Interconnect Analysis: 10GigE and InfiniBand in High Performance Computing

HPC ADVISORY COUNCIL
NETWORK OF EXPERTISE

WHITE PAPER

Highlights:

- There is a large number of HPC applications that need the lowest possible latency for best performance or the highest bandwidth (for example Oil&Gas applications as well as weather related applications)
  - o 10GigE has 5-6 times the latency of InfiniBand
  - o InfiniBand has 3.7x the throughput of 10GigE
  - o Beyond 1-8 nodes, many times InfiniBand provides much better performance than 10GigE and the performance difference grows rapidly as the number of nodes increases
- There are some HPC applications that are not latency sensitive. For example, gene sequencing and some bioinformatics applications are not sensitive to latency and scale well with TCP based networks including GigE and 10GigE.
  - o For these applications, both 10GigE and InfiniBand are appropriate solutions
- Putting HPC message passing traffic and storage traffic on a single TCP network may not provide enough data throughput for either.
  - o Many HPC applications are IOPS driven and need a low-latency network for best performance. 10GigE networks have 3-4 times the latency of InfiniBand.
  - o For most HPC applications it is recommended to use InfiniBand when storage and computational traffic is combined.
- There are a number of examples that show 10GigE has limited scalability for HPC applications and InfiniBand proves to be a better performance, price/performance, and power solution than 10GigE.

10 Gigabit Ethernet Background:

10GigE was standardized in 2002. Since then people have been watching it in the hope that it would become a commodity network as rapidly as GigE. The excitement behind 10GigE is that it is TCP based and virtually all admins understand TCP and are familiar with it.

Recall that TCP (http://en.wikipedia.org/wiki/Transmission_Control_Protocol) does not guarantee delivery of a packet. Indeed that was why it was developed – so a packet could be dropped or missed or whatever, and the protocol would just retransmit the missing packet. On the other hand InfiniBand is guaranteed to have in-order packet delivery with no dropped packets (http://www.mellanox.com/pdf/whitepapers/IB_vs_Ethernet_Clustering_WP_100.pdf). InfiniBand was designed for high performance networks that required guaranteed packet delivery (http://en.wikipedia.org/wiki/Infiniband).

In addition, if you build multiple tiers of Ethernet networks, as you would with medium to large number of nodes, you would need to ensure that every switch is running the spanning tree protocol (http://en.wikipedia.org/wiki/Spanning_tree_protocol). It makes sure there are no loops in the network and every node is reachable. This can introduce a fair amount of latency into the network (typically people recommend for simple Ethernet networks to turn off spanning tree to get better performance). On the other hand, InfiniBand does not add an additional protocol to ensure there are no loops. However, you have to design the network not to have any loops (loops mean that packets just move around the network without being delivered to the correct host). Recent IB fabric management tools have become much more adept at finding and adjusting to possible loops.

Recently, there have been efforts at what was called Data Center Ethernet (DCE - http://en.wikipedia.org/wiki/Data_Center_Ethernet) or CEE, and now it is being referred to as DCB (Data Center Bridging). One of the goals of DCB is to help with reliability of packet delivery (guaranteed in-order packet delivery). This might help in HPC as well. For example, the DCB initiative is proposing additions to the IEEE standards that allow for lossless TCP transmission (matching IB) and for better routing schemes to reduce the impact of spanning tree. But DCB has not yet been approved and its specification continues to evolve so it is not a standard at this time. In addition, if the lossless packet delivery is used it reduces the effective bandwidth of the network because of "pauses" that are needed to make sure a packet is delivered.

With these introductory comments, let's look at 10GigE performance, since HPC is all about performance (well usually).

Performance:

One of the key tenants of HPC is performance. This means "how fast does my application run?" not the performance of an individual component of the system. So when designing HPC systems one has to understand what aspects can heavily influence application performance. The primary aspects that this paper will discuss are network latency (there are several pieces to this), N/2, network throughput, and network topologies. These aspects are the most common parameters that influence application performance from a networking perspective.

Latency:

Latency is an important aspect of HPC systems because it can be a huge driver in performance and scalability for many applications and application classes. For HPC applications latency is the time it takes a zero-length packet to be transmitted from one process on one node, through the NIC, through the switch, to a NIC on a second node and to a process on the second node. This measure is very important because this is the latency that an application will experience (in the best case since no application sends zero length packets), consequently driving application performance. Also note that this can include the latency in the kernel and in the packet creation and teardown which can be considerable for various protocols such as TCP.

Sometimes latency will be stated as the latency through the switch which only gives a partial measure of the latency and isn't a good guide for application performance. The same is true for NICs. Stating the latency of just the NIC does not give an accurate enough picture as to the latency that the application experiences and which drives performance.

