



Start Voyager: Message Passing and DSM on SMP Clusters

**Arvind
Laboratory for Computer Science
Massachusetts Institute of Technology**



Massively Parallel Processors

Shared Memory

Cache-coherent

~~KSR~~

HP-Convex SPP 1200

Sequent Symmetry 5000

circa 1995

No global caches

Cray T-3D, T-3E

Distributed Memory, Message passing

~~TMC CM-5~~

~~Meiko CS-2~~

~~Intel Paragon~~

IBM SP2

nCube

Pyramid

Fujitsu VPP500, AP3000

~~C-DAC Param~~

Application: “Grand challenge” problems ?



Why MPPs have remained a fringe phenomenon ?

- driven by Grand Challenge problems and not by market forces
- too expensive in terms of both absolute cost and cost-performance
- MPP software is of little use in non-MPP environment
 - ↳ *few independent software developers*

MPPs = Massively Parallel Processors



What is needed for Parallel Computing to become Ubiquitous

- **Cost-effective Parallel Hardware**
- **Multiprocessing-capable standard Operating Systems**
- **Parallel programming models**
- **Scalability**

+

seamless transition from sequential to parallel computing



SMPs:

Main Stream Parallel Computing

**PC class SMPs are about to cause a revolution
(4-processor Pentium Pro Machine)**

- **cheap**
- **run NT, Solaris, Linux**
- **will track technology**

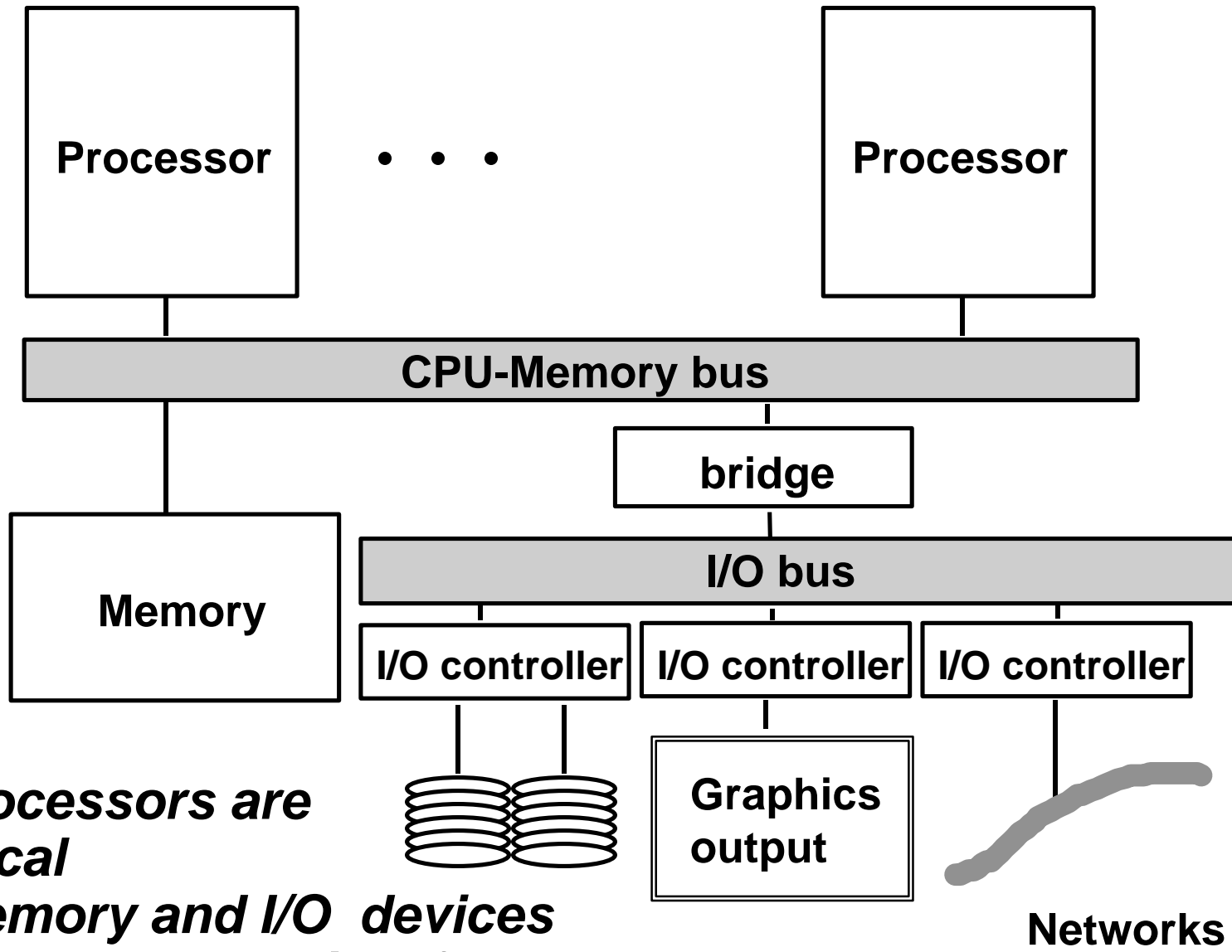


- ***sold in large numbers (100,000)***
- ***incentive for ISVs to produce parallel software
compilers, debuggers, performance tools
applications, libraries, ...***

SMPs = Symmetric Multi-Processors



Symmetric Multiprocessors



- *all processors are identical*
- *all memory and I/O devices are equally accessible from all processors*



Deluxe SMPs: Commodity Supercomputers

Deluxe SMPs have a superior memory system than low-end SMPs

- **SUN Enterprise Series:**
 - E5000 upto 12 procs**
 - E10000 upto 64 procs**
- **SGI**
 - Origin 2000 32 to 128 procs**
- **Digital**
 - Turbo Laser 8400 upto 12 procs**
 - Raw Hide 4 procs**

IBM, NEC, Hitachi, Fujitsu, ...

the cost

64-Processor SMP >> 16 x 4-Processor SMPs !!!



SMP Market

Current market : “servers”

Enterprise computing

\$\$\$\$

databases

OLTP

Web servers

Internet commerce ...

Few parallel applications

Potential Market

Technical and Scientific computing

\$

CAD

CAM

weather, climate...

drug design ...

