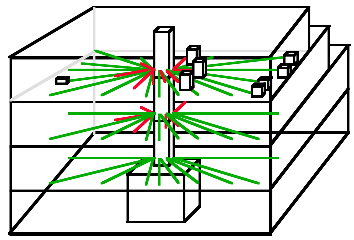
Overview of the Berkeley NOW Project



David Culler,
Tom Anderson,
Dave Patterson, et al.

Computer Science Division
University of California, Berkeley

Mosaic: http://now.cs.berkeley.edu/ Related paper: IEEE Micro, Feb. 1995.

NOW.1

Goal:

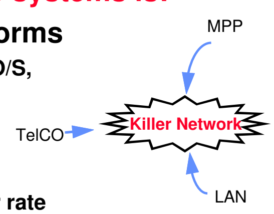
Impact the fundamental design techiques for Large Scale Systems

- **solve large problems,**
- **serve large number of users**
- **computing infrastructure of building, department, laboratory, ship, . . .**

The Basis for building large-scale systems is:

- **Complete mass-produced platforms**
  - **"killer" micro, memory system, disk, O/S,**

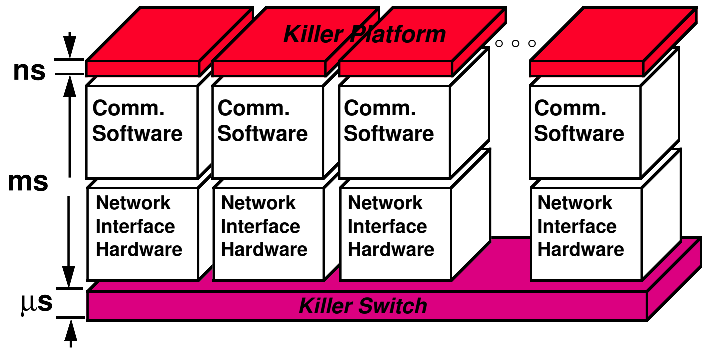
  plus SOFTWARE
  - **workstations or PCs**
- **Scalable (switched) Networks**
  - **high bandwidth, low latency, low error rate**
  - **ATM, repackaged MPP, or whoever wins**



NOW.2

–

Challenge: realizing fast communication



Goal: 10 µs end-to-end

NOW.3

Opportunity: Tremendous Resource Pool

- **500 workstations in 95-96 =**
  - 100s of GFLOPS,
  - 50-100 GB of memory (100s of MB of cache)
  - TB of disk
  - GB/s of disk bandwidth

- **Fast Communication => Remote memory is much closer than local disk**

=> New Approaches to Systems:
- file systems
- virtual memory
- parallel computing

NOW.4

Challenge: Tremendous Resource Pool

- Global Management of Resources
  - many highly interactive users
  - a few very demanding users (compute, storage, or I/O)
- Availability

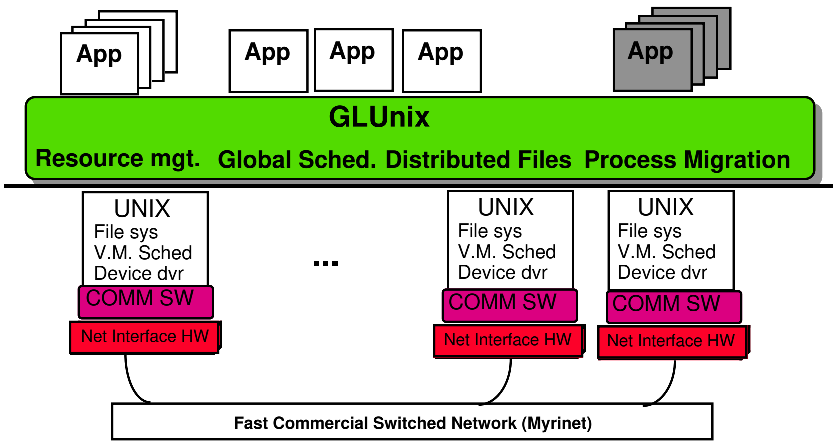
NOW.5

Project Components

- Fast Trusted-Area Communications (Comm)
- Serverless Distributed File System (I/O)
- Network Virtual Memory (Storage)
- Parallel Application Support (Proc)
- Global Resource Management (O/S)
- Benchmarking and Evaluation (###)

