34 Cluster Interconnect Technologies

If you have not read the book (*Performance Assurance for IT Systems*) check the introduction to "More Tasters" on the web site http://www.b.king.dsl.pipex.com/ to understand the scope and objectives of tasters.

The objective of this taster is to introduce the major cluster interconnect technologies: Gigabit; Myrinet; Quadrics; SCI; and Infiniband. This topic is a necessary precursor to the taster *Uses Of Cluster Interconnects: Distributed Shared Memory Systems, Computing Clusters, and Cluster Filesystems / Databases* (target date for publishing - September 2005). It may also be read in conjunction with the *Back-end Server Clusters* taster.

34.1 Starting Terminology

ASIC (Application Specific Integrated Circuit) chips are specialised components that have been developed to support a particular task.

microsecond is one millionth of a second (usually represented as µs).

millisecond is one thousandth of a second (usually represented as ms).

MPI (Message Passing Interface) is the standard mechanism by which processors communicate in distributed memory systems and clusters.

nanosecond is one thousand millionth of a second (usually represented as ns).

NIC (Network Interface Card).

PCI (Peripheral Component Interconnect) is an established bus architecture using parallel IO that was originally developed by Intel. PCI-X (short for PCI-extended) is an enhanced, higher-performing version of PCI, which was developed to support devices with high bandwidth requirements.

PVM (Parallel Virtual Machine) is an alternative to MPI. It provides a friendlier programming interface that is both flexible and portable. The main disadvantage is that it does not perform as well.

34.2 Technology Outline

34.2.1 Background

The term "cluster" has different meanings to different people, depending on the technology under discussion and the experience of the individuals concerned. To be clear: this taster is concerned with groups of servers that have an application requirement to communicate with each other quickly (low latency) and at high rates (large bandwidth). Obvious examples are scientific applications that use parallel processing techniques, loosely-coupled systems (sometimes called distributed shared memory systems) such as Tandem, databases that sit across multiple server nodes, e.g. Oracle RAC (Real Application Cluster), and optionally large-scale cluster filesystems. Clusters whose requirements are primarily limited to

supporting relatively free-standing server nodes, e.g. to provide simple failover, are not the focus of discussion.

Interconnect technology arguably commenced with the advent of the supercomputer, typically a large numbers of processor nodes connected together, possibly in a single cabinet. Some of the technologies that are discussed in this taster had their origins in supercomputing. A key driver in the scientific area in recent years has been to reduce the cost of high performance computing by deploying commodity components. The use of Linux-based servers and cheaper interconnects are major elements in this move. The advances in interconnect technology, coupled with the decreases in cost, has attracted interest in its use within commercial applications.

34.2.2 Hardware

The heart of the interconnect is the network, occasionally called the fabric. Various topologies can be used to connect server nodes, examples of which are shown in Figure 34-1 and 34-2.

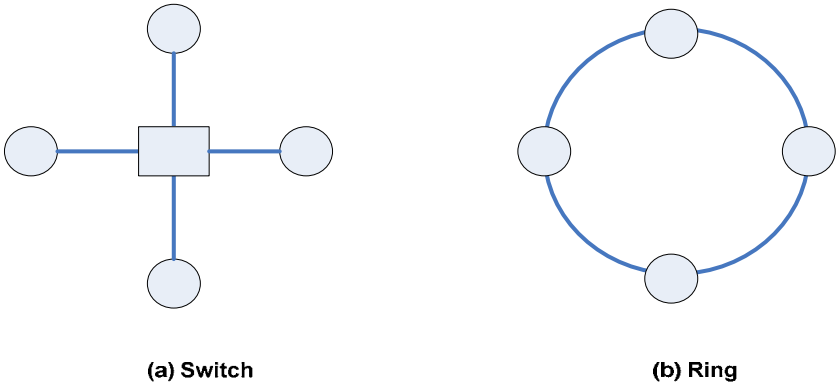

Figure 34-1 Basic Topologies (1)

