24 Lane, 6 Port Gen2 PCIe® Switch Performance Report

89PES24T6G2

Notes

Overview

This document presents performance measurements and benchmarking results for IDT's 89HPES24T6G2 24-lane, 6-port PCI Express® Gen2 switch, a member of IDT's PRECISE™ family of PCI Express Switching solutions. Any two x4 ports, including the upstream port, can be merged to create an 8-lane port. In the evaluation board for the PES24T6G2, ports 0 and 1 have been merged to form a x8 upstream port. The switch is compliant with PCI Express (PCIe®) base specification revision 2.0.

The test vehicle for the PES24T6G2 is the evaluation board IDT89EB24T6G2 which hosts the PES24T6G2. Accompanying the throughput performance metrics are descriptions and methodologies outlining the test setup and procedures.

The nature of tests and the equipment used for these tests varies significantly across the spectrum of tests performed. In the interest of readability and searchability the document is divided into various sections. Each section represents a single test suite that employs a single test setup. A single test suite is capable of highlighting several features of the switch device under test.

Section I provides some insight into issues that can affect the performance of a PCIe device. This includes overhead derived from the protocol, as well as the architectural decisions made while implementing the PCIe device.

Section III describes the performance of the PES24T6G2 with PCIe Gen2 Fiber Channel endpoints and a PCIe Gen1 10GE NIC attached to its downstream ports. Two Emulex PCIe Gen2 FC HBAs, one per downstream port, and one Myricom 10GE NIC are used for this test.

Section III describes the performance of the PES24T6G2 with PCIe Gen2 Fiber Channel endpoints and a PCIe Gen2 Gigabit Ethernet NIC attached to its downstream ports. Two Emulex PCIe Gen2 FC HBAs, one per downstream port, and one Broadcom GE-NIC are used for this test.

Appendix A gives a brief introduction to the SmartBits traffic generator and analyzer and the SmartFlow™ test software package used in conjunction with this test equipment.

Appendix B is an introduction to the software tool called IOmeter that is used in generating some of the storage and networking test results presented in this report.

Revision History

April 6, 2008: Initial version.

June 9, 2008: Moved previous section II to section III and added a new section II,.

SECTION I: PCIe Performance Basics

The PES24T6G2 primarily serves the purpose of high-performance I/O connectivity expansion in a typical system. The PES24T6G2 offers six 4-lane ports, and it is possible to merge any two 4-lane ports to create a single 8-lane port in this device. In fact, the evaluation platform used to run performance testing for this device has three downstream ports per the board design - one 8-lane port and two 4-lane ports.

Given that nothing ever comes for free, it is presumed that the addition of a port has some "cost" associated with it in the form of real estate on the system board, power/heat, design complexity, support circuitry/devices (clocks, hot plug controllers, EEPROMs, power regulators, jumpers, etc.), signal integrity, or adverse effects on throughput/latency. All but the last item in this list are unavoidable to some extent. It is the impact on throughput and latency (system performance in general) that is the least intuitive to predict without a reasonable understanding of the system and switching device architecture, the usage model of the switching device, and some basic understanding of the PCIe protocol itself. In this section, some of these elements are introduced to the users of the PES24T6G2, specifically those users who are new to PCIe and switching. Advanced users of PCIe and switches may skip the reminder of this section.

What Does Performance Mean?

PCIe switch performance can mean different things to different users. The following introduction to some basic terminology may clarify what ought to be important when selecting a switch for your system design.

Throughput

"Raw throughput" refers to the total number of bits that pass through the switch in a given period of time, regardless of function, source, or destination. The PES24T6G2 is designed to handle 5 Gigabits per second (Gbps) of raw throughput in each direction on each of its lanes. This switch is designed for IO expansion (or fan-out) where the traffic flows to and from the root complex via the switch, and as such the maximum width of the upstream port indicates the maximum throughput that this switch can achieve. This results in (5 Gbps) x (2 directions) x 8 (lanes) = 80 Gbps of raw switching capacity. In reality, the switch is not required to "switch" this amount of data, as seen below.

PCI express data bytes undergo 8b/10b encoding. Discussion of the 8b/10b mechanism is beyond the scope of this document. It is sufficient to note that two out of every ten bits passing across a PCIe link do not contribute to any meaningful user data and are stripped off before the data enters the switch core. Therefore, this 20% overhead must be deducted from the raw throughput that the switch must support in terms of actual switching capacity. For the PES24T6G2, the ideal "switch throughput" now becomes 80% of 80 Gbps, i.e. 64 Gbps, assuming simultaneous bidirectional traffic on all ports.

However, there is more overhead at play. Every payload packet (actual user data) is preceded and followed by a variable number of bytes as required by the PCIe protocol. These bytes include the frame K-code, sequence number, TLP header, optional ECRC, and LCRC. This is the "framing" overhead (see Figure 1).

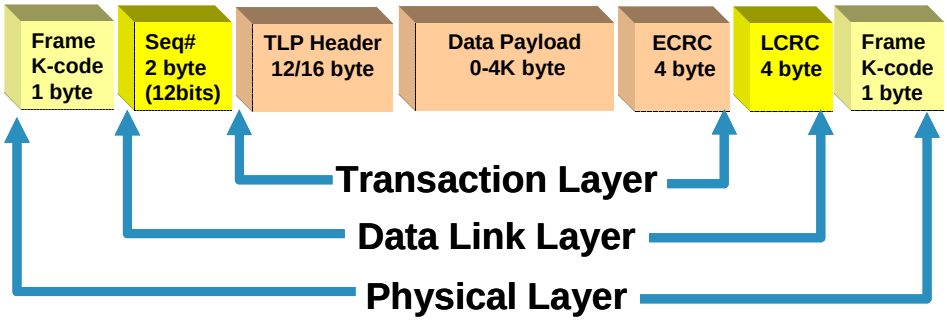

Figure 1 Framing Overhead in a Typical Transaction Packet

Figure 2 shows the effect of framing overhead on useful bandwidth for payloads at different PCIe link widths. A 20 byte overhead is assumed per payload packet for the purpose of this chart. This includes 1 byte of start of packet code, 2 bytes of sequence number, 12 bytes of TLP header, 4 bytes of LCRC, and 1 byte of end of packet code.