NOW.6

NOW System Architecture



NOW.7

Project Plan

- Initial 4-node HP 9000 735/99 Cluster
  - Medusa FDDI
- 32 Node Prototype of System components
  - Fast, Switch Based Communication
  - Glunix
  - XFS
  - Users
- >100 Node demonstration system

NOW.8

Fast Trusted Area Communication

Approach: Translate MPP experience to NOW:

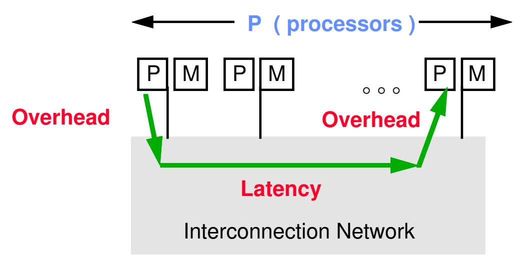
Active Messages <u>plus</u>

- **General Purpose Environment**
- **Serious Multiprogramming**
  - Protection, naming, Resource Sharing
- **Full Function Operating System**
  - page faults, daemons, . . .
- **Packet Loss**
  - infrequent (if we do our job right)

Traditional LAN: ~ 1,000 µs 1-way

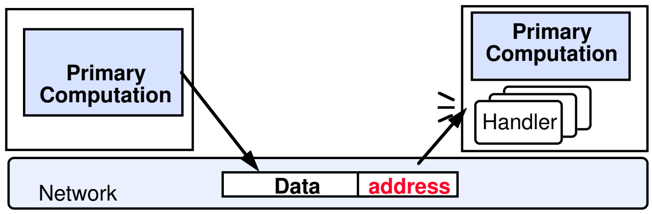
NOW Goal: 10 µs 1-way

Communication Cost (beyond BW)



- **Network Latency incurred in sending message between nodes (1-way)**
- **Processor Overhead to send *or receive* a message**

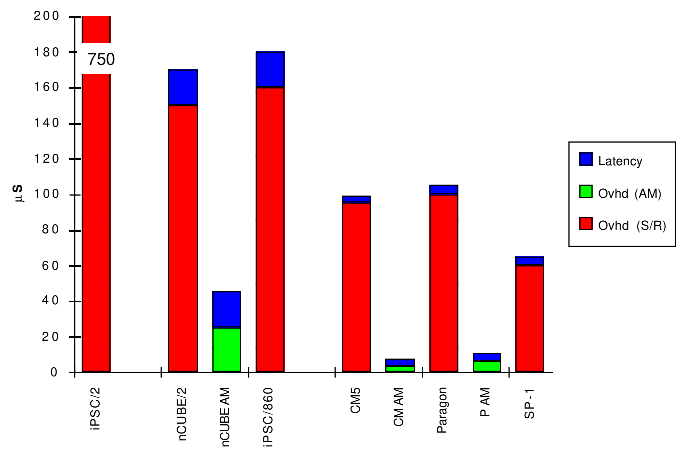
Active Messages



- **User generates msg => user knows how to handle it**
  - **Handler executes directly upon arrival**
  - **pulls message out of the network layer and integrates it into the computation or replies**
- **Simple protection, scheduling, allocation, liveness, . . .**

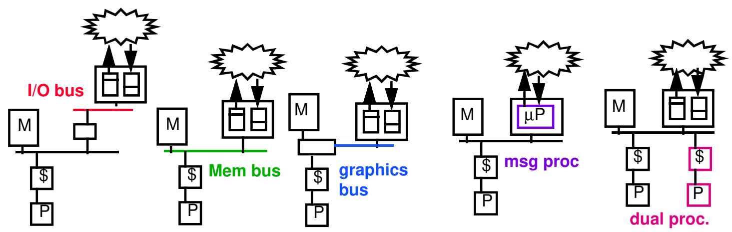
von Eicken, Culler, Goldstein, Schauser. “Active Messages: A Mechanism for Integrated Communication and Computation”, ISCA 1992.

