Viable opto-electronic HPC interconnect fabrics

Ronald Luijten Cyriel Minkenberg

IBM Research GmbH
Zurich Research Laboratory
Säumerstrasse 4
CH8803 Rüschlikon
Switzerland
+41-44-724 8348 +41-44-724 8670

{lui, sil} @zurich.ibm.com

Roe Hemenway Michael Sauer Richard Grzybowski

Corning Incorporated
Science and Technology Division
Science Center Drive, Sullivan Park
Corning, NY, 14831
USA
+1-607 974 9731 +1-607 974 1540 +1-607 974 0681

{hemenwaybr, sauerm, grzybowsrr} @corning.com

ABSTRACT

We address the problem of how to exploit optics for ultrascale High Performance Computing interconnect fabrics. We show that for high port counts these fabrics require multistage topologies regardless of whether electronic or optical switch components are used. Also, per stage electronic buffers remain indispensable for maintaining throughput, lossless-ness and packet sequence. Although the notion of true all-optical packet switching is not yet viable, we show that appropriate use of optical switching technology offers power and scaling advantages that can be leveraged economically, and propose a hybrid opto-electronic HPC interconnect fabric architecture that combines the strength of electronics in processing and storing information with the strength of optics in switching and transporting high bandwidths. Using Semiconductor Optical Amplifier technology, we are building a prototype demonstrator switch that we believe solves all the technical challenges. Having reached this threshold now enables commercialization of this technology, which we are currently pursuing.

Keywords

HPC, Interconnect, Switching, Optical Switching.

I. INTRODUCTION

It has been desirable to use optical switches for supercomputer interconnect fabrics for many years.[1] Deployment has been inhibited by high cost, inappropriate optical technologies, unsolved technical challenges and sub-optimal architectural choices. We address these challenges by analyzing key boundary conditions and by proposing a hybrid opto-electronic switch architecture.

With increasing data rates, copper cables between racks will likely be replaced by optical fibers. At 10 Gb/s per wire, copper runs into limits given by physics, mainly the skin effect. As a result, the wire diameter needs to be increased or more complex signal processing techniques need to be used to compensate for the frequency-dependent losses [1]. The first option is typically rejected because the cables become unmanageably thick. The second option requires too much power and chip area when many links are put in parallel to obtain higher bandwidths. Optical fibers solve this, at the cost of additional components to convert electronic to optical signals and back (EO and OE). Once these signals are in the optical domain, a logical desire is to keep them optical until they arrive at the destination host node, by means of an all-optical interconnect fabric.

Packet switching is needed for supercomputer interconnect fabrics to achieve high fabric efficiency while maintaining low latencies, ideally on the order of time-of-flight across the cables. Packet switching requires the following basic functions: routing, buffering, flow control, scheduling, and transmission. To only perform the routing function in the 'all-optical' domain, we will need to extract routing information from the packet header optically as well as configure the optical switch element with an optical control signal. We are not aware of a cost-effective technology to perform this function today. Hence, the notion of all-optical packet switching is currently beyond economical reach. Also, we can conclude that some electronics will be needed in the fabrics.

Notwithstanding, we are convinced that optical interconnects for supercomputers still have important merits. Electronic CMOS switches have electrical pins that are limited to a few tens of Gb/s per pin. Optical switches use fibers or other waveguides with a bandwidth that currently exceeds 1 Tb/s per waveguide. A key advantage of optical technology is that much larger transmission distances can be covered at these high bandwidths. With CMOS technology, implementation of a switch is restricted to a few chips that are tightly packaged together to maintain electrical signal integrity. With optics, we can package in a larger confinement at significantly higher base bandwidths. As a result, we see the value

[1] This research is supported in part by the University of California under subcontract number B527064.

SC|05 November 12-18, 2005, Seattle, Washington, USA
© 2005 ACM 1-59593-061-2/05/0011… $5.00

of optical switching in providing more ports at higher port bandwidths than is feasible with electronics. Although electronic switches organized in parallel multistage fabrics can always provide the required bandwidth and number of ports, we are convinced that lower fabric-level power consumption and cooling advantages will primarily drive the use of optical switch technology. The main advantage of current optical switching technology is that the optical switch element power consumption is independent of the data rate, whereas in CMOS power consumption is proportional to the clock (i.e. data) rates. The power consumption of the optical switch control function is proportional to the packet rate.

II. RELATED WORK

Optical switching technology has been used in telephony backbone networks for a number of years. This application uses the technique of circuit-switching, wherein connections through the backbone switch are provisioned on time scales of hours or longer. Hence, using optical switch components that change state in milliseconds is perfectly acceptable. A few examples include moving mirrors [2] and polymers using thermal control [3]. For packet-switching, where connections are ideally setup only for the duration of the packet transmission time only, optical components need to change state in the micro- to nanosecond range. We refer to the time to change state as the *guard-time*, which is inserted between packets and lowers effective user bandwidth.

Optical packet routers have already been deployed, for instance using the Chiaro beam-steering method, which exhibits guard times of around 20ns [4]. A well-known technique to address long guard times while maintaining good utilization is container switching, also known as burst or envelope switching [5][6]. This technique does not provide latency on the order of less than a few hundred ns, but this is typically not a requirement for Internet packet routers.

Tunable lasers [7] and Semiconductor Optical Amplifiers (SOA) [6] are suitable optical components to build high-port-count switches with low guard-band times. Reference [7] shows a 45 ns guard band, whereas SOAs can currently achieve around 5 ns.

DARPA has recently funded two projects, IRIS and LASOR [8][9] under the Data in the Optical Domain Networking (DOD-N) program. The DOD-N focus is on Internet routers and requires the use of optical buffers. Small packet-loss rates and out-of-order packet delivery can be accepted to accommodate the small optical buffer size.

The Data Vortex project [10] specifically targets HPC interconnect and uses SOA technology. Switch contention is resolved by deflection routing, keeping the packets in the optical domain. The architecture can scale to very high port counts but has limited throughput per port.