SMPs are poised to replace all computers larger than notebooks



Scalability Issue

Grand Challenge problems require “Jumbos” \$

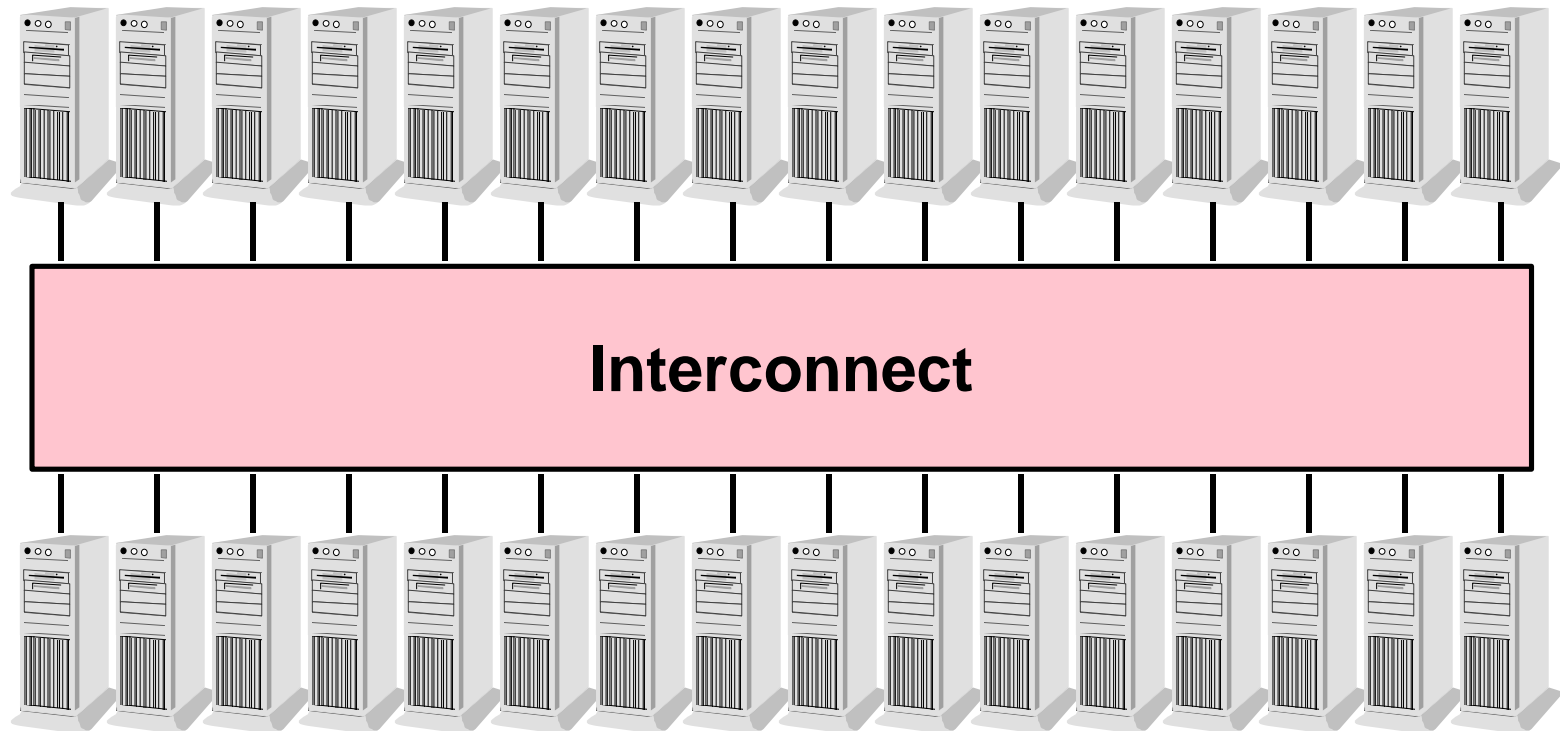
**Enterprise computing requires scalable SMPs
for growth** \$\$\$\$

*Unfortunately, SMPs don't scale well and don't
provide an incremental upgrade path*

Solution ⇒ **Clusters of SMPs**



Clusters: Scalable Parallel Machines



**Each SMP has an *adapter board* (with an embedded processor and network interface) to support *message passing* and *shared memory*
+ I/O + *Parallel OS layer* + ...**



ASCI Jumbos

Accelerated Strategic Computing Initiative (ASCI)
a U.S. Department of Energy program

- **ASCI Red machine** at Sandia \$54M
Intel: 9000 Pentium Pros (~1 Teraflops by 1997)
- **ASCI Pacific Blue** at Livermore \$96M
IBM: 512, 8-way SMPs (~3-4 Teraflops by 1998)
- **ASCI Mountain Blue** at Los Alamos \$120M
SGI: n, 32-way SMPs (~3-4 Teraflops by 1998)
- **ASCI White**
goal (~100 Teraflops by 2002)

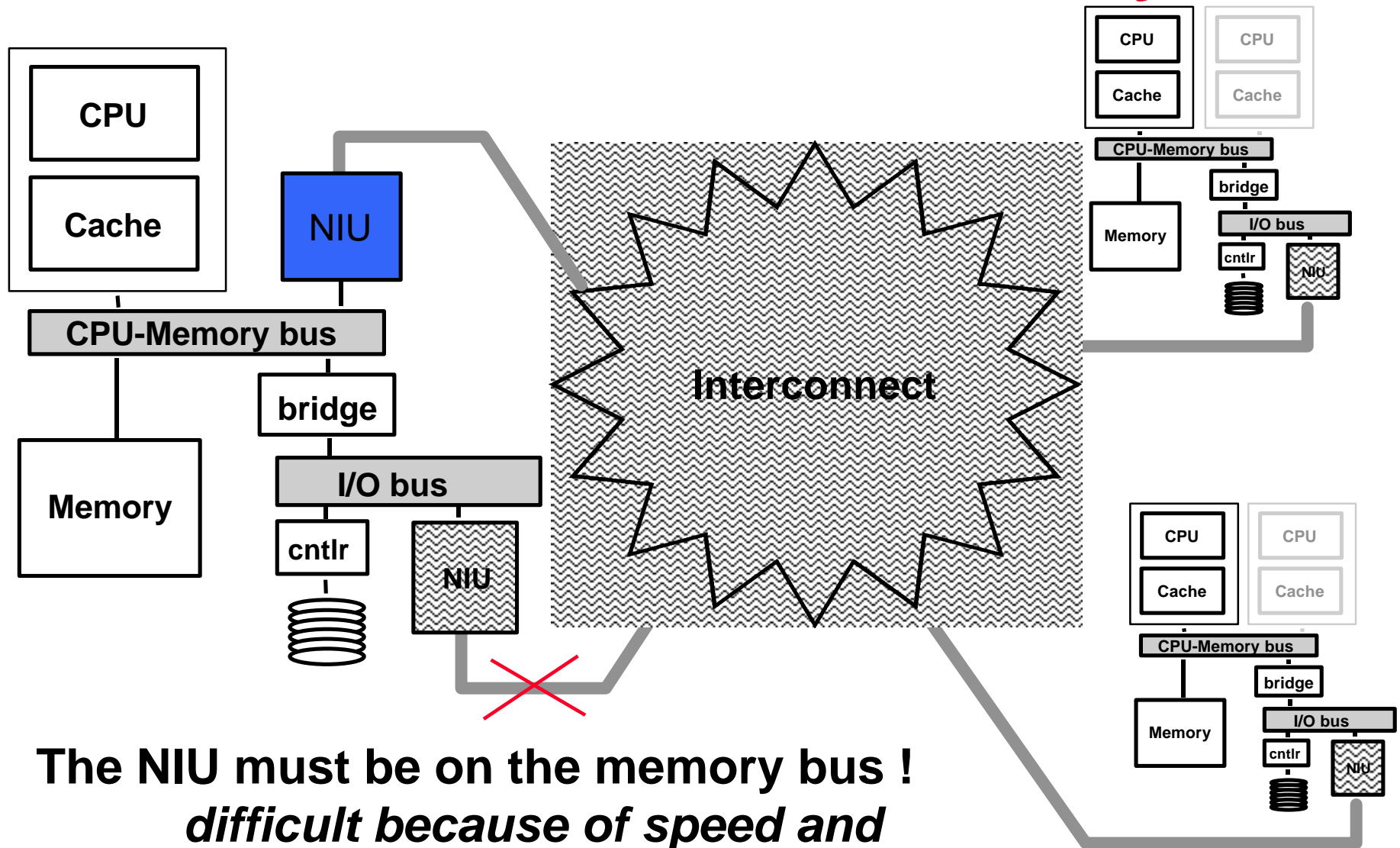


Challenges

- **How to make a cluster look like an SMP**
Shared Memory & Operating System issues
- **Cost effective networks and Network Interface Units (NIUs)**
- **Fault tolerance**
- **High-level multithreaded *programming model***

