Scalable Cache Coherent Shared Memory at Cluster Prices

White Paper

A Qualitative Discussion of NumaConnect vs InfiniBand and other high-speed networks

By: Einar Rustad

"Today's societal challenges require novel technological solutions and well informed opinions for adequate decision making. The accuracy of simulation predictions increases with the performance of the computing architecture; timely results are only avialable using powerful computing resources." NW2V2

ABSTRACT

Numascale's NumaConnect™ technology enables computer system vendors to build scalable servers with the functionality of enterprise mainframes at the cost level of clusters. The technology unites all the processors, memory and IO resources in the system in a fully virtualized environment controlled by standard operating systems with NUMA support. The idea of using low cost high volume servers that has made clustering successful is also used by NumaConnect.

Clustering technology has been dominating high-end computing for the last decade with hardware based on InfiniBand playing and important role for the recent development of clustering. NumaConnect supports the alternative shared memory programming model and integrates at a low enough hardware level to be transparent to the operating system as well as the applications. To aid the understanding of the concept the differences between the Numaconnect and InfiniBand approaches to system scaling are analyzed.

Contents

1 Technology Background

1.1 InfiniBand

InfiniBand is the result of a merging of two different projects, Future I/O and Next Generation I/O. As the names indicate, the projects were aimed at creating new I/O technology for connecting systems with peripherals and eventually replacing other I/O interfaces like PCI, Fibre Channel and even Ethernet and become the unified backbone of the datacenter. In short, InfiniBand entered the system scene from the I/O side (or the "outside") of systems. The main focus of InfiniBand was to be able to encapsulate any packet format and provide high bandwidth connections between systems and their peripherals as well as between systems. InfiniBand is a "shared nothing" architecture where the main processors at each end of a connection are not able to address each other's memory or I/O devices directly. This means that all communication requires a software driver to control the communication and handle buffers for the RDMA (Remote Direct Memory Access) engines.

1.2 NumaConnect

NumaConnect has its roots in the SCI standard that was developed as a replacement for the processor-memory bus structure inside systems. The main focus was to provide a scalable, low-latency, high bandwidth interconnect with full support for cache coherence. The main architectural feature is the notion of a global physical address space of 64 bits, where 16 bits are reserved for addressing of nodes and the remaining 48 bits for address displacement within the node for a total address space of 16 Exabytes. The shared address space allows processors to access remote memory directly without any software driver intervention and no overheads associated with setting up RDMA transfers. All memory mapping is handled through the standard address translation tables in the memory management controlled by the operating system. This allows all processors in a NumaConnect system to address all memory and all memory mapped I/O devices directly.

1.3 The takeover in HPC by Clusters

High performance computing (HPC) has changed significantly over the last 15 years. This is no big surprise to industry veterans who in the 15 years before that had seen the field move from single processor monsters with vector capabilities from the brains of Seymour Cray through the massively parallel machines from Thinking Machines. Everybody knew that MIMD systems would come, and when they did, there was always the problem of lacking software that could utilize the combined compute power of many processors working in parallel. One of the major problems had to do with data decomposition and how to handle

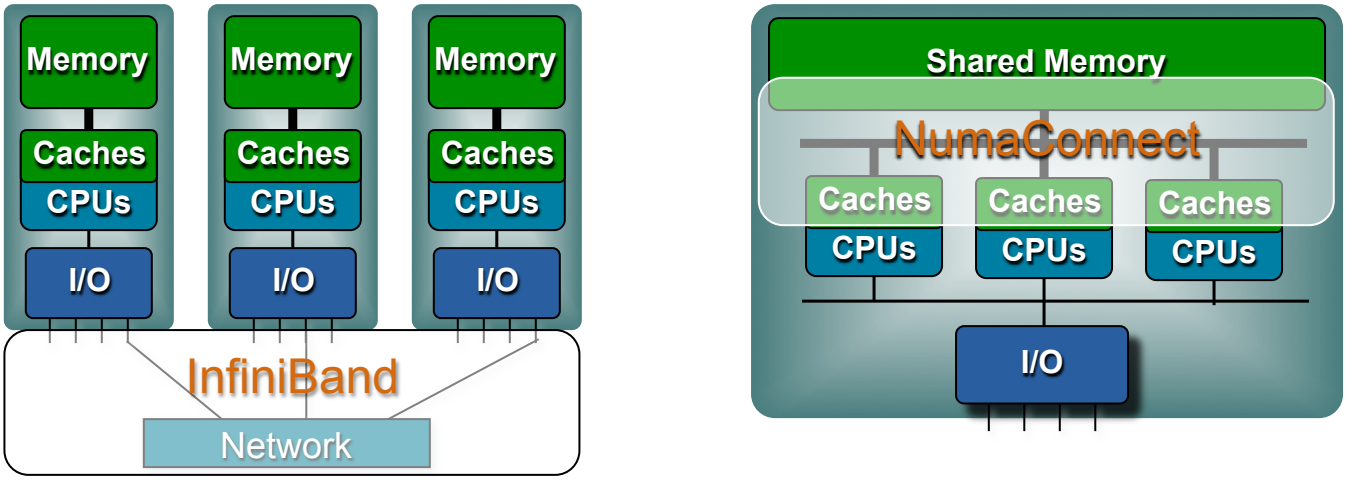

Figure 1, The System Positions of InfiniBand vs NumaConect

dependencies between processes and how to maintain the big picture when the problems were sliced to fit the individual memories of the computing nodes.

