for mono and poly variables, respectively, the time complexity to perform the register allocation is $O(P \cdot (Creg + Ctemp + Sreg + Stemp))$. This is the time to move the register and temporary data between machines, and does not include the time to do the mapping between the SPMD machine's register set and the SIMD machine's register set. The mapping time is expected to be $O(Creg + Ctemp + Sreg + Stemp)$ (i.e., performing a table lookup for each register or temporary value).

5. SUMMARY

For fault-tolerance, load balancing, or various administrative reasons, a task may need to be migrated dynamically between an SIMD virtual machine and an SPMD virtual machine. The mapping, H_M, from a interrupted point in an SPMD program to a viable state in an SIMD program was presented and the asymptotic time complexity was given. One of the assumptions made was that the task to be migrated was coded in a mode-independent programming language (e.g., ELP, HPF, Paralation Lisp, XPC). H_M performs a two step mapping. The first step maps the SPMD program to a somewhat inefficient intermediate SIMD program and the second step maps the intermediate SIMD program to a final SIMD program. The introduction of the intermediate SIMD program was found useful to meet the requirement that when the SPMD program is interrupted, it is moved "quickly" off the SPMD machine. The states of the PEs are synchronized by the intermediate SIMD program before the final SIMD program is invoked.

To limit the scope of this paper it was assumed that the hardware configuration of the SIMD virtual machine and SPMD virtual machine differed only to support the different modes of parallelism. Although this is not the case for most existing SIMD and SPMD machine pairs, the assumption is appropriate for mixed-mode machines (e.g., OPSILA, PASM, TRAC, Triton/1). However, the goal of the paper is to solve part of the more general problem of migrating a task between two arbitrary SIMD and SPMD machines. This work is seen as a necessary step in solving this more general problem in the field of heterogeneous computing.

REFERENCES

[1] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, Addison-Wesley, Reading, MA, 1986.

[2] J. B. Armstrong, M. A. Nichols, H. J. Siegel, and L. H. Jamieson, "Examining the effects of CU/PE overlap and synchronization overhead when using the complete sums approach to image correlation," *Third IEEE Symp. Parallel and Distributed Processing*, Dec. 1991, pp. 224-232.

[3] J. B. Armstrong, H. J. Siegel, W. E. Cohen, M. Tan, H. G. Dietz, and J. A. B. Fortes, *Dynamic Task Migration Between SIMD and SPMD Machines: A First Step*, Tech. Rep., EE School, Purdue, in preparation.

[4] J. B. Armstrong, D. W. Watson, and H. J. Siegel, "Software issues for the PASM parallel processing system," in *Software for Parallel Computation*, eds., Janusz S. Kowalik and Lucio Grandinetti, Springer-Verlag, Berlin, 1993, pp. 134-148.

[5] D.J.Batey and J. A. Padget, "Towards a virtual multicomputer," *Workshop on Heterogeneous Processing*, Apr. 1993, pp. 71-76.

[6] T. Blank, "The MasPar MP-1 architecture," *IEEE Compcon*, Feb. 1990, pp. 20-24.

[7] E. C. Bronson, T. L. Casavant, and L. H. Jamieson, "Experimental application-driven architecture analysis of an SIMD/MIMD parallel processing system," *IEEE Trans. Parallel and Distributed Systems*, Vol. 1, Apr. 1990, pp. 195-205.

[8] H. G. Dietz and G. Krishnamurthy, "Meta-State conversion," *1993 Int'l Conf. Parallel Processing*, Vol. II, Aug. 1993, pp. 47-56.

[9] P. Duclos, F. Boeri, M. Auguin, and G. Giraudon, "Image processing on SIMD/SPMD architecture: OPSILA," *9th Int'l Conf. Pattern Recognition*, Nov. 1988, pp. 430-433.

[10] F. B. Dubach, R. M. Rutherford, and C. M. Shub, "Process-originated migration in a heterogeneous environment," *ACM Conf. Computer Science*, Feb. 1989, pp. 98-102.

[11] R. F. Freund and H. J. Siegel, "Guest editors' introduction: heterogeneous processing," *IEEE Computer*, Vol. 26, No. 6, June 1993, pp. 13-17.

[12] J. P. Hayes and T. Mudge, "Hypercube supercomputers," *Proc. of the IEEE*, Vol. 77, No. 12, Dec. 1989, pp. 1829-1841.

[13] W. D. Hillis and Lewis W. Tucker, "The CM-5 Connection Machine: a scalable supercomputer," *Communications of the ACM*, Vol. 36, No. 11, Nov. 1993, pp. 30-40.

[14] Y. Hollander and G. M. Silberman, "A mechanism for the migration of tasks in heterogeneous distributed processing systems," *Int'l Conf. Parallel Processing and Applications*, Sept. 1988, pp. 93-98.

[15] High Performance Fortran Forum, "Draft: high performance Fortran language specification," *High Performance Fortran Forum*, Sep. 1992.

[16] Kendall Square Research, *Technical Summary*, Kendall Square Research, Waltham, MA, 1992.

[17] R. Keryell and N. Paris, "Activity counter: new optimization for the dynamic scheduling of SIMD control flow," *1993 Int'l Conf. Parallel Processing*, Vol. II, Aug. 1993, pp. 184-187.

[18] G. J. Lipovski and M. Malek, *Parallel Computing: Theory and Comparisons*, John Wiley & Sons, New York, NY, 1987.

[19] MasPar Computer Corporation, *MasPar MPL Manuals*, MasPar Corporation, Sunnyvale, CA, July 1993.

[20] M. A. Nichols, H. J. Siegel, and H. G. Dietz, "Data management and control-flow aspects of an SIMD/SPMD parallel language/compiler," *IEEE Trans. Parallel and Distributed Systems*, Vol. 4, No. 2, pp. 222-234.

[21] M. J. Phillip and H. G. Dietz, "Toward semantic self-consistency in explicitly parallel languages," *4th Int'l Conf. Supercomputing*, May 1989, pp. 398-407.

[22] M. Philippsen, T. Warschko, W. Tichy, and C. Herter, "Project Triton: towards improved programmability of parallel machines," *26th Hawaii Int'l Conf. System Sciences*, Jan. 1993, pp. 192-201.

[23] C. M. Shub, "Native code process-originated migration in a heterogeneous environment," *ACM Eighteenth Annual Computer Science Conf.*, Feb. 1990, pp. 266-270.

[24] H. J. Siegel, J. B. Armstrong, and D. W. Watson, "Mapping computer-vision-related tasks onto reconfigurable parallel-processing systems," *IEEE Computer*, Vol. 25, No. 2, Feb. 1992, pp. 54-63.

[25] H. J. Siegel, T. Schwederski, W. G. Nation, J. B. Armstrong, L. Wang, J. T. Kuehn, R. Gupta, M. D. Allemang, and D. G. Meyer, "The design and prototyping of the PASM reconfigurable parallel processing system," in *Parallel Computing: Paradigms and Applications*, Albert Zomaya, ed., Chapman and Hall, London, UK, 1994 (in press).

[26] J. M. Smith, "A survey of process migration mechanisms," *Operating Systems Review*, Vol. 22, No. 3, July 1988, pp. 28-40.

[27] M. M. Theimer, K. A. Lantz, and D. R. Cheriton, "Preemptable remote execution facilities for the V-System," *Tenth ACM Symp. on Operating Systems Principles*, Dec. 1985, pp. 2-12.

[28] L. W. Tucker and G. G. Robertson, "Architecture and applications of the Connection Machine," *IEEE Computer*, Vol. 21, No. 8, Aug. 1988, pp. 26-38.

[29] D. G. Von Bank, C. M. Shub, and R. W. Sebesta, *A Unified Model of Pointwise Equivalence of Procedural Computations*, Tech. Report TR-EE EAS-CS-93-3, Computer Science Department, University of Colorado, Apr. 1993.

[30] D. W. Watson, H. J. Siegel, J. K. Antonio, M. A. Nichols, and M. J. Atallah, "A framework for compile-time selection of parallel modes in an SIMD/SPMD heterogeneous environment," *Workshop on Heterogeneous Processing*, Apr. 1993, pp. 57-64.

LOOKING UNDER THE HOOD WHILE DRIVING THE INFORMATION HIGHWAY

Ed Krol, University of Illinois

I'd like to talk about the Internet today. My approach will be to tell you a bit about what it is, how its structured and how to use it, but at each point also give you a hand waving explanation of how it works as well. It turns out it may be big but the concepts behind it are pretty straightforward.

The Internet is a large number, over 29000, of networks who have all agreed to use the same basic technology and carry each others traffic. For the end user, this means that if you are connected to any one of those networks, you have access to computational and data resources on any of the others.

There is no central control or chief operating officer of the Internet. Each network is independently run. In its original incarnation if a network wanted to become part of the Internet, it would be connected over a dedicated data communications line. Routers, networking nodes on the joining network would send a message to the closest neighbor, saying that it knew how to get to and would act as an agent for a list of new networks. The neighbor would pass this information on until fairly quickly, the entire net would know how to reach those new networks.

This model has a number of consequences, the most basic is that we have know idea exactly how many people or computers there are on the Internet. When a network joins the Internet it can be counted, but the owner of that network did not have to register or ask permission to add any of the computers on it. Each of those computers may service any number of people, again we don't know how many. Estimates of these numbers run about 2 million machines and 20 million people.

The growth of the Internet has led to a slight change in this model for route acquisition. Consider the problem, what if a net tells its neighbor it can reach a network but it really can't? In this situation, communications destined for the orphaned network pour into the network claiming a connection and they are lost. To get around this problem, many of the major transit networks of the Internet maintain a believability database, which says which announcements should be believed when received by the transit network.

So we have this odd network of networks with no one in charge, what are the properties which makes it special? One major property is that it is peer to peer. Every computer on the Internet can either be a consumer or a provider of resources. This allows resources to be made available for really small clientele, with no necessity for their economic viability or profit potential.

The second property is that what is provided is a communications pathway, with all the smarts in the peer machines. This allows the end machines to do experimental things and in fact improve over time. All it takes is new software and there is new functionality.

To understand how these peer machines communicate, consider the global postal service. It's an internet, too. You can send a message from the US to France without ever knowing the underlying transport. This works because the postal services have agreed to act as each others agents, agreed on a standard format for handling mail, and will pass messages closer to a destination even if there is no direct route.

The Internet is the same. The agreement is the Internet Protocol (IP) which specifies the format of an envelope for a short packet of data, usually less than 1500 bytes, and an address format. An application on a computer merely has to put its message in one of these packets and address it correctly. The Internet will take it and attempt to get it to its destination.

Of course we all know the stereotypical stories about the post office being an unreliable delivery mechanism. The design criteria for the Internet assumes the same. At any time an IP packet might get lost, or delivered out of sequence. A basic IP network is not a particularly useful communications tool.

To fix this, the Transmission Control Protocol (TCP) is usually run between the application and the IP network. It assures that data is presented in the same order to the receiver as it is sent, and that all missing or corrupted packets are retransmitted. So together TCP/IP gets you a reliable byte stream between any two Internet connected computers in the world.

Most people don't get enthused by reliable byte streams - they want to do real work or real science. Doing real work on the Internet involves clients and servers. Clients are pieces of software which are manipulated by a human to do something useful. If necessary clients contact servers across the net to provide them with resources required to do the humans bidding. Everything on the Internet happens in this manner. And, one of the benefits of this type of computing, is that the server can handle requests from a variety of platforms.

In the late 60's or early 70's the original applications on the Internet were hammered out. These were electronic mail, remote login (timesharing across the Internet), and file transfer. As was typical of the time, before graphical user interfaces and mice became popular, these application were command line driven.

This is where the Internet's development faltered. The first problem encountered has been finding the resource to access. In its earlier days, the major means of finding resources, was human networking to support the Internet. People would go to conferences and meetings, hear about good things, and write down their location. When they returned home, they could use their new found information to access what was there.

This is where the other miscue occurred. Since a client program had to be written for each platform used, developers could have designed it use the command structure of the local machine wherever possible. This would have allowed the users knowledge to be applied to its new found tasks. Rather, they wrote a set of global commands for the Internet applications. Even if you were quite competent on a TOPS-10, Unix, or an IBM machine, you had to learn "ftp" to use the network. It kept people from trying.

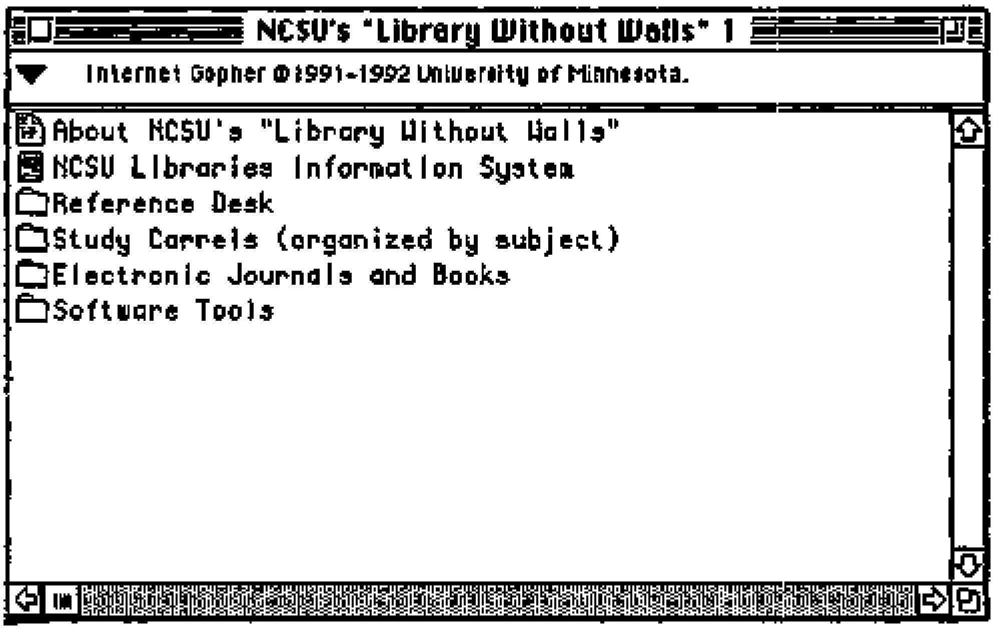
Its only been recently that this trend has reversed. Now we are starting to see clients, which use the intrinsic characteristics of the specific computer to do network access. This means that anyone competent on a Macintosh, or Microsoft windows machine, can use the Internet in a manner that they are familiar with.

Another trend we see in the development of new applications is a move to integrate just like the evolution that occurred in the PC world. The PC started as something that could run BASIC. Next, people developed word processors, spreadsheets, and graphics packages. You could write your text, do your calculations, and make graphs. You would then have to print them all out, cut and paste them with real scissors and paste. Now, there are integrated packages that allow you to do all of those things and print a finished product.

The same thing is happening on the Internet. We started with a bunch of discrete low level tools and have worked our way up to tools which allow you to find a resource and say give me that, and it appears. Interestingly enough, these new tools were so simple in concept that they took forever to be developed.

The first of these tools which brought the Internet out of the realm of computer geek and allowed its use by the common computer literate person was gopher. Gopher is a menu oriented tool for delivering primarily text files. It started out as a campus information system at the University of Minnesota. In the course of about 3 years it has gone from one site to well over 1000 servers.

A gopher client presents to the user menus. This is a screen from Turbogopher on a Macintosh, but it doesn't matter. You could have accessed the same data from just about any type of machine, including those with non-graphical displays. Notice that every line has a type icon to its left. The documents are documents - if you click on that you will see the document. The folders are sub-menus - click one of them and another menu appears. The computer icon is a timesharing resource. If you select it gopher will automatically remote login to that resource.



NCSU's "Library Without Walls" 1

Internet Gopher ©1991-1992 University of Minnesota.

About NCSU's "Library Without Walls"
NCSU Libraries Information System
Reference Desk
Study Carrels (organized by subject)
Electronic Journals and Books
Software Tools

How does this all work? Notice the similarity between a gopher server 's menu structure and a file structure. A menu is similar to a directory and files are files. In fact that's the implementation. You run server software on a computer somewhere and point it at a file structure. The one interesting twist is that it by default displays filenames and directories as menu items, but if you define a "titles" file it will substitute a human readable title for a filename.

```
Name=Study Carrels (organized by subject)
Type=1
Port=70
Path=1/library/disciplines
```

Host=dewey.lib.ncsu.edu

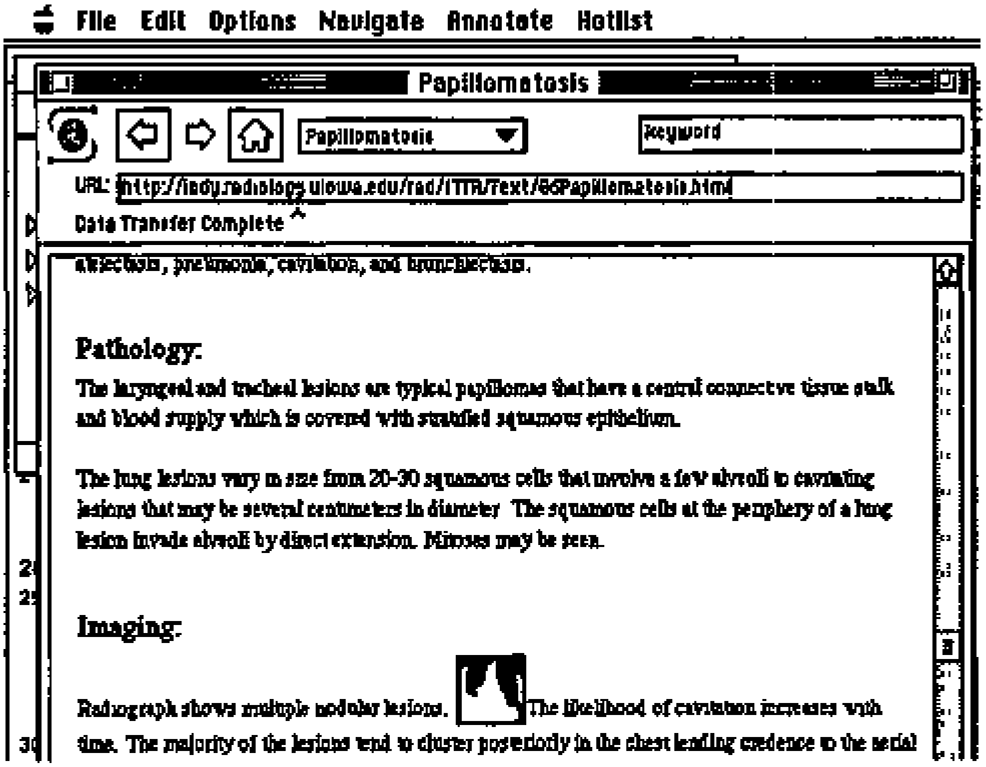
There is no long term relationship between a gopher client and a particular server. When you access a server it sends your client a menu and some hidden data which tells your client how to access the each item. That item may be served from the same or another server it doesn't matter. When you choose another item, your client will contact the host specified on the port specified, and ask for the path. This will return something to be displayed depending on the type.

That's how gopher works, but the engineering push for simplicity also limited its high end capabilities. The data displayed is a file of some type, normally text or a GIF image. This makes it quite easy to make documents available, but there is no real way to have imbedded images and other types present problems.

The alternative to gopher is to choose the higher end, but more maintenance intensive technology of the world wide web. It has been around longer than gopher but has always suffered, because the document preparation time was greater for little payback. The reformatting effort necessary to add hypertext links to other documents was just not worth it since what you got end looked a lot like gopher. This lack of payback was due to there being no good display clients. With the advent of a client called Mosaic this has all changed. Mosaic, from NCSA, is the best of the WWW browsers around. Allowing multimedia presentation of audio, pictures, video, and text in a complete package.

Web documents are prepared and stored in an SGML derived markup language call Hypertext Markup Language (HTML). When a client requests a document it is returned in raw form and formatted by the client. This allows the client to do formatting in the best possible manner for the display provided.

This is a display from Mosaic for a Macintosh (there are X and Microsoft windows clients as well). Notice the variety of text formats and the thumbnail sized image. This image is displayed and should the user want to pay the time to get a full sized higher resolution image it can be fetched by clicking on it.



URL: http://indy.radiology.uiowa.edu/rad/ITTR/Text/86Papillomatosis.html

Data Transfer Complete

atelectasis, pneumonia, cavitation, and bronchiectasis.

Pathology:

The laryngeal and tracheal lesions are typical papillomas that have a central connective tissue stalk and blood supply which is covered with stratified squamous epithelium.

The lung lesions vary in size from 20-30 squamous cells that involve a few alveoli to cavitating lesions that may be several centimeters in diameter The squamous cells at the periphery of a lung lesion invade alveoli by direct extension. Mitoses may be seen.

Imaging:

Radiograph shows multiple nodular lesions. The likelihood of cavitation increases with time. The majority of the lesions tend to cluster posteriorly in the chest leading credence to the aerial

The ability to have a link in one document to another of arbitrary format is the key to its success and centers around the Uniform Resource Locator (URL). A URL is an Internet standard method to describe the location of a particular document and the service necessary to access it. The basic format of a URL is a character string which begins with a service name and is followed by some service dependent information. Normally this service dependent part will look a lot like the gopher data structure we looked at previously. For example:

http://indy.radiology.uiowa.edu/rad/Text/86Papillamatosis.html

consists of a service "http", followed by the name of a machine
"indy.radiology.uiowa.edu", then name of a file path and file name.
Http is the name for the Web's transport, so this is a WWW resource. The suffix ".html" says this is in normal Web markup language which needs to be interpreted by the client.

Lets look at a bit of the HTML required to produce the above document.

```
<P>
<H3>Imaging:
</H3>
Radiograph shows multiple nodular lesions.
```

```
<A
HREF="http://indy.radiology.uiowa.edu/rad/ITTR/RadImages/papillomatosis.pa.
jpg">
<IMG
SRC="http://indy.radiology.uiowa.edu/rad/ITTR/RadImages/papillomatosis.pa.ic
on.gif">

</A>The likelihood of cavitation increases with time. The majority of the lesions
tend to cluster posteriorly in the chest lending credence to the aerial dissemination
theory.
```

This section of document starts with a paragraph marker (<P>). Then it puts out a level 3 section header beginning with <H3> and ending with a </H3>. There is some random text followed by some pointers to other documents between the <A> and </A>. The string beginning with HREF is a URL connection to another document. It could be text, audio or a picture. This happens to be a JPEG graphics image (the ending ".jpg"). The inline thumbnail image is located through the URL following the <IMG SRC... directive. After the </A> follows some normal text again.

So now we see how the Internet hangs together and how you might use it. Now lets think a little about the challenges facing it. First, the technology is neat, but people don't really know how to apply it. They are applying it inappropriately by trial and error, typically building a worse mousetrap. If you consider the medical resource early, ask yourself why they should build that. They are producing a medical textbook which can only be read while you are seated at a connected computer. If you want to study at the beach, you can only do it for 2 hours and lose significant resolution. Granted the technology will improve, but at this point its a waste of time.

Currently, we are in the trial and error phase of deploying Web technology. Not only don't we know when it should be deployed, but the actual format of these documents is also being hammered out. Currently authors don't really know when and how to include images or when to include links to other resources. Many web documents have so many links its like reading a magazine article with lots of side bars - there is no obvious way to proceed through it. The course will become more apparent as more documents get written, read, and critiqued.

Another problem we have is that the culture is biased against online resources. Libraries work pretty poorly, but they are tolerated. If you look up a book in the card catalog, and then proceed to the shelf and don't find it - you were unlucky. If you find a network resource and try to connect but can't access it - the network doesn't work.

People forget about how to do research in an online environment. Traditionally, if you need to learn about a new topic one of the ways of doing it is to get one article and look at the references. That method works on the net too, especially in the WWW. If people find one document they like, it will usually have links to others. People just refuse to do the same old stuff in a networked environment.