The only measure of latency that has much meaning is the measure that impacts application performance and scalability. In essence, the latency that the application experiences. This measure, sometimes called end-to-end latency, includes a NIC both at the send and receive nodes, and a switch in between them. It also includes a user-space process on each node that communicates with each other. This represents the configuration that an application actually uses when running.

Below is a table of latency results taken from an application named Netpipe (http://www.scl.ameslab.gov/netpipe). This is a commonly used application for testing latency and bandwidth as well as other measures.

Table 1 – Typical latencies for HPC networks taken from Netpipe results (smallest possible packet size, includes a switch in between two nodes)

Network Solutions	End-to-End Latency (µsec)
GigE	29-100 (47.1 for Netpipe results in Fig. 1)
10GigE (TCP)	12.51 Mellanox ConnectX + 10GigE
DDR InfiniBand	1.72
QDR InfiniBand	1.67

The Netpipe results for 10GigE, QDR InfiniBand, DDR InfiniBand, and GigE are shown in Figure 1.

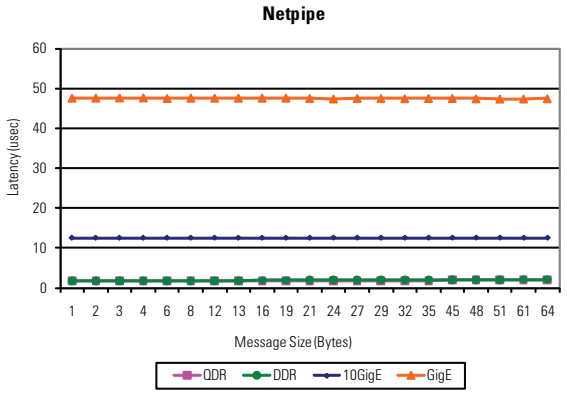


Figure 1 – Netpipe Results for DDR IB, QDR IB, 10GigE

Notice that 10GigE has a latency that is 6.8 times greater than InfiniBand. Please also note that latency also depends on the benchmark. For MPI latency (the de-facto application library for HPC), InfiniBand demonstrates 1.3us latency end-to-end versus 7us with 10GigE end-to-end (a factor of about 5.4).

We can dive into the specific components of the configuration that affect latency to illustrate that stating the latency of a single component does not give you enough information to understand the latency that the application will experience. For example, 10GigE switch latency varies between the different switches. 10GigE switches based on Fulcrum ASICs demonstrates 0.6us latency while others, such as Cisco demonstrates switch latency of 3us. Stating a latency of 0.6us does give you the complete picture where the overall latency may be 5-6 times greater.

Remember that so far this document has referred to latency in the general sense with a single process on one node sending data through a single NIC, through a switch, to a NIC on a second node. But HPC applications are rarely run with a single process per node. After all, there are typically 8 cores on a single node (assuming quad-core processors and dual sockets) so why not use all 8

cores for the HPC application? This means that instead of a single application putting messages onto the network, you will now have 8 processes sending and receiving messages per node. A logical question to ask is, "how does the latency change as the number of processes per node are increased?"

The following graph is a plot the latency as a function of the number of cores. It uses Mellanox DDR IB switches and Mellanox DDR ConnectX HCA's. It runs the Intel MPI benchmark suite (http:/software.intel.com/en-us/articles/intel-mpi-benchmarks) across a range of cores (this used to be called the Pallas benchmark suite).

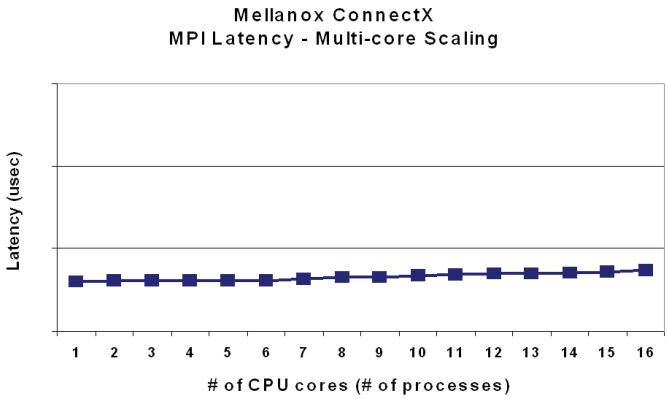


Figure 2 – Latency results using the Intel MPI Benchmarks for 1-16 cores with Mellanox ConnectX HCA's and a Mellanox InfiniBand switch

This graph illustrates that the per core latency stays the same using InfiniBand regardless of the number of cores that are communicating up to the range tested.

Now, let's look at the same type of data for 10GigE to determine what happens when we actually use more than 1 core per node. Figure 3 below is the same basic plot but only over 8 cores for QDR InfiniBand and 10GigE.