The vector machines grew their capabilities through a moderate amount of parallel processors operating on a huge, shared memory that allowed programmers to operate on the whole problem. The parallel machines forced programmers to write software to exchange data between processes through explicit messages and out of this grew the dominant communication library for HPC called MPI - Message Passing Interface.

Convex introduced their first parallel computer called "Exemplar" in 1994 with a global shared memory system. Soon after, Convex was acquired by HP and the Exemplar was reborn as "Superdome". About the same time, another parallel-processor project was undertaken by Floating Point Systems with a design based on using 64 SPARC processors operating on a shared memory. This was first acquired by Cray with little market success and it was quickly sold off to Sun Microsystems where it became a huge success as the Enterprise system E10000. All of these machines were used for HPC and they appeared in large numbers on the Top-500 list of their days. Common for all of them was that they carried the price tag of supercomputers. This kept the market for supercomputing from growing very fast.

In the early 1990s, a new paradigm started to appear. At this time, microprocessors had become powerful enough to do real computational work and scientists who are always looking for better tools to do their work saw this as an opportunity to get more computing power at their hands. An early approach was to use workstations in networks to run computing tasks during nights and other times when they were not busy serving the individual user. Platform Computing grew their business out of this idea through the LSF (Load Sharing Facility) product. Andersson, Culler and Patterson published their paper "A Case for NOW" (NOW = Network of Workstations) in 1994 advocating the use of cheap computing power in the form of mass-produced workstations to perform scientific computing.

The NOW approach could easily utilize the same programming paradigm as the high priced MIMD parallel machines, so some software was already available to run on these systems. About the same time several institutes started to set up dedicated systems instead of using other people's workstations and this gave birth to the concept of "Cluster". Thomas Sterling and Donald Becker started the "Beowulf" project to use commodity PCs, Linux and Ethernet to create a low cost computing resource for scientific workloads sponsored by NASA (http://www.beowulf.org/overview/history.html).

Since then, the HPC market has changed to be dominated by cluster computing due to the simple fact that the cost per processor core is 20 – 30 times less for a cluster than it is for large mainframe computers. Even though there are many tasks that are hard to do on a cluster, the huge price differential is a strong motivation for trying.

For applications that have few dependencies between the different parts of the computation, the most cost-effective solution is to use standard Gigabit Ethernet to interconnect the compute nodes. Ethernet is a serial interface with inherent potential for packet loss and requires quite heavy protocols in software (TCP-IP) for reliable operation on the application level. This means that the latency for messages sent between processes is quite high; in the order of 20 – 30 microseconds measured under Linux. Since HPC applications normally operate with quite short messages (many are dominated by messages smaller than 128 bytes), scalability for those is very limited with Ethernet as cluster interconnect.

Applications with requirements for high-speed communication use dedicated interconnects for passing messages between application processes. Among these are

Myrinet™ from MyriCom, InfiniBand™ from Mellanox and QLogic and proprietary solutions from IBM, NEC and others. The dominant high-speed interconnect for HPC clusters at this time is InfiniBand™, which is now used by 43% of the systems on the Top-500 list of supercomputers (Nov. 2010).

Common for all of these interconnects is that they are designed for message passing. This is done through specific host channel adapters (HCAs) and dedicated switches. MyriCom uses 10Gbit Ethernet technology in their latest switches and most InfiniBand vendors use switch chips from Mellanox. Common for all is also the use of high-speed differential serial (SERDES) technology for physical layer transmission. This means that many of the important parameters determining the efficiency for applications are very similar and that peak bandwidth is almost the same for all of them.

NumaConnect™ utilizes the same serial transmission line technology as other high speed interconnects, but it changes the scene for clusters by providing functionality that turn clusters into shared memory/shared I/O "mainframes".

2 Turning Clusters Into Mainframes

Numascale's NumaConnect™ technology enables computer system vendors to build scalable servers with the functionality of enterprise mainframes at the cost level of clusters. The technology unites all the processors, memory and IO resources in the system in a fully virtualized environment controlled by standard operating systems.

Systems based on NumaConnect will efficiently support all classes of applications using shared memory or message passing through all popular high level programming models. System size can be scaled to 4k nodes where each node can contain multiple processors. Total memory size is only limited by the 48-bit physical address range provided by the Opteron processors resulting in a total memory addressing capability of 256 TBytes.

At the heart of NumaConnect is NumaChip; a single chip that combines the cache-coherent shared memory control logic with an on chip 7-way switch. This eliminates the need for a separate, central switch and enables linear capacity and cost scaling. It also eliminates the need for long cables.

The continuing trend with multi-core processor chips is enabling more applications to take advantage of parallel processing. NumaChip leverages the multi-core trend by enabling applications to scale seamlessly without the extra programming effort required for cluster computing. All tasks can access all memory and IO resources. This is of great value to users and the ultimate way to virtualization of all system resources. No other interconnect technology outside the high-end enterprise servers can offer these capabilities.

