A Case for Networks of Workstations (NOW)

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

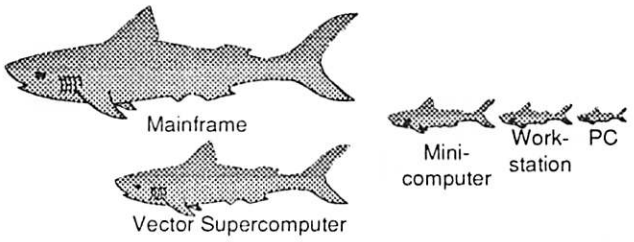
Computer Science Division
EECS Department
University of California, Berkeley

NOW 1

Original Food Chain Picture

NOW 2

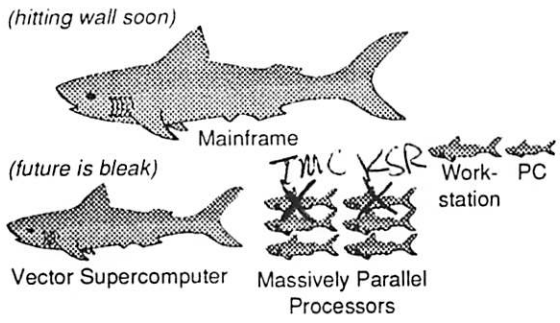
1984 Computer Food Chain



NOW 3

Minicomputer

1994 Computer Food Chain



NOW 4

MPP: A Near Miss

- near commodity procs,DRAMs, boards => delayed shipment:

	MPP	*Proc*	*Year*	*=WS*
≅	T3D	150 MHz Alpha	≅93/ 94	≅92/ 93
≅	Paragon	50 MHz i860	≅92/ 93	≅91
≅	CM-5	32 MHz SS-2	≅91/ 92	≅89/ 90

- proc perf. improves 50% / yr (4%/month)
  - 1 year lag:WS = 1.50 MPP node perf.
  - 2 year lag:WS = 2.25 MPP node perf.
- No economy of scale in 100s => +$
- SW incompatibility (OS & apps) => +$$$$

Volume vs. Cost/Price

- Rule of thumb on applying learning curve to Manufacturing:
- *When volume doubles, costs reduce 10%*

  A DEC View of Computer Engineering by C. G. Bell, J. C. Mudge, and J. E. McNamara, Digital Press, Bedford, MA., 1978.

- => Costs should be 1/2 when volume is 64X

e.g., 64X = 2^6 => 90% x 90% x 90% x 90% x 90% x 90% = 0.53 x Cost

- MPPs more expensive to develop than WS or networks yet amortized over fewer machines => Price multiplier over costs >> for MPPs to recover high development costs over few machines
- Moreover, Software costs >> Hardware costs

1990s Building Blocks

There is no near commodity component

Building block = complete computers (HW & SW) shipped in 100,000s: Killer micro, Killer DRAM, Killer disk, Killer OS, Killer packaging, Killer investment

- ≅ Leverage billion $ per year investment

Interconnecting Building Blocks => Killer Net

- ≅ High Bandwidth
- ≅ Low latency
- ≅ Reliable
- ≅ Commodity
- (ATM?)

NOW 7

Why NOW now? (Beyond technology and cost)

Building block is big enough (v. Intel 8086)

Networks are faster

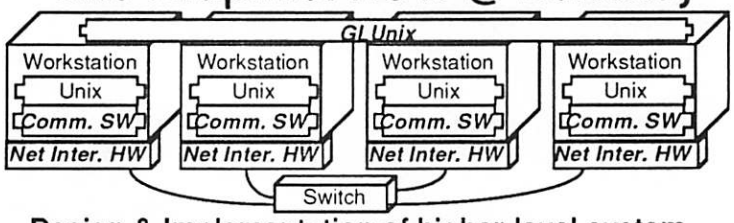
- ≅ Higher link bandwidth (v. 10 Mbit Ethernet)
- ≅ Switch based networks coming (ATM)
- ≅ New simple & fast paradigm for net. interfaces (Active Msgs)

Striped files preferred (RAID)

Demise of mainframes, supercomputers, & MPPs

NOW 8

One Exapmle: NOW @ Berkeley



Design & Implementation of higher-level system

- ≡ Global OS (Glunix)
- ≡ Parallel File Systems (xFS)
- ≡ Fast Communication (Low overhead HW for Active Messages)

NOW Goal: Faster for Sequential (and Parallel)