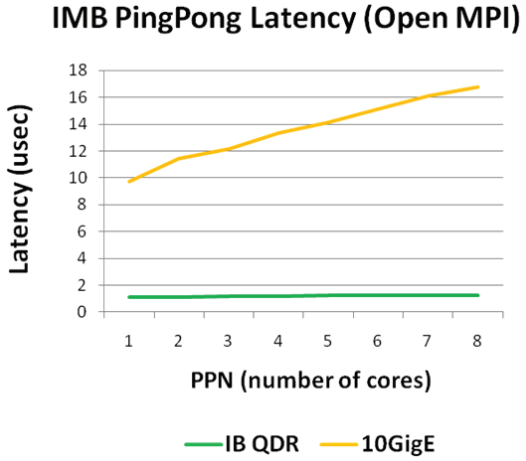


Figure 3 – Latency results using the Intel MPI Benchmarks for 1-8 cores with Mellanox InfiniBand QDR ConnectX HCA's and a Mellanox InfiniBand QDR switch

Notice that the latency of the cores goes up as you increase the number of cores used. From 1 core to 8 cores it increases by about 60%. Consequently even stating the single core latency for networks does give the entire picture. You need information like that in Figure 3 to understand what happens to the latency as you increase the number of cores per node.

But latency isn't the only micro-benchmark that tells us about application performance.

N/2:

Another important network measure is called N/2. It is a very important measure that tells us the smallest packet size that reaches full network speed in one direction. For example, in the case of GigE, 1,000 Mbit/s is the maximum "speed" in one direction (it's actually the bandwidth but it can be converted into time. Bandwidth is a more convenient measure but it is sometimes referred to as "speed").

The reason that N/2 is so important is that applications can sometimes send small packets and many times as the number of nodes in a run of an application increases, the number of smaller packets increases. Therefore if we had a small value of N/2 then it is very likely that the smaller packets would run at full wire speed, increasing performance.

To determine N/2, a plot of N, the packet size, versus bandwidth, is created. There are several applications that can create this type of plot, a common one being Netpipe. Below is an example of such a plot (http://www.clustermonkey.net//content/view/124/33/)

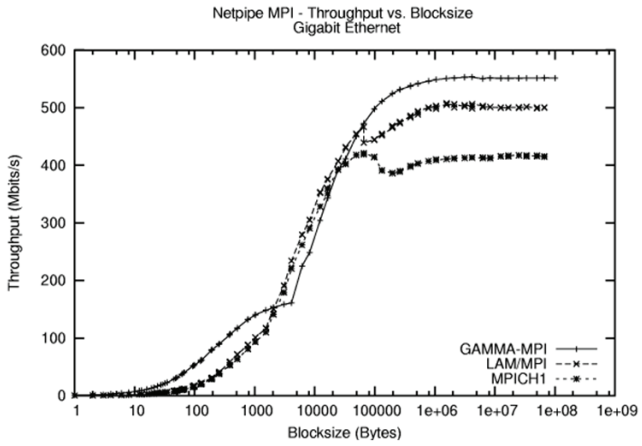


Figure 4 – Sample Netpipe Plot of Throughput (MB/s) vs. Packet Size (BlockSize)

This plot is for 3 different MPI libraries (MPICH1, LAM, and GAMMA-MPI) over a plain GigE network.

Table 1 - N/2 and Max Bandwidth for Figure 4

MPI Protocol	N/2 (bytes)	Maximum BW (Mb/s)	Latency (µs)
MPICH1	2,700	410	~45
LAM/MPI	3,100	500	~38
GAMMA/MPI	10,000	550	~11

Notice that the value of N/2 can vary quite greatly and the lowest N/2 does not always mean the lowest latency. In addition, when comparing N/2 values for networks, it's always appropriate to include the maximum bandwidth since that also impacts the comparison.

Figure 5 below is a combined plot of netpipe results for GigE, 10GigE, DDR InfiniBand, and QDR InfiniBand.

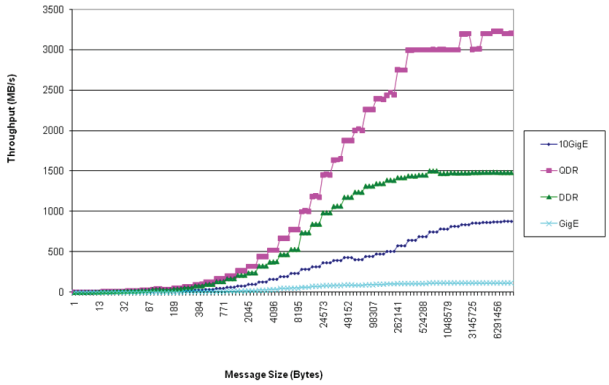


Figure 5 –Netpipe Plot of Throughput (MB/s) vs. Packet Size (Message Size) for GigE, DDR IB, QDR IB, and 10GigE

Table 2 below lists the maximum bandwidth in MB/s, the N/2 value in byes, and the latency in us for Figure 5.

