Performance Analysis of an Optical Circuit Switched Network for Peta-Scale Systems

Kevin J. Barker and Darren J. Kerbyson

Performance and Architecture Laboratory (PAL)
Los Alamos National Laboratory, NM 87545
{kjbarker,djk}@lanl.gov

Abstract. Optical Circuit Switching (OCS) is a promising technology for future large-scale high performance computing networks. It currently widely used in telecommunication networks and offers all-optical data paths between nodes in a system. Traffic passing through these paths is subject only to the propagation delay through optical fibers and optical/electrical conversions on the sending and receiving ends. High communication bandwidths within these paths are possible when using multiple wavelengths multiplexed over the same fiber. The set-up time of an OCS circuit is non-negligible but can be amortized over the lifetime of communications between nodes or by the use of multi-hop routing mechanisms. In this work, we compare the expected performance of an OCS network to more traditional networks including meshes and fat-trees. The comparison considers several current large-scale applications. We show that the performance of an OCS network is comparable to the best of the network types examined.

1 Introduction

Recent large-scale procurements indicate that a multi-petaflop system containing tens to hundreds of thousands of processors (or CPU cores) will be built by the end of this decade. The performance of such a system is directly related to the applicability and performance of the interconnection network. An Optical Circuit Switched (OCS) network has recently been proposed that is aimed at solving the low latency and high bandwidth requirements of these systems [1].

Even when considering four or eight cores per processing chip (or socket), the chip-count in such a system will be in the tens of thousands. Given the current trend of increasing chip density and advances in local (near-distance) interconnection technology, we also expect to see an increase in the number of sockets within a compute node. Even with 64 sockets per node, the number of compute nodes will likely be in the thousands. Therefore, the design and implementation of the inter-node communication network is critical in determining both the performance and cost of large HPC systems.

In this work we compare the expected performance of OCS to more traditional networks including meshes and fat-trees. This analysis uses performance models for large-scale applications that have been validated on current large-scale systems.

Traditionally large-scale system designers have had two choices for the high performance network: either topological inflexibility or topological flexibility.

A.-M. Kermarrec, L. Bougé, and T. Priol (Eds.): Euro-Par 2007, LNCS 4641, pp. 858–867, 2007.

Topologically inflexible: networks such as tori and meshes, have the advantage of a cost that scales linearly with machine size. Examples include the IBM BlueGene/L network (current largest is a 32x32x64 3D torus) and the Cray Red Storm network (current largest is a 27x20x24 3D torus/mesh hybrid). A drawback with such networks is that application performance can degrade if the logical communication topology does not map well to the physical network.

Topologically flexible: networks such as multistage Fat-tree networks include Quadrics [2], Myrinet [3], and Infiniband [4]. Such networks, when using adaptive routing, can reduce network contention and enable high performance for a variety of applications that utilize a large range of logical communication topologies. However, cost scales super-linearly with machine size ($N/X \cdot log_x(N)$, switches are required for an X-radix tree containing N nodes) and can become prohibitive in very large systems. In addition, the development cost of a new generation of switches and network adapters can add substantial cost to the system as communication protocols, signaling speeds, and switch radixes scale to keep pace with newer and faster processor chips.

The OCS has the potential to give the best of both worlds. The OCS uses a hybrid network consisting of both Electrical Packet Switching (EPS) and OCS planes constructed using Micro Electro-Mechanical Systems (MEMS) based switches. Although the switching latency associated with MEMS switches is non-negligible, the feasibility of such a hybrid approach was demonstrated for applications with static or slowly changing communication patterns [1].

Like the proposed OCS network analyzed in this work, the recently proposed HFAST (Hybrid Flexibly Assignable Switch Topology) network [5] also makes use of MEMS-based optical circuit switches. The Gemini and Clint projects [6,7] use two types of networks to handle communication traffic in a similar way to OCS. Gemini uses both OCS and Electronic Packet Switched (EPS) switches, while Clint's circuit and packet switched networks are both electronic. In distributed computing where nodes communicate over relatively large distances, there have been several proposed circuit switched networks. Cheetah [8] is aimed at transferring large files between storage centers, while the OptIPuter [9] is aimed at linking multiple resources including visualization and storage.

