Fabric Convergence Implications on Systems Architecture

Kevin Leigh, Parthasarathy Ranganathan and Jaspal Subhlok
kevin.leigh@hp.com; partha.ranganathan@hp.com; jaspal@uh.edu

Abstract

Converged fabrics that support data, storage, and cluster networking in a unified fashion are desirable for their cost and manageability advantages. Recent trends towards higher-bandwidths in commodity networks, physical-layer similarities across different communication protocols, and the adoption of blade servers along with the corresponding availability of "backplanes" to implement new networking methods, motivate revisiting this idea. We discuss various aspects of fabric convergence, and present some evaluation results from our experiments in the context of a specific I/O consolidation case study. Based on the insights from these experiments, we discuss opportunities for future research – in new instrumentation and evaluation methods, new cross-layer and application-agnostic designs for fabric convergence solutions, and new system architectures that leverage ensemble-level resource sharing. Our goal, through the discussions in this position paper, is to initiate a more general examination of these issues in the broader academic community.

1. Introduction

Fabric convergence, or the notion of a single network fabric to support data, storage, and cluster networking, has often been discussed as a desirable target for future enterprises, for its potential cost and manageability benefits. Several recent trends in interconnect fabrics motivate revisiting this idea.

Most modern fabrics including 10 Gigabit Ethernet (10GbE), Fiber Channel (FC), InfiniBand, PCI Express (PCIe), and even storage interfaces such as Serial-Attached SCSI (SAS) now have very similar physical layer characteristics. Specifically, they employ CMOS-based differential transceiver technology (SerDes or Serializer/Deserializer [2]), lane aggregation to scale bandwidth, and similar signal voltage swings. At the same time, blade servers are increasingly replacing conventional rack-optimized servers. These systems introduce the notion of *backplanes* that interconnect blade servers and switches, and eliminate expensive optical transceivers and cables. These two trends allow for interesting new system designs using fabric technologies.

Meanwhile, from a performance point of view, commodity fabrics are increasingly providing high bandwidth [5] and are being extended to support other applications (e.g., iSCSI [12] and iWARP [13] protocol layer extensions for storage and cluster networking over Ethernet). The availability of increased compute capacity at lower cost from Moore's Law has allowed for the design of more powerful controllers (e.g., TCP-offload engine network interface controllers or TOE NICs) to efficiently process packets to meet high bandwidth as well as low latency application needs.

Given all these trends, it is now realistic to consider some of these commodity protocols as potential targets for *converged* (or consolidated) fabrics that support data, storage, and cluster networking in a unified fashion. However, several issues and open questions still remain. Specifically, we are not aware of any prior work that has comprehensively evaluated the benefits and the challenges of fabric convergence at system architecture level. The lack of actual systems and models that can effectively characterize benefits has been the key limitation. New instrumentation methods and evaluation models are needed. In addition, there are several design challenges in terms of system architectures. Specifically, fabric convergence requires systems to be evaluated or designed at an *ensemble* level. Although fabric convergence scope for the ensemble can be as large as the entire datacenter (switches, routers, storage targets, management tool chain, virtualization services, etc.), in this paper, we limit our discussions to a blade enclosure. New blade enclosure designs are needed both to enable and to leverage fabric consolidation. Several challenges need to be addressed, in terms of addressing the design across multiple layers of the stack, and the design of an effective fabric manager that can potentially handle multiple fabrics. There are also opportunities to redesign system architectures for

individual blades while holistically addressing resource sharing at the ensemble level.

This paper seeks to initiate a more general examination of these issues in the broader academic community. We discuss various aspects of fabric convergence. As a specific case study, we consider PCIe-based *I/O consolidation* with respect to 10GbE-based *network consolidation*. We discuss our experience and some high-level experimental results characterizing the benefits of fabric convergence at the I/O level. Furthermore, going beyond network and I/O consolidation, opportunities exist for *coherent fabric consolidation* as well. Using the insights from the results of our case study, we then discuss challenges and opportunities for future research.

The rest of the paper is organized as follows. Section 2 provides a broad overview of fabric convergence. Section 3 discusses our experiments on evaluating the benefits from I/O consolidation. Section 4 discusses future challenges and opportunities, and Section 5 concludes the paper.

2. Fabric Convergence: Background

Many large datacenters traditionally have dedicated Local Area Networks (LANs) for client/server data communications and Storage Area Networks (SANs) for servers to connect to remote storage. Some datacenters also have high-bandwidth low-latency server networks for clusters of servers to inter-communicate. These three types of networks historically use different protocols – Ethernet for LAN, FC for SAN, and InfiniBand [11], Myrinet [16] or QSNet [20] for cluster networks.

We categorize fabric convergence at three levels from a server system architecture point of view – network level, I/O level, and coherent level, as shown in Figure-1. Each shaded box represents a physical module and each dash-lined box represents a logical system boundary comprising physical resource partitions, where P, M, IOB and D stand for processor, memory, I/O bridge and I/O device components, respectively. These components may also be replicated within a system, although only one instance of each is shown per system. Similarly, only two systems are shown for simplicity. The network fabric represents an edge network switch, the last switch stage (towards the servers) in a typical datacenter switch topology.

