Parallel Computing at UC Berkeley

Prof. David E. Culler
Computer Science Division

- Castle Project
- Parallel Computing Curriculum
- Distributed Supercomputing on a N.O.W.

© D.E.Culler 93

Software Support for Mainstream Parallel Computing

Faculty:

David Culler

James Demmel

Katherine Yelick

Thomas Anderson

Phillip Collela

Jerome Feldman

Susan Graham

Paul Hilfinger

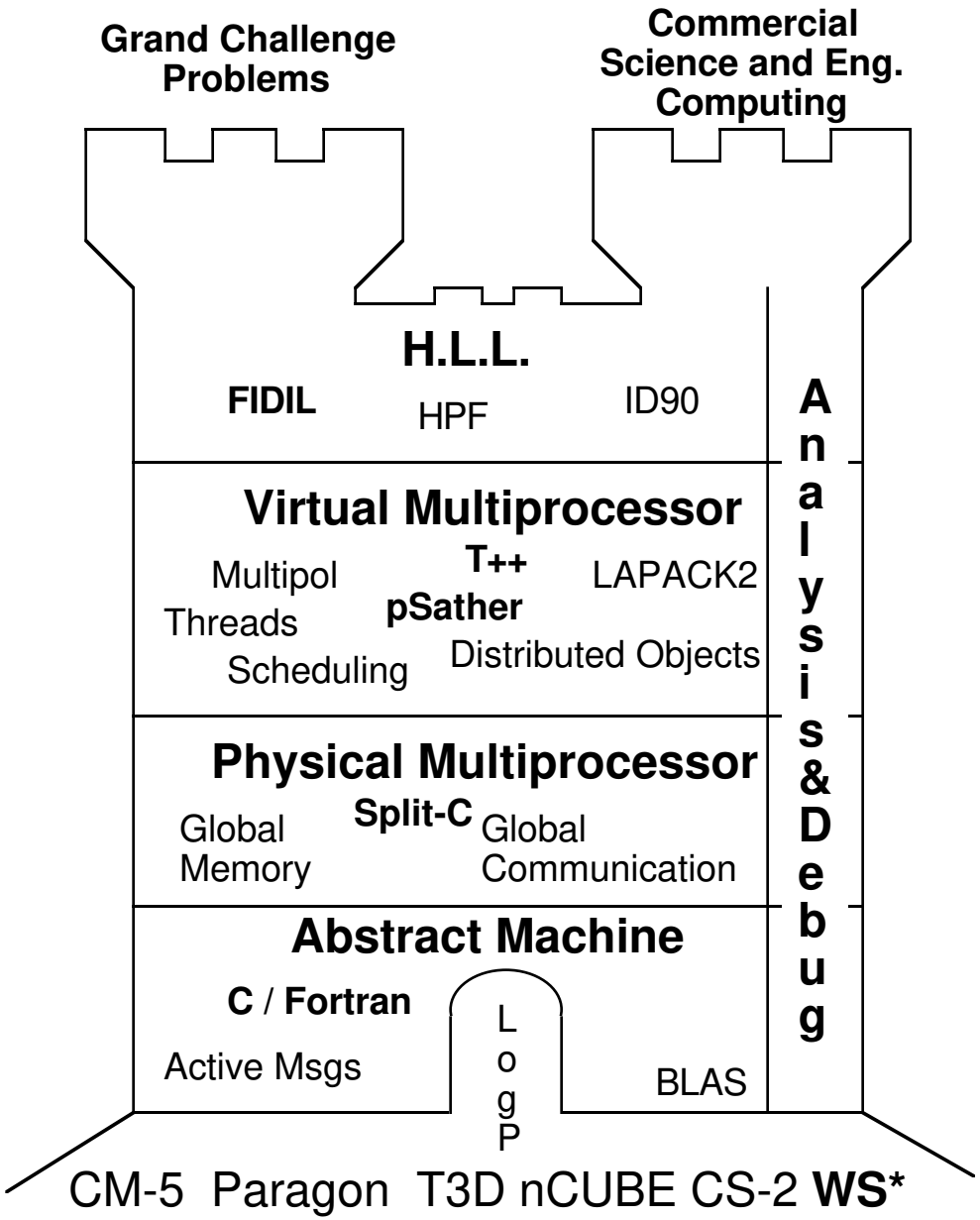
Richard Karp

Abhiram Ranade

David Patterson

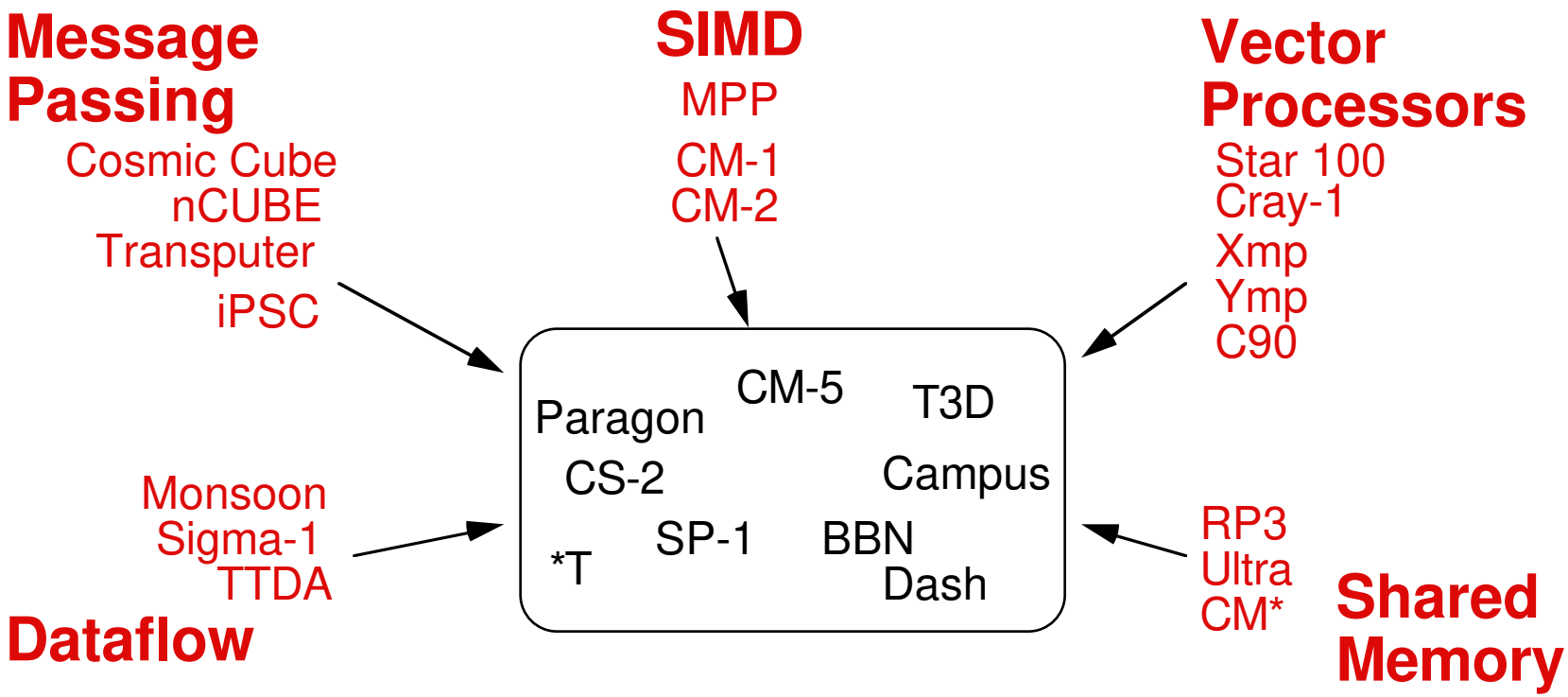
Alberto Sangiovanni-Vincentelli

James Sethian



© D.E.Culler 93

Supercomputer Convergence



- MPP node is the core of a workstation plus a dedicated network interface
- Supports all models via point-to-point messages