Table 2 - Values for Throughput (MB/s), N/2 (bytes), and Latency for Figure 5

Network Solution	N/2 (bytes)	Maximum BW (MB/s)	Latency (µs)
GigE	12,300	112	47.61
10GigE	98,300	875	12.51
DDR InfiniBand	12,285	1482	1.72
QDR InfiniBand	32,765	3230	1.67

From this data you can see the IB has a much lower N/2 meaning that smaller packets will get to take advantage of the bandwidth of the network. This is also why it is important for max bandwidth to be stated along with N/2. You could have a very good N/2, as in the case of GigE, but the network bandwidth is much lower than 10GigE or InfiniBand. Consequently, a lower N/2 does always mean the best performance for smaller packets.

Comparing DDR and 10GigE you can see that DDR IB has about 70% more bandwidth than 10GigE (that means packets can travel up to 70% faster) and an N/2 that is 8 times smaller, and a latency that is about 6.5 times smaller.

Comparing QDR and 10GigE you can see that QDR IB has about 3.7 times more bandwidth (speed) than 10GigE and an N/2 that is 3 times smaller, and a latency that is about 6.5 times smaller.

N/2 by itself is a good indicator of performance, but it needs to be combined with the max throughput numbers as well. As shown in the graph, InfiniBand provides higher throughput (speed) than 10GigE for every message size.

Network Topologies:

Constant Bi-Sectional Bandwidth (CBB) or Fat Tree networks have emerged as a key ingredient to deliver non-blocking, scalable bandwidth and lowest latency for high performance computing and other large scale data center clusters. In addition to wiring the network to provide a physical CBB network topology, system architects also need to pay close attention to the underlying networking technology. The spanning tree algorithm required by Ethernet layer 2 switches (to solve network loops) is not able to exploit physical CBB fat tree topologies. Moving to expensive layer 3 switches solves the spanning tree related scaling problem, but adds the overhead of additional layer 3 algorithms and store-and-forward switching.

InfiniBand on the other hand combines automatic configuration and flexibility of the forwarding algorithm, to be able to fully take advantage of the underlying CBB network. As such, InfiniBand can scale deterministically, maintaining full wire speed across the cluster, and through the use of simple cut-through switching, helps keep costs down. This enables building systems with multiple layers of switches fairly easily and cost effectively.

Applications:

Discussing micro-benchmarks such as latency, N/2, and throughput is good because it allows us to easily compare networks with simple easy to run benchmarks. However, there is nothing like real applications for comparing network fabrics. At the end of this document is a section entitled "Application test cases" that are applications that have been run on IB and 10GigE, and sometimes GigE. The results are compared in terms of performance and sometimes in terms of power, showing how faster results can actually save the customer power for the same amount of computation. As other benchmarks are run on both IB and 10GigE they will be added to this document.

Application test cases:

The following test cases are taken from the HPC Advisory Council website (http://www.hpcadvisorycouncil.com). The benchmarks and application characterization are a result of a joint project between Dell, AMD, and Mellanox. They compare the performance and in some cases the power consumption and productivity (number of jobs run per day) for 10GigE to DDR InfiniBand. The cases were run on Dell SC1435 using quad-core Shanghai processors.

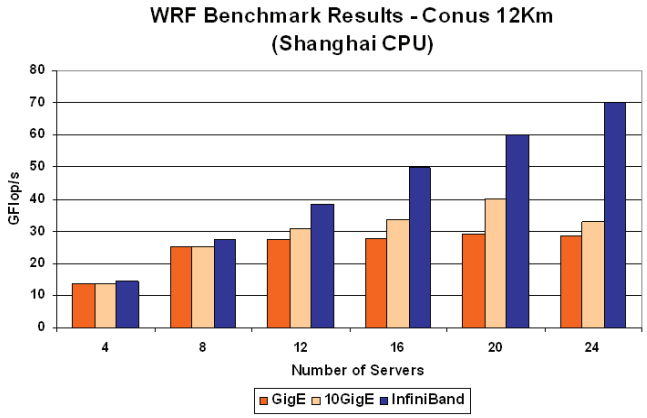
The information is freely available on the HPC Advisory Council and can be given to customers.

The applications examined are:

- WRF
- NAMD
- Schlumberger Eclipse
- LS-Dyna
- MM5
- Star-CCM+
- Star-CD
- MPQC
- GAMESS

WRF:

This is a very common weather modeling application. It is also part of the SpecMPI benchmark and is a very commonly requested benchmark for many RFP's.

