Commodity, High Performance Interconnects

Chris Wilson

EE482

Abstract

As communication limitations become the dominant factor in designing products, devices connected by networks will become common. Most companies will not want to design their own interconnection fabric but will, instead, opt to buy an off the shelf interconnect system. Thus, vendors will offer "commodity" interconnects that are easy to integrate into other products and have good performance over a wide range of applications. The challenge for these commodity interconnects will be to deliver high performance at a reasonable cost. This paper will examine the issues in designing a commodity, high performance interconnect.

1 Introduction

Interconnects can generally be classified by the maximum link distance they have to support. A Local Area Network (LAN,) for example, supports link distances up to one kilometer. Metropolitan and Wide Area Networks (MAN,WAN) support link distances up to thousands of kilometers. Recently, System Area Networks (SANs,) have been introduced that support maximum link distances up to 10 meters or so with higher maximum bandwidth than is available in LANs. The advent of SANs has been due to the increasing data rate demands between boards in a system cabinet. Traditionally, board interconnect is done on a backplane using busses. However, busses are limited in the amount of bandwidth they can support. SANs were introduced as a backplane/bus replacement for systems that demand higher bandwidth than a bus can support and to allow scaling the number of nodes beyond what a bus is capable of supporting.

SANs were initially introduced in high end massively parallel computers (MPPs) where high bandwidth was required, but the size of the system ruled out the use of busses. At the same time, it was recognized that there is a substantial time lag in designing a multiprocessor after a microprocessor is introduced [3]. The use of an off-the-shelf interconnect could dramatically decrease this lag time. The initial NOW project used Ethernet LAN interconnect technology, which, although cheap and readily available, did not have the performance to make this concept useful.

Since these beginnings, SANs have started to move into lower end systems which have higher volume and, consequently, are more cost and schedule sensitive. The need for high performance SAN interconnects coupled with reduced cost and schedules has driven the introduction of "commodity" SANs. These are general purpose interconnects designed to fill multiple needs. Some of these interconnects are specifically intended to be sold as OEM interconnects. Others are intended for internal use, but are adaptable to multiple products within a company.

This trend can be extrapolated [1] to the board level (BAN?) and chip level (CAN?) As signalling rates increase, interconnect technology will have to be used to send signals less than a meter on a board [8], and less than a centimeter on a chip once the data rates become high enough[2].

1.1 Application Meets Technology

In any new emerging technology, there are three factors that drive the market:

- Basic technology,
- applications for this technology, and
- middle-ware to bridge the two.

In the case of interconnects, the driving technology is pin signalling rate. Total chip pin bandwidth is increasing at not nearly the rate that the speed and density of the silicon inside the chip is. Thus, the most fundamental interconnect need is for higher pin bandwidth. Research has been ongoing to increase the pin data rate, but it is only recently that readily available technology has enabled high speed interconnects. In section 2 on page 2, we will look at issues involved with increasing the pin data rate beyond what a simple bus can provide.

In the future, almost all applications will need to use interconnect technology due to the demands of faster chips and the physical distances that signals must propagate. However, today, there are already a number of applications that have gone, or are undergoing, the transformation from bus/backplane based interconnect to a switched fabric interconnect. The applications most often cited for high performance interconnects are:

- Multiprocessing message passing (clustering)
- High end server I/O.
- Shared memory (CC-NUMA) communication.
- Embedded applications, such as giga-bit LAN switching and file server interconnect.

All of these collectively form the core applications for SANs, and the main thing they have in common is that the system size is large enough such that nodes that want to communicate are a relatively large distance apart (several meters) due simply to the size of the

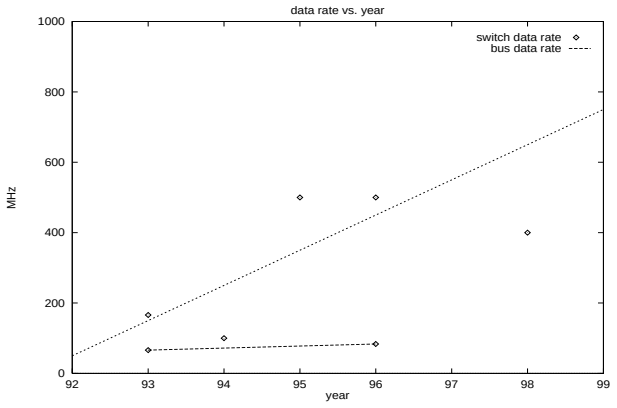
system, but at the same time, they want to communicate at as high a data rate as possible. We will examine these applications in section 3 on page 5 to determine the important parameters that drive their respective interconnect solutions.

What's left between the applications and the physical layer is the interconnect logical layers. This includes things like switching strategy, topologies, routing strategies, fault tolerance, and switch and node interface micro-architecture. We will examine these attributes in section 4 on page 8 for several commodity, high performance interconnects that are in current use. All of these interconnects share one thing, namely, that they are designed to accommodate more than one of the target SAN applications. In this sense, they have all been designed as somewhat "general purpose" interconnects. It is interesting, then, to study these architectures to determine what, potentially, may be the best general purpose interconnect solution in the future.

2 The Physical Layer

Figure 1.1. plots the increase in bus pin data rate for Sun's enterprise class servers as described in the Hot Interconnects conferences since 1993[4,5]. This figure also plots the fastest announced (at the same conference) pin data rate for commercially used switches over the same period [6,7,9,10,11]. While the bus signalling rate has increased at a rate of less than 10Mb/s/year, the switch signalling rate has increased at a rate of approximately 100Mb/s/year, with no end in sight [12].

Figure 1.1. Data Rate vs. Year Introduced



At the board level, Rambus introduced a signalling technology for DRAMs that has a 600Mb/s signalling rate [13]. They announced at this time that they would be able to increase this at a rate of 100Mb/s/year. Recently they announced Direct Rambus which increases the data rate to 800Mb/s. Thus, it appears that technology will be able to increase pin data rates at a steady rate for some time to come.

In the future, optical links, which have significantly higher data rates than electrical links, may become viable in SAN applications. Optical interconnects are already widely used in LAN and larger networks. For SAN applications, they are not as widely used due to the higher cost and longer latency of the optical/electrical interface, and the fact that there are no parallel optical interfaces commercially available yet.

The remainder of this section will look at the issues in high speed signalling that most affect SAN architecture design.

2.1 Signalling Basics

We can characterize physical signalling architectures into two types:

- Globally synchronous.
- Source synchronous.

Globally synchronous signals are those in which both the source and target chips are clocked by the same clock at the same time (within some skew tolerance.) In order for signals to be reliably transmitted across such an interface, it is generally necessary that only one physical digit (phit) be in flight at any time. Thus, the communication rate is limited by the electrical propagation delay plus any settling time, clock skew, and jitter. A bus signal traversing a backplane may have to travel up to one meter, requiring at least 6nS of flight time. Including settling, clock skew, jitter, and other effects, the data rate limit would then be around 100MHz.

A PC board bus trace may have to travel no more than a quarter of a meter. Modern microprocessors such as the Intel Pentium II, have bus speeds of 100MHz with four processors per bus. Next generation processors, such as Intel's IA64 architecture Merced and McKinley, will have bus speeds no higher than 200MHz and allow no more than two processors per bus. In the future, it is likely that processors will have only a point to point interconnect port to allow the maximum possible external bandwidth. Thus, the limit for bus-based globally synchronous clocking is probably around 200MHz at the board level.

Source synchronous clocking sends the clock with the data so that the receiving end can re-synchronize the incoming data. Source synchronous clocking allows multiple phits to be in flight at a time, which allows very high data rates over large distances to be achieved; the only limits being due to transmission losses and high frequency attenuation.

A recent talk by Mark Horowitz [12] suggests that there are no fundamental limits to increasing signalling rates. However, there are a number of issues that need to be resolved, with a number of trade-offs that need to be made. This paper will not deal with all the technical issues of designing a high speed physical link, but instead focus those issues that affect the application and network logical layers that are fundamentally different than a globally synchronous design.

2.2 Error Detection and Recovery

Traditional globally synchronous interfaces are generally considered to be error free and enough margin is included to make them so, within the limits of system reliability requirements. Consequently, error recovery is either non-existent or uses ECC, which primarily is used as a fault tolerance mechanism for hard errors (single wires breaking, for example.) However, for links that operate at GHz frequencies over meter distances, transient bit error rates can rise unacceptably high for ECC to correct due to lower voltage swings and narrower sampling windows. For commodity interconnect vendors, this is particularly true, since they can't necessarily control the noise environment that their interconnects are put in, unlike in a backplane/bus based system, in which the envi-

ronment is completely controlled by the designer.