So, for example, on a x8 link, at 5 Gbps per lane per direction, raw bidirectional bandwidth is 80 Gbps. Upon removing the 8b/10b overhead, the useful theoretical maximum bandwidth available is 64 Gbps. Similarly, the theoretical maximum useful bandwidth for a x4 link it is 32 Gbps, for a x2 link it is 16 Gbps and for a x1 link it is 8 Gbps.

To understand the calculations behind the chart shown in the figure, let us pick an example of a 64 byte payload packet on a x8 link to see how we come up with the corresponding data point on the chart. Total packet size with overhead becomes 84 bytes on account of the 20 byte overhead explained above. 64 Gbps (giga **bits** per second) of useful bandwidth is the same as 8 GBps (giga **bytes** per second). This is the same as 8000 MBps (mega bytes per second). In terms of packets, this means 8000/84 (i.e. 95.24) million packets. Payload bandwidth for 95.24 million packets is 95.24 multiplied by 64 bytes per packet, or a payload bandwidth of 48.76 Gbps. This is the 64 byte payload data point plotted on the x8 link chart in Figure 2). As seen in the chart, it is possible to achieve close to 64 Gbps (the theoretical maximum for a x8 link) under ideal conditions for payloads larger than 512 bytes.

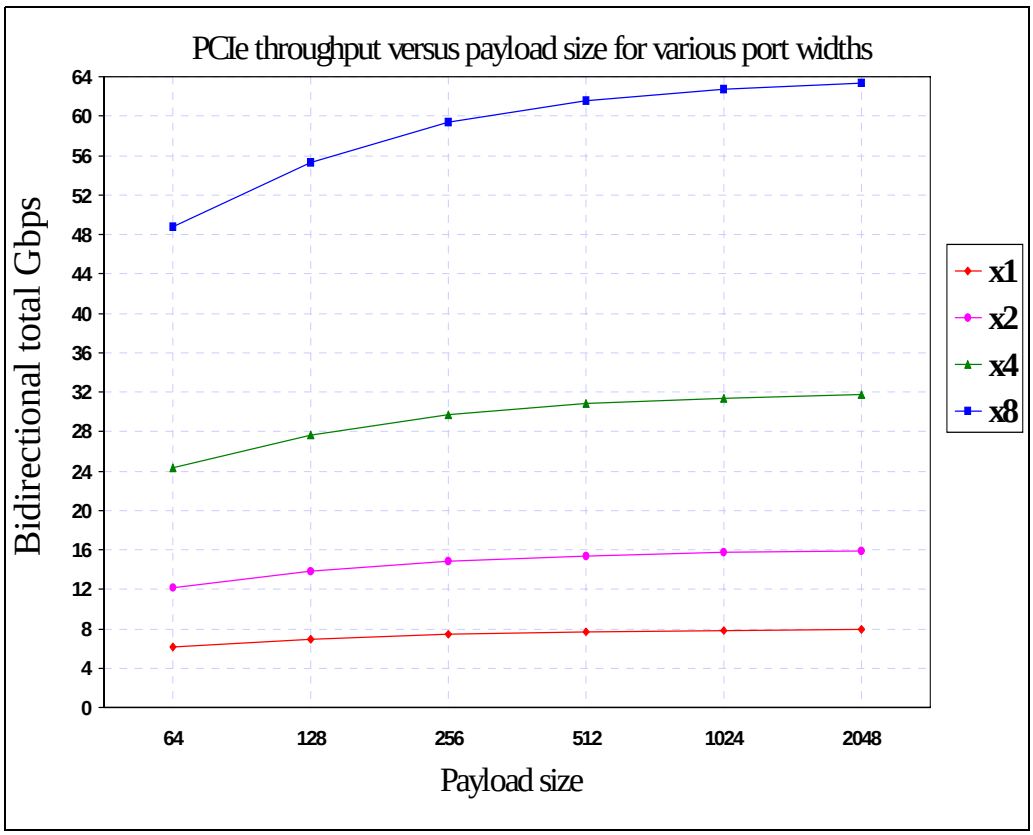

Figure 2 Effect of Framing Overhead on Link Efficiency

As calculated previously, at 64 byte payloads, the maximum throughput achievable on a x8 link is a bit over 48 Gbps. This means that out of the 64 Gbps useful bandwidth available, approximately 16 Gbps is spent on PCI Express framing overhead and approximately 48 Gbps on payload of 64 bytes payload per packet. This implies close to 75% efficiency on the “wire” (link). Since this framing overhead is constant irrespective of the link width, the wire efficiency is independent of link width in ideal conditions. For those who like to think in terms of wire efficiency as opposed to actual bytes or bits per second bandwidth, Figure 3 can help.

