Hierarchical clustering of the data vortex optical interconnection network

Cory Hawkins,[1,*] D. Scott Wills,[1,3] Odile Liboiron-Ladouceur,[2,4] and Keren Bergman[2,5]

[1]*School of Electrical and Computer Engineering, Georgia Institute of Technology,*
777 Atlantic Drive NW, Atlanta, Georgia 30332-0250, USA
[2]*Department of Electrical Engineering, Columbia University in the City of New York,*
500 West 120th Street, New York, New York 10027, USA
[3]*E-mail: scott.wills@ece.gatech.edu*
[4]*E-mail: ol2007@columbia.edu*
[5]*E-mail: bergman@ee.columbia.edu*
[*]*Corresponding author: cory@ece.gatech.edu*

Received April 18, 2007; revised July 13, 2007; accepted August 8, 2007;
published August 31, 2007 (Doc. ID 82213)

The data vortex photonic interconnection network is studied for application to clustering and hierarchical layering of nodes. Performance is examined for varying cluster counts and under loads of varying network locality. In today's technology, similar performance is attained at high network communication locality loads (>2/3), and a 19% latency reduction is obtained at the highest locality loads (>95%) for current optical switching technology. For projected future technology, the clustered system is shown to yield up to a 55% reduction in latency for applications with 2/3 or better locality. © 2007 Optical Society of America

OCIS codes: 060.0060, 060.2310, 060.4250, 200.4650.

1. Introduction

With today's high-performance computers being comprised of tens of thousands of processors (and the top performer, IBM's BlueGene/L, utilizing 131,072 processors), the trend is for more processors and more memory to achieve better performance [1]. In this paradigm of "throwing more resources at a problem" to achieve higher levels of performance, extra emphasis is placed on the interconnection network used. The processors must communicate with one another efficiently to work collaboratively on the same problem and dataset. To achieve high levels of performance, end-to-end communications latency must be minimized. Slow networks that yield multicycle delay on each message can become a bottleneck and drastically impact overall system performance [2]. To keep the message latency over long links that an expansive supercomputer would require to the desired low level and to keep throughput at a high level, optics can be leveraged to replace the commonly used electrical interconnection network. Optics allows greater data rates over long links than copper networks [3]. Optical technology affords the benefit of lack of required signal regeneration over long links, as previous research shows that amplifying the optical signal at nodes using semiconductor optical amplifiers (SOAs) is sufficient [4] and does not require explicit signal regeneration that copper technologies could require when carrying data over long distances. Optics also affords the possibility of dense wavelength division multiplexing (DWDM), which indirectly affects the latency. DWDM can be used to increase the data-carrying capacity of each link by allowing multiple wavelengths for communication of data from point to point, making messages wide in the frequency domain and short in the time domain. This makes the time to receive the entire packet shorter, even though the time to receive the first bit is the same, as multiple bits arrive in parallel at each time slice. For example, a packet can be split into ten chunks with each modulated at a different light wavelength, and the entire packet can physically traverse the long link then be received in 1/10 the time. The only stumbling block for wide-scale implementation of optical networking for supercomputing is the lack of random-access optical memory necessary for buffering. This leads to the need for optoelectric (O/E) conversions of messages within the network for storage in electrical memory. To avoid these costly conversions at every point of network contention, buffering can be eliminated through the use of deflection routing techniques that exploit an always-open path to create an all-optical end-to-end data path through the network that keeps data moving and entirely circumvents the dropping or pausing of data within the network. This effectively removes all message blocking from the system except that encountered at the input nodes when a message injection is attempted. If the network is designed with high message acceptance and with deflections that yield a low latency penalty, the full potential of the photonic network can be realized.

1.A. Data Vortex Interconnection Network

The data vortex interconnection network is explicitly designed to utilize deflection routing [5]. As illustrated in Fig. 1, the data vortex system comprises concentric, unidirectional routing cylinders (C) of nodes located at "angles" (A) around the circumference of each. Packets are input at the outermost cylinder and progress inward to the network outputs at the innermost cylinder like water swirling down a drain. Each routing cylinder has a fixed height (H), and each has a different intracylinder link arrangement that routes the packet by fixing one bit of the packet's destination height. Each node's switch uses its height as reference and compares it to the corresponding bit of the packet's destination height to make a routing decision (e.g., the outermost cylinder fixes the most significant bit of the packet height). If the bit does not match, the node uses a hop along the deflection link within the same cylinder to change the packet's height within the network. If the packet's destination height bit matches that of the current node's height, the node switches the packet inward along the ingression link to the next cylinder and toward the network's output ports. In the event that the bit matches but the inner destination node is busy with another packet, the packet is deflected along the intracylinder deflection link and suffers a two-hop deflection penalty (as the packet's bit will not match again for two more hops).

The data vortex has, since its inception, been studied for physical feasibility and function [6–9], for basic performance under synthetic traffic loads and relative performance against other well-known topologies [10], and finally has been tested as a full-scale 36-node, 12×12 switch physical implementation [11]. Most recently, Columbia University research has yielded a greater understanding of the timing and slot packet requirements of a data vortex network implementation and its underlying physical layer scalability and transparency to packet format [12–14]. Concurrently, recent Georgia Tech research has studied the impact of angle size selection on the perfor-