Above 500Mb/s, most interfaces use differential signalling to eliminate common mode noise, which is a major error contributor in single-ended systems. Generally, it is fairly easy to keep the bit error rate (BER) below one in 10^{-14} using differential signalling to well above 1Gb/s up to several meters. Note that at 1Gb/s, a BER of 10^{-14} means that an error will be detected more than once per day. A system with thousands of pins, all of which were signalling at this rate, would detect an error at a rate of almost one a second!

Most high speed physical links use re-transmission to recover from errors. Re-transmission is robust in that it can recover from any transient error that is a result of the high signalling rate and thus can be used in high error rate environments. In fact, some vendors take advantage of this, by pushing the technology aggressively, using reduced margins, to allow the highest possible data rate [14].

Re-transmission consists of several parts: first, there must be some sort of error detection mechanism. This must be robust enough to detect multiple bit errors as well as single bit errors. There is no perfect multiple bit error detection mechanism and so trade-offs must be made. Most vendors use a cyclic redundancy checking (CRC) type error checking mechanism, which is robust for burst errors common on serial links. However, for parallel links, it is not clear that errors are correlated into consecutive bits.

Another approach is to use an error detecting code (EDC) similar to ECC, but hardened against multiple bit errors. Using CRC, the entire packet must be received before the checking can occur. EDC checks each word as it comes in allowing earlier detection of errors. However, EDC is not as robust against burst errors as CRC.

Secondly, the receiving node must send an acknowledgment (ACK) to the source node that a packet was successfully received. If a packet is corrupted or lost, no acknowledgment is sent, causing the sender to time out and re-transmit the packet. This requires the sending side to save a copy of the packet in a buffer so that it can be re-transmitted. Since there may be a long latency in getting an ACK, the sending side usually has multiple buffers to increase throughput. Also, ACKs may or may not be sent in the same physical channels as packets. If they are sent on the same channel, extra bandwidth must be allocated for ACKs, if not, then additional pins must be allocated (which in a pin-limited system amounts to the same thing.)

There are two types of re-transmission mechanisms: link-level, and end-to-end. In link-level re-transmission, an ACK is generated at the next hop in the network and sent to the previous hop. The advantage of this is that the ACK latency is short, meaning that buffers can be used more efficiently since time-outs are short. End-to-end re-transmission sends ACKs from target nodes all the way back to source nodes. The primary advantage of this is that this allows the network to guarantee reliable, exactly once packet delivery, but requires many more buffers at the end points due to the much longer time-outs[1]. End-to-end re-transmission has the advantage that the switches are not burdened with re-transmission logic that potentially increases the latency at each hop.

1. Link level re-transmission time-outs are a function of only the round trip link latency while ene-to-end time-outs are a function of both raw latency and network queueing delay. In order t support high network utilizations, long queueing delays must be tolerated.

Re-transmission can be done on a flit basis or on a packet basis. Normally, re-transmission will be done at the same granularity as the flow control. Thus, if a network uses packet based flow control, it is easiest to use packet based re-transmission, and if it is using flit based flow control, it is easiest to use flit based re-transmission.

Packet based re-transmission has a disadvantage in that it incurs store-and-forward (SAF) delay at the end point. The reason for this is that, for most applications, a packet cannot be delivered until it is guaranteed to be error free. A packet is not guaranteed to be error free until the tail flit has been received and checked. Thus, the entire packet must be received before the head can be delivered. Flit based re-transmission can deliver each flit as it arrives and has been checked. This requires EDC type checking and not CRC (or CRC over each flit as is done in the SGI SPIDER [10].)

In large networks, the SAF delay is a small fraction of the total latency. However, for small networks, the limiting case being a single switch, it may be significant. As we shall see when we analyze applications, small networks are dominant.

On the other hand, it is important that the application interface be easy to use, since one of our prime motivations is to reduce the lag time between new microprocessor introductions and multiprocessor introduction. Reduced schedules can easily make up for slightly longer latency. Busses have reliable, in order, exactly once delivery of packets. A network that presents this same model makes the system designers task simpler. A network model that does not guarantee this means that more work must be done in the system design to compensate for this. It is more difficult to guarantee this with flit based re-transmission than with packet based re-transmission and this is a disadvantage of flit-based re-transmission compared to end-to-end re-transmission.[2]

Thus, the issue of packet re-transmission vs. flit re-transmission is an important one. We will re-visit this once we have examined our target applications and see how existing commodity interconnects handle this problem.

2.3 Cost

In order for a commodity interconnect to be successful, it must have a good cost/performance ratio. There are several cost factors, including packaging cost, cable cost, and design cost. We will briefly examine each of these.

As pin data rates go higher, it is usually necessary to have lower stray inductances and capacitances in order to prevent unacceptable signal degradation. This usually means using higher cost materials in the interconnect pathway. First, there is the chip package pin. Bonded leads have no ground plane protecting them, thus they can have intolerably high inductance. Packaging technologies, such as ball grid arrays (BGA,) have significantly lower lead inductance, as do ceramic or metal packages and thus, generally, high performance interconnects use ceramic or metal BGA packages, which have a high cost. As these packages become more common, their cost should go down somewhat, however, the man-

2. Presenting a bus type model is actually more difficult due to flit-based *switching* rather than re-transmission. Flit-based switching may interleave flits from different packets requiring packet re-assembly at the endpoint in order to deliver them in order. In most cases, however, flit-based switching implies flit-based re-transmission.

ufacturing process for these packages is expensive, and so costs will remain high relative to other package types. It is not inconceivable that new package types that are specifically designed for high speed signalling at a low cost will appear eventually. An example might be packages that interface directly to optical interconnects.

Cables, also, can be expensive. Better dielectrics are required at higher frequencies to reduce frequency dependent loss smaller. Also, tighter skew tolerance is required for parallel interfaces, which increases manufacturing costs. Currently, there are no standard cables that meet the needs of high-speed signalling. However, once these appear, cable costs should go down significantly.

These issues are largely beyond the control of the interconnect architect. As high performance interconnects become more common and are standardized, then it is likely that packaging and cable costs will decrease. So, in the long run, these costs should not be an issue, especially when compared to system costs. For example, a 32-way multiprocessor may have close to $100,000 worth of components, not including the interconnect. Switches that cost $500 and cables that cost $300 each will be a small fraction of the system cost. In other applications, such as gigabit LAN switching, however, these component costs are too high to be economically feasible.

Thus, the real cost issue is what to do today. An interconnect that limits its pin data rate to less than 500Mb/s will certainly have better cost/performance than one that runs at 1Gb/s, no matter what other design decisions are made. However, customers of commodity interconnects will expect their interconnect to increase its performance over time. Thus, at some point, the lower cost/lower performance manufacturers must bite the bullet and step up to more expensive packaging technology. Those vendors who take this step early may have an advantage in the future.

Indirectly related to this issue is design and fabrication cost. Most high performance links have been announced by large semiconductor companies [17,18,19]. These companies have an advantage in that they have access to the latest process technology first, and have legions of circuit designers to design the necessary high speed interface circuitry. They have the advantage of scale and can amortize their expensive chip/packaging cost over many different chips. So, for the small fab-less company, the most cost effective strategy will be to use a vendor's high performance I/O cell and pursue an aggressive switch micro-architecture. Large companies that can custom design their own interface circuitry can afford to push the signalling rate and leverage their technology advantage.

2.4 Interface Design: Electrical vs. Optical Links

Higher and higher pin data rates require more complex interface logic that, if not designed carefully, can introduce unwanted latency. In addition, since source clocking introduces an asynchronous interface at the receiving end, a synchronization point is introduced at each hop. As switch latencies become lower, interface and synchronization delay can be large fraction of the total hop latency.

Thus, it is necessary to pay careful attention to both the outgoing and incoming interface circuitry. Most of the work is done by the receiving side, since it has to recover the transmitted data and do the synchronization. The fastest commercially available general purpose switches have port to port fall through latencies on the order of 40nS [10,15] and the link interface represents 50% of the latency in the case of the SGI SPIDER. Switch latencies are likely to come down in the future, thus, the link transceiver portion must be significantly less than 10nS in order to allow these low latencies. Indeed, Fujitsu's recently announced Synfinity-II transceiver design has a latency of 7.7nS [17].

On the other hand, if latency is not a concern, then longer latency transceivers can be used. This leads us into considering optical transceivers. Optical links have a number of advantages over electrical links, the main ones being higher bandwidth over longer distances and lower signal to signal skew. Optical links are used heavily in long haul (MAN,WAN,LAN) applications, but have not yet been widely used in SAN applications.