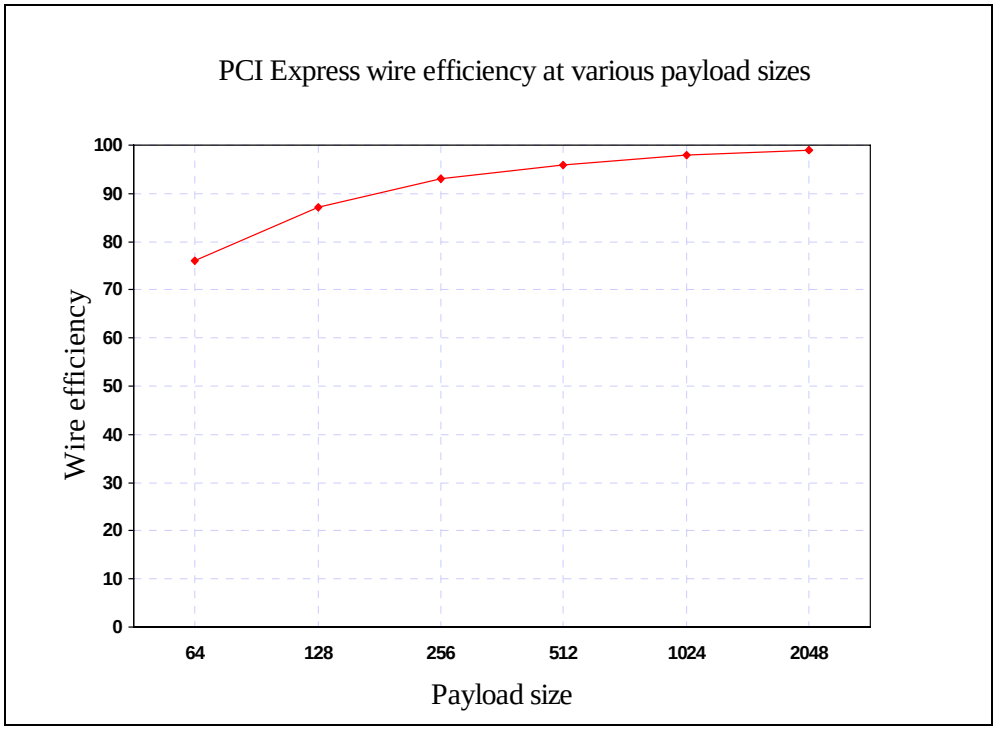

Figure 3 Data Path Efficiency of a PCI Express Link

There is more overhead to be considered in addition to the framing overhead. “Switch utilization” is the “switch throughput” described up until this point, less the overhead associated with the PCIe protocol infrastructure. Examples of this type of overhead traffic are TLPs containing no user data (messages related to interrupts, errors, hot plug, power management or vendor defined messages) and eight types of DLLPs (Ack/NAK, flow control, etc.). This overhead is variable in nature and can sometimes be fine tuned to meet system requirements by modifying the switch settings. Examples of such settings are, the ratio of ACK/NAKs to total packets, frequency of flow control updates, etc. In general, one can expect this overhead to be up to as much as 15% of switch throughput in several real life systems. So, for example, in a x8 link across the switch, for 64 byte payload size, starting from raw bits entering the switch as the base count, 20% is lost in 8b/10 encoding, 25% is lost in framing overhead and approximately 15% may be lost in other protocol overhead as described above.

A pictorial representation of the impact of this additional overhead is shown in Figure 4. This is similar to Figure 3 but also adds another line to the chart showing the effect of the additional DLLPs. The assumption here is that there are two DLLPs of 8 bytes each sent for every 4 TLPs. This equates to 16 bytes worth of DLLPs per 4 TLPs, or on average 4 bytes of DLLP overhead per TLP. This adds to the 20 bytes of framing overhead used previously as an example.

Clearly, the impact of fixed overheads such as these is minimized when the payloads are larger than 256 bytes.

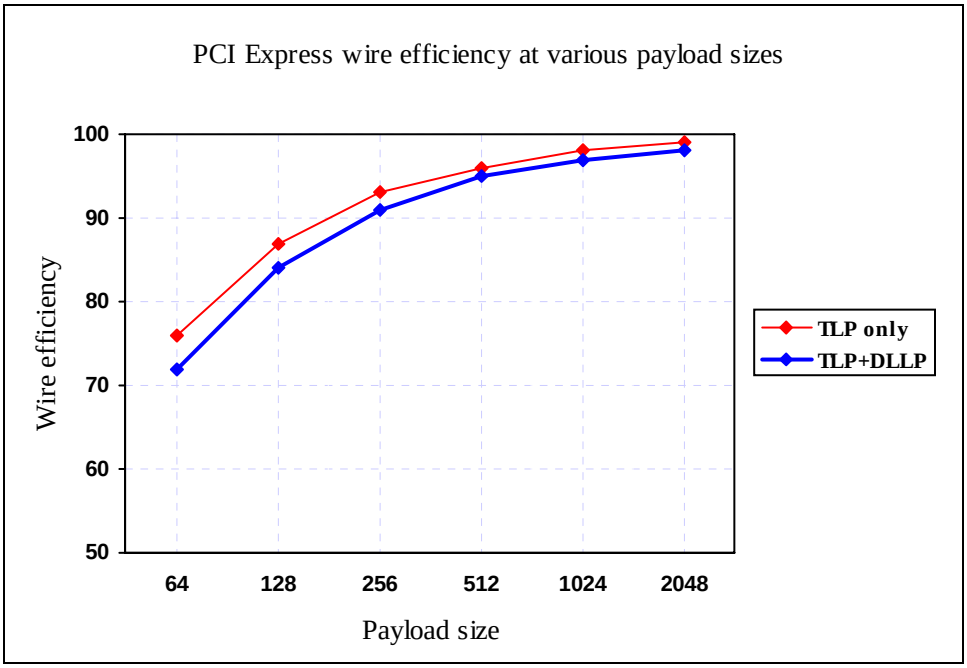

Figure 4 Effect of DLLP Overhead on Data Path Efficiency of a PCI Express Link

Latency

A different indicator of the performance of a switch is the switch "latency", which is defined as the time spent by a bit within the switch from the moment it enters the switch to the moment it exits. The latency number, typically low hundreds of nanoseconds, can be affected by several parameters including, but not limited to, switch architecture, traffic pattern, state of the switch in terms of loading, width of the ingress port, and width of the egress port.

It is crucially important to understand what matters and what does not matter when it comes to selecting a PCIe switch on the basis of latency. In general there is little correlation between the latency of a switch and the total throughput it can sustain across all its ports at the same time, which is the metric that truly matters for any system performance. An uninformed chase for a switch with the lowest latency number supplied by a switch vendor can inevitably lead to a wrong decision if no attention is paid to other performance metrics of a switch. Here is why...

Focus on the port width that matters to the application:

Some switch vendors tend to mislead customers by providing latency numbers which can only be realized when the switch is configured for the largest port width a switch can offer. In general, wider the port width, lower the latency. For a 16 lane switch, a vendor may offer a low latency number for data passing through the switch from an 8 lane ingress port to an 8 lane egress port. This information is worthless if your application requires data to move form a 4 lane or 1 lane port to a 4 lane or 1 lane port. The key is to focus on latency for the port widths actually required by your application.

