SE-292 High Performance Computing

Intro. To Concurrent Programming & Parallel Architecture

R. Govindarajan
govind@serc

PARALLEL ARCHITECTURE

Parallel Machine: a computer system with more than one processor

- Special parallel machines designed to make this interaction overhead less

Questions:

- What about Multicores?
- What about a network of machines?
  - Yes, but time involved in interaction (communication) might be high, as the system is designed assuming that the machines are more or less independent

2

Classification of Parallel Machines

Flynn's Classification

- In terms of number of Instruction streams and Data streams
- Instruction stream: path to instruction memory (PC)
- Data stream: path to data memory
- SISD: single instruction stream single data stream
- SIMD
- MIMD

3

PARALLEL PROGRAMMING

- Recall: Flynn SIMD, MIMD
  - Programming side: SPMD, MPMD
  - Shared memory: threads
  - Message passing: using messages
- Speedup = $\frac{T_{sequential}}{T_{parallel}}$

4

How Much Speedup is Possible?

Let ***s*** be the fraction of sequential execution time of a given program that can not be parallelized

Assume that program is parallelized so that the remaining part (***1 - s)*** is perfectly divided to run in parallel across ***n*** processors

$$\lim_{n\to\infty} \text{Speedup} = \frac{1}{s + \frac{1-s}{n}}$$

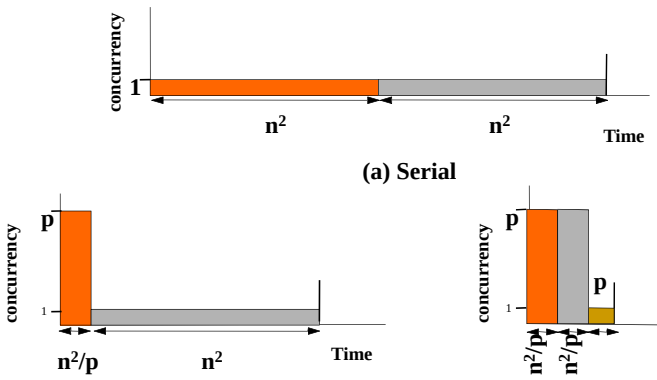
Maximum speedup achievable is limited by the sequential fraction of the sequential program

Amdahl's Law

5

Understanding Amdahl's Law

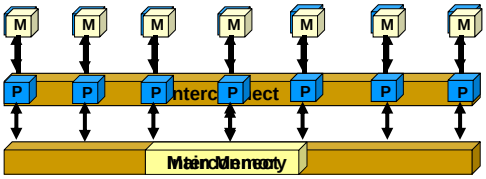
Compute : *min (|A[i,j]- B[i,i]|)*



6

Classification 2: Shared Memory vs Message Passing

- Shared memory machine: The n processors share physical address space
  - Communication can be done through this shared memory



- The alternative is sometimes referred to as a message passing machine or a distributed memory machine

7

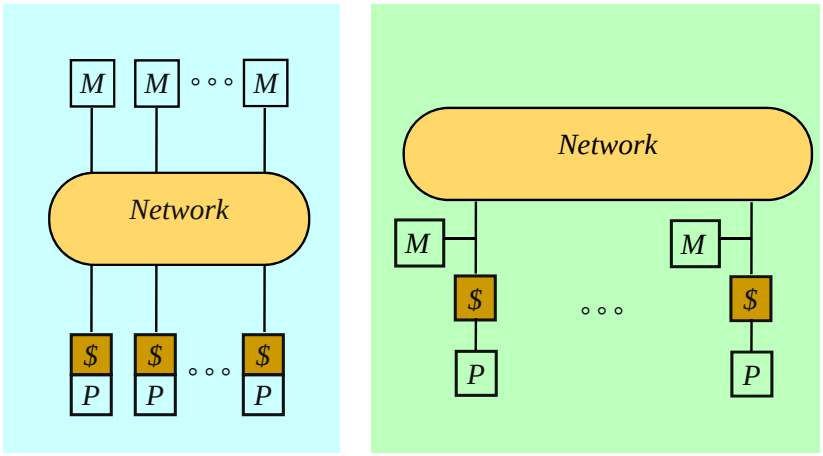
Shared Memory Machines

The shared memory could itself be distributed among the processor nodes

- Each processor might have some portion of the shared physical address space that is physically close to it and therefore accessible in less time
- Terms: Shared vs Private
- Terms: Local vs Remote
- Terms: Centralized vs Distributed Shared
- Terms: NUMA vs UMA architecture
  - Non-Uniform Memory Access