Optical technology that supports Gb/s data rates has been around for a number of years. Serial optical technology is currently reaching 10Gb/s in commodity designs, paralleling the introduction of 1 Gb/s electrical links. In general, the trend seems to be that current optical links have about ten times the data rate per signal of current electrical links.

Recently, parallel optical links [20,21,22] with 10-12 bit links giving an aggregate link bandwidth of 10Gb/s over inexpensive multimode fiber have started to become a reality. Efforts are underway to standardize these interfaces and, at least in the case of HP, they are targeting link bandwidths of up to 8GByte/s. This is significantly better than can be offered with electrical links and it should be easier to scale these numbers up than with electrical links. With the advent of Vertical Cavity Surface Emitting Lasers (VCSELs,) migration to parallel optical links for SAN applications should occur fairly soon

The biggest problem with optical links is the optical to electrical interface. Currently, this is done on separate transceiver chips resulting in 10-15nS of transceiver latency on both the incoming and outgoing sides. These transceiver chips currently mostly use "exotic" technologies such as GaAs, BiCMOS, bipolar silicon, and in at the 10Gb/s level, SiGe and thus are expensive and power hungry.

Until optical interfaces can be directly incorporated into switches, these types of links will not be used in the most latency sensitive applications, such as shared memory messaging. However, a large number of applications with less critical latency requirements, but high bandwidth requirements will likely have better cost/performance with optical links.

2.5 Power Consumption

Transceiver power consumption is another major issue. The best current cell designs consume around 0.2W per differential signal pair in CMOS technology. A maximum size chip may have as many as 512 signal pins. If all of these operated at 1Gb/s, the signal pin power consumption would be on the order of 125W using today's technology. Clearly, high speed links, are not going to be used in low power applications any time in the near future. The optical transceivers mentioned above consume roughly the same amount of power and so don't offer a solution in this area at presetn.

In addition, a thermal load of this magnitude requires expensive forced air cooling, meaning that at least for today, the concept of

high speed parallel links being ubiquitous is something that will only happen on expensive, performance oriented systems.

In the future, technology scaling will reduce power consumption somewhat, but most of the power consumed is in driving the wire. This is purely a function of the distance that needs to be driven and the error rate that can be tolerated. Thus, technology scaling will not help much here. However, as fast interconnects migrate to the board level where shorter distances are required, lower power drivers are possible. Some links have tunable drive strengths to allow minimizing power consumption on short links [17.]

2.6 Summary

The use of source synchronous clocking introduces the need for error recovery, which introduces the important issue of what model the interconnect presents to the system designer. Does the interconnect provide the same model as a bus-based system (reliable, ordered, exactly once packet delivery) or does the interconnect present an unordered interface, forcing the application to deal with these problems? This is a fundamental issue for which we need to examine SAN applications to determine the best solution.

High speed link designs also introduce cost, interface, and power consumption problems that make current high speed interconnects only viable only on high end systems at present. There is not much the interconnect architect can do to mitigate these problems.

So far, optical links have not been used on commodity, SAN interconnects, however, VCSELs will probably cause a migration to parallel optical links and this will become the dominant physical medium in the future.

3 Applications

The need for high performance interconnects has primarily been driven by the design of large multiprocessors, especially cache-coherent shared memory multiprocessors. It is not surprising then, that one of the primary application of emerging commodity, high performance interconnects is shared memory messaging. All processors need I/O, and so another driving force has been the need for increased I/O bandwidth and connectivity. In addition, the NOW project has indicated the need for high performance message passing (shared nothing) built using off the shelf components.

Designers realized that it was useful to use a single interconnect to implement each of these functions in a single system, thus saving development cost and effort. This section will analyze the characteristics of each of these applications in order that we may understand the trade-offs in designing commodity interconnects to fit these applications.

3.1 Shared Memory Messaging

High performance interconnects got their start in the implementation of massively parallel multiprocessors which were message passing machines designed to run large scientific applications. Scientific applications for these types of machines are usually fine tuned to maximize the performance of the interconnect. Because of this, these interconnects are primarily bandwidth sensitive.

With the introduction of the DASH system[24], interconnects became viable on shared memory (CC-NUMA) systems. Again, being a research machine, the primary benchmark workloads were scientific applications. However, shared memory communication is more latency sensitive than message passing communication. This is because shared memory communication is at a very fine grain and is highly unpredictable. A processor read miss usually stalls the processor, meaning that performance is directly a function of interconnect latency, especially if the latency is long, as is the case when using a switched fabric interconnect. Consequently, while massively parallel machines may have a thousand or more nodes, the DASH machine was only a 48-processor system.

This sensitivity to latency is further exacerbated by the fact that commercial, primarily on-line transaction processing (OLTP), applications have higher communication miss rates than scientific applications. While scientific applications typically have communication miss rates of less than 10%, over 50% of misses in OLTP applications can be communication misses. Communication misses also usually take longer to service than clean misses because the network must be traversed more times to get the data from a remote cache.

Today, commodity PC servers are emerging for the commercial market using commodity switched interconnect based memory systems [14,25,26]. All of these systems have between 16 and 32 processors only and processors are clustered onto 4-way busses. Thus, the interconnect need only support eight nodes at most. It's hard to consider an eight node network a network, so we need to understand why these systems are so small, if an interconnect is really necessary for these cases (Sun builds bigger bus based systems [27) and what the trend in the future will be for these types of systems.

The primary factor motivating the use of small shared memory systems is the lack of scalability in applications. In order to be scalable on a NUMA machine, the software must be tuned such that the most frequently accessed data is localized. Vendors running their own proprietary UNIX operating systems can do some tuning of the OS to localize static data, however, the true commodity PC server market uses the Microsoft Windows NT operating system, which has no NUMA tuning: memory requests are uniformly distributed to all nodes. Thus, on an eight node system, 7/8ths of the misses are remote (must traverse the network.) Consequently, no company markets a Windows NT based server having more than eight processors on two busses.

Thus, it would seem that there is little market for commodity, high speed interconnects for shared memory systems at present. However, in the future, this will not be the case. Slowly, Microsoft is increasing the scalability of Windows. Eventually, its page allocation and I/O will be completely NUMA-tuned allowing good scalability, even for applications that are not NUMA-tuned.

At the same time, processor bus speeds will continue to rise in order to meet the bandwidth demands of faster processors. As bus speeds rise, the maximum number of processors on a bus will decrease. This trend is evident in the evolution of Intel bus speed and maximum number of processors and attached chips over several generations as shown table 1 on page 6.

Eventually, the processor will not have a bus attachment, but, instead, will have an interconnect coming directly out of it. With the increasing scalability of Windows NT and the decrease in the number of processors per bus, this means that the number of interconnect nodes for these systems will increase even faster.

However, this does not necessarily mean larger interconnect topologies are necessary. The role of a bus in shared memory system is primarily as a concentrator. Shared memory traffic is bursty, and on the average, the data bus utilization is a small fraction of the peak bandwidth (25% bus utilization is typical.) However, it is necessary to have large peak bandwidths for two reasons: first to minimize memory access queueing delays and, secondly, to allow more data to be fetched per miss.

Table 1: Intel Bus Evolution

Processor	Bus Speed	Data B/W	Processors + Attached chips
Pentium Pro	66MHz	533MB/s	4+2
Pentium II Xeon	100MHz	800MB/s	4+1
Merced	133MHz	2GB/s	4+1
McKinley[a]	200MHz	6.4GB/s	2+1

a.Estimated

As was previously stated, latency is the primary factor in shared memory performance and so provisioning the memory system with extra bandwidth helps reduce latency due to queueing delay. Secondly, as processor performance increases, processor cache miss latency becomes a larger fraction of the application CPI (cycles per instruction) meaning designers must find ways to either reduce or hide the cache miss penalty. Making caches bigger decreases cache miss rates, but is prohibitively expensive, forcing designers to look for other ways to decrease miss rates.

One way to improve miss rates without increasing cache size is to increase the cache line size. Doubling the line size does not cut the miss rate entirely in half, but does double the bandwidth per miss resulting in a net increase in required bus bandwidth. This is reflected in the evolution of Intel processors. The Pentium Pro/ Pentium II Xeon has a line size of 32 bytes, Merced is 64 bytes, and McKinley (the successor to Merced) is projected to have a cache line size of 128 bytes. This accounts for the more than tripling of bus bandwidth between Merced and McKinley despite the fact that the McKinley processor is projected to be only twice as fast as Merced.

