Locality Aware MPI Communication on a Commodity Opto-Electronic Hybrid Network

Shin'ichiro Takizawa
Tokyo Institute of Technology
takizawa@matsulab.is.titech.ac.jp

Toshio Endo
Tokyo Institute of Technology
endo@matsulab.is.titech.ac.jp

Satoshi Matsuoka
Tokyo Institute of Technology
National Institute of Informatics
matsu@is.titech.ac.jp

Abstract

Future supercomputers with millions of processors would pose significant challenges in their interconnection networks due to difficulty in design constraints such as space, cable length, cost, power consumption, etc. Instead of huge switches or bisection bandwidth restricted topologies such as a torus, we propose a network which utilizes both fully-connected lower-bandwidth electronic packet switching (EPS) network and low-power optical circuit switching (OCS) network. Optical circuits, connected sparingly to only a limited set of nodes to conserve power and cost, are used in a supplemental fashion as "shortcut" routes only when a node communicates substantially across EPS switches, while short latency communication is handled by EPS only. Our MPI inter-node communication algorithm accommodates for such a network by appropriate scheduling of nodes according to application communication patterns, in particular utilizing relatively high EPS local switch bandwidth to forward messages to nodes with optical connections for shortcutting in order to maximize overall throughput. Simulation studies confirm that our proposal effectively avoids contentions in the network in high-bandwidth applications with nominal additions of optical circuitry to existing machines.

1. Introduction

Future petascale computing systems will embody hundreds to millions of processor cores due to moderate pace of clock frequency improvement of the cores themselves. For such systems, it will become unfeasible to construct a fully-connected packet networks with extremely high bisection bandwidth as had been done in the past machines, such as fat-trees or crossbars, due to various design constraints caused by superlinear growth of the switch fabric, such as power consumption, installation space, sheer cost, reliability, etc.

The only feasible solution has so far been to adopt topologies with restricted bisection bandwidth. For example, IBM Blue Gene/L embodies 65,536 nodes interconnected with a 3D torus network where processors can directly communicate with only six adjacent neighbors for inter-process communication [5], supplemented by separate networks for collective communication. Such a design can be considered feasible only because of customized design of the machine. As another example, TSUBAME Grid Cluster at Tokyo Institute of Technology facilitates 10,480 AMD Opteron processor cores on 655 nodes and achieves fully connectivity with a tree-topology network using eight 288-port InfiniBand switches, but suffers from lower bisection bandwidth as there is a 1 to 5 upstream vs. downstream bandwidth ratio at end-tier switches to reduce network complexity[13]. TACC Ranger [2] evolves from the TSUBAME design but for full connectivity it utilizes 3456-port InfiniBand switches—this is not without a tradeoff in cost, difficulty in cabling esp. the length and space restrictions, and moreover, cost issues, and may not be a feasibly applicable solution as systems grow larger in size.

On the other hand, it is reported that certain classes of MPI applications exhibits significant communication locality where each MPI process conducts point-to-point communication with only a limited number of other processes [14, 8]. For such applications, network topologies with high bisection bandwidth would not be necessary, resulting in a network with low bisection bandwidth, high locality and lower cost. However, such a network will suffer greatly when an application with high global communication re-

quirement arises. Hence past machines accommodated for the latter "worst case" but is starting to be unfeasible as described.

Instead, we propose a hybrid interconnect network that can be easily and cheaply constructed in commodity space, scalable, and can be designed in a way so as even to supplement existing networks on clusters. We combine an Electronic Packet Switching (EPS) network with an Optical Circuit Switching (OCS) network. Our EPS portion will embody relatively lower bandwidth in upstream links using inexpensive off-the-shelf components. Our OCS network, on the other hand, will exhibit fast point-to-point communication of direct optical paths, but will only be connected to a limited set of compute nodes, and moreover, would only be able to accommodate a limited set of connectivity at the same time. So the latter is used in a supplemental fashion as a "shortcut" path when contention would arise in the EPS portion due to restrictions of the upstream links—another reason for using as a shortcut is because inexpensive OCS networks involve ten millisecond latency in assign/release of an optical path due to the mechanical latency of micro electro-mechanical systems (MEMS) devices [7], which is too high for most MPI applications.