Focus on simultaneous multi-port activity:

If your application requires data to flow simultaneously between several ports of the switch, what matters is the total latency experienced by the last packet within the set of packets attempting to pass through the switch at the same time. A 4 port switch may have up to 4 different packets trying to get through the switch at the same instance, one from each port. If a vendor provides the latency for one data transfer across an empty switch, that information is worthless in a scenario such as this. Insist on the total latency for the last bit of the data attempting to go across the switch in a fully loaded condition.

Impact of Architecture on Switch Performance

Two high-level architectural decisions which will have the biggest impact on switch performance are "how" the data is forwarded from one port to the other within a switch and "when" the data is forwarded. System designers must make these decisions at the very beginning of the design process. The architectural choices available for the "how to forward" question are: Shared bus, Crossbar, and Shared memory, or a hybrid of some combination of the above. The PES24T6G2 is implemented in a shared bus style architecture. Explanation of these different types of switching architectures is beyond the scope of this document.

The architectural choices available for the "when to forward" question are: Cut-through (start forwarding a packet while it is being received) or Store and Forward (start forwarding only after an entire packet is received). The PES24T6G2 uses the Cut-through forwarding method.

There are several other micro-architectural features or implementation details of a switch that can also have noticeable impact on the performance of a switch. Discussion of the relationship between a feature choice and its impact on performance are beyond the scope of this document. It is relevant to note that several implementation details, such as the transmit retry buffer sizes, ingress buffer sizes, flow control mechanism, allowable maximum payload size (MPS), and controllable frequency of DLLPs including flow control updates and ACK/NACK, have an impact on the performance of the switch. Specifications related to these implementation details for the PES24T6G2 are found in the 89HPES24T6G2 User Manual, available by contacting IDT.

SECTION II: Fibre Channel & 10GE (multi-function) Throughput Measurements

The goal of this set of tests is to demonstrate the behavior of the PES24T6G2 with two FC (Fiber Channel) storage controllers and a 10GE (Ten-gigabits Ethernet) as endpoint devices connected to the PES24T6G2 downstream ports. This represents a multi-function add-on card or Server application scenario.

Dual 8G FC controller HBAs are used on two downstream ports while the third downstream port is populated with a 10GE NIC. Reads and writes to an array of disk drives (or Ramdisks emulating disk arrays) as well as ethernet transfers are done with the IOmeter software tool. IOmeter is also used to gather and analyze the performance data. See Appendix A for additional information on IOmeter. Performance measurements are made in two separate test cases, once with the IDT switch in the data path and once without the IDT switch in the datapath. A comparison of these two sets of results reveals the impact of the switch in the system.

Hardware Setup

Following is a list of system components used for this test:

- Intel Stoakley PDK (IDT switch is plugged into a slot in this machine)
  - Two Intel Xeon 5405 (each CPU was quad core - 2.8 GHz)
  - Intel 5400 (Seaburg) North Bridge
  - 4 GB FBDIMM memory
  - System clock 797 MHz
- Microsoft Windows 2003Server
- IDT PES24T6G2 - x8 upstream, two x4 Gen2 and one x8 Gen1 downstream ports
- Two Emulex LPE12002-M8 HBAs, two 8 Gbps FC ports per HBA
- One Myricom PCIe Gen1, 10GE NIC
- The storage was high speed RAM disks emulating FC disk arrays (a different set of Emulex FC HBAs in target mode acting as storage). Ramdisk emulation software was ThirdIO IRIS Ver 3.0 Rel 2.5 running on a different Gen2 PCIe server linux machine.

The FC controller cards and the 10GE NIC were plugged into the downstream port slots of the PES24T6G2 based board. The upstream port of the PES24T6G2 is at the edge connector of the PES24T6G2 evaluation board and is plugged into a x8 port slot of the motherboard. This way, the PES24T6G2 switch occupies one PCIe slot on the motherboard and creates a fan-out of three slots where two FC HBAs and one 10GE NIC can be used. Figures 9 and 10 illustrate the system setups used for the test.

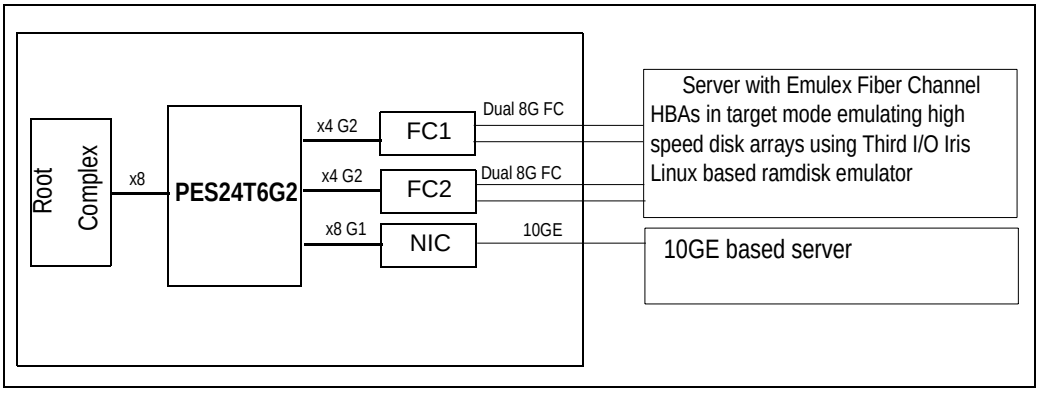

Figure 5 FC & GE Throughput Measurement Setup with the PES24T6G2 and 4 Fiber Ports