Thus. as we move toward having single processor nodes, the most cost effective solution will be to build a board level concentrator that maintains the four processor node illusion, connected to an external interconnect. The board level concentrator is an example of a board area network (BAN) and this will become a new technology in the near future.

In summary, commodity PC servers with more than eight processors and using switch based interconnects will become common in the future. Even in the best case scenario, however, 32-way systems with four to one concentrators connected using an eight node interconnect will be the largest configuration supported for the foreseeable future.

3.2 Messaging Passing

Also known as shared nothing messaging or clustering, this application is the most common one supported by the currently available commodity, high performance interconnects. The reason for this is simple; the hardware is simple, the software is simple, and the functionality has high value.

One way this is implemented in commodity systems is to build a PCI-to-interconnect interface and use a VI software driver. PCI is an I/O standard that most systems support. VI [28] is a standard user level messaging software/hardware protocol.

Although message passing machines started out as massively parallel number crunching machines, today's commodity message passing machines are used for an entirely different purpose. There are two primary applications for clustering. The first is load balancing of transaction-based jobs over a large server farm. This function has been done using LAN technology in the past; the primary advantage of SANs in this application is higher job throughput. However, since links are generally limited to 10 meters or less, the maximum number of system boxes that can be clustered using a SAN is limited by the physical space each box takes up. The practical upper limit for SAN clustering is between 10 and 100 nodes. Beyond this, the physical distances require LAN technology.

In most cases, though, the number of clustered systems will be small, between 10 and 30 at most. This is because these servers must attach to disk drives which also take up space and due to the fact that computing resources are usually scattered around large companies rather than being centralized.

The second major application of clustering is for fault tolerance. If large shared memory servers were feasible, load balancing would be done on these machines, however, clustering would still be used for fault tolerance. The reason for this is that a node failure in a shared memory system can bring the entire system down (and probably will.) Clustered systems, in which each node runs its own copy of the OS and communicates with other nodes through carefully controlled mechanisms can more easily tolerate node failures. In enterprise class servers, reliability, availability, and serviceability (RAS) is usually more important than pure performance, and this is primarily why clustering is a more common target of SAN vendors.

Message passing traffic patterns are different than shared memory. First, while shared memory uses mostly fixed data packet lengths, message passing packets can vary from a few bytes to thousands of bytes. Usually, however, message data transfer in most systems is done on a cache line basis using DMA transfer for large blocks [29].

Since the primary performance benchmarks for clustering are based on large block transfers, for these types of networks it makes sense to consider circuit switching instead of packet switching, which is required in shared memory applications. Some proprietary interconnects designed with MPP and clustering applications in mind, such as NCR's ByNET, use circuit switching [30].

In addition, unlike shared memory messaging, bandwidth is the

most important performance feature in clustered systems. Low latency is important, but interconnect latency is a smaller fraction of the total end to end latency due to the presence of software driver layers. Thus, switches with somewhat less than state of the art latency will perform well, and probably cost a lot less, resulting in better cost/performance, usually the dominant factor in choosing one interconnect over another.

Another performance factor in commodity systems is the fact that the network interface (NI) is connected to the system through the I/O interface (PCI.) This means that getting data in and out of memory is a two step process: first it must traverse the I/O controller and second it must traverse the network interface (NI,) making this path a bottleneck [31]. An alternative method that is used in proprietary systems is to connect the NI directly to the processor bus eliminating the I/O controller latency. However, this requires custom designed chips, negating the NOW concept. This is fine for high end MPP systems, but for commodity cluster systems, the advantage of PCI attached network interfaces is lower cost and faster time to market.

In summary, commodity clustered systems, while designed for large configurations, generally support less than 30 nodes. Secondly, a primary requirement for cluster interconnects is having good RAS characteristics[1] Third, bandwidth is a more important performance factor than interconnect latency due to software driver overhead.

3.3 I/O

I/O communication traditionally has lower bandwidth requirements than processor communication. Also, I/O communication architectures are highly standards based. Standards have an advantage in I/O for two reasons. First, there are many different I/O device types and having a common way of connecting them to a system greatly simplifies the system design. Secondly, standard interfaces allow mixing and matching I/O devices from any vendors allowing a low cost infrastructure for I/O.

There are two fundamental problems in I/O interconnection. First, processors may drive a large number of I/O devices, each of which has very low bandwidth requirements (A high end disk drive may not need more than a few tens of megabytes per second,) however, the combined I/O bandwidth of all these devices at the processor's memory system might be quite high (usually on the order of 1/2 the processor bus bandwidth.) Thus, this is a concentration problem. Secondly, in a multiprocessor, any processor must be able to get data from any device. This is an interconnection problem.

I/O devices are typically attached on busses, and busses may be bridged to allow attaching more devices than can fit on a single bus. In this case, the bus is acting as a concentrator and an I/O controller bridges the bus to the processor's memory system.

There are two ways of connecting processors to I/O. The first is to have an I/O network with multiple connection points. Each processor node connects to the I/O network. This allows any processor to connect to any device equally. The other alternative is that each processor node has its own locally connected I/O and for any other node to access this I/O, it must go through this node. This can be done in several ways. The simplest is through software; a node makes a request to another node to read some data from a device, which the target node does, delivering the data afterwards. This is usually very inefficient. Another way to do this is in hardware; the remote node talks directly to the local node's I/O controller to initiate the request. The data can either be streamed directly back to the requesting node or put into memory and them DMA'ed to the requesting node.

The primary disadvantage of locally connected I/O is that processor/I/O connectivity is provided by the processor's interconnect instead of a separate I/O interconnect. The need to do remote data transfers over the processor interconnect can cause bottlenecks which significantly affect system throughput. On the other hand, locally attached I/O allows the use of off-the-shelf chipsets, and sharing the processor interconnect to do I/O lowers cost and saves development time, both things that have high value in the commodity server market.

An alternative that bridges the gap between these two approaches is to use the same interconnect for inter-processor and I/O communication and have dedicated I/O nodes instead of locally attached I/O. This has the advantage of giving symmetric I/O access and removes the node bottleneck of using locally attached I/O, but requires a very high performance interconnect.

I/O traffic patterns are similar to message passing traffic; bandwidth is more important than latency. However, in I/O networks, message sizes are larger, Fibre Channel, a high performance I/O interconnect standard, has a maximum packet size of 2148 bytes, for example.

Latency between an I/O device and an I/O controller can be very long. For example, Fibre Channel uses essentially LAN technology with many protocol layers to meet its needs; it supports many different physical media and protocols. Consequently, it has switching times on the order of tens of microseconds instead of sub-microsecond switching times common in SANs.

Because of the need to support many different protocols and standards, commodity SAN technology is really most effective at solving the processor/device connectivity problem. New standards are being developed to address this problem, including Intel's Next Generation I/O (NGIO,) and the Future IO consortium led by HP, Compaq, IBM, 3Com, and Adaptec. Intel's NGIO[2] is a switched fabric interconnect that connects controllers to devices using channels called "thin-pipes." Controllers are attached to nodes through high bandwidth interconnects called "fat-pipes." Both of these areas will be opportunities for commodity, SAN technology.

In the long term, however, the picture is not as clear for I/O interconnects, compared to shared memory and clustering applications. As processor speeds increase, the major bottleneck to system throughput will be the time spent getting data to and from disk drives. It makes sense to move a lot of the processing into the disk drive or a nearby controller to reduce this latency. In this scenario, bulk data movement will be done over short distances and only control information will flow between the disk drive and processor.

1. Note that fault tolerance is one way to achieve good RAS characteristics, but fault tolerance and RAS are not the same thing.

2. Despite the name, this type of I/O architecture is not new, dating back to at least the IBM 360 series introduced in the mid-60's.

It is not clear what the workload characteristics would be in this type of processor/device interconnect, however since only control information is exchanged; it is possible that latency will be more important than bandwidth.

In summary, There are two problems in I/O interconnect: device to controller concentration and processor to controller connectivity. Concentration is addressed by LAN-like technology such as Fibre Channel while connectivity is addressed by SAN technology. Today's commodity systems are built using locally attached I/O because it is easy to build and cheap, but causes a bottleneck for remote I/O. Next generation commodity I/O interconnect will solve this problem using switched fabric which will be a prime application for high performance, commodity interconnects.

3.4 Miscellaneous Applications

The emergence of general purpose SAN interconnects has led to their adoption in applications other than the initial inter-processor and I/O communications applications. These include file server interconnects and LAN switching.

A file server is basically just a lot of disk drives attached to a system optimized to get high throughput for file access. Since disk drives are physically large, file servers are typically large boxes. Thus, communication distances are in the several meters range meaning that SAN technology is applicable.