As such, we devise a locality aware MPI communication algorithm that will properly allocate the rank processes so that communication will be mostly localized to end-tier switches, while long-haul, high bandwidth communication will be forwarded to the shortcut OCS pathways. Because of limitations in the number of pathways and the associated latency, we must "spread out" i.e., effectively load balance the communication such that OCS pathways are utilized in a best fashion, only used in inter-switch situations. Preliminary simulations confirm that our scheme will scale favorably for high-locality and high-bandwidth applications. We also confirmed that the load of the topmost switch in the EPS was very low, and the performance would be not affected by external communication such as storage I/O sharing the same network.

2. The Hybrid Commodity EPS-OCS Network

Our proposal is a hybrid interconnect as depicted in Figure 1. Each compute node connects to both the commodity EPS network and some to the commodity OCS network. A compute node which has access to OCS portion has two links to an edge OCS switch; one is for transmitting data on a light circuit and the other is for controlling circuits. We chose MEMS based OCS network because it is easy to create from current products and it is said that very large-port-count switches could be made using 3D MEMS technique[7].

EPS portion must be a fully-connected network, but low upstream bandwidth is allowed such as a simple tree. The

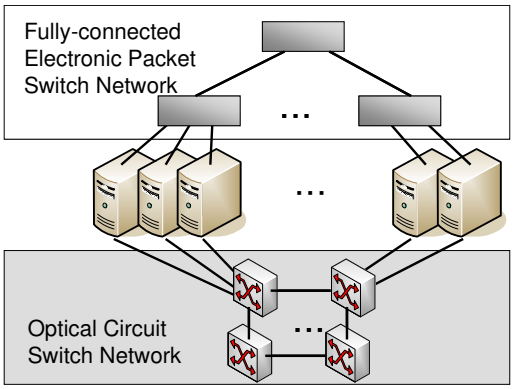

Figure 1. Hybrid EPS-OCS network

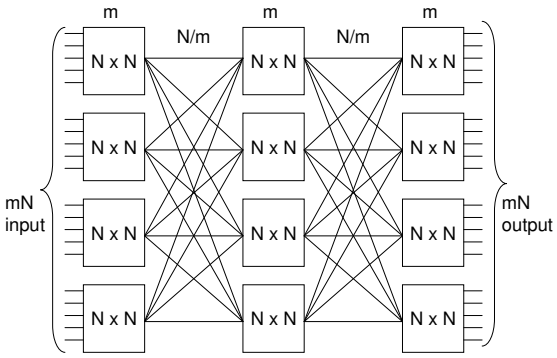

Figure 2. Rearrangeably nonblocking clos connected OCS network

OCS portion can be constructed in a low-cost fashion as a rearrangeably nonblocking clos network, meaning that an unused input on an ingress switch can always be connected to an unused output on an egress switch, using $3m$ banks of $N \times N$ (input port $\times$ output port) circuit switches as shown in Figure 2, supporting $m \times N$ compute nodes in total. We could further reduce cost by using smaller number of switches and constructing a non-rearrangeably nonblocking network, where OCS pathway assignment failure could occur as a tradeoff. Optical burst switching technique is used in the OCS such that assign/release delay of circuits consumes approximately ten milliseconds. Each compute node has only one optical NIC at max, meaning that only a single OCS pathway can be established with some other compute node.We assume that end-tier EPS bandwidth is largely sufficient for each compute node, and the fundamental bandwidth limitation occurs due to insufficiency in the bisection bandwidth. Optical circuits are used as shortcut paths only when a compute node under one end-tier switch communicates with another one under another switch.

Our proposed network can be constructed with low cost

because it does not use expensive high bandwidth EPS network, especially very large switches and its associated cabling. The OCS network is also low cost and low power in that it does not need expensive OEO (Optical Electrical Optical) conversions. Moreover, it is easy to construct our network by supplementing existing systems by only adding the OCS components.

3. Locality Aware MPI Communication Algorithm on Our Proposed Hybrid Network

In order to effectively exploit the proposed network, MPI communication must effectively take full advantage of both the locality and the availability of the shortcut OCS pathway. Here, we present two algorithms and compare the results in our evaluation section. For simplicity, we assume that a single process will execute on each compute node.

