COMS4040A & COMS7045A: High Performance Computing & Scientific Data Management Multiprocessors

Hairong Wang

School of Computer Science & Applied Mathematics
University of the Witwatersrand, Johannesburg

Contents

1. Key properties of HPC architecture
2. Parallel computers
    - Shared memory parallel computers
    - Distributed memory parallel computers
    - Networks
3. References

WITS
UNIVERSITY

Outline

1. Key properties of HPC architecture

2. Parallel computers
   - Shared memory parallel computers
   - Distributed memory parallel computers
   - Networks

3. References

Key properties of HPC architecture

Three key properties of an HPC architecture that determine delivered performance:

- The speed of the components comprising the system
- The parallelism or number of components that can operate concurrently doing many things simultaneously
- The efficiency of use of those components in the degree of utilization achieved

Read further on properties of HPC architecture in `Lec3_Sec1_note.pdf` provided.

Outline

1. Key properties of HPC architecture

2. Parallel computers
   - Shared memory parallel computers
   - Distributed memory parallel computers
   - Networks

3. References

Taxonomy of parallel computing paradigms

- **SIMD** Single instruction multiple data. A single instruction stream, either on a single processor (core) or on multiple compute elements, provides parallelism by operating on multiple data streams concurrently.
- **MIMD** Multiple instruction multiple data. Multiple instruction streams on multiple processors (cores) operate on different data items concurrently. The shared memory and distributed memory parallel computers described in this lecture are typical examples for the MIMD paradigm.
- **SPMD:** (Single program, multiple data stream) Instead of issuing a single instruction at a time, it issues a coarse-grained function or procedure, instead of a single instruction as in SIMD, that is performed by all the processing units. Depending on the varying levels of coarse grained tasks, most modern computing structures, such as GPU, and multicore processors, have this capability.

Outline

1. Key properties of HPC architecture

2. Parallel computers
   - Shared memory parallel computers
   - Distributed memory parallel computers
   - Networks

3. References

Shared memory parallel computers

- A shared memory parallel computer is a system in which a number of CPUs work on a common, shared physical address space.
- There are two varieties of shared memory systems that have different performance characteristics in terms of main memory access:
  - Uniform memory access (UMA): In a UMA system, the latency and bandwidth are the same for all compute elements and all memory locations. UMA is also called symmetric multiprocessing (SMP). A parallel computer with a single CPU chip which consists of multiple cores is a UMA system.
  - Cache-coherent non-uniform memory access (ccNUMA): Memory is physically distributed but logically shared. The physical layout of such systems is quite similar to the distributed-memory case, but network logic makes the aggregated memory of the whole system appear as one single address space.

Shared-memory parallel computers cont.

- Cache coherence: Cache coherence mechanisms are required in all cache-based multiprocessor systems, including both UMA and ccNUMA.
- Cache coherence — Cache lines that contain the same memory sections could exist in several processor caches. Cache coherence protocols ensure a consistent view of memory.

Shared-memory parallel computers – UMA

- **UMA:**
- The chipset enforces cache coherence and also connects cores to the memory.

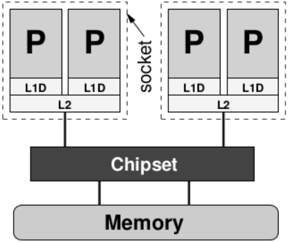


Figure: An example of UMA system

Shared-memory parallel computers – UMA - cont.

- A common problem with UMA systems is that bandwidth bottlenecks when the number of sockets (or FSBs) increases.
- If a point-to-point connections between CPUs and memory modules are built, then it can become expensive to build.
- If a common bus is used, then it can only be used to transfer data to one CPU at a time.
- Hence UMA is limited in terms of scalability to larger system.

Shared-memory parallel computers – ccNUMA

- ccNUMA is a collection of locality domains (LDs), where a LD is a set of processor cores together with locally connected memory.
- Multiple LDs are connected via a coherent interconnect.
- An LD is similar to a UMA.
- The intersocket connection is capable of handling cache-coherent memory accesses.