Active Messages on MPPs



Rich Design Space

- Network Interface HW Design Spectrum

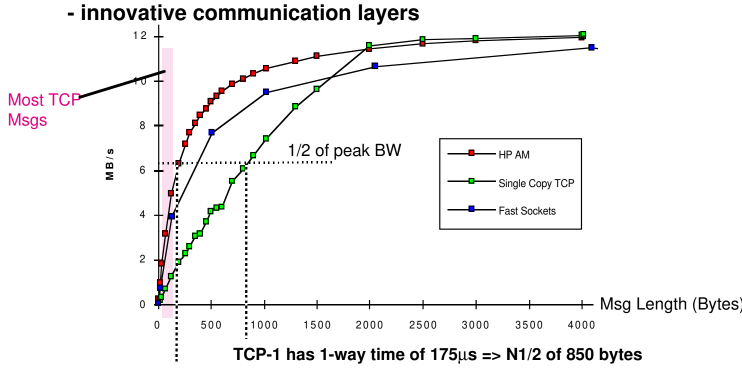


- Communication “Stack”
  - **simple protocols for “nice” networks**
  - **user level “virtual network”**

NOW.13

Prototype Comm. Performance

4 HP 9000/735 99MHz running HP UX
- experimental Medusa FDDI net. int. on the graphics bus
- innovative communication layers



TCP-1 has 1-way time of 175µs => N1/2 of 850 bytes
HP A.M. has 1-way time of 15µs => N1/2 of 175 bytes

Martin. “HPAM: An Active Message Layer for Networks of HP Workstations”, Hot Interconnects 1994.

NOW.14

Why it works

- Careful engineering of a simple abstraction
- Closer binding between network and user
  - external scheduler handles resource sharing
- Statefull NI Card + Request/Response permits very simple approach to
  - flow-control
  - time-out and retry
  -
  -
- Build traditional abstractions on top
  - MPI, Sockets, . . .

NOW.15

Status: Fast Communication

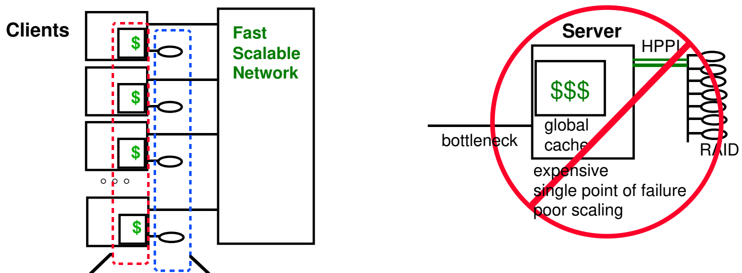
- Defined and implemented Generic Active Messages (GAM 1.0) on spectrum of NIs

Machine	NI Arch	Overhead+Latency	Bandwidth
Paragon	SMP	5.1 + 5.6 = 10.7 µs	140 MB/s
Meiko	CoP	1.1 + 11.4 = 12.5 µs	39 MB/s
Myrinet	*CoP*	*< 20 µs*	*>20 MB/s*
HP/Medusa	Graph	10.1 + 10 = 14.5 µs	12 MB/s *
SS10/SAHI	Ctrlr	11 + 200 = 211 µs	12 MB/s

- Communication models built on top
  - shared address, sockets, client/server
- Analysis of ATM options
- Ongoing
  - Broader, more complete A.M. Specification
  - HP GSC+/Myrinet Card
  - NIC hardware design

