
Tools and techniques for achieving optimal scaling on a large cluster of shared-memory nodes

Mark Bull and Gavin Pringle
EPCC, The University of Edinburgh, U.K.



Introduction



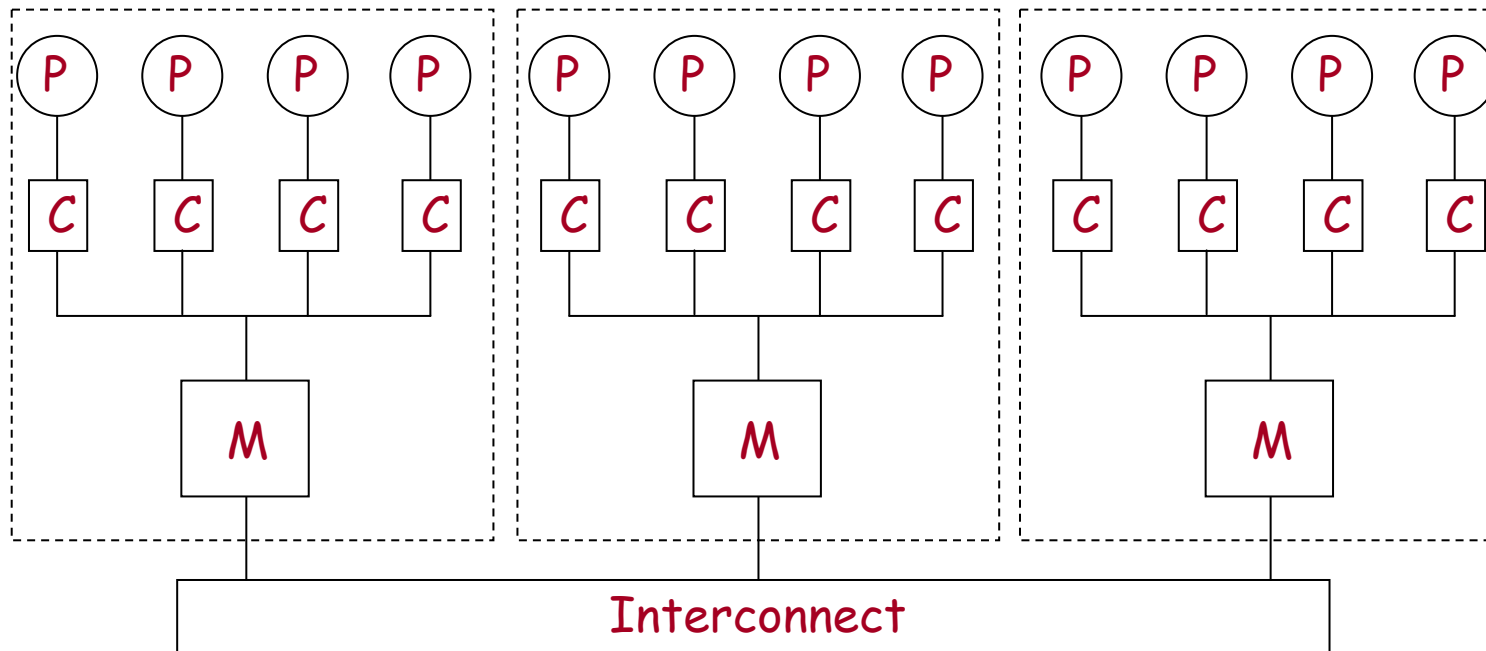
- In recent years there has been a trend towards large clusters of shared memory nodes
- For example
 - 3 of top 10 supercomputers in the world are SMP clusters
 - Reported in the top500 list, June 2006
 - Also most of the systems operated by DEISA partners

- **As these machines have become more prominent**
 - It has become essential to program these efficiently
- **Our experience on SMP clusters suggests**
 - Many applications do not consider the clustered nature of these machines
 - Can have significant influence on performance and scaling

- Hence our aim is to:
 - Provide an overview of the essential features of this type of architecture
 - Highlight the main techniques for achieving optimal performance and scaling

- Parallel systems traditionally fall into two categories of architecture:
 - Distributed memory systems
 - Separate nodes have separate address space
 - E.g. CrayT3E
 - Shared memory systems
 - Single address space shared between processors
 - E.g. SGI Altix, Sun E15K
- SMP clusters combines both types of architecture
 - Clusters of shared memory nodes
 - E.g. Sun Fire E2900 cluster, IBM p690+ cluster