With the introduction of giga-bit Ethernet LANs, switches for this technology will require large bandwidth and low latency at the same time. LANs usually have many attached nodes and hubs have a large number of ports. Lower cost is achieved by concentrating as many ports as possible into a hub. high bandwidth, low latency SAN switches allow a high degree of concentration and thus, are ideally suited to this application [34].

Taking this a step further, Internet (IP) routers are basically switches with many ports. These systems are physically large due to this large number of ports. The fact that they are also switches makes them ideal targets of SANs.

All of these applications use interconnect technology. What they have in common, though, is the fact that the systems are large, and have high data rate demands which call for switched fabric interconnects. Secondly, all these applications are potential commodity applications due to their high growth rate, ranging from 50-200% compound annual growth rate. So, while proprietary interconnect is certainly viable in these applications, in the future cost and time to market sensitivities may call for commodity interconnect solutions in these systems.

3.5 Summary

The common thread among all the applications discussed is a need for very high performance at a low cost and with fast time to market. This is due to the fast growth rates of demand for these applications.

Current applications require only small networks, no more than 30 nodes, with higher volumes occurring at the smallest sizes. What this calls for is a not only a high performance, low cost solution, but a scalable one as well. Ideally, a one node system could easily be expanded to maximum size with incremental cost and linear performance gain. Because of the small maximum sizes, network topology is not much of a concern.

Performance trade-offs were different between applications, shared memory being more latency sensitive, and message passing and I/O being more bandwidth sensitive. In addition, ease of system design and RAS are equally important.

The issue of standards conformance is not as clear. Proprietary, high performance interconnects generally do not follow any standards. Lower performance LAN and I/O interconnects are heavily standards oriented. Commodity SANs fall on the boundary between the two. Some commodity, higher performance interconnects are not standards based, but some are, notably Scalable Coherent Interface (SCI) based systems. The next section will talk more about SCI based systems to determine the impact that standards have on high performance networks.

4 Implementations

This section will discuss interconnect implementation issues based on the physical design and applications issues raised in previous sections. Several commodity and general purpose, high performance interconnects will be compared based on these issues. Afterwards, an attempt will made to arrive at a consensus on the design of high performance, commodity interconnects for the future.

The interconnects that will be studied are:

- Fujitsu Synfinity.
- Myricom Myrinet.
- SCI (Dolphin).
- Compaq/Tandem ServerNet.
- SGI SPIDER.

The next sections will give a brief overview of each interconnect.

4.1 Fujitsu Synfinity

Synfinity [14,15,16] was initially designed for shared memory applications in high end PC servers. It is also used for clustering and has the unique feature of supporting both clustering and shared memory on the same interconnect. Synfinity was designed by Fujitsu's Fujitsu System Technologies division to be sold as an OEM chipset. This is a new interconnect that is just starting commercial deployment. So far, the only customer has been Fujitsu itself, however, as the technology proves itself, it is expected that other vendors will incorporate it, especially those who need the highest performance available.

The Synfinity-I switch is a six port input buffered crossbar with 1.6Gbyte/sec. unidirectional links and 42nS port to port fall through latency with a pin data rate of 400Mb/sec. This combination of bandwidth and latency is unmatched by any industry designs, both proprietary and commodity, but is also a very costly solution.

In fact, most commodity applications don't need this level of performance and Synfinity can't compete on cost, thus, it is likely to win only the very highest performance designs. However, Synfinity is well positioned for the future since they are already well down the road to solving the physical link technology issues men-

tioned earlier. Most other lower performance, lower cost vendors will eventually have to deal with the problems that Synfinity has already solved. Synfinity will probably slowly gain market share as the company adjusts its price/performance to match industry needs.

Synfinity uses source routing and has a generic node interface chip called a MIC which holds the routing tables. The routing table size is 128 entries limiting the maximum network size to 128 nodes for the initial version. The use of table-based routing allows Synfinity to support arbitrary topologies and Synfinity supports two virtual channels which are used for the higher level cache coherence and message passing protocols. Thus, packets travelling in the network are confined to a single virtual channel, disallowing closed loop topologies such as tori.

Synfinity supports exactly once, in order, reliable packet delivery making it a good backplane bus replacement. Reliability is achieved using end-to-end re-transmission. Flow control is packet based on a per link basis, and uses a credit mechanism. The switch uses virtual cut through to reduce latency. The use of a reliability layer allows Synfinity to aggressively push technology margins in order to improve link speeds.

A packet consists of source route header, reliability layer header, and application layer header plus payload. It uses EDC, and so no trailing flit is needed. The maximum application layer payload length supported by the MIC is 128 bytes.

Links are 34-bits wide, source synchronous, single-ended. The maximum transmission distance is five meters at full speed, and 10 meters at half speed.

Synfinity also has a PCI adapter which supports 64-bit, 66MHz operation and has a software driver to support VI.

4.2 Myricom Myrinet

Myricom [1,32], descended from the Mosaic [33] technology developed at CalTech for MPP applications. The company was founded specifically to provide a commodity, high performance interconnect for general purpose SAN applications.

Myricom has a wider range of components than Fujitsu to accommodate as many applications as possible. Myricom has eight and 16-port switches, the 16-port switch having a cost of less than $500/port. The switches are pipelined crossbars and use wormhole flow control, however, the switch latency is around 100nS for an eight port switch with link bandwidth of 160MByte/sec at up to three meters.

The Myricom node interface (LANai) consists of a DMA engine, network interface, and processor to implement the node protocols and interfaces to a PCI bus. Also, they have a simple “FIFO” interface similar to the Synfinity MIC that is designed for use in higher performance systems.

A Myrinet packet is byte oriented and consists of a source route header, an application header, body, plus trailing CRC. The maximum node limit is very large, meaning that their source routing must be time consuming. This probably means that they have significantly longer node interface latencies than Synfinity, for example.

Myricom’s strategy is to not pursue technology aggressively. They only recently completed the transition to 0.35micron technology compared to Fujitsu which has used 0.35micron technology since 1994 and is currently transitioning to 0.18micron. They have conservative margins in their design in order to allow headroom for future improvements. [1]

The primary application Myricom is targeting is large message passing machines and clustering. Most of Myricom’s customers are government sponsored agencies or organizations building large MPP systems. Their penetration into the commodity clustering business, however, is less clear. Although their technology is sound, although somewhat on the low performance side, marketing is a big factor in determining the success of a commodity interconnect. Many companies don’t like buying high performance solutions from small companies. This is an area in which large companies like Fujitsu and Compaq/Tandem will have an advantage. However, Myricom’s niche should sustain them into the future.

4.3 SCI

One way of overcoming the “small company” problem that Myricom has is to use a standards based interconnect. Standards based systems allow inter-operability, which means that if one supplier goes out of business, it is possible to switch to another. This is the idea behind the SCI interconnect.

The SCI initiative was started in 1988 and completed standardization in the early 90’s [35]. Since then at least three companies have emerged as suppliers of SCI commodity components: Dolphin Interconnects[6,36,37] Vitesse, and ISS [38]. In addition, several large companies have developed SCI components for their own internal use [39,40].

SCI defines a physical layer and logical layer that supports cache coherent memory operations. The major goal of the SCI protocol is to allow scalability to large numbers of processors. The protocol is a distributed cache-based protocol. If a cache has a copy of a line, it adds itself to a linked-list of sharers. The home node for a line has a pointer to the head of a sharing list. Compared to central directory-based coherence mechanisms, this allows linear directory scaling up to any number of nodes. Central directory based systems suffer from quadratic scaling if they do not use some sort of compression scheme, which usually slows these protocols down significantly. The disadvantage of the SCI protocol is that it is inefficient for small systems. In [25] Lovett and Clapp report that the complexity of the SCI protocol forced them to choose a programmable coherence controller which has a high latency. In addition, the complex SCI protocols exchange many messages to do the coherence actions resulting in only 10-13% of the messages being data messages. This is compared to central directory-based systems which have 25-40% data messages.

In addition, being a cache-based protocol means that the caches must be designed to use this protocol. This makes SCI unsuitable as a processor L2 cache coherence protocol. Instead, most systems use a node level cache which is kept coherent using the SCI protocol.

1. which is a strategy that doesn’t make much sense. Why leave headroom, when the easiest way to increase performance in the future is simply to plan on using the vendor’s next technology spin?

The SCI physical standard defines a fixed data link of 1Gbyte/sec. using 500Mb/s signalling rate over 18-bit channels. SCI nodes are connected in a ring topology and rings can be connected by switches. The maximum number of nodes on a ring is <unknown>, and the total number of nodes in a system is over a million.