The two constituents of our hybrid network will behave entirely differently. The EPS will accommodate numerous small, low latency packets easily, while contentions will quickly occur in a disastrous fashion for high-bandwidth, all-to-all communication because of low bisection bandwidth on the upstream links. On the other hand, although sending multitudes of small packets over OCS will be disastrous due to substantial overhead in optical burst switching, once a link is established communication will be very fast in 10s of gigabits using wavelength division multiplexing (WDM) technologies. So the central issue is how to combine such divergent characteristics in our MPI communication algorithm so that only their advantages are exploited at appropriate times, such that high performance is achieved as if there is an EPS with extreme bisection while preserving low latency. To be more specific, the MPI communication algorithm should be able to make decisions not only merely to choose one network over the other. Rather, the key would be to investigate ways to utilize them effectively in a *combined* fashion, even within a single MPI communication primitive. For this purpose, the MPI system should be aware of both the network hardware as well as the communication characteristics of the application itself, and optimize accordingly with hiding details of EPS/OCS management from users.

As a first step, we identify two conditions that must be satisfied simultaneously when an MPI communication should be routed through the OCS network:

- When a message size is greater than a threshold determined by the bandwidth-delay product of the OCS link.
- When the message is routed to a different EPS switch in the EPS network.

We used the bandwidth-delay product, defined as the minimum message size that can theoretically saturate a link, as the threshold value in the first condition for effectively utilizing high bandwidth of an OCS link. If all nodes are connected to the OCS network, and we have arbitrary discretions on establishment of OCS pathways between all nodes simultaneously, our algorithm would be simple. However, in reality, limitations in both resources could arise in the optical network due to hardware or cost restrictions, as well as usages by other applications. In such a case, we construct hybrid electric-optical paths, where a pair of nodes with available OCS pathway effectively serves as router nodes that conduct EO-OE conversions.For example in Fig 3, there is an OCS pathway between node n_k and node n_{k+l}, such that when node n_0 sends a message to node n_m the message is forwarded along the path indicated by red dotted line. Note that, multiple EO-OE paths will be sharing the same optical paths; as such, effective scheduling and allocation of optical paths will become significant part of our overall system.

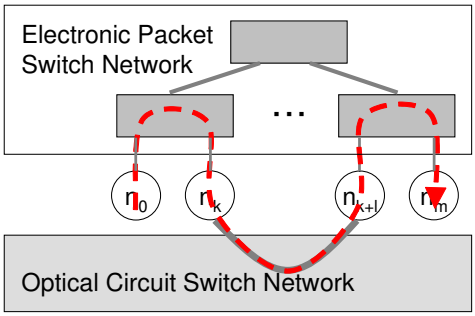

Figure 3. EO-OE Shortcut Forwarding of Messages Across Switches

To achieve this goal, we adopt a method which builds an EPS and OCS hybrid topology using available circuits based on application MPI communication patterns, such that the sharing is maximized up to the bandwidth afforded by the EPS end-tier bandwidth of the forwarding pair of nodes. A simpler approach to the problem is to pre-execute or sample-execute a given application beforehand to obtain its communication pattern by overwriting communication routines, and allocate the pre-built hybrid network, where OCS pathways are already determined, whereas decisions on routing either through the EPS network or EO-OE shortcut are performed dynamically. Especially in case of applications which have iterative codes, we can do all these automatically by inserting necessary codes with help of compiler technique at heads or tails of iterations.

We now describe the details of our algorithm, which consist of three major steps.

3.1. Obtaining network topology and application MPI communication pattern information

The network information required are 1) bandwidth of downstream links to the end-tier switches from individual nodes in the EPS network, 2) bandwidth of the OCS network, and 3) assignment of MPI processes onto the edge-level EPS switches. Although there are various ways of determining the network bandwidth automatically, here for simplicity we assume that theoretical bandwidth of links can be fetched from files. The assignment of MPI processes can be obtained from an automated means such as exchanging the IP address of each process using MPI_Allgather and estimating IDs of EPS switches where each process would be from a given address range. We assume that we also know the topology of the overall network. The information required as application communication pattern are ranks (process IDs) of message receiver processes and message size of point-to-point send calls. Only send calls whose message size are bigger than the bandwidth-delay product of the OCS network are of our interest.