2.1. Network consolidation

Fabric convergence at the network level, as shown in Figure-1(a), is also known as *network consolidation*, where a protocol such as Ethernet or InfiniBand is used for data, storage and cluster networks. In practice, there can be one physical network (as shown), or three physical networks using the same network layer protocol, for data, storage and cluster networking. Note that each system has its own set of I/O devices, and therefore the I/O domains are physically isolated across the systems, i.e., a system

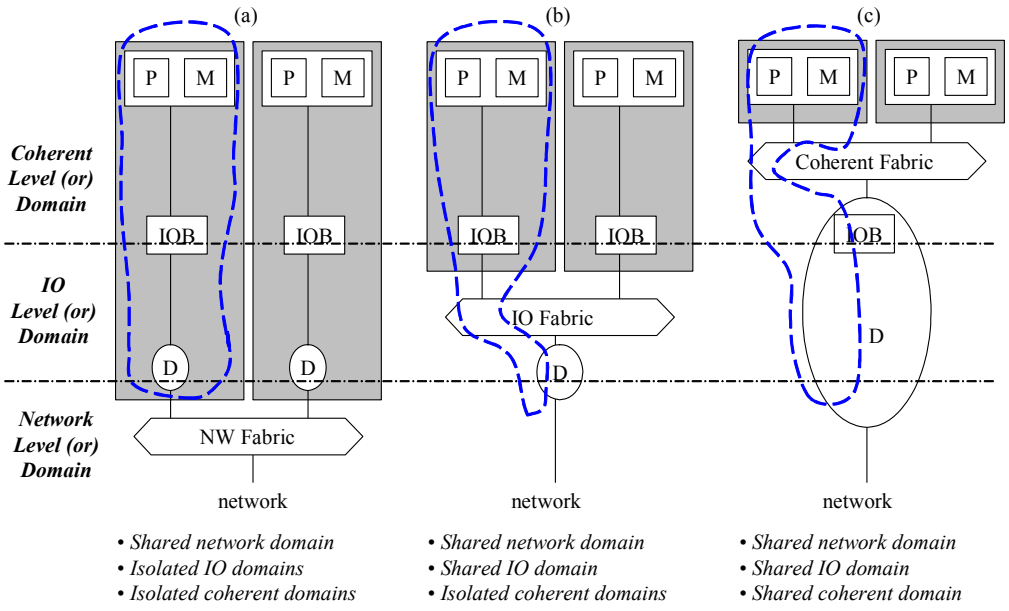

Figure-1. Fabric convergence at different levels: (a) network, (b) I/O, and (c) coherent.

has no visibility or access to another system's I/O devices. Similarly, the processor/memory interconnects, and their coherent domains, are physically isolated across the systems.

The goal for network consolidation is to lower capital expenses and/or operating expenses by reducing the number of network interface devices within servers, and the types of networks in datacenters to be managed from many to one. An important aspect of network consolidation is to provide mechanisms to manage the networks with minimum disruption to datacenter practices, and to provide cost-efficient gateways to connect to incumbent networks to protect the infrastructure investments already made in the datacenters.

Ethernet and InfiniBand have been considered as candidates for network consolidation in the past. InfiniBand [11] has been positioned to be a converged fabric, with high enough bandwidth along with RDMA for cluster networking, IP-over-IB for data networking and SCSI RDMA Protocol (SRP) for storage networking. However, despite being an industry standard, InfiniBand has not had enough support in its ecosystem to be high volume and cost effective. Besides, the existing Ethernet ecosystem and the large volumes cannot be easily displaced by any new networking technology, including InfiniBand. Ethernet has been the ubiquitous data network in datacenters, but 1GbE does not have all the attributes to replace higher performance storage networks (e.g., FC) or a cluster network (e.g., InfiniBand). Until recently, Ethernet did not have rich enough protocols to also be storage or clustering fabric. With relatively new protocol standards for storage networking (iSCSI/iSER [12], FCoE [4]), cluster networking (iWARP [13]), congestion/flow controls [9], TOE/RDMA NICs, and higher data rate of 10GbE [10], Ethernet is now the top candidate to be the consolidated network. For investment protection, iSCSI-to-FC gateways or FCoE bridges can be used to interface to existing FC SANs. It should be noted that FCoE is an emerging technology that is still work-in-progress.

2.2. I/O consolidation

Fabric convergence at the I/O level, shown in Figure-1(b), is also known as *I/O consolidation*, where dedicated I/O devices in Figure-1(a) are replaced with fewer I/O devices, that are shared by multiple systems via one or more I/O fabrics. The dash-lined box represents a logical system boundary combining the resources across the physical boundaries of a module and a component. A goal for I/O consolidation is to reduce costs for ensembles of systems, especially in blade environment where the expensive *edge* network switches are also eliminated. Figure-1(b) qualitatively illustrates a potentially lower cost blade solution of I/O consolidation with respect to network consolidation. We will quantify the cost and performance benefits of I/O consolidation later in this paper.

Furthermore, by physically disaggregating the I/O devices from the processor/memory complex, compute and I/O resources can be scaled independently for a system. This is an important attribute to foster simpler system designs, especially for server blades where real-estate, power and thermal tradeoff challenges are non-trivial. It is now obvious how I/O consolidation can impact system architecture.

2.3. Coherent-I/O consolidation

Fabric convergence at the coherent level, shown in Figure-1(c), is illustrated in this paper as a potential solution to further extend the cost-saving goals, where a common fabric is used for processor/memory coherent transactions as well as for I/O transactions. This makes sense only for applications where there are reasons for multiple processor/memory subsystems to be interconnected. Typically, coherent fabrics have higher bandwidth and lower latency requirements than I/O fabrics. At a high-level, if an environment already has a coherent fabric, then elimination of I/O fabrics could lower the hardware cost. However, for commodity components to interoperate there need to be standards and ecosystems for the system-level solutions to be cost-effective. We will discuss these tradeoffs later in the Challenges section.

2.4. Blade infrastructure for fabric convergence

It is important for datacenters to provide information services for business continuity at minimum capital and operational expenses. New technologies are adopted to meet these objectives where managing risk is an integral part. The areas of risks include supporting application and operating system software, interoperability of new hardware equipment with existing ones in a datacenter, compatibility with existing datacenter management tools, datacenter operational policies and practices, etc. Blade servers have been adopted in datacenters since the early 2000, because their architectures resemble traditional rack-mount servers, minimizing deployment risks while offering higher efficiencies in cost and manageability that directly contribute towards lower capital and operational expenses.