1.1 Contributions of This Work

A quantitative analysis is presented of the expected performance of a large-scale system that utilizes an OCS network compared to more conventional network architectures. We show that, for a number of applications, the OCS network architecture has excellent performance in both capacity and capability computing modes as well as for weak- and strong-scaling application modes. While the performance of the OCS network is on par with the best traditional network for each application examined, the key to the OCS approach lies in the dynamic allocation of bandwidth to where it is needed by the application, yielding a much more flexible network architecture. Such a broad analysis has not been done before; previous work was constrained to a single application on a single system configuration and focused on the feasibility of the hybrid OCS approach, not on its expected performance [1]. Results from this work indicate the hybrid OCS network architecture is a viable

contender for future HPC systems. We do not quantify the cost of the OCS network but expect this to be lower than existing networks since OCS can utilize high-radix MEMS switches that are commonly used in the telecommunications industry.

2 The OCS and Traditional Networks

The Optical Circuit Switching network consists of a hybrid architecture containing both EPS (Electronic Packet Switched) planes and OCS planes [1]. OCS planes handle the higher bandwidth traffic associated with larger messages and communication among persistent partner processors. The EPS planes handle lower bandwidth and collective communications, as well as communication between infrequently communicating processor pairs. Figure 1 illustrates this hybrid architecture consisting of L EPS planes and K OCS planes (where typically $K >> L$).

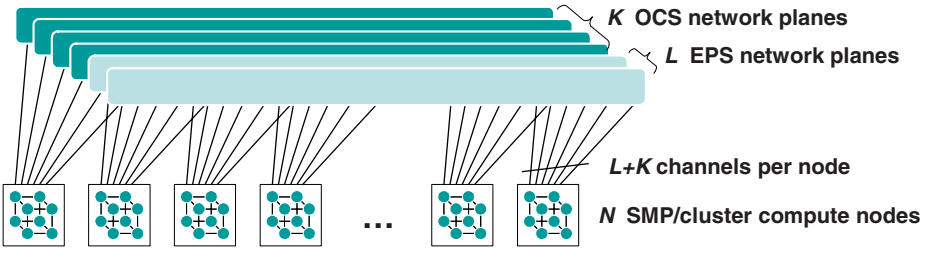

Fig. 1. The OCS network architecture illustrating multiple planes of two network types

A key characteristic of the OCS network is its ability to dynamically allocate network bandwidth between nodes as required by an application. The hybrid OCS is able to adaptively alter the set of partner processors directly connected to a particular node, either at job launch (if a job's static communication requirements are known) or during runtime. A further benefit is the ability to effectively partition a single system among several concurrently executing jobs (as is necessary in a capacity system), eliminating interference from communication traffic originating in independent jobs.

An OCS plane can be implemented using a single MEMS switch. This, a high radix crossbar (1024 ports currently possible), in which any input can be optically connected to any output. However, to switch an input from one output to another requires the mechanical movement of a micro-mirror which can take several milliseconds and significantly impact performance when processors in one node need to communicate with processors in more than one other node. To reduce the switching frequency, the following mitigations are possible:

***Multiple OCS planes*:** Each node is able to communicate with a single partner node per OCS plane without switching. If the number of OCS planes is greater than the application's communication degree (the maximum number of communication partners per node) and the application's communication pattern is static, the network can be configured without the need for further switching. From previous analysis we have seen that even when an application's communication pattern is

dynamic, it usually persists for some time before changing and hence the switching of OCS planes will not significantly impact performance [1].

***Multi-hop routing*:** The OCS network can use multi-hop routing to guarantee a path between any two nodes by routing messages through intermediate nodes. This increases the effective bandwidth between nodes (due to multiple routes between any source and target) but at the expense of increased latency as well as increased traffic on inter- and intra-node communication paths.