3.2. Possibly Grouping Processes and Assigning Optical Circuit Pathways

We next attempt to "group" the MPI processes into smaller number of end-tier switches in the EPS network as much as possible.In case of networks which have only EPS portion, the EPS upstream links are used for inter-group communication, but our plan is to use OCS pathways which connect representative processes in each group. An important thing here is to conserve the usage of OCS pathways.

If the mapping of MPI rank to physical nodes is fixed by the system, we employ a scheme, named **Switch Partitioning (SP)**, as follows: based on the communication pattern, we assign cross-switch OCS pathways to processes that communicate with other processes under other EPS switches, eagerly assigning available OCS pathways in a round robin fashion between EPS switch pairs. If there are process pairs that result in the same inter-switch communication, we assign OCS pathway to a pair which exchanged messages most frequently, if there is a need for a tiebreaker, that to a pair whose edge process has the smallest rank. If there are OCS pathways that remain after the initial assignment, then we can go on a second round of assignment. This scheme is easy to implement, but the drawback is that it requires relatively large number of OCS pathways because communication are not localized enough, and some inter-switch communication fall back on EPS links with possibly losing performance.

A better scheme, named **Communication Partitioning (CP)**, uses the communication pattern to physically grouping processes, re-allocating frequently-communicating processes to the same end-tier switch in an optimal fashion. It then uses the same strategy as SP to allocate the optical paths. Although effective, the scheme assumes the presence of intricate control in process assignment or some form of process migration.

Note that, in either scheme, because we use a physical node to forward messages to different EPS switches, contention can happen at the EPS links on such nodes. One way to avoid such a situation is to use fatter EPS links to selected number of nodes, and proactively assign those as the forwarding nodes. Many switches are in fact configured in this way—for example, a gigabit switch with 10GbE uplink is commonplace. In this case, the OCS pathways effectively form a distributed network as a replacement for a large central switch fabric that connects to such uplinks.

3.3. Creating the Forwarding Tables

Finally forwarding tables are created so that messages can be forwarded on the created network. Our algorithm is based on a distance vector that uses the (reciprocal value of) network bandwidth as the metric. When there are multiple OCS pathways between two EPS switches, forwarding tables are created to load balance the circuits. Conversely, when there are no paths between switches A and B but there are paths between A–C and B–C, multi-hop forwarding is performed (via C in this case) to avoid the use of EPS network. When multi-hop forwarding is not possible, the forwarding table is nullified to use EPS upstream links as a fallback.

4. Evaluation

4.1. Evaluation Settings

We evaluate our communication algorithm on our hybrid network through comparison with EPS-only networks using MPI application benchmarks. The first benchmark is five iterations of communication where each node is arranged on an 8×16 grid and exchanges 40MB messages with its four neighbors ("Neighbor" in the following). We also use CG and MG in NAS Parallel Benchmarks [6] with class C, whose behavior is expected to be similar to realistic HPC applications. Neighbor and MG represent the type of application whose communication pattern is locality dominant and CG represents bisection bandwidth dominated applications. Our evaluation consists of two steps, (1) obtaining traces of applications, and (2) replaying them on a simulator.

In the first step, we run applications on a parallel environment to create a trace file.During parallel execution, we record information of MPI events and CPU events, each of

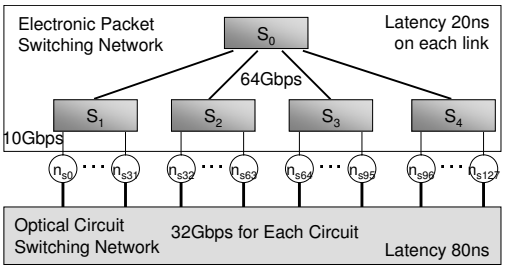

Figure 4. Simulation system environment

Table 1. Simulation system parameters

Parameter	Value
Node count	128
Node count under each packet switch	32
CPU speed of node	1.6GHz
Link speed in the OCS	32Gbps
Link speed of upstream links in the EPS	64Gbps
Link speed of downstream links in the EPS	10Gbps
Propagation delay in the OCS	80ns
One link propagation delay in the EPS	20ns
Switching delay in packet switches	420ns
MTU in the EPS	4096B

which is defined as a duration between two consecutive MPI events. For MPI events, we record types of events, message sizes, source or destination process and so on. For CPU events, processing times are recorded.