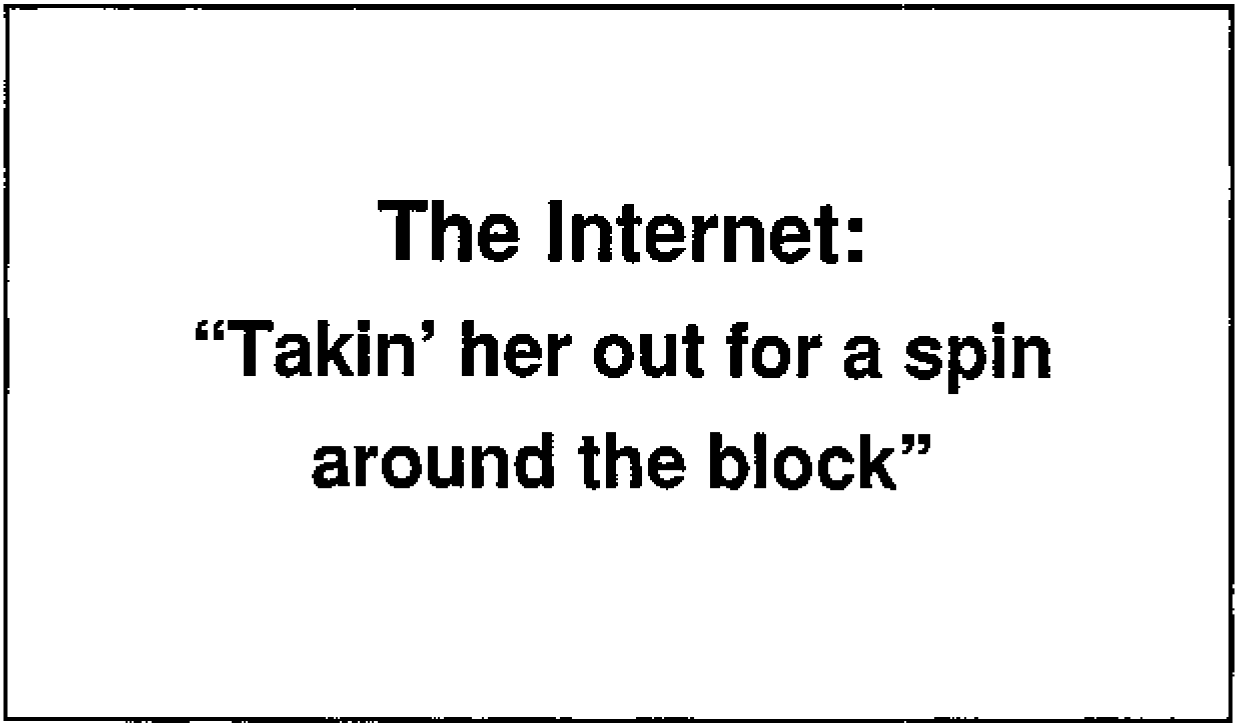
There is also a problem with citability of online resource. How do I know the author is authoritative and how do I know a work has not been and will not be altered after it has been cited? This is purely a technological problem, solved by deploying digital signatures.

Finally, there is support inertia. As the Internet grows larger and larger, it gets harder and harder to do both support and experimentation. You need to sacrifice technological solutions in deference to the installed base. And, you might need to slow deployment of new technology because having many versions deployed leads to too much of a support problem. This in fact will culminate when the NII is deployed because the software will probably be locked in silicon.

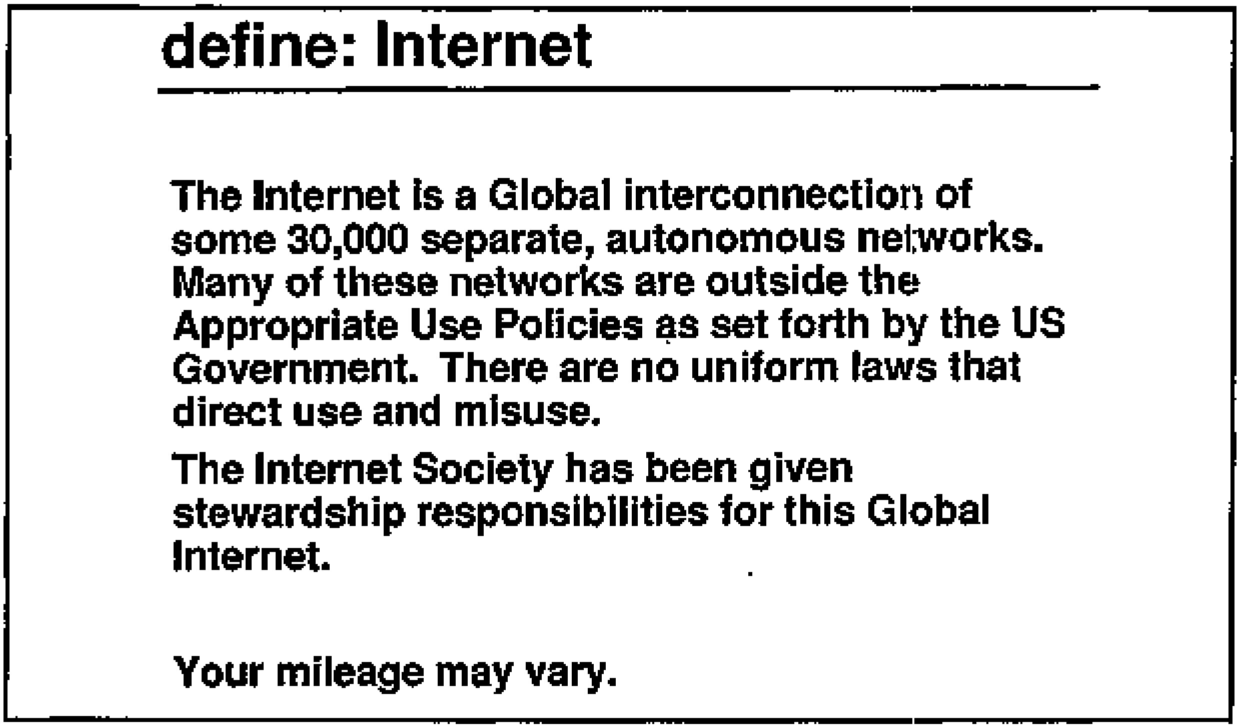
In conclusion, the Internet is a communication pipe usable for whatever you want. There are standard things to do, but you can use it to do any kind of human collaboration or computer collaboration you can imagine.

INTERNET DEMO

Tom DeBoni, LLNL, and Dale Land, LANL



The Internet:
"Takin' her out for a spin around the block"



define: Internet

The Internet is a Global interconnection of some 30,000 separate, autonomous networks. Many of these networks are outside the Appropriate Use Policies as set forth by the US Government. There are no uniform laws that direct use and misuse.

The Internet Society has been given stewardship responsibilities for this Global Internet.

Your mileage may vary.

The Scenery

- Basic Functionality
  - Remote login, file transfer, remote process communication .
- Services and Servers
  - E-mail, FTP, USENET News
- Information retrieval agents
  - Gopher, WAIS, Archie, WWW and hypermedia.
- The "Killer" application
  - Mosaic
- Inter-human communication
  - Multicast; NV, VAT, and WB

Basic Functionality

- Remote login
  - Virtual terminal sessions with non-local computer
- File transfer
  - Machine to machine file copy
  - The beginning of moving information, not people
- Remote process communication
  - Fundamental for all uses to come
  - Building block of higher level services

Services and Servers

- E-mail
  - Asynchronous person to [one, some, many]
  - Straight text initially, MIME to the rescue!
- FTP
  - Scalable, low-cost information distribution
- USENET News
  - Thousands of global bulletin boards
  - Over 6000 subject areas
  - Infinite time sync for the undisciplined
  - Open ended technical advice from experts

Information Retrieval Agents

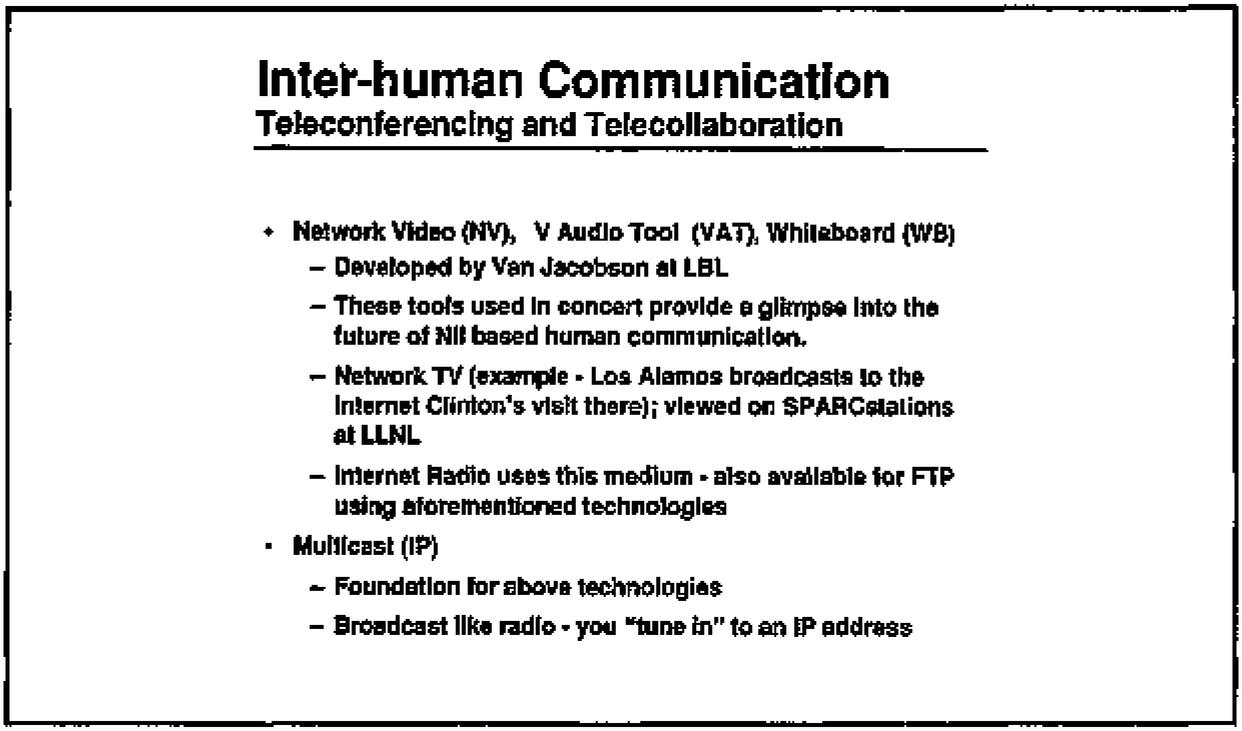
- Gopher
  - FTP client/server pair
  - Stateless - less impact on server
  - Visual - more impact on the user
- Archie
  - Filename search service - use it to find which servers have named files.
  - Database built by automatic monthly combing through anonymous FTP servers on the Internet.
  - Queries to Archie are searched in its database
  - Queriable by Telnet or E-mail.

Information Retrieval Agents

- Wide Area Information Systems (WAIS)
  - Content addressable FTP server search service
  - Invented by TMC.
  - General keyword searches of servers and their directories.
- World Wide Web (WWW) and hypermedia.
  - Integrates all of the above services.
  - WWW protocol defined at CERN
  - Hypertext + Multimedia = Hypermedia
    - » Hypertext - semantic network overlaid on linear text
    - » Multimedia - multi-modal data delivery

The "Killer" Application Mosaic

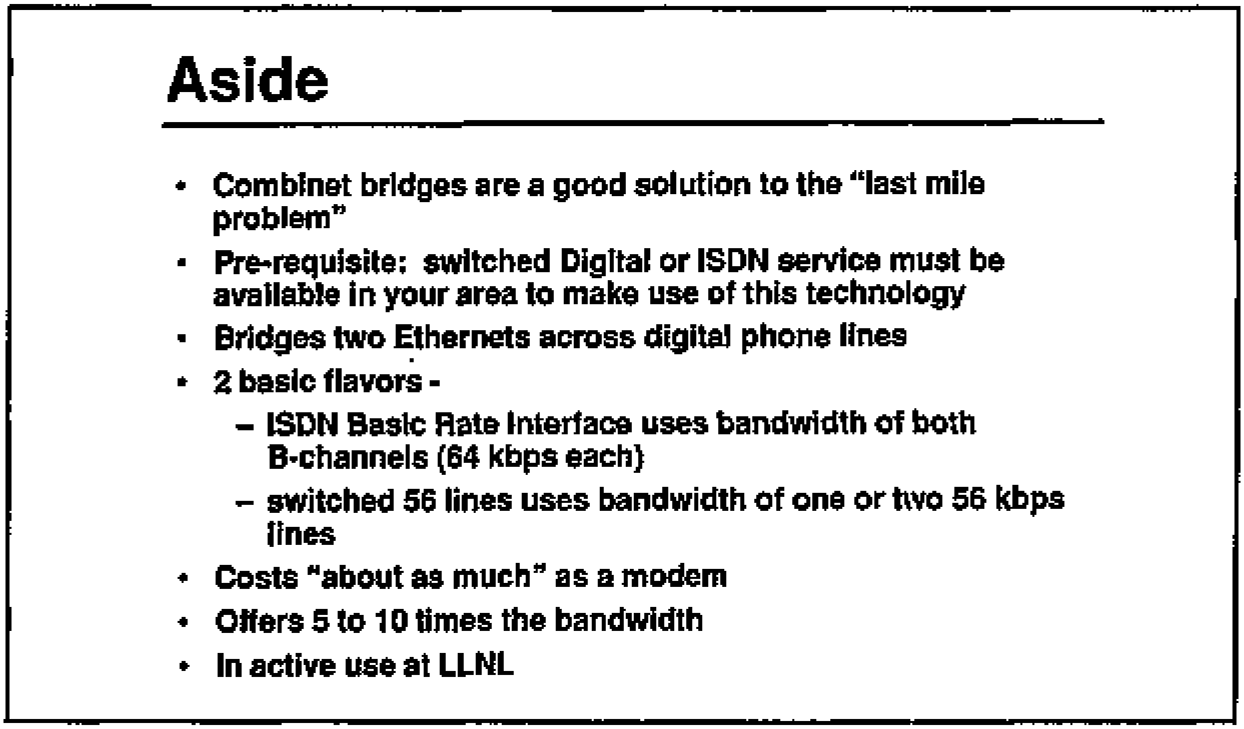
- Developed by NCSA - freeware
  - Available on MAC, UNIX / X11, PC platforms
- Implements WWW protocols with pleasing user interface
- Provides common interface to FTP, telnet, gopher, USENET News, and multimedia applications
- Responsible for large jump in Internet traffic
- Home pages are appearing with increasing frequency -or- more info available daily
- Demo (with any luck at all . . .)



Inter-human Communication

Teleconferencing and Telecollaboration

- Network Video (NV), V Audio Tool (VAT), Whiteboard (WB)
  - Developed by Van Jacobson at LBL
  - These tools used in concert provide a glimpse into the future of NII based human communication.
  - Network TV (example - Los Alamos broadcasts to the Internet Clinton's visit there); viewed on SPARCstations at LLNL
  - Internet Radio uses this medium - also available for FTP using aforementioned technologies
- Multicast (IP)
  - Foundation for above technologies
  - Broadcast like radio - you "tune in" to an IP address



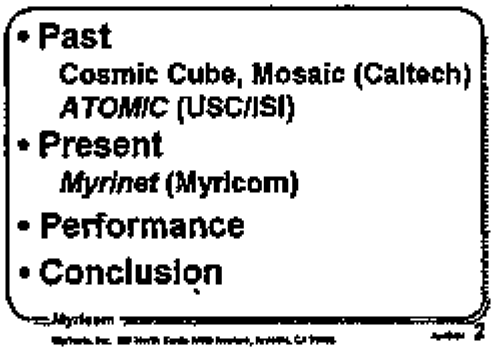
Aside

- Combinet bridges are a good solution to the "last mile problem"
- Pre-requisite: switched Digital or ISDN service must be available in your area to make use of this technology
- Bridges two Ethernets across digital phone lines
- 2 basic flavors -
  - ISDN Basic Rate Interface uses bandwidth of both B-channels (64 kbps each)
  - switched 56 lines uses bandwidth of one or two 56 kbps lines
- Costs "about as much" as a modem
- Offers 5 to 10 times the bandwidth
- In active use at LLNL

ATOMIC/MYRINET - A NEW GIGABIT LAN TECHNOLOGY

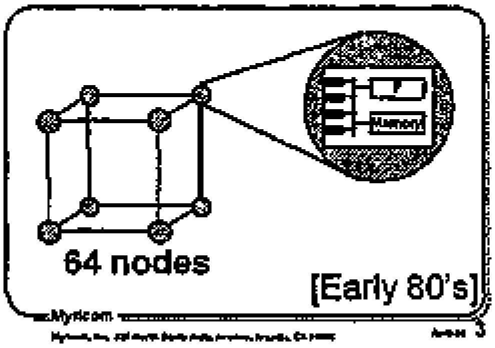
Danny Cohen, Myricom, Inc.

Overview



- Past
  - Cosmic Cube, Mosaic (Caltech)
  - *ATOMIC* (USC/ISI)
- Present
  - *Myrinet* (Myricom)
- Performance
- Conclusion

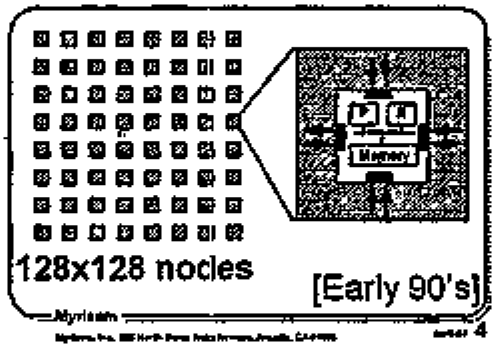
Seitz's Cosmic Cube



64 nodes

[Early 80's]

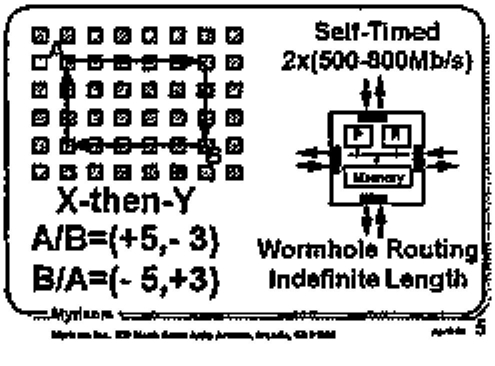
Seitz's Mosaic



128x128 nodes

[Early 90's]

Mosaic Communication

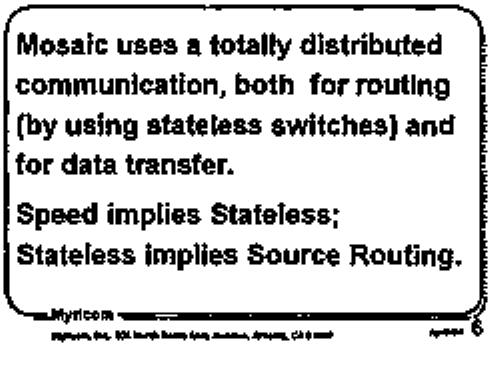


Self-Timed
2x(500-800Mb/s)

X-then-Y
A/B=(+5,- 3)
B/A=(- 5,+3)

Wormhole Routing
Indefinite Length

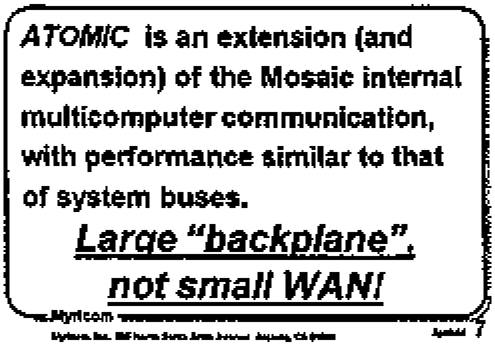
Distributed Comm'n



Mosaic uses a totally distributed communication, both for routing (by using stateless switches) and for data transfer.

Speed implies Stateless;
Stateless implies Source Routing.

USC/ISI's ATOMIC

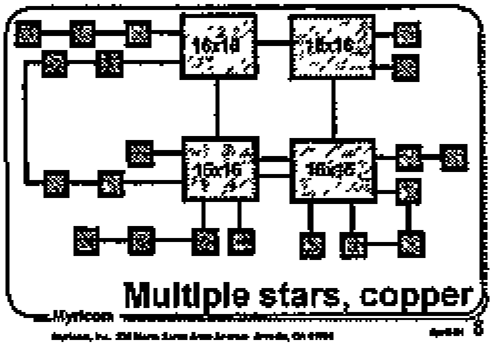


ATOMIC is an extension (and expansion) of the Mosaic internal multicomputer communication, with performance similar to that of system buses.

Large "backplane", not small WAN!

Myricom 7

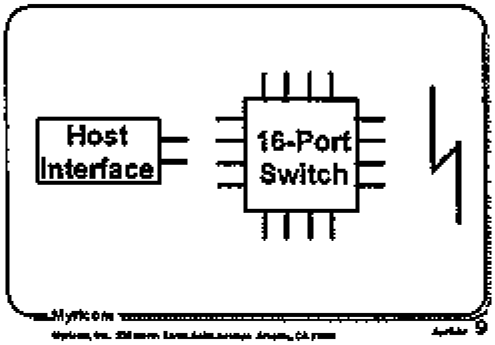
ATOMIC LAN



Multiple stars, copper

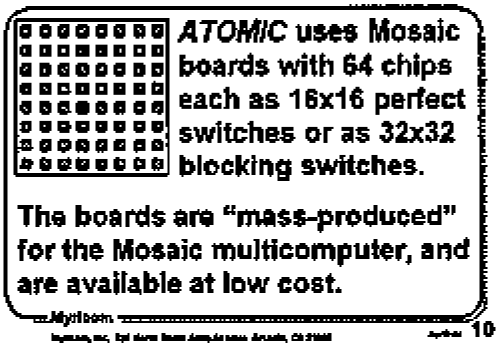
Myricom 8

ATOMIC's Components



Myricom 9

ATOMIC Switches

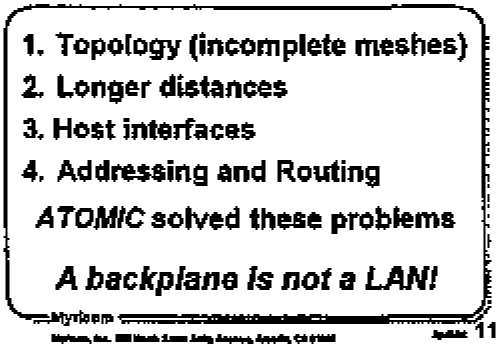


ATOMIC uses Mosaic boards with 64 chips each as 16x16 perfect switches or as 32x32 blocking switches.

The boards are "mass-produced" for the Mosaic multicomputer, and are available at low cost.

Myricom 10

Problems



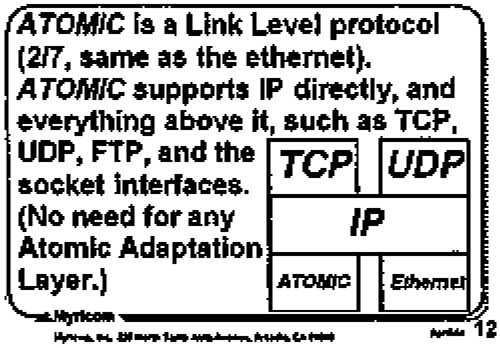
1. Topology (incomplete meshes)
2. Longer distances
3. Host interfaces
4. Addressing and Routing

ATOMIC solved these problems

A backplane is not a LAN!

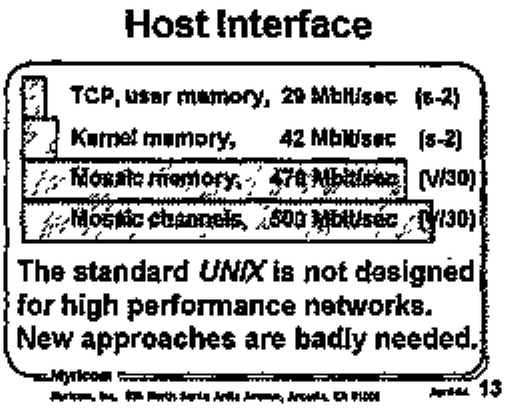
Myricom 11

Protocol Level



ATOMIC is a Link Level protocol (2/7, same as the ethernet). *ATOMIC* supports IP directly, and everything above it, such as TCP, UDP, FTP, and the socket interfaces. (No need for any Atomic Adaptation Layer.)