Early generations of blades, shipped during 2000 through 2006, were more or less a repackaging of

rack-optimized servers in an enclosure to provide the key benefits of cable reduction, ease of use, and ease of management. The cable reductions, and associated cost efficiency benefits, stem from the replacement of cables and transceivers with backplane traces for server blades and edge network switches to communicate within the same enclosure. Therefore, a blade enclosure is an infrastructure for servers and network switches to be connected directly, in addition to providing shared infrastructure elements such as power, cooling and management. Most blade enclosures in the market offer infrastructure that supports multitude of network switches. However, network switches for different network link widths typically require different size switch bays. The form factors of almost all different-size switch bays in most blade enclosures are disjoint, and each switch bay is typically limited to a specific network protocol switch usage. Nevertheless, modern blade enclosures [7][8][21] not only provide high efficiencies in power, cooling, management and ease-of-use, but also provide modern high-speed fabrics for I/O and network consolidations.

3. Fabric Convergence: Case Study

In this section, we present selected methods and results from our evaluation study comparing I/O and network consolidation. We first provide the framework for our evaluation space. We then discuss the challenges in performing such an evaluation and our methodology, followed by the results and insights from the study. Details of our study are described in [14][15].

3.1. Evaluation framework

We designed a blade enclosure to be a general-purpose infrastructure (GPI) to enable a smooth migration path from traditional-fabric-based system architectures to converged-fabric-based system architectures. The most notable part of the GPI is the backplane and switch bay designs, where different link-width protocol switches can share the same switch bays, as shown in Figure-2. This approach simplifies the designs of enclosure, backplane and server blade, while providing the same infrastructure for fabric consolidation at different levels as defined earlier in this paper. We tradeoff multiple parameters in designing a GPI, to eventually arrive at the architecture with volumetric positioning of switch bays and adaptive grouping of signal lanes illustrated in Figure-2. Volumetric positioning allows multiple switch bays to be used together for larger switch modules. Adaptive grouping allows multiple backplane trace groups to be used together when larger switch modules are used. Physical implementations of blade enclosures can vary in terms of the volume space and the granularity of the signal lanes per switch bay. For an implementation similar to the switch bays layout of a blade enclosure described in [7], a pair of 1-lane GbE or 10GbE switches, or one 4-lane InfiniBand switch can be used in the side-by-side adjacent bays as shown in Figure-2.

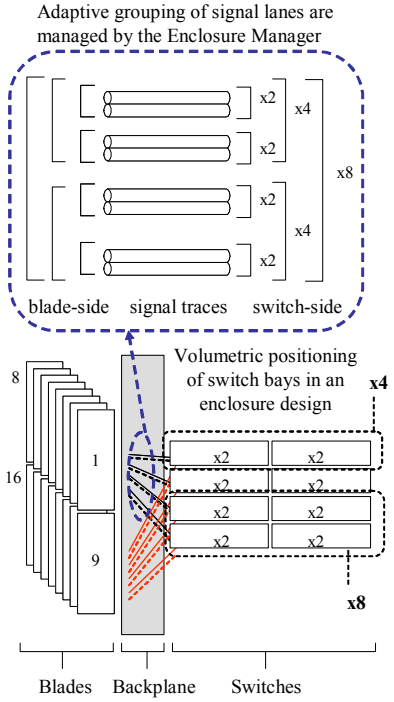

Figure-2. GPI architecture.

Alternatively, a PCIe x2 I/O switch along with shared I/O devices can be used in each of the side-by-side adjacent switch bays. With this level of flexibility, a well-designed blade enclosure can easily support various types of fabric convergence.

We assumed a GPI-based blade enclosure for our evaluations of I/O consolidation vs. network consolidation since the same switch bays in an enclosure can be used for 10GbE or PCIe, simplifying our cost and performance models.

Figure-3(a) shows a blade enclosure with blade servers consisting of (oval-shaped) dedicated I/O devices interfacing to a pair of 10GbE consolidated network switches via a backplane. Figure-3(b) shows the same blade enclosure with blade servers sharing a set of I/O devices via a pair of PCIe I/O fabric. The interconnect links for the devices and the switches are also labeled as either *PCIe* or *10GbE*. The same

backplane traces and switch bays are used (at different times) for PCIe and 10GbE fabrics.

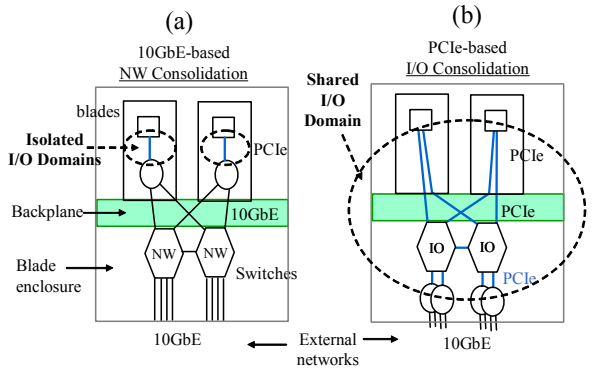

Figure-3. Network vs. I/O consolidation.

3.2. Methodology

Comparing these two fabric consolidations will be straightforward if full systems were available for both architectures. The major challenges in evaluating I/O consolidation are that there are no specific metrics, and there are no full systems available for I/O consolidation.

For our study, we first defined metrics and models for performance and cost. We used 10GbE-based network consolidation as a reference configuration, and we defined several practical configurations within a GPI to evaluate PCIe-based I/O consolidation, as shown in Figure-4. Each dash-lined box in Figure-4 represents a blade enclosure. *10-10* is the reference network consolidation configuration where the internal and external network links are all 10GbE. Only two server blades are shown here for simplicity, but there can be *N* numbers of blades, e.g., *N=16*. A network interface controller (NIC) device in each server blade is shown with an oval block interfacing to a root (CPU/memory complex) with a PCIe x8 (by-8 or 8-lane) interface, and the network switches are shown with hexagon blocks.

The remaining configurations are for I/O consolidation and vary the PCIe link widths across the PCIe switches (in hexagon blocks) for the roots (square blocks) and the shared devices (oval blocks). For example, *R4D8* means the PCIe link widths for the roots and the shared devices are x4 and x8, respectively.

