COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY **Special Issue 2006**, 33-45

A scalability Study of Columbia using the NAS Parallel Benchmarks

Subhash Saini, Johnny Chang, Robert Hood and Haoqiang Jin

NASA Advanced Supercomputing Division
NASA Ames Research Center
Moffett Field, California 94035-1000, USA
e-mail: {ssaini/jchang/rhood/hjin}@mail.arc.nasa.gov

(Rec. 12 July 2006)

Abstract: The Columbia system at the NASA Advanced Supercomputing (NAS) facility is a cluster of 20 SGI Altix nodes, each with 512 Itanium 2 processors and 1 terabyte (TB) of shared-access memory. Four of the nodes are organized as a 2048-processor capability-computing platform connected by two low-latency interconnects – NUMALink4 (NL4) and InfiniBand (IB). To evaluate the scalability of Columbia with respect to both increased processor counts and increased problem sizes, we used seven of the NAS Parallel Benchmarks and all three of the NAS multi-zone benchmarks. For NPB we ran three Classes B, C, and D of benchmarks. To measure the impact of some architectural features, we compared Columbia results with results obtained on a Cray Opteron Cluster consisting of 64 nodes, each with 2 AMD Opteron processors and 2 gigabytes (GB) of memory, connected with Myrinet 2000. In these experiments, we measured performance degradation due to contention for the memory buses on the SGI Altix BX2 nodes. We also observed the effectiveness of SGI's NL4 interconnect over Myrinet. Finally, we saw that computations spanning multiple BX2 nodes connected with NL4 performed well. Some computations did almost as well when the IB interconnects was used.

Key words: Computer architectures, benchmarking, performance evaluation

1. INTRODUCTION

In summer 2004, NASA installed a large SGI Altix cluster named Columbia, which consists of twenty 512-processor nodes, and was ranked second on the Top500 computer list of November 2004 based on its LINPACK benchmark performance of 52 teraflops (Tflops) [1].

On the most recent (November 2005) Top500 list, four computing systems perform above the 50 Tflops point, and ten above 20 Tflops. The IBM Blue Gene/L [2-3, 15] tops the list, registering a stunning 280.6 Tflops on the LINPACK rating [1]. It should be noted, however, that while performance of the LINPACK benchmark is about 90-95 percent of the peak performance, user applications are unable to achieve sustained performance at that same level. There are several reasons for this.

First, a common vendor approach to providing high peak performance is to design systems with large processor counts. Often, however, applications are unable to use the additional processors effectively, either because of poor communication infrastructure on the system or because the codes use inappropriate algorithms.

Another issue is the fact that clock speeds double roughly every two years, whereas the speed of memory doubles every seven years, thus hindering good sustained performance of applications on these systems. It is important for applications scientists to recognize this bottleneck and determine methods of enhancing the performance of the general applications on these large systems with complex memory hierarchies [2].

To better understand Columbia's performance with respect to these potential issues, we conducted several scalability experiments using the NAS Parallel Benchmarks (NPB) [4]. We examined its strong scalability by using increased numbers of processors to solve fixed-sized problems. We also investigated weak scaling by studying larger problem sizes. In addition, we tested the impact of the SGI Altix memory organization and interconnect design by comparing benchmark results from Columbia with results from a Cray Opteron cluster. Finally, we examined Columbia's potential for scaling beyond 512 processors through experiments that spanned multiple Altix nodes.

The rest of this paper is organized as follows: In Section 2, we present the architectural details of the Columbia system and the Cray Opteron cluster. In Section 3, we describe the NPB and NPB-MZ benchmarks used in this study. In Section 4, we present and analyze the results of the benchmarking study. We conclude in Section 5 with a discussion of future work.

2. ARCHITECTURAL OVERVIEW

Here, we describe the architecture of the computing systems used in our experiments. While the focus of our work is the SGI Altix architecture in which comprises Columbia, we also describe the Cray Opteron cluster used for comparison purposes.