The remainder of this paper is organized as follows: Section III defines the requirements, derives constraints and some key conclusions. In IV we synthesize the interconnect fabric architecture. Section V describes the demonstrator hardware being built, and VI discusses the results. The next steps are discussed in VII, followed by the conclusion in VIII.

III. PROBLEM DESCRIPTION

Interconnect fabrics are an essential part of large High Performance Computing (HPC) machines, whether they are used for message passing, shared memory, coherent or other programming models. We consider fabrics that address HPC machines requiring port counts from a few hundred to many thousands at a port speed of 10 GByte/s and higher. We do not address direct topologies such as *k*-ary *n*-cubes and tori which use low-radix switches. Starting from the machine level requirements, we determine the architecture for the interconnect fabric, aiming to strike a balance between technical capability and constraints as well as economical factors.

We define *fabric* as the interconnect structure between all the compute nodes of a machine, built using one or more single-stage *switches*.

The following table summarizes the fundamental requirements we assumed for the interconnect fabric.

Table 1: Key HPC fabric requirements

Switch latency	100 – 250 ns
Port count	≥ 2048
Port BW	12 GByte/s in each direction
Sustained throughput	> 95%
Minimum packet size	64 - 256 Bytes
Packet loss	Acceptable only if due to transmission errors (and corrected by retransmission)
Effective user bandwidth	≥ 75% of raw transmission bandwidth
Packet ordering	Maintained between in- and output pairs

Latency is the single most important characteristic for HPC interconnect fabrics. A contemporary target is 1 μs application to application. This includes the driver software stack and Host Channel Adapter (HCA) latency at the source and destination nodes, the switch fabric elements and time-of-flight in cables. Our target is to have less than 500 ns latency in the switch fabric, including the machine-room cabling. As an engineering choice we split the 500 ns switch fabric delay equally between the switch elements and the total cable delay. This supports fiber cabling with 250 ns time-of-flight delay for a 50-m-diameter machine room. We assume bimodal traffic: short (control) packets that require low latency and long (data) packets that require high utilization. The fabric must be able to deliver performance required for both types of traffic simultaneously.

We have observed that over the past few years, a growing fraction of the HPC machine cost goes into the interconnect fabric. A significant contributor to the cost is the cables. Therefore, we require high utilization on those cables and need to ensure that the fabric can sustain throughput close to 100%. To achieve this, the switches must be work-conserving [11], which can be viewed as that a switch output may never be idle when a packet is available somewhere in the switch for transmission on this output.

A clear trend is the growth in the number of compute nodes to several thousands as shown in recent top500 lists [12]. Hence we set the goal of at least 2048 ports at the switch level, targeting Infiniband 12x QDR (quad data rate) port rates. This yields an aggregate bandwidth of 25 TByte/s. Based on our electronic-switch sizing studies, we are convinced this is not feasible in a single-stage fabric while maintaining all requirements in Table 1 [13].

We are also convinced that using current technologies, an all-optical single-stage solution is not economically feasible whilst meeting all requirements of Table 1. The reason lies foremost in the lack of a viable optical memory technology, compounded with the time-of-flight problem. Although optical buffers exist in the form of fiber loops, and some promising new technologies are being researched such as photonic crystals [14] and ring lasers [15], there currently is no cost-effective optical memory technology available with the density and random access times offered by electronic memories. Switch buffers typically need to hold a few hundred packets in order to prevent packet loss, and need to have a per-output FIFO behavior to maintain packet ordering. Traditional supercomputing interconnect fabrics have typically used output-queued electronic switches with integrated buffers [16]. Owing to the lack of optical buffers inside the crossbar, an optical packet switch becomes a bufferless crossbar. Hence it becomes an input-queued switch which requires a mechanism to resolve the switch contention.

We choose a central scheduler approach because we need high sustained throughput and we must maintain packet ordering. The buffers are now located at the ingress nodes. The scheduler receives packet transmission requests from all of the inputs, performs arbitration, informs the inputs, and configures the switch. This scheduler is typically located very close to the switch crossbar to avoid long delays between the crossbar and the scheduler. In the case of a single-stage fabric with thousands of ports, this would be at a central location in the machine room. The minimum latency for a single-stage optical crossbar is twice the cable latency plus the scheduling and switching delay.

The cable delay is labeled RTT (round-trip time) in Fig. 1. One RTT is required to perform the request/grant cycle to scheduler *S*, one more RTT is required to transmit the data packet. This 2 RTT latency exceeds our latency goals. Furthermore we do not consider a 2048 port scheduler feasible at high speed and low latency. Therefore in conclusion we generalize that a multistage topology is required irrespective of whether electronic or optical switch elements are used.

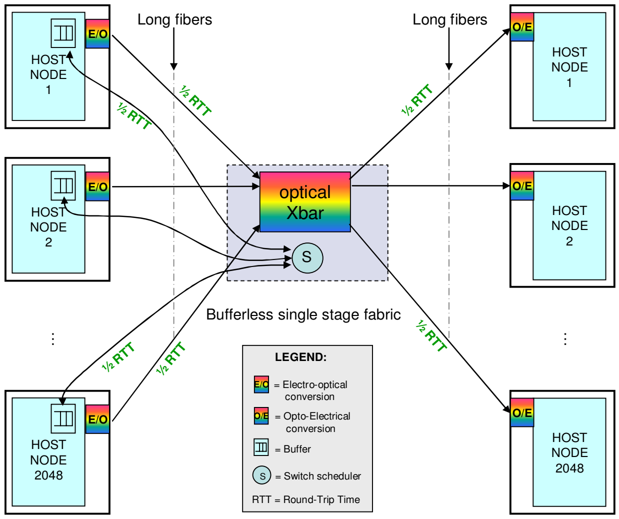

Figure 1: Control and data latency with single stage fabric

Achieving high throughput requires the use of the well-known Virtual Output Queuing (VOQ) method to resolve head-of-line blocking in bufferless crossbars [17]. Schedulers exist that can reach high throughput, but the challenge remains to achieve low scheduling latency.