© D.E.Culler 93

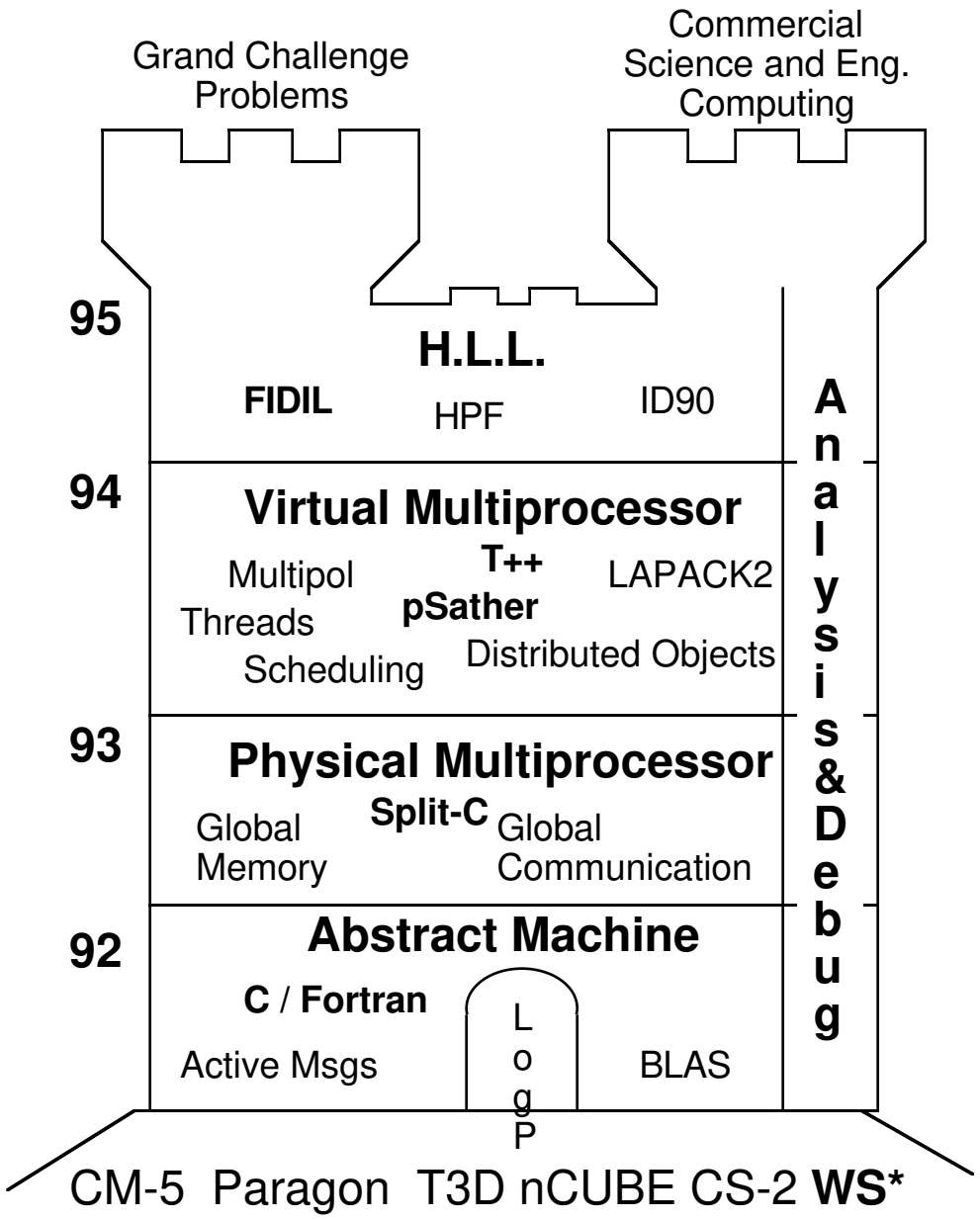
Time Critical

- **Grand Challenge PR has been successful**
- **Companies are positioned to start advanced product R&D**
- **Vendors are looking for standards**