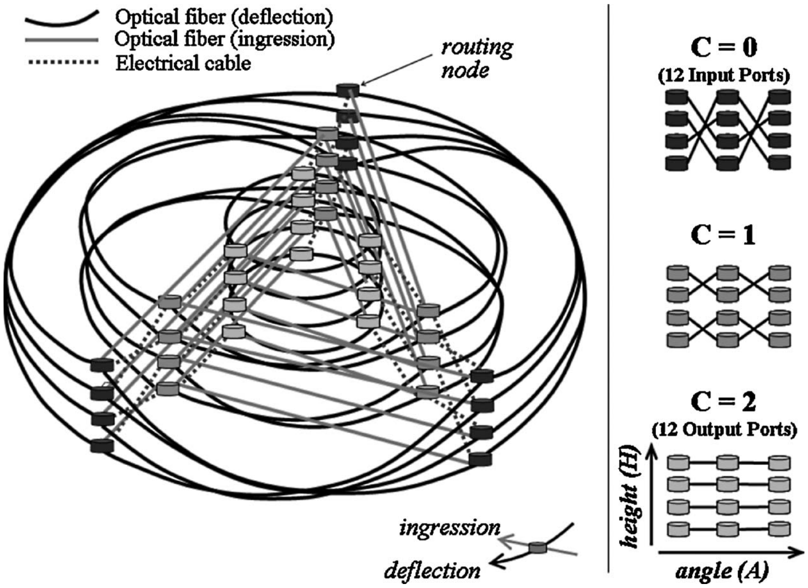

Fig. 1. Data vortex topology routes packets based on comparisons of the current switching node's height and the desired destination height in the packet's header. Each node has a deflection link to the next angle in the same cylinder and an ingression link to the next angle in the inner cylinder, and electrical signals are used to avoid contention by notifying the nodes if they need to deflect rather than ingress.

mance of a data vortex [15]. One aspect that has not been studied previously is how the data vortex could be applied to the clustering of computers to form a larger, hierarchical system to exploit potential communications locality within applications. This research addresses that option and explores the performance ramifications of clustering.

1.A.1. Hierarchical Network Layering

Applications for distributed computers often exhibit a level of spatial locality, in which processors communicate more often with their closest neighbors. To exploit this characteristic, clustering of processors can be used to keep those nearby neighbors even closer, in which subsets of the total processors are connected by smaller networks to create local clusters. These clusters are connected together by a higher-layer network to form a network hierarchy in which local data stays on the bottom (cluster) level, and less frequent traffic for other clusters utilizes the upper-level network to reach the destination cluster. One example of a network that has been studied for hierarchical layering is the de Bruijn graph. A 160-node system is proposed by Ramaswami and Sivarajan in a 1994 IEEE Transactions on Communications paper [16] in which two de Bruijn graphs are connected through 32 intermediate nodes to connect 32 clusters of five stations per cluster to form a system with 160 stations (processors) total. The clustering idea for de Bruijn graphs is continued in the work of Liu *et al.* in their SUPERCOMM/ICC '94 paper [17] in which they suggest a two-layered hierarchy of optical networks with comparisons between the de Bruijn and shufflenet topologies. The bottom layer of each of the proposed networks consists of processors connected in clusters of either shufflenets (SH) or de Bruijn (dB) networks. The clusters are connected at the top level by simple rings in opposite directions (SH–ring and dB–ring), another de Bruijn network (dB–dB), or another shufflenet (SH/SH). The results of each when simulated with the assumption of a fixed probability of intracluster communication are compared, illustrating that for larger networks (32 or more clusters of 64 processors) the rings perform almost as well as the other much more complex networks for the top-layer network. Not only does the hierarchical layering net greater performance in lower expected number of hops by exploiting intracluster locality, but this type of clustering also allows greater tolerance of link failure and a simple way to connect less-scalable, more complex networks with desired properties (like the desirable smaller diameter of de Bruijn networks) together to form much larger networks.

Like the de Bruijn graph, the data vortex can benefit from clustering. The use of clustering can improve the best-case number of hops through the network by reducing the number of cylinders experienced. In nonclustered implementations, the number of cylinders (C) in a data vortex is set by Eq. (1), and the number of input–output (I/O) ports (N) for each data vortex is set by Eq. (2), where A' is the number of angles used for injection, and H is the network height.

$$C = \log_2(H) + 1, \tag{1}$$

$$N = H * A'. \tag{2}$$

To keep the height (and thereby the number of cylinders and network diameter) small and still meet the fixed system I/O number requirement, more injection angles must be used. As shown in previous research, in order to get desirable message acceptance from the data vortex, a ratio of ~1:5 injection angles to purely routing (virtual buffering) angles is needed [15]. As defined in previous publications, "virtual buffering" refers to the capacity of non-I/O (purely routing) angles to house additional packets. A hurdle is thereby presented because when the total number of angles used in a data vortex increases beyond a certain point, the resultant backpressure from angle resolution (the circulation of message packets until they reach the destination angle) can severely degrade the performance. Having too many total angles must be avoided while meeting the virtual buffering requirement by limiting the number of injection angles used in the network.

1.A.2. Data Vortex Clustering

To cluster computers or processors and memories using data vortex networks, multiple methods can be used to connect the clusters, involving everything from the addition of angles or heights to connect to the upper-level network to simply using the existing links more effectively. The main cost of a data vortex network lies in the I/O ports because of the price of the necessary laser drivers, modulators, receivers, deserializers, and demultiplexers [15]. One of the extra incentives of applying the clustering idea to the data vortex is that the cost is very low because only additional links and low-cost switching nodes are needed: the number of I/O ports remains the same. To utilize the existing links more effectively, the upper-level data vortex network can connect the lower-level data vortex clusters at non-I/O angles. Currently, in a nonclustered data vortex network, one of the input links of the outermost cylinder's non-I/O angles and one of the output links of the innermost cylinder's non-I/O angle nodes are not utilized. These "free" links can be easily used to connect the clusters together with another (upper-level) data vortex arrangement of the same height, as shown in Fig. 2. If the upper-level network is underbuffered, one can add angles between the cluster-linked angles to form an upper level "buffer factor" (BF) as needed. Along the same vein, if the upper-level network has too many angles, one can limit the upper-level network angle count and simply use a fraction of the available angles from clusters to form a fractional BF. Using this simple methodology (free links) of connecting the clusters together with an upper-level network, no change in the topology or constituent nodes is required. Data still progresses from the outermost to the innermost cylinders of each network, and each node is still composed of a simple 2×2 optical switch as used in prior research [11,13].