Dolphin and Vitesse' chipsets implement the physical layer functionality of the SCI protocol. The coherence protocol is left to the system designer. This negates some of the advantages of using an off-the-shelf interconnect since implementing the coherence protocol is complex. On the other hand, transport protocols on rings are also complex and the data pump chips hide this.

Dolphin's chipset is CMOS based, and Vitesse uses GaAs. Consequently, the Vitesse solution is more expensive, but does have a higher link data rate and lower latency. The basic function of these chips is to insert and extract packets and to bypass packets in the ring not destined to this node. The incoming and outgoing ports are in different clock domains and so there is a slack buffer between the incoming and outgoing ports. The node interface is basically a simple incoming and outgoing FIFO. Node bypass latency is around 100nS, and injection/extraction latency is around 300nS for the Dolphin chipset.

Dolphin has a four port switch with 200MByte/s links, and 480nS fall through latency and also sells a PCI to SCI bridge for clustering.

4.4 Tandem ServerNet

In the early 90's, Tandem recognized the need for processor/I/O connectivity. They found that high performance I/O interconnects such as Fibre Channel did not have the characteristics to support this connectivity efficiently [41] and the commodity LAN peer to peer solutions such as Ethernet were too low performance. They consequently decided to design their own interconnect, called ServerNet.

The original ServerNet had low performance by today's standards; 50MByte/sec. links and 300nS switch fall through time. ServerNet-II [42] has increased the link bandwidth to 150MBytes/s. The primary reason for ServerNet's relatively lower performance is the use of less aggressive technology. Tandem's ServerNet-I switch, designed in 1994, was implemented using 0.5micron gate array technology, in comparison to the semi-custom designs of Fujitsu and SGI designed at the same time, which achieve an order of magnitude better latency and bandwidth.

Despite its physical design shortcomings, ServerNet uses competitive logical layer techniques. The switch is an input buffered crossbar using wormhole routing. It is unusual in that it has only one virtual channel, but only 64bytes of flit storage per input port. With a link bandwidth of only 50MBytes/sec., this may not cause a significant performance degradation, though.

The routing is table-based in the switch and care must be taken to avoid routes that may deadlock. Since, as we have seen, most SAN networks are small, this does not impose much of a restriction on the topologies that can be supported.

Although originally designed as a processor-I/O connectivity solution, as most general purpose SANs have, ServerNet has started to offer clustering support through a PCI interface card. This is a natural direction for ServerNet to move, since it was originally intended for highly reliable systems, Tandem's core market. The original deployment of ServerNet was in Tandem's proprietary fault-tolerant Himalaya systems. Tandem/Compaq have also been trying to OEM ServerNet for clustered systems, thus, making ServerNet a true commodity SAN.

4.5 SGI SPIDER

The Silicon Graphics Scalable Pipelined Interconnect for Distributed End point Routing (SPIDER) chip [10] is the only one of these interconnects designed for proprietary use. However, Silicon Graphics designed it with multiple purposes in mind. Its primary use is as a high performance shared memory messaging interconnect in SGI's Origin 2000 CC-NUMA multiprocessor [43]. However, SGI claims the design will work well for distributed graphics, and high-end LAN switching.

As befitting a shared memory switch, it has very high performance. Remarkably, the SGI SPIDER switch and Fujitsu Synfinity switch have very similar performance characteristics, which is a result of targeting the same application and using aggressive technology and design techniques. The SPIDER has slightly lower latency[1], but Synfinity has double the link bandwidth, making it the performance winner. However, the SPIDER switch does more work in the switch than Synfinity, which pushes as much as possible to the end points. This, coupled with the fact that Synfinity's switch is implemented in 0.35micron technology while SPIDER is 0.5micron, is the reason that Synfinity achieves higher performance at the switch level.

However, the actual performance difference between the two is less clear. First, SGI uses virtual channel flow control and has four virtual channels, although the ability to do adaptive routing is somewhat limited since these virtual channels are used by their coherence protocol also. This should mean that SPIDER can attain higher link utilizations, but also means that SPIDER is a completely unordered network; Synfinity has endpoint to endpoint ordering guarantees which simplifies its coherence protocol design in comparison to SPIDER.

While Synfinity is completely packet-based, SPIDER uses flit based flow control. SPIDER has 1Kbytes of buffer at each input port vs. Synfinity's ~900bytes. However, since SPIDER uses virtual channels and flit based flow control, it probably makes more efficient use of its buffers. On long links, this may make SPIDER competitive with Synfinity with respect to link bandwidth. However, even if Synfinity achieved only 50% of its maximum throughput, this will still be better than SPIDER, no matter how well it does.

The SPIDER table-based routing uses a two-level lookup, with both accesses being done in parallel. The reason for this is to limit the size of the tables in the switch. SPIDER supports up to 512 end points, but with the two level lookup mechanism, the two tables have only 48 entries each. The two-level table does impose some restrictions. Topologies tend to be topologies of topologies (such as cubes of hypercubes) in large configurations due to the limited radix of the switch. This is reflected in the hierarchical nature of

1. However, I believe that the actual latencies are the same, and it is only a difference in accounting. Fujitsu adds 1/2 cycle to account for the random phase difference between the incoming signal and internal clock, while SGI does not.

the two-level lookup, meaning that this is an efficient mechanism that imposes little restriction on topologies.

Like Synfinity, SPIDER is a very costly solution. The SGI Origin 2000 system is primarily intended for what would be considered supercomputer applications, meaning that cost is a less important factor. Thus, SPIDER has been successfully deployed in its intended application. Apparently, SGI intends to use SPIDER in other applications. It will be interesting to see if they run into the same cost structure problems as Synfinity for these applications.

4.6 Summary

table 2 on page 11 summarizes the major characteristics of each interconnect.

All of these interconnects have their own strengths and weaknesses.To summarize these:

- Synfinity has by far the best performance, however, it also has the highest cost. The current market does not need this level of performance and, thus, it is priced out of the market. However, this will probably change in the future.
- Myricom offers good performance and good cost, but suffers from being a small company. Customers don't want to buy interconnects from small un-established companies. Consequently, Myricom relies on government contracts and "one off" projects that allow them to claim a lot of customers, but not commodity, high volume customers.
- SCI is a standard allowing small companies like Dolphin to sell to commodity customers. However, SCI's ring topology has poor bandwidth scaling and long latency, and the complicated SCI protocols are inefficient for the small network size seen in average commodity systems. It is likely SCI in its current form will be phased out eventually and replaced with something that better matches the needs of commodity SANs.
- ServerNet has the advantage of a large company backing it and being the first SAN in the market. However, its performance is low and Tandem's lack of custom circuit design experience may hurt it in the long run against competitors like Fujitsu.
- SGI SPIDER is a proprietary interconnect with high performance, and cost structure to match. It should continue to evolve and be used within SGI products.

From the discussion, it is clear that, while they all solve part of the SAN problem, no company has a perfect solution. In fact, while they all show promise, they all have major weaknesses that must be overcome.

Thus, none of these is a perfect SAN solution. The next section will discuss the direction to take to improve SAN technology as data rates move up and the need for board area networks becomes greater.i

Table 2: Interconnect Comparison

Feature	Fujitsu Synfinity-I	Myricom	Dolphin SCI (LC-2)	Tandem Servernet-II	SGI SPIDER
link signalling rate	400MHz	160MHz	125MHz	125MHz	400MHz
link technology	single-ended low voltage	single-ended low voltage	LVDS	LVDS	single-ended low voltage
link width	34 uni-dir.	10 uni-dir.	18 uni-dir.	9 uni-dir.	21 uni-dir.
link B/W/dir.	1.6GByte/s.	160MByte/s.	1GByte/s	125MByte/s	0.8GBytes/s.
max link distance	5m	3m	?	30m	5m
topologies supported	arbitrary[a]	arbitrary	hierarchical ring	arbitrary[b]	hierarchical
routing method	source	source	?	table	two-level table
maximum nodes	128	millions	millions	1 million	512
maximum payload size	128 bytes	?	64 bytes	64 bytes	?
overhead/packet	16 bytes	variable	?	16 bytes	20%
request/response messaging deadlock free?	Y	?	?	N	Y