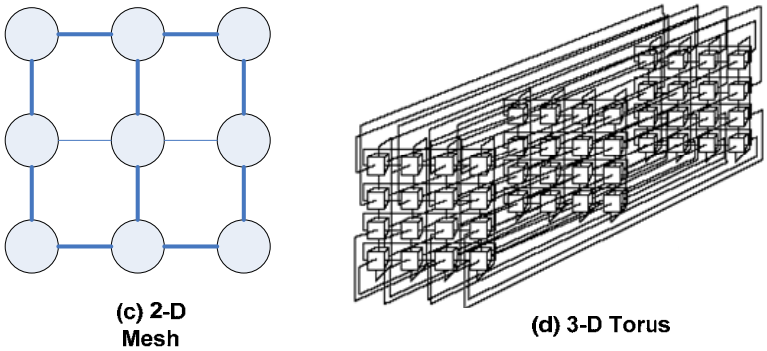

Figure 34-2 Example Topologies (2)

Note that Torus means that the ends are connected (wrap round) such that the cube has no boundary edge. Speed through the network is extremely important. The first objective must be to minimise the number of hops that are necessary to traverse the network. While this may be relatively straightforward on a system with a modest number of server nodes (say up to 24), it can be more problematic on larger systems where it is necessary to support hundreds, and possibly thousands, of server nodes. One approach is to use a “fat-tree” design, as shown

in Figure 34-3. This is a hierarchy. Packets travel through the entry ports (at the bottom of the diagram) up the hierarchy as far as necessary and back down to the destination port.

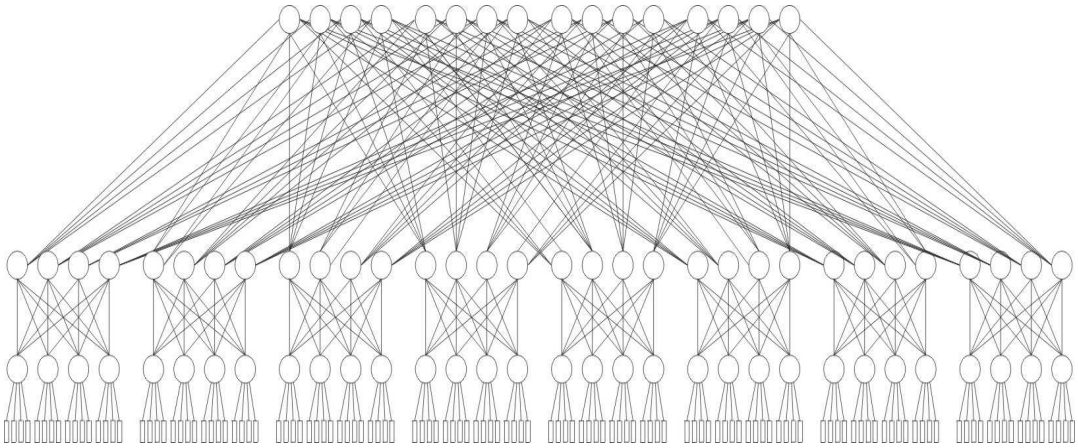

Figure 34-3 "Fat-Tree"Design

The second objective must be to reduce the time taken to traverse the network. The simple, albeit slowest, approach is the use of store and forward where, for each hop, a packet is copied from the source to destination point in its entirety before the next hop can commence. A variation on this approach is to split the packet up into smaller units. As soon as a unit has arrived at an intermediate switch, it can be forwarded without waiting for the other units to arrive. An alternative to store and forward is circuit switching. Here, the first switch establishes a path across the network. Once it is set up the data can be sent across the network without the need for buffering at each intervening point. The path is torn down once the transfer is complete. While there is some overhead in establishing and tearing down paths, overall performance should be better. A refinement is to split the packet up into elements and allow the first element to flow even before the full path has been established; this is called wormhole routing.