However, for any clustered system, there is usually a price to pay. Using clusters for a reduction in latency is made possible in the eight-cluster example case at the cost of 25% more switching nodes (184,320 nodes versus 147,456 for the nonclustered system), but it should be kept in mind that switching nodes are simple in design and cost much less than an I/O node (currently 1/10 the cost of an I/O node). Thus, the expense is small compared to the overall system expense and is worth the performance increase gained if the system is to run loads with moderate to high locality.

2. Data Vortex Timing Requirements

The data vortex 2×2 optical switching node is illustrated in Fig. 3. The unclocked node routes one packet at a time within the slot time duration. The routing decision is made electronically using the header and frame information encoded on dedicated wavelengths in the optical packet. Meanwhile, the optical packet remains in the optical domain and is evenly split to the two SOAs used as switching elements. After extracting the frame and header information with optical bandpass filters, their signals are converted to electrical signals. The routing logic enables one of the two SOA devices such that the packet ingresses to the next cylinder or is deflected. The simple routing logic can be programmed in a complex programmable logic device (CPLD). While the routing decision is made, the packet is delayed in optical delay line. The

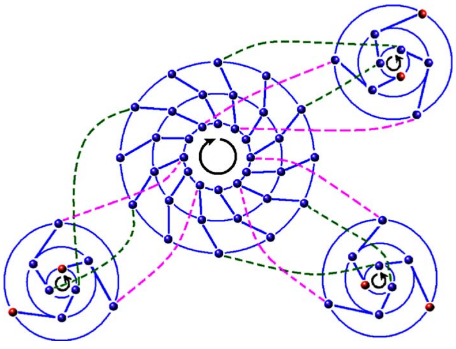

Fig. 2. Clustered data vortex system with three clusters having one input and one output angle (in red) in each, and a height of four (three cylinders) for a 12×12 network switch. The system has four processors–memories in each cluster. The upper-level network (center) is connected to the clusters at its inputs by the green links and at its outputs by the pink links. The upper network utilizes a BF of two, with twice as many angles as necessary to connect it to the clusters to add virtual buffering. Note: The intracylinder links in both the clusters and upper network are the same length, and the cluster-to-upper network links are much longer in this design. They are not shown to scale, so the connections can be seen clearly.

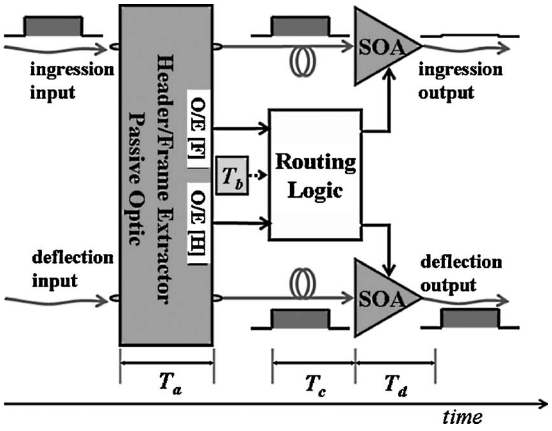

Fig. 3. Schematic of the data vortex 2×2 switching node showing latency associated with the passive components and optical receivers (O/E) converting the frame [F] and header [H] information (T_a), the deflection signal (T_b), the routing logic (T_c), and the SOA devices (T_d).

node latency is determined by the packet optical time-of-flight latency in the header–frame extractor (T_a), the CPLD propagation delay (T_c), and the SOA device transition times (T_d). The deflection mechanisms for contention resolution impose an additional timing constraint for a packet to be properly routed. The downstream stage to the routing node sends an electrical deflection signal used as an input to the routing logic. The latency (T_b) corresponds to the propagation time of the signal in a coaxial cable.

The time of propagation in a single-mode optical fiber (SMF-28) is 4.897 ns/m. In a system limited by time-of-flight latencies, the distance between in-network nodes and between a cluster and the upper-level cluster is determined by the slot time. It should be noted that the interconnected fiber length can also correspond to a multiple of the slot time duration. For example, if the slot time is 14 ns, then the interconnecting SMF-28 optical fiber must be 2.86 m for 14 ns or 5.72 m for 28 ns and so on. Because the electrical deflection signal must be received in time to generate the routing decision, the connecting deflection fiber is longer than the ingression fiber (Fig. 4). The deflection fiber length is determined by the sum of the total latencies after the header–frame extractor of the routing node ($T_b+T_c+T_d$) and the ingression fiber length. Any timing mismatch between the ingression fiber and the deflection fiber may result in fatal packet truncation. Finally, the slot time is the sum of the node latency and the deflection fiber.