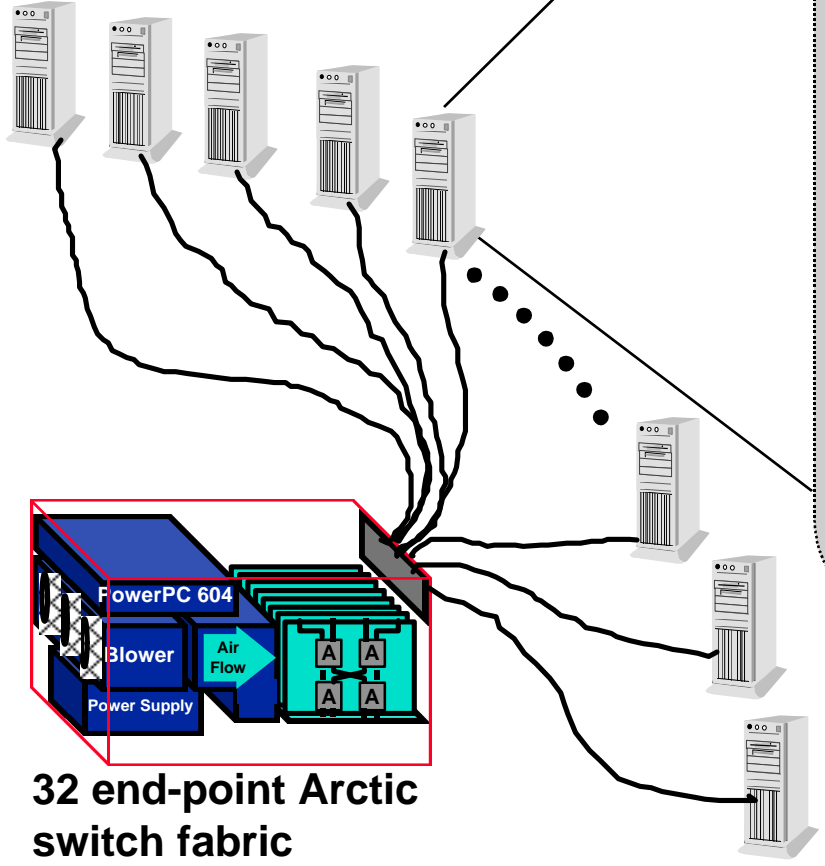
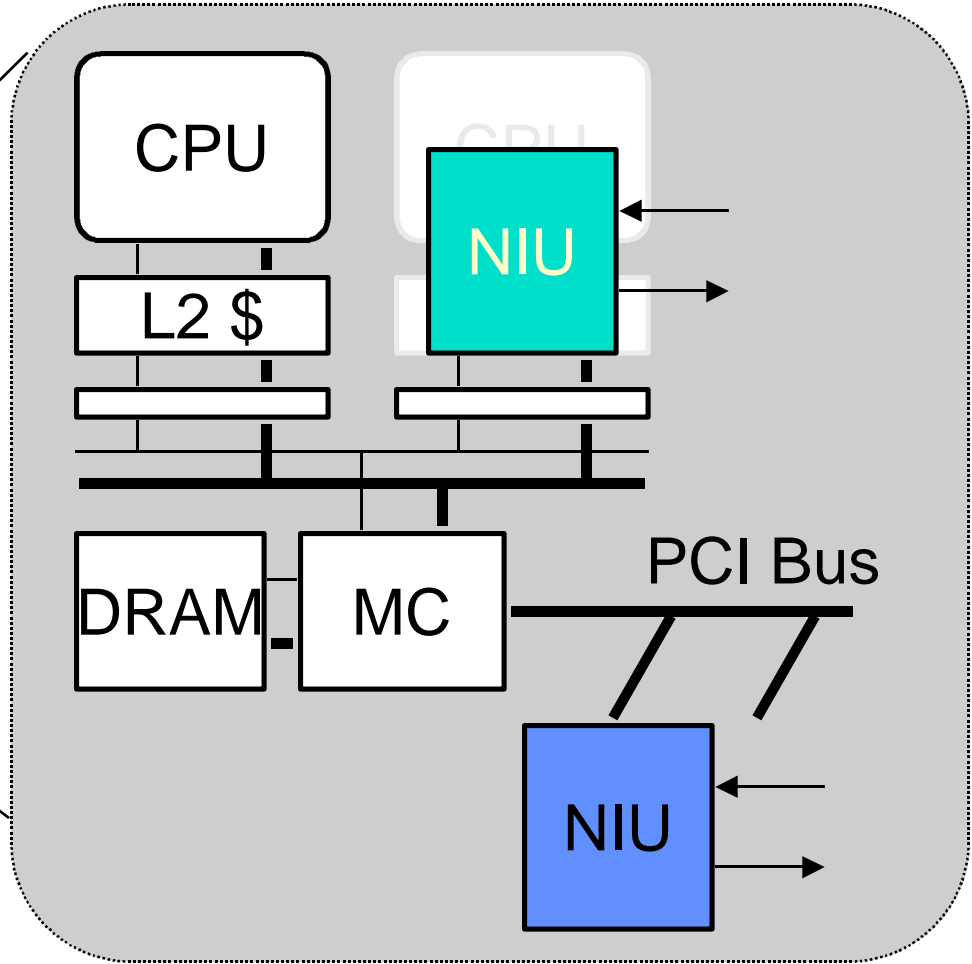