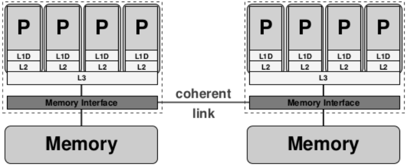


Figure: An example of ccNUMA system with two LDs

Shared-memory parallel computers – ccNUMA cont.

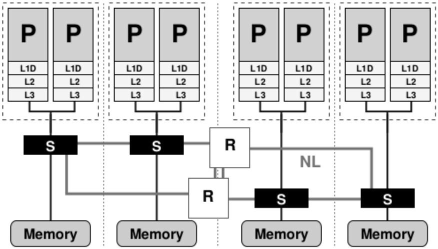


Figure: An example of ccNUMA system with four LDs

- The LDs are connected via a routed NUMALink (NL) network using routers (R).
- Each processor socket is connected to a communication interface (S).
- The NL network relies on Rs to switch connections for nonlocal access.

Shared-memory parallel computers – ccNUMA cont.

- In ccNUMA systems, network connections need to match the local memory access performance in order to achieve good performance.
- Another issue is contention problem, where two processors from different LDs access memory in the same LD. In this case, there is a contention of memory bandwidth.

Outline

1. Key properties of HPC architecture

2. Parallel computers
   - Shared memory parallel computers
   - Distributed memory parallel computers
   - Networks

3. References

Distributed memory

- In a distributed memory parallel computer, each processor is connected to exclusive local memory.

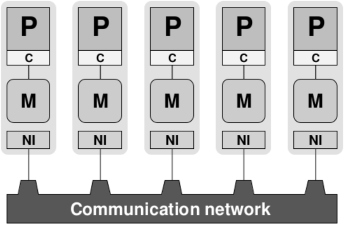


Figure: Simplified programmer's view, or programming model of a distributed memory parallel computer

Distributed memory cont.

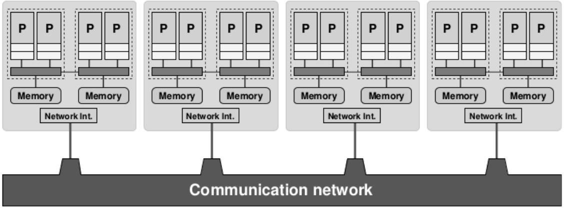


Figure: A typical hybrid system with shared memory nodes.

- Figure shows a more realistic (distributed memory) parallel computer which is of a hybrid system shared memory and distributed memory systems.

Outline

1. Key properties of HPC architecture

2. Parallel computers
   - Shared memory parallel computers
   - Distributed memory parallel computers
   - Networks

3. References

Basic performance characteristics of networks

- The simplest and cheapest network solution is Gigabit Ethernet, which suffices for many throughput applications but is far too slow for parallel programs with any need for fast communication.
- **Point-to-point connections:** The communication characteristics of a single point-to-point connection can be described by a simple model with *B* - network (maximum) bandwidth in MB/sec, N - message size in byes, T_ℓ - latency.

$$T = T_\ell + \frac{N}{B} \tag{1}$$

- The effective bandwidth:

$$B_{eff} = \frac{N}{T_\ell + \frac{N}{B}} \tag{2}$$

Basic performance characteristics of networks cont.

- To measure the latency and effective bandwidth, the pingpong benchmark is often used. The code sends a message of size *N* bytes back and forth between two processes running on different processors.

```
myID = get_process_ID()
if (myID == 0)
  targetID = 1
  S = get_walltime()
  send_message(buffer,N,targetID)
  recv_message(buffer,N,targetID)
  E = get_walltime()
  MBYTES = 2*N/(E-S)/1.e6 %data transfer rate
  TIME = (E-S)/2*1.e6 % transfer time in microsecs
else
  targetID = 0
  recv_message(buffer,N,targetID)
  send_message(buffer,N,targetID)
```

Basic performance characteristics of networks cont.