Myricom 12



Host Interface

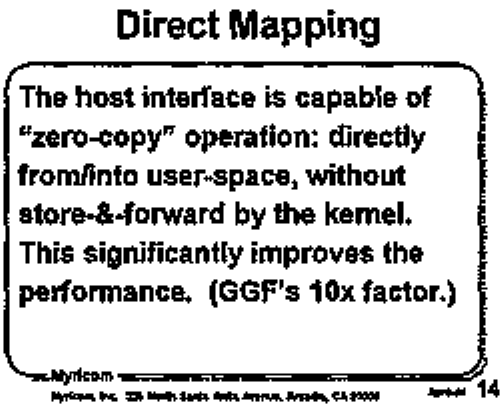
TCP, user memory, 29 Mbit/sec (s-2)

Kernel memory, 42 Mbit/sec (s-2)

Mosaic memory, 478 Mbit/sec (V/30)

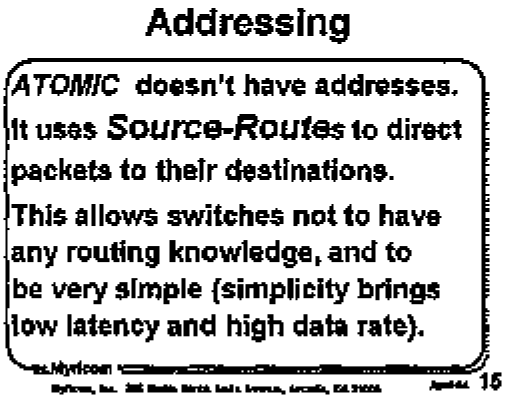
Mosaic channels, 500 Mbit/sec (V/30)

The standard *UNIX* is not designed for high performance networks. New approaches are badly needed.



Direct Mapping

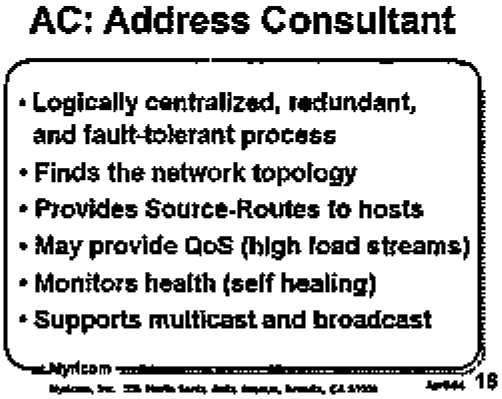
The host interface is capable of "zero-copy" operation: directly from/into user-space, without store-&-forward by the kernel. This significantly improves the performance. (GGF's 10x factor.)



Addressing

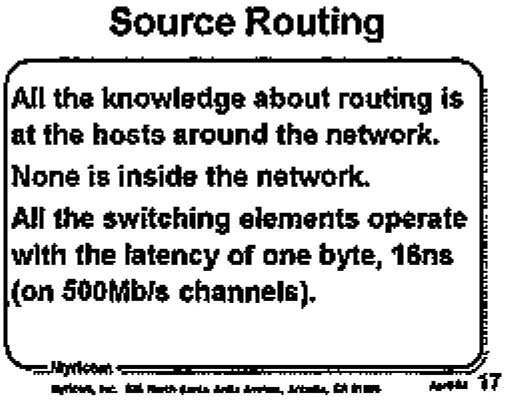
ATOMIC doesn't have addresses. It uses *Source-Routes* to direct packets to their destinations.

This allows switches not to have any routing knowledge, and to be very simple (simplicity brings low latency and high data rate).



AC: Address Consultant

- Logically centralized, redundant, and fault-tolerant process
- Finds the network topology
- Provides Source-Routes to hosts
- May provide QoS (high load streams)
- Monitors health (self healing)
- Supports multicast and broadcast

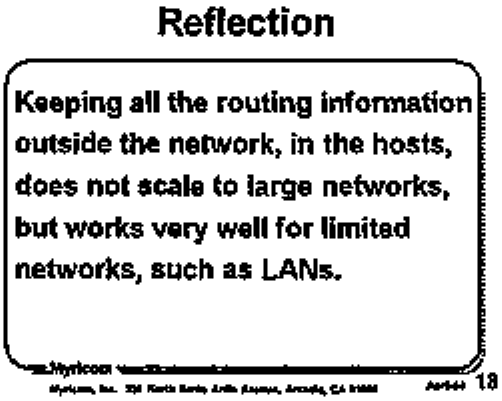


Source Routing

All the knowledge about routing is at the hosts around the network.

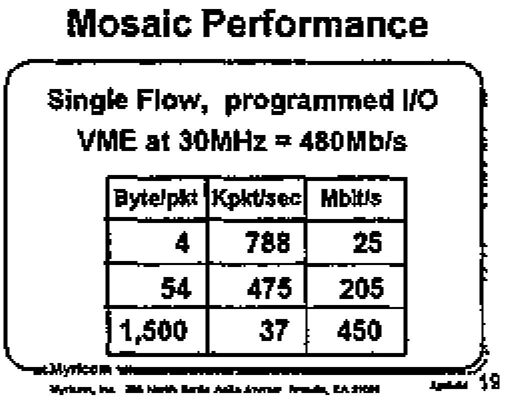
None is inside the network.

All the switching elements operate with the latency of one byte, 16ns (on 500Mb/s channels).



Reflection

Keeping all the routing information outside the network, in the hosts, does not scale to large networks, but works very well for limited networks, such as LANs.

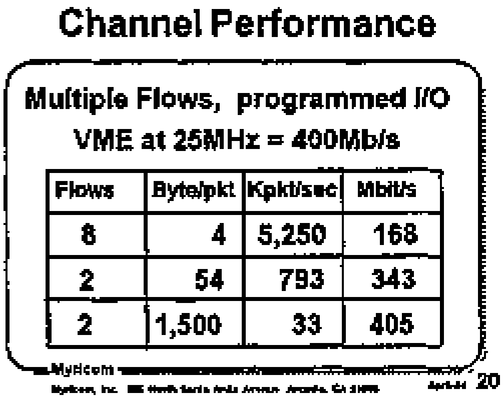


Mosaic Performance

Single Flow, programmed I/O
VME at 30MHz = 480Mb/s

Byte/pkt	Kpkt/sec	Mbit/s
4	788	25
54	475	205
1,500	37	450

Myricom
Myricom, Inc. 325 North Santa Anita Avenue Arcadia, CA 91006 April-94 19

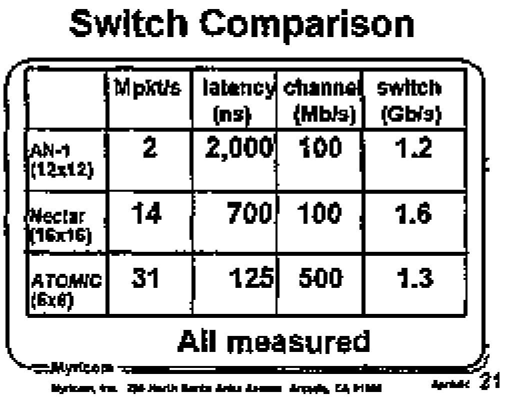


Channel Performance

Multiple Flows, programmed I/O
VME at 25MHz = 400Mb/s

Flows	Byte/pkt	Kpkt/sec	Mbit/s
8	4	5,250	168
2	54	793	343
2	1,500	33	405

Myricom
Myricom, Inc. 325 North Santa Anita Avenue Arcadia, CA 91006 April-94 20

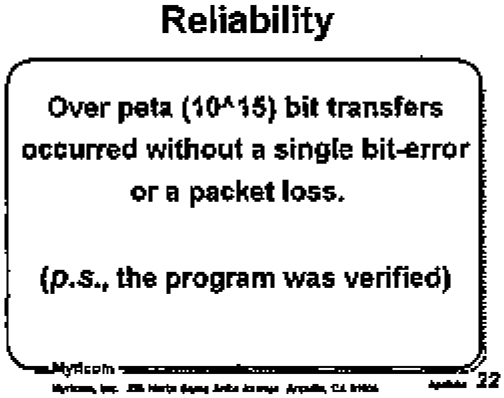


Switch Comparison

	Mpkt/s	latency (ns)	channel (Mb/s)	switch (Gb/s)
AN-1 (12x12)	2	2,000	100	1.2
Nectar (16x16)	14	700	100	1.6
ATOMIC (8x8)	31	125	500	1.3

All measured

Myricom
Myricom, Inc. 325 North Santa Anita Avenue Arcadia, CA 91006 April-94 21

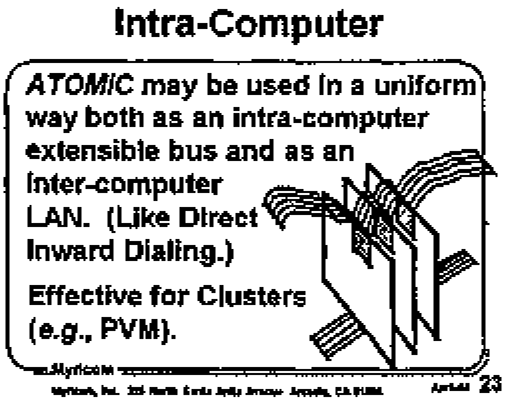


Reliability

Over peta (10^15) bit transfers occurred without a single bit-error or a packet loss.

(*p.s.*, the program was verified)

Myricom
Myricom, Inc. 325 North Santa Anita Avenue Arcadia, CA 91006 April-94 22



Intra-Computer

ATOMIC may be used in a uniform way both as an intra-computer extensible bus and as an inter-computer LAN. (Like Direct Inward Dialing.)

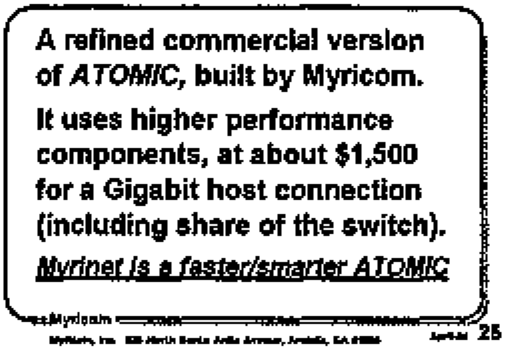
Effective for Clusters (*e.g.*, PVM).

Myricom
Myricom, Inc. 325 North Santa Anita Avenue Arcadia, CA 91006 April-94 23

Myrinet

Myricom
Myricom, Inc. 325 North Santa Anita Avenue Arcadia, CA 91006 April-94 24

Myrinet



A refined commercial version of *ATOMIC,* built by Myricom.

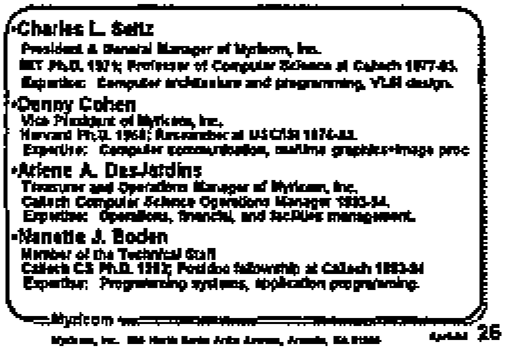
It uses higher performance components, at about $1,500 for a Gigabit host connection (including share of the switch).

Myrinet is a faster/smarter ATOMIC

Myricom

Myricom, Inc. [illegible] North Santa Anita Avenue, Arcadia, CA [illegible] April-94 25

The People of Myricom

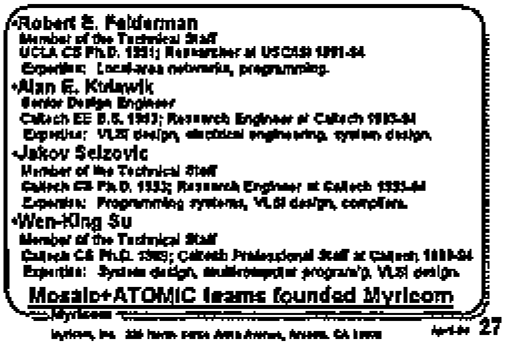


- Charles L. Seitz
  President & General Manager of Myricom, Inc.
  MIT Ph.D. 1971; Professor of Computer Science at Caltech 1977-93.
  Expertise: Computer architecture and programming, VLSI design.
- Danny Cohen
  Vice President of Myricom, Inc.
  Harvard Ph.D. 1969; Researcher at USC/ISI 1976-93.
  Expertise: Computer communication, realtime graphics+image proc
- Arlene A. DesJardins
  Treasurer and Operations Manager of Myricom, Inc.
  Caltech Computer Science Operations Manager 1983-94.
  Expertise: Operations, financial, and facilities management.
- Nanette J. Boden
  Member of the Technical Staff
  Caltech CS Ph.D. 1993; Postdoc fellowship at Caltech 1993-94
  Expertise: Programming systems, application programming.

Myricom

Myricom, Inc. [illegible] North Santa Anita Avenue, Arcadia, CA [illegible] April-94 26

The People - 2



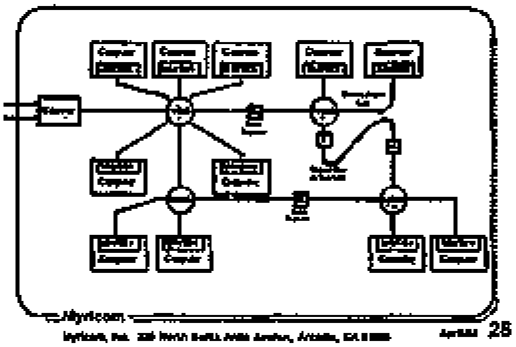
- Robert E. Felderman
  Member of the Technical Staff
  UCLA CS Ph.D. 1991; Researcher at USC/ISI 1991-94
  Expertise: Local-area networks, programming.
- Alan E. Kulawik
  Senior Design Engineer
  Caltech EE B.S. 1983; Research Engineer at Caltech 1983-94
  Expertise: VLSI design, electrical engineering, system design.
- Jakov Seizovic
  Member of the Technical Staff
  Caltech CS Ph.D. 1993; Research Engineer at Caltech 1993-94
  Expertise: Programming systems, VLSI design, compilers.
- Wen-King Su
  Member of the Technical Staff
  Caltech CS Ph.D. 1983; Caltech Professional Staff at Caltech 1983-94
  Expertise: System design, multicomputer program'g, VLSI design.

Mosaic+ATOMIC teams founded Myricom

Myricom

Myricom, Inc. [illegible] North Santa Anita Avenue, Arcadia, CA [illegible] April-94 27

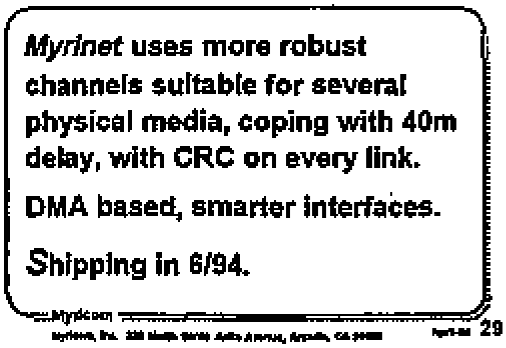
Myrinet, an Example



Myricom

Myricom, Inc. [illegible] North Santa Anita Avenue, Arcadia, CA [illegible] April-94 28

Myrinet



Myrinet uses more robust channels suitable for several physical media, coping with 40m delay, with CRC on every link.

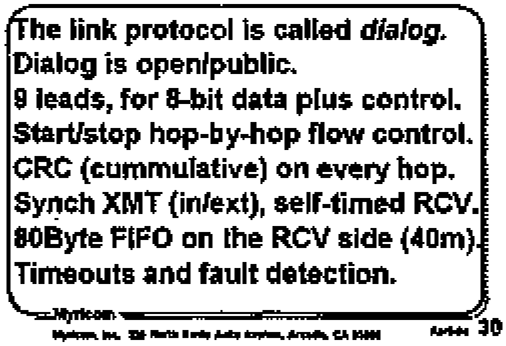
DMA based, smarter interfaces.

Shipping in 6/94.

Myricom

Myricom, Inc. [illegible] North Santa Anita Avenue, Arcadia, CA [illegible] April-94 29

Myrinet Channels

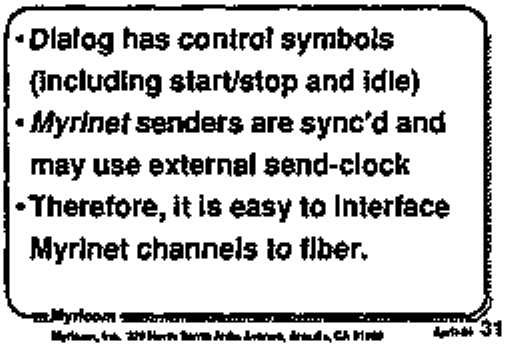


The link protocol is called *dialog.*
Dialog is open/public.
9 leads, for 8-bit data plus control.
Start/stop hop-by-hop flow control.
CRC (cummulative) on every hop.
Synch XMT (in/ext), self-timed RCV.
80Byte FIFO on the RCV side (40m).
Timeouts and fault detection.

Myricom

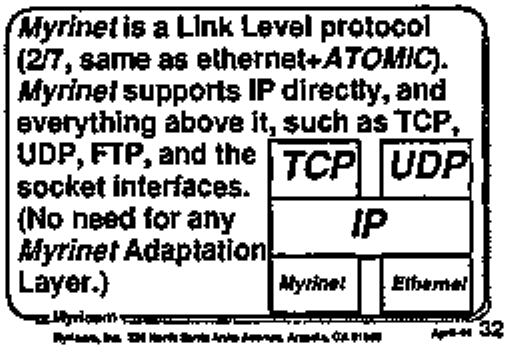
Myricom, Inc. [illegible] North Santa Anita Avenue, Arcadia, CA [illegible] April-94 30

Myrinet and fiber



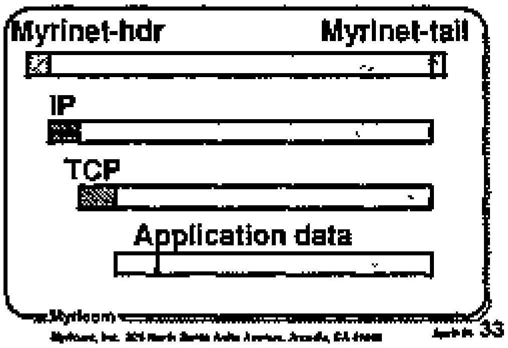
- Dialog has control symbols (including start/stop and idle)
- *Myrinet* senders are sync'd and may use external send-clock
- Therefore, it is easy to interface Myrinet channels to fiber.

Protocol Level

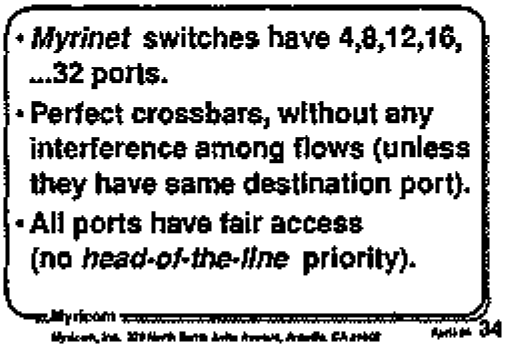


Myrinet is a Link Level protocol (2/7, same as ethernet+*ATOMIC*). *Myrinet* supports IP directly, and everything above it, such as TCP, UDP, FTP, and the socket interfaces. (No need for any *Myrinet* Adaptation Layer.)

Encapsulation

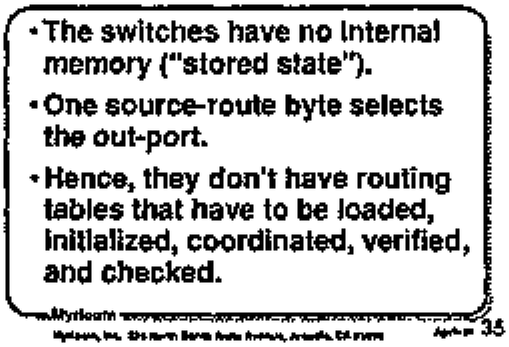


Myrinet Switches - 1



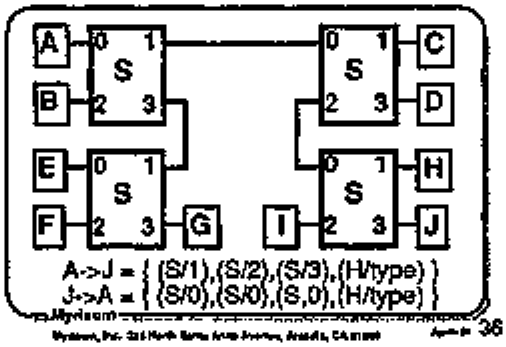
- *Myrinet* switches have 4,8,12,16, ...32 ports.
- Perfect crossbars, without any interference among flows (unless they have same destination port).
- All ports have fair access (no *head-of-the-line* priority).

Myrinet Switches - 2



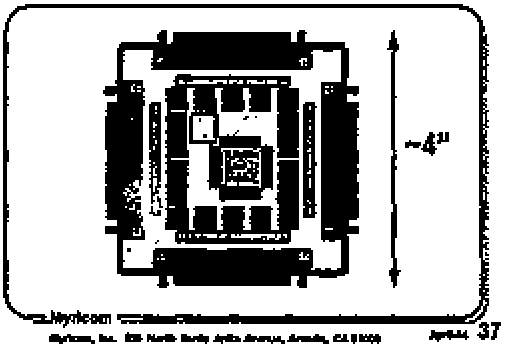
- The switches have no internal memory ("stored state").
- One source-route byte selects the out-port.
- Hence, they don't have routing tables that have to be loaded, initialized, coordinated, verified, and checked.

Myrinet Routing



A->J = { (S/1),(S/2),(S/3),(H/type) }
J->A = { (S/0),(S/0),(S,0),(H/type) }

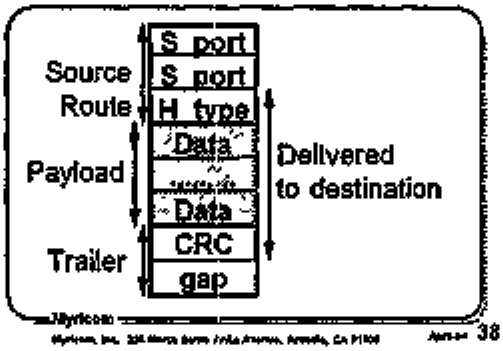
Myrinet 4-Port Switch



Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 [illegible] 37

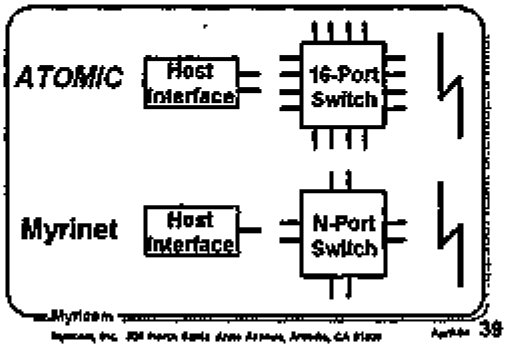
Myrinet Packet



Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 [illegible] 38

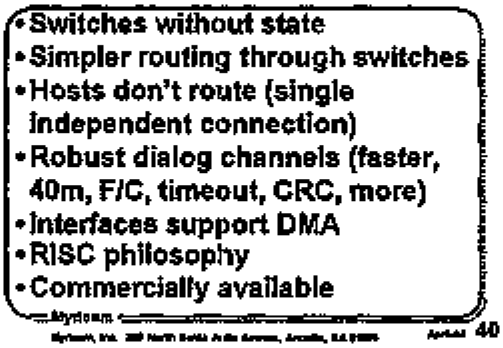
Components



Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 [illegible] 39

Myrinet vs. *ATOMIC*

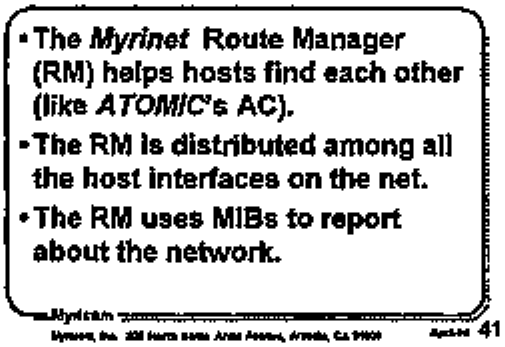


- Switches without state
- Simpler routing through switches
- Hosts don't route (single independent connection)
- Robust dialog channels (faster, 40m, F/C, timeout, CRC, more)
- Interfaces support DMA
- RISC philosophy
- Commercially available

Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 [illegible] 40

Myrinet Management

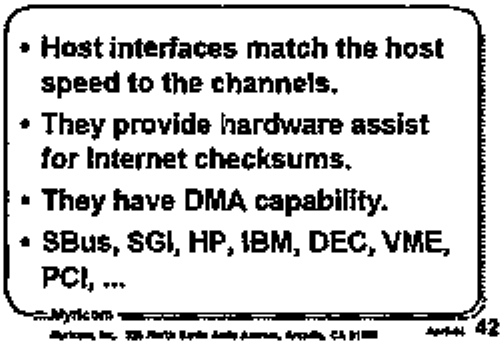


- The *Myrinet* Route Manager (RM) helps hosts find each other (like *ATOMIC*'s AC).
- The RM is distributed among all the host interfaces on the net.
- The RM uses MIBs to report about the network.

Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 [illegible] 41

Myrinet Host Interfaces

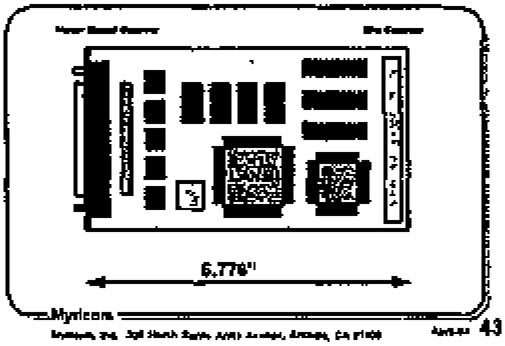


- Host interfaces match the host speed to the channels.
- They provide hardware assist for Internet checksums.
- They have DMA capability.
- SBus, SGI, HP, IBM, DEC, VME, PCI, ...

Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 [illegible] 42

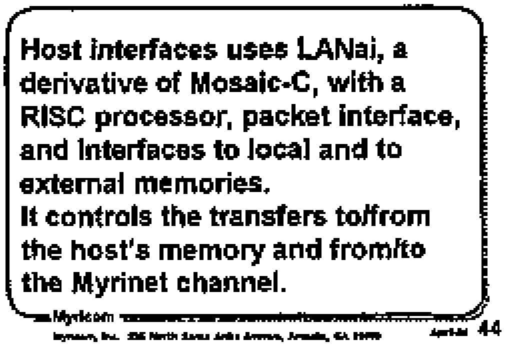
Myrinet/Sbus Interface



Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 April-94 43

The LANai Chip



Host interfaces uses LANai, a derivative of Mosaic-C, with a RISC processor, packet interface, and interfaces to local and to external memories.
It controls the transfers to/from the host's memory and from/to the Myrinet channel.

Myricom

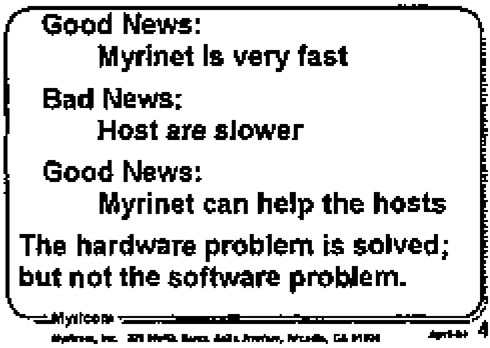
Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 April-94 44

Performance

Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 April-94 45

The News



Good News:
Myrinet is very fast

Bad News:
Host are slower

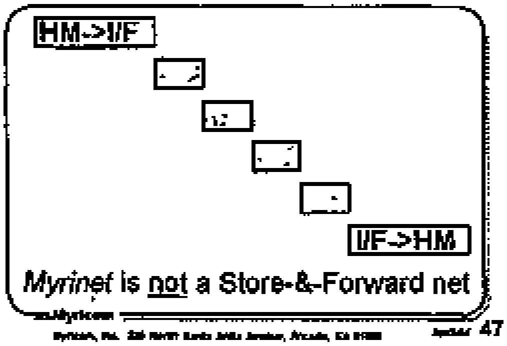
Good News:
Myrinet can help the hosts

The hardware problem is solved; but not the software problem.

Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 April-94 46

Store-&-Forward Timeline

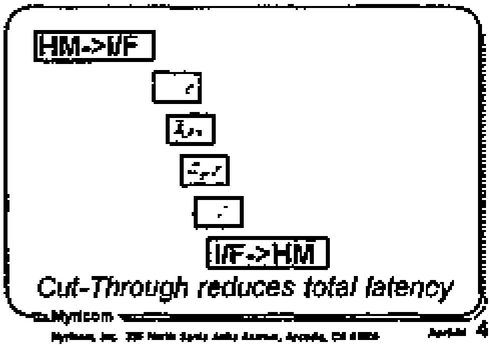


Myrinet is not a Store-&-Forward net

Myricom

Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 April-94 47

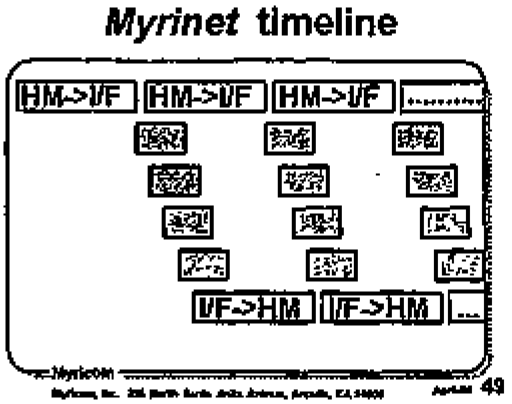
Cut-through Timeline



Cut-Through reduces total latency

Myricom

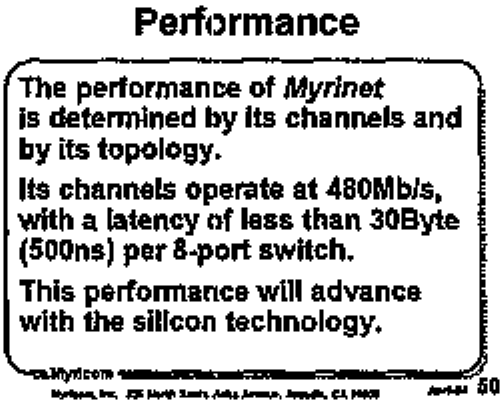
Myricom, Inc. 325 North Santa Anita Avenue, Arcadia, CA 91006 April-94 48



Myrinet timeline

Myricom

49



Performance

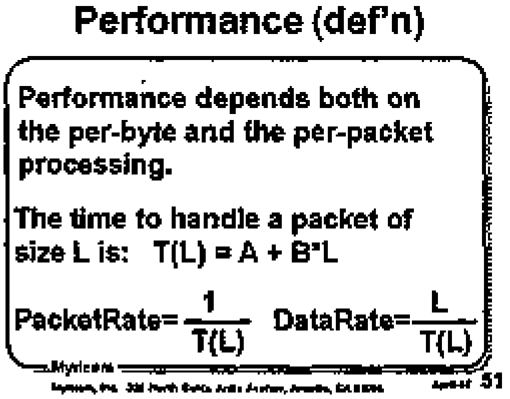
The performance of *Myrinet* is determined by its channels and by its topology.

Its channels operate at 480Mb/s, with a latency of less than 30Byte (500ns) per 8-port switch.

This performance will advance with the silicon technology.

Myricom

50



Performance (def'n)

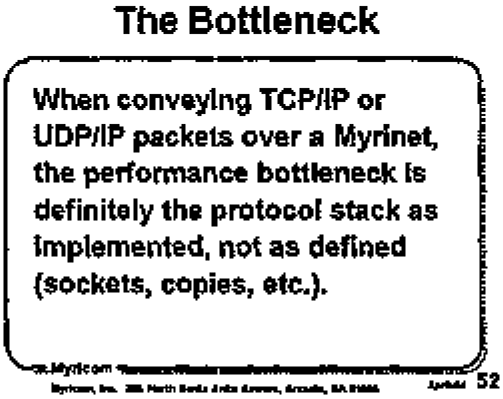
Performance depends both on the per-byte and the per-packet processing.

The time to handle a packet of size L is: $T(L) = A + B*L$

$\text{PacketRate}=\frac{1}{T(L)}$ $\text{DataRate}=\frac{L}{T(L)}$

Myricom

51

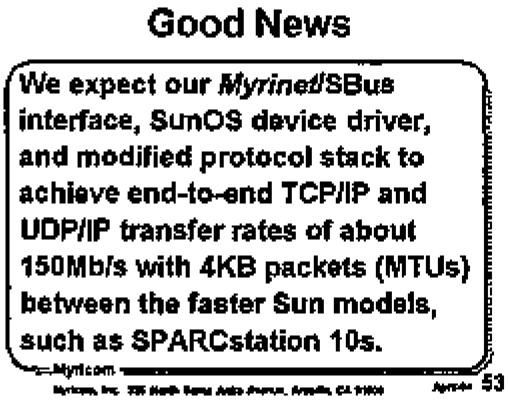


The Bottleneck

When conveying TCP/IP or UDP/IP packets over a Myrinet, the performance bottleneck is definitely the protocol stack as implemented, not as defined (sockets, copies, etc.).

Myricom

52

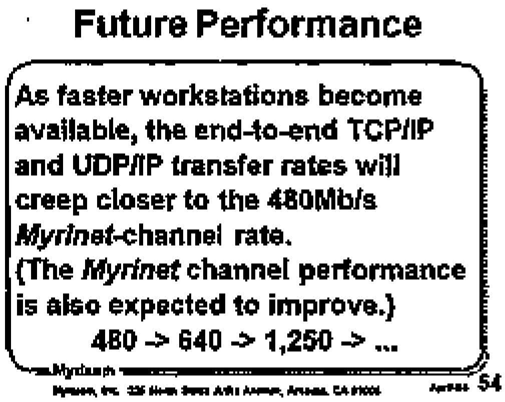


Good News

We expect our *Myrinet*/SBus interface, SunOS device driver, and modified protocol stack to achieve end-to-end TCP/IP and UDP/IP transfer rates of about 150Mb/s with 4KB packets (MTUs) between the faster Sun models, such as SPARCstation 10s.

Myricom

53



Future Performance

As faster workstations become available, the end-to-end TCP/IP and UDP/IP transfer rates will creep closer to the 480Mb/s *Myrinet*-channel rate.

(The *Myrinet* channel performance is also expected to improve.)

480 -> 640 -> 1,250 -> ...

Myricom

54

One or Zero Copies?

In contrast to this "one-copy" implementation, a "zero-copy" TCP/IP or UDP/IP is compatible with other implementations at the network level, but, is lacking the kernel-user copy, necessarily presents a different programming interface.

Zero-Copy

Programs that need the increased communication efficiency and performance of the "lighter-weight" protocol require modest changes.

At such time that Sun may start to distribute zero-copy TCP and UDP implementations, Myricom plans to support them on *Myrinet* products.

Myrinet's Own

Myricom will also provide a "feather-weight" protocol, essentially native Myrinet packets -- for use between hosts on the same *Myrinet* network, for applications such as MPI.

SBus Performance

The performance bottleneck when using this protocol is the SBus, limiting the transfer rates between the interface's and the SPARC's memory to somewhat less than 40MB/s (320Mb/s) on models with a 20MHz SBus, or to somewhat less than 50MB/s (400Mb/s) on models with a 25MHz SBus.

Expectations from SBus

We expect to be able to achieve end-to-end feather-weight transfers at about 300Mb/s in the best-case benchmarks on these SBus systems.

(The SBus guarantees that these figures will not be exceeded.)

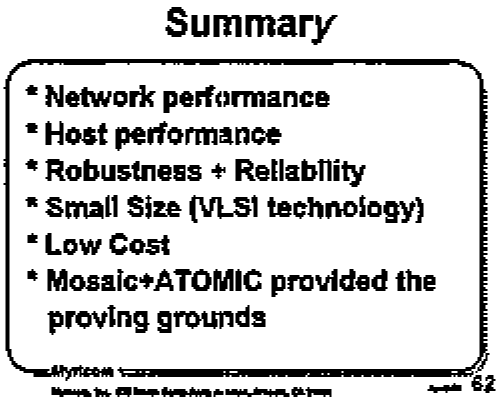
Host Performance

	Typical	1-copy	0-copy
Per-byte:			
User/Kernel copy	Host	Host	—
Checksum	Host	I/F	I/F
DMA	I/F	I/F	I/F
Per-packet:			
TCP/IP	Stack	Lighter	
Op-Sys			

Summary

Myricom

[illegible] 61

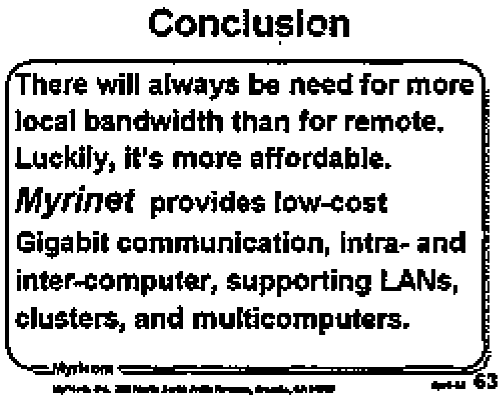


Summary

* Network performance
* Host performance
* Robustness + Reliability
* Small Size (VLSI technology)
* Low Cost
* Mosaic+ATOMIC provided the proving grounds

Myricom

[illegible] 62



Conclusion

There will always be need for more local bandwidth than for remote. Luckily, it's more affordable.

Myrinet provides low-cost Gigabit communication, intra- and inter-computer, supporting LANs, clusters, and multicomputers.

Myricom

[illegible] 63

The End

Myricom

[illegible] 64

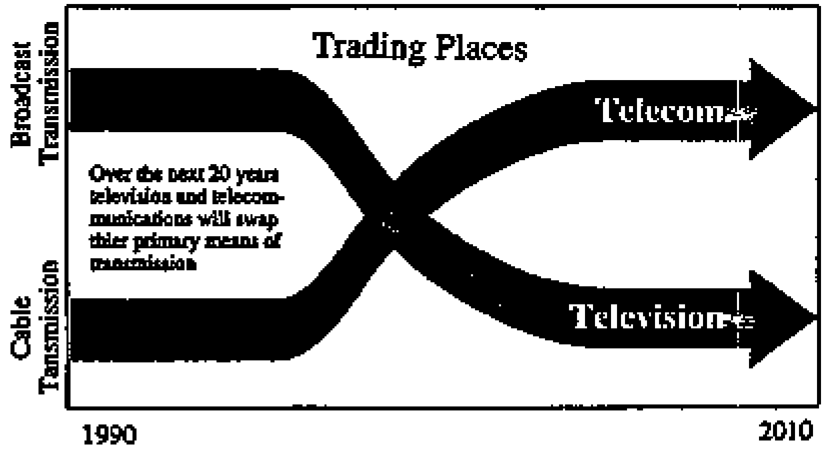
NEW FRONTIERS IN WIRELESS COMMUNICATIONS

James Stuart, Calling Communications

- Low Earth Orbit (LEO) Wireless Communications Revolution
    - 'Negroponte Flip' and 'Pelton Merge'
    - National Information Infrastructure (NII) and LEO's Role
- Wireless Satellite Services on the Horizon
- GSO Mobile and Wideband Systems on the Horizon
- LEO Communications Advantages
- LEO MSS Frequency Allocations (WARC-92)
- LEO Mobile Satellite Services (MSS) Categories and Systems on Horizon
    - LEO MSS Systems and Comparisons
    - FCC Approval Status of MSS LEO Systems
    - Contacts for 'Little', 'Big' and 'Mega' LEO Satellite Communications Systems
- Comparison of Capital Cost per Subscriber for GEO, 'Big' LEO and 'Mega' LEO Systems
- Wideband 'Mega' LEO (Teledesic) Wireless Global Network/System Features
    - Teledesic Corporation Background and Status
    - Teledesic Network and System Features
    - Teledesic Space and Ground Segment Features
- Wireless Revolution in New Service Providers and New Equipment Suppliers

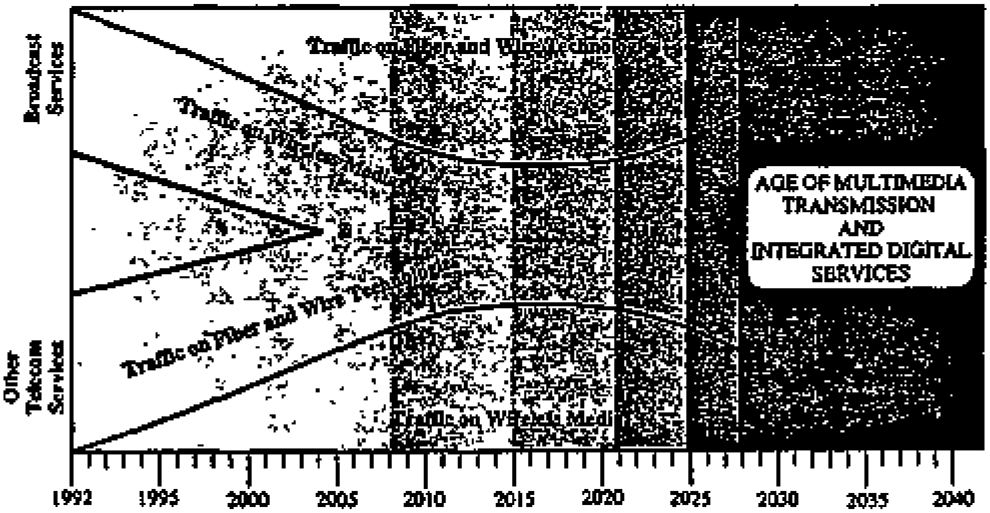
An Early View of the Communications Revolution

The "Negroponte" Flip


Trading Places
Broadcast Transmission
Cable Tansmission
Over the next 20 years television and telecommunications will swap thier primary means of transmission
Telecom
Television
1990
2010

A More Recent View of the Communications Revolution

The "Pelton" Merge



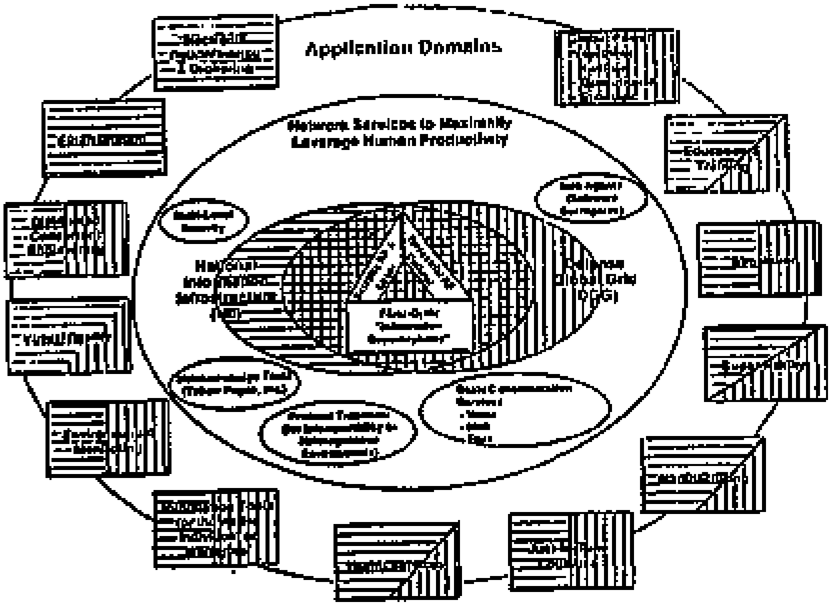
Low Earth Orbit (LEO) Wireless Communications Revolution

- LEO Communications Services Will Be Available Globally and Economically:
  - Wide Band Voice and Data, DAB, Mobile Services, Personal Communications
  - FAX , E-mail, Short messages, Monitoring, Alarms, Positioning, Tracking and Location
- Personal Ground Terminal Business Is Enormously Larger Than Space Segments
  - LEO Constellations Enable This Much Larger Business
  - Hottest New Personal Electronic Products Since PC's and VCR's Will Be:
    - Mobile Communicators, Wireless Modems, Pocket Videophones, etc.
- Shift from Last 30 Years of Satellite Communications Evolution:
  - Bigger, More Powerful, Longer Lifed Satellites
  - Hierarchical Point-To-Point Communications Architectures
- Biggest Advance in Satellite Communications in 30 Years:
  - Lightsat Networks, Intersatellite Links, Network Thinking, New Competitive Multiple-Choices, Interconnectivity, Interoperablity. Global Marketplace Determination of 'Best'
- Future Will Be Networks Of Hybrid Systems Connecting Everyone To Everyone
  - Overlaid Interconnected and Interoperable Networks
    - Terrestrial Wire, Cellular, Coaxial Cable, Fiber Optic Cable, etc.
    - GSO Large Satellites, and the New LEO, MEO and GSO Lightsats
  - Large, Competitive, Open, Diverse Global Markets
  - Multiple Service Approaches Will Become Available to All Customers
  - Continuous Evolution Of Most Effective Set of Communications Networks
    - 'One Size Fits All' is Victim to More Convenient 2nd-to-Market Choices
    - Bandwidth/Quality/Price/Convenience-On-Demand (Interoperable Choices)

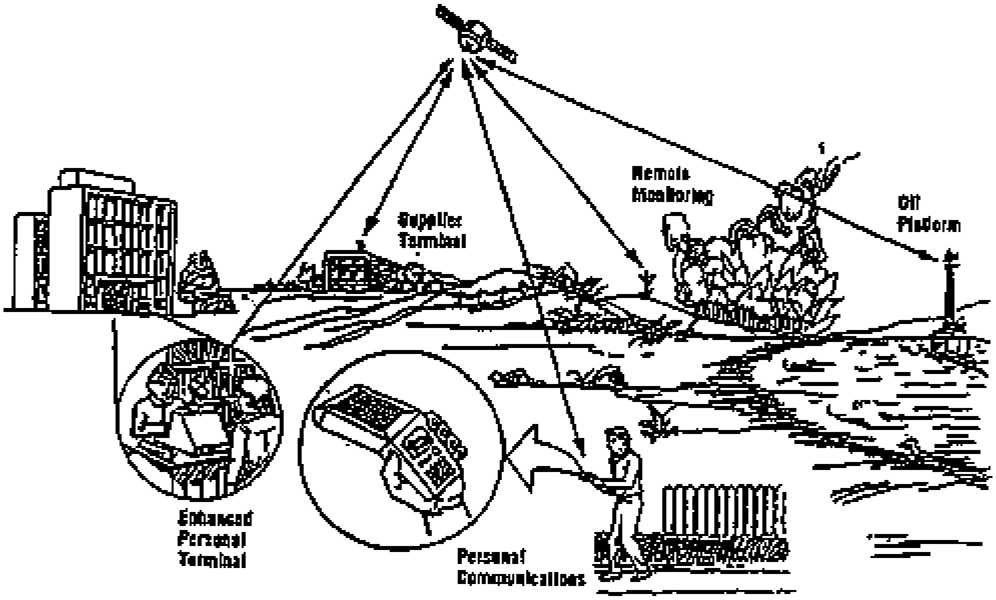
National Information Infrastructure (NII)

- NII is a Vision of a Universally Accessible Web of Multiple Interconnected Networks
  - Permitting Access to Widely Distributed Private/Public Data Bases
  - Providing Ready Transmission of Information (Voice, FAX, Text, Images, Video, etc.)
    - In Any Format, To Anyone, In Any Place, and At Anytime
- NII is an Entire NII System:
  - Human Users (and Developers)
  - User's Information Appliances (Computing and Consumer Electronics)
  - Accessed Information, Data Bases and Computing Resources
  - Networks
- The NII Network Will Be an Intricately Tangled Web of Multiple Overlaid Networks
  - Wired and Wireless
  - Terrestrial and Space
  - Physical and Virtual
  - Private, Commercial and Government
- NII (and Large Evolving Commercial Market) Will Migrate to Efficient Web Elements:
  - Reliable, Ubiquitous, Seamless, Interconnected, Flexible, Cost effective
  - Successful Elements will be Interoperable
    - 'Open' Interfaces with Accepted Standards
    - Wide Array of Competing Information Appliances and S/W Tools
      - Interoperable and Interchangeable by Design
      - Standard User-Friendly (Easy) Interfaces for H/W and S/W;
      - (e.g. Discovery/Recovery Applications, Operating Systems, etc.)
    - Many Interchangeable Computing Service Providers and Equipment Suppliers

A Current View of LEOs Role in the National Information Infrastructure (NII)


Application Domains
Network Services to Maximally Leverage Human Productivity
Multi-Level Security
National Information Infrastructure (NII)
Defense Global Grid (DGG)
Education & Training
Basic Communication Services
- Voice
- Data
- Fax