Cache-Coherent Distributed Shared Memory



**The NIU must be on the memory bus !
*difficult because of speed and
proprietary information***

The StarT Project



 **StarT-Voyager**
(with IBM)  **StarT-Jr**

32 end-point Arctic
switch fabric



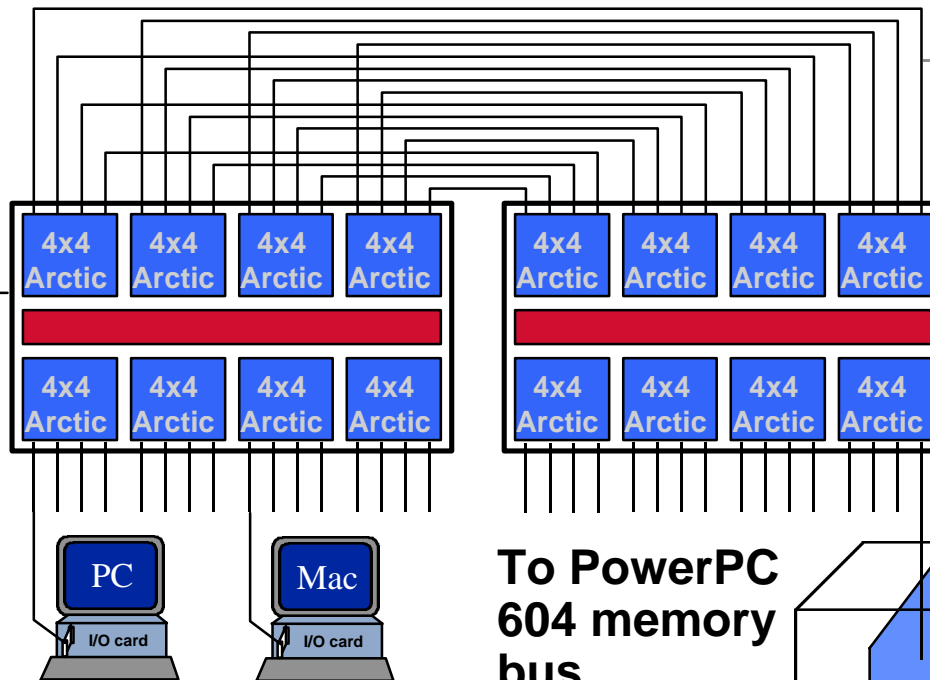
Arctic Network

Andy Boughton

Extensible Fat Tree

Network monitor

- control
- test
- statistics



320 MB/sec
full-duplex-links

- 2 priority levels
- 16-96 Byte packet

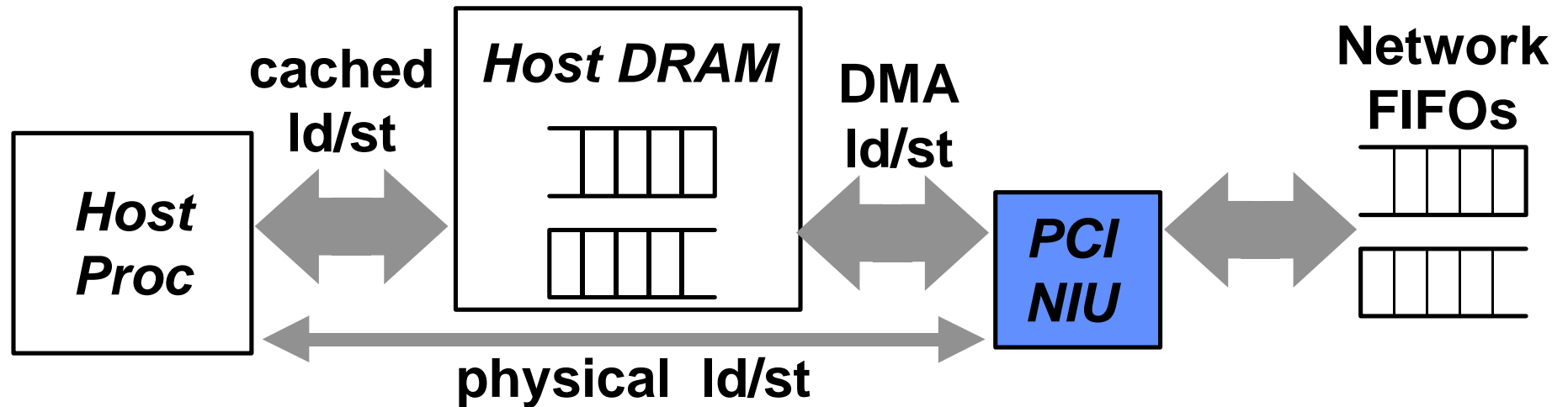
StarT-Jr : A High-performance network of PC's via I/O bus (PCI)

StarT-Voyager site



Arctic Network Infrastructure

James Hoe



Arctic network will connect several SMP clusters via PCI-based NIU's

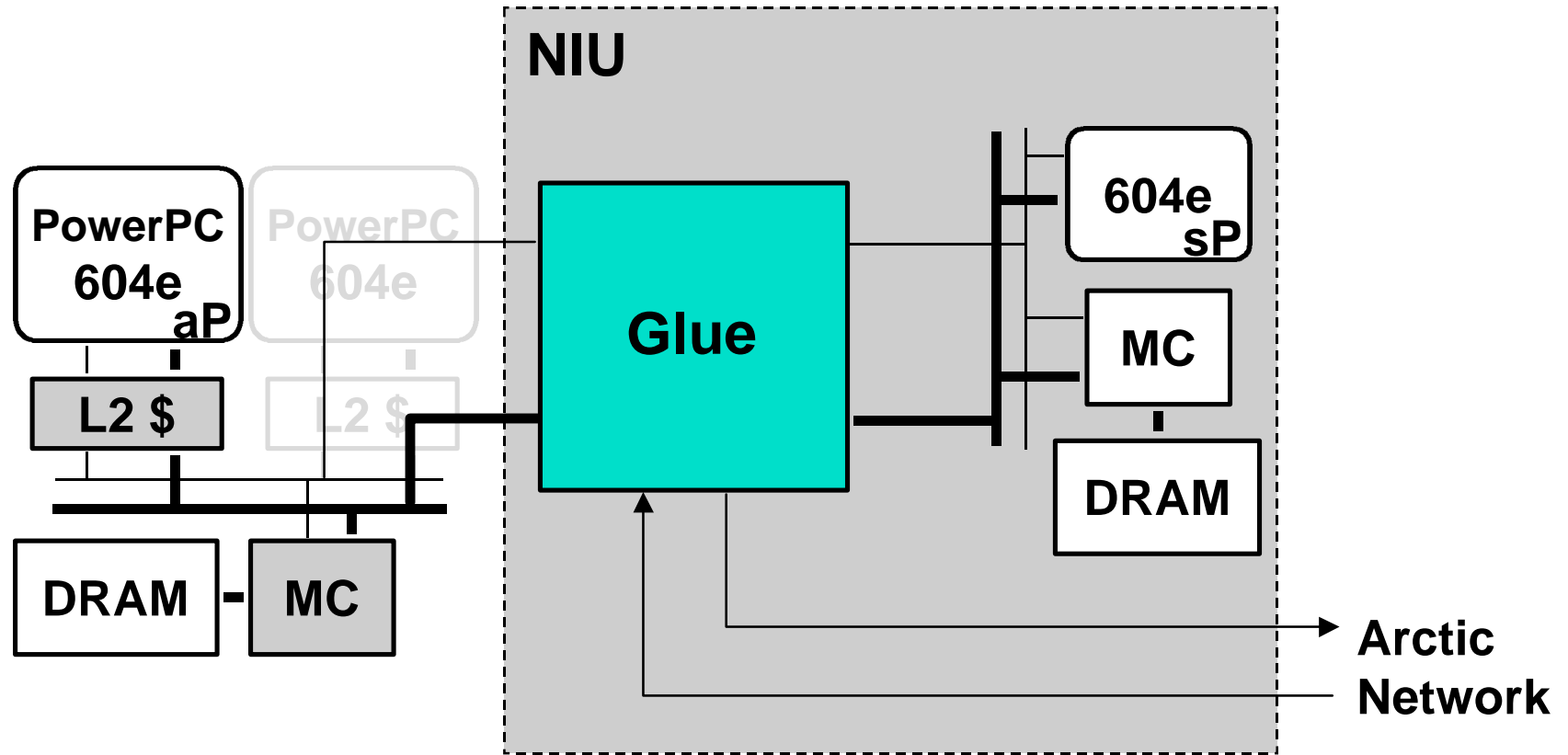
<i>SUN:</i>	9 8-processor E5000
<i>Digital:</i>	7 4-processor Raw Hides
<i>Intel:</i>	32 or more Quads

DMA performance determined by the host PCI bus
50 to 70 MB/s for sends, upto 90MB/s for receives