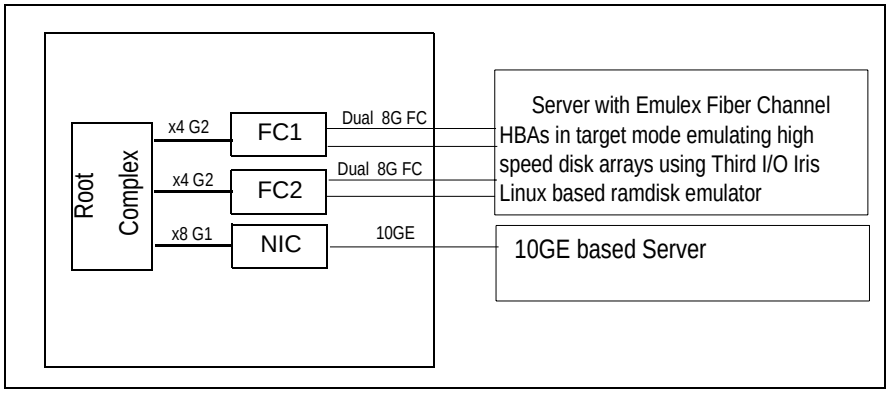

Figure 6 FC & GE Throughput Measurement without the PES24T6G2 and 4 Fiber Ports

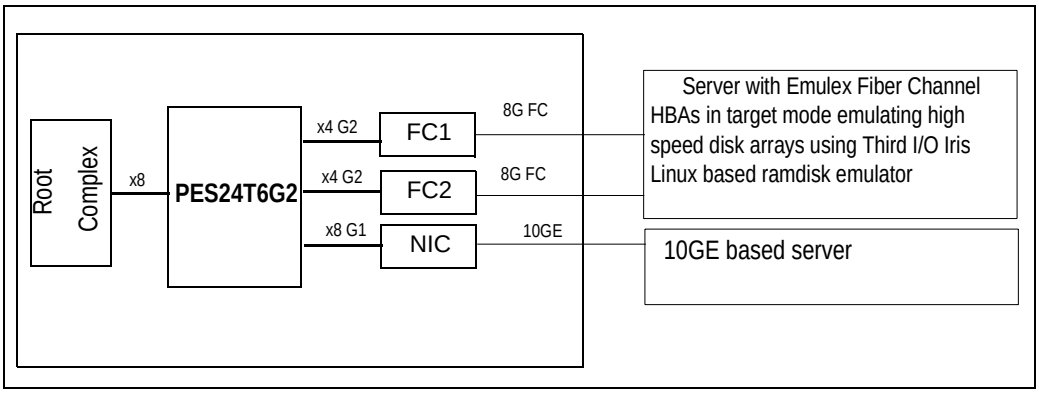

Figure 7 FC & GE Throughput Measurement Setup with the PES24T6G2 and two Fiber Ports

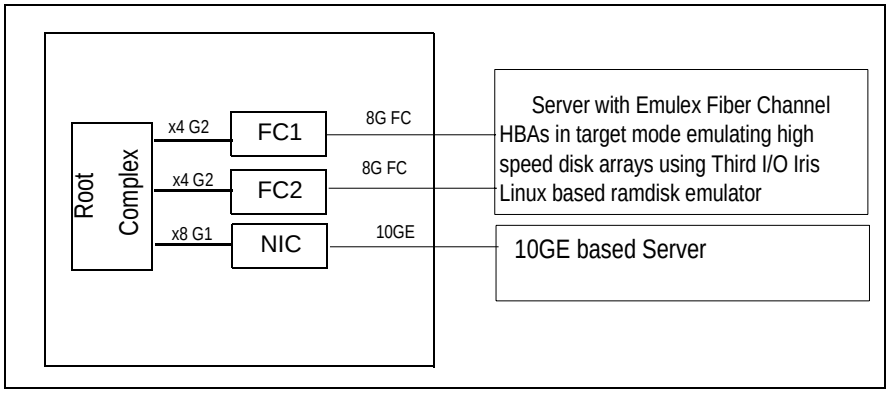

Figure 8 FC and GE Throughput Measurement without the PES24T6G2 and Two Fiber Ports

Software Setup

Various settings of IOMeter were tried for both network and FC IO traffic and once the best setting for the hardware set up was found, measurements were taken with and without the switch in the path. Details related to IOmeter software package can be found in Appendix A. IOmeter version 2006 was used. For disk targets four physical drives, one worker per drive, maximum disk size of 2048 sectors and 16 outstanding IOs were used in the IOmeter settings. For network targets, on both sides of the 10GE link, 3 network workers accessing 1 IP target at the other end of the link were used with transfer request size of 1MB with burst length of 16 and 100% sequential access.

Test Procedure and Methodology

No data loss was permitted along the entire data path in either direction. Combined IOmeter recordings of total throughput including FC and 10GE traffic were noted. 100% writes (i.e. traffic flowing from upstream port to downstream ports of the switch), 100% reads (i.e. traffic flowing from downstream ports to upstream port of the switch) and mixed (half upstream and half downstream traffic) were noted. All tests were run twice, once with all 4 fibers connected, and once with only one fiber per HBA connected.

Results

Type of traffic	No switch	With switch
50% Read / 50% Write	3133 MB/S	3275 MB/S
100% Read	2301 MB/S	2276 MB/S
100% Write	3643 MB/S	3487 MB/S

Table 1 Mixed 10GE and FC Traffic with all 4 Fibers Connected as in Figures 5 and 6

Type of traffic	No switch	With switch
50% Read / 50% Write	3440 MB/S	3543 MB/S
100% Read	2505 MB/S	2564 MB/S
100% Write	2939 MB/S	2838 MB/S

Table 2 Mixed 10GE and FC Traffic with only 1 Fiber per HBA as in Figures 7 and 8

Analysis

Working with different types of PCIe endpoints in the same system creates some interesting scenarios that beg a serious look into the system behavior by the system architect and the software developer. Clearly the switch is placed in an over-subscription situation where the upstream port is a x8 Gen2 interface to the root complex, while the total number of downstream lanes adds up to 16 lanes (8 of them at Gen1 data rate). Secondly, it should be noted that the root system needs to be extremely powerful to sink and source data worth of 8 PCIe Gen2 lanes in order to stress the switch to the limit. Given the CPU, memory and (mainly) the operating system overhead and/or limitations, it is abundantly clear that the system used in this experiment (the best one available in the market at the time of running the experiment) lacks this level of performance. For example, when 50% Read / 50% Write performance is compared between the two results tables, overall performance (with and without the switch) is actually better with fewer fiber connections to the HBA. The CPU utilization with all four fibers connected approached 100%, indicating that the CPU had no time to process the incoming data. This pressure is removed when two of the four fibers are disconnected, as in Figures 7 and 8 and in Table 2.