NOW.16

XFS: NOW Serverless File System



Software RAID using LFS (RAWD)
- **link BW determines striping**

Huge Cooperative File Cache
- **private and global**
- **dynamic location (COMA + fault tolerance)**
- **latency critical**

NOW.17

XFS Performance

Time to transfer 8KB block:

	Remote Mem	Remote Disk
Ethernet	6.9 ms	21.9 ms
ATM (155Mb/s)	1.1 ms	16.1 ms

Simulation results: (Sprite file trace)

	Miss Rate	Response Time
Client/Server	16%	2.8 ms
Coop Caching	8%	1.6 ms

(42 WS, 16 MB/WS, 128 MB/server)

Dahlin, Wang, Anderson, and Patterson, “Cooperative Caching: Using Remote Client Memory to Improve File System Performance”, OSDI 1994.

NOW.18

Why it works

- Disk access is the slowest part of the system
- Small increase in local miss rate
  - => More communication (but with less contention)
- Larger decrease in global miss rate
  - => Fewer disk accesses
  -
  -
- Selective (almost greedy) replacement
  - inactive workstations become more “server-like”
  - active workstations hold onto what they use

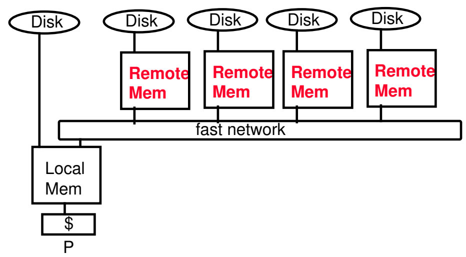
NOW.19

Status: XFS

- High Level Design Complete
- Detailed simulations
  - 80% improvement in response time
- Operational Feasibility Prototype
- Implemented
  - workstation disk striping
  - co-operative caching
  - single-sever data failure recovery
- On-going
  - location data failure recovery
  - distributed managment of location data and meta data

NOW.20

Network Virtual Memory



$$4\text{ ms} + \frac{4\text{ KB}}{2\text{ MB/s}} = 6{,}000\ \mu s$$

$$50\mu s + \frac{4\text{ KB}}{20\text{ MB/s}} = 250\ \mu s$$

- **Potential 25x improvement in miss cost**
- **Two Prototype nRAM distributed server**
  - **external user-pager for Solaris and user-level library**
  - **Prototype user level pager: + 500 µs**
  - **Current VM systems: + 500 µs => limit of 5x in miss cost**
  - **Invaluable insight toward communication abstractions**

NOW.21

Parallel Application Support

Performance of a Climate Model

Machine (Procs)	***ODE***	***Transport***	***I/O ****	***Total (s)***
Cray C-90 (16)	**7**	**4**	**25**	**38**
Intel Paragon (256)	**12**	**24**	**10**	**46**
NOW:				
RS/6000 (256) + PVM / Ether	**4**	**23,340**	**4,030**	**27,374**
+ ATM bandwidth (20 MB/s)	**4**	<u>192</u>	<u>2,015</u>	**2,211**
+ Parallel FS	**4**	**192**	<u>10</u>	**206**
+ low overhead (10 µs)	**4**	<u>11</u>	**10**	**25**

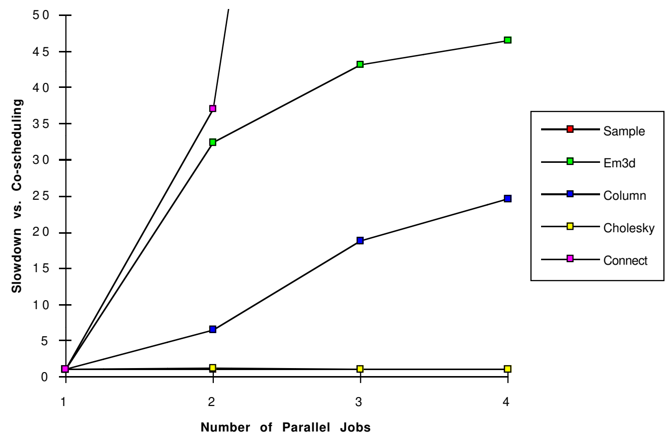
Order of Importance

*(1 disk / processor, parallel FS for C-90, Paragon)

Demmel and Smith, "Parallelizing a Global Atmospheric Chemical Tracer Model", SHPCC 1994.