In conclusion we need to solve two issues: 1) how to build low-latency HPC fabrics out of bufferless crossbar elements, which is a switch fabric architecture question. 2) Which optical switching technology to use and address additional requirements imposed by the use of optics.

IV. FABRIC ARCHITECTURE

With our insight that a multistage fabric is required, an important question is where to place the buffers. It is desirable to have the buffers at the ingress and egress of only the multistage fabric, thus, avoiding intermediate OEO conversions. Each optical switch stage needs a scheduler to configure the switch. If the buffers are placed only at the perimeter of the fabric, the schedulers need to be synchronized to set up a path through the entire multistage fabric to allow packet transmission through the fabric without storing the packet intermediately. Two issues preclude this approach: 1) the additional latency incurred in synchronizing all schedulers globally across the fabric and 2) the complexity of these schedulers to achieve high utilization.

Synchronizing these schedulers reaches the complexity of a single global multistage scheduler. We do not consider this global scheduling a viable approach for fabrics of thousands of ports and packet times of tens of nanoseconds. We thus conclude that, unfortunately, we need buffers between each optical switch element. This allows us to operate the schedulers independently of each other, thus achieving multistage scalability. It also offers lower latency. This turns the fabric into a store-and-forward architecture, which is often considered undesirable for HPC interconnects. By choosing small packet sizes at high data rates, the store-and-forward penalty becomes negligible compared with the cable delay. For instance, at 12 GByte/s a 64-Byte packet takes 5.33 ns to store in a buffer. Using small packets allows us to maintain low latency for control packets while achieving high utilization for data packets by choosing a strict priority selection mechanism at the output of each buffer throughout the fabric.

A. *Physical Buffer Placement*

Placing the buffers with each stage offers three options: 1) place buffers at the both in- and outputs, 2) at the outputs only, and 3) at the inputs only. Figure 2 shows these options. For cost reasons, we assume that the fabric is built using identical switches in each stage. Note that each switch is logically an input-buffered switch with a central scheduler, independent of what physical buffer placement option is chosen. In case of option 2 the input buffers are located in the preceding stage, hence the long scheduler connections.

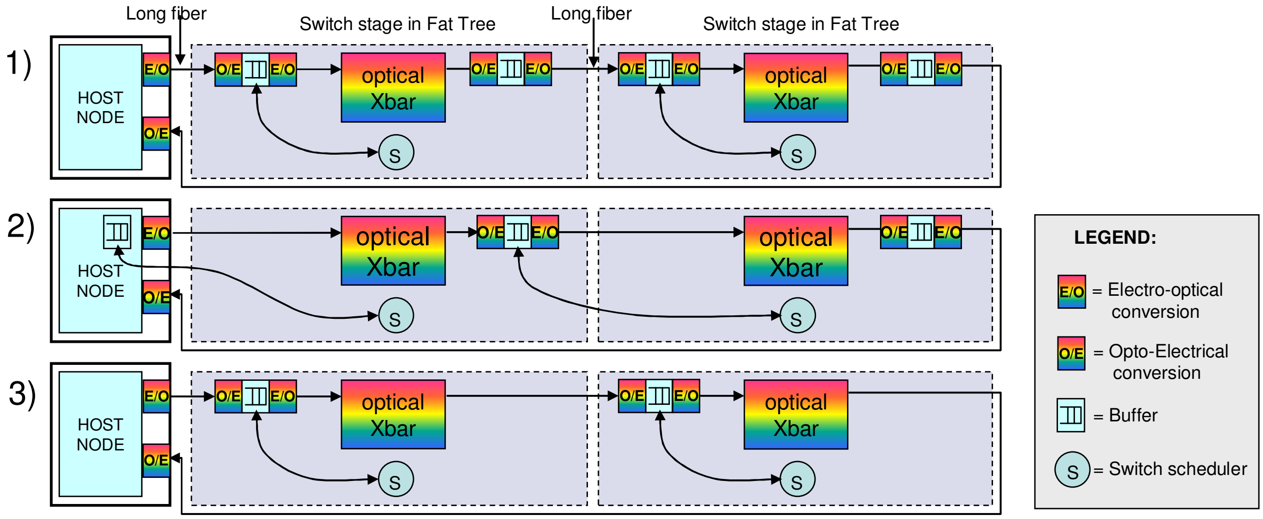

Figure 2: Buffer placement options around optical crossbar

Figure 2 shows a representative diagram of a two-stage fat tree. Although having buffers at both the in- and outputs of each stage would simplify flow control, this would require twice as many OEO conversions as the other two options, and is therefore discarded. The second option shows the buffers at the output, which results in two problems: 1) the request/grant protocol to the scheduler requires a separate long cable, and 2) this protocol gets subjected to the flight time of this cable, adding to the scheduling latency. Therefore, we select option 3, which hides the request/grant protocol inside the single-stage switch fabric, and allows us to place the buffers as close to the switch as possible, minimizing the request/grant latency. The flight time on the control cables has been studied in [18]. Option 3 combines the output buffers with the input buffers of the next stage, which has an impact on the size: These buffers must be large enough to avoid data underrun due to the long cables, and flow control becomes more complex.

B. *Flow Control*

Consider the multi-stage fabric of Fig. 3, where port 4 of switch S_1 is connected to port 2 of switch S_2. Each stage has in- and output buffers, corresponding to Fig 2 option 1. We distinguish between *local* and *remote* flow-control (FC) loops. The local loop controls the flow of packets from ingress to egress buffers within a stage, whereas the remote loops controls the flow from egress to ingress buffers between switch stages. Local FC takes advantage of the control channels between the ingress adapters and the scheduler, whereas remote FC can travel across the data channels.