The system had better performance in some scenarios when the endpoints are directly plugged into the motherboard because the links as well as the system resources are now stressed less since three ports are used with the motherboard (versus one port between the switch and the motherboard). On the other hand, as seen in the results table, there are several scenarios where the overall performance actually improves with the switch in the path. This is as a result of the internal buffering of TLPs within the device.

Various experiments (outside the scope of this document) can be performed with different endpoint vendors, different number of fiber connections between the HBAs and the switch, different MPS settings of the system, different loads between each of the downstream ports of the switch, etc. Depending on the intended usage scenario, customers are encouraged to tailor their own performance measurement tests following the guidelines offered in this performance report.

SECTION III: Fibre Channel & GE (multi-function) Throughput Measurements

The goal of this set of tests is to demonstrate the behavior of the PES24T6G2 with FC (Fiber Channel) storage controllers and GE (Gigabit Ethernet) as endpoint devices connected to the PES24T6G2 downstream ports. This represents a multi-function add-on card or Server application scenario.

8G FC controller HBAs are used on two downstream ports while the third downstream port is populated with a dual-GE NIC. Reads and writes to an array of disk drives (or Ramdisks emulating disk arrays) are done with the IOmeter software tool. IOmeter is also used to gather and analyze the performance data. See Appendix A for additional information on IOmeter. For the GE NIC, Smartbits (see Appendix B) is used for traffic generation and performance measurement.

Performance measurements are made in two separate test cases, once with the IDT switch in the data path and once without the IDT switch in the datapath. A comparison of these two sets of results reveals the impact of the switch in the system.

Hardware Setup

Following is a list of system components used for this test:

- Tyan S5396 motherboard (IDT switch is plugged into a slot in this machine)
  - Two Intel Xeon 5405 (Quad core - 2 GHz)
  - Intel 5400 (Seaburg) North Bridge
  - 4 GB FBDIMM
  - PCIe slots: Three PCIe Gen2 slots (2 x16 and one x8)
- Microsoft Windows 2008 Server
- IDT PES24T6G2 - x8 upstream and three x4 downstream ports, all Gen2
- Two Emulex LPE12000 HBAs, one 8 Gbps FC ports per HBA
- One Broadcom PCIe Gen2, dual-GE NIC
- The storage was high speed RAM disks emulating FC disk arrays (a different set of Emulex FC HBAs in target mode acting as storage).

The FC controller cards and the dual-GE NIC were plugged into the downstream port slots of the PES24T6G2 evaluation board hosting the PES24T6G2 switch. The upstream port of the PES24T6G2 is at the edge connector of the PES24T6G2 evaluation board and is plugged into a x8 port slot of the motherboard. This way, the PES24T6G2 switch occupies one PCIe slot on the motherboard and creates a fan-out of three slots where two FC HBAs and one dual-GE NIC can be used. Figures 9 and 10 illustrate the system setups used for the test.

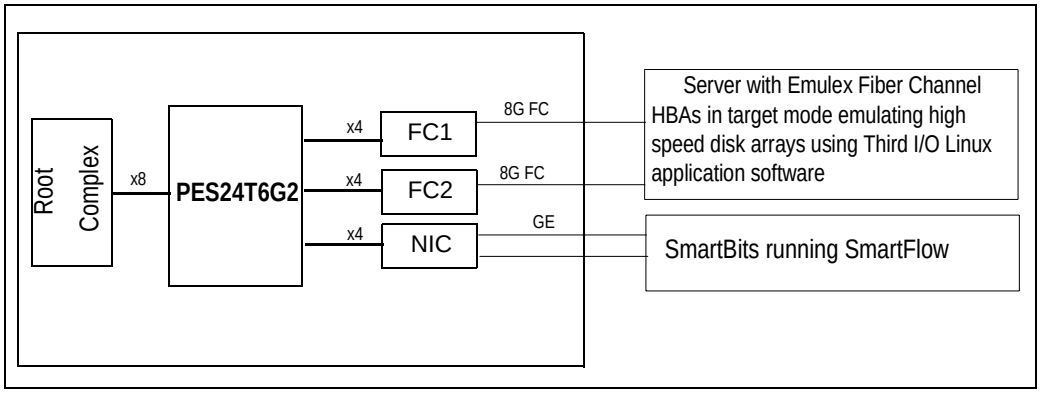

Figure 9 FC & GE Throughput Measurement Setup with the PES24T6G2

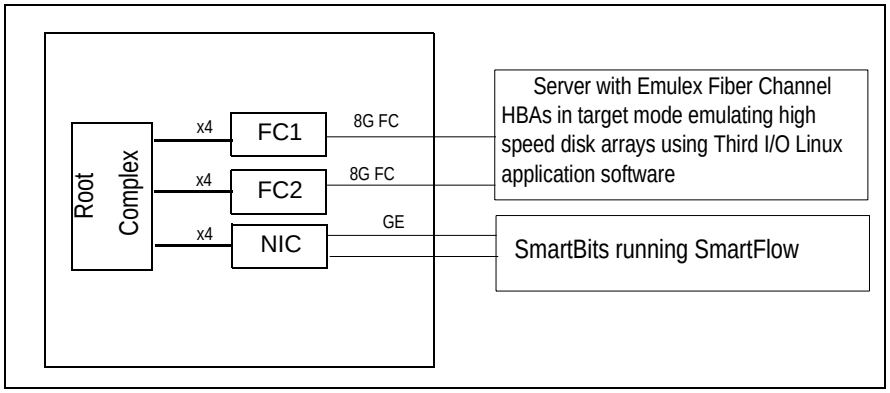

Figure 10 FC & GE Throughput Measurement without the PES24T6G2

Software Setup

For the GE controller cards, the SmartBits 600 Gigabit Ethernet traffic generator is controlled by the SmartFlow software package to generate and sink Ethernet traffic in a loopback mode. Details related to SmartBits setup can be found in Appendix B. The PCI Express-enabled server system is controlled by the operating system and implements bridging of Ethernet traffic from one Ethernet port to another.