R8D8 was chosen for the roots to have connectivity to a shared I/O device similar to the *10-10* case. *R4D8* was chosen to reduce the root link width and therefore use a smaller I/O switch with the intention to lower the components cost. The number of roots is the highest multiplier for an I/O switch for an ensemble of blade servers. The lower performance of smaller root link width is justified by the fact that the I/O switch is bandwidth over-subscribed for the shared I/O devices. Similarly, *R2D8* and *R2D4* were chosen to further reduce the solution cost by reducing the root link width to x2. Note that there is no x2 ports supported in existing PCIe Gen1 components. With the smallest *max_payload_size* implementation found in almost all the PCIe Gen1 components, a x2 port with 5 Gbps raw bandwidth can support only 3.3 Gbps, after factoring the packet and 8b10b encoding overheads. This useable bandwidth is adequate for very limited number of protocol types (e.g., GbE, FC) even after considering lower bandwidth requirements of each device due to bandwidth over-subscription in network switches.

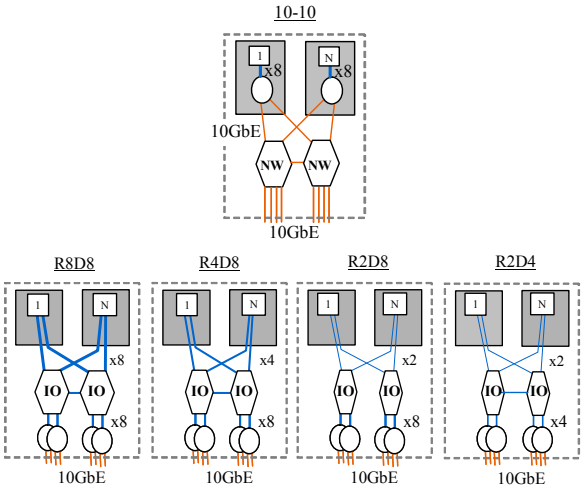

Figure-4. Practical configurations in a general-purpose blade infrastructure.

For a PCIe Gen2 x2 port with a useable bandwidth of 6.6 Gbps, more protocol types can be considered (e.g., 10GbE) especially for the cases where bandwidth over-subscription is inevitable and/or a full line-rate bandwidth cannot be realized for most common applications running on a system. Although, current common wisdom in the industry is to not use smaller than x4 ports for the PCIe Gen2 switches, we are proposing PCIe Gen2 x2 ports to be considered for a good cost/performance tradeoff. *R2D2* was not considered because a x2 link width would be too limited for a shared dual-port 10GbE device. We will later refer to I/O fabrics with root link widths of x8 and x4 as *large* I/O fabrics, and x2 as *small* I/O fabric.

We derived the cost models for the practical configurations illustrated in Figure-4, based on the

10GbE NIC and switch chips' per-port cost trends as well as PCIe switch chips' per-lane cost trends as shown in Figure-5, Figure-6 and Figure-7.

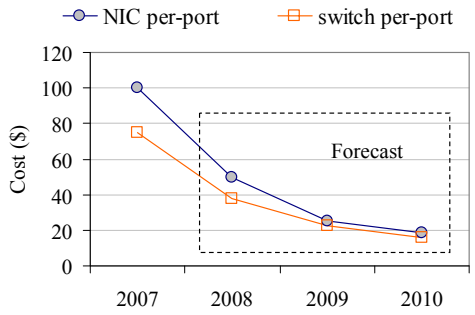

Figure-5. 10GbE NIC and switch components cost trends.

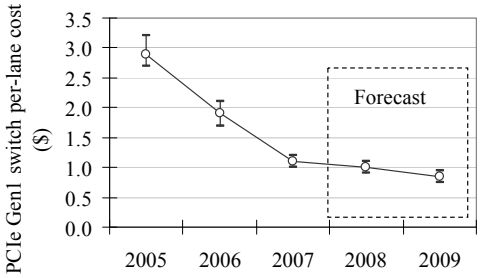

Figure-6. PCIe1 switch component cost trends.

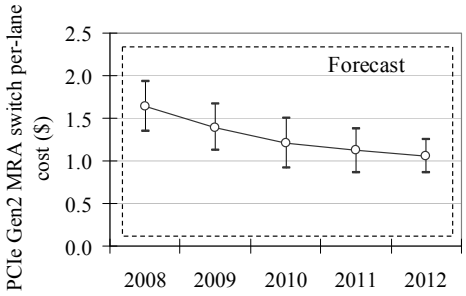

Figure-7. PCIe2 switch component cost trends.

It should be noted that the cost trends illustrated in the above figures are forecasted basic component costs. The costs for the I/O card and switch module solutions will be higher after considering packaging and supporting electronics (management controllers, voltage regulators, etc.) as well as the costs for the management software and development costs.

We also introduced new cost parameters, e.g., multi-root aware (MRA) premium factor, to calculate the costs as shown in Table-1. We used a wide range of base costs for the I/O device and network switch per-port for sensitivity analysis.

A PCIe switch per-port cost can be derived by multiplying its per-lane cost with the number of lanes per port, which may be 2, 4 or 8 for popular link widths. Assuming a $1.50 per lane, a x4 (or 4-lane) port's cost will be $6. Even if 10GbE per-port cost drops to about $20 in the future, it will still be multiple times higher than a PCIe x4 per-port cost. This cost differential is one of the main reasons for 10GbE-based network consolidation hardware components cost to be higher than PCIe-based I/O consolidation cost.

Table-1. Cost model parameters summary.

Description	Range (Eval Cost Point)
IO switch per-lane base cost	$1.00-$1.50
IO switch interface cost	$50
IO device base cost	$10-$200 ($80)
network switch per-port cost	$5-$100 ($40)
MRA premium factor for an IO switch per-lane cost	0.4
MRA premium factor for a shared device	0.5
Shared IO device's local RAM premium factor	0.1

Note that the above example, on x4-lane based PCIe switch component cost comparing with 10GbE switch, is the *R4D8* configuration scenario. Also note that a PCIe Gen2 x4 port can support about 13 Gbps to 16 Gbps, depending on the *max_payload_size*, which is adequate to support one 10GbE port at full line rate. Here, we are not assuming peer-to-peer communication across the I/O switch in addition to the blade servers communicating to external of the enclosure via a shared NIC. If an application calls for peer-to-peer through an I/O switch as well as through the shared NICs, then *R8D8* should be considered.