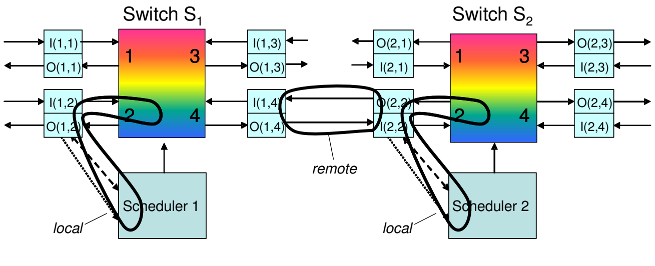

Figure 3: Local and remote flow control loops in a fat tree comprised of switches arranged with buffers at in- and output per stage

To implement option 3 of Fig. 2, we need to eliminate the output buffers, which complicate remote FC in two crucial ways: First, it is now no longer possible to have point-to-point FC on the links between switches. Instead, *every* port of S_1 must be aware of the state of the FC of the buffers of I(2,2), i.e., the one-to-one sender-receiver relation of Fig. 3 is now many-to-one. Moreover, the receiving ingress buffer can no longer immediately transfer FC information on the reverse data link, because packets are not buffered on the output side.

To avoid having to add a new out-of-band FC channel, we solve this problem by taking advantage of the centralized schedulers as FC managers and FC information relays. The ingress buffers forward incoming remote FC events to the local scheduler, which takes these into account by only issuing transmission grants for links/buffers that are available and performs the necessary bookkeeping. The ingress buffers forward their local FC information to their local scheduler, which pairs up FC information with transmission grants such that the packet launched in response to this grant will carry the FC information to the correct stage. For example, in Fig. 4, FC information produced by I(2,2) must be transported to S_1, which is only reachable via output 2 of S_2. Figure 4 illustrates the remote FC loop between S_1 [I(1,2)] and I(2,2); here, scheduler 1 relays the FC via I(2,4) to I(1,4).

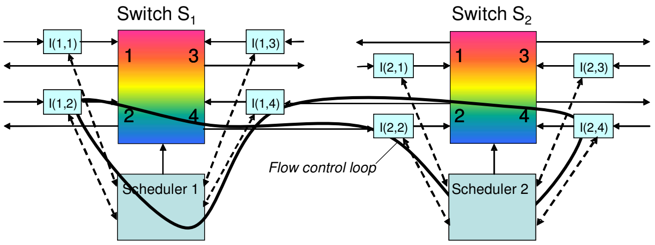

Figure 4: Flow control loop in a fat-tree network comprised of switches with input buffers per stage

This scheme solves FC using input buffers only, exploiting the central scheduler and existing links. It does not interfere with data traffic or otherwise cause performance degradation, and the FC loop has a deterministic RTT, which allows straightforward buffer sizing. The scheme is also suitable for relaying ACKs for link-level-reliable delivery. Furthermore we have shown how to make these control channels reliable in [19].

C. *Using Optics*

To use optical switching efficiently, the optical switch elements must switch with nanosecond speeds or better. This prohibits technologies that use slower physical effects (moving mirrors, heating/cooling) such as MEMS [2], LCD shutters, and index changes in waveguide structures [3]. Tunable lasers [7] are much faster, as is beam steering [4]. We consider SOAs (semiconductor optical amplifiers) to offer the best combination of optical bandwidth scalability and switching speed, and have selected this technology as the basis for our optical switch. Still, we need to budget a guard-time of a few nanoseconds during which no user data can be sent.

The total guard-time is not only caused by the optical switch element, but also by the serializer and deserializers, which for a high-speed links are no longer statically connected, as they are in electronic switches. With an optical switch, a given deserializer receives bitstreams from different serializers for different packets coming from different inputs. These bitstreams have independent phase and frequencies. We partially address this problem by ensuring a central reference-clock distribution, but phase re-acquisition is still required. In the optical community this is known as burst mode receiving (not to be confused with burst switching). Finally some portion of the guard-time is used as the packet-arrival jitter time. All packets need to arrive at the optical switching elements at the same time, while the switch reconfigures. A solution for this timing issue is proposed in [20].

Finally, for optical links the best raw Bit-Error Rate (BER) is in the range of 10^{-10} to 10^{-12}, owing to the lower dynamic range of optics as compared to copper links, which can be engineered to a raw BER of better than 10^{-17}. With the higher number of ports and links in large HPC interconnect fabrics, we chose a two-tiered approach to solving the raw BER limit. We first employ a forward error-correcting code (FEC) that results in better than 10^{-17} user BER, on top of which a hop-by-hop hardware retransmission mechanism improves this BER to better than 10^{-21}. The chosen FEC optimizes between low coding latency (i.e., short block lengths) and low overhead (i.e., long block lengths). No standard FEC code meets our requirements and we have selected a code in the class of *generalized non-binary cyclic Hamming codes (272, 256, 3)* with Galois field size 2^8 and the generator polynomial

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1.$$

This code has a block length of 256 bits, and a coding overhead of 6.25%. It corrects all single bit errors and detects all double bit and most multi-bit errors.

V. OSMOSIS DEMONSTRATOR

Based on the above considerations, IBM and Corning, Inc. are jointly building a hardware demonstrator that addresses all these technical challenges. We call our project OSMOSIS (Optical Shared MemOry Supercomputer Interconnect System) [21], in which we employ SOA technology. For reasons of cost and flexibility, we use commercially available components and Field Programmable Gate Arrays (FPGA) instead of custom ASICs. As a result, some compromises have to be made with respect to the requirements of Table 1. Notwithstanding, we can demonstrate that the requirements are achievable when building an ASIC-based and more integrated version for commercialization.