=> need to deliver in short order

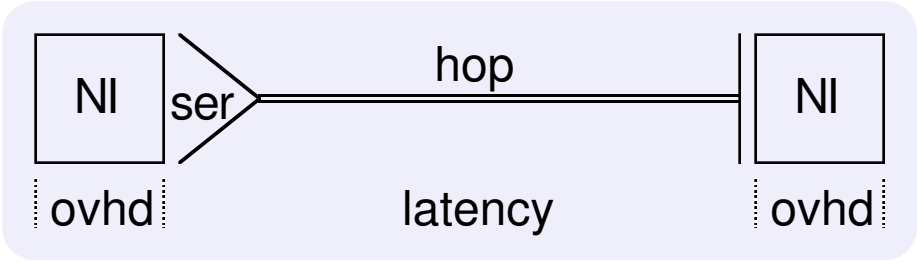
=> cannot wait for "breakthroughs "in architecture, compilation, or language design

=> need to build from current investment.



© D.E.Culler 93

Msg Passing Performance



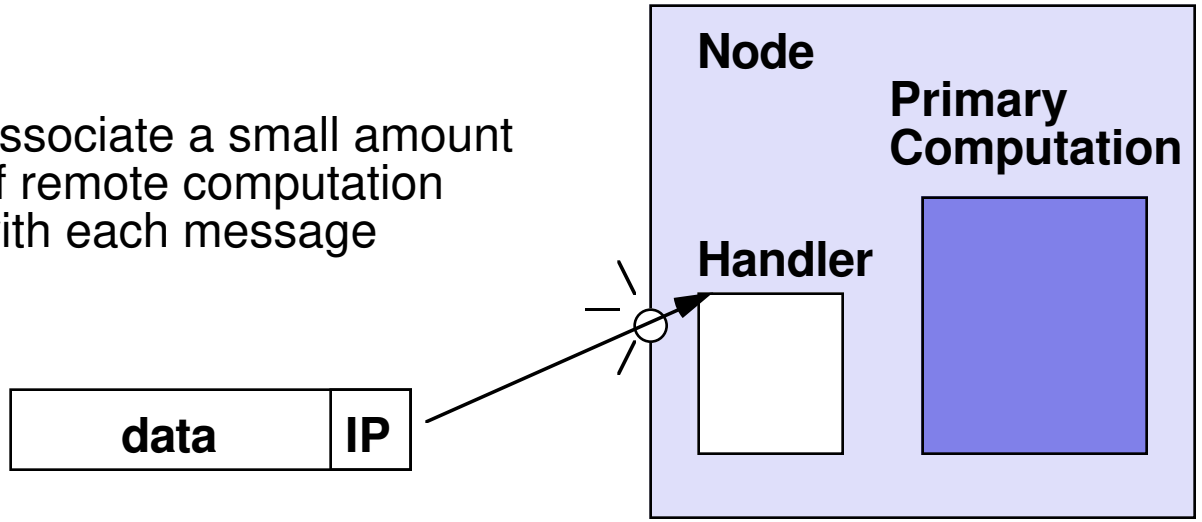
machine	year	send+recv overhead	flop cost	flops / snd+recv	cycle time
nCUBE/10	87	400µs	6.5us	62	100ns
IPSC/2	88	700µs	3.1µs	226	62ns
nCUBE/2	90	150µs	0.45µs	333	50ns
w/Act.Msg.		**25**µs		**55**	
IPSC/860	91	160µs	0.05µs	3200	25ns
Delta	91	55µs	0.05µs	1100	25ns
CM-5	92	95µs	0.31µs	306	30ns
w/Act.Msg.		**3**µs		**10**	

© D.E.Culler 93

Active Messages

Idea:

Associate a small amount of remote computation with each message



CM-5

1.6μs send,

1.7μs recv

nCUBE/2

11μs send,

13μs recv

Head of the message is the address of its handler

Handler executes immediately upon arrival

- extracts msg from network and integrates it with computation, possibly replies
- handler does not ``compute´´