In order to quantitatively evaluate the performance of the OCS hybrid network, we compare it against several typical high-performance networks. These are summarized in Table 1 and include 2D and 3D meshes, Fat-Trees (FT), and fully-connected networks (FC1 & FC2). The OCS networks studied include fully connected (OCS-FC1 & OCS-FC2), as well as a dynamic OCS configuration where bandwidth is dynamically allocated to coincide as closely as possible with application communication requirements (OCS-D). While both OCS and fully-connected networks can be operated in single-hop or two-hop mode, we will present results for the fully-connected network in single-hop mode only; in two-hop mode both traditional FC2 and OCS-FC2 exhibit nearly identical performance at full system size.

Note that all OCS network topologies can be provided by the same networking hardware. The difference between fully-connected OCS and electrical networks is that the OCS network provides full connectivity for the nodes participating in a single *job*, while a standard network fully connects the entire *machine* regardless of job size.

In all cases, the latency per hop is assumed to be 50ns and the bandwidth per network link is 4GB/s. In the case of the mesh networks, neighboring nodes are assumed to be physically close (at a cost of 1-hop in each direction), while the fully-connected network will exhibit a worst-case latency which includes transit time between two nodes that are physically farthest apart (incorporating a speed-of-light, SOL, component). Similarly, all OCS network topologies require a transit time to the single OCS switch rack (assumed to be in the middle of the machine layout) and back – effectively the same distance as the fully-connected network.

We assume the switch residency time in the fat-tree network is 50ns, the mesh latency per hop is 20ns, and signal propagation time is 5.2ns per meter. It is also

Table 1. Network characteristics used in this analysis, assuming a 64 quad-core sockets and 256 inter-node network links per node

	Network	Latency per hop	Links per neighbor	Notes
2D	2D Mesh	50ns	64	
3D	3D Mesh	50ns	42	
FT	Fat-tree	50ns	-	24-ary fat-tree, 256 planes
FC1 FC2	Fully-Connect, 1-hop or 2-hop	50ns+SOL	1	Static fully-connected network – 1 link between all node-pairs.
OCS-FC1 OCS-FC2	OCS (fully-connect) 1-hop or 2-hop	50ns+SOL	256/JobSize	OCS as a fully-connected network based on job-size
OCS-D	OCS (dynamic)	50ns+SOL	as needed	Application determines connectivity

assumed that a node, which contains 64 quad-core sockets each with 4 inter-node communication links, fits into a single rack and racks are spaced 2m apart in a 2D floor layout. The fat-tree is assumed to use 48-port switches (a 24-ary tree), and the MPI software overhead is 500ns on both the sender and receiver sides.

We assume that messages can be striped across available links. Messages greater than 16 KB are striped across all links within a node, while messages between 2 KB and 16 KB are striped across links available from a single socket. Messages smaller than 2 KB are not striped.

3 Performance Analysis Methodology

Rather than simply considering network latency and bandwidth characteristics, we compare the expected performance of large-scale applications executing on a parallel machine equipped with the networks whose configurations are described in Section 2. Applications considered include a generic one representing a 2D or 3D partitioning of a 3D regular data grid as well as several large-scale scientific codes. Strong and weak scaling modes are used in the generic case while each application is considered in its most appropriate mode of execution.

We utilize detailed performance models for all of the applications. This approach has been used extensively in the past; models for the applications have been published and validated on current systems including 64K nodes of the IBM Blue Gene/L system installed at Lawrence Livermore National Laboratory and 10K nodes of the Cray ASC Red Storm machine at Sandia National Laboratory [10]. On these machines prediction error was less than 10% [11,12,13,14,15]. The use of performance models enables the determination of potential application performance in advance of system implementation.

Our performance models were constructed from a detailed examination of the application both in terms of its static characteristics (as defined by the source code) as well as its dynamic characteristics (the parts of the code that are actually used by an input deck). Through the use of profiling, a structural model is determined which describes the functional "flow" of the application. The structural model includes characteristics such as communication type (e.g., MPI_Send, MPI_Isend, etc.), frequency, and size, as well as computation characteristics such as number of data cells processed on each processor. These characteristics are typically dependent on the particular input deck and the size of the parallel system.

