Parallel Processors

Krste Asanovic
Laboratory for Computer Science
Massachusetts Institute of Technology

http://www.csg.lcs.mit.edu/6.823

Parallel Processing: The Holy Grail

- **Use multiple processors to improve runtime of a *single* task**
  - **technology limits speed of uniprocessor**
  - **economic advantages to using replicated processing units**
- **Preferably programmed using a portable high-level language**

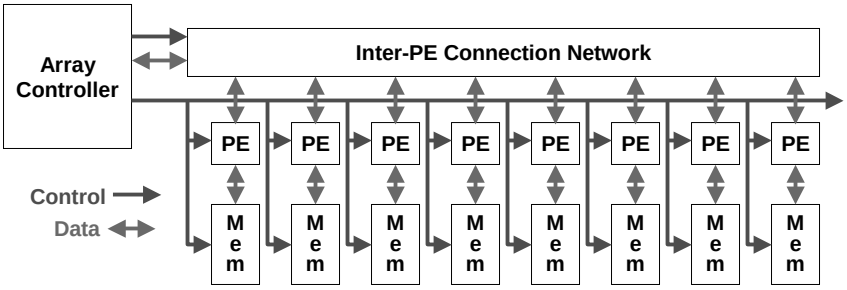
Flynn's Classification (1966)

Broad classification of parallel computing systems based on number of instruction and data streams

- **SISD: Single Instruction, Single Data**
  - **conventional uniprocessor**
- **SIMD: Single Instruction, Multiple Data**
  - **one instruction stream, multiple data paths**
  - **distributed memory SIMD (MPP, DAP, CM-1&2, Maspar)**
  - **shared memory SIMD (STARAN, vector computers)**
- **MIMD: Multiple Instruction, Multiple Data**
  - **message passing machines (Transputers, nCube, CM-5)**
  - **non-cache-coherent shared memory machines (BBN Butterfly, T3D)**
  - **cache-coherent shared memory machines (Sequent, Sun Starfire, SGI Origin)**
- **MISD: Multiple Instruction, Single Data**
  - **no commercial examples**

SIMD Architecture

- **Central controller broadcasts instructions to multiple processing elements (PEs)**



- **Only requires one controller for whole array**
- **Only requires storage for one copy of program**
- **All computations fully synchronized**

SIMD Machines

- Illiac IV (1972)
  - 64 64-bit PEs, 16KB/PE, 2D network
- Goodyear STARAN (1972)
  - 256 bit-serial associative PEs, 32B/PE, multistage network
- ICL DAP (Distributed Array Processor) (1980)
  - 4K bit-serial PEs, 512B/PE, 2D network
- Goodyear MPP (Massively Parallel Processor) (1982)
  - 16K bit-serial PEs, 128B/PE, 2D network
- Thinking Machines Connection Machine CM-1 (1985)
  - 64K bit-serial PEs, 512B/PE, 2D + hypercube router
  - CM-2: 2048B/PE, plus 2,048 32-bit floating-point units
- Maspar MP-1 (1989)
  - 16K 4-bit processors, 16-64KB/PE, 2D + Xnet router
  - MP-2: 16K 32-bit processors, 64KB/PE

(Also shared memory SIMD vector supercomputers
TI ASC ('71), CDC Star-100 ('73), Cray-1 ('76))

SIMD Today

- Distributed memory SIMD failed as large-scale general-purpose computer platform
  - required huge quantities of data parallelism (>10,000 elements)
  - required programmer-controlled distributed data layout
- Vector supercomputers (shared memory SIMD) still successful in high-end supercomputing
  - reasonable efficiency on short vector lengths (10-100 elements)
  - single memory space
- Distributed memory SIMD popular for special purpose accelerators
  - image and graphics processing
- Renewed interest for Processor-in-Memory (PIM)
  - memory bottlenecks => put some simple logic close to memory
  - viewed as enhanced memory for conventional system
  - technology push from new merged DRAM + logic processes
  - commercial examples, e.g., graphics in Sony Playstation-2