Wireless Satellite Services on the Horizon



GSO Mobile and Wideband Communications Systems on the Horizon

- MSS GSOs' Above 1 GHz:
  - AMSC (MSAT), USA
  - Celsat (Celstar), USA
  - INMARSAT (Inmarsat), Europe
  - Mexico (Solidaridad), Mexico
  - Telesat (MSAT), Canada
  - (+Other National and Regional GEO's)
- MSS GSOs' at 20-30 GHz:
  - ISAS (ETS-VI), Japan
  - Norris (Norstar), USA
  - NTT (N-Star), Japan
  - NASA (ACTS), USA
    - High Power, Satellite-Switched, Multi-Beam, Ka-Band Satellite
      - 2-4 Year Lifetime, in 100°W Slot,
      - 6 Ka-Band TWTA's (20 GHz, 46W) through 3.3 m Antenna
      - 3 Fixed Beams (Cleveland, Atlanta, Tampa)
      - Steerable Hopping Spot Beam (650 km diam., <1 ms dwell)
      - 64 kbps - 1.54 Mbps (64 kbps increments, Bandwidth on Demand)
    - ACTS Satellite Time Grants for ACTS Experimenters
      - >70 Experimenters to Date
        - 10 Service Providers, 8 Equipment Providers,
        - 16 Non-Gov't End Users, 18 Gov't End Users,
        - 17 Universities, e.g. University of Colorado at Boulder (CU)
          (CU has T-1 Earth Station, etc. Available for Joint Experiments)

Contact for ACTS Experiment Info: Tom Meyer (303) 494-8144

ACTS Mobile Experiments (Examples)

Experiment Title	*Principal Investigator*	*Experiment Description*
Land Mobile	Jet Propulsion Laboratory	Mobile Terminal Verification, Propagation
Secure Mobile Communications	National Communications System	Secure (STU-III) Land Mobile Communications for National Security
C2 On-the-Move	U.S. Army	Military Land Mobile Communications for Command and Control
Emergency Medical Land Mobile Satellite Communications	EMSAT, Advanced Technology For Emergency Medical Services	Emergency Land Mobile Communications For Paramedics
Telemedicine	University of Washington Medical Center	MRI and CT Transmissions Between Hospitals And Mobile Units
Satellite News Gathering	IDB Communications	Remote Broadcast For Satellite News Gathering
Satellite News Vehicle	NBC	Mobile Communications For Remote News Vans
Satellite/Terrestrial Personal Communications Services	Bellcore	Nomadic End-User Interface To Personal Communications Networks
High Quality Audio	CBS Radio	Direct Broadcast Audio Experiments
Aero-X	JPL/NASA LeRC	Low Data Rate Aeronautical Technology Verification
Aeronautical Tracking and High Data Rate Communications	Rockwell/Collins	Compressed Full Motion Video, Small Steerable Satellite Antenna

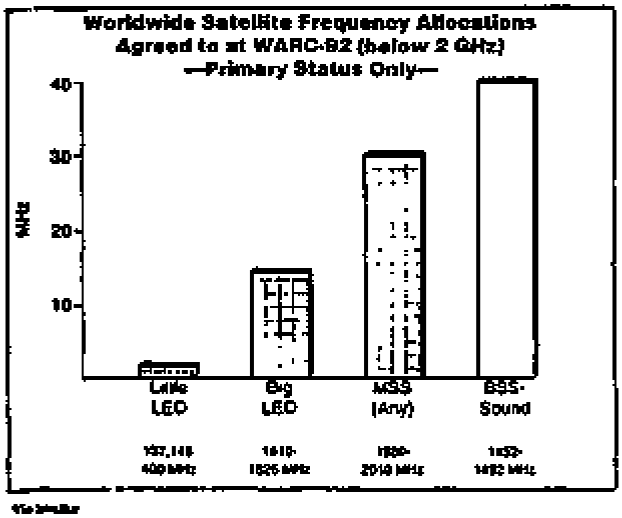
Communications Advantages of LEO Systems

- LEO/MEO Constellations Competitive Advantages over GSO (by factors of 10)
- LEO Performance Advantages
  - Communications Uniformity
  - Communications Time Delay: LEO/MEO Orbits
  - Increased Link Margins (50 times closer than GSO)
  - High Frequency Re-Usage (smaller footprints)
  - Reliability: Small SSPA's (versus HPA TWTA's)
  - Redundancy: On-Orbit Spare Lightsats
  - Graceful Introduction Of New Technologies
  - Inherent Doppler Shift For Position Determination
- LEO Price Advantages
  - Volume Production Methods, Economies Of Scale (Satellites and Launchers)
  - Smaller Launchers and Piggyback Launch Opprotunities
  - Reduced Insurance Costs and Debt Service (Smaller Capacity Demands)
  - Smaller User Terminals (50 times closer than GSO)
  - Incremental Increase In Capacity Can Follow Actual Demand
    - Investment Capital Able to be Coupled To Revenue Flow
  - LEO Network Vastly Superior In Invested Capital Required per World Subscriber
    - 'Last Mile' (Remote In Most of World)
    - Marginal Cost Of New Subscribers, etc.
    - Construction, Operation, Maintenance, Improvements, etc.

WARC-92 LEO MSS Frequency Allocations

WARC 92 (Torremolinos, Spain) Concluded on 3 March 1992

New Worldwide Frequency Allocations for MSS:



20-30 GHz Previously Allocated Worldwide for FSS/MSS

LEO Mobile Satellite Service (MSS) Categories

- MSS "Little" LEO's Below 1 GHz

    Noncontinuous Worldwide Coverage,
    No intersatellite Links (yet), "Store-and-Forward"
    Gateways, PSTN Connections, By-pass option
    Non-RealTime and Near-Real Time Digital Mobile Services (~2.4-9.6 kbps)
        Short Digital Messages, Alarms, Monitoring Data, Positioning, Tracking
        E-Mail Text pages, FAX pages
    Typical Delivery Delay Times
        Within Footprint (~4000km Diameter): ≤10-30 minutes
        International (e.g., USA-Europe): 30 minutes - 8 hours
    Typical Subscriber Costs
        Terminals: $500-$100
        Data: 1.0¢ -0.001¢ per byte

- MSS "Big" LEO's Above 1 GHz

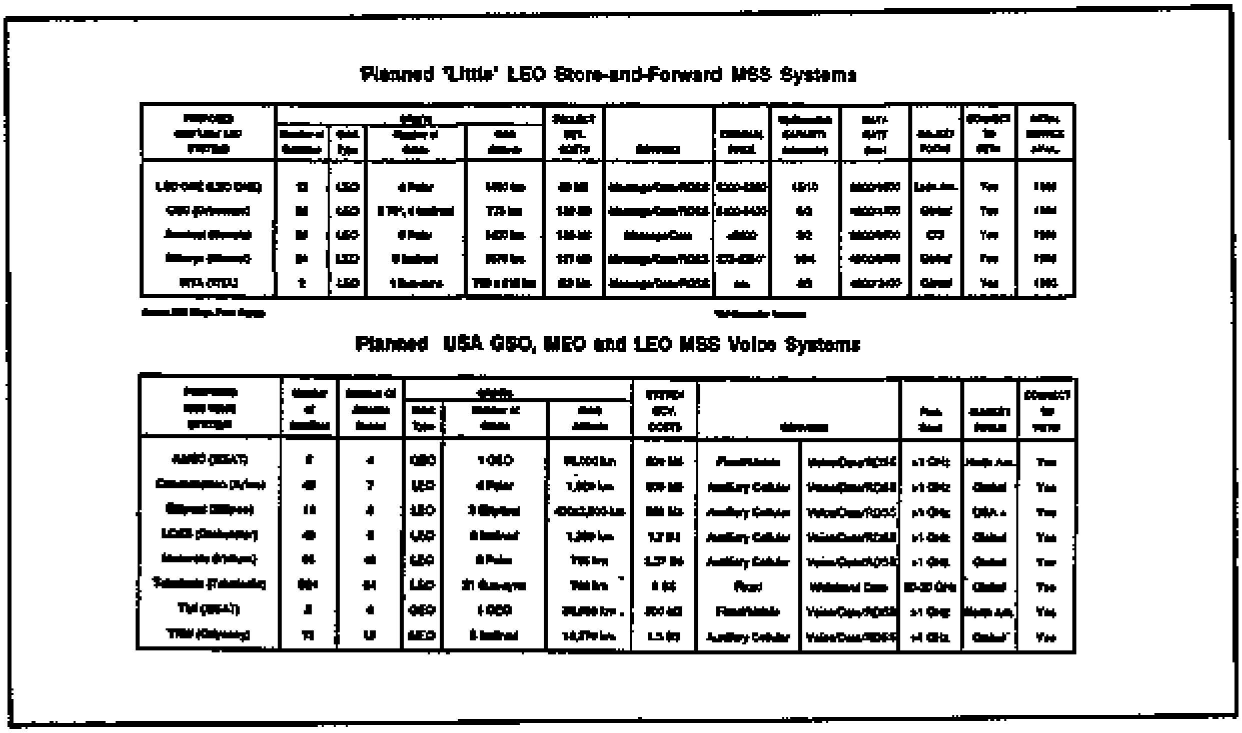
    Continuous Worldwide Coverage (<Terrestrial Dial-tone Availability)
    With/without Intersatellite Links, Gateways, PSTN Connections
    Local Cellular Company Size (largest: ~250,000 Subscribers at 0.1 *Erlang*)
        Real Time Mobile Services (~ 4.8 kbps)
        Digital Voice, Narrowband Data (<Toll Quality)
    Typical Long Distance Delay Times: ~Terrestrial Delays
    Typical Subscriber Costs
        Terminals: $1000-$500
        Data: $3.00-$0.50 per minute

LEO Mobile Satellite Service (MSS) Categories

- FSS Wideband "Mega" LEO's at 20-30 GHz
  - Continuous Worldwide Coverage (Terrestrial Dial-tone Availability)
  - Regional Bell Operating Company Size
    - > 20,000,000 Subscribers at 0.1 Erlang
  - Intersatellite Links, Gateways, PSTN
  - Real Time Fixed and Mobile Services (16 kbps-1.2 Gbps)
    - Bandwidth On Demand
      - 16 kbps - 2 Mbps (using Teledesic Standard Terminals)
      - 155 Mbps - 1.2 Gbps (using Teledesic 'GigaLink' Terminals)
    - Wideband Data, Video, Digital Voice (Toll Quality)
    - All Typical Phone Company Services and Features
    - Typical Long Distance Delay Times: < Fiber
  - Typical Subscriber Costs
    - Terminals: ~$1,500 (and falling sharply with volume and competition)
~$7,500 for 'Gigalink' Terminals (and falling sharply)
    - Data: Comparable to local PTT charges (≤few ¢ per minute)

LEO Mobile Satellite Communications Systems on the Horizon

- MSS "Little" LEO's Below 1 GHz
  - LEO ONE Panamericana (LEO ONE), Mexico
    12 Satellites (4 inclined 1100 km orbits)
  - OSC (Orbcomm), USA
    36 Satellites (2 ~polar and 4 inclined 775 km orbits)
  - Smolsat (Gonetz), Russia
    36 Satellites (6 inclined 1400 km orbits)
  - Starsys (Starnet), USA
    24 Satellites (6 inclined 1,370 km orbits)
  - VITA (VITA), USA
    2 Satellites (1 sun-sync 798 x 815 km orbit)
- MSS "Big" LEO's Above 1 GHz
  - Constellation Communications, Inc. (Aries), USA
    48 Satellites (4 polar 1,020 km orbits)
  - Ellipsat Corp. (Ellipso), USA
    16 Satellites (3 elliptical ~500 x 7,800 km orbits)
  - Loral/Qualcomm Satellite Services (Globalstar), USA
    48 Satellites (8 inclined 1,414 km orbits)
  - Motorola (Iridium), USA
    66 Satellites (6 polar 780 km orbits)
  - TRW (Odyssey), USA
    12 Satellites (3 inclined 10,370 km orbits)
- FSS Wide-band "Mega" LEO's at 20-30 GHz
  - Teledesic Corporation (Teledesic), USA
    924 Satellites (21 sun-sync, 700km orbits)



Planned 'Little' LEO Store-and-Forward MSS Systems

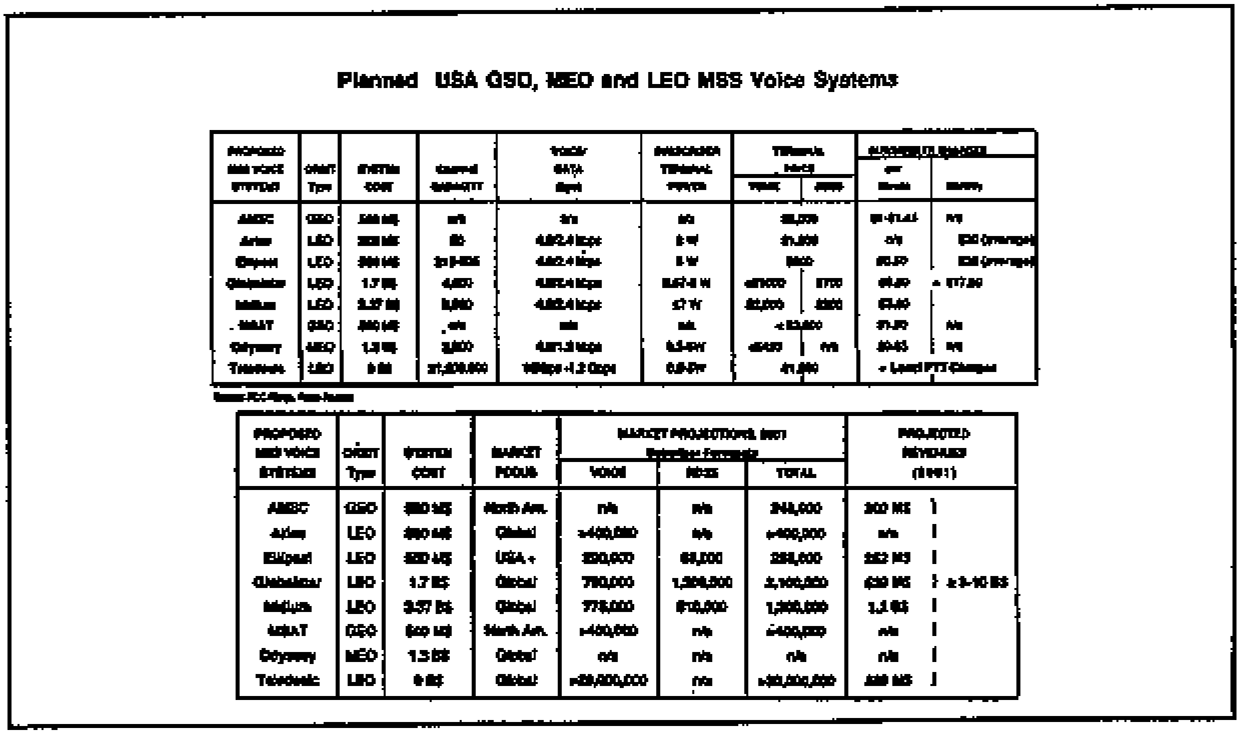
PROPOSED [illegible] LEO SYSTEM	ORBITS				PROJECT SYS. COSTS	SERVICES	TERMINAL PRICE	[illegible] CAPACITY	DATA RATE	MARKET FOCUS	CONNECT to PSTN	INITIAL SERVICE
	Number of Satellites	Orbit Type	Number of Orbits	Orbit Altitude								
LEO ONE (LEO ONE)	12	LEO	4 Polar	950 km	49 M$	Message/Data/RDSS	$300-$500	15/10	2400/9600	Latin Am.	Yes	1996
[illegible]	36	LEO	3 78°, 4 Inclined	775 km	160 M$	Message/Data/RDSS	$400-$400	60	4800/2400	Global	Yes	1996
[illegible]	24	LEO	6 Polar	1300 km	180 M$	Message/Data	<$100	32	[illegible]	US	Yes	1996
[illegible]	24	LEO	6 Inclined	970 km	177 M$	Message/Data/RDSS	[illegible]	164	[illegible]	Global	Yes	1996
VITA (VITA)	2	LEO	1 Sun-sync	750 x 815 km	20 M$	Message/Data/RDSS	n/a	40	[illegible]	Global	Yes	1995

Source: [illegible]

*[illegible]

Planned USA GSO, MEO and LEO MSS Voice Systems

PROPOSED MSS VOICE SYSTEMS	Number of Satellites	Number of [illegible]	ORBITS			SYSTEM EST. COSTS	SERVICES		Freq. Band	MARKET FOCUS	CONNECT to PSTN
			Orbit Type	Number of Orbits	Orbit Altitude						
AMSC (MSAT)	2	4	GEO	1 GEO	35,000 km	500 M$	Fixed/Mobile	Voice/Data/RDSS	<1 GHz	North Am.	Yes
Constellation (Aries)	48	7	LEO	4 Polar	1,020 km	270 M$	Auxiliary Cellular	Voice/Data/RDSS	<1 GHz	Global	Yes
Ellipsat (Ellipso)	14	4	LEO	3 Elliptical	4200x7,800 km	560 M$	Auxiliary Cellular	Voice/Data/RDSS	<1 GHz	USA +	Yes
LQSS (Globalstar)	48	8	LEO	8 Inclined	1,389 km	1.7 B$	Auxiliary Cellular	Voice/Data/RDSS	<1 GHz	Global	Yes
Motorola (Iridium)	66	48	LEO	6 Polar	765 km	3.37 B$	Auxiliary Cellular	Voice/Data/RDSS	<1 GHz	Global	Yes
Teledesic (Teledesic)	840	64	LEO	21 Sun-sync	700 km	9 B$	Fixed	Wideband Data	20-30 GHz	Global	Yes
TRI (MSAT)	2	4	GEO	1 GEO	35,800 km	500 M$	Fixed/Mobile	Voice/Data/RDSS	<1 GHz	North Am.	Yes
TRW (Odyssey)	12	12	MEO	3 Inclined	10,370 km	1.3 B$	Auxiliary Cellular	Voice/Data/RDSS	<1 GHz	Global	Yes



Planned USA GSO, MEO and LEO MSS Voice Systems

PROPOSED MSS VOICE SYSTEMS	ORBIT Type	SYSTEM COST	Channel CAPACITY	VOICE/ DATA Rate	HANDHELD TERMINAL POWER	TERMINAL PRICE		SUBSCRIBER CHARGES	
						[illegible]	[illegible]	per Minute	Monthly
AMSC	GSO	560 M$	n/a	n/a	n/a	$2,500		$1-$1.45	n/a
Aries	LEO	300 M$	80	4.8/2.4 kbps	5 W	$1,500		n/a	$30 (average)
Ellipsat	LEO	560 M$	[illegible]	4.8/2.4 kbps	5 W	$800		$0.50	$36 (average)
Globalstar	LEO	1.7 B$	4,800	4.8/2.4 kbps	[illegible] W	<$1000	$700	$0.30	+ $17.50
Iridium	LEO	3.37 B$	3,840	4.8/2.4 kbps	0.7 W	$2,000	$300	$3.40	
MSAT	GSO	500 M$	n/a	n/a	n/a	< $2,500		$1.50	n/a
Odyssey	MEO	1.3 B$	2,300	4.8/1.2 kbps	0.5 W	<$400	n/a	$0.65	n/a
Teledesic	LEO	9 B$	21,000,000	16 kbps - 1.2 Gbps	0.3 W	$1,000		+ Local PTT Charges	

Source: FCC Filings, [illegible]

PROPOSED MSS VOICE SYSTEMS	ORBIT Type	SYSTEM COST	MARKET FOCUS	MARKET PROJECTIONS [illegible] Subscriber Forecasts			PROJECTED REVENUES ([illegible])	
				VOICE	RDSS	TOTAL		
AMSC	GSO	560 M$	North Am.	n/a	n/a	240,000	300 M$	
Aries	LEO	300 M$	Global	>400,000	n/a	>400,000	n/a	
Ellipsat	LEO	560 M$	USA +	200,000	68,000	268,000	262 M$	
Globalstar	LEO	1.7 B$	Global	780,000	1,320,000	2,100,000	630 M$	≥ 3-10 B$
Iridium	LEO	3.37 B$	Global	775,000	610,000	1,385,000	1.1 B$	
MSAT	GSO	500 M$	North Am.	>400,000	n/a	>400,000	n/a	
Odyssey	MEO	1.3 B$	Global	n/a	n/a	n/a	n/a	
Teledesic	LEO	9 B$	Global	>20,000,000	n/a	>20,000,000	400 M$	

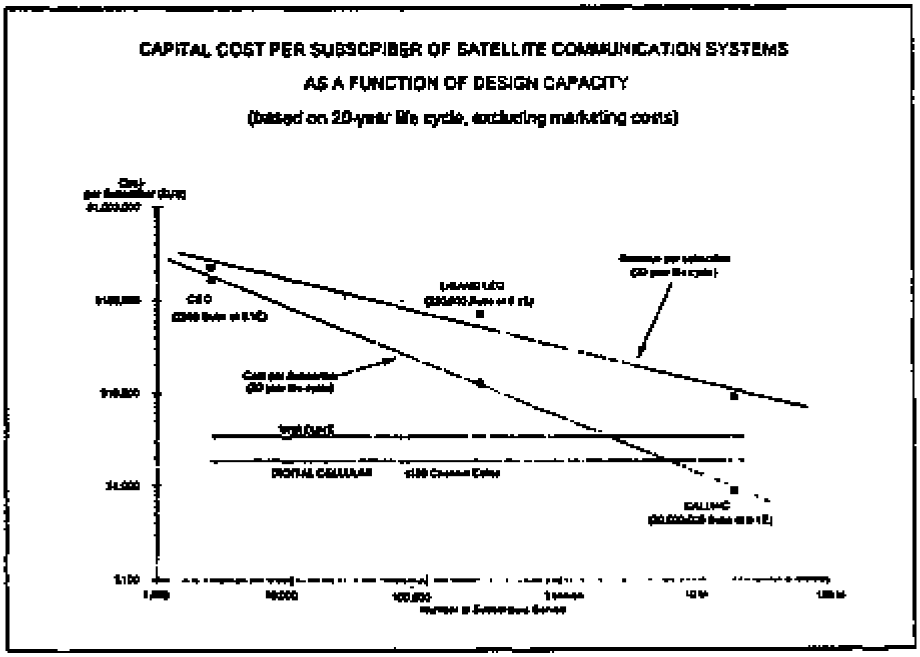
FCC Approval Status of MSS LEO Systems

- "Little" LEO's (Below 1 GHz)
  - Experimental Licenses Granted to 3 of the Little LEO's
    - OSC (Orbcomm): 2 satellites
    - Starsys (Starnet): ground tests
    - VITA (VITASAT): 1 satellite
  - Pioneer's Preference Designation Awarded to VITA
  - "Little" LEO's Agreed on Recommendations to FCC for Rulemaking
    - Notice of Proposed Rulemaking Issued (Band Segmentation)
- "Big" LEO's (Above 1 GHz)
  - Experimental Licenses Granted to 4 of the 5 Big LEO's
    - Constellation Communications, Inc. (Aries): 2 satellites
    - Ellipsat (Ellipso): 6 satellites
    - Motorola (Iridium): 5 satellites
    - TRW (Odyssey): ground tests
  - Pioneer's Preference Designation Denied to All the Big LEO's
  - "Big" LEO's unable to Agree on Recommendations for FCC Rulemaking
    - Notice of Proposed Rulemaking Issued (Band Segmentation)
- "Mega" LEO's (at 20-30 GHz)
  - 20-30 GHz Already Allocated Worldwide for FSS/MSS by ITU
  - FCC Filing Submitted by Teledesic Corporation (21 March 1994)

Contacts for LEO Mobile Satellite Communications Systems

- "Little" LEO Contacts
  - Juan F. Gomez, LEO ONE Panamericana (LEO ONE)
    (202) 429-2023, fax (202) 626-6282
  - Alan Parker, OSC (Orbcomm)
    (703) 818-3762, fax (703) 631-3610
  - Vern Riportella, SmalSat (Gonetz)
    (914) 986-6904, fax (914) 985-3875
  - Dr. Ashok Kaveeshwar, Starsys (Starnet)
    (301) 459-8832, fax (301) 794-7106
  - Dr. Gary Garriott, VITA (VITA)
    (703) 276-1800, fax (703) 243-1865
- "Big" LEO Contacts
  - Bruce D. Kraselsky, Constellation Communications, Inc. (Aries)
    (703) 733-2819, fax (703) 733-2827
  - Dr. David Castiel, Ellipsat Corp. (Ellipso)
    (202) 466-4488, fax (202) 466-4493
  - Douglas G. Dwyre, Loral Qualcomm Satellite Services (Gobalstar)
    (301) 805-0591, fax (301) 805-0591
  - Robert W. Kinzie, Motorola (Iridium)
    (202) 944-5109, fax (202) 842-0006
  - Roger J. Rusch, TRW (Odyssey), USA
    (310) 814-5927, fax (310) 813-7535
- "Mega" LEO Contact
  - Mark Lawrence, Teledesic Corporation (Teledesic)
    (818) 896-0671, fax (818) 962-0758