All high speed interconnects now use the same kind of physical interfaces resulting in almost the same peak bandwidth. The differentiation is in latency for the critical short transfers, functionality and software compatibility. NumaConnect™ differentiates from all other interconnects through the ability to provide unified access to all resources in a system and utilize caching techniques to obtain very low latency.

Key Facts:

- Scalable, directory based Cache Coherent Shared Memory interconnect for Opteron
- Attaches to coherent HyperTransport (cHT) through HTX connector, pick-up module or mounted directly on mainboard
- Configurable Remote Cache for each node 2-4GBytes/node
- Full 48 bit physical address space (256 TBytes)
- Up to 4k (4096) nodes
- Sub microsecond MPI latency (ping-pong/2)
- On-chip, distributed switch fabric for 1, 2 or 3 dimensional torus topologies

2.1 Expanding the capabilities of multi-core processors

Semiconductor technology has reached a level where processor frequency can no longer be increased much due to power consumption with corresponding heat dissipation and thermal handling problems. Historically, processor frequency scaled at approximately the same rate as transistor density and resulted in performance improvements of most all applications with no extra programming efforts. Processor chips are now instead being equipped with multiple processors on a single die. Utilizing the added capacity requires software that is prepared for parallel processing. This is quite obviously simple for individual and separated tasks that can be run independently, but is much more complex for speeding up single tasks. The complexity for speeding up a single task grows with the logic distance between the resources needed to do the task, i.e. the fewer resources that can be shared, the harder it is.

2.2 SMP is Shared Memory Processor – not Symmetric Multi-Processor

Multi-core processors share the main memory and some of the cache levels, i.e. they classify as Shared Memory Processors (SMP). Modern processor chips are also equipped with signals and logic that allow connecting to other processor chips still maintaining the same logic sharing of memory. The practical limit is at two to four processor sockets before the overheads reduce performance scaling instead of increasing it. This is normally restricted to a single motherboard. With this model, programs that need to be scaled beyond a small number of processors have to be written in a more complex way where the data can no longer be shared among all processes, but need to be explicitly decomposed and transferred between the different processors' memories when required.

NumaConnect™ uses a much more scalable approach to sharing all memory based on directories to store information about shared memory locations. This means that programs can be scaled beyond the limit of a single motherboard without any changes to the programming principle. Any process running on any processor in the system can use any part of the memory regardless if the physical location of the memory is on a different motherboard.

3 Numaconnect Value Proposition

NumaConnect enables significant cost savings in three dimensions; resource utilization, system management and programmer productivity.

According to long time users of both large shared memory systems (SMPs) and clusters in environments with a variety of applications, the former provide a much higher degree of resource utilization due to the flexibility of all system resources. They indicate that large mainframe SMPs can easily be kept at more than 90% utilization and that clusters seldom can reach more than 60-70% in environments running a variation of jobs. Better compute resource utilization also contributes to more efficient use of the necessary infrastructure with power consumption and cooling as the most prominent ones with floor-space as a secondary aspect.

Regarding system management, NumaChip can reduce the number of individual operating system images significantly. In a system with 100Tflops computing power, the number of system images can be reduced from approximately 1 400 to 40, a reduction factor of 35. Even if each of those 40 OS images require somewhat more resources for management than the 1 400 smaller ones, the overall savings are significant.

Parallel processing in a cluster requires explicit message passing programming whereas shared memory systems can utilize compilers and other tools that are developed for multi-core processors. Parallel programming is a complex task and programs written for message passing normally contain 50%

- 100% more code than programs written for shared memory processing. Since all programs contain errors, the probability of errors in message passing programs is 50% - 100% higher than for shared memory programs. A significant amount of software development time is consumed by debugging errors further increasing the time to complete development of an application.

In principle, servers are multi-tasking, multi-user machines that are fully capable of running multiple applications at any given time. Small servers are very cost-efficient measured by a peak price/performance ratio because they are manufactured in very high volumes and use many of the same components as desk-side and desktop computers. However, these small to medium sized servers are not very scalable. The most widely used configuration has 2 CPU sockets with 4 to 16 CPU cores. They cannot be upgraded with more than 4 CPUs without changing to a different main board that also normally requires a larger power supply and a different chassis. In turn, this means that careful capacity planning is required to optimize cost and if compute requirements increase, it may be necessary to replace the entire server with a bigger and much more expensive one since the price increase is far from linear. For the most expensive servers, the price per CPU core is the range of USD 50,000 – 60,000.

NumaChip contains all the logic needed to build Scale-Up systems based on volume manufactured server components. This drives the cost per CPU core down to the same level as for the cheap volume servers while offering the same capabilities as the mainframe type servers.

Where IT budgets are in focus the price difference is obvious and NumaChip represents a compelling proposition to get mainframe capabilities at the cost level of high-end cluster technology. The expensive mainframes still include some features for dynamic system reconfiguration that Numascale will not offer initially. Such features depend on operating system software and can be also be implemented in NumaChip-based systems.