2.1. Columbia

Columbia consists of twenty 512-processor SGI Altix computers. Twelve of these are model 3700, and eight are model 3700 BX2 (hereafter called "BX2"). Since the experiments reported in this paper were conducted on BX2s, we'll confine our discussion to that architecture [5-8].

Each Altix node has global shared memory and is characterized as a Cache Coherent – Non-Uniform Memory Access (CC-NUMA) computer. Columbia is a single- system image (SSI) computer, which means that a single memory address space is visible to all the computing system resources. On a model BX2, SSI is achieved through NUMALink4 (NL4), a Non-Uniform Memory Access Flexible (NUMAflex) memory interconnect, where scaling can be done in three dimensions; namely the number of processors, memory capacity, and I/O capacity. This NUMAflex architecture supports up to 2,048 Intel Itanium 2 processors and 4 TB of memory capacity. With its fat-tree network topology, the bisection bandwidth scales linearly with the number of processors

Local cache-coherency between processors is implemented on the Front Side Bus (FSB). The Scalable Hub (SHUB) chip implements the global cache coherency protocol, which is a refinement of the protocol used in the directory-based DASH computing system developed at Stanford University [9]. The advantage of the directory-based cache–coherent protocol is that only the processors playing an active role in the usage of a given cache line need to be informed about the operation. This reduces the flow of information, using about three percent of the memory space for the directory.

In the SGI BX2 system, eight Intel Itanium 2 processors and four SHUB ASICs are grouped together in a brick, called a C-brick, which is connected by an NL4 interconnect to another C-brick. Each pair of processors shares a peak bandwidth of 3.2 gigabytes per second (GB/s). Peak bandwidth between nodes is 1.6 GB/s [2].

The 64-bit Itanium 2 processor runs at 1.6 gigahertz (GHz). It can issue two MADD (multiply and add) instructions per clock and has a peak performance of 6.4 Gflop/s. The memory hierarchy of a BX2 consists of 128 floating-point registers and three-level-on-chip data caches: 32-kilobytes (KB) of L1; 256-KB of L2 cache; and 9 megabytes (MB) of L3 cache.

At the NASA Advanced Supercomputing (NAS) facility, we have 12 SGI 3700 computers and eight BX2's. Four of the BX2s are organized as a capability platform by interconnecting them with two low-latency networks – NL4 and InfiniBand (IB) [1, 10]. The IB connects to the other sixteen 512-processor Altix nodes as well [2, 10].

IB is a network technology that defines very high-speed networks for interconnecting compute nodes and I/O nodes [2, 10]. It is an open industry standard for interconnecting both high-performance clusters of SMP (e.g., clusters of IBM POWER 5 or SGI Altix or NEC SX-8) and off-the-shelf processors, such as the Intel Itanium 2 or Intel Xeon [2].

Columbia's LINPACK results used in the TOP500 ranking were obtained using the IB network. IB cards and switches used at NAS are from Voltaire; measured latency and bandwidth are 10.5 microseconds and 855 megabytes per second (MB/s). This bandwidth is comparable to the measured bandwidth of NL4, used in the Columbia 2,048 system. However, the latency of IB is slower than NL4 by a factor of five.

2.2. Cray Opteron Cluster

In this work, we also used a 64-node Cray Opteron cluster located at NASA Ames [2, 6-8,11]. Each node has two AMD Opteron processors running at two GHz, and the nodes are connected with Myrinet. One node of the cluster is used as the server node and has four GB of memory. The remaining 63 nodes (126 processors) each have 2 GB of memory and are used as compute nodes. Peak performance of the system is 504 Gflop/s.

The Opteron processors in the cluster use a 0.13 micron copper CMOS process technology and can perform two floating-point operations per clock, giving a peak performance of four Gflop/s per processor. Each processor has an integrated memory controller that is, the memory controller is no longer in the Northbridge, but instead, is on the chip. This reduces the performance bottleneck, which in turn increases the application performance by reducing memory latency. Each processor can issue nine superscalar out-of-order instructions. Processor uses the HyperTransport technology [12], which is a high-speed, high-performance, point-to-point link for interconnecting integrated circuits on the motherboard. It also provides multi-processing with a "glue-less" chip-to-chip interconnect, thereby enabling scalability.