8

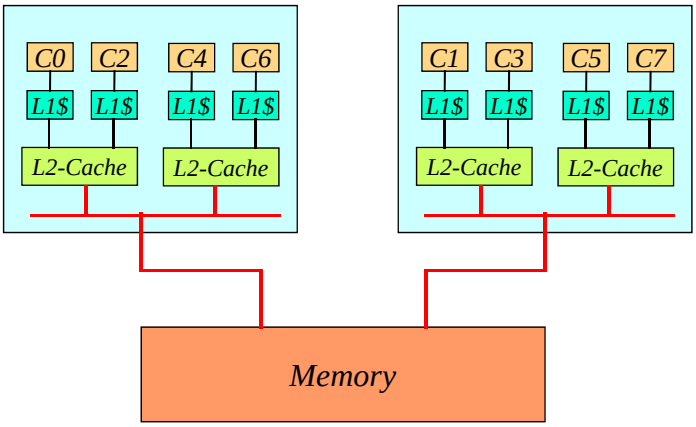
Shared Memory Architecture



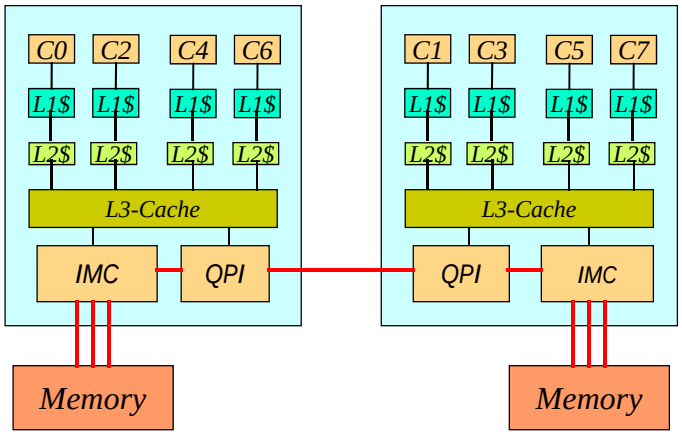
Centralized Shared Memory
Uniform Memory Access (UMA)

Distributed Shared Memory
Non-Uniform Memory Access (NUMA)

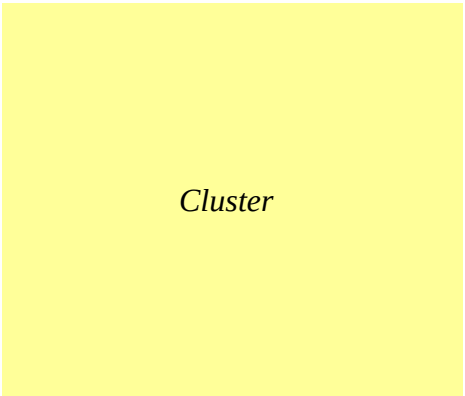
MultiCore Structure



NUMA Architecture



Distributed Memory Architecture



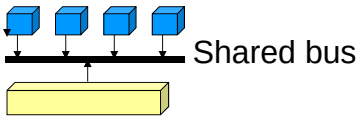
- Message Passing Architecture
  - Memory is private to each node
  - Processes communicate by messages

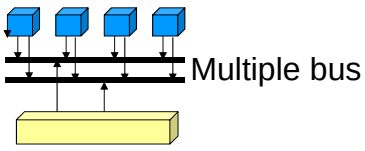
Parallel Architecture: Interconnections

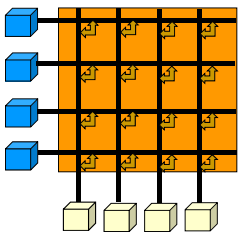
- Indirect interconnects: nodes are connected to interconnection medium, not directly to each other
  - Shared bus, multiple bus, crossbar, MIN
- Direct interconnects: nodes are connected directly to each other
  - Topology: linear, ring, star, mesh, torus, hypercube
  - Routing techniques: how the route taken by the message from source to destination is decided