*Note: **user**-level handler*

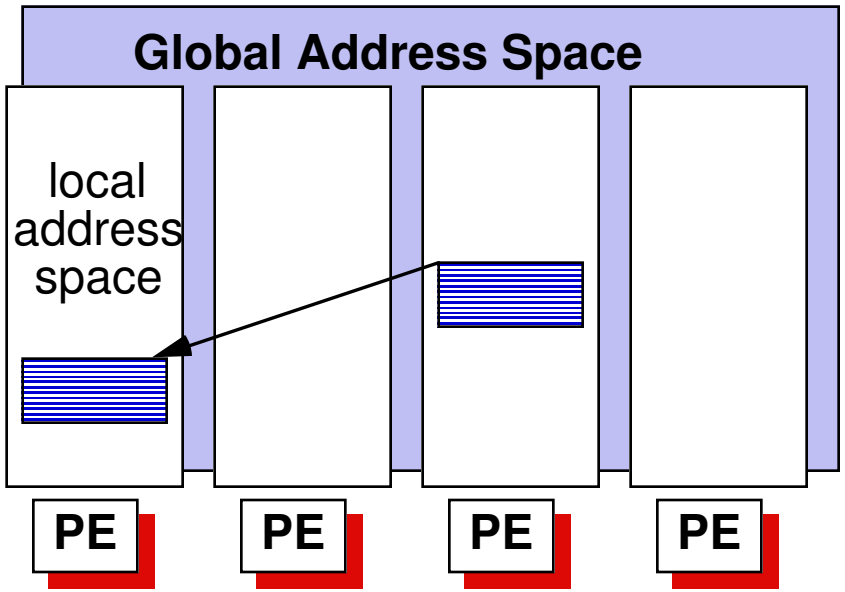
No buffering beyond transport

- data stored in pre-allocated storage
- quick service and reply, e.g., remote-fetch

© D.E.Culler 93

Split-C: a simple, efficient parallel language

Optimize remote access in a global address space



- **2-D Global Address Space**
  - physically distributed memory
  - no specified topology

```
double *global P;
x = *P
```

- **Split-phase access**
  - overlap communication

```
x := *P
```

- **Bulk transfer**

```
foo x = *Q;
```

- **Distributed Arrays**

```
double array A[n][m]::[k][k];
```

- **Signalling Stores**

```
P :- y;
```

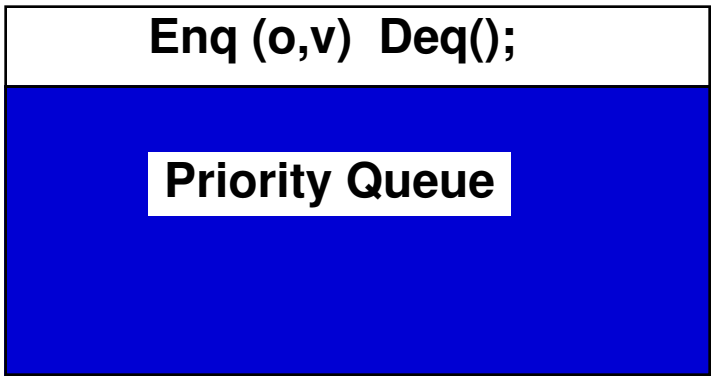
- **Locking access**

© D.E.Culler 93

MultiPol: Data structures for distributed memory machines

A Library of Multiported Objects

– Examples: work "queue", multiset, hash table, . . .



- Clean, meaningful interface
  - atomicity
- Portability above lib
- Weak consistency
- Multi-access

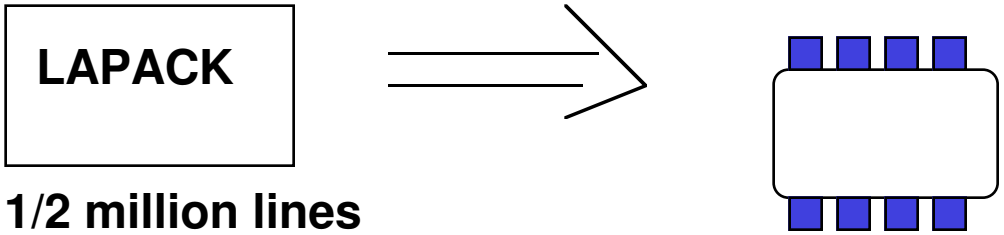
A suite of test applications
– Circuit simulation, Sparse Cholesky, Gröbner Basis

A methodology for developing correct parallel programs
– formal approaches to programming