In the current data vortex test bed implementation [11], the slot time is 25.7 ns, resulting from the total node latency of 15.8 ns and the latency associated with the deflection signal processing (9.9 ns). In this node design, the routing decision is implemented using discrete logic gates with a total latency of approximately 5.5 and 11.0 ns when including the O/E conversion and the SOA transition times. The latency associ-

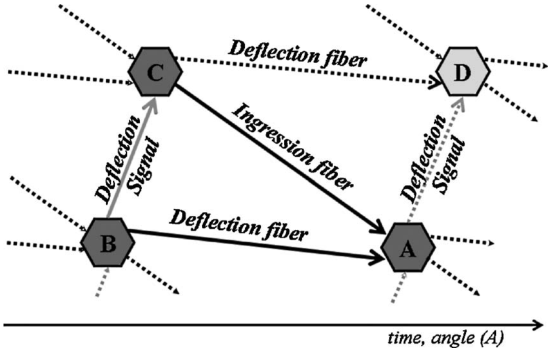

Fig. 4. Schematic representation of the in-network switching node interaction with deflection mechanism. To avoid packet collision at node A, the electrical deflection signal is transmitted by node B to node C in time to route the packet present at node C onto the deflection fiber (to node D) instead of the ingression fiber (to node A). The illustrated structure is repeated throughout the network within the clusters and the upper-level cluster regardless of the angle coordinate.

ated with the passive optical components is 4.3 ns. Consequently, the deflection fiber is approximately 2 m. In a more aggressive design approach, a CPLD with 2.5 ns propagation delay can be used and the passive optical components can be tightly spliced to minimize the superfluous fiber pigtails. With today's technology, the node latency could be minimized to 10.0 ns and the deflection signal latency down to 4 ns resulting in a packet slot time of 14 ns. This node latency can be reduced even further with integration of the components. The prospect of even lower latencies for future data vortex node generations makes the hierarchical layering of data vortex arrangements more rewarding, as will be shown in the results in later sections of this manuscript.

The total packet end-to-end delay through the nonclustered network consists of the delay incurred traversing the long link from the source node to the network input, the delay incurred routing through the network (delay from in-network hops), and the delay incurred traversing the long link from the network to the destination node. Future optical switching technology advances can reduce the middle delay factor by reducing the slot time and the interconnecting fiber length. For example, if the current 10 Gbits/s switching technology was replaced by 40 Gbits/s switching technology, the optical switching component of the delay could be cut by 75%. However, the time of propagation in single-mode fiber dictates that the delays to traverse the long links to and from the network remain large. The lengths of the fiber links to and from the network are required by the physical location of the system's processing nodes and network inputs–outputs. They are not determined by the switching time, other than the fact that they should exhibit a delay that is a multiple of the selected slot time as per the timing requirements. This long link delay can be a dominant factor versus the in-network routing delay in current technology and can be more dominant in future technology with potentially smaller switching–slot times. Clustering allows reduction of these long fiber lengths by placing a smaller network physically closer to each cluster of processing nodes.

3. Performance Study

To study the performance of data vortex hierarchical clustering, a series of data vortex configurations are examined. All systems are simulated using a custom cycle-accurate data vortex simulator written in C++ that simulates the entire network, with packets injected in the first 50,000 time slots and with 1000 subsequent noninjection time slots to clear the network of all data. The primary metric for comparison is the total percentage of packets offered that are accepted for injection, with message inputs only occurring at the outermost cylinder, as the network definition dictates. The average packet latency as measured in network hops from input to output is considered as well. In all studied systems, it is assumed that all packets are exactly one cycle in length (i.e., there is only one packet per node on any given cycle), each message is composed of exactly one packet, and packets have a randomly chosen destination address. Likewise, it is assumed that each link has the same physical latency (one hop). These assumptions are made to make the simulational results as general as possible for ease of understanding, generality of design and application, and for extrapolation to other parameter values. For instance, the assumption of one hop per physical link means that the interconnecting fiber and the switch have a combined delay that equals that of one slot–cycle time. As mentioned previously, the length of fiber could be chosen to yield a delay that is a multiple of the slot time for proper routing, but the single-slot simulation yields a baseline, clear representation of the performance under both nonclustered and clustered implementations. In addition, all messages are comprised of exactly one packet with a randomly chosen destination address. This gives a fair representation of performance under baseline synthetic traffic without favoring any one traffic type or any one supercomputing application type. Choosing a burst of packets of a chosen length for each application would represent only applications that exchange large amounts of data or that use small packet sizes. This would yield results that are harder to apply to general system design and would actually tax a recirculating network like the data vortex less, as random destination selection yields more cross-route packets than bursts of packets destined for the same address. The total packet hops are computed as the number of hops along the identical fiber links between optical switches. The message is routed to a node with the correct height in the innermost (output) cylinder, and the correct angle value is determined by angle resolution timing—represented in this simulation as a header match with an explicit header field for destination angle as proposed in previous research [11,15]. Finally, the output node is determined by random selection, as in previous studies involving the data vortex [15], but a variable factor (a locality variable) has been added to the simulation to test the impact of same-cluster communication.

3.A. Performance With Purely Random (No Locality) Traffic

When using a method such as clustering–layering to exploit locality, the system should still perform adequately with random locality applications so as to produce a general purpose system that is not handicapped to acceptable performance only with workloads that exhibit locality. The first step in an investigation into optimum network parameters for clustering performance should therefore be to test all systems with random (no locality) traffic. The clusters should be high-performance data vortex networks with adequate virtual buffering for each, using the results of the parameter study in previous research (1:5 I/O to non-I/O angles) [15]. The virtual buffering of the upper-level data vortex is important as well, as the figures in this section indicate. As in the angle study, it is shown that too little virtual buffering in the upper-level network yields poor performance, and too much buffering yields a latency penalty [15]. With the clustering–layering arrangement, however, an additional issue arises. With no locality and four clusters, there is a 75% probability that the packet is destined for another cluster, so the upper-level network gets a heavy workout at higher (45% or greater) loadings and can become saturated more easily than a cluster. The saturation of the upper-level network is a result of the fact that each cluster is injecting a majority of its packets into the upper-level network, and each packet has to progress around the circumference of the upper-level network to the destination cluster's connected links. When a packet gets to the destination cluster links, it can encounter newly injected packets in the outermost cylinder of the destination cluster. Those newly injected packets can cause deflection of the upper-level packets and force them to progress around the inner cylinder of the upper-level network again, thereby further exacerbating the backlog of packets in the upper-level network. Thus, things can slow down in the upper-level network when high loads that exhibit no locality are placed on the clusters. However, normal applications for supercomputing like those represented by the SPLASH-2 benchmark suite generate relatively infrequent memory accesses and thus infrequent interconnection network accesses [18,19]. That fact is coupled with the fact that even a 0.45 loading (a 45% probability that a packet injection will be attempted on every cycle) is a massive, unlikely loading for a current-day electrical processor utilizing an optical system running at 10 Gbits/s (with slot times in the functioning test bed currently measured at ~25 ns [14]). This makes the saturation–overloading issue a minor concern.