- Distributed memory system
 - Separate nodes have separate address spaces



- Single address space within each node
 - A traditional shared memory multiprocessor (SMP)

- Results from three SMP clusters will be used throughout this tutorial:
- IBM p690+ Cluster
- Sun Fire SMP Cluster
- AlphaServer SC cluster

- 50 IBM p690+ shared memory nodes
- Each p690+ node
 - 32 POWER 4+ 1.7 GHz processors and 32Gb main memory
- Supported by the University of Edinburgh (EPCC), Daresbury Laboratory and IBM
- Since upgraded to Power5...



- 16 Sun Fire 6800 nodes
 - 24 900Mhz UltraSparc III processors with 24 GB memory
- 4 Sun Fire 15K nodes
 - Each with 72 900MHz UltraSparc III processors with 144 GB memory
- Hosted and run by Aachen University, Germany

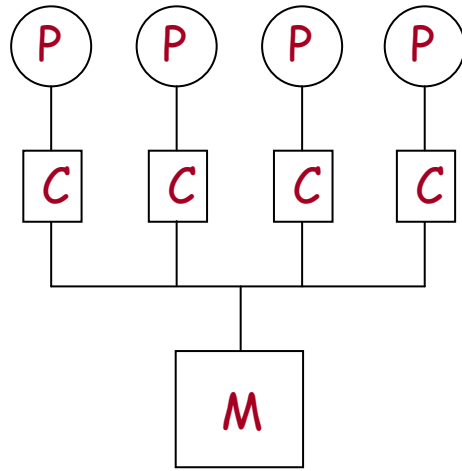


- 127 HP AlphaServer SC ES45 nodes

- 4 1GHz ev68 (Alpha) processors
- Between 4GB and 16GB of memory per node



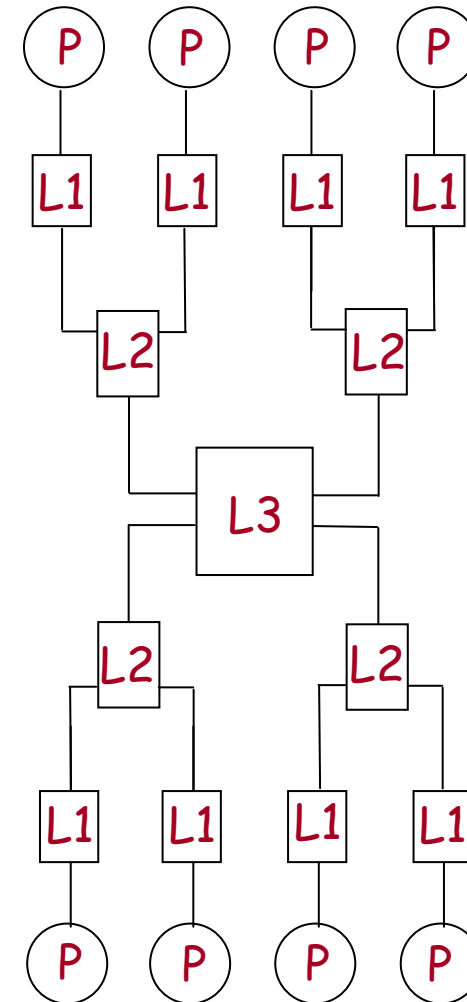
- Run by the Australian Partnership for Advanced Computing National Facility (APAC)



- These can vary in size and complexity
 - E.g. 4 processor Intel Itanium2 Tiger4, 32 processor p690+

- The memory hierarchy can also be quite complex
 - E.g. L1, L2, L3 and main memory shared between varying numbers of processors
 - Memory access speeds are influenced by this complexity

- Each p690+ node is complex
 - 4 Multi-chip modules, 16 chips, 32 processors
- L1 cache
 - instruction cache of 128 kbytes
 - data cache of 64 kbytes
- Power4+ chips
 - 2 independent CPUs
 - shared L2 cache - 1.5 Mbytes
- Multi-Chip Modules (MCM)
 - Consist of four Power4+ chips
 - All share L3 cache - 128 Mbytes
- Main memory - 32 Gbytes