The connection between the server nodes and the "fabric" varies, depending on the implementation, but they can include standard Ethernet cables, proprietary copper or fibre optic cables. Proprietary solutions may consist of multiple links that are aggregated to provide higher bandwidth capabilities. Terms such as channel, bonding, and virtualisation are used to describe these multiplexing techniques. There are varying cable length restrictions, depending on the product.

At the server node, standard NICs are used on rudimentary Ethernet-based systems. Proprietary solutions are based on bespoke PCI or PCI-X cards, an example of which is shown in Figure 34-4. The cards will typically contain cache memory to buffer incoming and outgoing packets. Transfer of data between this cache and the local server's main memory can be achieved by using DMA (Direct Memory Access) channels. An on-board CPU, coupled with DMA, will help to offload much of the work that the server's CPU(s) would otherwise have to do to control this data flow. PCI-X provides a faster platform than PCI, supporting up to 8Gbps; it can operate at 133MHz/64 bit, as opposed to PCI's 66MHz/32 bit. A recent refinement to PCI-X implements DDR mode (Double Data Rate) which means that data can be transferred on both the rising and falling edges of a single clock cycle.

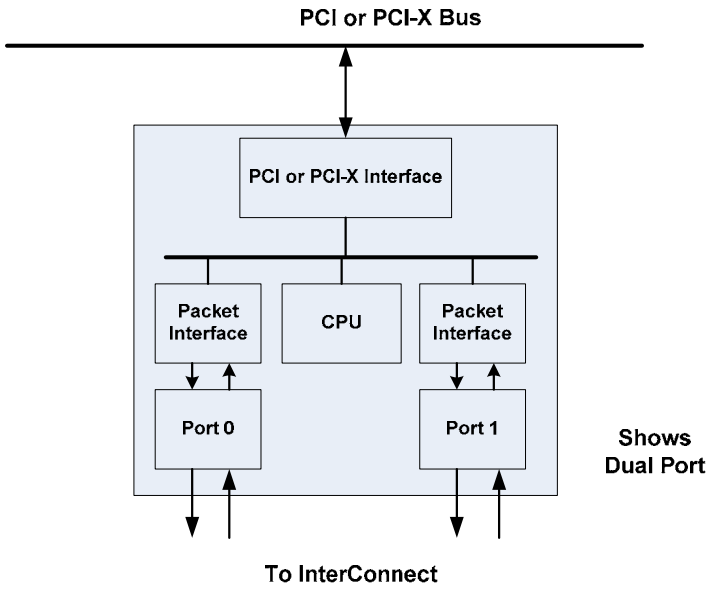

Figure 34-4 Example Host Channel Adaptor

34.2.3 Software

Each proprietary solution offers its own individual low level interface. At the next layer above, where application software is more likely to interface, they typically offer a message passing interface. There are two standard variants in this area: Parallel Virtual Machine (PVM) and Message Passing Interface (MPI). In general, MPI provides better performance while PVM provides superior portability and flexibility.

In addition to basic send and receive operations, the key facility that is offered by proprietary solutions is the ability to allow an application in one server node to have direct access to the address space in another node. This feature is usually termed RDMA (Remote Direct Memory Access). It operates in user space, thereby avoiding the need for system calls and the associated context switches and buffer copying between user and kernel. Vendors usually call this feature “OS Bypass”, while “zero copying” is another acronym that is frequently used.

Standard Ethernet solutions will be based on TCP/IP and, not having the go-faster techniques that have been mentioned above, inevitably have higher latency figures, as well as lower bandwidth throughput due to the fat nature of the protocol. Some cluster software vendors provide protocols that run directly on top of Ethernet.

34.2.4 Summary of Main Interconnect Technologies

This section provides a brief introduction to the current major technologies and their claimed bandwidth and latency performance. Beware that the quoted bandwidth figures are for large messages while the latency figures are for short messages, i.e. they are arguably best case. Note that, although the proprietary products tend to support multiple software communication mechanisms, the focus here is primarily limited to MPI. Also note that this is a rapidly changing area of technology and that therefore the information may become dated in a short period of time. Check the appropriate web sites for current details.