The structural model does not include information related to time, such as computation rate or message latency and bandwidth. Rather, the structural model is combined with hardware performance characteristics, such as message latency, bandwidth, and computation rate obtained from benchmarks or from system specs when the system cannot be benchmarked. System architecture characteristics including network topology are also used. It should be noted that we do not model single processor performance from first principles. Rather we relay on measuring the single processor performance (or using cycle-accurate simulation for a future system) for each of the applications, and concentrate on modeling its parallel behavior.

Once constructed, the model is used to predict performance of current (measurable) systems and hence validated. This is an iterative process which stops when prediction accuracy reaches desired levels – our goal has been to have an error of less than 20%

in this process but we have found that typically the error is considerably less than this. Given a validated model, performance on a non-existent future system can be predicted with some confidence.

4 Performance Analysis – Generic Boundary Exchange

We begin with an analysis of a generic application model that uses a regular data partitioning and boundary exchange communication. This is followed by an analysis of large-scale scientific application performance in Section 5. The generic application partitions a 3D data grid in either two or three dimensions (referred to here as Generic-2D and Generic-3D). Note that in all the following analyses a node is assumed to contain 64 sockets arranged internally in a 6D hypercube.

Performance predictions for a boundary exchange are shown in Figure 2 for strong scaling (for a fixed global grid size of 10^9 cells), while Figure 3 shows predictions for weak scaling (for a fixed problem per processor of 10^7 cells). A single word per boundary cell is communicated in the relative directions. The communication times generally decrease with node count in the strong-scaling case since the boundary surfaces, and hence communication volume decreases. In the weak scaling case, the boundary surface sizes are fixed, resulting in near constant communication times for 2^d or more nodes for a d-dimensional partitioning.

In all cases, fully connected networks with single-hop routing are the worst performers due to lack of available bandwidth between any pair of nodes. The OCS fully-connected network exhibits linearly decreasing bandwidth between node pairs as job size grows; at maximum job size (utilizing the full machine of 256 nodes, 16384 sockets) both fully connected single-hop networks offer the same performance. At smaller configurations, however, the OCS proves to be superior.

The regular communication topologies of the Generic-2D and Generic-3D applications map directly to the 2D and 3D mesh networks, respectively, leading to near optimal performance. However, in cases in which the physical mesh network does not exactly match the application's communication topology, the incurred latency and reduction in bandwidth resulting from multiple hops through the network negatively impact performance, particularly at large scale. In addition, because only a fraction of the inter-node links connect each pair of neighboring nodes, the peak inter-node bandwidth is not as great as with the multi-hop or fat-tree networks. The reconfigurable OCS network is able to match the communication topology of either Generic application, yielding the maximum possible node-to-node bandwidth. However, each packet suffers an SOL latency penalty relative to the mesh networks imposed by having only a single, centrally-located switch component.

In short, if the benefit of reduced latency outweighs the penalty of reduced bandwidth, the 2D or 3D mesh will likely obtain the best performance followed closely by the OCS and Fat-Tree networks (assuming the application's communication topology maps relatively well to the physical machine). The OCS's flexibility puts its performance on par with Fat-Tree networks and far ahead of mesh networks in those cases in which the application's logical and the network's physical topologies do not match. In addition, the OCS network's reduced number of components is attractive in terms of cost and reliability, giving it superior price/performance characteristics.