For the performance metrics, we defined a range of bandwidths to bracket the best-case and worst-case workload scenarios. We defined the best-case workloads to be *no-share* when a root has access to all the shared I/O devices connected to the I/O switches, and the worst-case workloads to be *full-share* when all the roots share all the shared I/O devices. We further sub-divide the workload conditions into *max* (maximum), *typ* (typical) and *low*, where we did not consider bandwidth degradations for the *max* cases. We considered severe bandwidth degradations for the *low* cases, and mid-point degradations for the *typ* cases. Therefore, we will have a total of six cases to span the performance range – *no-share-max, no-share-typ, no-share-low, full-share-max, full-share-typ* and *full-share-low*, where *no-share-max* is the best-case and the *full-share-low* is the worst-case bracketing the performance range (while the rest may overlap). To derive the performance of each of these six cases, we defined four key parameters to factor the degradation of the I/O bandwidth that we derived from the current system measurements, as follows:

Throughput-overhead [θ] is the amount of additional I/O bandwidth required to sustain the network bandwidth of a NIC. Throughput-overhead depends on network packet size, NIC settings to handle system interrupts, how the descriptors are written to a NIC within a system environment, and how a NIC manages its resources (e.g., how a TCP-offload NIC manages its connection context cache).

Switch latency penalty [δ] is the amount of bandwidth degradation because of latencies through an I/O switch. This can be due to physically longer paths such as multiple switch chip hops, switch internal buffer scheduling causing congestion, head-of-the-line blocking, etc.

Link width ratio (LWR) penalty [ω] is the amount of bandwidth degradation because of the root and device link width difference across an I/O switch. It is easy to derive the bandwidth ratio of links having different widths when the links are fully utilized. For example, half the link width will reduce the smaller link's full bandwidth to half that of the wider link's full bandwidth. However, it is not apparent how the bandwidth will get degraded when the link utilizations are low.

Sharing penalty [λ] is the amount of bandwidth degradation on each root link, when multiple roots are sharing a device link. Intuitively, a device link bandwidth will be equally divided among the sharing roots if the roots have uniform workloads. However, PCIe protocol parameters such as *max_payload_size*, link width and link *flow control credits*, as well as implementation choices such as I/O port buffer depths (and count) and I/O switch internal data path forwarding methods can cause uneven bandwidth allocations across the switch ports, due to resource starvation, contention or scheduling.

Next, we defined a set of performance models, based on bandwidth bottlenecks within the GPI-based configurations for network and I/O consolidations. Below, we show only two expressions to illustrate how the bandwidth degradation parameters are used to derive the effective I/O bandwidths. A complete list of equations is described in [15].

Eq(1) and Eq(2) show the I/O consolidation performance equations for a *no-share* and a *full-share* cases, respectively, where θ, δ, ω and λ were described above, ρ is the PCIe payload efficiency factor which depends on the payload size, and ε is the 8b10b PCIe line encoding factor which is 0.8 (80%) [17]. These parameters factor either the total bandwidth of a root or all the roots for *no-share* or *full-share*, respectively. B_u is the bandwidth of a root link (i.e., an upstream link of an I/O switch), $\sum B_d$ is the aggregate bandwidth of all the downstream links of an I/O switch, and N_{is} is the number of I/O switches.

$$B_{IOC\_no\_share} = \Sigma B_d \cdot \frac{\rho.\varepsilon}{1+\theta} \quad \text{Eq(1)}$$

$$B_{IOC\_full\_share} = \mathrm{B_u}.N_{is}.(1-\delta).(1-\omega).(1-\lambda).\frac{\rho.\varepsilon}{1+\theta} \quad \text{Eq(2)}$$

To determine the values for each of these parameters for the entire performance range discussed earlier, we derived a hybrid methodology based on physical system measurements, hardware FPGA-based emulation, and simulation models. The procedures for our approach involve the steps using, (1) current systems to characterize PCIe and 10GbE network performances, (2) FPGA-based hardware emulators and PCIe switches to evaluate the I/O throughput and latency effects for the selected I/O consolidation configurations by varying I/O parameters on the hardware, (3) simulation models to extend our evaluation for the cases that we could not cover with hardware emulators, and (4) limited-function hardware system prototypes to verify the sensitivity of I/O parameters at application level.

3.3. Results and discussion

We ran several sets of experiments to quantify the bandwidth degradation parameters, but only show a few sample results in this paper due to space constraints.

Figure-8 shows one sample set of link width ratio (LWR) penalty results of *R4D8* configuration compared to *R8D8* configuration, where the difference is on the roots' I/O link width (x4 vs. x8). Figure-8(a) shows I/O bandwidths for network transmits for both x8 and x4 cases measured on a limited-function system prototype for varying application message sizes. Note that the I/O bandwidth of the x4 case is expected to be about half of the x8 case at saturation (for message size 4KB and above).

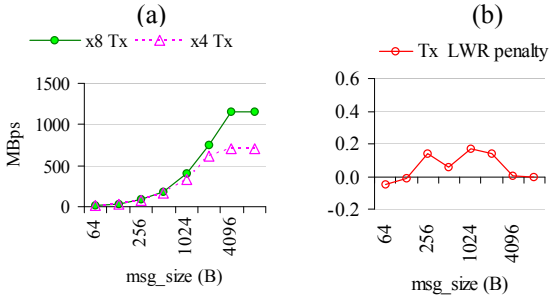

Figure-8. A set of example results for link width ratio penalty for R4D8.

Figure-8(b) shows the *Link Width Ratio Penalty* of less than 20%, derived from Figure-8(a), representing the bandwidth degradation of a narrower link width (x4) when the I/O link is not fully saturated (for message sizes smaller than 4KB).