For the dual FC controller card, each port was connected to 4 disk drives. Traffic was generated and measurements were taken using the IOmeter software package running on the PCIe host system. Details related to IOmeter software package can be found in Appendix A. IOmeter version 2004.07.30 was used.

Test Procedure and Methodology

For the GE controller card, each port of the SMB600 transmits Ethernet packets of predefined sizes targeted at another port. Each packet transmitted by Port 1 travels through the corresponding NIC in the PCIe system, through the PCIe switch, if present, through the memory in the PCIe system, gets bridged over to Port 2 via the PCIe switch, if present, and returns to Port 2 of the SMB600. Packets starting at Port 2 of the SMB600 traverse the exact opposite path described above. Combined throughput measurements of these two flows for each packet size, with and without the PCIe switch in the path, are recorded. No data loss is permitted along the entire data path in either direction. The tests were run by sweeping through the ethernet packet sizes while running the IOmeter program continuously to monitor any throughput changes worth recording on the storage throughput (FC) side of the test.

For the FC controller card, the IOmeter settings were as follows:

- **100% sequential or random "reads"**
  Sequential 1MB transfers set as 100% Read & 0% Write; 1 Manager (Dynamo) with 6 Workers and 6 Logical Drives (341 MB/drive).
- **100% sequential or random "writes"**
  Sequential 1MB transfers set as 0% Read & 100% Write; 1 Manager (Dynamo) with 6 Workers and 6 Logical Drives (341 MB/drive).
- **50% sequential "writes" and 50% sequential "reads"**
  Sequential 1MB transfers set as 0% Read & 100% Write; 1 Manager (Dynamo) with 6 Workers and 6 Logical Drives (341 MB/drive).

Results

FC Throughput	Ethernet	Throughput in Megabits/Second						
100% Write	GE Packet size (bytes) -->	64	128	256	512	1024	1280	1518
1554 MB/S	<-- Without switch -->	99	166	341	622	1198	1494	1747
1350 to 1100 MB/S	<-- With switch -->	99	177	327	636	1241	1409	1747
100% Read	GE Packet size (bytes) -->	64	128	256	512	1024	1280	1518
1364 MB/S	<-- Without switch -->	99	177	327	622	1212	1494	1775
1360 to 1100 MB/S	<-- With switch -->	99	177	327	636	1241	1409	1747
50% Read, 50% Write	GE Packet size (bytes) -->	64	128	256	512	1024	1280	1518
2220 MB/S	<-- Without switch -->	99	177	327	650	1156	1466	1662
1915 to 1701 MB/S	<-- With switch -->	99	177	327	650	1198	1522	1775

Table 3 Broadcom Ethernet Performance, with and without PCIe Switch, for Mixed FC and GE Traffic

Analysis

Working with different types of PCIe endpoints in the same system creates some interesting scenarios that beg a serious look into the system behavior by the system architect and the software developer. Clearly the switch is placed in an over-subscription situation where the upstream port is a x8 Gen2 interface to the root complex while the total number of downstream lanes adds up to 12 lanes. Secondly, it should be noted that the root system needs to be extremely powerful to sink and source data worth of 8 PCIe Gen2 lanes in order to stress the switch to the limit. Given the CPU, memory and (mainly) the operating system overhead and/or limitations, it is abundantly clear that the system used in this experiment (the best one available in the market at the time of running the experiment) lacks this level of performance. The system has better performance when the endpoints are directly plugged into the motherboard because, obviously, the links as well as the system resources are now stressed less since three ports are used with the motherboard (versus one port between the switch and the motherboard).

As shown by the Broadcom GE NIC test scenario above, the IDT switch actually helps the system throughput in some cases. It should be noted, however, that the switch tends to slightly reduce storage performance as the ethernet packet sizes become larger. Without the switch, storage performance is able to sustain at a constant throughput value at all ethernet packet sizes.

Various experiments (outside the scope of this document) can be performed with different endpoint vendors, different number of fiber connections between the HBAs and the switch, different MPS settings of the system, different loads between each of the downstream ports of the switch, etc. Depending on the intended usage scenario, customers are encouraged to tailor their own performance measurement tests following the guidelines offered in this performance report.

Appendix A Introduction to SmartBits and SmartFlow

Note: Information contained in this section pertains to tools offered by a third party. The information is provided for the convenience of the reader and is not guaranteed to be complete or accurate.

The following document was used for reference while generating this text: Spirent Communications, Inc., 2005. “Introducing SmartFlow.” SmartFlow User Guide (5.0).

SmartFlow is a performance analysis tool to test Layers 2, 3, and 4 on Class of Service devices and networks built with Class of Service priority strategies. SmartFlow allows the setup of multiple flows of IP frames to simulate network traffic and measures latency, frame loss, and throughput. It presents results in charts and tables that include measurements for latency, frame loss, and standard deviation of flows. Results can be tracked by priority or by type of traffic to determine the effect a prioritizing Class of Service device has on the network.

Since our primary goal was to measure throughput through the PCI Express switch, we used the SmartFlow Group Wizard to simply generate flows, track them, and group them. SmartFlow is used in conjunction with a Spirent Communications SmartBits chassis and at least two SmartMetrics or TeraMetrics (or TeraMetrics-based) ports.

SmartFlow includes the following tests:

- Throughput
- Frame Loss
- Latency
- Latency Distribution
- Latency Snap Shot
- Smart Tracker

Below is a general description of the tests that were used for our measurements.

Throughput

Measures the maximum rate at which frames from flows and groups can be sent through a device without frame loss. A sequence of transmissions from one port on the SmartBits chassis to the other port on the chassis is setup. This traffic flows through the device under a test (PCI Express switch) which has Ethernet NICs connected to its downstream ports. An OS-based bridge is created between these two NIC, causing traffic entering one NIC to get forwarded to the other NIC. Bidirectional traffic is used, and each test consists of several sequential transmissions of Ethernet packets varying in size from 64 bytes to 1518 bytes with each type of packets getting transmitted in a single flow for several seconds at a time.

SmartFlow and SmartFlow Demos are available at **support.spirentcom.com**. Path: Self Service Tools –> Download Software Updates –> All Software –> SmartBits –> Applications or Demo. It is necessary to obtain a support account from Spirent to login to this site.