The demonstrator is a single-stage, 64-port optical switch. Each port runs at 40 Gb/s. Using a two-level (i.e., three-stage) fat-tree topology, this yields 2048 ports at the fabric level. For flexibility, the demonstrator uses buffers at both in- and egress. It uses fixed-size packets (also known as cells) of 256 byte, including the guard time, resulting in a 51.2 ns packet cycle time. We consider this baseline as one of the most difficult challenges to realize in our demonstrator because it determines the FPGA cycle times, scheduling time, and the effective user-data bandwidth. The demonstrator is expected to be complete in the first half of 2006. Figure 5 shows the system diagram. The optical crossbar implements a broadcast-and-select architecture, using Wavelength Division Multiplexing (WDM) and multiple fibers. Eight ingress

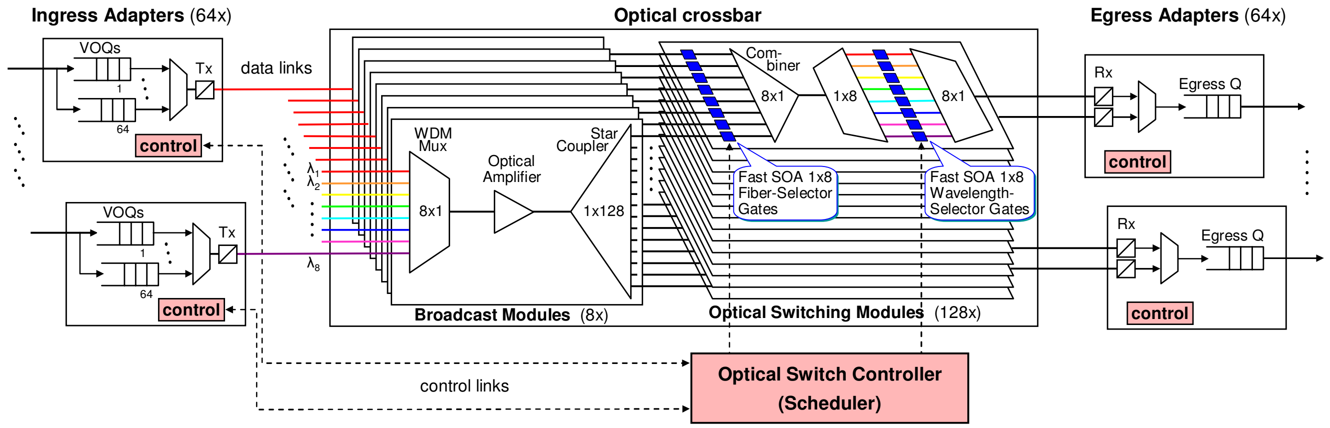

Figure 5: OSMOSIS demonstrator single stage switch system diagram

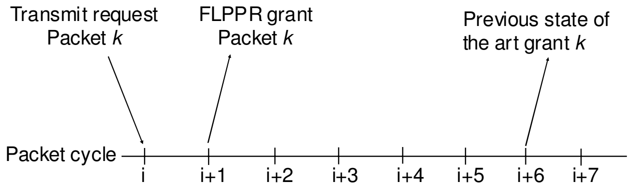

Figure 6: FLPPR request to grant latency for a 64 port switch

adapters, each using a different WDM color, are optically multiplexed onto a single fiber. Hence, eight fibers carry all the data from 64 inputs. Each of these eight fibers is optically split 128 ways, requiring optical amplification to compensate for the split loss. The switching modules consist of a fiber and a color selection stage, both built using SOAs. The SOAs can be considered as on/off devices: only one of the eight fiber-select SOAs will be turned on, allowing light to pass from one of the eight fibers. Similarly the color-select SOAs select a single color on the previously selected fiber. As the SOAs require electrical control signals and the ingress packet buffers are also electronic, the scheduler is best implemented in electronics. Our demonstrator exploits the broadcast-and-select architecture by having two paths from each input to a given output. This choice improves latency at medium loads.

It is well known that in order to achieve good utilization under various traffic conditions, the scheduler needs to perform $\log_2 N$ iterations for an N-port switch [17]. These iterations need to be completed for each packet scheduling decision, and the allowable time to complete all iterations is given by the shortest packet time. We have developed a novel crossbar scheduler algorithm that allows a parallel implementation in FPGAs for our 51.2 ns packet cycle time and 64 ports, assuming the time required to perform one iteration is also 51.2 ns. This work was recently published [22], and is called FLPPR (Fast Low-latency Parallel Pipelined aRbitration). The novelty of this algorithm is the use of parallel (sub-) schedulers in a way that minimizes the time between a request and a grant. Figure 6 shows that our algorithm needs a single packet request-to-grant latency under light to moderate loads versus $\log_2 N$ iterations for prior art.

Before building OSMOSIS, we performed detailed delay versus throughput analyses in our Omnet++ based simulation environment; some of these results were reported in [22]. The architectural choice to have dual paths improves delay versus load further, as shown schematically in Fig. 7 by the curve labeled Dual Receiver. The main advantage of the dual receiver architecture is that the delay is more or less constant for a large range of loading, and only increases significantly for high loads.

VI. RESULTS

A. System design

The OSMOSIS system design is complete. We have completed the multistage delay versus load performance analysis and closed the optical power, latency, utilization and jitter budgets. All hardware modules have been prototyped, and we are currently integrating these modules into the demonstrator system. Figure 8 shows a representative example of such a module. This picture shows half of the Optical Switching Module, where the 8 fiber select SOAs can be clearly seen. We have designed a software-based management system with a graphical user interface for the tasks of configuring and testing the system, monitoring demonstrator operation, and extracting performance values. We consider the resultant system architecture an optimal choice to meet the requirements in Table 1, and it follows our philosophy of using optics for what optics does best and electronics for what electronics does best. Forcing the entire system into a single domain does not result in an economically optimal system. Optics is used for switching and transmission at very high bandwidths, whereas electronics is used for storing data and performing the complex switch control.