StarT-Voyager

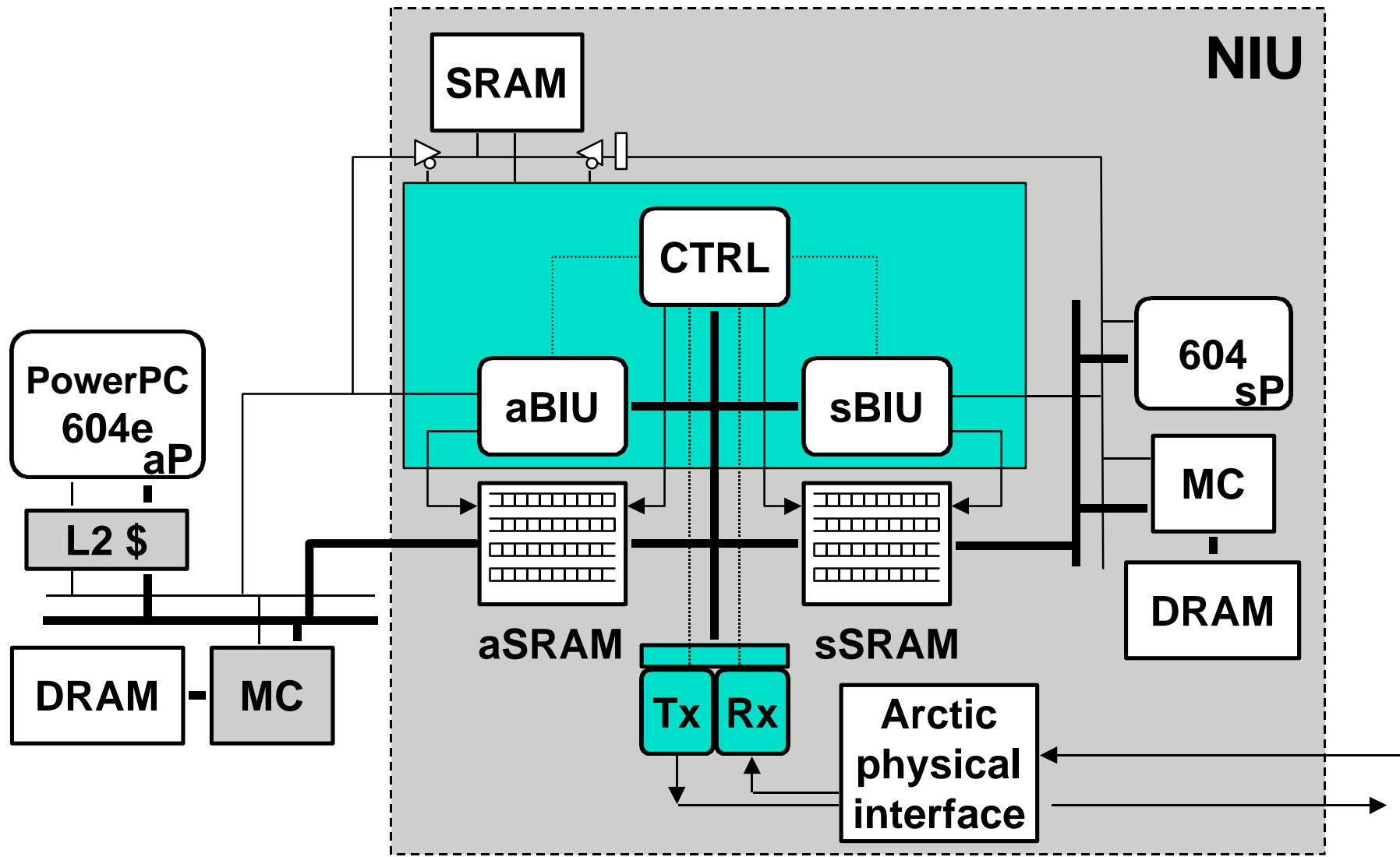
Boon S. Ang & Derek Chiou



Full protection in multi-user time-sharing environment



Network Interface Unit





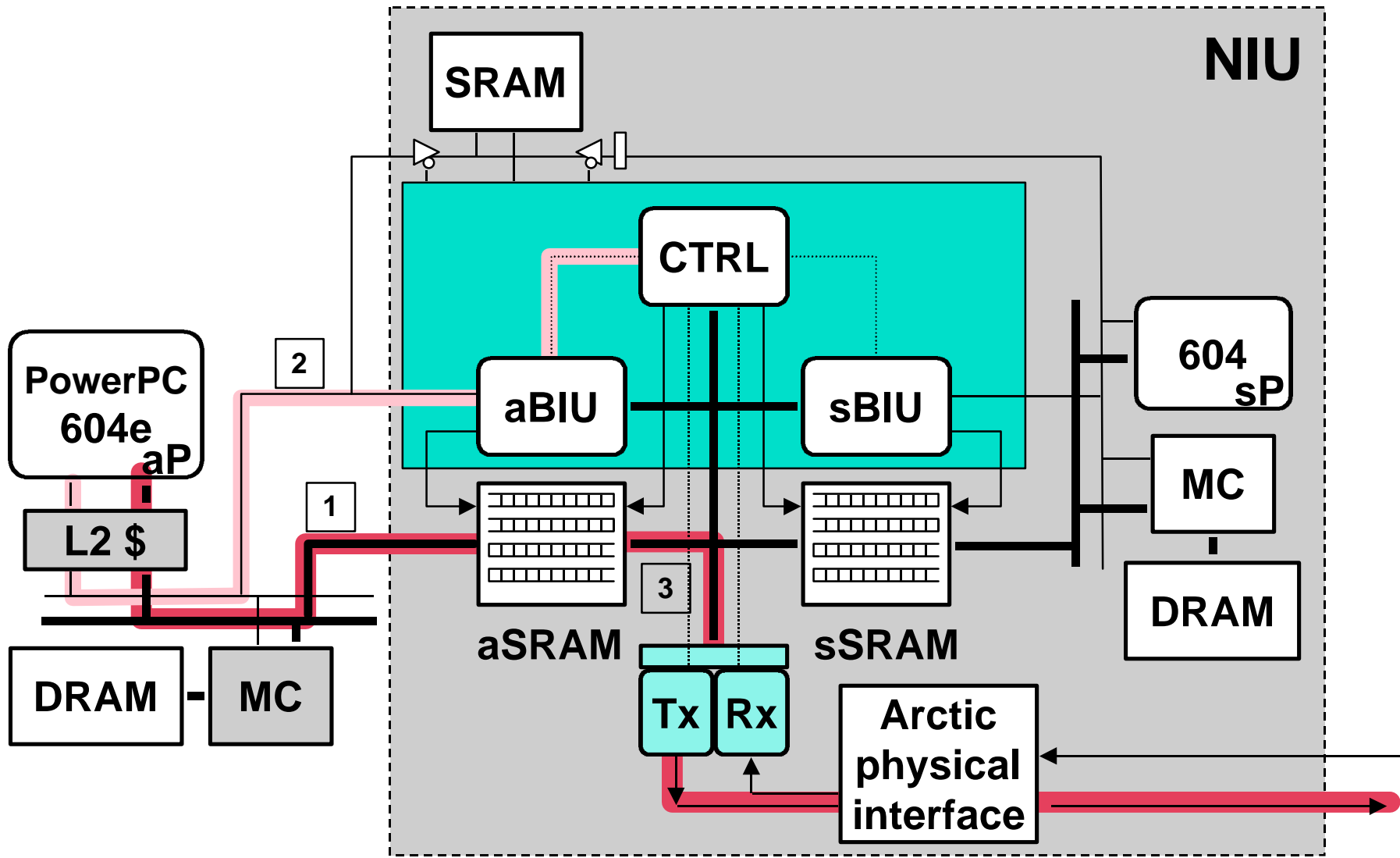
StarT- Voyager Features

- **Four message passing mechanisms optimized for different types of communication**
basic, express, DMA, tag-on
- **Global shared memory**
 - two mechanisms for inter-site memory accesses
S-COMA, NUMA
 - various memory models & associated protocols
- **Full protection in multi-user, time-sharing environment**