NOW.22

Sensitivity to Scheduling



Parallel Pgms like to Run in Parallel

NOW.23

Status: Parallel Application Support

- **Basic Message Passing (PVM, ...)**
- **Shared Address Space (Split-C)**
- **Debugger (Mantis)**
- **Direct simulations (on CM-5) of program behavior under possible NOW schedulers**

on-going:

- **MPI on active messages**

NOW.24

The Design of a Global O/S

- How do you build it?
  - Out of off-the-shelf commercial components
    - » leverage huge investment / volume, reduce lag time, . . .
    - » technology transfer
  - Enabling technology
    - » Software fault isolation (SFI)
    - » Proxy
- What should it do?
  - Provide each active user with the interactive performance of one dedicated workstation
  - Provide a small number of users a large fraction of resources
    - » sequential or parallel
  - Be Available

NOW.25

Software Fault Isolation

- Traditional approach
  - use hardware to enforce protection between different address spaces
- SFI: enforce fault isolation within single address space
  - Insert language-independent software checks into object code (stores and indirect branches), optimize
  - Overhead for SPEC is 7% on MIPS, 3% on Alpha, 2% on PowerPC
- Enables "fuzzing" of kernel-user boundary
  - Run protected kernel code in application libraries
- 1.1X to 1.3X slower v. all DRAM:
  => Virtual Memory is reborn
- (4X to 9X faster v. DRAM+disk)

Wahbe, Lucco, Anderson, Graham. "Efficient Software-Based Fault Isolation", SOSP 1993.

NOW.26

Preserving Interactive Performance

- Pay attention to the <u>state of the storage hierarchy</u>
  - Dedicated workstation is aligned to "user's working set"
    - » You notice intrusions that cause page faults
  - Primary metric is the cost of restoration
    - » CPU is almost free
    - » Memory footprint is critical
  - Desktop: give back what you borrow
  - Basement: create affinity or lose
- Use Fast network and Parallel File System
  - Cost of restore
    - » 64MB over Ethernet, single disk 60 seconds
    - » 64MB over ATM, parallel file sys 4 seconds

NOW.27

Demanding Applications

- Sequential Applications
  - network RAM
  - file system cooperative caching, disk striping
- Parallel Applications
  - co-scheduling
  - fast communication
  - parallel file systems
- Availability
  - engineering for fault isolation (not shared memory)
  - Hot Swap of HW and SW components
  - automated system management

NOW.28

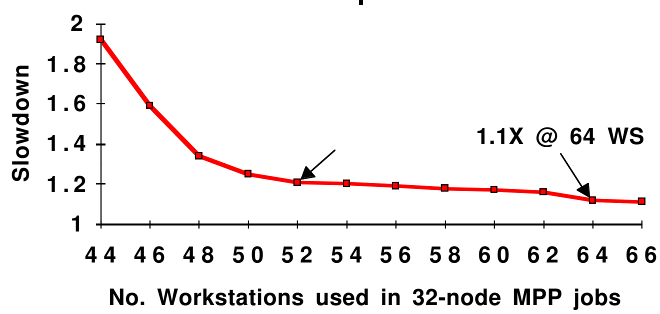
Desktop vs. Machine Room ?