The upper-level network can be over-buffered and have a latency penalty from angle resolution backpressure as well, as Fig. 5 indicates. However, the number of

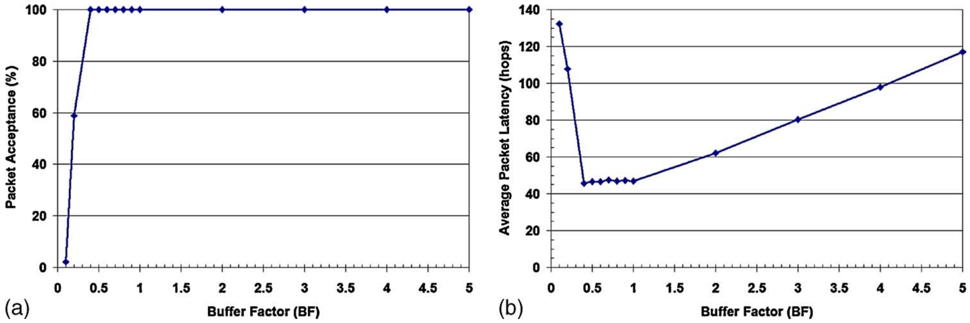

Fig. 5. Performance measures versus BF for an example system. (a) Packet acceptance versus BF for a 20% load of no locality random traffic and a system with four clusters of data vortex networks with $H=256$, $A=12$, and $A'=2$ (2048 I/O). As the plot indicates, once the virtual buffering requirement is met (BF > 0.3), the system accepts 99.9% or more of the traffic offered. (b) Average packet latency versus BF for a 20% load of no locality random traffic and a system with four clusters of data vortex networks with $H=256$, $A=12$, and $A'=2$ (2048 I/O). As indicated by the plot, there exists a BF range (BF = 0.4 to 1) in which the latency is lowest, and over BF = 1 yields reduced performance from angle resolution delay, as in nonclustered systems.

angles before the penalty is seen is larger than that of a lower-level cluster. If a packet in the top-level network is in an angle that is connected to the desired destination cluster and experiences a deflection, it simply tries again at the next available link to the same cluster, possibly the very next angle (with only a one hop penalty). As the plot in Fig. 5(b) shows, however, overbuffering by doubling the number of angles by increasing from BF=1 to BF=2 in the upper-level network while keeping the same number of links (i.e., only ten free links per cluster but 80 angles total) shows the expected performance penalty from angle resolution latency. Whole number buffer factors are mainly useful in systems that have too few free links (i.e., too few non-I/O angles) to comprise adequate buffering for the top-level network like the example in Fig. 2. Adequately buffered lower-level clusters should have enough non-I/O angles to adequately connect the clusters to the upper-level network and not require a whole-number buffer factor, unless very few clusters are used.

Table 1 contains a list of same-I/O-number (2048) clustered systems with their performance measures under a 20% load of random (nonlocality) synthetic traffic and the BF value selected for the best performance under the fixed 20% load. The fact that they each perform best with a BF of 0.8 is coincidental, as results indicate that systems with different system I/O port counts often perform better with different BF values.

Increasing the number of clusters from four to eight causes a severe penalty with nonlocality traffic. This is because the destination for each packet injected has a 7/8 probability of not being in the local cluster. Extracluster packets must traverse the entire source cluster, the upper network until they reach the destination cluster, and the destination cluster from input to output. They experience three times the number of hops or higher when contention occurs in any of the three networks—source, upper level, or destination. To minimize this effect, fewer clusters should be used or the load should exhibit enough intracluster locality to make the three-network traversal the uncommon case.

3.B. Performance With Traffic Exhibiting Locality

The locality type that is of interest to a supercomputer designer (as far as interconnection network is concerned) for a distributed shared memory machine is a combination of both spatial locality of data reference and network locality. Spatial locality is the notion that if a specific data item is accessed in memory, another data item that is spatially near that one in memory is likely to be accessed as well. Network locality refers to the way in which a processor working on a portion of the parallel program communicates with the other processors. A processor working on an application with strong network locality will communicate most with its nearest neighbors. Combining the two, one can visualize that placing processors wanting to communicate primarily with each other and with certain memory units into tight clusters can improve performance on applications that exhibit such locality. These two types of locality are both represented in this study as a “locality percent” that represents the probability that communication will take place between a processor and memory or another processor within the same cluster. Applications that exhibit such locality are those similar to the Ocean program from the SPLASH benchmark suite [20], which primarily uses communication between nearest neighbor processors to model oceanic changes and currents. Other examples include programs that model particle dynamics and force interactions between planetary bodies that are close to one another.