© D.E.Culler 93

LAPACK II

Developing Scalable Codes for Dense and Sparse Linear Algebra



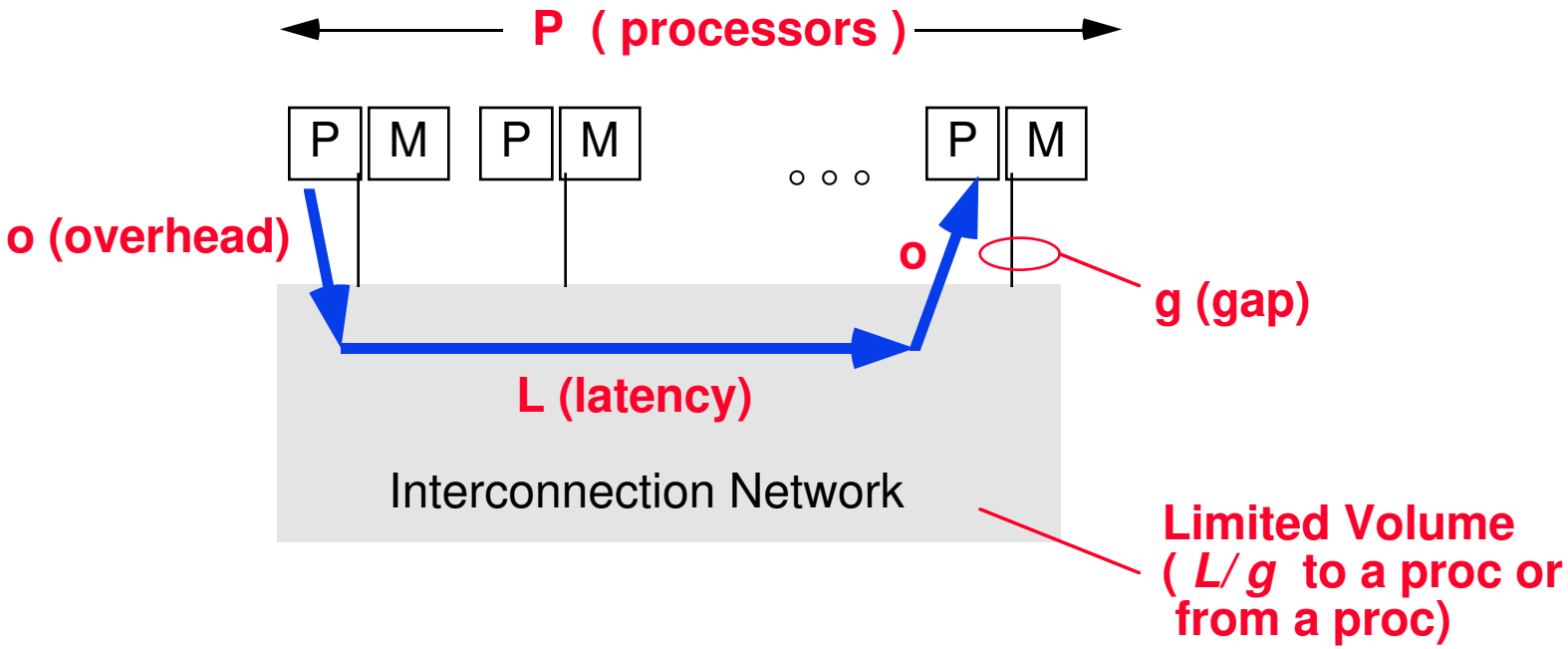
1/2 million lines
Fortan 77
Shared Memory

Distributed Memory

- **Initial study:**
  - **factorizations (A = LU, A = LL^T , A = QR)**
  - **reductions to condensed forms (UAU^T = H, UAU^T = T, UAV^T = B)**
  - **Eigenproblems / SVD for condensed forms.**
- **Issues:**
  - **new || algorithms, high performance via local BLAS**
  - **data structures, programming paradigm, IEEE arithmetic**
  - **communication primitives (LACS, Edelman)**