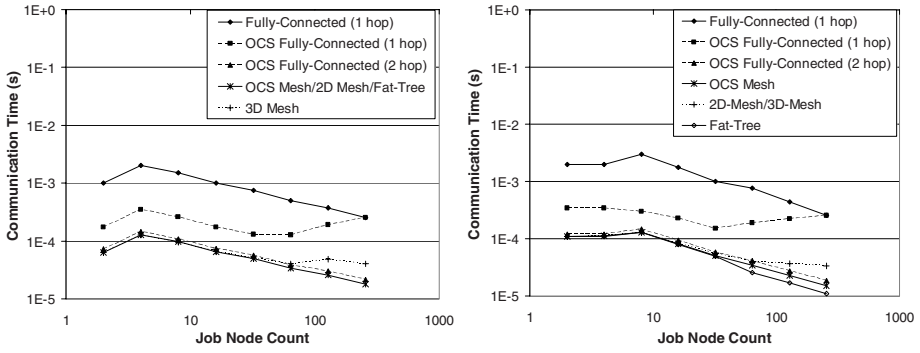

Fig. 2. Boundary exchange times for Generic-2D (left) and Generic-3D (right), strong scaling

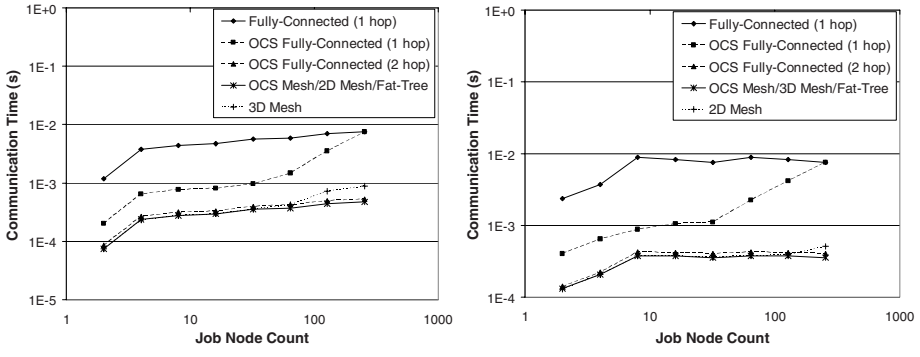

Fig. 3. Boundary exchange times for Generic-2D (left) and Generic-3D (right), weak scaling

5 Application Performance Analysis

Performance models of the applications listed in Table 2 were used to compare the performance of the networks listed in Table 1. We consider two system scenarios here: a 512-socket job executing on a larger capacity system and a 16384-socket capability system in which a single job utilizes the full machine. If we assume each node contains 64 sockets and a socket size of four cores with each core capable of 16 GF/s, then the capability mode provides a peak performance of 1.05 PF/s, while in capacity mode the peak performance available to each job is 32 TF/s.

We calculate the runtime for each application and compare it to the best runtime across networks (Figure 4). The runtime includes both communication and computation performance, allowing us to examine how network performance contributes to overall application performance. Figure 4 shows the relative performance with a value of one indicating the network with the best runtime, and a value greater than one indicating a longer runtime than the best by that factor.

Table 2. Details of the applications used in this analysis

Application	Scaling	Main Communication	Comment and Input
Generic-2D	S or W	2-D regular	2-D partitioning of 3-D data 10^9 cells (s), 10^7 (w)
Generic-3D	S or W	3-D regular	3-D partitioning of 3-D data 10^9 cells (s), 10^7 (w)
HYCOM	Strong	2-D (mod), Row/Col	Hybrid Ocean Model [11]. 1/12 degree (4500x3398x26)
KRAK	Strong	2-D Irregular	Hydrodynamics [12] 4 material cylinder (204,800 cells)
LBMHD	Weak	2-D regular	Magneto-hydrodynamics, 128x128 cells
POP	Strong	2-D partitioning	Parallel Ocean Program [15] 0.1 degree (3600x2400x40 cells)
RF-CTH2	Strong	3-D partitioning	Shock-dynamics (833x191x191 cells)
SAGE	Weak	modified 1-D, reduction	Shock-wave hydrodynamics [14] (100,000 cells)
Sweep3D	Weak	2-D regular with pipeline	Deterministic S_N transport [13] (8x8x1200 cells)