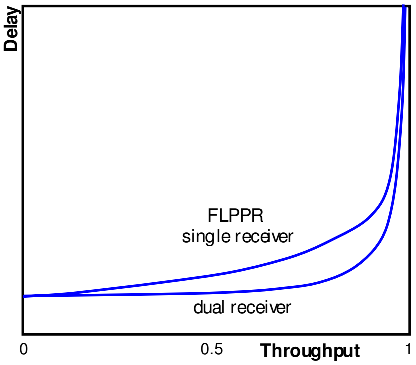

Figure 7: OSMOSIS delay versus throughput

B. Scheduler and latency

To the best of our knowledge, the OSMOSIS scheduler is the first solution for building a 64-port opto-electronic packet switch, running at 40 Gb/s, without using container switching. The scheduler is implemented in 40 high-end Xilinx and Altera FPGAs. Figure 9 shows a picture of the Optical Switch Scheduler prototype hardware. More details on the scheduler implementation have been published in [23].

The demonstrator also employs FPGAs to implement the switch data path and input control, located at the ingress adapters. Implementing the 40 Gb/s data path logic requires a large amount of pipelining, for instance for the FEC coding and decoding functions. The scheduler function, being distributed over multiple FPGAs, results in a high number of chip crossings. Notwithstanding, the demonstrator prototype has only around 1200 ns latency. A straightforward mapping of the FPGAs into ASIC technology will reduce the latency down to a few hundred nanoseconds. Latency can be further reduced by tighter integration of optics and electronics, resulting in shorter connections between the scheduler and the SOAs. The demonstrator uses multi-meter optical cables for this link. Our size analysis shows that the scheduler can be built with no more than four identical ASICs.

C. Bandwidth

OSMOSIS has an effective user bandwidth close to 75%, sustained high utilization and 64 ports at 40 Gb/s. Building fabrics with 64-port switches requires fewer stages than using electronic switches. We expect the highest possible electronic switch port count to be 32 ports for the IB 12x QDR rates, and commodity parts will probably offer only 8 to 12 ports. A 2048-port fabric needs 3

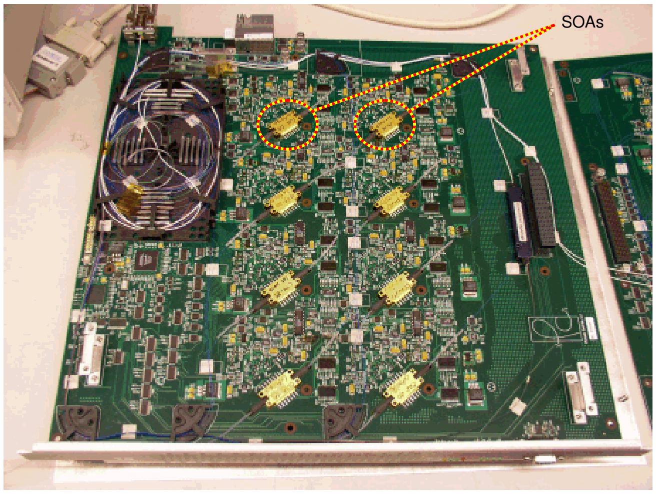

Figure 8: Optical Switching Module prototype

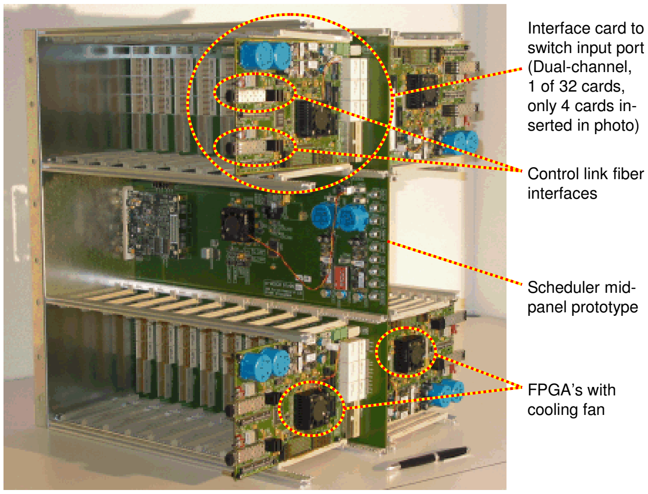
Interface card to switch input port (Dual-channel, 1 of 32 cards, only 4 cards inserted in photo)
Control link fiber interfaces
Scheduler mid-panel prototype
FPGA's with cooling fan

Figure 9: Optical Switch Scheduler Prototype

OSMOSIS stages, 5 high-end electronic switch stages and 9 stages of commodity switch chips. Each stage contributes to latency and power consumption. Compared with the high-end electronic solution, OSMOSIS saves two layers of OEO conversions in the fat tree.

D. *Comparison With Other Switch Architectures*

A single-stage OSMOSIS switch belongs to the family of input-queued centrally arbitrated bufferless crossbars using Virtual Output Queuing [17]. High-port-count and high-speed centrally scheduled crossbars have been built using burst-mode (envelope; container) switching [5][6] as a workaround to relax scheduling time constraints. Owing to the packet burst size, these architectures exhibit latencies on the order of the packet burst time for unloaded switches, which is not attractive for HPC interconnect fabrics. A key OSMOSIS novelty is the FLPPR scheduler algorithm [22], which achieves a scheduling latency of a single packet for an unloaded switch with high port counts at port speeds of 40 Gb/s and beyond. The LASOR and IRIS programs in DOD-N [8][9] use optical buffers and accept packet loss when these small buffers become full. HPC fabrics only accept packet loss due to transmission errors but not due to buffer overflow. The Birkhoff-von Neumann switch [24] is a multistage switch architecture with distributed scheduling. It can be seen as a space-time-space switch, where the first stage has a round-robin scheduler that forwards an arriving packet to the output given by the round-robin pointer. In effect, this stage shapes the arriving traffic into a uniform traffic pattern, which is sent to the second stage, which contains the buffers. The largest merit of this architecture is scalability. It is not attractive for HPC applications because of the high average switching latency of *N/2* packets for an unloaded *N*-port switch, and also because of the out-of-order packet delivery.