MIMD Machines

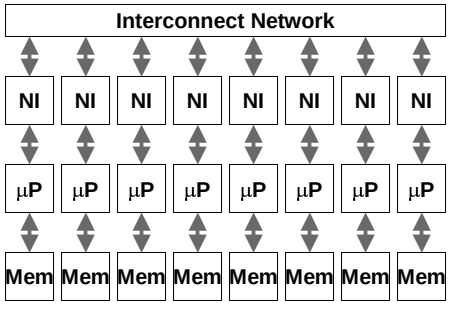
- **Message passing**
  - » **Thinking Machines CM-5**
  - » **Intel Paragon**
  - » **Meiko CS-2**
  - » **many cluster systems (e.g., IBM SP-2, Linux Beowulfs)**
- **Shared memory**
  - – **no hardware cache coherence**
    - » **IBM RP3**
    - » **BBN Butterfly**
    - » **Cray T3D/T3E**
    - » **Parallel vector supercomputers (Cray T90, NEC SX-5)**
  - – **hardware cache coherence**
    - » **many small-scale SMPs (e.g. Quad Pentium Xeon systems)**
    - » **large scale bus/crossbar-based SMPs (Sun Starfire)**
    - » **large scale directory-based SMPs (SGI Origin)**

Message Passing MPPs

(Massively Parallel Processors)

- **Initial Research Projects**
  - – **Caltech Cosmic Cube (early 1980s) using custom Mosaic processors**
- **Commercial Microprocessors including MPP Support**
  - – **Transputer (1985)**
  - – **nCube-1(1986) /nCube-2 (1990)**
- **Standard Microprocessors + Network Interfaces**
  - – **Intel Paragon (i860)**
  - – **TMC CM-5 (SPARC)**
  - – **Meiko CS-2 (SPARC)**
  - – **IBM SP-2 (RS/6000)**
- **MPP Vector Supers**
  - – **Fujitsu VPP series**

Designs scale to 100s-10,000s of nodes



Message Passing MPP Problems

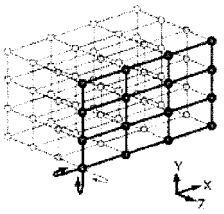
- All data layout must be handled by software
  - cannot retrieve remote data except with message request/reply
- Message passing has high software overhead
  - early machines had to invoke OS on each message (100μs-1ms/message)
  - even user level access to network interface has dozens of cycles overhead (NI might be on I/O bus)
  - sending messages can be cheap (just like stores)
  - receiving messages is expensive, need to poll or interrupt

Shared Memory Machines

- Two main categories
  - non cache coherent
  - hardware cache coherent
- Will work with any data placement (but might be slow)
  - can choose to optimize only critical portions of code
- Load and store instructions used to communicate data between processes
  - no OS involvement
  - low software overhead
- Usually some special synchronization primitives
  - fetch&op
  - load linked/store conditional
- In large scale systems, the logically shared memory is implemented as physically distributed memory modules

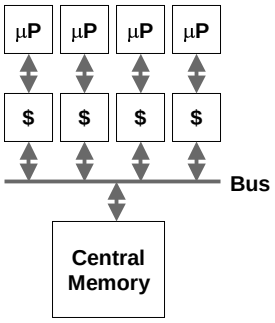
Cray T3E

Up to 2,048 675MHz Alpha 21164 processors connected in 3D torus



- **Each node has 256MB-2GB local DRAM memory**
- **Load and stores access global memory over network**
- **Only local memory cached by on-chip caches**
- **Alpha microprocessor surrounded by custom "shell" circuitry to make it into effective MPP node. Shell provides:**
  - **multiple stream buffers instead of board-level (L3) cache**
  - **external copy of on-chip cache tags to check against remote writes to local memory, generates on-chip invalidates on match**
  - **512 external E registers (asynchronous vector load/store engine)**
  - **address management to allow all of external physical memory to be addressed**
  - **atomic memory operations (fetch&op)**
  - **support for hardware barriers/eureka to synchronize parallel tasks**