In the second step, we predict execution times of applications using our home-grown network simulator, which takes network parameters and trace files as inputs. The hybrid network architecture in our simulation is shown in Figure 4. The system contains 128 compute nodes each of which has NICs for both networks, a simplest case of our hybrid network. The detailed system parameters are listed in Table 1. We let CPU speed of nodes be 1.6GHz, which is determined by that of nodes used for obtaining traces of applications. The OCS network is rearrangeable and non-blocking. We assume bandwidth of single OCS pathway is sufficient and only limited by bus speed of nodes, which is set to be 32Gbps of PCI Express x16. We ignore switching delay in the OCS, because once a pathway is assigned, we can expect that there are no delays in switches. We assume that EPS switches conduct store-and-forward communication. Bandwidths of upstream vs. downstream links are set so that its ratio would be 1 to 5. The switching delay and MTU size of the EPS come from specification of InfiniBand switches used in TSUBAME. And we let propagation delay in both network be calculated by light transmission speed.

Based on above network parameters, our network simulator replays events in the trace files as follows. When a CPU event is found, the simulator simply increments its virtual clock by the processing time of the event. For MPI events, the simulator increments its virtual clock by time spent in communication, obtained as follows. Communication time of a message is the sum of times spent in all links and switches where the message passes through. The communication time on a link is calculated as $\alpha + n/\beta$, where α is propagation delay, β is bandwidth and n is message size. Furthermore, we take contention on each link into account. For this purpose, we store time stamps of past messages and postpone successive ones if they are conflicted. When a process forwards a user message from one link to the other link, it sends the message after complete reception. This time, we ignored overhead on processors incurred by forwarding messages for ease. For collective communication events, we simulate the same algorithms as in MPICH2 [15]. When we are going to evaluate SP scheme on our network or EPS-only networks, ranks of processes are assigned sequentially from left to right as in Figure 4. With the CP scheme, process allocation is determined by METIS [1], a graph partitioning library which uses edge cut algorithm. We set the threshold of message sizes described in Section 3 to be 8192 bytes.

4.2. Result and Discussion

The result of Neighbor is shown in Figure 5. The x-axis means the number of OCS pathways, and the y-axis means the relative execution time against the case in the EPS-only network with 64Gbps upstream links. CP U $\times$ n and SP U $\times$ n correspond to each scheme on the hybrid network with n $\times$ 64Gbps upstream links. EPS U $\times$ n corresponds to cases in EPS-only networks, where bandwidth of upstream links is set as n $\times$ 64Gbps, $n = 5$ achieves full bisection. $n = 1/4$ is used for simply simulating a contention on the EPS upstream links and in this case we assume the bandwidth remainder, $3/4 \times 64Gbps = 48Gbps$, is used for other communication such as network storage and other applications. It can be seen that our schemes maintain performance even when EPS upstream links are narrow. This is because end-tier processes forward messages and bottleneck links are never used. Furthermore, performance of full bisection case is achieved with about 20 pathways. For CP, processes are partitioned into four 8x4 matrices, so that inter-switch communication between each switch are only 8 and in total 24. This is because only 24 pathways are used. On the other hand, SP partitioned processes into four 2x16 matrices, and because there are 16 inter-switch communication between each switch and in total 48, 48 pathways are used. There is a big performance difference when there are fewer pathways than six. The reason of this is that the relay nodes must forward many messages and their EPS links are congested, because SP has more inter-switch communication

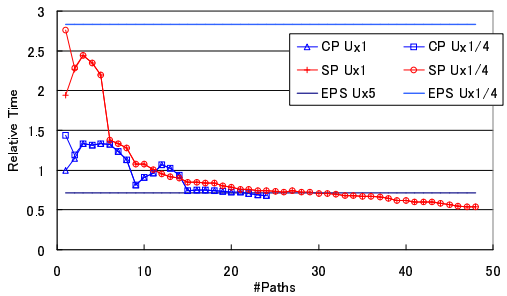

Figure 5. Relative performance of Neighbor

than CP.