Gigabit Ethernet. Ethernet has always been used in clusters in some form. 100Mbps Ethernet may still be satisfactory for clusters where the amount of inter-node traffic is modest, e.g. support for a simple cluster failover system. Ethernet is popular simply because of its general use in networks over the last 20 years, plus its use of TCP/IP. Typical peak throughput on a standard gigabit implementation is in the region of 300Mbps with a latency of 80-90µs. The high latency (relative to proprietary products) is in part due to the CPU overheads of TCP/IP. Improvements can be made via the use of Jumbo Ethernet frames (9KB rather than the standard 1.5KB) and with specialised NIC ASIC chipsets. Peak throughput of 900Mbps and a latency of 30-40µs are claimed with these techniques. An alternative approach is to run proprietary cluster protocols directly over Ethernet rather than TCP/IP, e.g. LLT and GAB are used in the Veritas Cluster Server product. The other key issue is the use of standard Ethernet switches. They used to limit the number of nodes to 24 unless more expensive switches were used. Part of the cost was attributable to the provision of layer 3 (and above) routing features which are not required by cluster interconnects. However, the prices of switches are coming down and larger networks are now feasible (up to 480 ports). 10 Gigabit Ethernet has recently appeared and potentially holds out some promise, but it is arguably not cost-effective at the moment.

Myrinet. Myricom (founded in 1994) originally developed interconnect technology for massively parallel processor systems (MPPs). Current switch topologies support anything from 8, up to a claimed 1000+ server nodes. Small switches, up to 32 ports, employ a crossbar. Larger infrastructures typically employ a fat-tree design of small switches. Fibre optic cables provide the host links, each running at 2Gbps (bi-directional). A dual port card can virtualise the two ports, providing a 4Gbps channel. The host network card is based on 133MHz/64bit PCI-X. It has a programmable processor that can access host server memory via a DMA controller. As with all proprietary solutions, messages are sent and received without the need for system calls; this is usually termed OS bypass in the literature. GM is the low level message passing interface. Peak throughput using a dual port card is ~3.6Gbps per link with MPI latency (using GM) of 6-7µs. MX, an alternative low level software interface with a promise of improved performance, has recently been released.

Quadrics. Similar to Myricom, Quadrics is a mature player in this market. It arguably rose to fame when it was used on the Compaq AlphaServer SC (supercomputer). The network uses a fat-tree design, employing multiple small switches, to support up to 4096 ports. Copper cables are used, providing 3.2Gbps in each direction. The host-attached network cards are based on 66MHz/64 bit PCI. ElanLib provides the low level programming interface. OS bypass is used and the communication overheads are offloaded from the CPU. A global address space concept is supported. This allows communication between any areas of paged virtual memory across nodes. Peak throughput is ~6.3Gbps with an MPI-level latency of ~2µs.

Infiniband is the least mature of the technologies in this section, albeit possibly the one with the greatest potential. It has been developed by an industry consortium to provide general high bandwidth and low latency communication. It was previously called Future IO and was in part based on Tandem's ServerNet technology. In addition to cluster interconnects, it is planned to be used as a storage fabric, and even as a system bus. Several vendors are likely to offer switches. Mellanox, arguably the leading vendor at the moment, provides basic switches with up to 144 ports. The basic link speed is 2.5Gbps (bi-directional). However, multiple links can be aggregated to form a channel, e.g. 4X (10Gbps) and 12X (30Gbps). A number of vendors will probably offer host-attached network interface cards which will be

based on PCI-X, as the current Mellanox offering is. VAPI is the low level programming interface. OS bypass is used and the communication overheads are offloaded from the CPU. Peak performance on a 4X card is ~6.75Gbps with an MPI latency of 6-7µs.