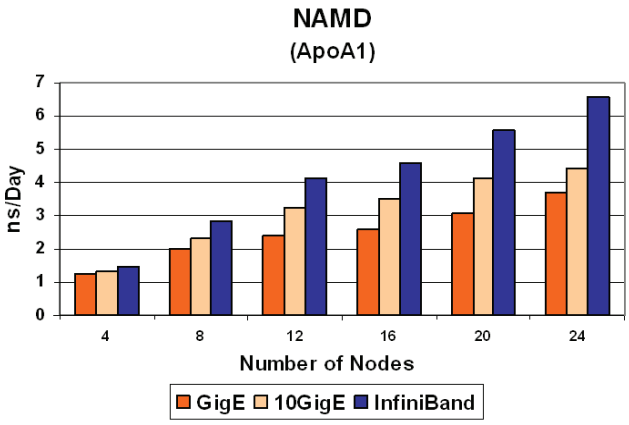


Observations:

- At 24 nodes, DDR IB is 113% faster than 10GigE
- Performance of 10GigE actually decreases from 20 to 24 nodes

NAMD:

NAMD is a molecular modeling application that is freely available. It is quite common in the chemistry and biosciences fields and has a wide range of applications in molecular dynamics.



Observations:

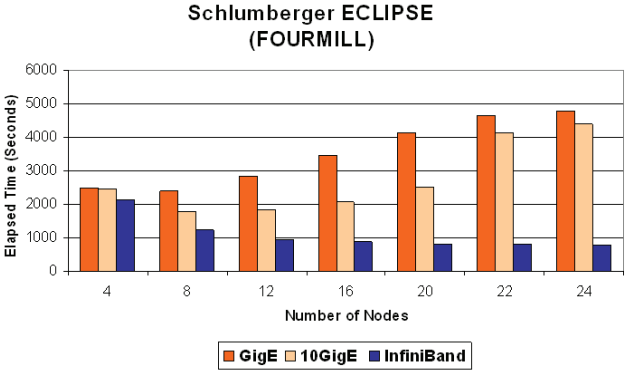
- DDR IB is 49% faster at 24 nodes than 10GigE
- DDR IB scaling is almost linear over the range tested
  - o 10GigE scaling is less than linear

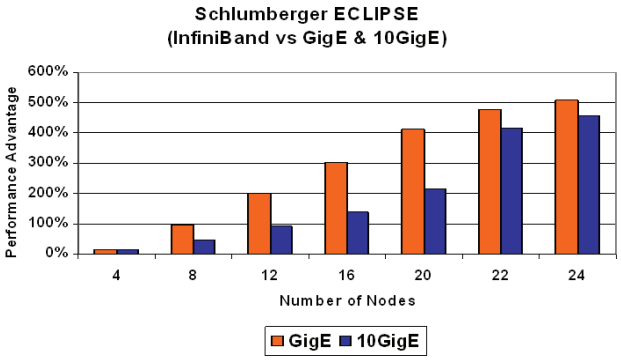
Eclipse:

Eclipse is a commercial product of Schlumberger and is a Oil and Gas reservoir simulation application. It is extremely popular in the Oil and Gas industry. The testing below was performed jointly between Dell, AMD, Mellanox, and Schlumberger.

The first two graphs show the performance for GigE, 10GigE, and IB in elapsed time (seconds) as the number of nodes is varied. The second graph shows the performance advantage IB has over GigE and 10GigE for the same range of nodes.

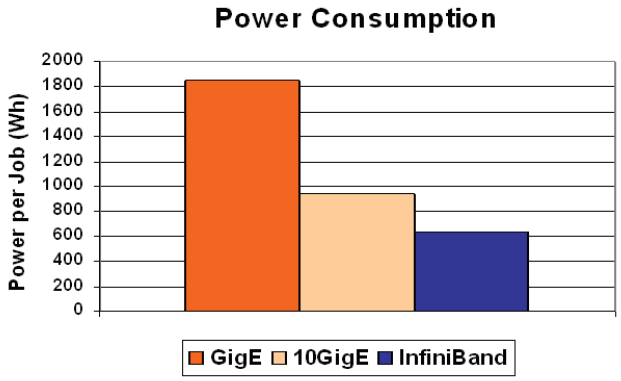
Finally, the last graph shows the power consumption comparison between IB, 10GigE, and GigE when 4 jobs were run on the cluster (the comparison is power per job).





Observations:

- DDR IB is 457% faster than 10GigE at 24 nodes
- Beyond 16 nodes, 10GigE performance improves very little (see first graph)



Observations:

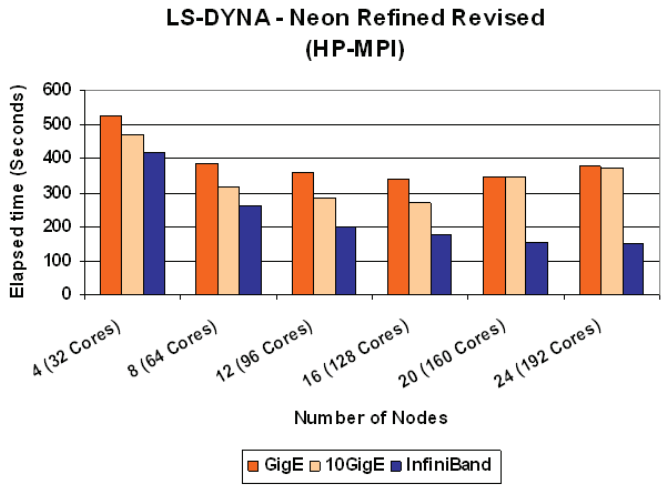
- DDR IB improves power per job by 33% over 10GigE

LS-DYNA:

LS-Dyna is one of the premier impact analysis applications. It is used in many, many industries. For example, aerospace companies use it to examine impact of weapons on a fighter or the impact of a bird on wind screens. The car industry uses it to test car impact. Proctor & Gamble uses it to design detergent containers so that they don't break open if dropped at home or in the store.

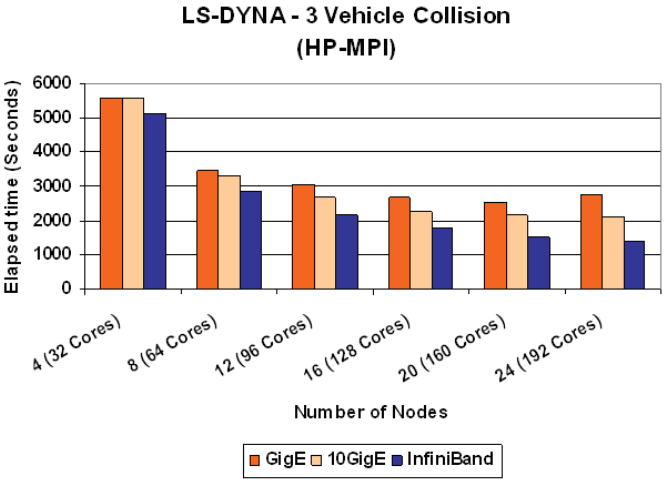
The benchmarks were run under a project with Dell, AMD, and Mellanox. Advice and support was also provided by LSTC, the parent company developing and marketing LS-Dyna, and with GM.

Two different cases were tested in the HPC Advisory Council report. The performance of the two cases as well as the power consumption of the cluster using IB, 10GigE, and GigE are presented.



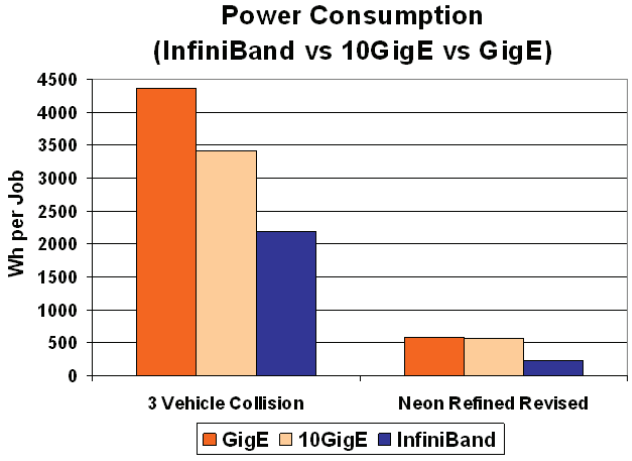
Observations:

- DDR IB is 60% faster than 10GigE at 24 cores
- 10GigE actually gets slower after 16 nodes



Observations:

- DDR IB is again about 60% faster than 10GigE
- 10GigE performance improves very little with increasing core count

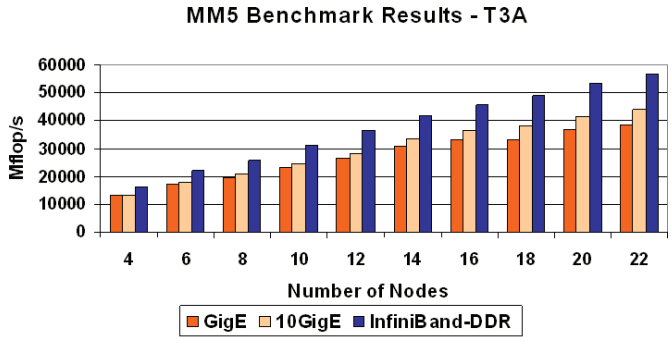


Observations:

- DDR IB improve power per job by 62% over 10GigE for the Neon Refined case
- For the 3 car case, DDR IB is about 33% better than 10GigE (watts per job)

MM5:

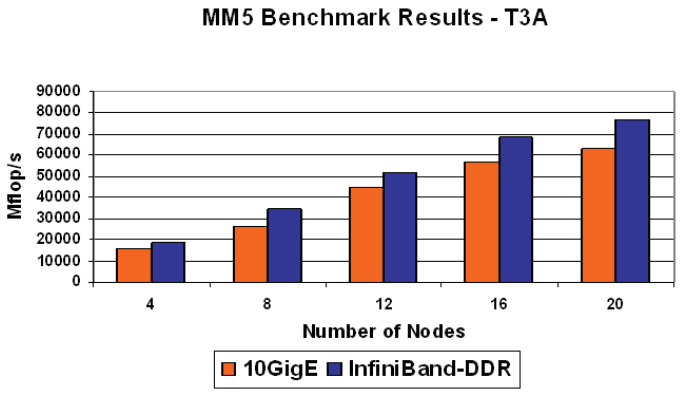
This is a very common weather modeling application



Observation:

- DDR IB is 30% faster than 10GigE at 24 nodes

The following chart shows the productivity – the number of jobs per day that can be run using IB, 10GigE, and GigE for various node counts. For all 3 networks, 4 jobs per cluster were run (this was determined to be the optimal number of jobs for maximum productivity).



Observations:

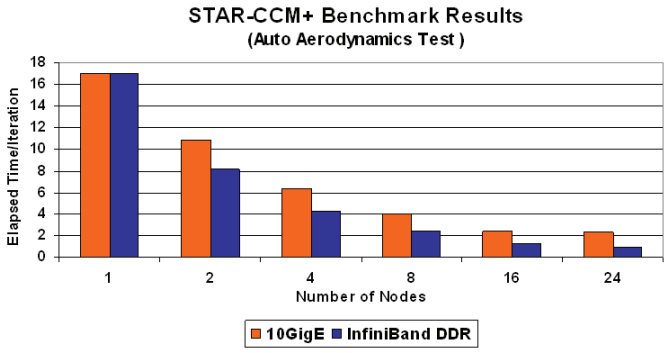
- DDR IB offers more throughput (productivity) than 10GigE (about 25%) when 4 jobs per cluster were run.

Star-CCM+:

Star-CCM+ is a commercial CFD application from CD-Adapco. It is used in a wide variety of industries such as aerospace, automotive, medical (blood flow), marine (ships and submarines), chemical production, etc.

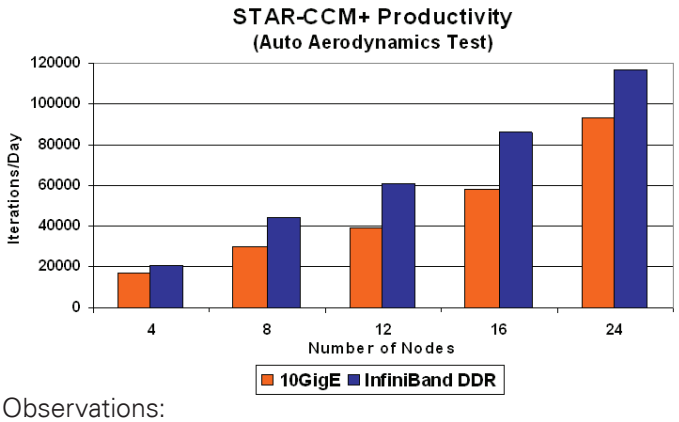
Star-CCM+ is one of the new CFD applications that are commercially available and is enjoying great success and its market share is rapidly increasing.

The first chart shows performance and the second chart shows productivity (jobs per day).



Observations:

- DDR IB is 146% faster than 10GigE at 24 nodes
- 10GigE stops scaling at 16 nodes



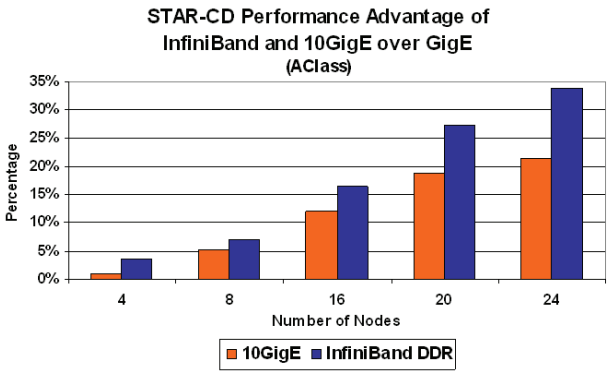
Observations:

- DDR IB has 25% more productivity (jobs per day) than 10GigE

Star-CD:

Star-CD is the flagship CFD package from CD-Adapco. Overall it is second in worldwide use compared to Fluent but in some industries it is the #1 application.