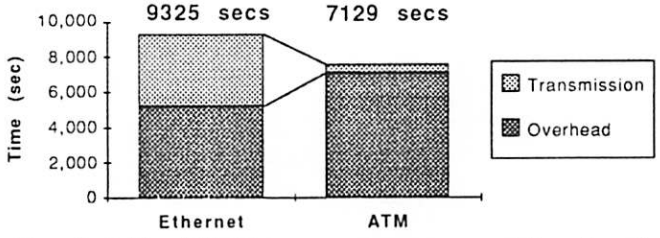
NOW 9

Importance of Overhead (and Latency)

NFS trace over 1 week: 95% msgs < 200 bytes

Ethernet: 9 Mb/s BW, 456 secs overhead

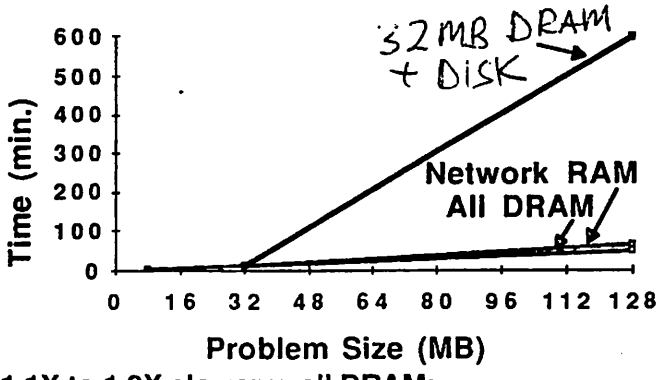
ATM Synoptics: 78 Mbit/s BW, 626 secs ovhd.



Bandwidth MIPS for processors; misleading?
(625 sec overhead ATM vs. 155 Mb/s BW ATM)

NOW 10

Network RAM simulation



- 1.1X to 1.3X slower v. all DRAM:
  => Virtual Memory is reborn
  (4X to 9X faster v. DRAM+disk)

NOW 11

3 Paths for Applications on NOW?

- Revolutionary (MPP Style): write new programs from scratch using MPP languages, compilers, libraries, ...
- Porting: port programs from mainframes, supercomputers, MPPs, ...
- Evolutionary: take sequential program & use

increasing programming difficulty ↓

1) Network RAM: first use memory of many computers to reduce disk accesses; if not fast enough, then:
2) Parallel I/O: use many disks in parallel for accesses not in file cache; if not fast enough, then:
3) Parallel program: change program until it uses enough processors that it is fast

=> Large speedup without fine grain parallel program
=> Enabling new parallel programs without MPP cost

NOW 12

Pitfalls for NOWs

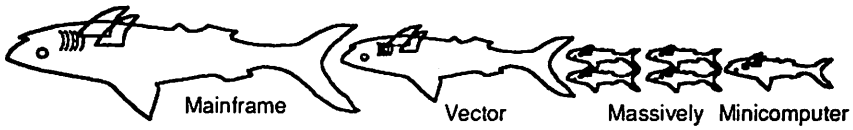
- Design NOW to only help parallel programs
  - Few applications are parallel => hard to justify fast NOW
  - Many large programs just need memory and disk BW
- Rewrite/Modify WS operating system for NOW
  - New HW waits for OS port => lower performance
  - Leverage of OS improvements by vendors?
- Custom Network Interface per WS for NOWs
  - New HW waits for NI => lower performance
  - OK for now research; NOW success => commercially available, low NI as soon as next WS announced
- Serial file system
  - can't take advantage of 100s of parallel disks

Research Focus at New Level

- "Higher Order" Systems Research: building on top of other systems vs. bottom-up
  - Must avoid time lag: neither HW nor OS can delay putting new machines to use
  - NOW is one example
- Advantages:
  - + easier to track technological advances
  - + less development time
  - + easier to transfer technology (reduce lag)
- New challenges:
  - maintaining performance goals
  - system is changing underneath you
  - underlying system has other people's bugs
  - underlying system is poorly documented

Conclusion

- Technologies aligned to exploit NOW now
  - 32-bit μprocessors, switch based LANs, active messages, striped files, file caches, process migration
- 1990s building block is desktop HW&SW
  - Need higher-level system research use building blocks: stand on shoulders, not toes
  - Challenges for NOW: Leveraging technology yet add low overhead user communication, global OS, parallel file system
- Comparative Value of Metrics: Network Bandwidth = MPP MFLOPS?
- Goal: Make sequential users happy NOW



2004 Computer Food Chain