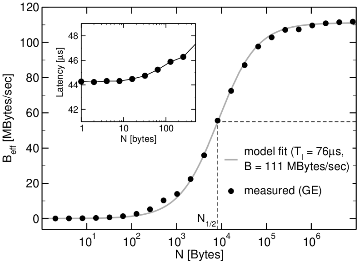


Figure: Fit of the model for effective bandwidth to data measured over GigE.

- Figure shows a benchmarking over Gigabit Ethernet. The model parameters (latency and bandwidth) are fitted to real measurements. The latency can be measured setting $N = 0$.

Basic performance characteristics of networks cont.

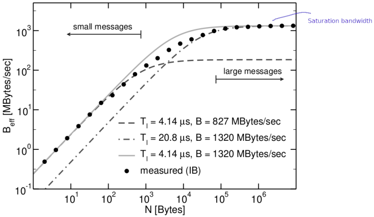


Figure: Fit of the model for effective bandwidth to data measured over DDR Infiniband.

- The dotted-dash curve is obtained by restricting the fit to the large message size regime,
- The dashed curve is obtained by restricting the fit to the small-message size regime.
- The former yields a good estimate of B, while the latter of T_ℓ.
- To quantify this problem $N_{1/2}$ is often used. $N_{1/2}$ is the problem size where $B_{eff} = B/2$. For model (2), $N_{1/2} = BT_\ell$.

Basic performance characteristics of networks cont.

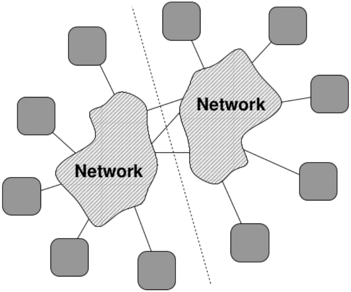
- What is the effect like when we increase the maximum bandwidth by a factor of β, i.e., $B_{new} = \beta B$? At message size N, the improvement in effective bandwidth is

$$
\begin{aligned}
\frac{B_{eff}(\beta B, T_\ell)}{B_{eff}(B, T_\ell)} &= \left(\frac{N}{T_\ell + \frac{N}{\beta B}}\right) / \left(\frac{N}{T_\ell + \frac{N}{B}}\right) \\
&= \left(\frac{T_\ell + \frac{N}{B}}{T_\ell + \frac{N}{\beta B}}\right) \\
&= \frac{T_\ell + \frac{N}{N_{1/2}/T_\ell}}{T_\ell + \frac{N}{\beta N_{1/2}/T_\ell}} \\
&= \frac{1 + N/N_{1/2}}{1 + N/(\beta N_{1/2})}
\end{aligned}
\tag{3}
$$

- When $N = N_{1/2}$ and $\beta = 2$, the gain is only 33%.

Bisection bandwidth

- Bisection width: The minimal number of connections cut when splitting the system into two equal-sized parts.
- Bisection bandwidth: The sum of the bandwidths of the minimal number of connections cut when splitting the system into two equal-sized parts.
- In hybrid/hierarchical systems, we can use available bandwidth per core - bisection bandwidth divided by the overall number of compute nodes.



Bus

- A bus is a shared medium that can be used by exactly one communication device at a time. They are easy to implement.
- A bus is blocking - Only one communication device can use the bus at one time, and others have to be blocked until the bus becomes free.
- Buses are susceptible to failures.
- The bisection width of a bus is 1.

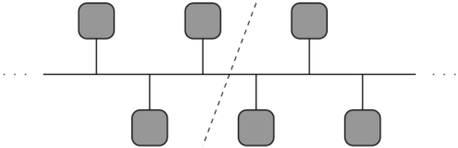


Figure: A bus network

Fully-connected crossbar network

- Switches are used to subdivide a network into groups. The communication happens via the switches.
- Switch: consists of input and output ports.
- A single switch can either support a fully nonblocking operation, which means that all pairs of ports can use their full bandwidth concurrently, or it can have partly or completely, a bus like design where bandwidth is limited.