***Two 32-Processor machines to be delivered to
LCS and IBM Research in 1997***

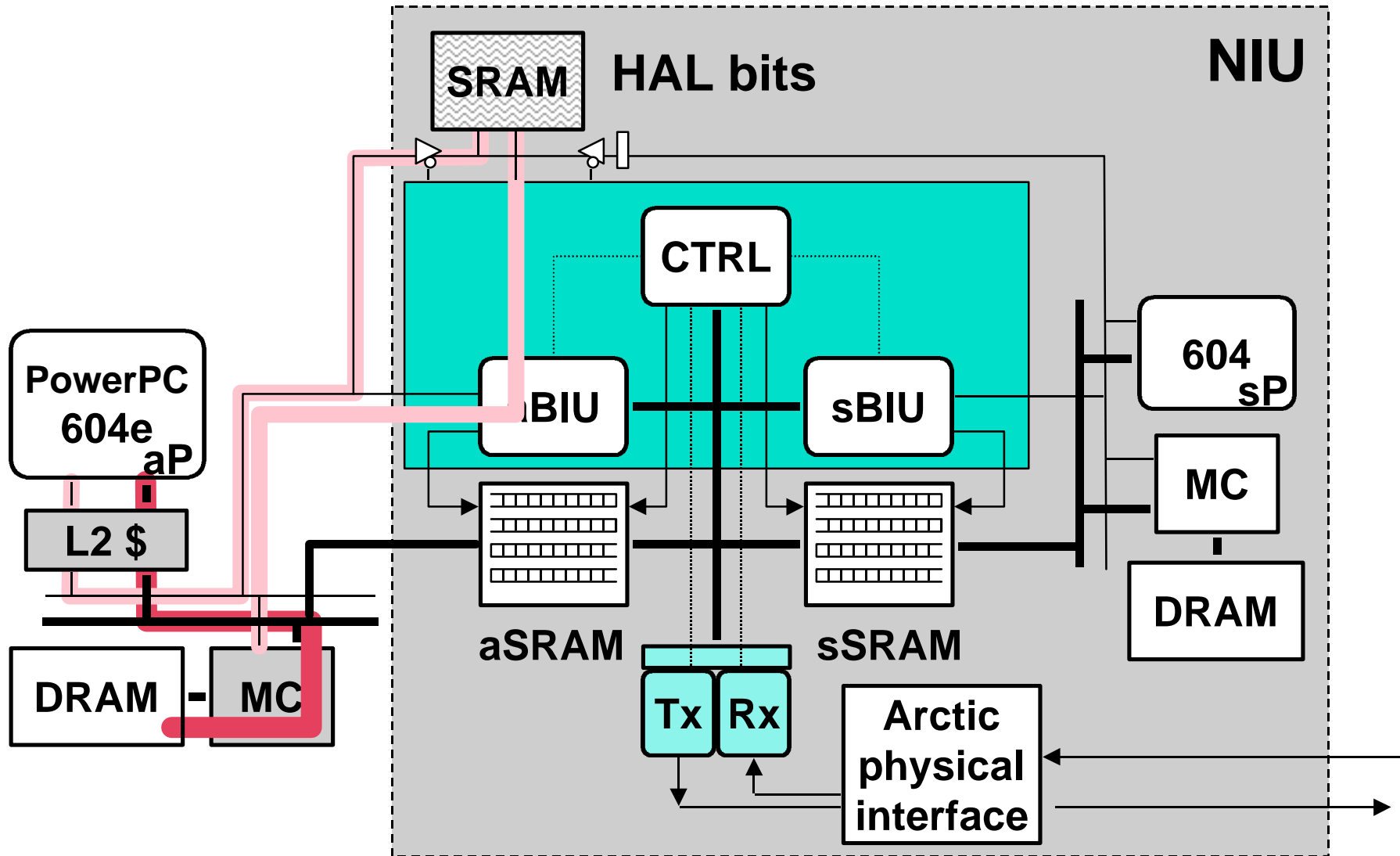


Sending a Basic Message





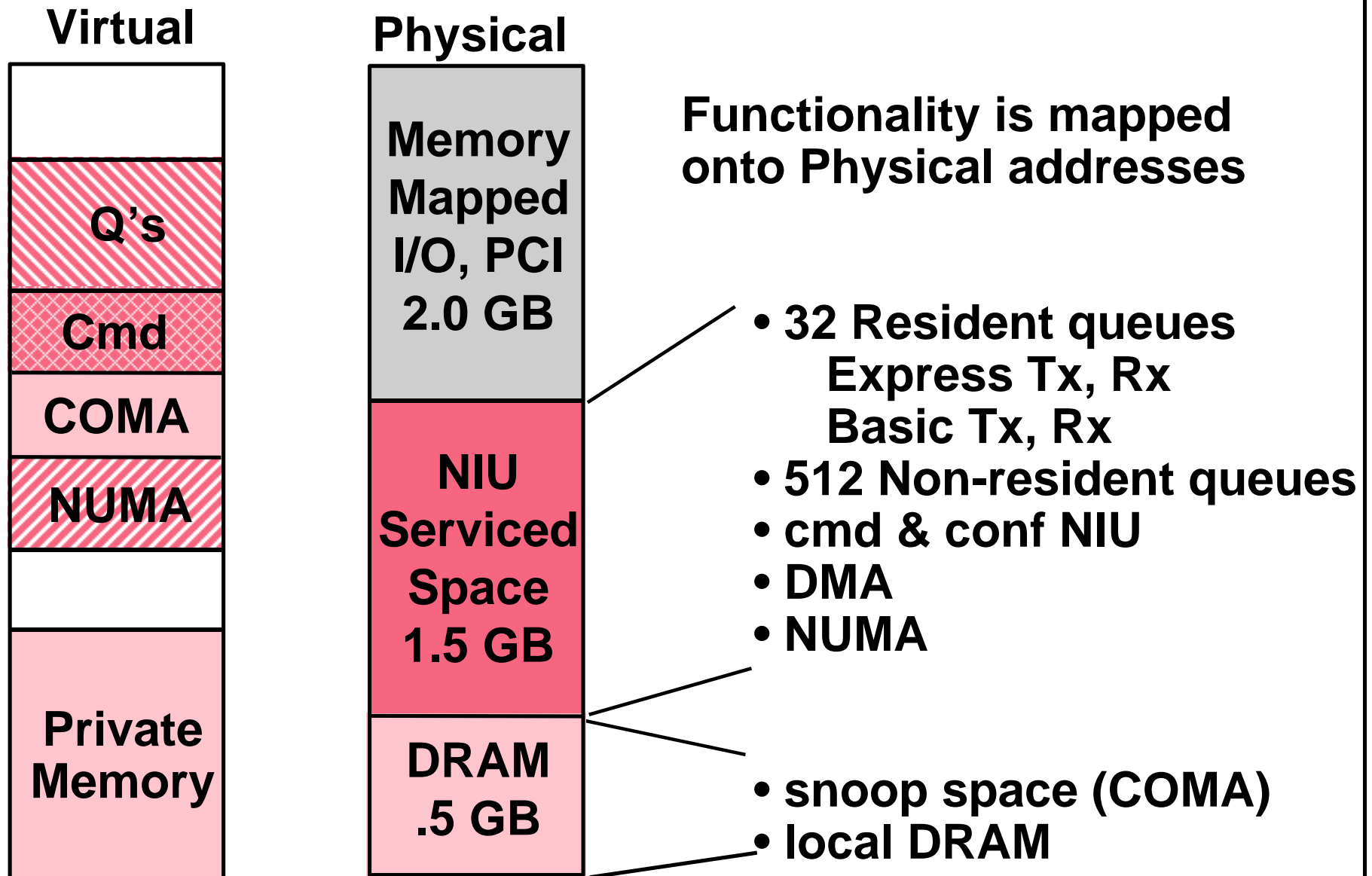
S-COMA Style Memory Access



HAL: F(Bus action, Address) \Rightarrow (proceed/retry) X (notify/~notify)



Functionality Map





Protection and Multitasking

- **Message queues and global address space are viewed as *resources* to be shared amongst processes**
- **Once a process has acquired a resource, virtual memory translation mechanisms protect access to it**
- **NIU provides protection in accessing remote resources**
- **Resident queues are dynamically allocated to the current process (soft context switch)**



Protecting Receive Queues

TxQ	Dest 0	...	Dest n
0	$\langle d_{00}, q_{00} \rangle, s_{00}$...	$\langle d_{0n}, q_{0n} \rangle, s_{0n}$
...			
m	$\langle d_{m0}, q_{m0} \rangle, s_{m0}$...	$\langle d_{mn}, q_{mn} \rangle, s_{mn}$

destination
site

destination
Rx queue

reply
Rx queue
(logical name)