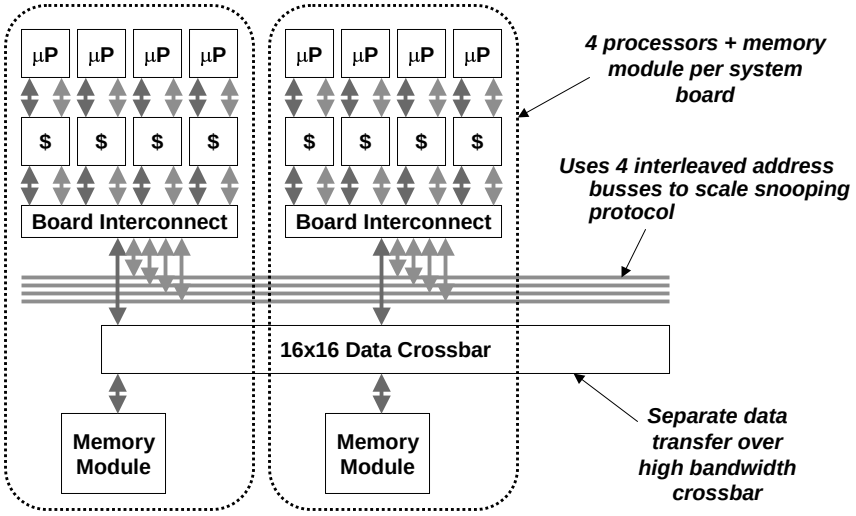
Bus-Based Cache-Coherent SMPs



- **Small scale (<= 4 processors) bus-based SMPs by far the most common parallel processing platform today**
- **Bus provides broadcast and serialization point for simple snooping cache coherence protocol**
- **Modern microprocessors integrate support for this protocol**

Sun Starfire (UE10000)

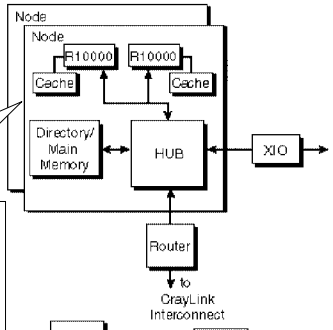
• Up to 64-way SMP using bus-based snooping protocol



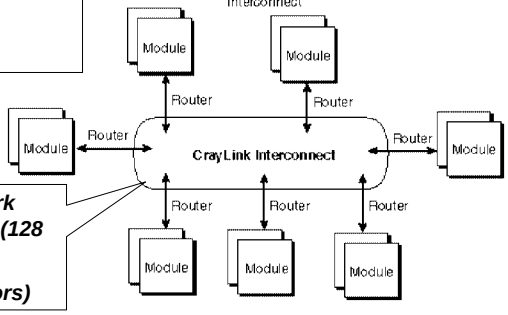
SGI Origin 2000

• Large scale distributed directory SMP

• Scales from 2 processor workstation to 512 processor supercomputer



Node contains:
- *Two MIPS R10000 processors plus caches*
- *Memory module including directory*
- *Connection to global network*
- *Connection to I/O*



Scalable hypercube switching network supports up to 64 two-processor nodes (128 processors total)

(Some installations up to 512 processors)

Origin Directory Representation

(based on Stanford DASH)

H	S	bit-vector C	Memory Block

Bit-vector is a representation of which children caches have copies of this memory block

At home

(H=1, S= _) : no cached copy exists (R[ε])

Read Only Copies

(H=0, S=1) : for all C_i=1, i^{th} child has a copy (R[Dir])

Writable Copy at C_i

(H=0, S=0) : for C_i=1, i^{th} child has the Ex copy (W[id])

size?

Directory Size

Directory size = (M / B) · [s+N] / 8 Bytes

where

M = Memory size
B = Block size
N = number of children
s = no. of bits to represent the state

For M= 2^{32} Bytes, B=64 Bytes, s = 2 bits

Directory size = $(2^{(32-6)}) \cdot (2+N) / 8$ Bytes
= $2^{23} \cdot (2+N)$ Bytes