Dolphin's **Scalable Coherent Interface (SCI)** is not switched; the nodes are connected in a ring, 2-D wrapped mesh or 3-D torus. It has been used to connect multiple boards in SMP systems, e.g. Sequent's NUMAQ machines. The network interface card handles pass through traffic (i.e. traffic not destined for this node); there is no host CPU overhead for this task. As the links are shared, this impacts on the number of nodes that can be put in a loop before it is saturated, typically 8-10. This means that a 2D configuration will be 64-100 nodes and a 3D torus 640-1000 nodes. Host adaptor cards are based on 66MHz/64bit PCI. Third-party MPI products are used, e.g. Scali's MPI Connect (Linux-based) and MP-MPICH. Peak throughput is ~2.4Gbps with an MPI latency of ~3µs.

34.3 Performance

For high performance clusters the key performance criterion is the increased delay that is acceptable when accessing memory on a remote server node. Remember that a simple main memory access on the local server node will typically be of the order of 50-100ns (nanoseconds). At the MPI level proprietary solutions have an optimum latency of under 10µs (microseconds). Although this sounds quick, it is 50-200 times slower than local memory access. In fact, the delay may be even greater, depending on the implementation, particularly if there are further layers to consider, e.g. application protocols sitting on top of MPI. One Oracle RAC (multi-node database) benchmark accessing 6-7 blocks per second shows ~200µs for a buffer access (proprietary solution) and ~1100µs (using Gigabit Ethernet).

Before discussing the performance of cluster interconnects *per se* the point should be made that the impact on the host servers needs to be considered. Technologically, much work has gone into increasing the bandwidth that can be supported by cluster interconnects *in toto* and by the individual host channel adapter (HCA) cards. Indeed, a number of HCA products have the potential to exhaust a PCI or PCI-X bus. Where this is an issue it may be necessary to investigate newer server bus technologies (see observations in the next section). One approach is to configure multiple HCA cards spread across multiple PCI buses, where the server hardware supports this (usually on larger SMP systems).

In terms of benchmarking, custom built benchmarks are preferable to artificial benchmarks, as they are likely to be the most useful in terms of assessing the extent and impact of inter-node communication. However, there is no shortage of "artificial" benchmarks. The figures that everybody quotes, including me in the previous section, are bandwidth throughput and latency. However, similar to any other area of technology, benchmarks can vary in their complexity and their usefulness (comparison with real world applications). See *Lies, Damned Lies and Benchmarks* (Chapter 3 of the book – *Performance Assurance for IT Systems*) for a general discussion on the pitfalls of benchmarking.

The lower down the stack (Figure 34-5) that performance is monitored, the more impressive the performance figures will be. Therefore, it is important to understand (a) the level where monitoring takes place and (b) how the figures may relate to user-observable performance in real-world applications.

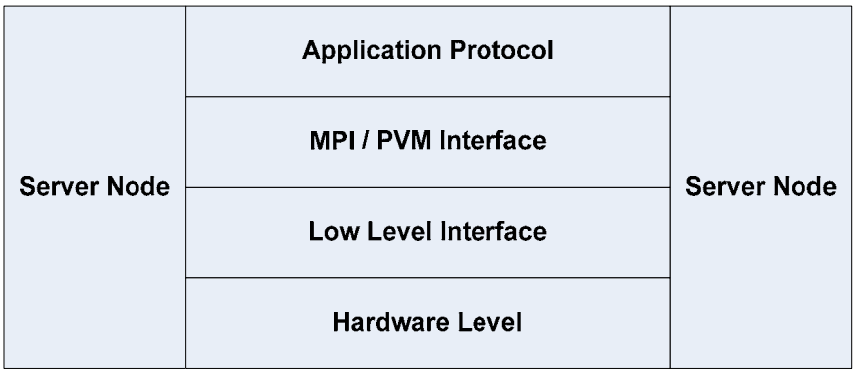

Figure 34-5 Potential Levels for Benchmarks To Report Performance

Other factors to look out for in artificial benchmarks include:

- Sustained levels of performance (a minimum of 20 minutes running at peak rates, ideally more). Beware of short tests

- The effect of varying message sizes. Throughput tends be poor (relatively speaking) at low message sizes, often not getting close to vendor-quoted figures below 20KB and often closer to 100KB