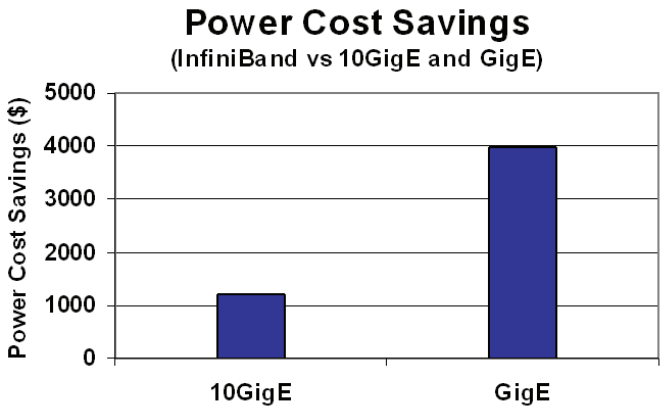
The first chart shows the performance advantage of IB over 10GigE and GigE as a function of the number of nodes.



Observations:

- DDR IB is 22% faster than 10GigE at 24 nodes
- DDR IB allows 10% more jobs per day than 10GigE

The next chart shows the power savings in $$ of IB compared to 10GigE and GigE.

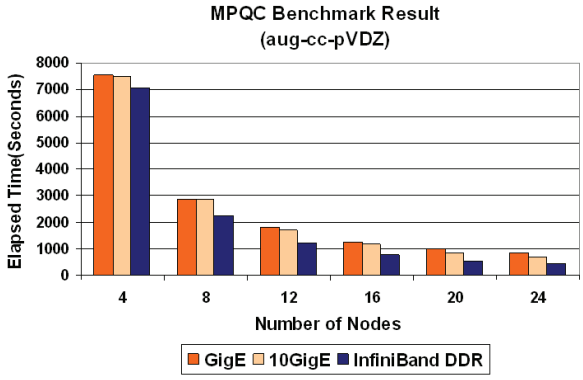


Observation:

- DDR IB saves $1200/year for 24 nodes to run Star-CD compared to 10GigE

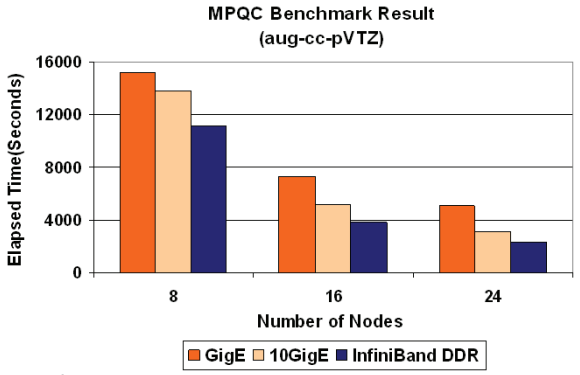
MPQC:

MPQC is the Massively Parallel Quantum Chemistry Program. It computes properties of atoms and molecules from first principles using the time independent Schrödinger equation.



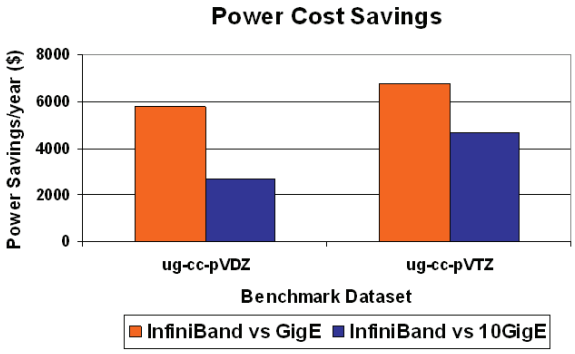
Observations:

- DDR IB is 47% faster than 10GigE at 24 nodes



Observation:

- DDR IB is 38% faster than 10GigE at 24 nodes

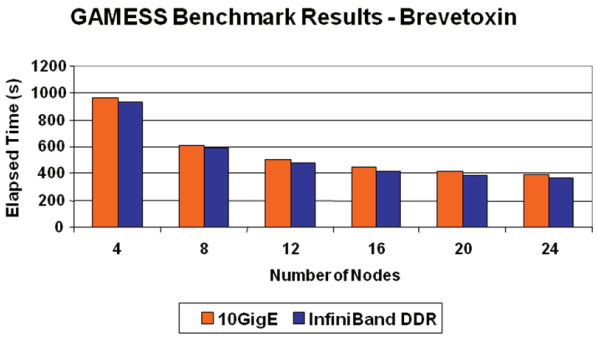


Observation:

- IB saves $3400/year for one test case over 10GigE
- Second test case it saves $4500/year versus 10GigE

GAMESS:

The General Atomic and Molecular Electronic Structure System (GAMESS) is a general ab initio quantum chemistry package.



Observations:

- 10GigE scales well with number of nodes
- 10GigE is only about 6% slower than IB over the range of nodes tested.

HPC ADVISORY COUNCIL
NETWORK OF EXPERTISE
350 Oakmead Pkwy, Sunnyvale, CA 94085
Tel: 408-970-3400 • Fax: 408-970-3403
www.hpcadvisorycouncil.com