Table-2 shows a sample set of the bandwidth degradation parameter values for the six workload conditions we defined.

Table-2. Bandwidth degradation parameters results chart sample.

	Throughput overhead	Switch latency penalty	Link width ratio penalty*	Sharing penalty
To calculate:	θ	δ	ω	λ
No-share-max	0	0	0	n/a
No-share-typ	0.1	0.1	0.10	n/a
No-share-low	0.5	0.15	0.31	n/a
Full-share-max	0	0	0	0
Full-share-typ	0.1	0.1	0.10	0.1
Full-share-low	0.5	0.15	0.31	0.2

These bandwidth degradation parameter values were generated for various application message sizes, and are applied to the performance model equations for the I/O consolidation configurations.

Figure-9 shows the performance brackets of the I/O consolidation configurations with respect to the *10-10* network consolidation.

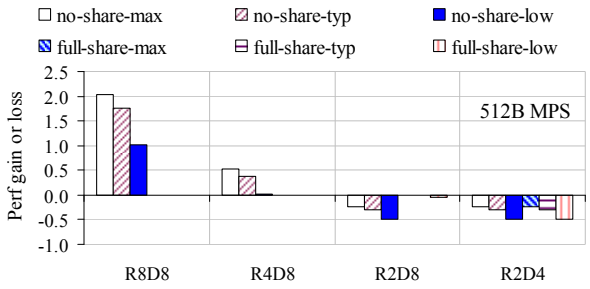

Figure-9. I/O consolidation configurations' performances compared to 10-10 for 512-byte message size.

R8D8 and *R4D8* have bandwidth advantages over *10-10* for the *no-share* cases, since there are more 10GbE links available across the I/O switches, compared to *10-10* where each root has the network links of only one dedicated NIC. *R2D8*'s *no-share* cases have lower performance than *10-10* because we assumed that the dedicated NIC in *10-10* has a x8 host interface. *R8D8, R4D8* and *R2D8* have negligible bandwidth degradations for *full-share* cases compared to *10-10*, since the effects of bandwidth division of the I/O links for the sharing roots in I/O consolidation configurations were comparable to the effects of bandwidth division of the over-subscribed network links in *10-10*. *R2D4*'s *no-share* cases have lower performance due to its x2 link compared to the x8 link in *10-10*, and also for the full-share cases, the x4 link width of the shared devices became the bottleneck.

Figure-10 shows the best-case performance/cost comparison of the I/O consolidation configurations with respect to the *10-10* network consolidation. Recall that the cost comparison is based on only hardware component costs. *R8D8* has 200% higher performance (i.e., about 30 Gbps bandwidth) at about 30% cost saving. For those applications that do not need this bandwidth, *R4D8* still offers 50% performance gain (i.e., about 15 Gbps) and just over 50% cost savings. *R2D8* offers slightly more cost savings at 60%, but the best-case bandwidth is about 25% lower than *10-10*, i.e., about 7.5 Gbps, which might be acceptable for many applications, especially considering network over-subscription farther up in the datacenter switch hierarchy.

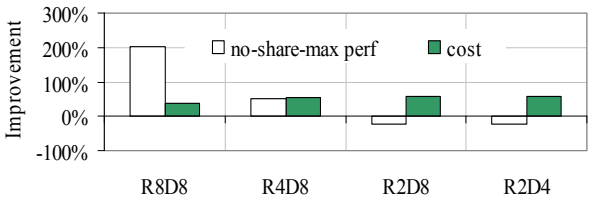

Figure-10. I/O consolidation configuration's cost/ performance compared to 10-10 for 512-byte message size.

Another important point to consider is that, for smaller root link widths (x2) the I/O switches fit in smaller switch bays in a GPI-based blade enclosure [7]. This allows more fabrics to be used, and consequently enabling higher level of fabric isolation and/or finer granular scaling of fabric connectivity.

The PCIe specification [17] defines the *max_payload_size* to be from 128 bytes through 4096 bytes with power-of-2 increments, i.e., 128, 256, 512, etc. A common wisdom in the industry for PCIe components is to support the largest *max_payload_size* for best link efficiency. Note that, although existing PCIe components in the industry mostly implemented up to 256-byte *max_payload_size*, newer generation of PCIe components are expected to implement higher *max_payload_size* (e.g., 2048-byte). We learned from our experiments that a) fair bandwidth allocation across I/O switch ports is difficult to achieve for large payload sizes; b) flow control credits are less flexible to be managed for payload size beyond 512 bytes for most PCIe switches (with about 2048-byte buffer per port); and c) link width ratio penalties decreased only

slightly for payload size beyond 512 bytes. Besides, PCIe link efficiency is about 95% and 76% before and after considering 8b10b encoding, respectively, where only a few more percent efficiency can be gained for 4096-byte payloads. One other important point is that supporting a large *max_payload_size* significantly impacts silicon cost of high port-count PCIe switch components if large buffer queues are desired. Based on all these reasons, we believe 512-byte *max_payload_size* enables a good cost/performance balance for high port-count PCIe switches.

3.4. Summary

Our case studies show that I/O consolidation *can* be more cost-efficient than network consolidation, *but* requires careful trade-offs in system and component configurations.

Larger I/O fabrics offer best performance gains at attractive cost savings if they are fully utilized, and smaller fabrics offer best cost savings at acceptable performance gains. Smaller fabrics also offer better fabric isolation and finer grain scalability for both performance and cost. Note that PCIe does not have native congestion management, and quality of service (QoS) is managed by means of *Traffic Classes* and *Virtual Channels*. Physically isolated fabrics enable separation of traffic with varying QoS requirements.

Network consolidation does not affect physical server architecture, but I/O consolidation does. I/O consolidation allows designers to define the hardware boundaries more flexibly to accommodate system resources. I/O consolidation eliminates dedicated I/O devices in each server, and allows multiple servers to share fewer I/O devices. Since I/O consolidation enables physical decoupling of dedicated I/O devices while enabling shared I/O devices for storage and network resource connectivity, processor and memory resources can be scaled independently from the storage and network resources for a system.