- The effects of throughput and message size on latency. Higher throughput and / or increased message sizes will affect latency adversely

- Bi-directional bandwidth and latency. Many tests may be limited to one-way traffic. Bi-directional testing is much more likely to find bottlenecks and thereby affect latency

- Host server communication overhead. There is still significant work for the operating system to do notwithstanding all the sterling efforts of vendors to minimise it

- The overhead of completion notification

- Buffer re-use. Server communication buffers have to be registered prior to use. The reason for this is that they have to be pinned in memory to avoid being paged. This process involves the operating system and obviously constitutes an overhead. Therefore, the re-use of buffers can be important. Simple single-threaded tests may be limited to the use of a single buffer. Tests that use multiple buffers can illustrate how effective the product is at re-using buffers. Low buffer re-use may impact on performance

- Hot-spot tests will help to identify bottlenecks under spikes (short bursts) of high activity

- Benchmarks are typically run on clusters of small servers, typically uniprocessors or dual processors. The question is how this may relate to your problem. If it is necessary to support larger SMPs there may well be a requirement to support a higher volume of inter-node traffic. This may result in larger (and more expensive network / fabric), and there may be issues relating to the host scalability (the possible need for multiple PCI gateways / buses to support the required number of host adapters).

34.4 General Observations

It is a truism that you tend to get what you pay for. This certainly applies to cluster interconnect technology. Systems that have low levels of inter-node traffic may be satisfied with the use of relatively cheap Gigabit Ethernet, providing that the higher levels of latency are acceptable. More significant volumes of inter-node traffic and/or stricter performance requirements will necessitate the use of a proprietary cluster interconnect solution.

Commercial applications do not have the same requirement as scientific applications for large interconnects with many hundreds of ports. At the current time it is difficult to imagine more than a handful of commercial applications worldwide that are likely to get close to 100 ports.

Future changes in this field are likely to include changes to speed up data transmission within the server nodes. A number of technologies, using high speed, switch-based architectures, are lining up to compete in this area, including:

- PCI-Express is a proposed replacement for PCI (or PCI-X) and AGP. Individual links (termed data lanes) run at 2.5Gbps at present (5 and 10 Gbps are being talked about in the future). Data can be transmitted across a "channel" with multiple lanes
- Infiniband itself
- HyperTransport, which is arguably better known as an processor interconnect that is used on AMD's Opteron systems, is used as an IO bus on Apple's G5 PowerMac
- RapidIO from Motorola, although this is slanted more at embedded systems.

Ethernet-based solutions are also changing in an effort to remain competitive. Apart from 10 Gigabit Ethernet, there are specialised NICs that run RDMA over TCP/IP. Certain products also offer TCP Offload Engines (TOE) on the NIC to reduce the impact on the servers of running TCP. These products are expensive at the time of writing but it is an area that is worth monitoring.

34.5 Further Reading

Significant material can be located in this field. I have limited the references to a couple of interesting papers and web sites of note.

Liu, J., Chandrasekaran, B., Yu, W., *et al*, *Micro-Benchmark Level Performance Comparison of High-Speed Cluster Interconnects*, Ohio Supercomputer Center. An excellent paper that summarises the results of detailed benchmarking.

Bode, B.M., Hill, J.J., Benjegerdes, T.R., *Cluster Interconnect Overview*, See the web site of the SCL Ames Laboratory (Iowa State University), http://www.scl.ameslab.gov. This is a very useful introduction to current technologies, along with basic bandwidth and latency benchmark results.

Understanding PCI Bus, PCI-Express, and Infiniband Architecture, Mellanox whitepaper. This is a balanced and useful document on possible future changes.

Web sites, primarily of the main cluster interconnect vendors, include:

http://www.myri.com Myrcom

http://www.quadrics.com Quadrics

http://www.dolphinics.com Dolphin

http://www.infinicon.com) and Network Xpress.

http://www.mellanox.com) Mellanox (Infiniband vendor)

http://www.hoti.org Hot Interconnects

http://www.scali.com Scali (software vendor).

