Lecture 2
Understanding Parallel Computers

- Recap: Flynn's taxonomy
- Communication abstractions
- Shared Memory Multiprocessors
- Distributed Memory Multiprocessors
- Network Topologies

1

Recap: Flynn's taxonomy

2

Flynn's taxonomy (1966)

- {Single, Multiple} {Instructions, Data}

SISD Single Instruction, Single Data	SIMD Single Instruction, Multiple Data
MISD Multiple Instruction, Single Data	MIMD Multiple Instruction, Multiple Data

3

SISD

UNIVAC1

IBM360

Dell Laptop

CDC7600

PDP1

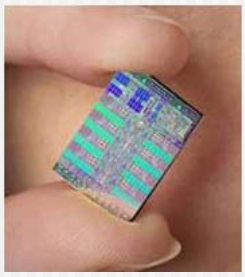
4

SIMD

ILLIAC IV

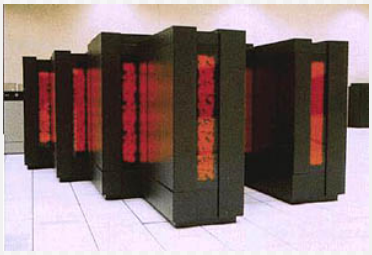
MasPar

Cell Processor (GPU)



Cray 1

CM-2

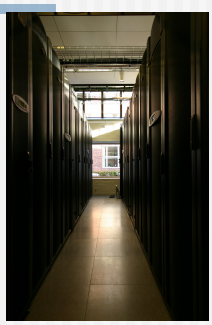


5

MIMD

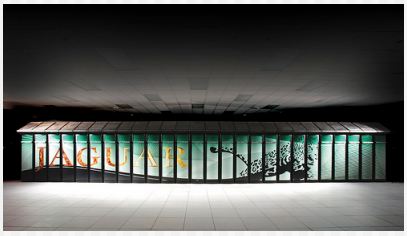
IBM Power5

AMD Opteron



IBM BlueGene/P

Cray XT5



6

Communication Architecture

7

A parallel computer is

"a collection of processing elements that **communicate** and **cooperate** to solve large problems fast"

(Almasi and Gottlieb 1989)

8

Communication Architecture

- Defines basic communication and synchronization operations
- Addresses the organizational structures that realize these operations

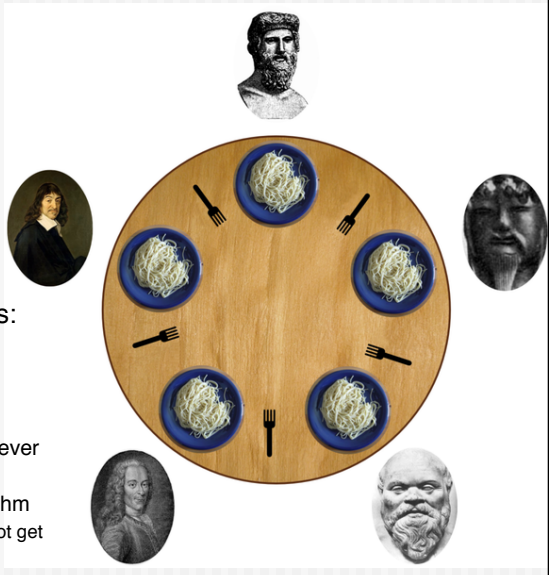
- Communication: exchange of data between processing units
- Synchronization: coordinate parallel activities

9

Synchronization: Dining Philosophers

- Algorithm:
  - Think
  - Take left fork
  - Take right fork
  - Eat
  - Release right fork
  - Release left fork

- Synchronization Problems:
  - Dead lock:
    - All have left fork
  - Starvation:
    - One philosopher can never get hold of two forks
    - Only in modified algorithm
      - Release fork if cannot get hold of second fork



10

Common Synchronization Patterns

- Barrier
  - Hold activities until all processes have reached the same point
- Semaphore
  - Finite resources
  - Two operations: P - wait for free resource and lock it; V - release resource
- Mutex
  - Only one process can access a shared resource
- Events
  - Process waits until notified by another process

11

Typical Communication Architectures

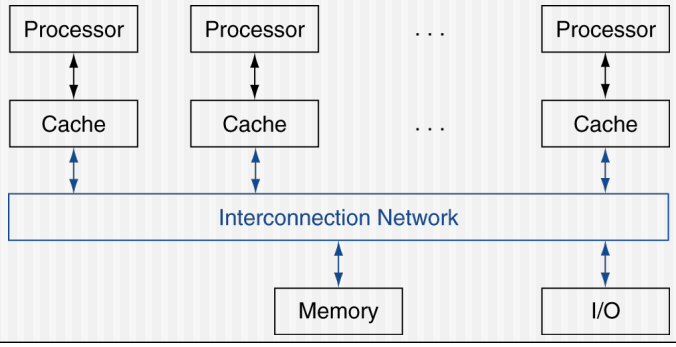
- Shared Memory
- Distributed Memory

12

Shared Memory

Shared Memory Multiprocessor

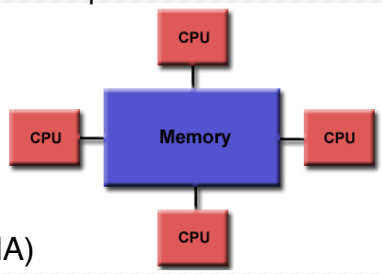
- Hardware provides single physical address space for all processors
- Global physical address space and symmetric access to all of main memory (*symmetric multiprocessor - SMP*)
- All processors and memory modules are attached to the same interconnect (bus or switched network)