- **BOTH !!!**
- **Desktop**
  - Tremendous idle capacity, even at coarse grain
  - Very few unexpected reboots
  - Institutional infrastructure will support it
  - Want fast communication to my display
- **Machine room**
  - Simpler adminstration
  - Electrical vs optical
  - Always want some headroom in systems
  - Want baseline of resources for production
- **Economics and workload will determine mix**

NOW.29

Sequential & Parallel on One System

- **Sequential has priority**
- **Ratio MPP nodes:desktops 3:5=>1.2x slowdown**



Arpaci, Dusseau, Vahdat, Lok, Anderson, Patterson. "The Interaction of Parallel and Sequential Workstations on a NOW," submitted.

NOW.30

Status: GLUNIX

- Prototype operational:
- Distributed Job Control
- Idle resource detection
- User-level co-scheduling
-
- On-going
- Migration
- Hot Swap

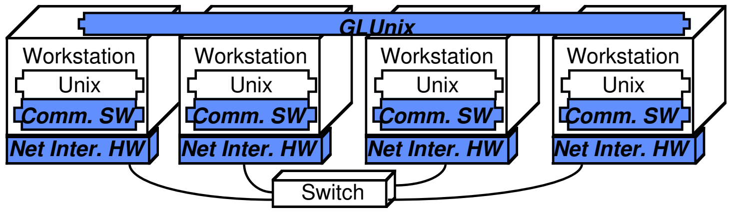
NOW.31

Status: Benchmarking

- Template for “NOW marks”
  - beyond scientific computing
- Conduits to real users
- Bringing up data base
  - Informix

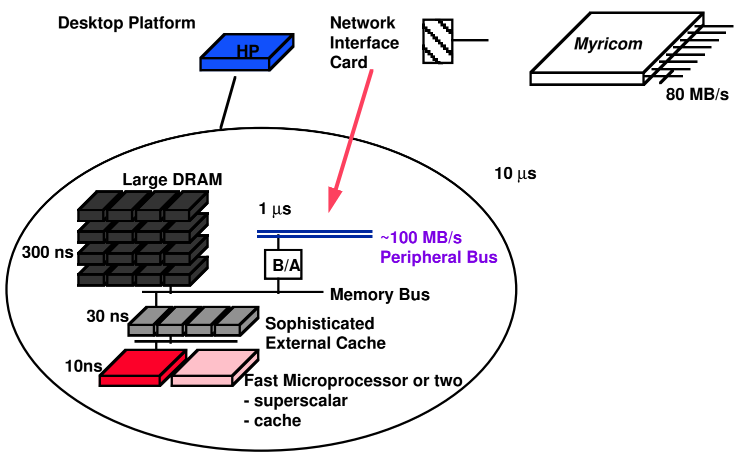
NOW.32

NOW @ Berkeley



- **Design & Implementation of higher-level system**
  - **Fast communication (low overhead SW / HW for Active Messages)**
  - **Global OS (Glunix)**
  - **Serverless Network File Systems (xFS)**
- **Goal: Better for Sequential and Parallel Appln's**
  - **recruit DRAM, disks, and CPUs**
  - **provide cheap, high availability**
  - **simplify system management**

Building Blocks



HP Options for Final System

- **100 HP PA 700J @ 100MHz**
  - 14,000 = 100 x 140 SpecInt92 22,000 = 100 x 220 SpecFP92
  - 25,000 = 100 x 250 TPS
  - 50 MB Cache, 3.2 GB DRAM, 100 GB Disk, 400 MB/s SCSI
  - 230 cu ft, 8,000 Lbs, 100 KW
  - proprietary graphics bus
  - HPUX
- **1,000 HP Pentium Vectra @ 90MHz**
  - 100,000 = 1000 x 100 SpecInt92 80,000 = 1000 x 80 SpecFP92
  - 100,000 = 1000 x 100 TPS ???
  - 256 MB Cache, 8 GB DRAM, 540 GB Disk, 4000 MB/s SCSI
  - 1090 cu ft, 24,800 Lbs, 45 KW
  - PCI
  - Linux

NOW.35

1000 PCs !!!