Table 2: Interconnect Comparison

Feature	Fujitsu Synfinity-I	Myricom	Dolphin SCI (LC-2)	Tandem Servernet-II	SGI SPIDER
in order, reliable, exactly once delivery?	Y	?	?	Y	N
switch ports	6	8,16	4	6 (or 12?)	6
switch unloaded latency (port to port)	42nS	~100nS for the 8-port switch	48nS for node bypass in LC-2, 600nS for switch	300nS	40nS
switch bisection B/W	9.6Gbyte/s.	1.28Gbyte/s.	1Gbyte/s LC-2	750MByte/s	4.8Gbytes/s
switch type	input buffered crossbar	?	-	input buffered crossbar	input buffered crossbar
cost/port	>$1000/port	$250/port	-	?	-
virtual networks	1[c]	?	?	1	2
link flow control unit	packet	flit	?	flit?	flit
link flow control method	credit	optimistic	?	optimistic	credit
switching algorithm	VCT	wormhole	-	wormhole	virtual channel?
virtual channels	No	No		No	2/network
switch buffer size	6 packets/port (912bytes/port)	?	?	64 bytes/port	256bytes/VC/ port
link error recovery	end to end re-transmission	?	?	?	link level re-transmission
error checking	EDC	CRC	?	CRC	CRC
switch technology parameters	0.35micron 200MHz 14.2x12.9mm 1.9M transistor 15W ceramic flip chip BGA		0.5micron 100MHz 7.3x7.3mm ? 2.5W PBGA		0.5micron 100MHz 12.7x12.7mm 850K gates 29W ceramic flip chip BGA

a.Synfinity-I does not have enough virtual channels to support torus type topologies.
b.ServerNet cannot support cyclic routes, generally limited to acyclic networks.
c.Synfinity-I uses its second virtual channel for network ACKs. The application only sees 1 VC.

5 Analysis and Future Directions

5.1 Network Size, Routing and Topologies

Maximum nodes supported ranges from 128 to millions. However, like most commodity systems, the volume is in the low end, thus, configurations with more than 30 nodes are rare, with the volume market being less than 10[1]. In addition, most applications have high peak bandwidth, but usually moderate average bandwidth requirements, and thus, concentration is a cost effective solution, but has the effect of requiring small interconnects. This trend will continue into the future, having two effects.

First, the growth rate in the maximum number of nodes that need

1. Customers who buy switch based systems over bus based systems at the low end want the scalability that switch based systems provide.

to be supported will be slow. Secondly, the need for switch based concentration at the board level will dramatically increase, starting with the next generation of high end microprocessors.

As a result of the low maximum number of nodes that need to be supported, topologies and routing algorithms are not that important. The ability to support flexible routing requirements is the most important factor here. Thus, table-based routing is the algorithm of choice. In the examples we examined, there was a split between source routing and switch table routing. Either of these will work; vendors will make incremental improvements to allow more path diversity and larger numbers of nodes without dramatically increasing table size.

5.2 Performance

All of these switches except the SCI-based ones, use advanced techniques such as virtual cut through or wormhole switching to reduce latency and improve throughput. Examining the flow control and switching choices by each vendor, shows that each vendor chose a different combination of methods. This probably reflects the fact that these choices mostly affect cost/performance and are secondary to the main performance considerations of raw bandwidth and fall through latency.

It is likely that vendors will continue to adjust these parameters to best fit their needs. Since these architectural features are mostly hidden from the user, vendors will probably feel free to change their choices as they develop their technologies. So, in the future, we can expect to see continuing development in this area, with probably no real consensus on the best choices.

5.3 Application Interface

As was mentioned previously, one of the most important considerations in choosing a high performance interconnect is the services it provides to the application. Currently, there are several useful features that an interconnect can provide besides the basic functionality of switching:

- Reliable, in-order, exactly once packet delivery (i.e. the backplane bus model.)
- End to end ordering guarantees with multiple virtual networks. (Useful for deadlock avoidance in request/response type protocols such as cache coherence or message passing.)
- Built-in cache coherence protocol (SCI for example.)
- Built-in message engine. (PCI interfaces for example.)
- Physical layer standards conformance (to allow plugging in different vendors components.)

No one network provides all these features and architecting generic application level services into an interconnect does not necessarily make the system design easier or higher performance as was seen with the Sequent SCI design.

In the future, these features will continue to be included. However, beyond the first two, there is no clear consensus on generally useful application level features that will be universally adopted.

5.4 Standards

This is the hardest to judge. SCI has probably been the most successful SAN in terms of revenue generated for any high performance commodity SAN, however, it has no real future in its present form due to its high overhead and lack of scalability.

However, it also seems clear that as commodity, high performance SANs become more common, the need for standardized physical layers, protocols, and application interfaces will become greater. High volume customers will simply not want to get locked into using a single companies non-standard interconnect with the possibility that company may go out of business. Designing an interconnect into a system requires too much time and work for companies to be able to shift vendors easily.

In the near future, a switched I/O SAN which replaces PCI will become a big SAN application. This SAN will be either NGIO or Future I/O or maybe both. It is likely that NGIO/Future I/O components will be integrated into other products beyond their originally intended targets, similar to how SCI and ServerNet have evolved into different markets than they were originally intended.

There is also a proposed IEEE standard for SANs called IEEE 1355, which may encompass all of these things. In summary, for current vendors, this is probably the biggest issue in designing their next generation interconnect architectures: which, if any, standard to support.

5.5 Business Models

On the business side, it is also necessary to determine what model best suits selling commodity, high performance interconnects. As was previously suggested, companies make too much of an investment in a high performance interconnect to risk buying it from a company that may not be around tomorrow. Thus, it is a big advantage to be a large company. Compaq and Fujitsu have the advantage here.

In addition, large semiconductor companies have a technology edge over fab-less companies, especially small companies. This allows them to bring out faster systems earlier than their competitors. It also means that they can be more aggressive in their circuit designs than fab-less companies who must rely on external vendors who typically characterize their libraries more conservatively than an in-house semiconductor company.

Small companies can survive by selling standards based components. However, as in the Dolphin and Vitesse cases, they don't have much value added. Myricom survives by not selling into the commodity market, but instead concentrating on higher end systems and "one-off" prototype systems. While Dolphin should continue to be successful, the future for Myricom is less clear.

The future is also less clear for Vitesse and its GaAs solution. Exotic technologies (i.e. non-CMOS technologies) are always going to be relatively expensive and have difficulty gaining acceptance. However, in the next generation, optical interconnects, which generally use exotic technologies will likely carve out a large part of the market due to their superior performance, especially if the cost is roughly the same as a high performance electrical link of today.

In the long term, each of these companies should survive in its niche. However, it is expected that the larger companies will begin

to dominate as they focus more on standards based systems, especially NGIO or Future I/O.

5.6 Beyond SANs towards BANs and CANs

Board area networks and chip area networks are the logical successors to SANs as pin data rates continue to go up. The trend toward BANs has already started with Rambus using high speed signalling techniques to address the DRAM pin data rate problem.

Rambus' solution, although touted as a general solution to the DRAM data rate problem, really only addresses a subset of this problem. Rambus really only works for small DRAM sizes typical in desktop or embedded systems. When moving to server sized memory arrays, Rambus provides no better pin data rates overall than soon to be available DDR-SDRAM. Consequently, it is not really the preferred choice for implementing server memory systems. Except for the fact that Intel chipsets will only support Rambus in the future, most PC servers would probably not use Rambus.

In addition, although technically, Rambus makes sense for desktop and embedded systems such as graphics due to its higher level of integration, it has a high cost compared to traditional SDRAMs due to the overhead of its link drivers. Thus, for low end systems where it makes the most technical sense, it makes the least business sense. Thus, Rambus success really lies in Intel's commitment to supporting it as the primary choice for future PC memory systems. As volume goes up costs should go down, but will never be as low as standard SDRAMs due to the licensing costs that Rambus charges. However, Rambus future is still quite good because, as has been mentioned throughout this paper, those companies that are paying the price of overcoming the high cost and power consumption of high speed signalling now will be better off in the future, since it is inevitable that anybody who wants to be in this business will have to go through this same process.

We have also seen the need for board level networks that replace processor busses as concentrators for SANs as an emerging application. One example that already exists is Sun's UPA architecture. This is a globally synchronous interface, however, with a separate control chip and data crossbar, thus, it only marginally qualifies as an interconnect. In the near future, however, we will see true packet switched board level cache coherent concentrators based on high-speed source synchronous signalling technology.

6 Conclusion

We have presented high performance, commodity interconnects as a technology (high-speed signalling) meeting a need (shared memory, clustering, I/O etc.) We analyzed the basic signalling technology and issues, analyzed the application characteristics and then presented several examples of commodity interconnects designed to bridge this gap. We compared each interconnect to see how well it met the needs and found that no existing interconnect exactly was the right solution. We then analyzed these solutions to determine future trends for SANs and beyond.