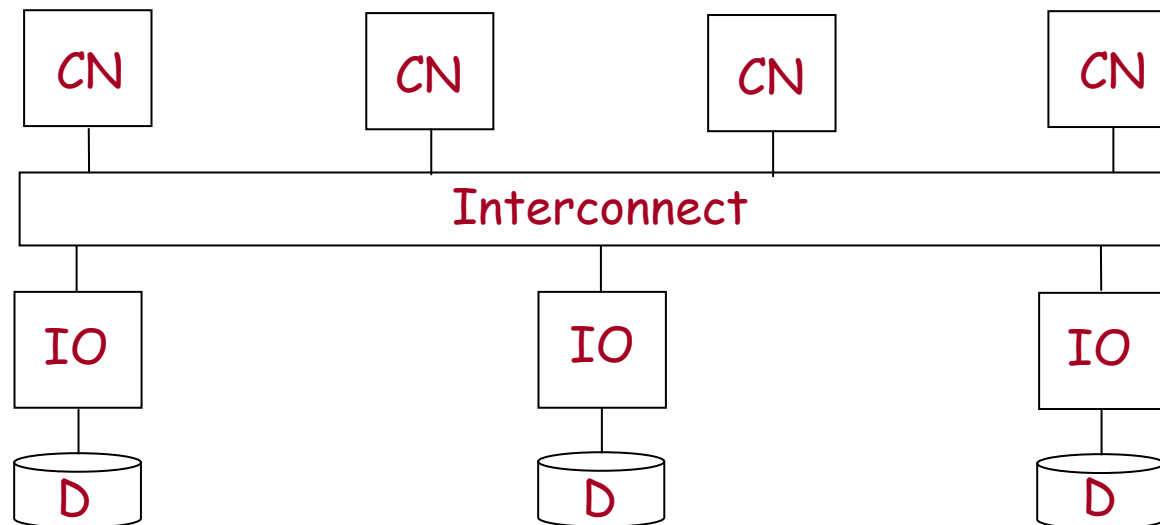
- UltraSparc III chip
 - L1 cache 64Kbytes data, 32 Kb instruction
 - L2 (off chip) 8MB data and instruction
- CPU board
 - Consists of 4 processors
 - L2 cache and local interleaved memory
- Sun Fire 6800
 - 6 boards coupled with a crossbar
- Sun Fire 15K
 - 18 boards coupled with a crossbar

- Each Es45 node is relatively simple
 - 4 ev68 (Alpha) processors
- Cache
 - L1 cache (on chip): 64 kbytes (I) + 64 Kbytes (D)
 - L2 cache (off chip): 8 Mbytes per cpu
- Main memory
 - Between 4 and 16GB of RAM (currently 36 8GB nodes and 4 16GB nodes)

- These can vary substantially on different systems
 - Different topologies
 - Different mechanisms for handling congestion
 - Different mechanisms for dealing with message routing
- These in turn can influence performance
 - Wide variety of latency and bandwidth characteristics

- SMP clusters combine the benefits of both shared and distributed memory systems
- Communication within a node is fast
 - Processors can access global memory via load operations within a node
 - However SMPs do not scale to large processor numbers (due to contention on the bus)
- Distributed memory systems
 - Scale to larger numbers of processors
 - However communication between nodes is slower

- IO architectures can have significant influence on speed of data access
 - Often have separate IO nodes connected to the same interconnect as compute nodes
 - These are connected to storage device(s)
 - Each compute node may also have its own local disk



- Can be programmed using a range of parallel programming paradigms
 - Distributed memory models can be used across the whole system
 - Shared memory models can be used within a node
 - Combination of shared and distributed memory models can also be used (mixed mode programming)
 - Can use MPI across the whole system
 - Can use OpenMP or MPI within a node
 - Can use a mixed MPI / OpenMP model

- These features can significantly influence performance
 - For example, communication between nodes usually slower than within a node
- Can be important to consider architecture features when optimizing your code
- In this tutorial we aim
 - To highlight the main techniques for achieving optimal performance and scaling on these systems

- **Lecture 2**
 - Communication optimization
- **Lecture 3**
 - Decomposition strategies
 - IO optimization
- **Lecture 4**
 - Mixed programming models
- **Lecture 5**
 - Performance analysis tools

- *Sourcebook of Parallel Computing, J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, Morgan Kaufmann, 2003*
- *High Performance Computing: Clusters, Constellations, MPPs, and Future Directions, J. Dongarra, T. Sterling, H. Simon, E. Strohmaier, 2003.*