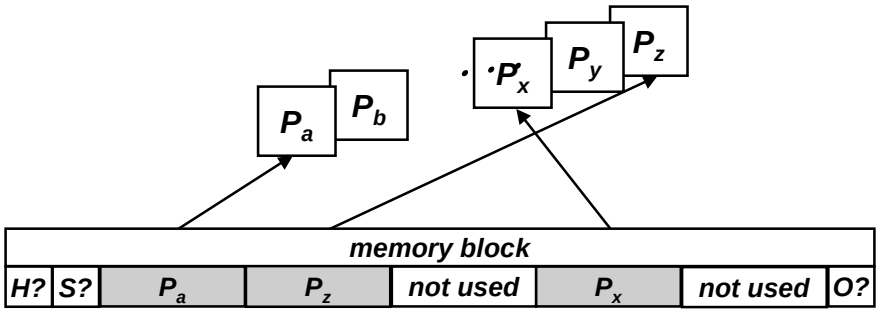
N=16 ⇒ directory ≈ 2^{27} Bytes or ~ *4% overhead*
N=256 ⇒ directory ≈ 2^{31} Bytes or ~ *50 % overhead*

This directory data structure is practical for small N but does not scale well !
(Origin shares 1 bit per 2 processors for <= 64 processors, 1 bit per 8 processors in 512 processor version)

Krste
May 14, 2001
6.823, L24--17

Reducing the Directory Size

***Limitless directories*- Alewife, MIT**



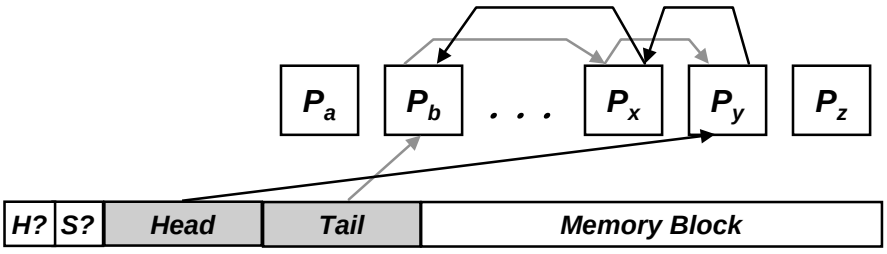
Instead of a N-bit-vector, keep n (*lg N*- bit) pointers; if more than n children request a copy, handle the overflow in software

effective for large N and low degree of sharing

Krste
May 14, 2001
6.823, L24--18

Reducing the Directory Size

***linked-list* - SCI (Scaleable Coherent Interface)**



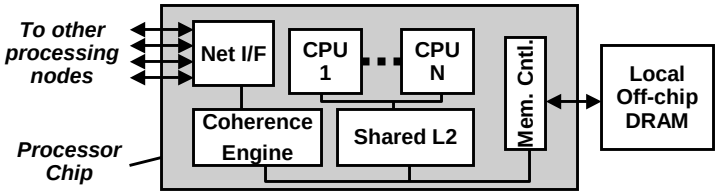
- **Part of the directory is attached to each cache**
- **Home and each cache block keep two (*lg N)*-bit pointers per memory block**
- **A *doubly linked-list* of the cache blocks holding the same memory block is maintained, with *the root* at the home site**

  ⇒ *less storage but bad performance for many readers*

Trends in High-End Server CPUs

Higher transistor counts enable greater integration

- **DRAM controllers on chip (UltraSPARC-III, Alpha 21364)**
  - reduce main memory latency (from 250ns->170ns in UltraSPARC-III)
  - increase memory bandwidth (21364 has over 12 GB/s peak memory bandwidth to 8 Rambus DRAM channels)
- **On-chip network routers (Alpha 21364, Power-4)**
  - cheaper and faster connectivity for cache coherence traffic
- **Multiple processors on one chip**
  - separate cores for chip-scale multiprocessing (IBM Power-4)
  - simultaneous multithreading (Alpha 21464)



Diseconomies of Scale

- **Few customers require the largest machines**
  - much smaller volumes sold
  - have to amortize development costs over smaller number of machines
- **Different hardware required to support largest machines**
  - dedicated interprocessor networks for message passing MPPs
  - T3E shell circuitry
  - large backplane for Starfire
  - directory storage and routers in SGI Origin