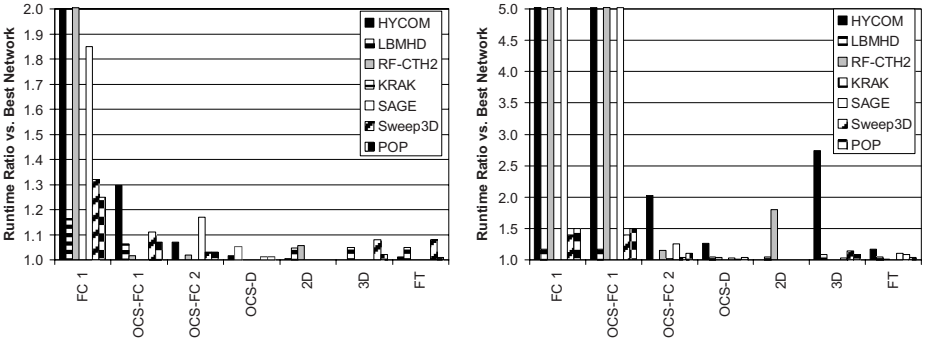

Fig. 4. Application performance on a capacity sized job of 512 sockets (left) and a capability sized job of 16384 sockets (right). In both instances, system size is 16384 sockets.

It can be seen in Figure 4 (left) that there is very little difference in application runtimes on systems equipped with OCS-Dynamic, Fat-Tree, and 2D and 3D mesh networks. Note that the fully-connected network with single-hop routing is a factor of 3.4 and 4.9 times slower than the best network for HYCOM and RF-CTH2 respectively – this is due to the larger bandwidth requirements of these codes.

The situation changes slightly when considering a capability system in Figure 4 (right). Here, the mesh networks start to perform poorly on RF-CTH2 (a 3D code which does not map well to a 2D mesh) and HYCOM (a 2D code which does not map nicely to a 3D mesh). The performance of the OCS-FC1 network and the FC1 network is almost identical at the largest scale. This is expected since these two networks exhibit essentially the same topology at this scale (in a capacity system, the OCS network provides full connectivity for the job only, not the entire system). Note

that the ratio of runtimes for HYCOM, RF-CTH2, and SAGE on these network topologies is 6.5, 10.2, and 12.6 respectively. Again the OCS-FC2 and OCS-Dynamic networks result in a performance close to the best performing network in all cases.

A summary of the relative runtimes is included in Table 3. Here, the average runtime ratio is shown across all applications for each network. It can be seen that the OCS-Dynamic network provides the best average runtime, although the average performance of the OCS-Dynamic and Fat-Tree networks are similar. However, recall that we have used an idealized fat-tree network with an assumed minimal congestion (resulting from ideal adaptive routing) within the network; therefore the Fat-Tree result is optimistic. Note that we have not considered network cost in this analysis, but expect that an OCS network will be cheaper and hence a lower price/performance.

Table 3. Average runtime ratio (to the best network) for each network

	FC1	OCS-FC1	OCS-FC2	OCS-D	2D Mesh	3D Mesh	Fat-Tree
Capacity	2.13	1.08	1.05	1.01	1.02	1.02	1.02
Capability	4.87	4.87	1.23	1.06	1.12	1.30	1.06

We do not describe the case of a fully-connected network with two-hop routing (FC2). We would expect the performance of this network to match that of the OCS-FC2 for a job equal to the system size (capability mode). However, when multiple jobs or jobs utilizing less than the full system are executed, it can be expected that the FC2 network will provide inferior performance due to contention between jobs that may arise within intermediate nodes during message routing.

6 Conclusion

We have discussed potential benefits of Optical Circuit Switch (OCS) based networks over traditional direct and indirect networks for large-scale parallel computing systems. Although the number of available OCS planes is limited and optical circuit set-up cost is non-negligible, these factors can be mitigated through architectural decisions such as the number of OCS switch planes and multi-hop routing strategies.

Through the use of detailed and previously validated application performance models, we have been able to study the potential performance of an OCS network relative to several common high performance network types, including mesh networks (2D and 3D), fat-tree networks, and variants of fully connected networks. Such a broad analysis of the potential performance of an OCS network in the realm of high performance computing has not been previously done. The results indicate that the performance of an OCS network should be comparable to the traditional network type that is currently best suited to each of the applications. This results from the flexibility of the OCS design, allowing it to effectively mimic the connectivity of more common direct and indirect network topologies. The true advantage of the OCS network may come from also considering its cost – though has not been quantified.