Continuing the example from the previous section (2048 I/O), the packet acceptance results are shown in Fig. 6 for the same system ($H=256$, $A=12$, $A'=1$, BF=0.8, four clusters) but with data added for loads exhibiting cluster locality. The plot shows the expected increase in packet acceptance under locality loads versus nonlocality load, as the average packet no longer has to take that long route from cluster to upper level

Table 1. Comparison Systems With 2048 I/O Ports and 20% Nonlocality Load

H	A	A'	Number of clusters	BF	Acceptance (%)	Average hops
512	6	1	4	0.8	99.998	37.7
256	12	2	4	0.8	99.98	46.8
256	6	1	8	0.8	96.9	250.7

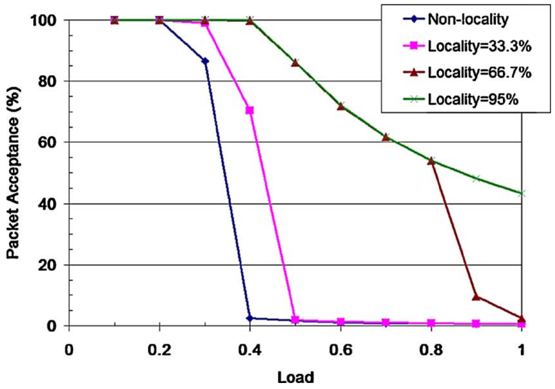

Fig. 6. Packet acceptance versus load for differing locality values random traffic and a system with four clusters of data vortex networks with $H=256$, $A=12$, $A'=2$, four clusters, and BF=0.8 (2048 I/O). As the plot indicates, higher locality levels yield increasing performance, as expected, due to the lesser strain placed on the upper-level network.

and back to a cluster. The results of locality traffic are even more pronounced when one observes the latency for a fixed 20% load. Table 2 shows the results for the comparison systems from the previous section with the same number of I/O ports under the same (now locality-based) loads. The performance results show that if locality is expected in the workload, having smaller clusters and more of them yields better performance, as the “closest neighbors” that are communicating are closer to each other. The best general purpose system from the previous section ($H=512$, $A=6$, $A'=1$, with four clusters) is no longer the winner under high-locality traffic. It should also be noted that all three systems with 2/3 or better locality perform on par with a nonclustered system with the same number of I/O ($H=2048$, $A=6$, $A'=1$), which accepts 100% of traffic and exhibits an average latency of 21.1 hops under 20% random synthetic load. They also all three outperform the single system in latency for 95% locality traffic while maintaining the acceptance of more than 99.999% of all offered packets.

4. Designing Using the Results

Figure 7 shows a sample supercomputing complex that houses 2048 processing nodes with a central data vortex network. The scale of the figure is based on that of the Japan Earth Simulator (JES) [21,22]. The JES has a facility size of 50 m by 65 m (3250 m^2) with 640 processing nodes (1.4 m by 1.0 m cabinets that each contain eight processors and 16 Gbytes of memory). If that number of processing nodes is multiplied by 3.2 to form a 2048-node system, it stands to reason that the facility size would at least double to accommodate the extra nodes. The greater number of processors yields additional need for cooling space to help dissipate the large quantity of heat produced, more maintenance–access areas around nodes, and more networking cabinets, possibly requiring an even larger facility. However, the 2048 nodes could be arranged in an array of 32 by 64 nodes within a conservatively sized 64 m by 128 m system facility, as shown in Fig. 7(a). Assuming uniform distribution and 2 m^2 for each processing node cabinet, space around it for access and cooling, and supporting network fibers, wires, and cables, the average distance from any cabinet in the facility to the central network is ~38 m (represented by the red line), and the worst-case distance is ~68.8 m.

Table 2. System Performance for 2048 I/O and Fixed 20% Load

Locality	(H,A,A')	(512,6,1)	(256,12,2)	(256,6,1)
	Clusters	4	4	8
(%)	BF	0.8	0.8	0.8
33.3	Acceptance (%)	99.98	99.98	99.999
	Latency (hops)	35.2	41.7	43.1
66.7	Acceptance (%)	99.998	99.998	99.999
	Latency (hops)	26.2	29.1	26.5
95	Acceptance (%)	99.9995	99.9998	99.9998
	Latency (hops)	18.7	21.1	16.8

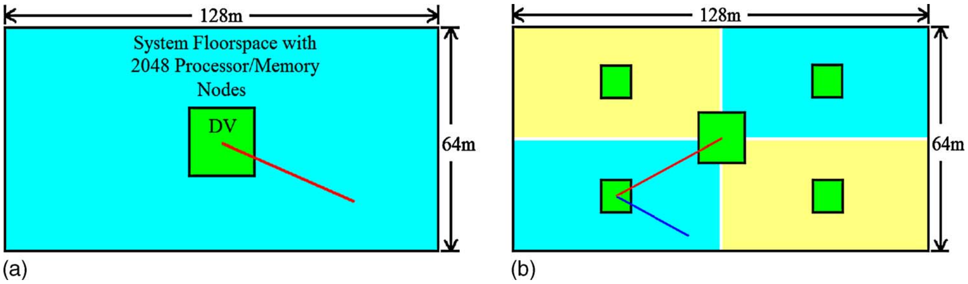

Fig. 7. Floor plan of an example supercomputing facility with 2048 processor–memory nodes. (a) Central system requires long fiber links (in red) from the nodes to the switch in the middle. (b) Clustered system with just four clusters of 512 nodes each cuts the average fiber length between processor–memory nodes and the network (in blue) by about half that of the nonclustered system.