4. Future Challenges and Opportunities

We have described fabric convergence at different levels, namely network, I/O and coherent. We illustrated an evaluation of I/O consolidation (with several simplifying assumptions) as a case study to assess the benefits and tradeoffs of fabric convergence at the I/O level. However, several challenges remain to be addressed, presenting opportunities for future work. We list some of them as follows:

4.1. Evaluation methods

Application-specific message size profiling – We have seen in our experiments that application message size has significant impacts in all of our performance studies. Also, large PCIe packet sizes (limited by the *max_payload_size* parameter) can benefit large application message sizes, but at higher component costs. We collected information on a few application message size distributions, including prior work on internet traffic [6] and high-performance technical computing [15]. It is important to better understand application message size profiles of various applications, and categorize them based on similar message size characteristics such as sequence pattern, distribution profiles, etc. These profiles can be used for system performance sensitivity analysis with respect to application message size, *max_payload_size* and network packet size. Knowledge gained from these analyses can guide chip architects to size the components and system architects to size the system designs for efficient cost/performance tradeoffs.

Network performance analysis tools – We used network load generation tools such as netperf, ntttcp, etc., where every test-run requires a command line entry with various arguments, including message size. It will be invaluable to apply message size profiles or trace sequences, to mimic certain application classes.

Workload correlation – In studying I/O or network transactions of an ensemble of systems, it will be useful to easily source workloads that can be forced to be maximally interleaved, maximally contentious, or to have some defined degree of contentions.

Total costs of ownership and industry ecosystem dynamics – A key challenge is how to capture the total cost of ownership associated with each solution. In our example case study, we focused primarily on hardware component costs. But for the correct conclusions to be drawn, one would need to look at total costs of ownership including costs for management software. Another intangible that is often hard to characterize in a formal model is the dynamics around industry ecosystem and volume, and better methods are needed to capture these in formal models.

Robust instruments – We studied I/O consolidation in the absence of a full system environment using hardware emulations and partial system simulations. This method was useful to study the sensitivity of the desired I/O parameters in a controlled environment, i.e., we were generally able to isolate the study of individual parameters.. However, the FPGA-based hardware emulators we used were bulky and not agile enough to carry out multiple experiments in a timely manner. There are opportunities for PCIe instrument vendors to consider hardware and software tools to assist the analysis of single or multiple I/O domains. Suggested instrument features are: user-defined traffic generation, emulated device endpoints with

configurable parameters (e.g., *max_payload_size*, flow control credits), post-processing multiple data sets with time synchronization, robust trace statistics, and import/export data formats to exchange trace data sets across different vendors' instruments.

4.2. Resource management

Multi-level resource allocations – In a traditional server, the dedicated I/O devices are physically part of the server. In an I/O consolidation environment, I/O device resources are physically disaggregated from the processor/memory complexes of a system. There is a need for management mechanisms to configure these I/O device resources, partition them and allocate portions of the resources to each system. In addition, a partition within I/O fabrics and I/O devices allocated to a system needs to be well isolated from other systems. Methods to configure resources to be shared by multiple systems are not new, and they can be extended to shared I/O devices. However, more work is needed on how to map the system-level and I/O level resource allocation. For example, what configuration parameters (e.g., payload size, flow control credits) should be setup in a device, the I/O fabric it attaches to, and the system it is allocated to, to provide a certain I/O bandwidth for a system while minimally impacting the resources already allocated to other systems?

Hardware/software layer interoperability – In today's server system deployment in large datacenters, IT organizations spend significant amount of time and resources to enable combinations of hardware and software layers to interoperate, before deploying them in their production environment. This includes testing the combinations of layers of hardware (processor, chipsets, devices, system BIOS, device firmware) and software (application, OS, network protocol stack, device driver, device option ROM code). Fabric consolidation allows disaggregating of physical resources and means for multiple systems to logically share these resources, which can be independently updated or replaced. There are significant challenges to validate the hardware and software interoperability within a logical system when its resources are disaggregated. These challenges are compounded as multiple hardware and software versions are updated for disaggregated devices, system firmware, operating systems and application software. Should the entire hardware and software architectural stacks be revisited for better hardware/software layer interoperability in a shared resource environment such as I/O consolidation?

4.3. System architecture implications

Disaggregated system resources – A server system architecture typically consists of four main resources – processor, (volatile) memory, (non-volatile) storage and network. Storage and network resource connectivity is achieved by the use of I/O devices. I/O consolidation by definition disaggregates the dedicated I/O devices from each server's processor and memory complex, enabling independent scaling of system resources. This enables independent sizing of form factors, capacity, bandwidth, etc. so much so that system OEMs can create building block modules to serve a wide customer base instead of developing different size servers as they traditionally do. Another major advantage is for users to flexibly configure a server system by logically binding disaggregated resources. However, what are the metrics and mechanisms to enable and ensure reliability, QoS, data security, ease of servicing, etc. for the disaggregated resources?

Hierarchical fabrics – We evaluated I/O consolidation in a flat I/O fabric topology where there was only one layer of I/O fabric that the roots and the shared I/O devices connect to. As silicon compaction trend continues, it will not be a surprise to see more System-on-Chip (SoC) products. Instantiating multiple SoCs in a blade server form factor will require a fabric within a blade to aggregate the connectivity, be it a network or I/O. This introduces a nested hierarchy of fabrics. What are the metrics and models to effectively evaluate hierarchical fabric consolidations?

Level-merging of converged fabrics – Mainstream processors are expected to have integrated memory controllers and multiple coherent links for processor chips to interconnect among them, similar to HyperTransport [1] connecting AMD Opteron processors in a NUMA topology. In this topology, I/O devices can be interfaced to each processor, or only to a few processors. Regardless, a processor can *reach* an I/O device connected to another processor by *tunneling* I/O transactions within the coherent links that connect the processors. In other words, for I/O transactions to traverse within a relatively small coherent domain comprising of a few processor elements is not new. What are the challenges for scaling the coherent domain? In addition, traversing coherent transactions in an I/O domain, or a network domain, would require more investigation to identify and address the challenges, e.g., How to manage coherency snoops as the size of an I/O domain and/or coherent nodes are scaled larger?