Differences in Memory Access

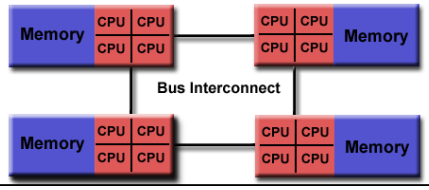
- Uniform Memory Access (UMA)

  Memory access takes about the same time independent of data location and requesting processor



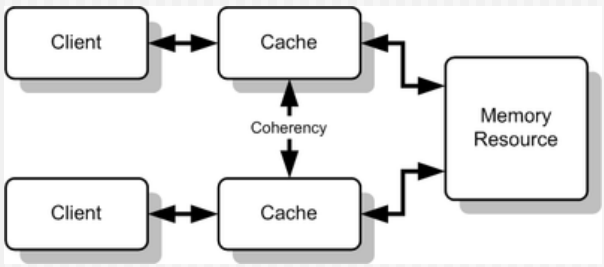
- Nonuniform memory access (NUMA)

  Memory access can differ depending on where the data is located and which processor requests the data



Cache coherence

- While main memory is shared, caches are local to individual processors
- Client B's cache might have old data since updates in client A's cache are not yet propagated
- Different cache coherency protocols to avoid this problem
  - Subject of subsequent lectures



16

Synchronization

- Access to shared data needs to be protected
  - Mutual exclusion (mutex)
  - Point-to-point events
  - Global event synchronization (barrier)
- Generic three steps:
  1. Wait for lock
  2. Acquire lock
  3. Release lock

17

Waiting Algorithms

- Busy-waiting
  - Process spins in a loop that repeatedly tests for a variable to change its value
- Blocking
  - Process suspends (blocks) and will be awakened and made ready to run again when the release occurs
- Trade-off:
  - Blocking has higher overhead (suspend resume) but frees processor for other tasks
  - Busy-waiting favorable for short waiting times

18

SMP Pros and Cons

- **Advantages:**
  - Global address space provides a user-friendly programming perspective to memory
  - Data sharing between tasks is both fast and uniform due to the proximity of memory to CPUs
- **Disadvantages:**
  - Primary disadvantage is the lack of scalability between memory and CPUs. Adding more CPUs can geometrically increases traffic on the shared memory-CPU path, and for cache coherent systems, geometrically increase traffic associated with cache/ memory management.
  - Programmer responsibility for synchronization constructs that insure "correct" access of global memory.
  - Expense: it becomes increasingly difficult and expensive to design and produce shared memory machines with ever increasing numbers of processors.

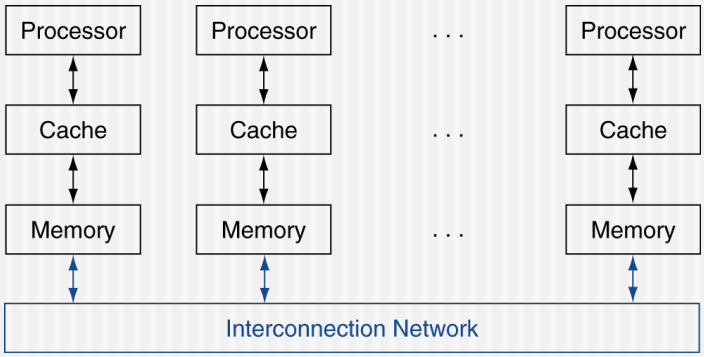
19

Distributed Memory Multiprocessors

20

DMMPs

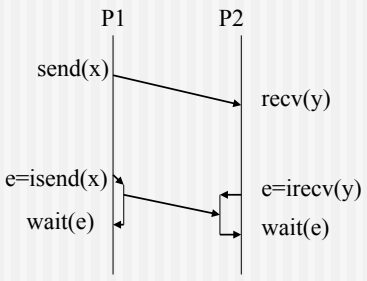
- Each processor has private physical address space
  - No cache coherence problem
- Hardware sends/receives messages between processors
  - Message passing



21

Synchronization

- Synchronization via exchange of messages
- Synchronous communication
  - Sender/receiver wait until data has been sent/received
- Asynchronous communication
  - Sender/receiver can proceed after sending/receiving has been initiated
- Higher level concepts (barriers, semaphores, …) can be constructed using send/recv primitives
  - Message passing libraries typically provide them