4 Numaconnect Vs Infiniband

4.1 Shared-all versus shared nothing

In a NumaConnect system, all processors

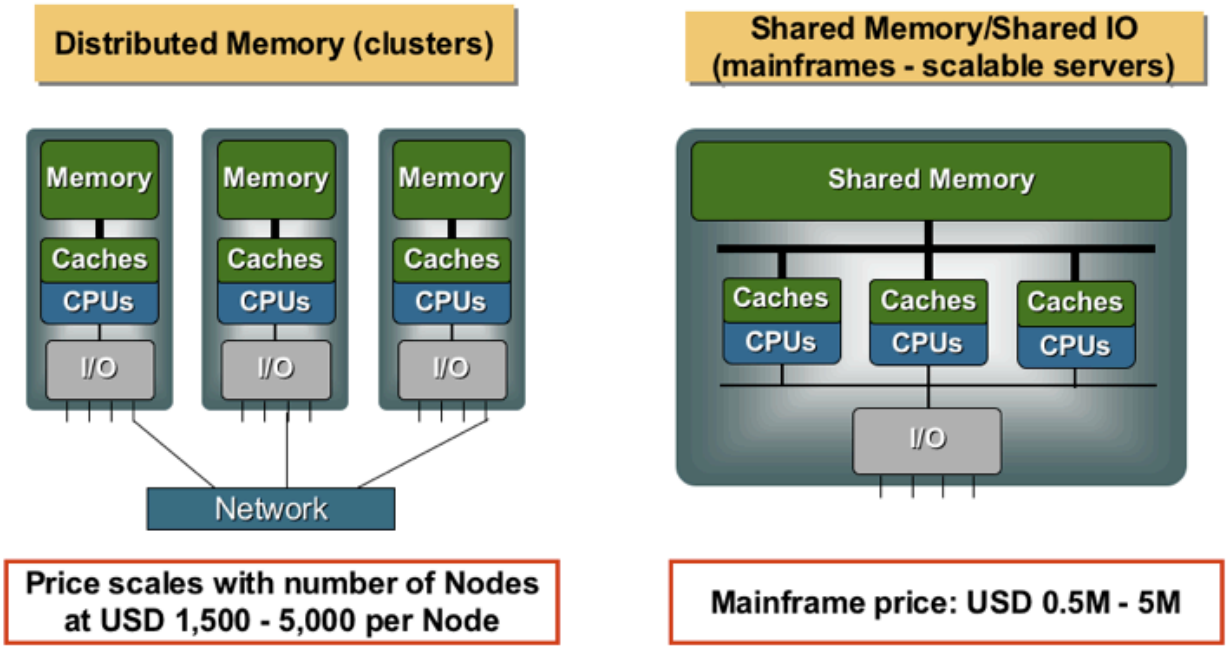

Figure 2, Price levels for clusters and mainframes

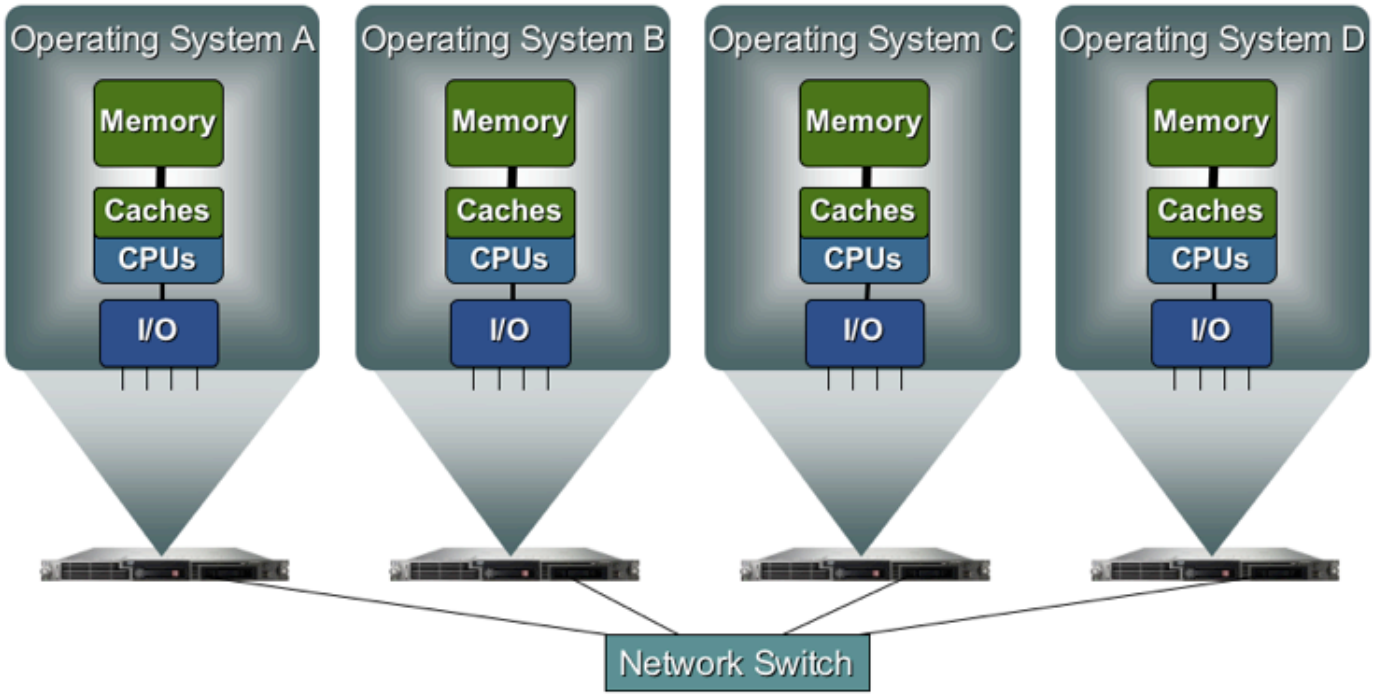

Figure 3, In a cluster, nodes are connected through the I/O system only

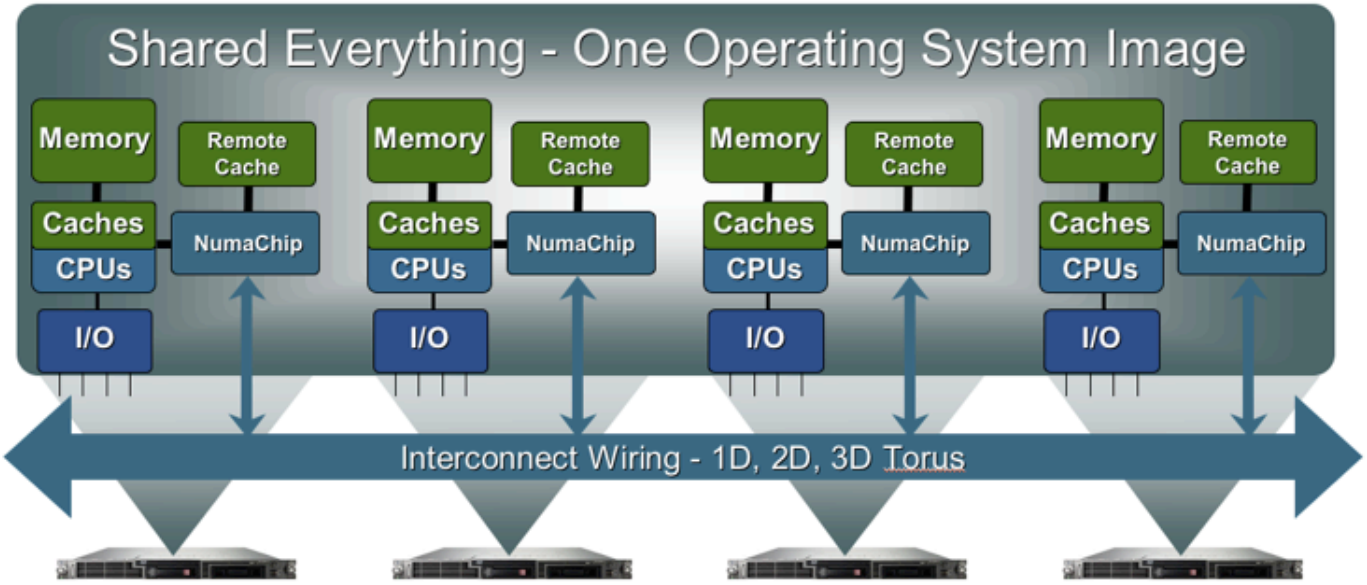

Figure 4, In a NumaConnect system, all resources can be shared for all processes

can share all memory and all I/O devices through a single image operating system instance. This is fundamentally different from a cluster where resource sharing is limited to the resources connected to that node only and where each node has to run a separate instance of the operating system.

For a program that can exploit parallel processing, the difference is also fundamental by the fact that a cluster requires the program to perform explicit passing of messages between the processes running on the different nodes whereas a NumaConnect system allows all processors to access all memory locations directly. This reduces the complexity for programmers and is also identical to the way each node in a cluster can operate. It is also important to note

here that a system with NumaConnect will execute message passing programs very efficiently, whereas a cluster cannot efficiently execute programs exploiting shared memory programming.

4.2 Shared Memory vs Message Passing

On a more detailed level, the main difference between NumaConnect and other interconnects like InfiniBand is that NumaConnect allows all processors in a system to access all resources directly, whereas InfiniBand only allows this to happen indirectly. This means that in a NumaConnect system a program can store data into the memory that resides on a remote node. With InfiniBand, this can only be done through the sending program initiating a remote direct memory access (RDMA) that can undertake the task on behalf of the CPU. This means that the sending program must call a routine that initiates the RDMA engine that is located on the InfiniBand host channel adapter and that in turn fetches the data to be sent from local memory and sends it across the network to the other side where it is stored into the remote memory. With NumaConnect, all that is required for the sending process is to execute a normal CPU store instruction to the remote address. The data will most likely be in the processor's L1 cache so only one transaction is required on the link between the processor and the NumaConnect module. The combination of one load and one store instruction will require about 0.6ns of CPU time and the store operation will proceed automatically across the NumaConnect fabric and be completed when the data is stored in the remote memory. For InfiniBand, the sending program must set up the RDMA engine through a number of accesses to the IB adapter in the I/O system and then the RDMA engine will read the data from memory and send it across the interconnect fabric.