13

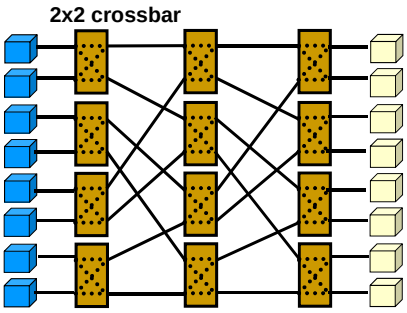
Indirect Interconnects







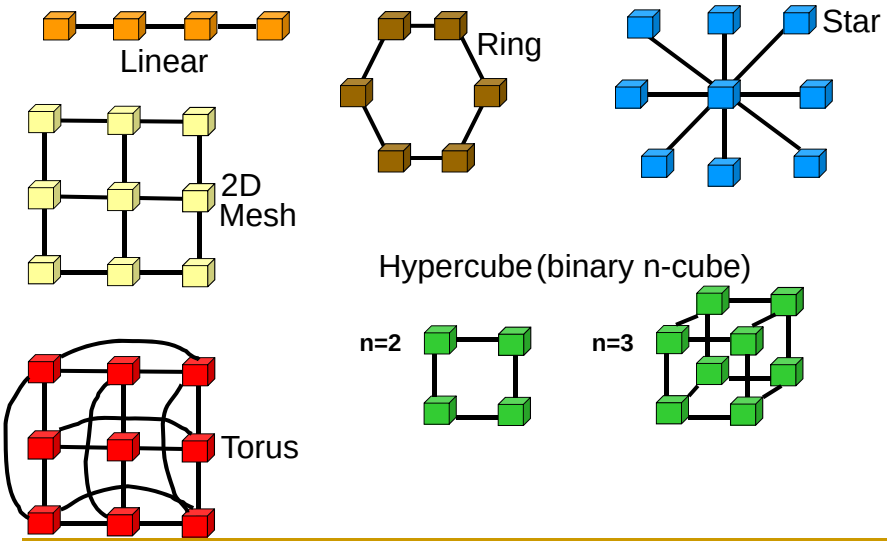
Crossbar switch



Multistage Interconnection Network

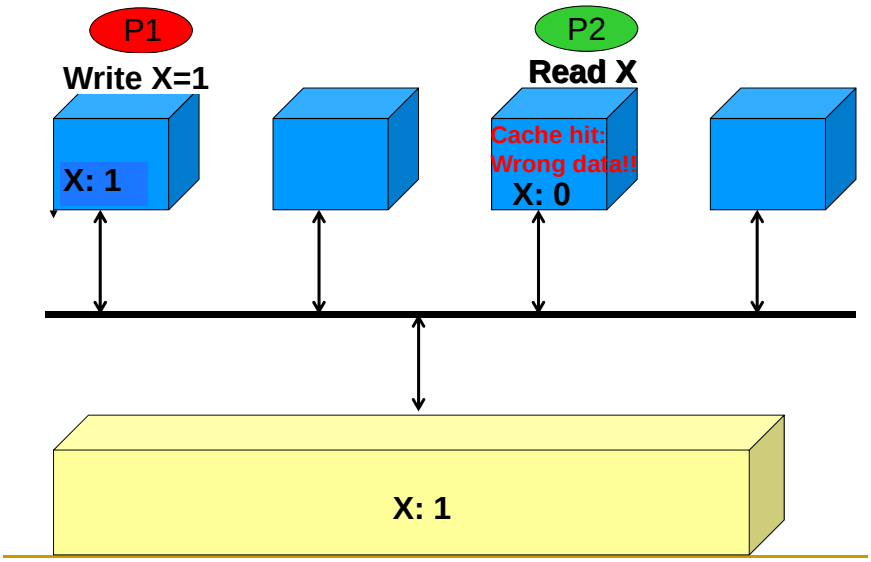
14

Direct Interconnect Topologies



15

Shared Memory Architecture: Caches



16

Cache Coherence Problem

- If each processor in a shared memory multiple processor machine has a data cache
  - Potential data consistency problem: the cache coherence problem
  - Shared variable modification, private cache
- Objective: processes shouldn't read `stale' data
- Solutions
  - Hardware: cache coherence mechanisms
  - Software: compiler assisted cache coherence