© D.E.Culler 93

The LogP Model



L: Latency incurred in sending message between modules

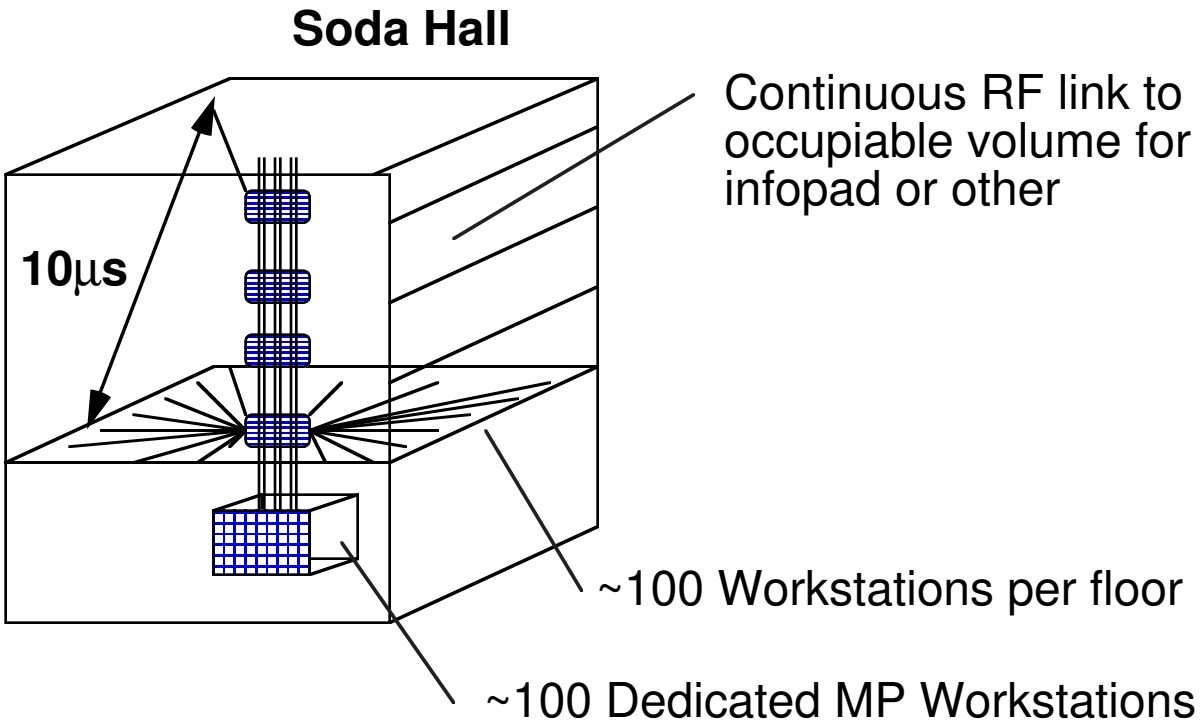
o: processor overhead to send *or receive* a message

g: Gap between messages sent or received by processor

P: number of processors/memory modules

© D.E.Culler 93

The Distributed Supercomputer



400 Snakes @ 25 MFLOPS => 10 GFLOPS

1.2 GBytes => 500 GB disk with 800 MB/s BW

64 MBytes => 25 GB memory

128 KBytes => 50 MB cache

Scale!