In summary, we found:

- high speed signalling requires expensive packaging and consumes a lot of power and that companies that deal with this today will be well positioned for the future.
- Parallel optical links will be the next major technology that drives SAN technology.
- Most commodity network sizes are small, less than 30 nodes, and the higher the volume, the lower the number of nodes. Therefore, commodity interconnects should be optimized for small sizes.
- Switch architecture decisions such as routing, flow control, etc. are secondary to the basic cost/performance model.
- Large companies with their own foundries and lots of circuit design experience are best positioned technically to take advantage of the explosion in commodity SAN applications that is likely to occur in the near future.
- In the future, SANs will adhere to standards at all levels, and companies will have to offer standards based products to survive.
- Board area interconnect applications are just around the corner: DRDRAM and processor node concentration being the first applications.

7 References

[1] Seitz, C., "Making Interconnect Commodity", *Hot Interconnects IV Symposium*, Stanford, CA, 1996.

[2] Dally, W.J., "Interconnect-Oriented Architecture and Circuits", Computer Systems Lab, Stanford University, 1999.

[3] Patterson, D. et al., "A Case for Networks of Workstations (NOW)", *Hot Interconnects II Symposium*, Stanford, CA, 1994.

[4] Cheng, C., Yuan, L., "Electrical Design of the XDBus Using Low Voltage Swing CMOS (GTL) in the SparcCenter 2000 Server," *Hot Interconnects I Symposium*, Stanford, CA, 1993.

[5] Singhal, A. et. al., "Gigaplane: A High Performance Bus for large SMPs," *Hot Interconnects IV Symposium*, Stanford, CA, 1996.

[6] Bugge, H. and Alnes, K., "The SCI NodeChip, a 500MByte per se Virtual Backplane Chip," *Hot Interconnects I Symposium*, Stanford, CA, 1993.

[7] Duzett, B., "nCUBE3 Communications Architecture," *Hot Interconnects II Symposium*, Stanford, CA, 1994.

[8] Slater, M., "Rambus Unveils Revolutionary Memory Interface," *Microprocessor Report*, March 4, 1992.

[9] Engebretsen, D. et. al., "Using a 2GBytes/Second Optical Data Link to Extended High-Bandwidth Busses," *Hot Interconnects III Symposium*, Stanford, CA, 1995.

[10] Galles, M. "Scalable Pipelined Interconnect for Distributed Endpoint Routing," *Hot Interconnects IV Symposium*, Stanford, CA, 1996.

[11] Dally, W., Carvey, P., Dennison, L, "The Avici Terabit Switch/Router." *Hot Interconnects VI Symposium*, Stanford, CA, 1998.

[12] Horowitz, M., "The Limits of Electrical Signalling," *Hot Interconnects V Symposium*, Stanford, CA, 1997.

[13] Crisp, R., "Direct Rambus Technology: The New Main Memory Standard," *IEEE Micro*, Dec. 1997.

[14] Weber, W-D., et. al., “The Mercury Interconnect Architecture: A Cost-effective Infrastructure for High-performance Servers,” *Proceedings of the 24th Annual International Symposium on Computer Architecture*, Denver, Co, 1997.

[15] Mu, A. et. al., “A 9.6GigaBytes/s Throughput Plesiochronous Routing Chip,” *Digest of Papers of the 41st IEEE Computer Society International Conference*, Feb. 1996.

[16] Mu. A. et al., “A 285MHz 6-port Plesiochronous Router Chip with Non-Blocking Cross-Bar Switch,” *1996 Symposium on VLSI Circuits: Digest of Technical Papers*, 1996.

[17] Gotoh, K, et. al., “A 2B Parallel 1.25Gb/s Interconnect I/O Interface with Self-Configurable Link and Plesiochronous Clocking,” *1999 IEEE International Solid-State Circuits Conference*, San Francisco, CA, 1999.

[18] Takahashi, T., et. al., “110GB/s Simultaneous Bi-Directional Transceiver Logic Synchronized with a System Clock,” *1999 IEEE International Solid-State Circuits Conference*, San Francisco, CA, 1999.

[19] Haycock, M. and Mooney, R., “A 2.5Gb/s Bidirectional Signaling Technology,” *Hot Interconnects V Symposium*, Stanford, CA, 1997.

[20] Buckman, L. et. al., “Parallel Optical Interconnects,” *Hot Interconnects VI Symposium*, Stanford, CA, 1998.

[21] Kaminishi, K. et. al., “Si Bipolar 3.3V Transmitter/Receiver IC Chip Set for 1Bg/s 12-Channel Parallel Optical Interconnects,” *1999 IEEE International Solid-State Circuits Conference*, San Francisco, CA 1999.

[22] Hatakeyama, I., “A 12-Channel Data-Format-Free 1Gb/s/ch Parallel Optical Receiver,” *1999 IEEE International Solid-State Circuits Conference*, San Francisco, CA, 1999.

[23] Theorin, C. et. al., “A ‘Seamless Migration’ to VCSEL-Based Optical Data Links,” *Hot Interconnects V Symposium*, Stanford, CA, 1997.

[24] Lenoski, D. and Weber, W-D., *Scalable Shared-Memory Multiprocessing*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1995.

[25] Lovett, T. and Clapp, R., “STiNG: A CC-NUMA Computer System for the Commercial Marketplace,” *Proceedings of the 23rd International Symposium on Computer Architecture*, Somewhere, 1996.

[26] Clark, R., “SCI Interconnect Chipset and Adapter: Building Large Scale Enterprise Servers with Pentium Pro SHV Nodes,” *Hot Interconnects IV Symposium*, Stanford, CA, 1996.

[27] Singhal, A. et. al., “Gigaplane (TM): A High Performance Bus for Large SMPs,” *Hot Interconnects IV Symposium*, Stanford, CA, 1996.

[28] Dunning, D. and Regnier, G., “The Virtual Interface Architecture,” *Hot Interconnects V Symposium*, Stanford, CA, 1997.

[29] Ibel, M. et. al., “High-Performance Cluster Computing using SCI,” *Hot Interconnects V Symposium*, Stanford, CA, 1997.

[30] McMillen, R., “The BYNET v2.0 Interconnection Network,” *Hot Interconnects VI Symposium*, Stanford, CA, 1998.

[31] Mukherjee, S. and Hill, M., “A Case for Making Network Interfaces Less Peripheral,” *Hot Interconnects V Symposium*, Stanford, CA, 1997.

[32] Seitz, C., “Myrinet -- A Gigabit-per-Second Local-Area Network,” *Hot Interconnects II Symposium*, Stanford, CA, 1994.

[33] Seitz, C. et. al., “The Design of the CalTech Mosaic C. Multicomputer,” *Proceedings of the Washington Symposium on Integrated Systems*, Seattle, WA, 1993.

[34] Cohen, D. et. al., “The Use of Message-Based Multicomputer Components to Construct Gigabit Networks,” *ACM Computer Communication Review*, July 1993.

[35] James, D., “The Scalable Coherent Interface: Scaling to High-Performance Systems,” *Spring COMPCON 94, Digest of Papers*, San Francisco, CA, 1994.

[36] Alnes, K. and Johnsen, B., “Gigabit SCI Cluster Interfaces,” *Hot Interconnects III Symposium*, Stanford, CA, 1995.

[37] Gustad, P. et. al., “A Low Cost CMOS 500MByte/Sec SCI Link Controller,” *Hot Interconnects IV Symposium*, Stanford, CA, 1996.

[38] Kibria, K., “Lightweight SCI, the low cost interconnect solution,” *Wescon IC Expo/96*, 1996.

[39] Cecchi, D. et. al., “A 1.0 GB/Second SCI Link in 0.8u BiCMOS,” *1995 IEEE International Solid-State Circuits Conference*, San Francisco, CA, 1995.

[40] Scott, S., “The SCX Channel: A new supercomputer-class system interconnect,” *Hot Interconnects III Symposium*, Stanford, CA, 1995.

[41] Horst, R., “TNet: A Reliable System Area Network for I/O and IPC,” *Hot Interconnects II Symposium*, Stanford, CA, 1994.

[42] Horst, R., and Garcia, D., “Servernet SAN I/O Architecture,” *Hot Interconnects V Symposium*, Stanford, CA, 1997.

[43] Laudon, J. and Lenoski, D., “The SGI Origin: A ccNUMA Highly Scalable Server,” *Proceedings of the 24th International Symposium on Computer Architecture*, Somewhere, 1997.