Appendix B: Introduction to IOmeter

Note: Information in this section pertains to tools offered by a third party. The information is provided for the convenience of the reader and is not guaranteed to be complete or accurate.

The following document was used for reference while generating this text: Intel Corporation: Iometer User's Guide December 16, 2003, which is available at:

http://cvs.sourceforge.net/viewcvs.py/*checkout*/iometer/iometer/Docs/Iometer.pdf

The latest version of Iometer, including the documentation, can be obtained from the Iometer project Web Site at the following URL: http://www.iometer.org/

An Iometer is an I/O subsystem measurement and characterization tool for systems. It is both a *workload generator* (that is, it performs I/O operations in order to stress the system) and a *measurement tool* (that is, it examines and records the performance of its I/O operations and their impact on the system). It can be configured to emulate the disk or network I/O load of any program or benchmark, or it can be used to generate entirely synthetic I/O loads. It can generate and measure loads on single or multiple (networked) systems.

An Iometer can be used for the measurement and characterization of:

- Performance of disk and network controllers.
- Bandwidth and latency capabilities of buses.
- Network throughput to attached drives.
- Shared bus performance.
- System-level hard drive performance.
- System-level network performance.

The Iometer tool consists of two programs, *Iometer* and *Dynamo*.

Iometer is the controlling program. Using the Iometer's graphical user interface, you configure the workload, set operating parameters, and start and stop tests. Iometer tells Dynamo what to do, collects the resulting data, and summarizes the results in output files. Only one copy of Iometer should be running at a time; it is typically run on the server machine in which the devices under test are plugged.

Dynamo is the workload generator. It has no user interface. At Iometer's command, Dynamo performs I/O operations and records performance information, then returns the data to Iometer. There can be more than one copy of Dynamo running at a time; typically one copy runs on the server machine and one additional copy runs on each client machine.

Dynamo is multithreaded; each copy can simulate the workload of multiple clients programs. Each running copy of Dynamo is called a *manager*; and each thread within a copy of Dynamo is called a *worker*. A system can simulate stress conditions by deploying several managers and workers. The worst case combination can be determined by experimenting and noting the results.

Once the IOmeter program has been started, a screen similar to that in Figure 11 is displayed. The "Results Display" tab displays performance statistics while a test is running. A user can choose which statistics are displayed, which managers or workers are included in a particular run of the test, and how often the display is updated in real time. A user can change the settings of all controls in the Results Display tab while the test is running. Changes take immediate effect.

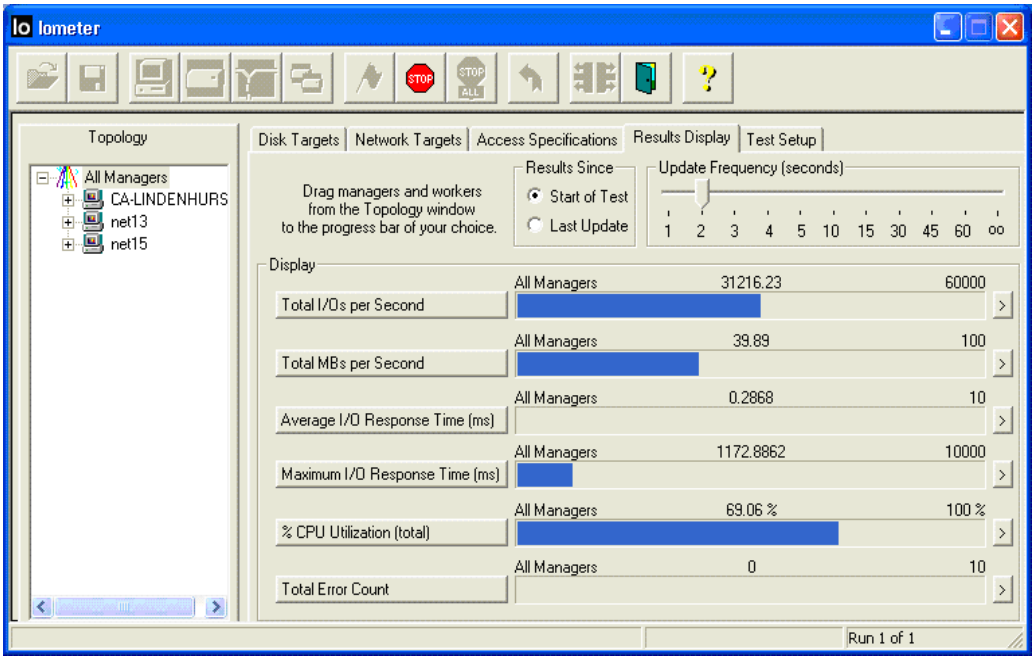

Figure 11 IOmeter Main Screen

As seen in Figure 11, six performance metrics can be displayed as bar charts on the program screen at one time. There are seven main categories of performance metrics. Within each main category, there are several sub-categories of more refined information. Any six out of these large number of metrics can be displayed and tracked as charts in real time. At the end of a test run, results of all sub-categories can be saved as tabulated text files in various formats.

The following is a list of the main metrics and sub-metrics within each main metric.

Operations per Second

- Total I/Os per Second
- Read I/Os per Second
- Write I/Os per Second
- Transactions per Second
- Connections per second

Megabytes per Second

- Total MBs per Second
- Read MBs per Second
- Write MBs per Second

Average Latency

- Average I/O response time (ms)
- Average Read response time (ms)
- Average Write response time (ms)
- Average Transaction time (ms)
- Average Connection time (ms)

Maximum Latency

- Maximum I/O response time (ms)
- Maximum Read response time (ms)
- Maximum Write response time (ms)
- Maximum Transaction time (ms)
- Maximum Connection time (ms)

CPU

- % CPU utilization (Total)
- % User time
- % Privileged time
- % DPC time
- % Interrupt time
- Interrupts per Second
- CPU effectiveness

Network

- Network packets per Second
- Packet Errors
- TCP segments retransmitted per Second

Errors

- Total Error Count
- Read Error Count
- Write Error Count