NIU consults the table on each message send



StarT-Voyager Research

Ideal test-bed for exploring

- Effect of various message passing mechanisms
 - *word size, cache-line size, page size*
- Effect of cache-coherent shared memory mechanisms
 - *S-COMA, NUMA, ...*
- New Memory models
- Integrated message passing and shared memory
- Adaptive cache-coherence protocols
- Distributed/Parallel OS's
- Macro-speculative execution
- Memory hierarchy
-

while running realistic applications



Status: April 1997

- **StarT-Jr: 4-Processor demo using Firewire 1394
August 1996**
- **Arctic chip: In hand - March, 1997**
- **StarT-Jr: 4-Processor demo using Arctic- June, 1997**
- **StarT-Jr: demo using new NIU and Arctic- 3Q,1997**
- **StarT-Voyager: 4-Processor demo- 3Q,1997**
- **StarT-Voyager: Two 32-Processor machines- 4Q,1997
*one for LCS and one for IBM Research***



StarT-Voyager Team

April 1997

Architects

Boon S. Ang & Derek Chiou

Implementors

*Mike Ehrlich, Chris Conley, Jack Costanza,
Daniel L. Rosenband, Brad Bartley*

CC protocols

Xiaowei Shen

Arctic

G. Andy Boughton, Jack Costanza

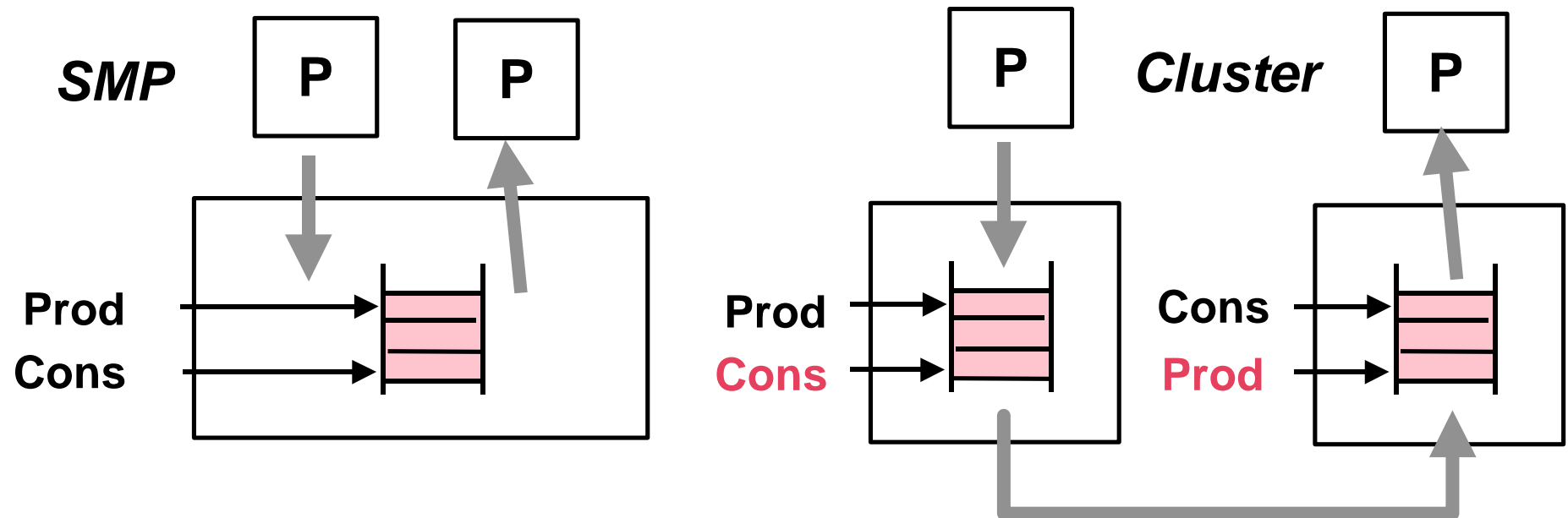
Software

*Larry Rudolph, Andy Shaw, Alex Caro
Paul Johnson,*



Integrated View of Programming

SMP's and Clusters



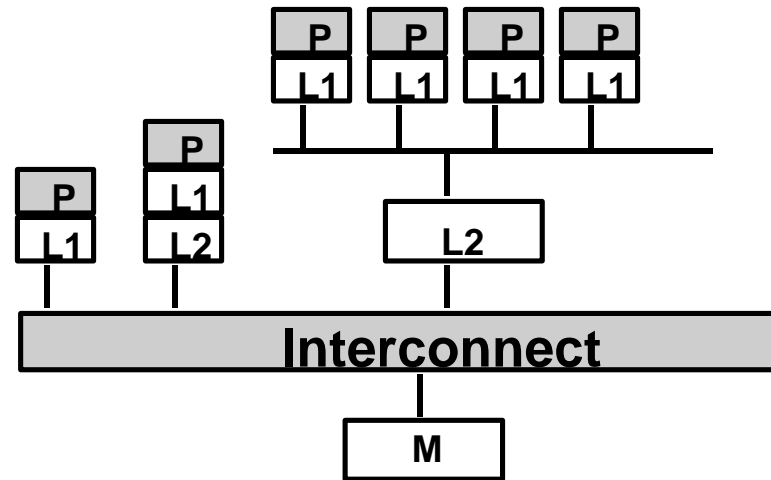
Message passing can be viewed as shared memory with strange semantics

- updating a sender's Producer pointer causes a ***message to be sent***
- receipt of a message causes ***update of receiver's **Producer pointer*****

Sender's view of buffers & pointers may not match receiver's.



Adaptive Cache Coherence Protocols



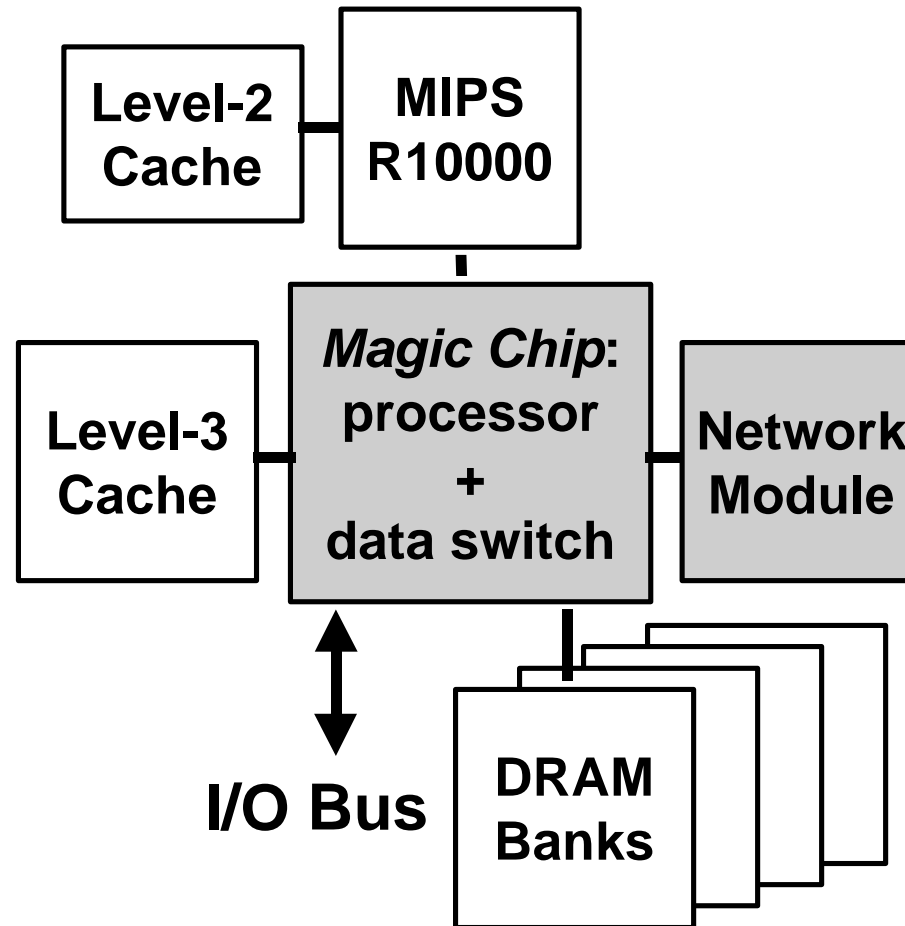
- Different protocols at different levels for efficiency
network vs bus, invalidate vs update, ...
- Adjust the protocol based on the usage pattern
or user/compiler directives

Major hurdle: *design and verification of new protocols*

We are developing a new formalism and tools to define
memory models and associated protocols precisely.