As Fig. 7(b) illustrates, slicing the facility into four clusters of processing nodes, each with its own local data vortex, yields much shorter connections to the local network (19 m on average represented by the blue link and ~34.4 m worst case), and a link from each cluster to the upper-level network in the center of the facility is ~35.8 m long, making the average worst-case, one-way link from processing node to the central system only 54.8 m long instead of the 68.8 m connection fiber length of the nonclustered system. Utilizing eight clusters cuts the average link from local cluster network to processor–memory node almost in half again (to ~12.2 m).

Using this supercomputer facility layout, the effects of the latency from long links to and from the network and from the clusters to the upper-level network can be illustrated. As Fig. 8 shows, cutting the long links using clustering reduces that component of the total end-to-end latency as workload locality increases. When the most common packet only has to travel to the local cluster and back and avoids the (farther away) upper-level network, the average packet experiences less delay from the links. However, the switching time is still dominant, due to the long slot time of 25.7 ns that is currently in use. In addition, contention within the network from increasing the workload is illustrated, as switching latency increases with greater workload. Latency versus loading for the nonclustered data vortex has been studied in previous published works [7,10,15]. The benefits of clustering when using locality-based workloads include a reduction of the switching component of the latency as well as a reduction of the latency from long fibers to and from the network.

The end-to-end latencies are shown in Fig. 9 for a 512-height system with four clusters, a 256-height system with eight clusters, and a single, nonclustered system for comparison. As mentioned in Section 2, the CPLD that controls the switching logic of the 2×2 nodes can be replaced by a faster CPLD, the SOA transition time can be minimized with better packaging, and the optical passive components can be integrated while taking advantage of future advances in semiconductor components [23]. Hence, it is realistic to assume that the minimum slot time could be shorter than the

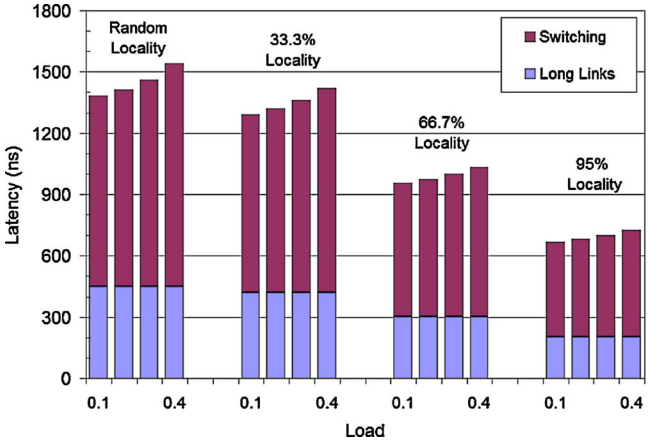

Fig. 8. Message total time of flight, factoring in fiber length and time per in-network hop with current day (25.7 ns) slot time and increasing workload for varying levels of workload locality. The system has four clusters, $H=512$, $A=6$, $A'=1$, and BF=0.8.

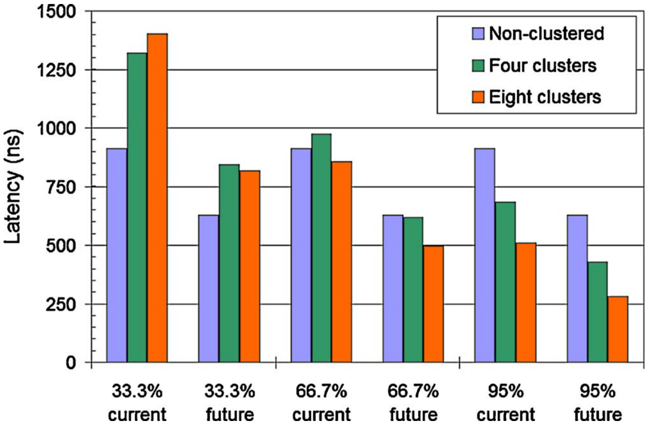

Fig. 9. Message total time of flight, factoring in fiber length and time per in-network hop with current day (25.6 ns slot time) and projected future (12 ns slot time) switching times. The single, nonclustered system has $H=2048$, $A=6$, and $A'=1$; the four-clusters system has $H=512$, $A=6$, $A'=1$, and BF=0.8, the eight-clusters system has $H=256$, $A=6$, $A'=1$, and BF=0.8, and all are under a 20% load.

actual propagation time between nodes, such that the switching time becomes less than the time of flight along the length of fiber. Clustering large systems allows reduction in the fiber length between nodes and between processors–memories and the network, and hence the overall latencies are reduced. With the reduction in node latencies, the communication latency in future systems will be determined by the distance separating the clusters and by the number of hops through which packets propagate. As the figure indicates, the performance of the data vortex when hierarchically layered with four or eight clusters is on par with a same-size nonclustered data vortex for 66.7% locality with current technology and much better only for higher locality levels. However, once the impact of reduction of switching time is factored in to the latencies (with a slot time of 12 ns), it is evident how much clustering can improve performance in future systems. Utilizing eight clusters, it is possible to reduce latency by 20.6% for only 66.7% locality loads, and it is possible to reduce the latency by 55.3% for 95% locality loads. In applications such as ocean modeling, particle dynamics, and astrophysical studies of planetary body interaction, nearest neighbor locality is especially high, and 66.7% locality is not an unreasonable amount for other applications as well—one in three messages is exchanged with an extracluster node, representing a large amount of sharing. For instance, in image processing, most nodes only communicate with their four nearest neighbors when the nodes are logically arranged in a grid arrangement. Intelligently mapping the processor–memory nodes onto clusters can effectively employ a clustered data vortex topology arrangement and exploit that locality to keep latency much lower on average.