⇒ ***Large machines cost more per processor than small machines!***

Clusters

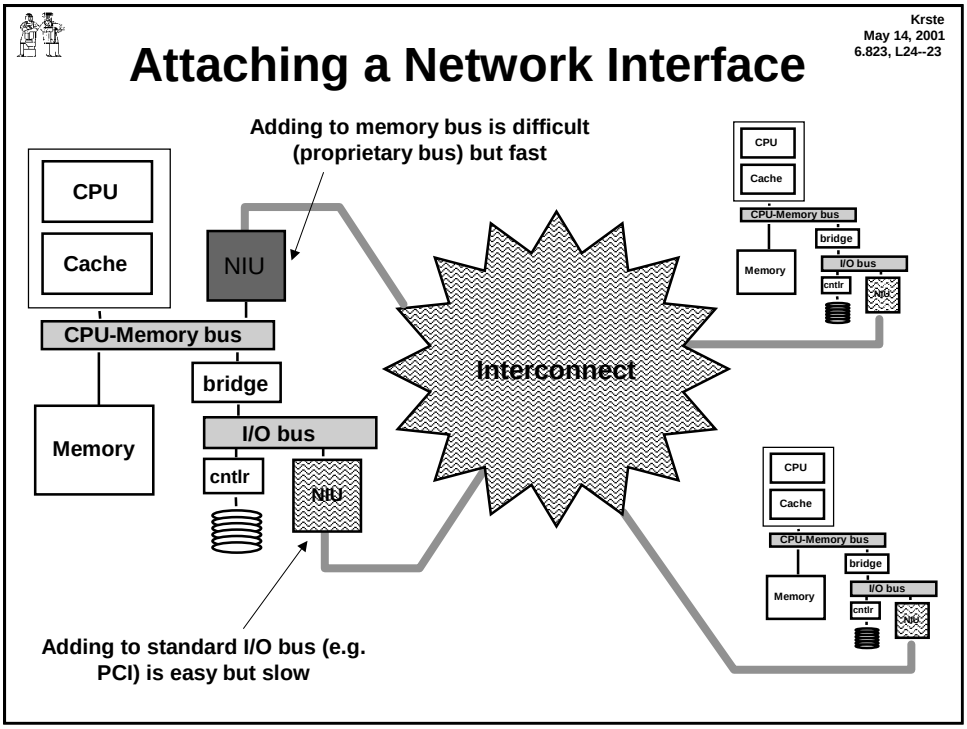
Connect multiple *complete* machines together using standard fast interconnects

- **Little or no hardware development cost**
- **Each node can boot separately and operate independently**
- **Interconnect can be attached at I/O bus (most common) or on memory bus (higher speed but more difficult)**

Clustering initially used to provide fault tolerance (DEC, IBM)

Networks of Workstations

- **Build message passing MPP by connecting multiple workstations together using fast interconnect attached to I/O bus**
  - **UC Berkeley NOW, 100 Sun Ultra-1 workstations connected by Myrinet network**
  - **Shrimp @ Princeton**
- **Node performance tracks workstation technology (same boxes!)**
- **Sold commercially by IBM (SP-2), and many other vendors (including Cray selling Alpha clusters)**



Attaching a Network Interface

Clusters of SMPs (CluMPs)

- **Connect multiple n-way SMP boxes using a cache coherent interface on memory bus**
  - **HP Convex Exemplar (SCI interconnect between 8-way SMPs)**
  - **Sequent NUMA-Q (SCI interconnect between 4-way SMPs)**
- **Connect multiple n-way SMPs using a fast message passing network**
  - **SGI PowerChallenge Array (HIPPI network connecting 18-way SMPs)**
- **Connect multiple n-way non-cache coherent SMPs using a fast non-coherent interconnect**
  - **Vector supercomputer clusters (e.g., connect up to 32 SX-5 16-way SMPs)**

Portable Parallel Programming?

- **Most large scale commercial installations emphasize throughput**
  - **database servers, web servers, file servers**
  - **independent transactions**
- **Wide variety of parallel systems**
  - **message passing**
  - **shared memory**
  - **shared memory within node, message passing between nodes**

$\Rightarrow$ ***Little commercial software support for portable parallel programming***

Message Passing Interface (MPI) standard widely used for portability

- **lowest common denominator**
- **"assembly" language level of parallel programming**