Acknowledgements. This work was made possible by OCS network concepts initially published in [1] and driven by Eugen Schenfeld of IBM T.J. Watson Research Center. The authors would like to thank Eugen for his enthusiasm and support. This work was funded in part by the DOE Accelerated Strategic Computing (ASC), and the DARPA High Productivity Computing Systems (HPCS) programs. Los Alamos National Laboratory is operated by Los Alamos National Security LLC for the US Department of Energy under contract DE-AC52-06NA25396.

References

1. Barker, K.J., Benner, A., Hoare, R., Hoisie, A., Jones, A.K., Kerbyson, D.J., Li, D., Melhem, R., Rajamony, R., Schenfeld, E., Shao, S., Stunkel, C., Walker, P.: On the Feasibility of Optical Circuit Switching for High Performance Computing Systems. In: Proc. Supercomputing, Seattle (2005)
2. Petrini, F., Feng, W., Hoisie, A., Coll, S., Fractenberg, E.: The Quadrics Network: High-Performance Clustering Technology. IEEE Micro. 22(1), 46–57 (2002)
3. Myricom, http://www.myri.com
4. Infiniband Trade Association, http://www.infinibandta.org/
5. Shalf, J., Kamil, S., Oliker, L., Skinner, D.: Analyzing Ultra-Scale Application Communication Requirements for a Reconfigurable Hybrid Interconnect. In: Proc. Supercomputing, Seattle (2005)
6. Chamberlain, R., Franklin, M., Baw, C.S.: Gemini: An Optical Interconnection Network for Parallel Processing. IEEE Trans. on Parallel and Distributed Processing 13(10), 1038–1055 (2002)
7. Eberle, H., Nilsm Gura, N.: Separated High-bandwidth and Low-latency Communication in the Cluster Interconnect Clint. In: Proc. Supercomputing, Baltimore (2002)
8. Veeraraghavan, M., Zhenga, X., Leeb, H., Gardnerc, M., Fengc, M.: CHEETAH: Circuit-Switched High-Speed End-to-End Transport Architecture. In: SPIE Proc. vol. 5285, pp. 214–225 (2003)
9. Defanti, T., Brown, M., Leigh, J., Yu, O., He, E., Mambretti, J., Lillethun, D., Weinberger, J.: Optical switching middleware for the OptIPuter. IEICE Trans. Commun. E86-B(8), 2263–2272 (2003)
10. Hoisie, A., Johnson, G., Kerbyson, D.J., Lang, M., Pakin, S.: A Performance Comparison Through Benchmarking and Modeling of Three Supercomputers: Blue Gene/L, Read Storm and ASC Purple. In: Proc. SuperComputing, Tampa FL (2006)
11. Barker, K.J., Kerbyson, D.J.: A Performance Model and Scalability Analysis of the HYCOM Ocean Simulation Application. In: Proc. IASTED Int. Conf. on Parallel and Distributed Computing, Las Vegas NV (2005)
12. Berker, K.J., Pakin, S., Kerbyson, D.J.: A Performance Model of the KRAK Hydrodynamics Application. In: Proc. Int. Conf. on Parallel Processing, Columbus OH (2006)
13. Hoisie, A., Lubeck, O., Wasserman, H.J.: Performance and Scalability Analysis of Teraflop-Scale Parallel Architectures using Multidimensional Wavefront Applications. Int. J. of High Performance Computing Applications 14(4), 330–346 (2000)
14. Kerbyson, D.J., Alme, H.J., Hoisie, A., Petrini, F., Wasserman, H.J., Gittings, M.L.: Predictive Performance and Scalability Modeling of a Large-scale Application. In: Proc. Supercomputing, Denver CO (2001)
15. Kerbyson, D.J., Jones, P.W.: A Performance Model of the Parallel Ocean Program. Int. J. of High Performance Computing Applications 19(13) (2005)