5. Conclusions

Clustering affords many benefits for a photonic network such as the data vortex. The most important reason to use clustering with the data vortex if high locality is expected is the fact that the long fibers that connect processors and memories to the input and output nodes can be greatly shortened. This reduces the time of flight for messages traveling to the network and returning. Additional reasons to use clustering are the performance gain from making intracluster message delays smaller for locality loads. In addition, clusters can be added to an existing clustered system to get higher I/O counts without having to tear down all of the connections and start over again, and there is greater link failure tolerance from the smaller probability that a given packet will need to traverse a link in another cluster that may have failed.

A designer of a supercomputing interconnection network who expects no nearest neighbor locality from his or her application is not going to benefit as much in performance from the clustering and layering of data vortex networks to form a hierarchy, versus a single data vortex system. However, for current-day technology and 1/3 or better locality, the clustered system will perform on par with a nonclustered system, and the clustered system will actually outperform the nonclustered system if 95% or better locality is expected. Future technology trends uncover the prime reason to use clustering in systems such as the data vortex: reduction in average fiber length traversed. As the switching time decreases with improving technology, the time of flight largely becomes dependent on the distance that each packet must traverse along the optical fiber. Reducing that distance equals reducing average packet latency, and clustering is a good way to reduce it for high- or even moderate-locality workloads. With projected future switching nodes, there is a 55% reduction in average latency for loads with only 66.7% locality.

References and Links

1. TOP500.org, "TOP500 List for June 2006," http://www.top500.org/lists/2006/06.
2. D. Dai and D. K. Panda, "How much does network contention affect distributed shared memory performance?" in *Proceedings of the International Conference on Parallel Processing* (IEEE, 1997), pp. 454–461.
3. G. P. Agrawal, *Fiber-Optic Communication Systems* (Wiley, 2002).
4. O. Liboiron-Ladouceur, W. Lu, B. A. Small, and K. Bergman, "Physical layer scalability demonstration of a WDM packet interconnection network," in *Proceedings of the 17th Annual Meeting of the IEEE Laser and Electro-Optics Society* (IEEE, 2004), pp. 567–568.
5. C. S. Reed, "Multiple level minimum logic network," U.S. patent 5,996,020 (30 November 1999).
6. Q. Yang, K. Bergman, G. D. Hughes, and F. G. Johnson, "WDM packet routing for high-capacity data networks," J. Lightwave Technol. **19**, 1420–1426 (2001).
7. Q. Yang and K. Bergman, "Performances of the data vortex switch architecture under nonuniform and bursty traffic," J. Lightwave Technol. **20**, 1242–1247 (2002).
8. Q. Yang, "Optical packet switching for high-performance computing," Ph.D. dissertation (Princeton University, 2002).
9. B. A. Small, O. Liboiron-Ladouceur, A. Shacham, J. P. Mack, and K. Bergman, "Demonstration of a complete 12-port terabit capacity optical packet switching fabric," in *Proceedings of the Optical Fiber Communications (OFC) Conference* (Optical Society of America, 2005), paper OWK1.
10. C. Hawkins, B. A. Small, D. S. Wills, and K. Bergman, "The data vortex, an all optical path multicomputer interconnection network," IEEE Trans. Parallel Distrib. Syst. **18**, 409–420 (2007).
11. A. Shacham, B. A. Small, O. Liboiron-Ladouceur, and K. Bergman, "A fully implemented 12×12 data vortex optical packet switching interconnection network," J. Lightwave Technol. **23**, 3066–3075 (2005).
12. B. A. Small and K. Bergman, "Slot timing considerations in optical packet switching networks," IEEE Photon. Technol. Lett. **17**, 2478–2480 (2005).
13. O. Liboiron-Ladouceur, B. A. Small, and K. Bergman, "Physical layer scalability of WDM optical packet interconnection networks," J. Lightwave Technol. **24**, 262–270 (2006).
14. B. A. Small, B. G. Lee, and K. Bergman, "Flexibility of optical packet format in a complete 12×12 data vortex network," IEEE Photon. Technol. Lett. **18**, 1693–1695 (2006).
15. C. Hawkins and D. S. Wills, "Impact of number of angles on the performance of the data vortex optical interconnection network," J. Lightwave Technol. **24**, 3288–3294 (2006).
16. R. Ramaswami and K. N. Sivarajan, "A packet-switched multihop lightwave network using subcarrier and wavelength division multiplexing," IEEE Trans. Commun. **42**, 1198–1211 (1994).
17. G. Liu, K. Y. Lee, and H. F. Jordan, "Hierarchical networks for optical communications," in *Proceedings of the IEEE International Conference on Communications (SUPERCOMM/ICC)* (IEEE, 1994), Vol. 3, pp. 1664–1668.
18. S. Petit, J. Sahuquillo, and A. Pont, "Characterizing parallel workloads to reduce multiple writer overhead in shared virtual memory systems," in *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-Based Processing* (IEEE, 2002), pp. 261–268.
19. D. Marinov, D. Magdic, A. Milenkovic, J. Protic, I. Tartalja, and V. Milutinovic, "Scowl: a tool for characterization of parallel workload and its use on SPLASH-2 application suite," in *Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (IEEE, 2000), pp. 207–213.
20. J. P. Singh, W. Weber, and A. Gupta, "SPLASH: Stanford parallel applications for shared memory," Technical Report, Computer Systems Laboratory (Stanford University, 1991).
21. M. Yokokawa, "Present status of development of the Earth Simulator," in *Innovative Architecture for Future Generation High-Performance Processors and Systems* (IEEE, 2001), pp. 93–99.
22. Earth Simulator Center, "Earth Simulator: hardware," http://www.es.jamstec.go.jp/esc/eng/ES/hardware.html.
23. International Technology Roadmap for Semiconductors, "ITRS 2005 Edition," http://www.itrs.net/Links/2005ITRS/Home2005.htm.