Capital Cost Per Subscriber for Generic GEO, 'Big' LEO and 'Mega' LEO Systems



CAPITAL COST PER SUBSCRIBER OF SATELLITE COMMUNICATION SYSTEMS

AS A FUNCTION OF DESIGN CAPACITY

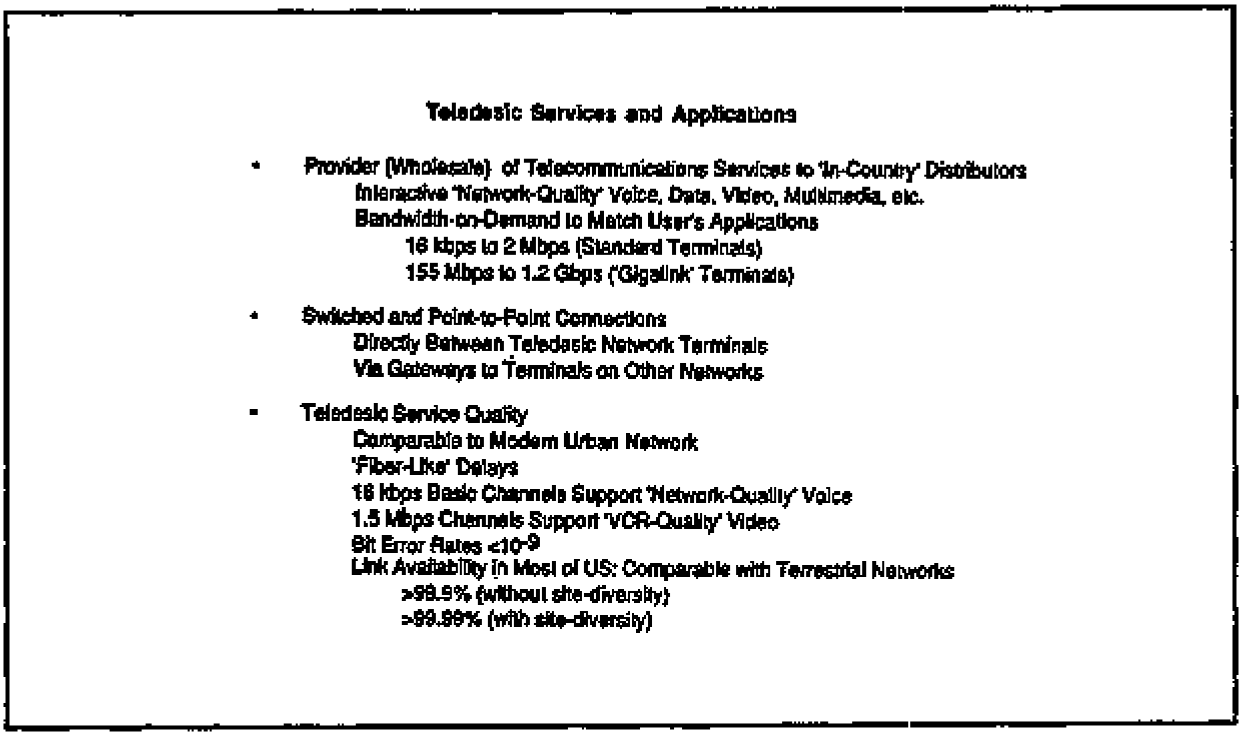
(based on 20-year life cycle, excluding marketing costs)

Teledesic Corporation Background and Status

- Teledesic Background
  - Founded in July, 1990 (as Calling Communications Corp.)
  - Concept Originally Developed (reduced to writing) in 1988
  - Current (and Original) Corporate Mission Statement:

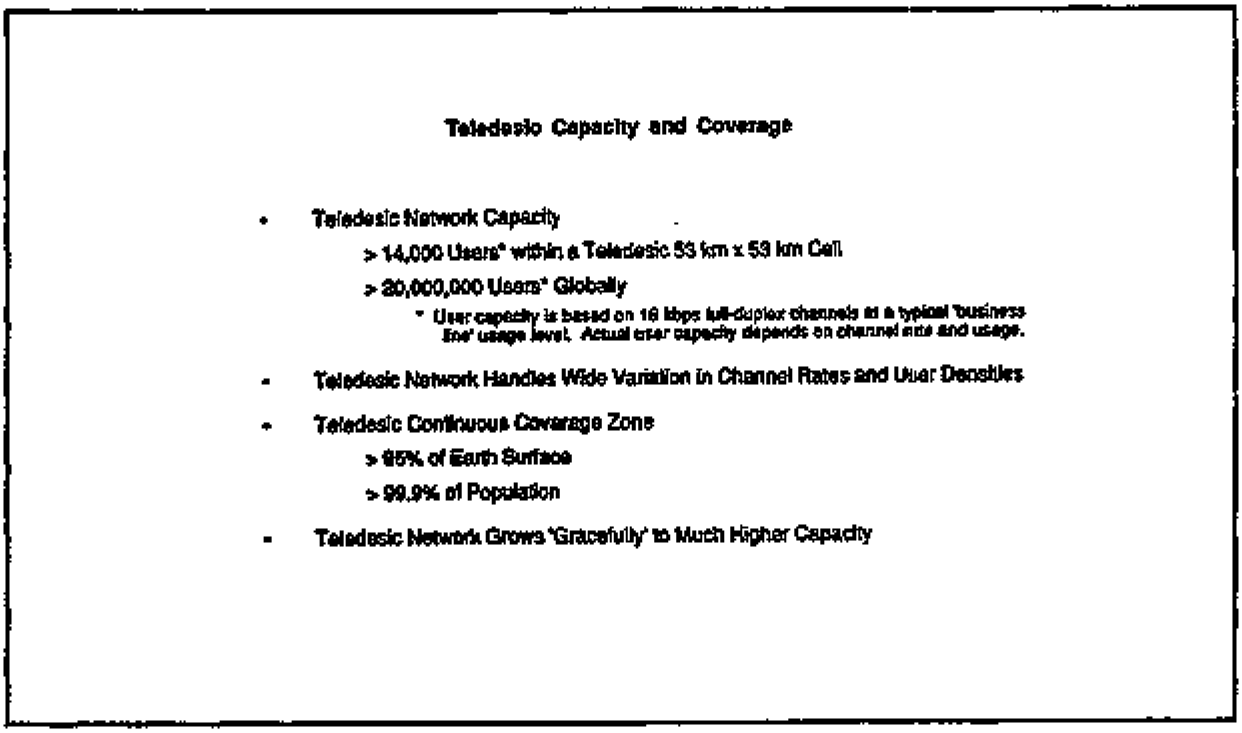
> "Our goal is to build, as rapidly as possible, a privately-owned system to provide telephone and data service, with quality at least equal to the best service available anywhere . . . between randomly selected points on Earth at any time . . . The system must grow smoothly to carry a substantial portion of Earth's . . . traffic in the 21st Century."

- Teledesic Status
  - Feasibility Study and Point Design (Phase A) Completed
    - > 3.5 Years by Extraodinary Team of Full-time Employees, Consultants and Selected Subcontractors (Led by Ed F. Tuck, Kinship Partners II)
    - - Continuous Internal Peer and Periodic External Critical Supplier Reviews
    - - External Teledesic-Contracted Technical Reviews /Design Audits
      - NASA/JPL (3 Reviews: Nov/Dec. 1993)
      - A Major Aerospace Prime Contractor (April 1994)
  - FCC Application Filed (3/94)
  - Primary Shareholders:
    - - Mr. Craig O. McCaw
    - - Mr. William H. Gates
    - - McCaw Development, Inc.
    - - Kinship Partners-II
  - Headquarters in Kirkland, WA (President: W. Russell Daggatt)
    - - Distributed Program Development Team



Teledesic Services and Applications

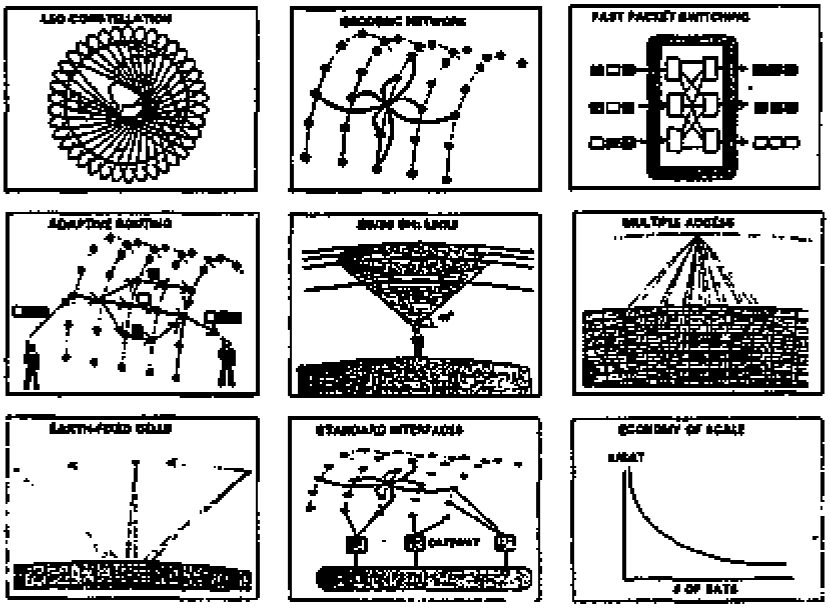
- Provider (Wholesale) of Telecommunications Services to 'In-Country' Distributors
  - Interactive 'Network-Quality' Voice, Data, Video, Multimedia, etc.
  - Bandwidth-on-Demand to Match User's Applications
    - 16 kbps to 2 Mbps (Standard Terminals)
    - 155 Mbps to 1.2 Gbps ('Gigalink' Terminals)
- Switched and Point-to-Point Connections
  - Directly Between Teledesic Network Terminals
  - Via Gateways to Terminals on Other Networks
- Teledesic Service Quality
  - Comparable to Modem Urban Network
  - 'Fiber-Like' Delays
  - 16 kbps Basic Channels Support 'Network-Quality' Voice
  - 1.5 Mbps Channels Support 'VCR-Quality' Video
  - Bit Error Rates $<10^{-9}$
  - Link Availability in Most of US: Comparable with Terrestrial Networks
    - >99.9% (without site-diversity)
    - >99.99% (with site-diversity)



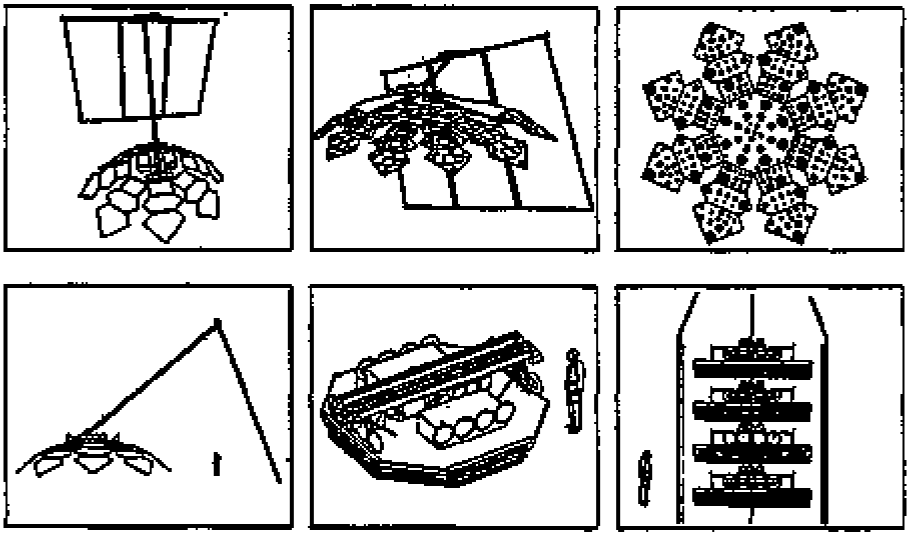
Teledesic Capacity and Coverage

- Teledesic Network Capacity
  - > 14,000 Users* within a Teledesic 53 km x 53 km Cell
  - > 20,000,000 Users* Globally
    - * User capacity is based on 16 kbps full-duplex channels at a typical 'business line' usage level. Actual user capacity depends on channel size and usage.
- Teledesic Network Handles Wide Variation in Channel Rates and User Densities
- Teledesic Continuous Coverage Zone
  - > 95% of Earth Surface
  - > 99.9% of Population
- Teledesic Network Grows 'Gracefully' to Much Higher Capacity

'Mega' LEO Wideband Network and System Features



Teledesic Satellite Configuration Features



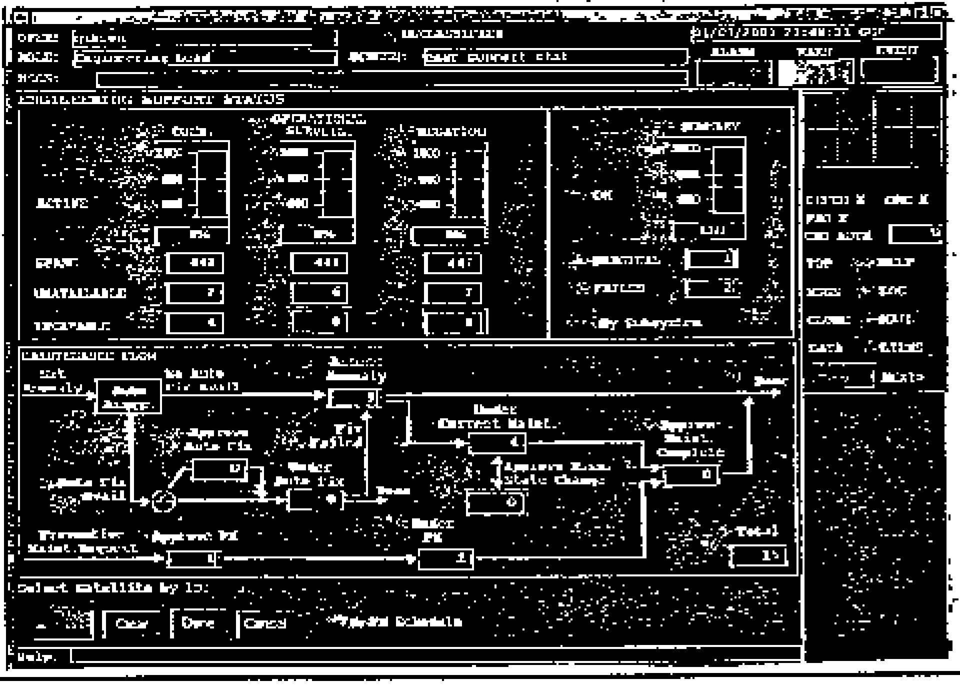
Teledesic Space Segment Key Features

- Modern, High Performance, High Power, Mass-Producible Satellite System
  - Identical 3-Axis Stabilized Satellites for All Constellation Positions
  - High Performance, High Reliability, 10 year Lifetime Satellite System
    - High Power (>6.6 kW EOL, 15 kW surge capability)
    - High Computational Power (>300 MIPs, >2Gbytes RAM)
    - High ΔV Low-Thrust Propulsion (>1000 mps)
    - Lightweight (750 kg)
  - Robust Phase A Point Design with Large Design Margins
    - > 20% in Mass, Volume
    - > 40% in Power
    - > 85% in Propulsive ΔV
    - > 300% in MIPS and 200% in RAM
    - > 9% in Reliability
- Design Features Tailored Specifically for Large Constellation
  - High Volume Production of Components (Large Economies of Scale)
  - Automated Integration and Test of Satellite Systems (On-Board Test S/W)
  - Self-Stacked, Self-Deployed Group Launch by Variety of Launchers
  - Automatic Orbit Transfer, Insertion and Gap-Filling
  - Autonomatic On-Orbit Health Monitoring and Constellation Control
  - Active On-Orbit Spares (Routine Block Replenishments)
  - Reliable End-of-Life Disposal/Deorbit Capability
- Modern Technology and Architecture Baseline (Phase A Point Design)
  - Current, Costable, Mass-Producible Technologies and Components
  - Multiple Existing Aerospace Suppliers and Estimates for All Components
  - Existing USA and International Launchers for Performance/Cost Estimates

Teledesic Ground Segment Key Elements

- Terminals
  - Standard Terminals: 16 kbps to 2 Mbps
  - 'Gigalink' Terminals: 155 Mbps to 1.2 Gbps
  - COCC, NOCC, SPAC Gateways (1.2 Gbps)
- Network Operations and Control Centers (NOCC)
  - Redundant Facilities, providing e.g.,
    - Subscriber and Network Databases
    - Feature Processors
    - Network Management
    - Global Administration and Billing Systems
  - Owned and Operated by Teledesic
- Service Provider Administration Centers (SPAC)
  - Redundant Gateway Antennas
  - Regional Administration and Billing Systems
  - Owned and Operated by Service Provider
- Constellation Operations and Control Centers (COCC)
  - Redundant Facilities for 4 Teams
    - Launch/Initialization/ReplacementTeam
    - Health Monitoring/Failure DetectionTeam
    - Diagnostic Team
    - Disposal/Deorbit Team
  - Owned and Operated by Teledesic

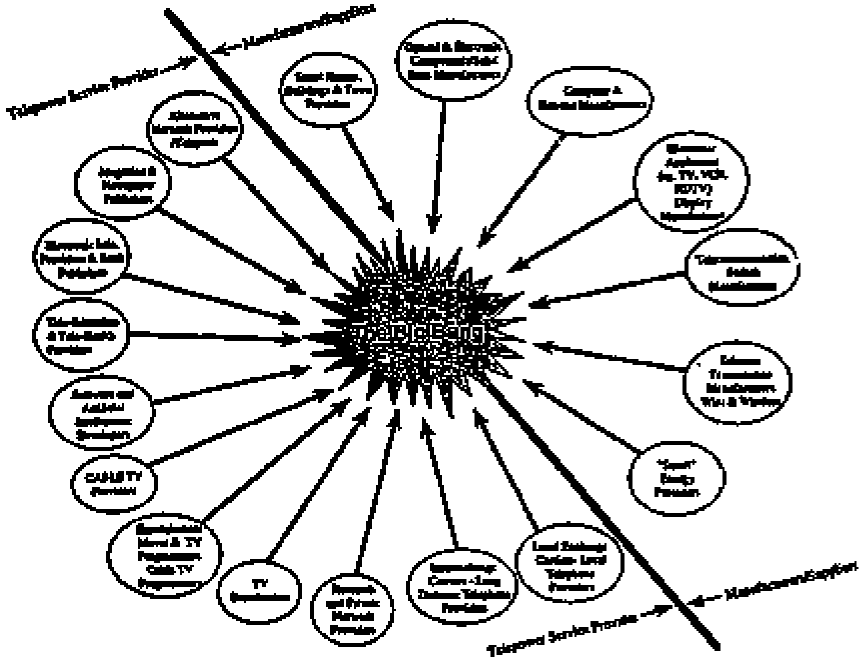
Teledesic Ground Control Display Example



THE TELEDESIC SPACE SEGMENT KEY TECHNOLOGIES

Baseline Modern Space Technologies	Technology Back-ups	Enhanced Technology Alternatives
Pulse Plasma Electric Thrusters (Teflon, 60 kN-s,1200 Isp)	Hall Thrusters Arc-Jets	Deflagration Thrusters
Amorphous Silicon Thin Film Solar Array (4%EOL)	Crystal Si, GaAS Multi-junction Concentrators	Amorphous Si (6.5%EOL) Thin Film CIS (copper indium diselenide, 10%) GaAS/CIS (16%)
NiMH Batteries	NiH2 (CPV) Batteries	Lithium Ion Batteries Thin Film Polymer Batteries
High Performance Microprocessors (PC603)	R58000,1750, 68020	Pentium, PC604, etc.
Paraffin (HOP) Latch/Deploy Mechanisms	Motors, Spring/Dampers	Shape Memory Mechanisms
Inflatable Solar Array Booms	Bistem Booms Cont. Longeron Booms	Shape Memory Extensions
Advanced Composite Structures (Integrated Cabling and Thermal)	Standard Composites Aluminum	Smart Structures
VLSI Dig. Signal Processors, Fast Packet Switches (GaAS, CMOS)	(Terrestrial Suppliers)	Optical Processing
SC-cut Crystal Oscillators	(Terrestrial Suppliers)	--
GaAs MMIC's (High Volume/Low Cost)	--	--
Active Phased Array Antennas	Gimballed GSL Arrays	--
60 GHz Intersatellite Links (Phased Arrays)	Gimballed 60 GHz ISL Arrays	Optical Intersatellite Links Superconducting 60 GHz
Autonomous On-orbit Operations S/W	Partially Autonomous S/W	Autonomous COCC S/W

Wireless Revolution in New Service Providers and New Equipment Suppliers


Manufacturers/Suppliers
Telepower Service Providers
Magazine & Newspaper Publishers
Tele-Education & Tele-Health Providers
CABLE TV Providers
TV Broadcasters
Research and Private Network Providers
Interexchange Carriers - Long Distance Telephone Providers
Local Exchange Carriers - Local Telephone Providers
"Smart" Energy Providers
Telecommunication Switch Manufacturers
Computer & Terminal Manufacturers
Consumer Appliances (e.g. TV, VCR, HDTV) Display Manufacturers
Telepower Service Providers
Manufacturers/Suppliers

Source: Dr. Joseph Pelton (CU), Feb. '94

New Frontiers in Low Earth Orbit Wireless Communications

Dr. James R. Stuart
Director, Space Segment, Teledesic Corporation
1062 W. Alder Street, Louisville, CO 80027 (303) 666-0662, fax (303) 666-0388

Conference On High Speed Computing, 1994: High Performance Systems
Session 8: New Frontiers in High Performance Systems II (10:15AM - 12:15PM)
Salishan Lodge, Gleneden Beach, Oregon, 21 April 1994

Abstract:

This presentation will summarize the new mobile and wideband capabilities of currently planned low Earth orbit (LEO) satellite communications systems and their relation to the National Information Infrastructure (NII). The current technologic and economic trends are driving inevitably to a future that includes 'Little', 'Big' and the new wide-band 'Mega' LEO's (with much larger constellations of 1000 or more satellites). For example, Teledesic will provide a space-based, wireless, digital, wide-band global communications utility. Comparisons of the currently proposed 'Little', 'Big' and 'Mega' LEO systems will be presented (e.g, characteristics, services, market projections, capacities, subscriber costs, and program development costs, etc.). Current and planned wideband geostationary (GSO) satellites (e.g., ACTS, etc.) will be also be addressed.

The features of the extremely high-performance and high-power Teledesic LEO satellites will be described (e.g., many kW's, 100's of MIPS, 1000 mps, 1000's of spot beams, etc.). Many of the subsystem components, materials and processes developed for space exploration and national defense have application directly or indirectly in Teledesic's space segment. The Teledesic satellites are a new class of small satellites, which demonstrate the important commercial benefits of using technology developed for other purposes by U.S. National Laboratories (such as JPL and LLNL). The new Teledesic satellite manufacturing, integration and test approaches use modern high volume production techniques that result in surprisingly low space segment costs. The current surge in space-based LEO wireless communications systems and architectures demonstrates the important commercial benefits being derived from using technology, components, materials and processes developed for space exploration and national defense by U.S. National Laboratories, which are now being applied to the information superhighway. The unprecedented volume of advanced components and services (e.g, launch services), required to construct and replace these large LEO constellations will be sufficient to propel certain industries to world market leadership positions, and enable a revolution in the price and performance of LEO and GSO commercial communications satellite systems that will affect us all.