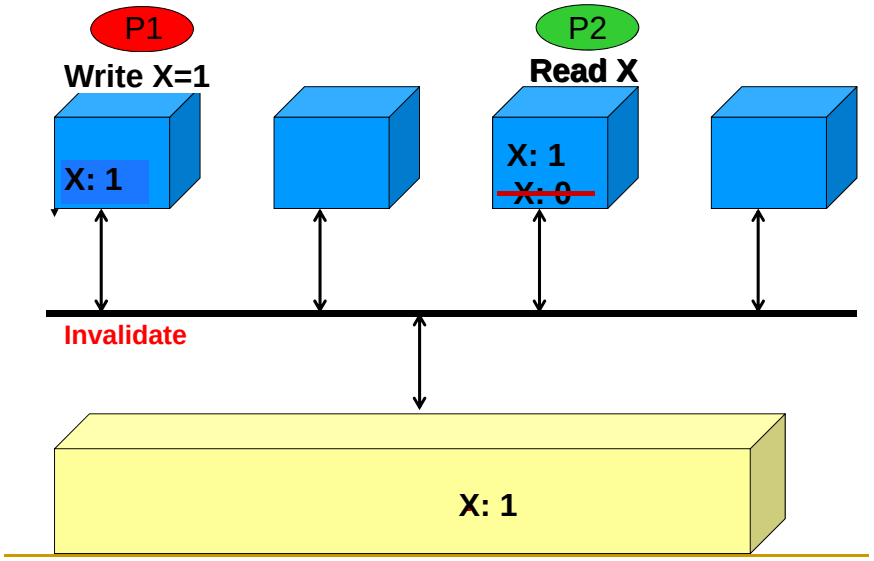
17

Example: Write Once Protocol

- Assumption: shared bus interconnect where all cache controllers monitor all bus activity
  - Called snooping
- There is only one operation through bus at a time; cache controllers can be built to take corrective action and enforce coherence in caches
  - Corrective action could involve updating or invalidating a cache block

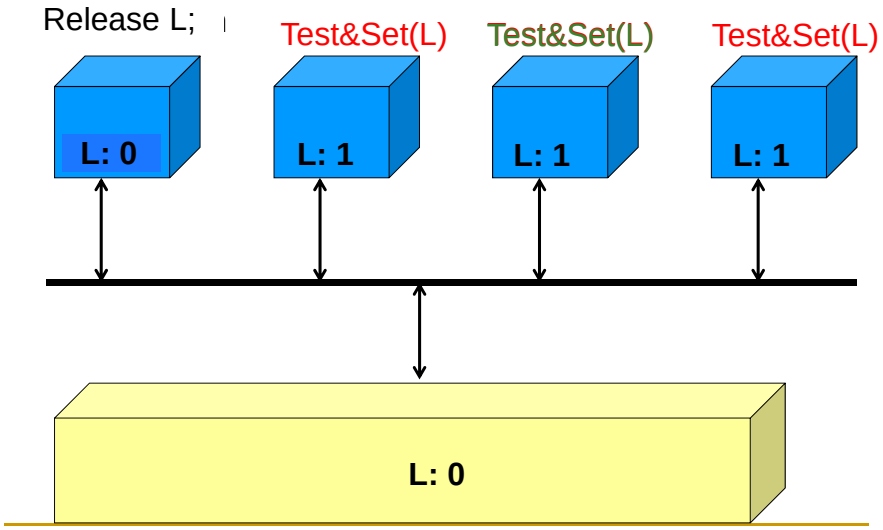
18

Invalidation Based Cache Coherence



19

Snoopy Cache Coherence and Locks



20

Space of Parallel Computing

Programming Models

- What programmer uses in coding applns.
- Specifies synch. And communication.
- Programming Models:
  - Shared address space
  - Message passing
  - Data parallel