22

DMMPs Pros and Cons

- **Advantages:**
  - Memory is scalable with number of processors. Increase the number of processors and the size of memory increases proportionately.
  - Each processor can rapidly access its own memory without interference and without the overhead incurred with trying to maintain cache coherency.
  - Cost effectiveness: can use commodity, off-the-shelf processors and networking.
- **Disadvantages:**
  - The programmer is responsible for many of the details associated with data communication between processors.
  - It may be difficult to map existing data structures, based on global memory, to this memory organization.
  - Non-uniform memory access (NUMA) times
  - Administration and software overhead (essentially N systems vs. 1 SMP)

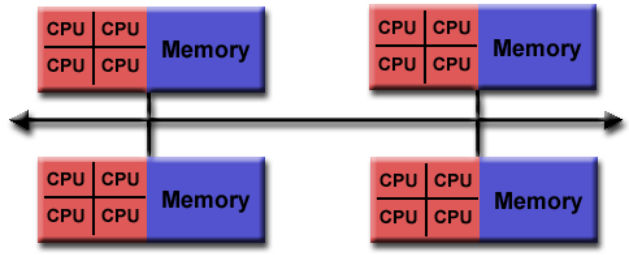
23

Hybrid Approaches

24

Combining SMPs and DMMPs

- Today, DMMPs are typically built with SMPs as building blocks
  - E.g. Cray XE6 has two CPUs with 12 cores each per DMMP node
  - Soon systems with more CPUs and many more cores will appear
- Combine advantages and disadvantages from both categories
  - Programming is more complicated due to the combination of several different memory organizations that require different treatment



Network Topologies

26

The Role of the Network

- The overall performance of DMMPs depends critically on the performance of the network used to connect the individual nodes
  - How fast can messages be transmitted and how much data can be exchanged
  - Also applies to networked SMPs
- **Latency**: time between *start* of packet *transmission* to the *start* of packet *reception* (but typically measured as round-trip of zero sized messages
- **Bandwidth**: how much data can be transmitted over the network (bit/s)

27

Different Technologies

- Ethernet
- Myrinet
- Infiniband
- Proprietary networks
  - Cray Gemini
  - IBM BlueGene
- Differ in bandwidth and latency but most notably in sustained performance through e.g. MPI

28

Network Topologies

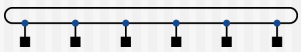
- Networks can be arranged in a number of ways
- Typical design goals is to balance performance and cost
- Factors in addition to latency and bandwidth:
  - Fault tolerance
  - Power consumption
  - Number of switches
  - Number of links
  - Cable length
- Additional considerations:
  - Total Network Bandwidth
    - Bandwidth of each link multiplied by the number of links
  - Bisection Bandwidth
    - Worst case bandwidth if nodes are divided into two disjoint sets

29

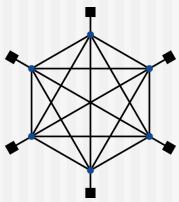
Common Topologies

- Bus
  - Total bandwidth is 2 times bandwidth of the link
  - Bisection bandwidth is bandwidth of the link

- Ring
  - TB: P times the bw of one link
  - BS: 2 times the bw of one link



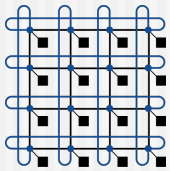
- Fully connected network
  - TB: P x (P-1)/2
  - BS: $(P/2)^2$



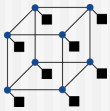
30

Common Topologies Cont'd

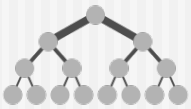
- Mesh
  - Typically 2D or 3D



- N-cube
  - Hypercube
  - 2^n nodes



- Fat tree
  - Common in Infiniband based systems



31

Summary and Outlook

- Synchronization and Communication
- Common parallel computer organizations
- Common network topologies

- Lecture 3: Performance, Scaling, Benchmarking

32

Further Readings

- **Introduction to Parallel Computing,** https://computing.llnl.gov/tutorials/parallel_comp/
- **Computer Organization and Design - The Hardware/Software Interface,** 4th Edition, David A. Patterson and John L. Hennessy
  - Chapter 7 (Multicore, Multiprocessors, and Clusters)
  - Chapter 2.11 (Parallelism and Instructions: Synchronization)

33

Exercises

4. What is the sequence of events that lead up to the problem where none of the philosophers in the dining philosophers problem ever eats (i.e. dead lock)
5. How can we avoid the problem of a dead lock? Can the solution introduce starvation?
6. Suppose a single shared memory processor has 20GB of main memory, five clustered computers each have 4 GB, and the OS occupies 1 GB. How much more space is there for users with shared memory?
7. Assume 2 communicating processors issuing synchronous and asynchronous communication requests (send/recv). Draw up sequences of events in a timeline that can lead to dead locks.
8. Discuss the advantages/disadvantages of hybrid architectures wrt network topology.

34