Dr. James R. Stuart Biography:

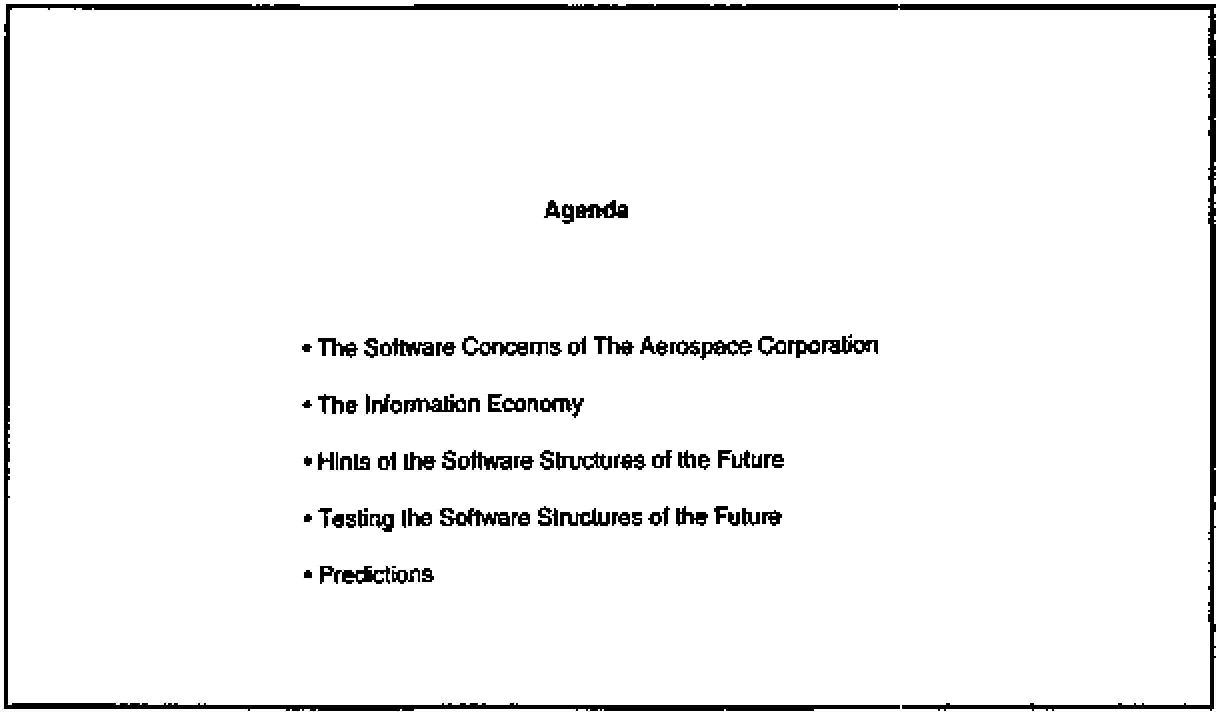
Dr. James R. Stuart is the Director, Space Segment for Teledesic Corporation (a large U.S. wide-band, wireless Ka-band LEO satellite constellation), and an independent, internationally recognized aerospace consultant specializing in advanced space systems design, development and management. He has played an important role in the creation and development of LEO and GSO communications lightsats, and is currently an active principal and board member in several entrepreneurial technology and space companies involved with communications, satellites and small launch vehicles. Dr. Stuart is, for example, concurrently Chief Technical Advisor for LEO ONE Panamericana (a Mexican store-and-forward 'Little LEO').

Dr. Stuart previously held positions as Chief Scientist and Chief Engineer at Ball Space Systems Division in Boulder, CO. He was also founding Chief Engineer of Orbital Sciences Corporation, Assistant Laboratory Director of the Laboratory for Atmospheric and Space Physics at the University of Colorado. At NASA/Jet Propulsion Laboratory he was the first Project Manager of Mars Observer, Manager of Advanced Planetary Programs among other positions. Dr. Stuart has been on various graduate faculties of the University of Colorado at Boulder for over 13 years: in the Electrical Engineering, Telecommunications and Aerospace Engineering Sciences Departments, as well as in the Center for Space Construction. He received his Ph.D. in Systems Engineering (1979), M.S. in Operations Research (1977), and M.S. in Electrical Engineering (1974) from the University of Southern California, and his B.S. in Physics (1968) from the University of Washington.

Dr. Stuart has received numerous professional awards, including NASA's Exceptional Service Medal for his project management of the Solar Mesosphere Explorer Project, JPL's highly successful, first modern small satellite project. He is also listed in Via Satellite's "Top 100 Executives in the Satellite Communications Industry". Dr. Stuart has published over 80 professional papers on the topics of small satellite systems, space technologies and communications satellite economics.

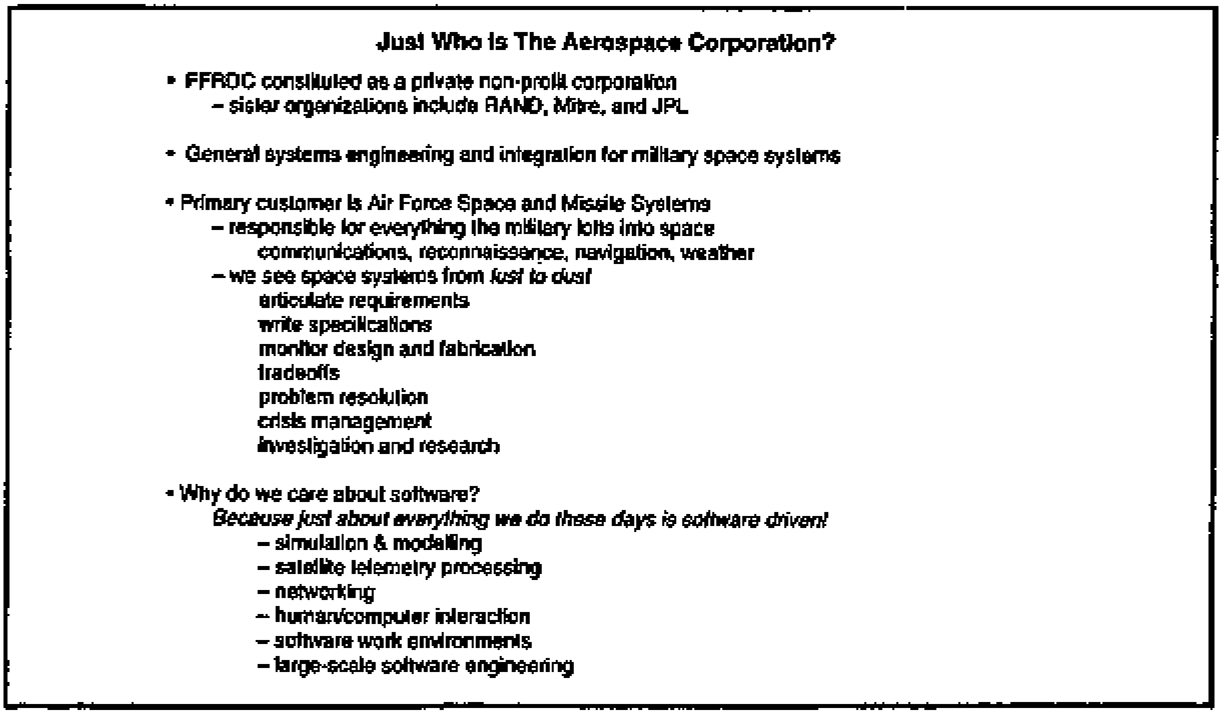
SPECULATIONS ON THE STRUCTURE OF SOFTWARE IN THE 21ST CENTURY

Michael Gorlick, Aerospace Corp.



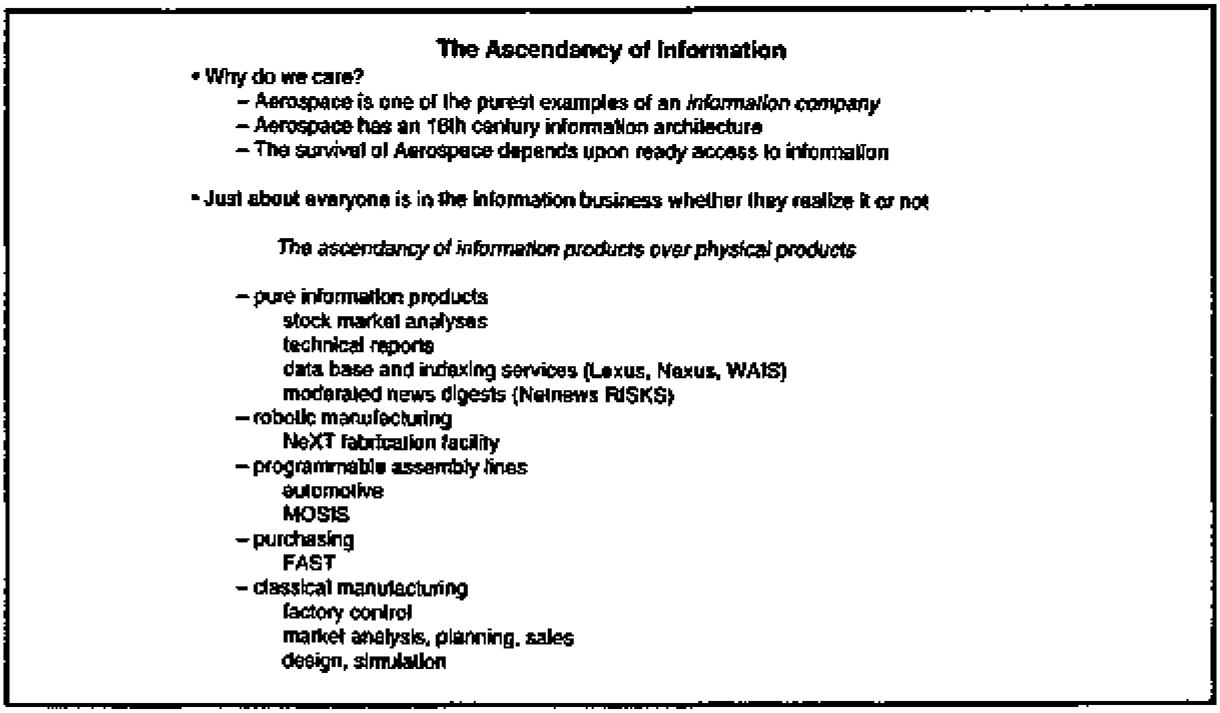
Agenda

- The Software Concerns of The Aerospace Corporation
- The Information Economy
- Hints of the Software Structures of the Future
- Testing the Software Structures of the Future
- Predictions



Just Who is The Aerospace Corporation?

- FFRDC constituted as a private non-profit corporation
  - sister organizations include RAND, Mitre, and JPL
- General systems engineering and integration for military space systems
- Primary customer is Air Force Space and Missile Systems
  - responsible for everything the military lofts into space
    communications, reconnaissance, navigation, weather
  - we see space systems from *lust to dust*
    articulate requirements
    write specifications
    monitor design and fabrication
    tradeoffs
    problem resolution
    crisis management
    investigation and research
- Why do we care about software?
  Because just about everything we do these days is software driven!
  - simulation & modelling
  - satellite telemetry processing
  - networking
  - human/computer interaction
  - software work environments
  - large-scale software engineering



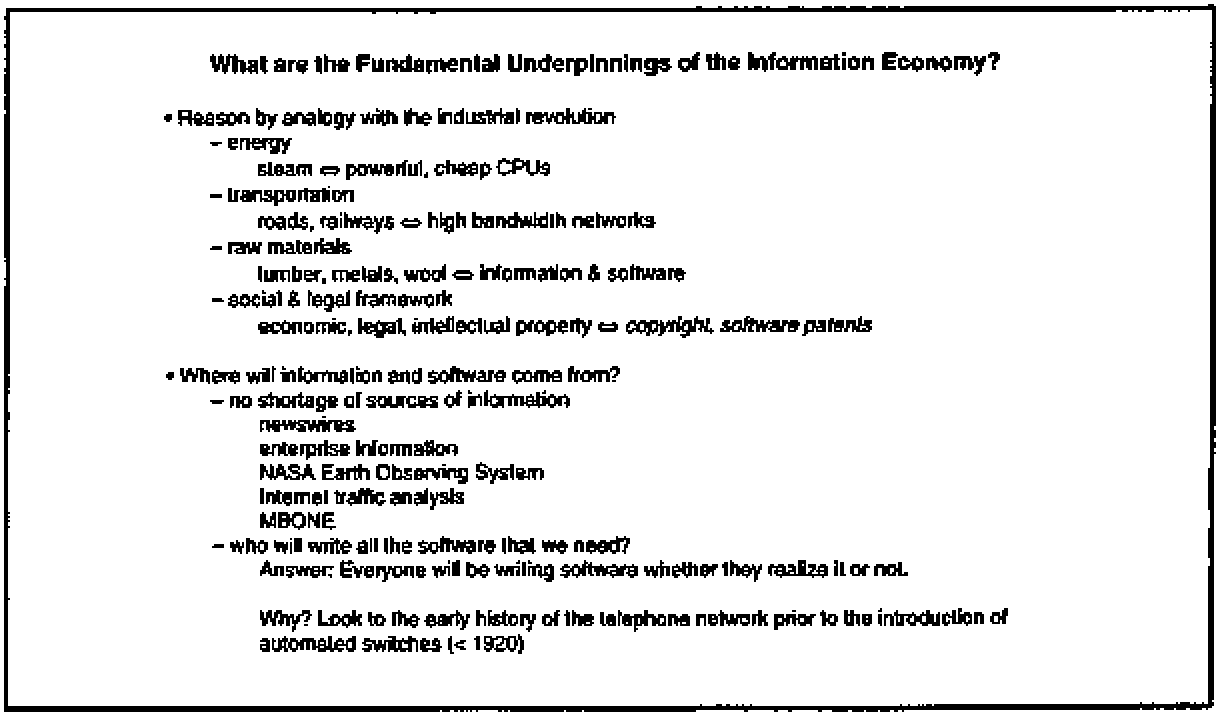
The Ascendancy of Information

- Why do we care?
  - Aerospace is one of the purest examples of an *information company*
  - Aerospace has an 18th century information architecture
  - The survival of Aerospace depends upon ready access to information

- Just about everyone is in the information business whether they realize it or not

The ascendancy of information products over physical products

- pure information products
  - stock market analyses
  - technical reports
  - data base and indexing services (Lexus, Nexus, WAIS)
  - moderated news digests (Netnews RISKS)
- robotic manufacturing
  - NeXT fabrication facility
- programmable assembly lines
  - automotive
  - MOSIS
- purchasing
  - FAST
- classical manufacturing
  - factory control
  - market analysis, planning, sales
  - design, simulation



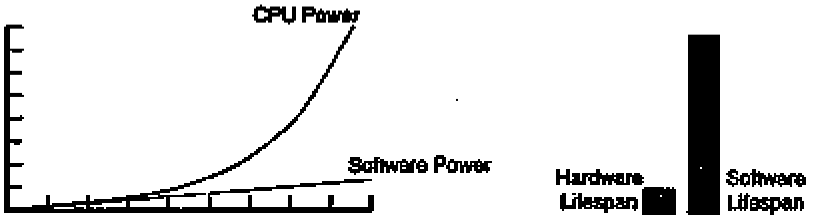
What are the Fundamental Underpinnings of the Information Economy?

- Reason by analogy with the industrial revolution
  - energy
    - steam ⇔ powerful, cheap CPUs
  - transportation
    - roads, railways ⇔ high bandwidth networks
  - raw materials
    - lumber, metals, wool ⇔ information & software
  - social & legal framework
    - economic, legal, intellectual property ⇔ *copyright, software patents*

- Where will information and software come from?
  - no shortage of sources of information
    - newswires
    - enterprise information
    - NASA Earth Observing System
    - Internet traffic analysis
    - MBONE
  - who will write all the software that we need?
    - Answer: Everyone will be writing software whether they realize it or not.

      Why? Look to the early history of the telephone network prior to the introduction of automated switches (< 1920)

The Software Capability Gap



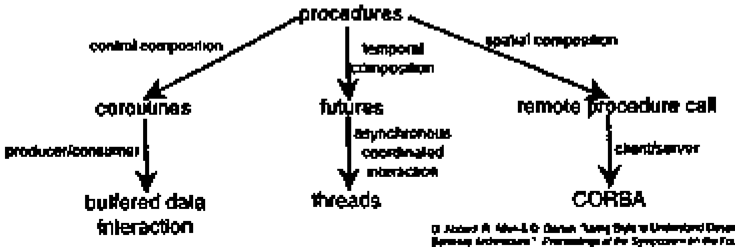
- Processors will be everywhere, all dressed up and with nowhere to go
- Societal computing will swamp everything else
- Software curve must move from linear to non-linear growth
- Shorten the software lifespan
  - – flexibility
  - – reliability
  - – multicultural versus monocultural software
    Biological monocultures are subject to catastrophic failure
    Internet worm
    PC and Macintosh viruses
    Internet-wide password collection
  - – performance
    improvements in algorithms

Flexible Software is the Holy Grail of the 21st Century

- Drivers for change
  - – revolution in computing hardware every 2 years
  - – the information economy is insatiable
    - growth in Internet traffic
    - Mosaic, Gopher, WAIS
    - Interpedia
    - MBONE
  - – information space will be highly chaotic and easily disrupted
    - hardware failures
    - service failures
    - errors
    - deliberate sabotage
- We don't know what we want until we've built it
  - – spiral model of software development
  - – rapid prototyping
    Everything is a prototype
  - – systems so complex that no one is smart enough to get them right the first time
- Resilience in everything
  - – IP/TCP dictum
    Accept anything from anyone and adhere strictly to the protocol

Compositional and Integration Services as Leverage

- The analytical maxim of computer science is *divide and conquer*
  - the synthetic maxim will be *compose and unite*
- The elaboration of architectural structures is the search for compositional operators



G. Abowd, R. Allen & D. Garlan, "Using Style to Understand Descriptions of Software Architecture," *Proceedings of the Symposium on the Foundations of Software Engineering*, Redondo Beach, California, Dec 1993

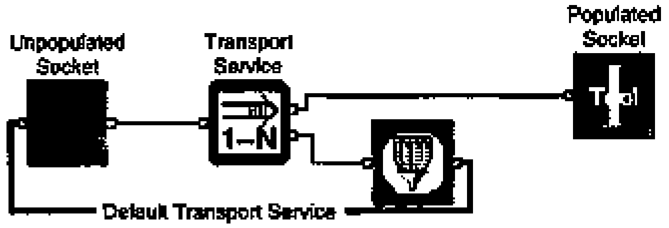
R. Allen & D. Garlan, "Formalizing Architectural Connection," Proceedings of the 16th International Conference on Software Engineering, Sorrento, Italy, May 1994

J. Purtilo "The POLYLITH Software Bus," ACM TOPLAS 16(1), January 1994

K. Sullivan & D. Notkin, "Reconciling Environment Integration and Software Evolution," ACM TOSEM, 1(3), July 1992

- The secret is in the glue
  - Unix pipes and filters
  - internals of Unix utilities
    - *it's all smoke and mirrors*
  - Internet
    - ftp (data composition)
    - telnet (control composition)
  - Mosaic
- Compositional services are the large force multipliers

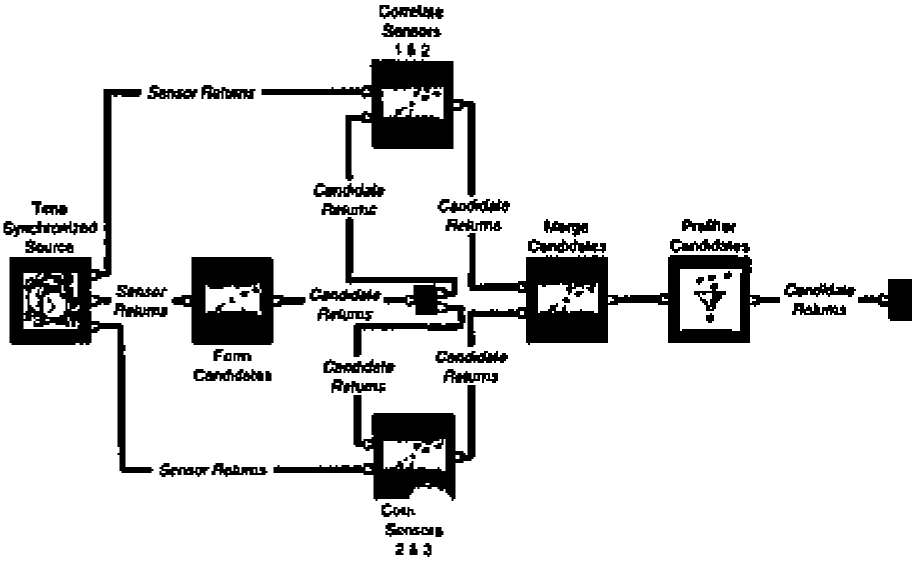
Weaves



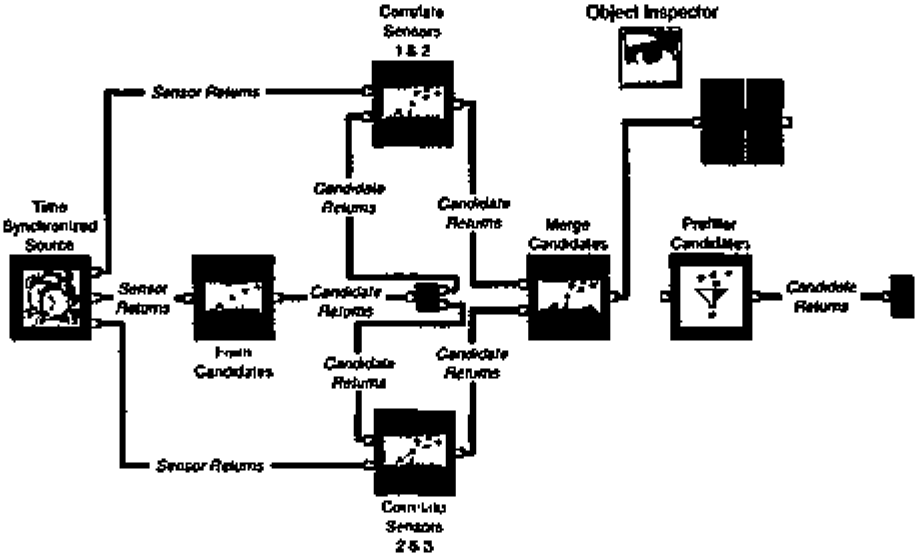
- Weaves are networks of interconnected tool fragments communicating (local or remote) by sending and receiving objects
- Computation is strictly separated from communication
  - Tool fragments are oblivious to connectivity (*blind communication*)
  - Run-time reconfiguration of the network
- Each tool fragment runs in parallel with other components
- Applications characterized by streams of data

M. Gorlick & R. Razouk, "Using Weaves for Software Construction and Analysis," *Proceedings of the 13th International Conference on Software Engineering*, Austin, Texas, May 1991

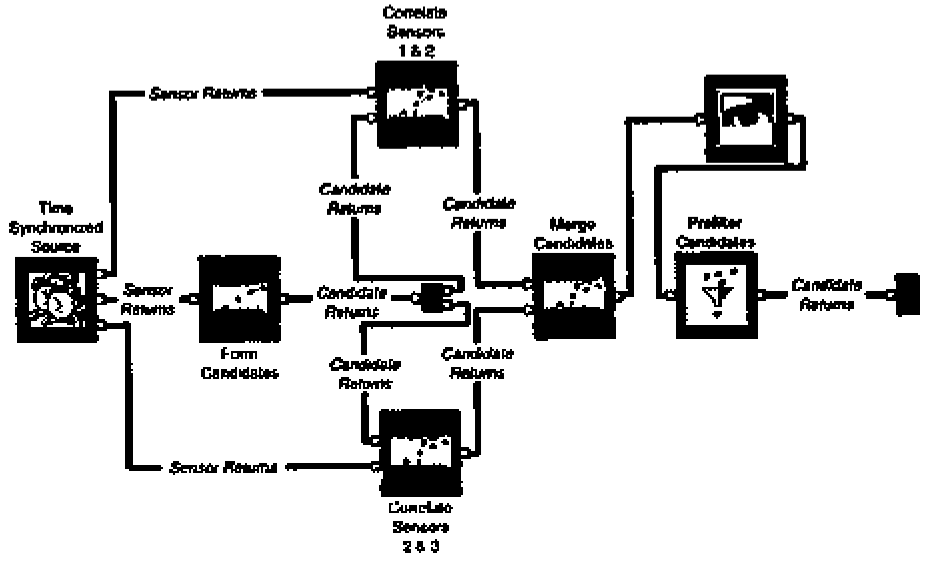
A Portion of a Weave for Stereo Tracking