Figure 6 shows loads on topmost switch (S_0) and nodes.The load on S_0 switch is defined as the total amount of messages it receives from all its links. The load on node is calculated as maximum total amount of messages that each node sends/receives to/from its EPS link. We can see loads of S_0 switch drastically decreases when we have more pathways, and no load when number of pathways is bigger than three. This is because all four switches are connected by three pathways in a chain, then all messages pass through the OCS network and the EPS network is never used. Hence, contention in the EPS network would be avoided. On the other hand, we see the load of node gets larger than the EPS-only case. This is because partial nodes have to forward messages and their EPS links are heavily loaded. However, this situation is mitigated when we have plenty of pathways, since the chances for direct communication increase.

Next, from CG in Figure 7, we see that our schemes also maintain performance even when EPS upstream links are narrow and have better performance than full bisection case. When the number of pathways is less than six, our schemes have large overhead. This behavior is explained by the fact that there are communication between all the pairs of end-tier switches as shown in Table 2. Items in the table mean the number of node pairs that have inter-switch communication; we see all items are positive in the case of CG. Thus when we have only less than six ($=_4 C_2$) pathways, we suffer from message forwarding of excessive number of hops. As the number of pathways increases, performances are improved because we can reduce the average number of hops. We see no improvement with more than 48 pathways for the following reason. In Table 2, there are only eight node pairs having communication between two end-tier switches; thus 48 pathways are sufficient to cover all inter-switch communication. We can also see from Figure 7 that the full bisection EPS network can not improve performance, because CG only has 48 inter-switch communication against its 8192 ($=_{128} C_2$) process pairs and then upstream links can not be filled and utilized enough.

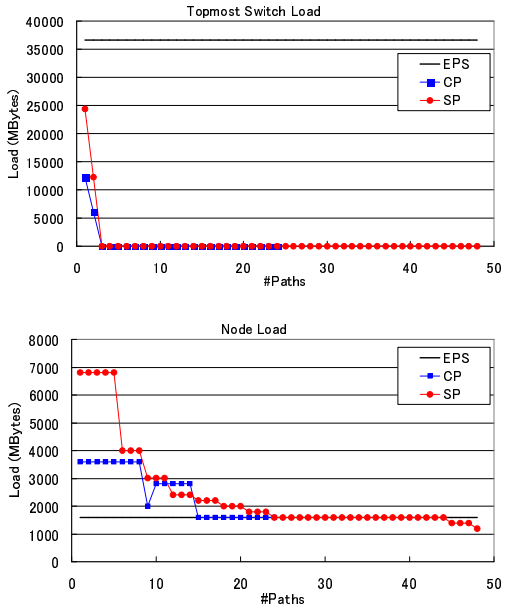

Figure 6. Load in Neighbor

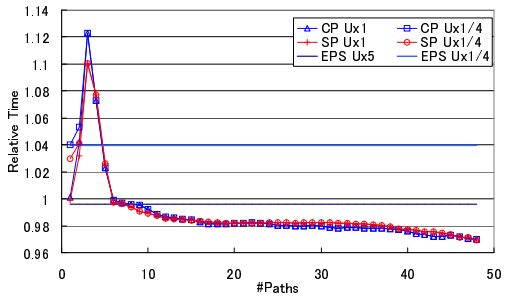

Figure 7. Relative performance of CG

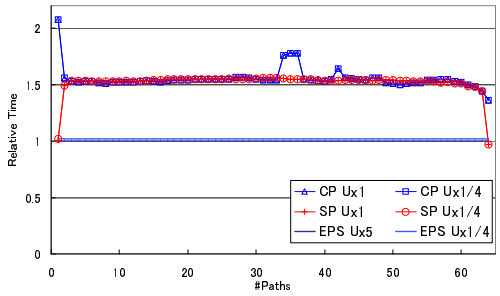

Figure 8. Relative performance of MG

Finally, from MG in Figure 8, we see execution times on our schemes are 50% to 110% longer than on EPS-only cases, whose values are almost one. One of the reasons is that there are more node pairs that give rise to inter-switch