The most important factor for application performance is the speed for the most common message sizes. Most applications are dominated by relatively small messages,

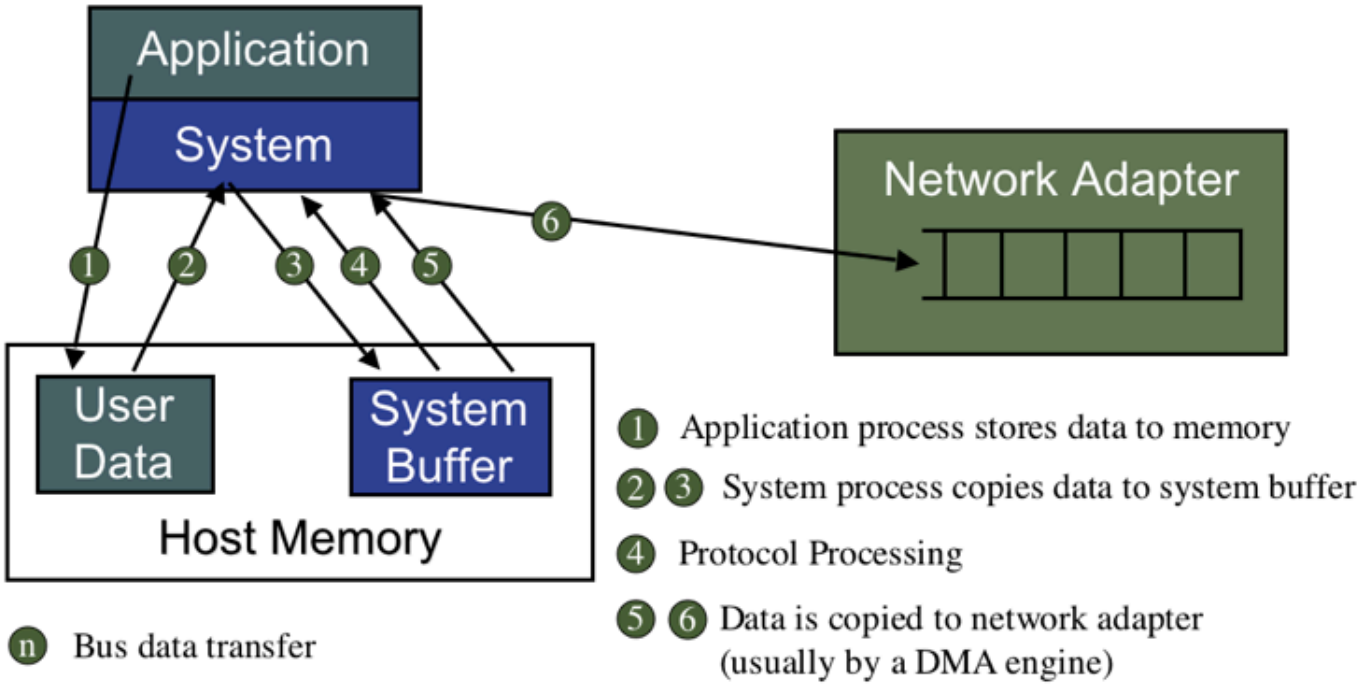

Figure 5, Shared Nothing Communication (Ethernet, InfiniBand, Myrinet etc.)