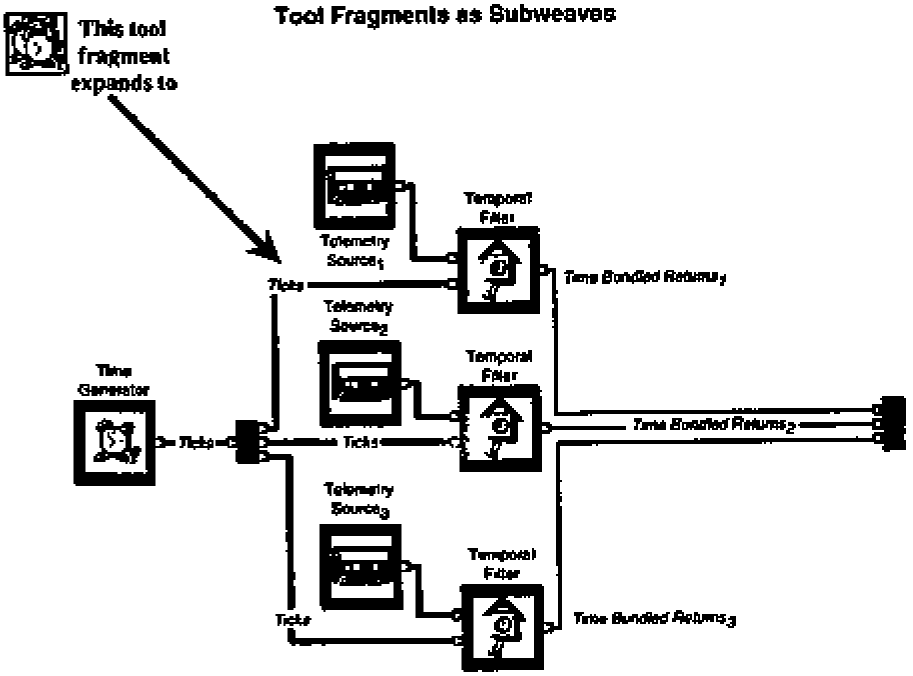
Editing the Weave to Insert Instrumentation



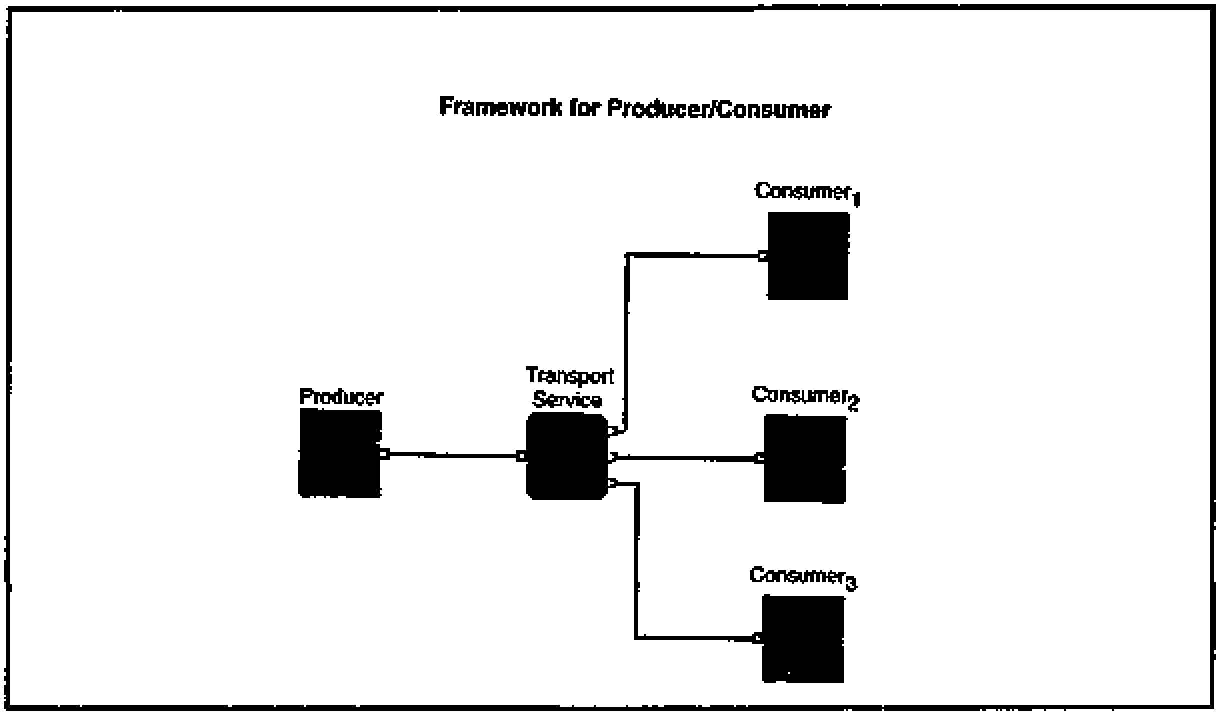
Editing the Weave to Insert Instrumentation



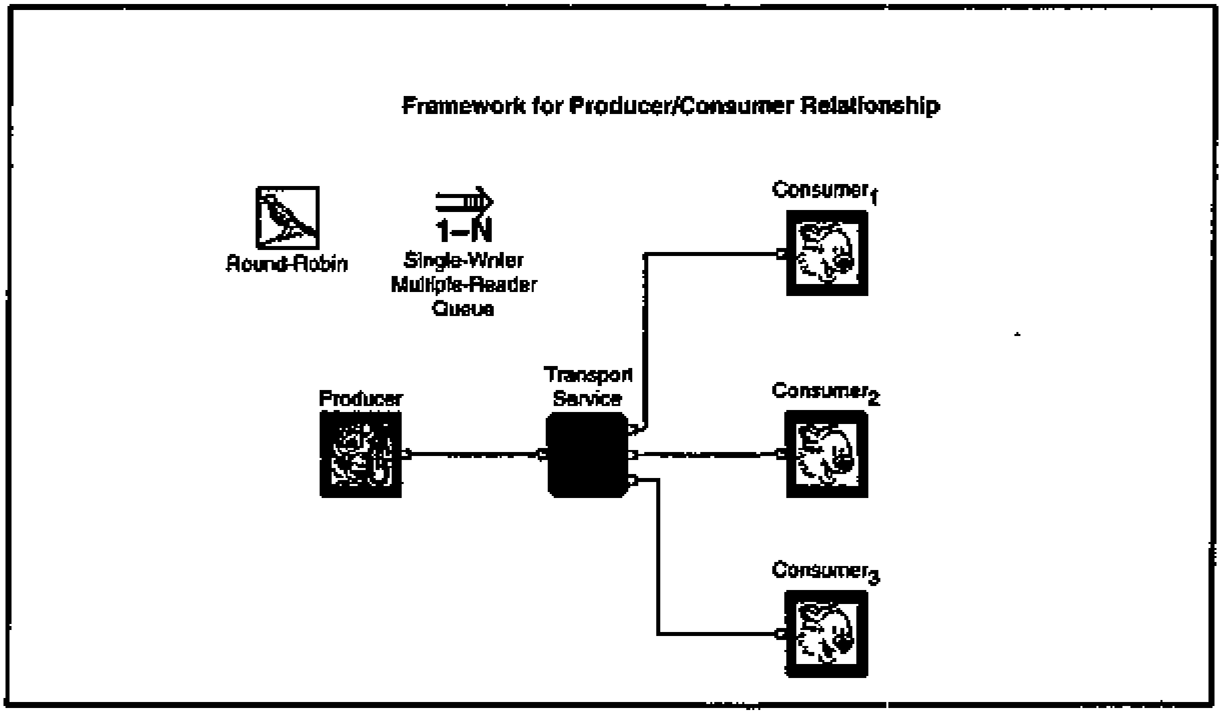
Tool Fragments as Subweaves

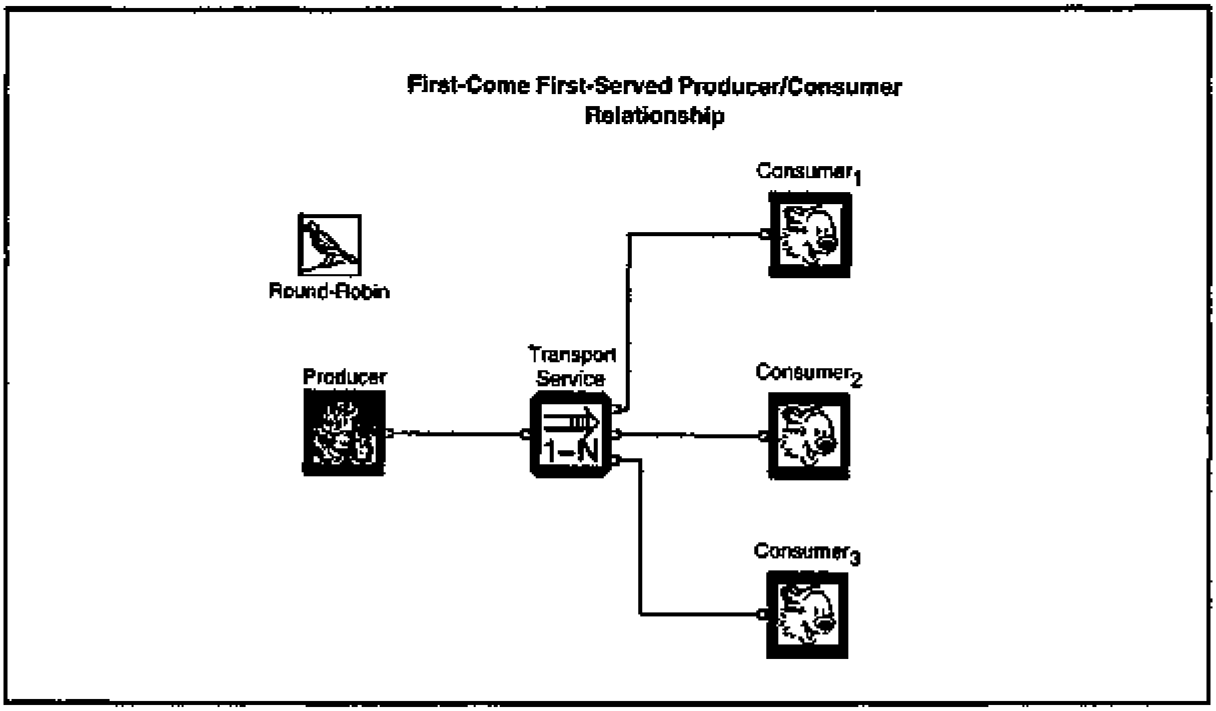


Framework for Producer/Consumer

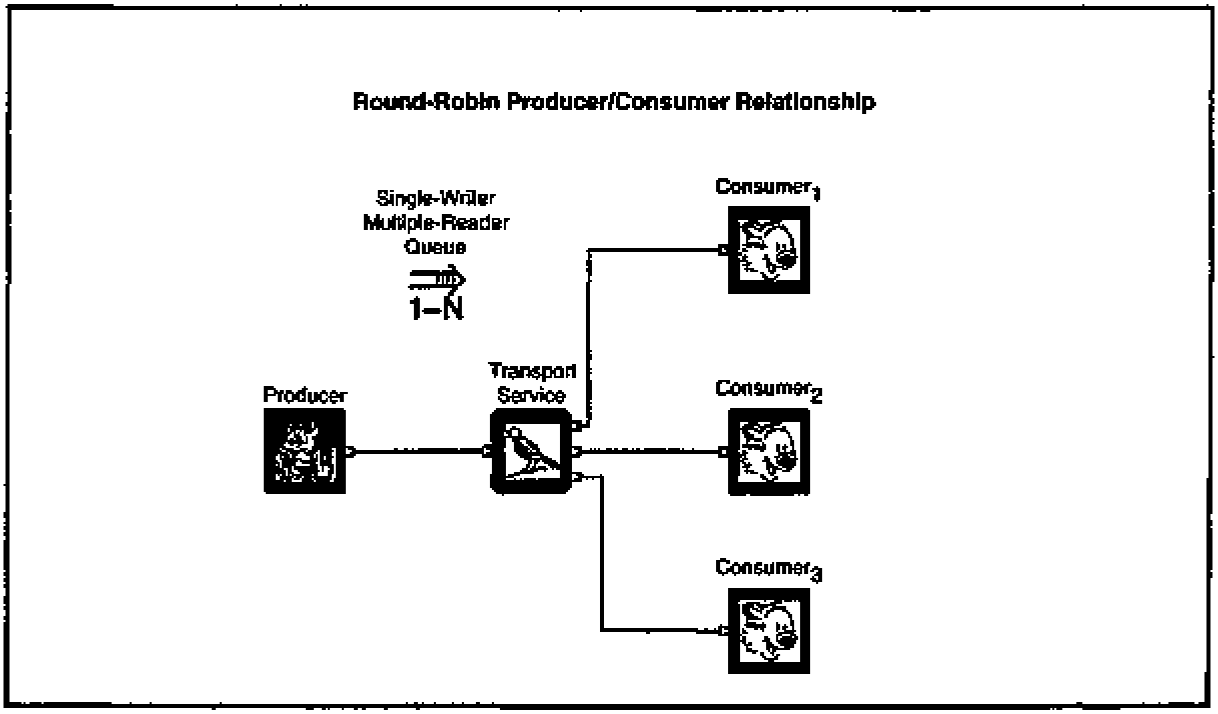


Framework for Producer/Consumer Relationship



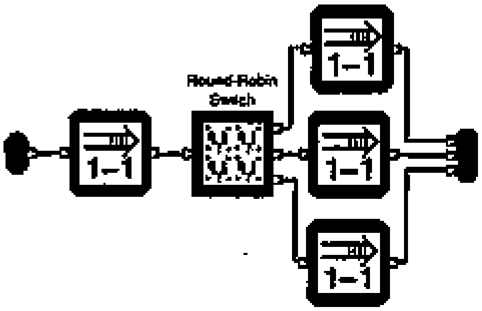


First-Come First-Served Producer/Consumer Relationship

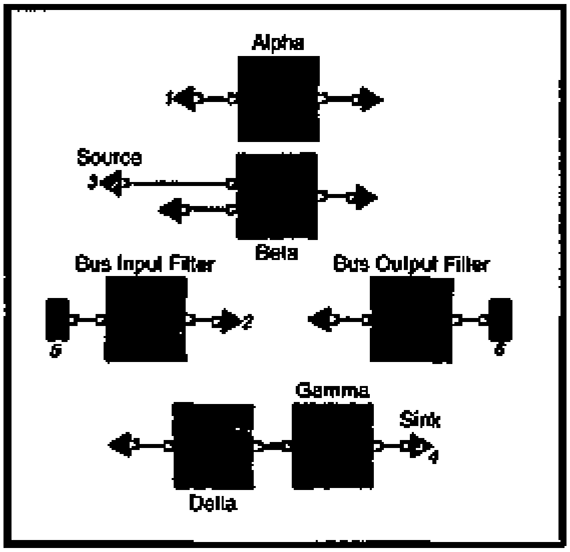


Round-Robin Producer/Consumer Relationship

Round-Robin Transport Service



Hierarchical Software Message Buses

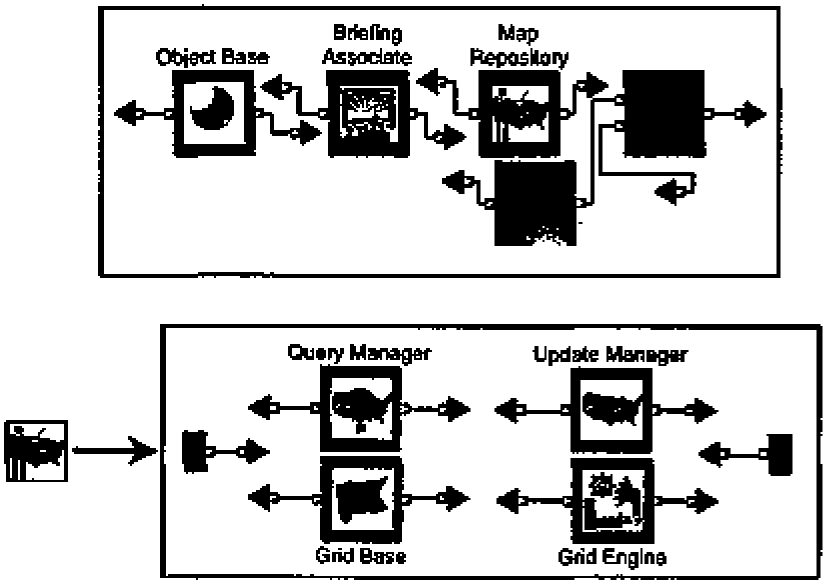


Legend
1 – On-sheet Bus Receiver
2 – On-sheet Bus Transmitter
3 – Off-sheet Bus Receiver
4 – Off-sheet Bus Transmitter
5 – Input Terminal
6 – Output Terminal

Sheet Boundary

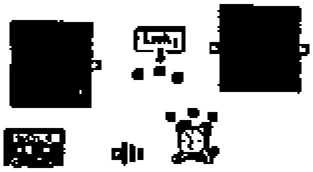
M. Gorlick "Cricket: A Domain-Based Message Bus for Tool Integration," *Proceedings of the 2nd Irvine Software Symposium*, Irvine, California, UC Irvine Research Unit in Software, March 1992

Composition of Hierarchical Software Message Buses

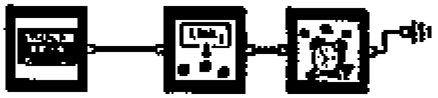


Telescript

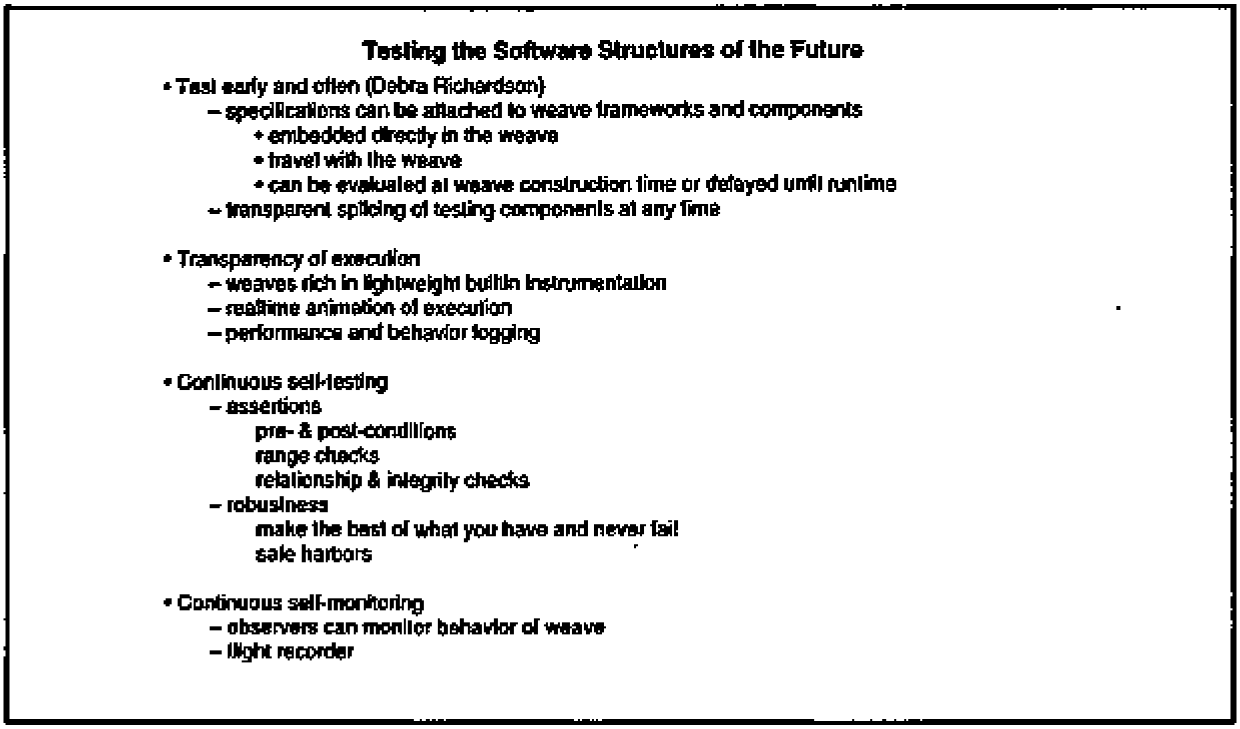
- Commercial product from General Magic
  - Telescript is to networking as Postscript is to page description
  - interpreted, object-oriented remote programming language
  - concepts
    - places
    - agents
    - travel
    - meetings
- Obverse of RPC
  don't send data send programs instead
- Different view of composition and integration from weaves
- One possible marriage of the two approaches is to use Telescript agents as *weavers*



Component Repository



Agent leaves with assembled weave



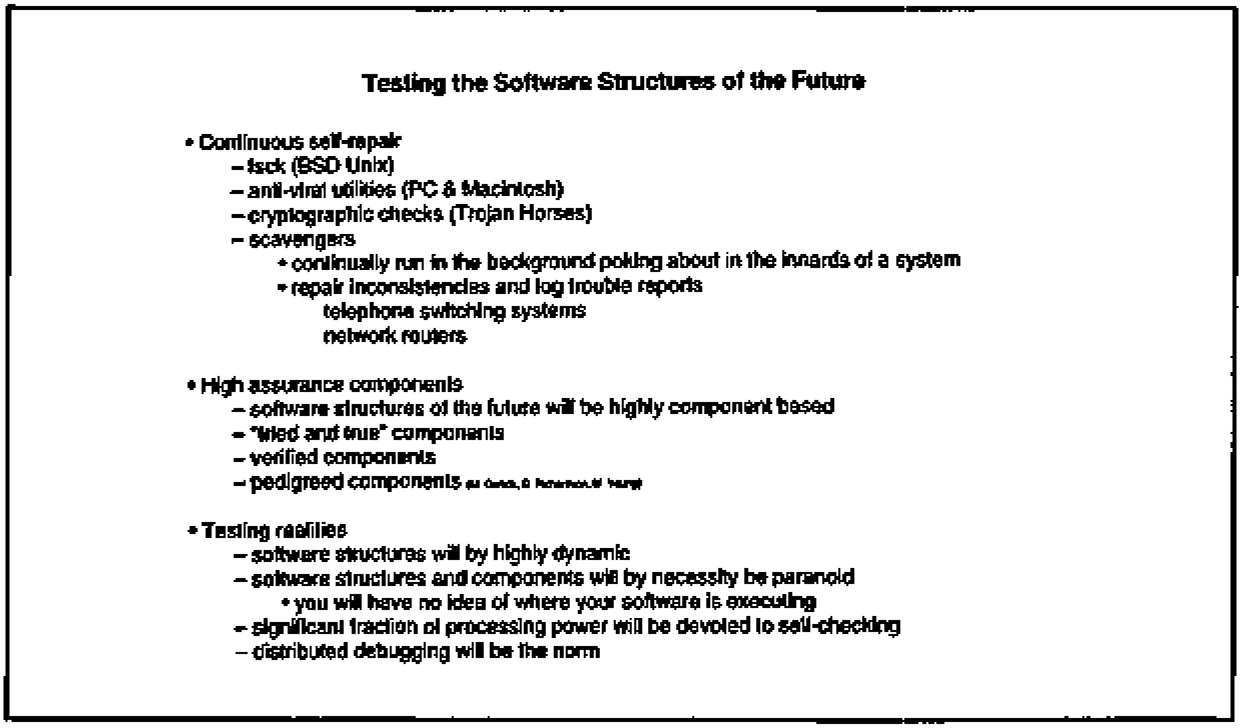
Testing the Software Structures of the Future

- Test early and often (Debra Richardson)
  - specifications can be attached to weave frameworks and components
    - embedded directly in the weave
    - travel with the weave
    - can be evaluated at weave construction time or delayed until runtime
  - transparent splicing of testing components at any time

- Transparency of execution
  - weaves rich in lightweight builtin instrumentation
  - realtime animation of execution
  - performance and behavior logging

- Continuous self-testing
  - assertions
    pre- & post-conditions
    range checks
    relationship & integrity checks
  - robustness
    make the best of what you have and never fail
    safe harbors

- Continuous self-monitoring
  - observers can monitor behavior of weave
  - flight recorder



Testing the Software Structures of the Future

- Continuous self-repair
  - fsck (BSD Unix)
  - anti-viral utilities (PC & Macintosh)
  - cryptographic checks (Trojan Horses)
  - scavengers
    - continually run in the background poking about in the innards of a system
    - repair inconsistencies and log trouble reports
      telephone switching systems
      network routers

- High assurance components
  - software structures of the future will be highly component based
  - "tried and true" components
  - verified components
  - pedigreed components [illegible]

- Testing realities
  - software structures will by highly dynamic
  - software structures and components will by necessity be paranoid
    - you will have no idea of where your software is executing
  - significant fraction of processing power will be devoted to self-checking
  - distributed debugging will be the norm

Predictions

Near-term is going to be very unpleasant

inadequate development methodologies
inadequate composition mechanisms
inadequate network technology
inadequate testing theory and mechanics

The information economy is going to steamroller everyone

We are all going to be roadkill on the information highway
Traditional software methods amount to criminal negligence

The government and marketplace will demand a new research agenda

Don't write software generate it
Don't generate software compose it
Software as a by-product of other processes
Don't do it here when you can do it there

* U.S. GOVERNMENT PRINTING OFFICE 1995-0-673-828/7083