Table 2. Number of inter-switch communication in CG

Switch ID	S_1	S_2	S_3	S_4
S_1	132	8	8	8
S_2	8	132	8	8
S_3	8	8	132	8
S_4	8	8	8	132

Table 3. Number of inter-switch communication in MG with CP (and SP)

Switch ID	S_1	S_2	S_3	S_4
S_1	144 (160)	16 (16)	0 (0)	32 (16)
S_2	16 (16)	144 (160)	32 (16)	0 (0)
S_3	0 (0)	32 (16)	144 (160)	16 (16)
S_4	32 (16)	0 (0)	16 (16)	144 (160)

communication in MG. The number is 64 (4×16, from Table 3) in SP and 96 ($2 \times 16 + 2 \times 32$, from Table 3 inside parenthesis) in CP. Another reason is that MG can not utilize bandwidth of pathways because the amount of single pair transfer is smaller than in CG. The amount is about 740MBytes in CG, while it is 44MBytes in MG. Generally, smaller messages suffer largely from increase of propagation delay by forwarding. Moreover, in CP, number of pairs with inter-switch communication is increased. This comes from the edge cut algorithm, and we are considering using another one which can localize communication.

The loads of S_0 switch and nodes in CG and MG have the same trends with Neighbor, except that small count of small messages are transferred on the EPS network.

5. Related Work

Interconnects that utilize both the EPS network and the OCS network are proposed [3, 11, 12]. A system where each node has a NIC to low bandwidth EPS network and multiple NICs to OCS networks is proposed [3]. In their work, an application on the system uses just the EPS network for collective communication, and for point-to-point communication it first uses EPS network while the system monitors the traffic, and an OCS pathway is assigned and the traffic is migrated to the circuit when a certain threshold is exceeded. If a node uses up all its optical NICs, it releases the least used circuits to accommodate a new one in a LRU fashion. They also proposed a hybrid OCS network consists of k planes OCS network, l planes EPS network and SMP nodes each of which has $k + l$ links, and compare its performance with traditional networks such as fat-tree and mesh by using application performance models [4]. Although their approaches are effective, the proposed systems would incur substantial cost and installation space because it needs multiple OCS networks. Another interconnect called HFAST, where each node is connected to a single low bandwidth EPS network and a network which is a hybrid of EPS and OCS network, is proposed [11]. The HFAST puts an OCS network between compute nodes and an EPS network, and optical circuits are assigned in a manner such that localities in MPI application communication pattern are satisfied. However, the network incurs high cost and efficiency of the EPS network usage is low, because large sized messages are exchanged exclusively on HFAST and the EPS network is used only for small size messages. The E-RAPID network which uses EPS networks for intra-board (node group) communication and OCS networks for inter-board communication is proposed [12]. It can adjust the bit rate to reduce power consumption and assign circuits between boards where traffic is high. However, it lacks system scalability because all boards must have optical transmitters matching the number of boards.

Compared to the above works, we attempt to achieve compromise between performance, cost, and scalability by targeting just the problem of attaining high bisection bandwidth, using OCS in a more supplemental role, using the software approach for making effective assignments. As another software approach, an MPI collective communication library which tests performances of various algorithms and selects the best one for use is proposed [9]. However, it incurs profiling of many collective algorithms repeated number of times for effective selection, making it feasible only for iterative applications. There is similar work [10] targeting 3D torus of Blue Gene/L using the simulated annealing to allocate MPI processes to localize communication. Such works are similar but different from ours because of the divergent characteristics of EPS versus OCS networks resulting different solutions.

6. Conclusion

We proposed a hybrid network which combines a low bandwidth, low latency EPS network and a high bandwidth, high latency OCS network in a cost-effective fashion, both networks being commodity in nature for low cost and low power consumption. All compute nodes in this network have one NIC for EPS network while some compute nodes also have a connection to the OCS network, using the latter as a high-bandwidth, cross-switch shortcut to avoid contentions in upstream links in the EPS network. We also proposed a locality aware MPI communication algorithm on the network, where processes are grouped, OCS pathways are assigned by their initial location or their communication pattern and messages are forwarded by processes connected

to both networks. Benchmark results demonstrate that our scheme is very effective when application has high locality and high traffic. However, some applications could not exploit the added bandwidth. In all cases, the load of the topmost EPS switch was very low, indicating high scalability and better results with orders of magnitude increase in the number of nodes in future petascale systems and beyond.