Address Translations – I/O transactions have memory semantics, i.e., memory-address based transactions for data movements. Multiple hosts sharing I/O devices via I/O fabrics involve (memory) address translations since hosts can potentially use the same memory address ranges for a logical partition of a physical I/O device. For shared I/O devices, an IOV specification for Address Translations [18] was released relatively recently. However, this is just a *mechanism* specification. There is a need to develop *policies* developed at ensemble of system level, to study tradeoffs in address translation methods, e.g., where to place them, what table sizes, security aspects of the transactions and tables, etc. Interesting challenges exist in integrating these policies with the virtualization layer and other layers of services that control resource management.

5. Conclusions

Recent trends towards high bandwidth in commodity networks and physical layer similarities of commodity PCI Express I/O fabric and network fabrics such as 10GbE and InfiniBand, has made fabric convergence for blade infrastructures a reality. In this paper, we discussed various aspects of designing systems with fabric convergence. We presented a specific case study comparing I/O consolidation and network consolidation with technologies that are available today. As part of this case study, we also illustrated how new metrics and models are needed for cost/performance evaluation. For example, our methodology introduced a combination of actual system measurements, FPGA-based hardware emulations to source and sink I/O transactions, and simulation to draw conclusions. The result was a range of cost and performance results to compare PCIe-based I/O consolidation benefits with respect to 10GbE-based network consolidation. For costs, we focused on hardware component costs of the practical configurations defined in the context of a general-purpose bladed infrastructure. We will need to extend this to include formal ways to capture total solution cost including full product design, management tool layers, license fee, etc. Regarding performance, I/O consolidation inherently allows a processor/memory complex (root) to have full access to all the I/O devices connected to the I/O switch that it connects to. This allows a system's best-case performance to be multiple times higher for I/O consolidation compared to network consolidation at a competitive cost.

Our work shows the conditions that enable promising results for I/O consolidation, and lays the groundwork for evaluation of other types of fabric convergence. However, a lot more work is needed in this area. Specifically, we discuss some key research opportunities – in evaluation of fabric converged systems, resource management across ensembles beyond blades, and opportunities for new system architectures. We hope that the discussions in this position paper provide a starting point for the broader academic community to initiate a more general examination of these issues, and look ahead to the next generation of system architectures that build on these recent exciting developments in the area of fabric convergence.

6. Closing Remarks

We would like to thank the anonymous reviewers, and Robert Elliott, Michael Krause, and Jayaram Mudigonda (all at HP), for their valuable feedback on the paper.

All opinions in this position paper represent the views of the authors and do not represent official HP positions on these subjects.

7. References

[1] D. Anderson, "HyperTransport Architecture," Addison-Wesley Longman Pub, Co., Inc., February 2003.

[2] T. Beukema, M. Sorna, K. Selander, S. Zier, B.L. Ji, P. Murfet, J. Mason, W. Rhee, H. Ainspan, B. Parker, M. Beakes, "A 6.4-Gb/s CMOS SerDes Core with Feed-Forward and Decision-Feedback Equalization," IEEE Journal of Solid-State Circuits, Vol. 40, Issue 12, December 2005.

[3] Cisco, "Scalable Fabric Design – Oversubscription and Density Best Practices," Cisco Systems Inc., 2004.

[4] C. Desanti et al., "Fibre Channel over Ethernet," T11/07-303v0, May 2007.

[5] Ethernet Alliance, "Ethernet Alliance Supports Progress Towards Higher Speed Ethernet and Energy-Efficient Ethernet Standards," press release, July 2007.

[6] J.L. Hennessy, D.A. Patterson, "Interconnection Networks and Clusters" in "Computer Architecture – A Quantitative Approach," 3rd Edition, pp. 791-801, Morgan Kaufmann Publishers, Elsevier Science, 2003.

[7] "HP BladeSystem c-Class Architecture: Technology Brief," HP, 2006. www.hp.com/go/blades

[8] IBM BladeCenter, www.ibm.com/bladecenter

[9] IEEE 802.1au, "IEEE Standard for Local and Metropolitan Area Networks – Virtual Bridged Local Area Networks – Amendment 10: Congestion Notification, Draft 0.1," IEEE Computer Society, September 2006.

[10] IEEE 802.3ap-2007, "Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification. Amendment 4: Ethernet Operation over Electrical Backplanes," IEEE Computer Society, March 2007.

[11] "InfiniBand Architecture Specifications, Rev. 1.0," InfiniBand Trade Association, October 2000. www.infinibandta.org

[12] "iSCSI Extensions for RDMA (iSER) Specification," IETF/RFC 5046, October 2007. tools.ietf.org/html/rfc5046

[13] "iWARP: An RDMA Protocol Specification, Version 1.0," InfiniBand Trade Association, October 2002.

[14] K. Leigh, P. Ranganathan, J. Subhlok, "General-Purpose Blade Infrastructure for Configurable System Architectures," Journal on Parallel and Distributed Databases (JPDD), Springer Publishers 0926-8782 (Print) 1573-7578, March 2007.

[15] K. Leigh, "Design and Analysis of IO Consolidation in a General-Purpose Infrastructure for Blade Servers," Ph.D. Thesis, University of Houston, August 2007.

[16] Myrinet, "Myrinet 2000 Serial Link, Rev. 1.0," 2002. www.myri.com/open-specs/serial.pdf

[17] PCI SIG, "PCI Express Base Specification Revision 2.0," December 2006.

[18] PCI SIG, "Address Translation Services, Revision 1.0," March 2007.

[19] PCI SIG, "Multi-Root I/O Virtualization and Sharing, Revision 0.7," June 2007.

[20] Quadrics, "QSNet," 2007. www.quadrics.com

[21] Sun Blade Systems, www.sun.com/blades