Parallel Architecture

- *Shared Memory*
  - *Centralized shared memory (UMA)*
  - *Distributed Shared Memory (NUMA)*
- *Distributed Memory*
  - *A.k.a. Message passing*
  - *E.g., Clusters*

21

Memory Consistency Model

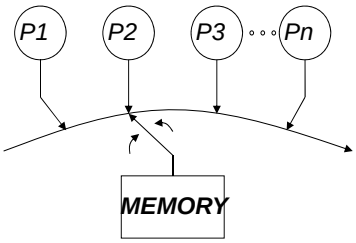
- Memory consistency model
  - Order in which memory operations will appear to execute
    - ⇒ What value can a read return?
  - Contract between appln. software and system.

⇒ Affects ease-of-programming and performance

22

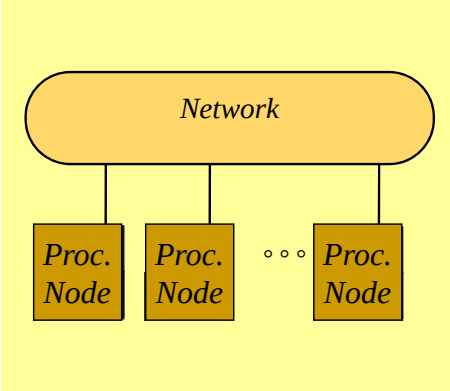
Implicit Memory Model

- Sequential consistency (SC) [Lamport]
  - Result of an execution appears as if
    - Operations from different processors executed in some sequential (interleaved) order
    - Memory operations of each process in program order



23

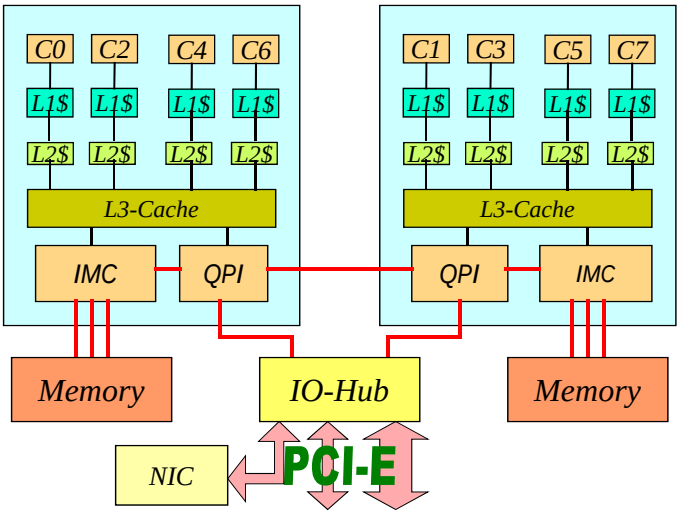
Distributed Memory Architecture



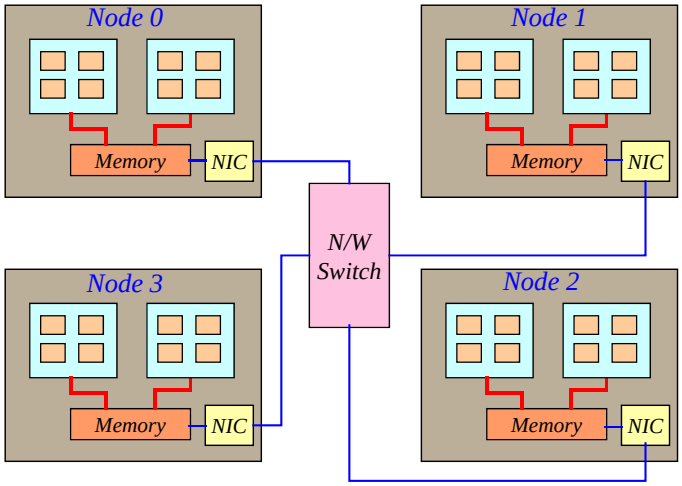
- Message Passing Architecture
  - Memory is private to each node
  - Processes communicate by messages

24

NUMA Architecture



Using MultiCores to Build Cluster



Using MultiCores to Build Cluster