For future work, we will test our hybrid network with such a condition that only some nodes have access to the OCS network We also plan to verify its scalability with large number of processes and many other applications. In case of testing with large system, we would have to rely on modeling method such as used in [4] instead of simulating. Moreover, we will propose schemes for multi-application scenarios where conflicts in OCS pathway will result in allocation failures, compromising performance. In such a case, mutual re-planning spanning the applications would likely be ideal. We also hope to test our scheme on a real hardware testbed.

Acknowledgment

This work is supported in part by the Ministry of Education, Culture, Sports, Science, and Technology, Grant-in-Aid for Scientific Research on Priority Areas, 18049028 and JSPS Global COE program "Computationism as a Foundation for the Sciences".

References

[1] METIS - Family of Multilevel Partitioning Algorithms. `http://glaros.dtc.umn.edu/gkhome/views/metis/`.

[2] TACC HPC Systems. `http://www.tacc.utexas.edu/resources/hpcsystems/`.

[3] K. J. Barker, A. Benner, R. Hoare, A. Hoisie, A. K. Jones, D. J. Kerbyson, D. Li, R. Melhem, R. Rajamony, E. Schenfeld, S. Shao, C. Stunkel, and P. Walker. On the Feasibility of Optical Circuit Switching for High Performance Computing Systems. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, November 2005.

[4] K. J. Barker and D. J. Kerbyson. Performance Analysis of an Optical Circuit Switched Network for Peta-Scale Systems. In *Euro-Par 2007*, pages 858–867, August 2007.

[5] K. Davis, A. Hoisie, G. Johnson, D. J. Kerbyson, M. Lang, S. Pakin, and F. Petrini. A Performance and Scalability Analysis of the BlueGene/L Architecture. In *Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, 2004.

[6] R. F. V. der Wijngaart. NAS Parallel Benchmarks Version 2.4. Technical Report NAS Technical Report NAS-02-007, NASA Ames Research Center, 2002.

[7] P. D. Dobbelaere, K. Falta, L. Fan, S. Gloeckner, and S. Patra. Digital MEMS for Optical Switching. *Communications Magazine, IEEE*, 40:88–95, March 2002.

[8] A. Faraj and X. Yuan. Communication Characteristics in the NAS Parallel Benchmarks. In *International Conference on Parallel and Distributed Computing and Systems*, pages 729–734, November 4-6 2002.

[9] A. Faraj, X. Yuan, and D. Lowenthal. STAR-MPI: self tuned adaptive routines for MPI collective operations. In *Proceedings of the 20th annual international conference on Supercomputing*, pages 199–208, 2006.

[10] G. Ghanot, A. Gara, P. Heidelberger, E. Lawless, J. C. Sexton, and R. Walkup. Optimizing task layout on the Blue Gene/L supercomputer. In *IBM Journal of Research and Development*, volume 49, pages 489–500, 2005.

[11] S. Kamil, A. Pinar, D. Gunter, M. Lijewski, L. Oliker, and J. Shalf. Reconfigurable Hybrid Interconnection for Static and Dynamic Scientific Applications. In *ACM International Conference on Computing Frontiers*, 2007.

[12] A. K. Kodi and A. Louri. Power-Aware Bandwidth-Reconfigurable Optical Interconnets for High-Performance Computing Systems. In *International Conference on Parallel and Distributed Processing Symposium*, March 2007.

[13] S. Matsuoka. The Road to TSUBAME and Beyond. *Petascale Computing: Algorithms and Applications*, 2007.

[14] S. Shao, A. K. Jones, and R. Melhem. A Compiler-based Communication Analysis Approach for Multiprocessor Systems. In *Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium*, April 2006.

[15] R. Thakur, R. Rabenseifner, and W. Gropp. Optimization of Collective Communication Operations in MPICH. *International Journal of High Performance Computer Applications*, 19(1):49–66, 2005.