Stanford's CCDSM: *Flash*



**Magic is a static two-way superscalar processor.
Has to be fast because *all* memory traffic goes through it
⇒ *lots of I/O pins !***



Parallel Programming



Parallel Programming Models

High-level

Data parallel: *Fortran 90, HPF, ...*

Multithreaded: *Cid, Cilk, ...*
Id, pH, Sisal, ...

Low-level

Message passing: *PVM, MPI, ...*

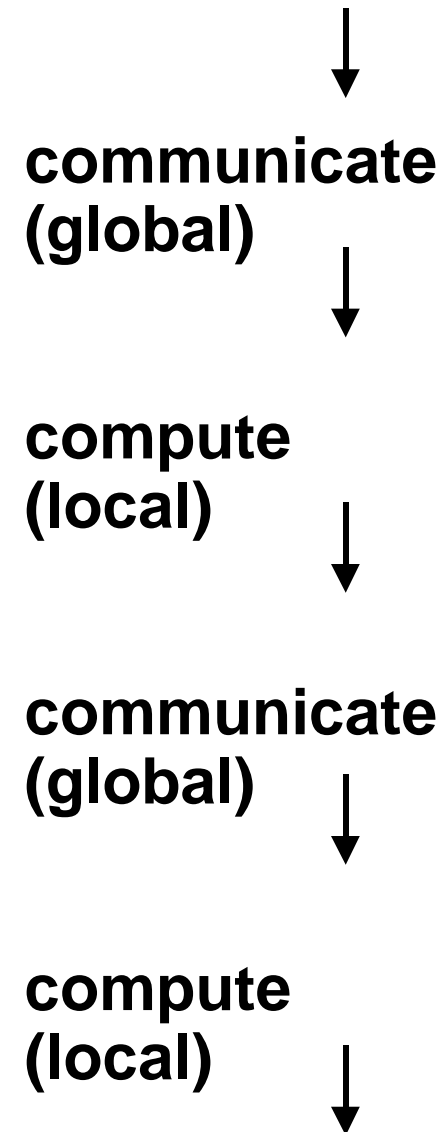
Threads & synchronization:
Futures, Forks, Semaphores, ...



Data Parallel Model

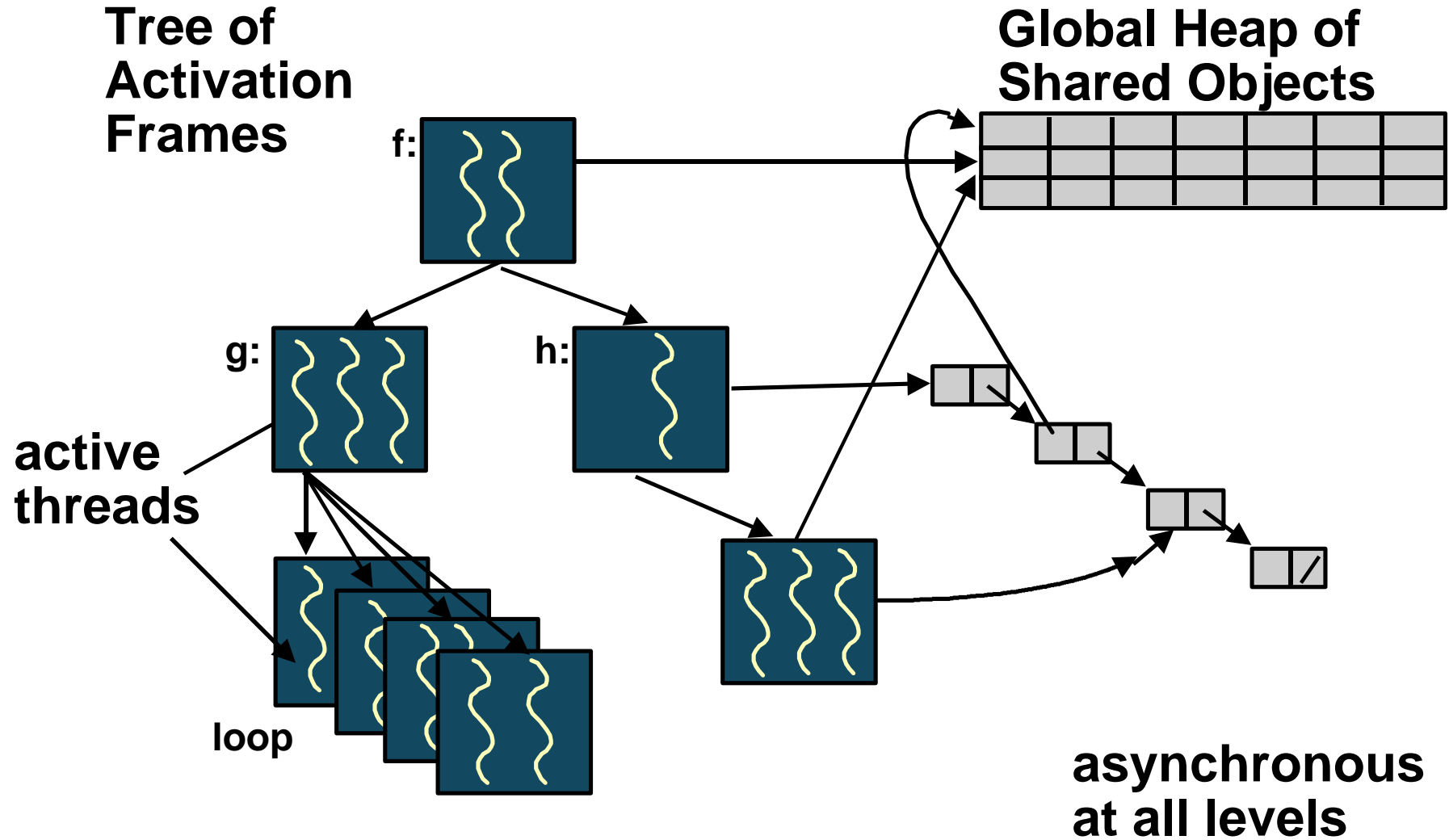
- All processors execute the *same program*
- *Global communication* primitives allow processors to exchange data
- *Implicit global barrier* after each communication

Too restrictive for General Purpose computing





Multithreaded Model





Explicit vs Implicit Multithreading

Explicit:

C + forks + joins + semaphores

***multithreaded C:* Cilk, Cid, ...**

***quick access to coarse-grain parallelism
in existing codes but ...***

Implicit:

**languages that specify only *a partial order*
*on operations***

***functional languages:* Id, pH, ...**

***safe, high-level, but difficult to implement
efficiently without shared memory & ...***



Future

Id pH Cilk HPF

```
graph TD; Id --> IL; pH --> IL; Cilk --> IL; HPF --> IL; IL --> Notebooks; IL --> SMPs; IL --> Clusters; style IL fill:none,stroke:none
```

multithreaded intermediate language

notebooks

SMPs

Clusters
of SMPs

*Freshman in a decade from now will be taught
sequential programming as a special case of
parallel programming*