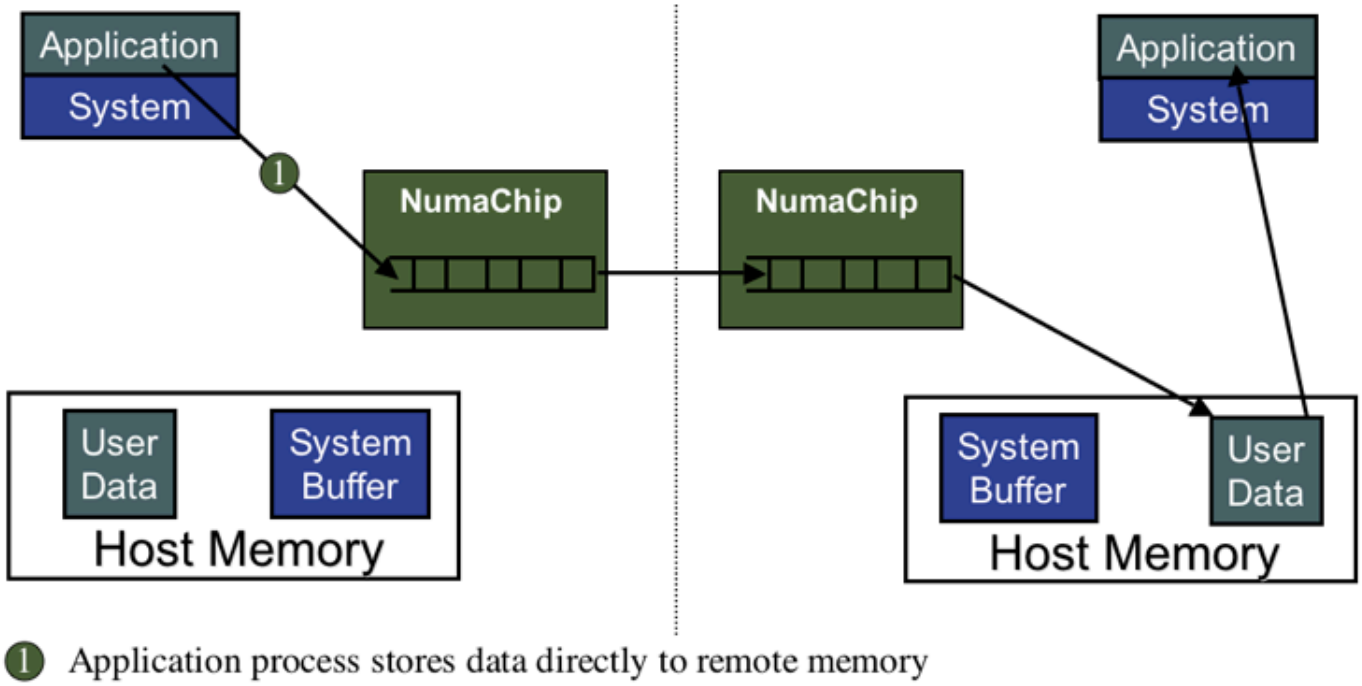

Figure 6, Shared Address Space Communication (Dolphin, Numascale)

typically in the order of 128 bytes. This means that the speed with which those messages are sent is the determining factor for the overall application performance. The measure for this speed is message latency and it is measured as the time it takes from one application process sends data until another application process receives them. The message latency measured by ping-pong/2 time for InfiniBand is in the order of 3 – 10 microseconds and for 10Gbit Ethernet it is in the order of 20-30microseconds.

Figure 7 shows the MPI message size for the Fluent medium class benchmarks. The dominant message size is less than or equal to 128 bytes. With this size of messages, the latency and message overhead will dominate the performance effects of the inter-process communication. In turn this limits the performance scalability for systems with long latency interconnects.

4.3 Cache Coherence

Truly efficient shared memory programming is only possible with hardware support for cache coherence. The reason is that modern processors are extremely much faster than the main memory and therefore rely on several layers of smaller and faster caches to support the execution speed of the processors. Current processor cycle times are in the order of 300 – 500 picoseconds whereas (DRAM) memory latency is in the order of 100 nanoseconds. This means that the processor will be stalled for approximately 300 cycles when waiting for data from memory and without caches, the utilization of the processor's computing resources would only be 0.3%. With caches that are able to run at the same speed as the processor, utilization can be kept at reasonable levels (depending on the application's access patterns and data set size).

For multiprocessing, where several processors operate on the same data set, it is necessary to enforce data consistency such that all processors can access the most recent version of data without too much loss in efficiency. Memory hierarchies with multiple cache levels make this task a grand challenge, especially when considering scalability. Microprocessors normally use a so-called snooping technique to enforce consistency in the cache memories. This was a quite viable solution with traditional bus structures since a bus in principle is a party line where all units connected can use all information that is transferred on the bus. This means that all transactions that can be relevant for updating the cache