© D.E.Culler 93

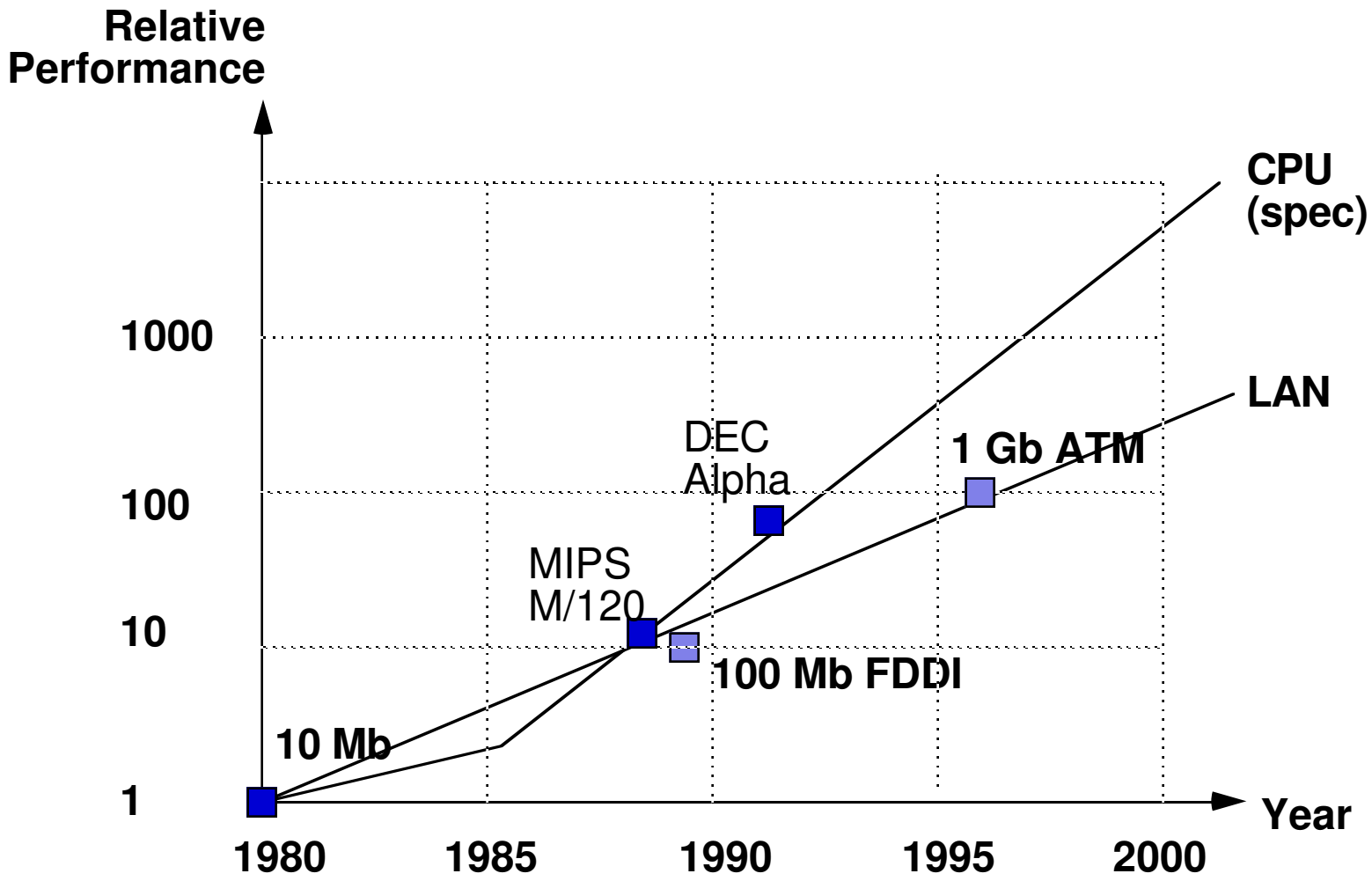
Workstations Falling Behind on Networks

- Combersome protocols
  - within a building is like across the world
- Layers and layers and layers
  - all do buffering, flow control, error handling
- Multiple copies
- I/O bus interface
- Poor O.S. support
- Wrong abstractions

Millisecond overheads, few megabits of bandwidth

© D.E.Culler 93

Divergence of CPU and LAN Performance



© D.E.Culler 93

Transfering Supercomputer Technology to NOW

- Dedicated, reliable (virtual) network
- Switched (point-to-point and very high aggregate BW)
- Low overhead, high bandwidth interface
  - Memory bus, user level, zero copies
- Low Latency (virtual cut-thru)
- Global address space
- Coherent scheduling
- High-performance numerical software

While retaining workstation advantages
- Cost effective
- Software base

© D.E.Culler 93

Proposed Effort

- Network Interface Card
  - **tight integration (mem bus)**
  - **provide user-level virtual network**
  - **support get / put and active messages**
  - **flow control and protection**
- Network fabric
  - **switches**
    - **+ ATM, CM-5, Paragon, CIT, PARC, . . .**
  - **links**
  - **maintenance**
- Operating systems
  - **NIC support**
  - **Global PIDs, coherent scheduling**
  - **Distributed file system**
- Mobile access

© D.E.Culler 93