VII. NEXT STEPS AND TECHNOLOGY OUTLOOK

Given the signaling speed, pin limits and the current CMOS technology limits, we consider 6 – 8 Tb/s aggregate switch bandwidth around the maximum single-stage electronic limit [13]. The OSMOSIS architecture can scale to at least 50 Tb/s aggregate per stage. This aggregate can be used to scale up the number of ports, the WDM wavelengths per fiber, and the speed per port. The strength of the OSMOSIS architecture lies in the combination of WDM and space multiplexing (i.e., number of fibers) and the bandwidth supported by SOA technology. Thus 256 ports at 200 Gb/s per port are feasible, in a single stage. The FLPPR scheduler can exploit higher parallelism to perform the required additional iterations in the same time.

We expect that a straightforward mapping of the scheduler logic to ASIC will speed up the scheduler by at least a factor of four. This can be invested in making the fixed-size packet shorter and the port bandwidth higher at the same size, or a combination thereof.

The electrically controlled SOA technology can be scaled to sub-nanosecond guard times by operating it in a high current-density mode with tight optical confinement. Under such conditions, the primary optical impairment is due to cross-gain modulation (XGM) distortion wherein the return-to-zero bit pattern of one WDM channel in the SOA modulates the gain and thus distorts the other channel's amplitude. To improve operation, we propose to use differential phase-shift-keyed (DPSK) modulation rather than the usual non-return-to-zero modulation. This technique is well-known in conventional RF systems. We have conducted measurements with our SOAs operating with 8 DPSK channels at 40 Gb/s. Such a constant-envelope modulation format contains no fast optical power transients, enabling the SOAs to operate very deeply into saturation. This reduces the SOA guard times to sub-nanosecond, and moreover improves the SOA power efficiency. Figure 10 shows that a 14 dB improvement measured in SOA input loading at 1 dB OSNR penalty can be achieved by adopting DPSK rather than return-to-zero modulation. Two bit-error rates are shown, 10^{-6} and 10^{-10}. In separate measurements (not shown), the SOA-switched link operates with 3 dB lower OSNR than NRZ at any given bit-error rate, thereby increasing the signal margin or relaxing the link tolerance.

Faster SOA guard times down to the femtosecond range have been demonstrated by the University of Cambridge [25]. This approach uses the fast dynamics of cross-phase modulation within an SOA. In such optically-switched implementations, the presence of a

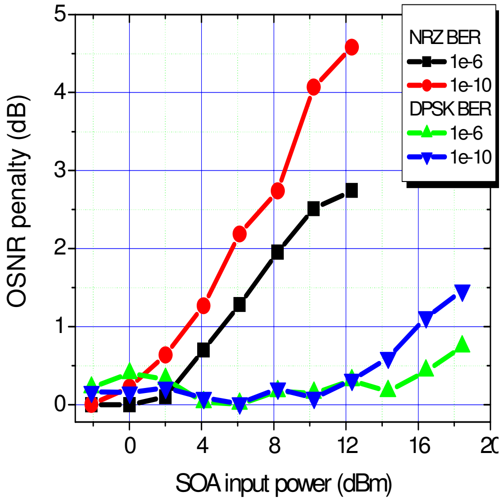

Figure 10: Optical signal-to-noise penalty as a function of input power to the SOA for DPSK and NRZ modulation formats

synchronized short optical strobe pulse almost instantaneously switches the output of a Mach-Zehnder optical interferometer.

Reducing the SOA guard times facilitates maintaining a high user-payload fraction when reducing the minimum packet size. More guard time can be removed by building custom clock and data recovery circuits that have a fast phase-lock time constant during the first few bits of a packet followed by a slow time constant to facilitate long run lengths during the remainder of the packet.

At this phase of the OSMOSIS program, we feel we have overcome all fundamental technical challenges to build a commercial high-end HPC opto-electronic interconnect fabric. We are currently pursuing commercialization of this technology, which is targeted for product availability around the end of this decade. We are confident this technology has wide applicability, and have begun to develop additional markets beside HPC using a platform-based approach. We think this approach allows us to introduce the technology faster than only pursuing the HPC market. Key to market acceptance will be to reach a fabric-level aggregate cost per bandwidth unit (e.g. $/Gb/s) that is on par with electronics-based solutions. To reach this cost point, a further integration of the optical components is an essential first step which we are currently pursuing.

In this paper, our intent was to show that scaling to large interconnect fabrics using hybrid opto-electronic switching technology is technically feasible. We rely on node designers to make high-speed busses available on the compute nodes to connect our 12 – 25 GByte/s fabric ports to, and can fully leverage the low latencies, data and packet rates supported by the interconnect.

VIII. Conclusions

Based on ultrascale HPC interconnect fabric requirements, we have shown that multistage fabric technology is required, regardless of whether electronic or optical switch elements are used. We have further shown that buffers cannot be eliminated between stages, primarily because of cable delays and scheduling complexity and lossless operation. Therefore, the notion of all-optical packet switching is not viable until optical buffers capable of storing the several hundred packets being sorted per output become economically feasible. We have shown that the use of optical switching technology is attractive for reasons of power consumption and scaling. Based on the above, we have developed an optimal hybrid opto-electronic interconnect fabric architecture targeting thousands of ports at 12 GByte/s per port with low latency. A hardware demonstrator (OSMOSIS) is currently being built. OSMOSIS shows that all fundamental technical challenges have been sufficiently addressed to build a hybrid opto-electronic packet switch and commercialization of the technology can start. We are pursuing HPC and additional markets to enable product availability around the end of this decade.

Acknowledgements

A project of this size relies on the work of many individuals. Foremost, we thank our sponsors who have gotten us thinking in this direction and subsequently have enabled this work.

Furthermore we thank all the people involved at Corning, Inc., IBM, Photonics Controls, LLC and G&O GmbH for their dedication, contributions and looking each other in the eyes.