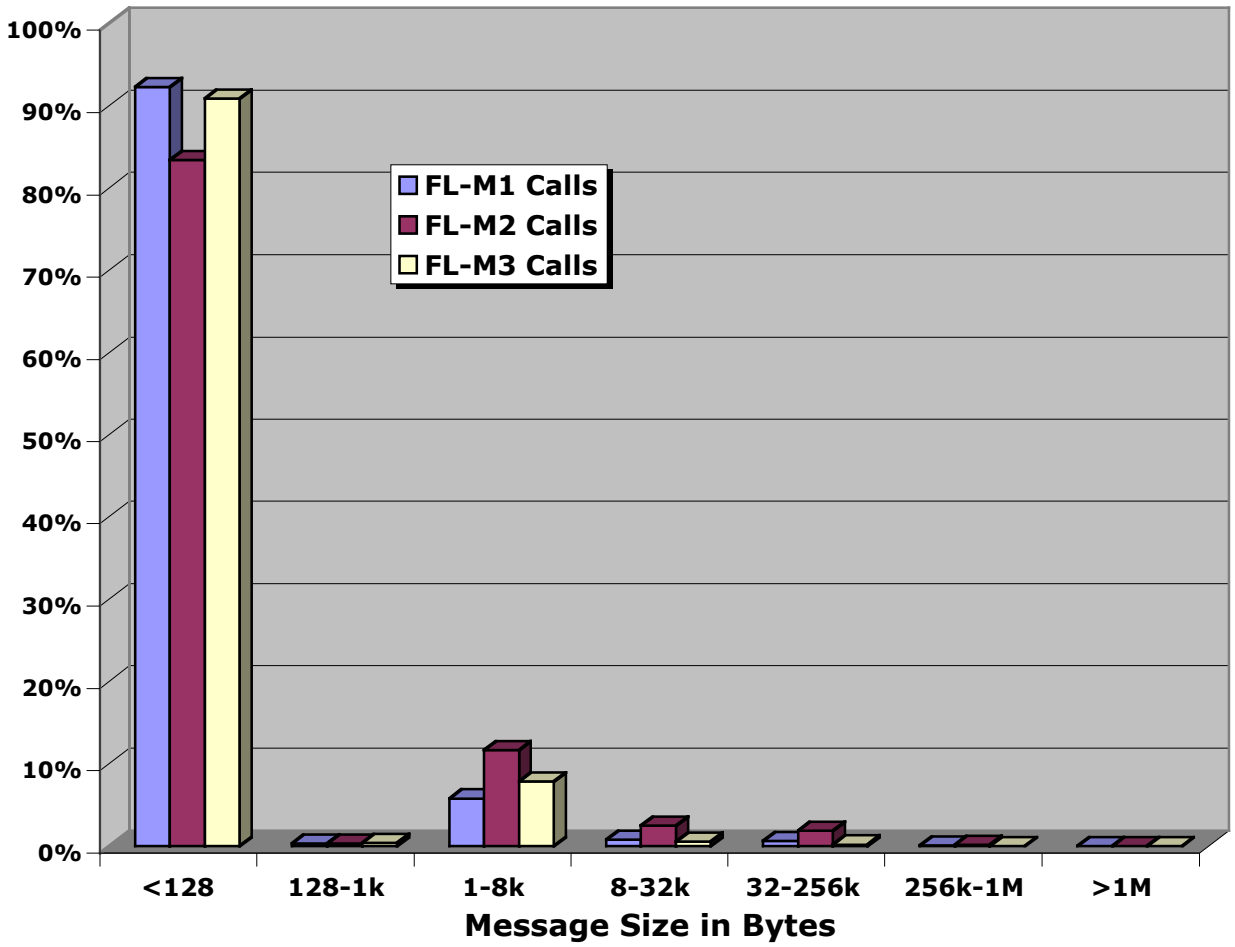

Figure 7, Fluent, MPI message size histogram

state must be available on the bus from the processor that operates on the data and that all other caches must listen – "snoop" - on those transactions to see if their cache state for the given cache line must be changed. When microprocessors move away from bus structures to point-to-point interconnects like HyperTransport, the snoop information must be broadcasted to all processors.

Buses are inherently very limited in scalability due to electrical considerations; a bus with many connections comes very far from being a perfect transmission line. In addition the physical length of the bus limits the frequency with which signals can be transferred. This inherent limit to scalability has resulted in modern bus based systems with only a couple of processor sockets connected to the same bus. With the electrical limitations already there, the snooping principle fits in quite well since it with a small number of processors (i.e. 2) it does not represent a limiting factor beyond the bus limitations themselves. When systems are to be scaled beyond a couple of processors, the snooping traffic that needs to be broadcasted will represent a significant overhead proportional to $O(n^2)$. This efficiently limits scalability to a few processor sockets (≈4) even if the point-to-point interconnect like HyperTransport can handle much more traffic than any party-line bus.

NumaConnect represents an efficient solution to the scalability issue by using a cache coherence protocol based on distributed directories. The coherence protocol is based on the IEEE standard SCI – Scalable Coherent Interface.

4.3.1 Coherent HyperTransport

The introduction of AMD's coherent HyperTransport (cHT) opened a new opportunity for building cache coherent shared memory systems. The fact that HyperTransport is defined as a point-to-point interconnect instead of a bus allows for easier attachment for third party components. The Opteron architecture with on-chip DRAM controllers was also more suitable for bandwidth-hungry applications due to the incremental memory bandwidth from additional processor sockets. By mapping the coherency domain from the broadcast/snooping protocol of Opteron into the scalable, directory based protocol in NumaConnect, customers can now build very large, shared memory systems based on high volume manufactured components.

4.3.2 Cache coherence complexity

Cache coherence is a complex field, especially when considering scalable high performance systems. All combinations of data sharing between different processors and their caches have to be handled according to a strict set of rules for ordering and cache states to preserve the semantics of the programming model. The SCI coherency model has been both formally and practically verified through work done at the University of Oslo, through expensive test programs and through large quantities of Aviion numaserver systems from Data General operational in the field over many years. NumaChip translates the cHT transactions into the SCI coherency domain through a mapping layer and uses the coherency model of SCI for handling remote cache and memory states.

4.4 Shared Memory with Cache Coherent Non-Uniform Memory Access – CCNUMA

The big differentiator for NumaConnect compared to other high-speed interconnect technologies is the shared memory and cache coherency mechanisms. These features allow programs to access any memory location and any memory mapped I/O device in a multiprocessor system with high degree of efficiency. It provides scalable systems with a unified programming model that stays the same from the small multicore machines used in laptops and desktops to the largest imaginable single system image machines that may contain thousands of processors.

There are a number of pros for shared memory machines that lead experts to hold the architecture as the holy grail of computing compared to clusters:

- Any processor can access any data location through direct load and store operations - easier programming, less code to write and debug
- Compilers can automatically exploit loop level parallelism – higher efficiency with less human effort
- System administration relates to a unified system as opposed to a large number of separate images in a cluster – less effort to maintain
- Resources can be mapped and used by any processor in the system – optimal use of resources in a virtualized environment
- Process scheduling is synchronized through a single, real-time clock - avoids serialization of scheduling associated with asynchronous operating systems in a cluster and the corresponding loss of efficiency

Such features are available in high cost mainframe systems from IBM, Oracle (Sun), HP and SGI. The catch is that these systems carry price tags that are up to 30 times higher per CPU core compared with commodity servers. In the low end, the multiprocessor machines from Intel and AMD have proven multiprocessing to be extremely popular with the commodity price levels: Dual processor socket machines are by far selling in the highest volumes.