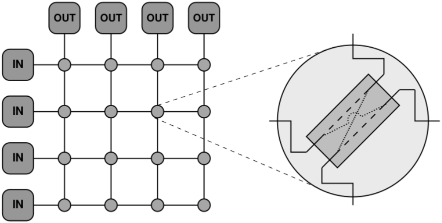


Figure: A fully-blocking 2D crossbar network. Each circle represents a possible connection between two devices from the ‘IN’ and ‘OUT’ groups, respectively, and is implemented as a 2×2 switching element. The whole circuit can act as a four-port nonblocking switch.

Switched and fat-tree networks

- Switches can be combined and cascaded to form a fat tree switch hierarchy;
- The diameter of the network is the maximum number of hops required to connect two arbitrary devices.

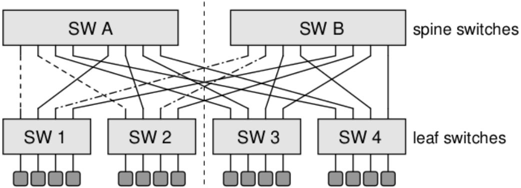


Figure: A non-blocking fattree network

Switched and fat-tree networks cont.

- Figure below shows a system that has thinner connections towards the root of the tree, in which only 4 pairs of compute elements are allowed to communicate at one time, and the rest will be blocked.

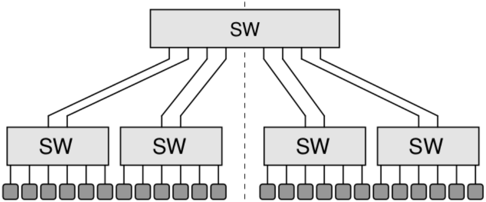


Figure: A fattree network with a bottleneck

Switched and fat-tree networks cont.

- Even in a fully nonblocking fat-tree switch hierarchy, not all possible combinations of $N/2$ point-to-point connections allow collision free operation under static routing (or ‘hardwired’).

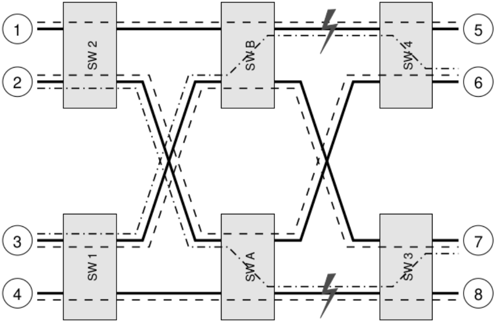


Figure: A fat-tree network

Mesh networks

- In mesh network, there are no direct connections between elements that are not next neighbors.

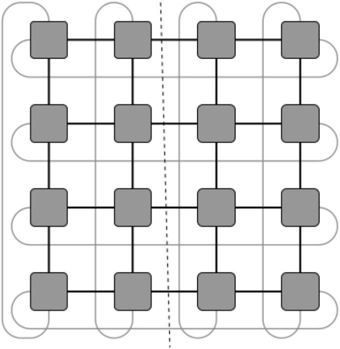


Figure: 2D torus

Hybrids

- **Hybrids:** If a network is a combination of at least two of the topologies described above, it is called hybrid.

Outline

1. Key properties of HPC architecture

2. Parallel computers
   - Shared memory parallel computers
   - Distributed memory parallel computers
   - Networks

3. References

References

- Chapter 2 of *High Performance Computing: Modern Systems and Practices*, by Thomas Sterling, Matthew Anderson, and Maciej Brodowicz, Morgan Kaufmann Publishers, 2018.
- Chapter 4 of *Introduction to High Performance Computing for Scientists and Engineers*, by Georg Hager and Gerard Wellein, CRC Press, Inc, 2010. ISBN:978-1-4398-1192-4.