References

[1] David A. B. Miller, “Rationale and challenges for optical interconnects to electronic chips,” Proc. IEEE 88 (6), 728-749 (2000).

[2] A. Neukermans and R. Ramaswami, “MEMS technology for optical networking applications,” IEEE Commun. Mag. 39 (1), 2001.

[3] T. Goh, M. Yasu, K. Hattori, A. Himeno, M. Okuno and Y. Ohmori, “Low loss and high extension ratio strictly nonblocking 16X16 thermooptic matrix switch on 6-in wafer using silica based planer lightwave circuit technology,” J. Lightwave Technol., Vol. 19, no. 3, Mar. 2001.

[4] E. Shekel, A. Feingold, Z. Fradkin, A. Geron, J. Levy, G. Matmon, D. Majer, E. Rafaely, M. Rudman, G. Tidhar, J. Vecht and S. Ruschin, **“**64 x 64 fast optical switching module,” Proc. Optical Fiber Communication Conference and Exhibit, 2002. OFC 2002, pp. 27–29.

[5] F. Masetti, D. Chiaroni, R. Dragnea, R. Robotham and D. Zriny, “High-speed high-capacity packet-switching fabric: a key system for required flexibility and capcity,” J. Opt. Networking 2 (7), July 2003

[6] S. Araki, Y. Suemura, N. Henmi, Y. Maeno, A. Tajima and S. Takahashi, “Highly scalable optoelectronic packet-switching fabric based on wavelength-division and space-division optical switch architecture fof use in the photonic core node,” J. Opt. Networking 2 (7), July 2003

[7] J. Gripp, M. Duelk, J.E. Simsarian, A. Bhardwaj, P. Bernasconi, O. Laznicka and M. Zirngibl, “Optical switch fabrics for ultra-high-capacity IP routers,” J. Lightwave Techn. 21 (11), 2839-2850, Nov. 2003.

[8] A. Von Lehmen, G. Clapp, J.W. Gannett, H. Kobrinski and R.A. Skoog, “Data in the optical domain technology: A network-level perspective,” Proc. 17th Annual Meeting of the IEEE Lasers and Electro-Optics Society, LEOS 2004., Vol. 1, Nov. 2004

[9] A. Willner, “Physical impairements and network limitations when interconnecting multiple DOD-N routers,” Proc. 17th Annual Meeting of the IEEE Lasers and Electro-Optics Society, LEOS 2004. Vol. 1, Nov. 2004

[10] Q. Yang and K. Bergman, “Performances of the Data Vortex switch architecture under non-uniform and bursty traffic," IEEE/OSA J. Lightwave Technol. 20 (8), 1242-1247, Aug. 2002.

[11] C. Minkenberg, “Work-conservingness of CIOQ packet switches with limited output buffers,” IEEE Commun. Lett. 6 (10), 452-454, Oct. 2002.

[12] The 24th TOP500 List, released November 8, 2004, during SC2004 in Pittsburgh, PA, http://www.top500.org

[13] F. Abel, C. Minkenberg, R. Luijten, M. Gusat, I. Iliadis, “A Four terabit packet switch supporting long round trip times“, IEEE Micro 23 (1), 10-24, Jan.-Feb. 2003.

[14] S. Noda, “Photonic nanostructures and devices based on photonic crystals,” Proc. 7th Annual Meeting of the IEEE Lasers and Electro-Optics Society, LEOS 2004, Vol. 1, Nov. 2004

[15] M.T. Hill, H.J.S. Dorren, T. de Vries, X.J.M. Leijtens, J.H. den Besten, B. Smalbrugge, Y.-S. Oei, H. Binsma, G.-D. Khoe and M.K. Smit, “A fast low-power optical memory based on coupled micro-ring lasers,” Nature 432, 2004.

[16] C.B. Stunkel and P.H. Hochschild, “SP2 high-performance switch architecture,” Proc. Hot Interconnects II, pp. 190-195, Aug. 1994.

[17] N. McKeown, V.T. Anantharam and J. Walrand, “Achieving 100% throughput in an input-queued switch,” in Proc. IEEE Infocom '96, vol. 1, pp. 296-302, San Francisco, Mar.1996.

[18] C. Minkenberg, “Performance of i-SLIP scheduling with large round-trip latency “,Proc. Workshop on High Performance Switching and Routing HPSR 2003, pp. 49-54, Turin, Italy, June 2003.

[19] C. Minkenberg, F. Abel, M. Gusat, “Reliable control protocol for crossbar arbitration”, IEEE Commun. Lett. 9 (2), 178-180, Feb. 2005.

[20] P. Mueller, R. Luijten and U. Bapst, “Hierarchical system synchronization and signaling for high-performance – low-latency interconnects”, IEEE Electro/Information Technology Conf. “EIT 2005,” Lincoln, NE, May 22-25, 2005.

[21] R. Hemenway, R. Grzybowski, C. Minkenberg, and R. Luijten, “Optical-packet-switched interconnect for supercomputer applications,” OSA J. Opt. Network. 3 (12), 900-913, Dec. 2004.

[22] C. Minkenberg, I. Iliadis, F. Abel, “Low-Latency Pipelined Crossbar Arbitration,” Proc. IEEE Globecom 2004, paper GE15-2, Dallas, TX, Nov. 2004.

[23] C. Minkenberg, F. Abel, P. Mueller, R, Krishnamurthy, M. Gusat and R. Hemenway, “Control Path Implementation for a Low-Latency Optical HPC Switch”, HOTi 2005, Stanford, CA, Aug 2005

[24] C. Chang; D. Lee; C. Yue, **“**Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches”, INFOCOM 2003, Volume 3, 30 March-3 April 2003

[25] C.K. Yow, Y.J. Chai, C.W. Tee, R. McDougall, R.V. Penty and I.H. White, “All-optical multiwavelength bypass-exchange switching using a hybrid-integrated Mach-Zehnder switch,” Proc. ECOC 2004, Paper WE4P118, Stockholm, Sweden, Sept. 2004.