The 64 nodes of the Cray Opteron cluster are interconnected via a Myrinet network [13]. Myrinet is a packet-communication and switching technology widely used to interconnect servers or single-board computers. Myrinet uses cut-through routing and remote memory direct access (RDMA) to write to/read from the remote memory of other host adaptor cards, called Lanai cards. These cards interface with the PCI-X bus of the host they are attached to. Myrinet offers three ready-to-use switches with 8-256 ports each. The 8- and 16-port switches are full crossbars.

3. THE NAS PARALLEL BENCHMARKS

The NPB suite [4] contains eight benchmarks comprising five kernels (CG, FT, EP, MG, and IS) and three compact applications (BT, LU, and SP). The conjugate gradient (CG) benchmark is used in many spectral methods and is a good test of long-distance communication performance. In this benchmark, a CG method is used to compute an approximation to the smallest eigenvalue of a large, sparse, symmetric positive definite matrix. This kernel is typical of unstructured grid computations in that it tests irregular long-distance communication and employs sparse matrix-vector multiplication. In the FT benchmark, a 3D partial differential equation is solved using Fast Fourier Transforms (FFTs). This kernel tests global all-to-all communication. EP accumulates certain statistics of Gaussian random numbers and has virtually no interprocessor communications. MG performs simple multigrid calculations and has highly structured short- and long-distance communications. IS performs a sort operation that is important in “particle” codes.

In addition, there are three compact applications: BT, LU, and SP. LU is a regular-sparse, block (5x5) lower and upper triangular system solver. This code is typified at NASA Ames by the code INS3d-LU. SP computes the solution of multiple, independent systems of non-diagonally dominant, scalar penta-diagonal equations. BT performs solutions of multiple, independent systems of non-diagonally dominant, block tridiagonal equations with a 5x5 block size. Both SP and BT are typified at NASA by the code ARC3D.

Recent effort in NPB development was focused on new benchmarks, including the new multi-zone version, called NPB-MZ [4, 14]. While the original NPBs exploit fine-grain parallelism in a single zone, the multi-zone benchmarks stress the need to exploit multiple levels of parallelism for efficiency and to balance the computational load. NPB-MZ contains three application benchmarks: BT-MZ, SP-MZ, and LU-MZ, which mimic the overset grid (or zone) system found in the OVERFLOW code. BT-MZ (uneven sized zones) and SP-MZ (even sized zones) test both coarse- and fine-grain parallelism and load balance. LU-MZ is similar to SP-MZ but has a fixed number of zones (4 × 4 = 16).

For our experiments, we used the message passing interface (MPI) implementation of the original NPBs and the hybrid MPI + OpenMP implementation of the NPB-MZ. All code came from the latest NPB3.2 distribution [4].

4. RESULTS

In this section, we present the results of our experiments on performance and scalability. First, we show the impact of memory bandwidth on Altix scalability. We then show how the NAS Parallel Benchmarks scale on Columbia with respect to both increasing processor counts and increasing problem sizes. Next, we compare results with a Cray Opteron cluster. Finally, we show how computations scale beyond a single Altix node.

4.1. Memory Contention

The effect of memory contention is estimated by running simultaneous copies of a serial benchmark and comparing performance to that of a single copy. Like *DGEMM and *FFT in the High Performance Computing Challenge (HPCC) benchmarks [16], we define an NPB-STAR benchmark that consists of running N simultaneous copies of each NPB on an N-processor system [16].

In Figure 1 we plot the percentage of degradation of six Class A and B benchmarks (CG, MG, FT, BT, SP, and LU) for an SGI Altix BX2 and a Cray Opteron Cluster.

As expected, performance degradation is highest (about 36-40 percent) for the CG and MG benchmarks on the Altix because their memory usage patterns cause contention on the shared-memory bus. This was expected because MG is memory bound while CG is memory sensitive and involves indirect addressing (Sparse BLAS 1) with a dot product (two loads and one store). Performance degradation for the FT and SP benchmarks is about half that of CG and MG. For BT and LU, the performance degradation is less than five percent.

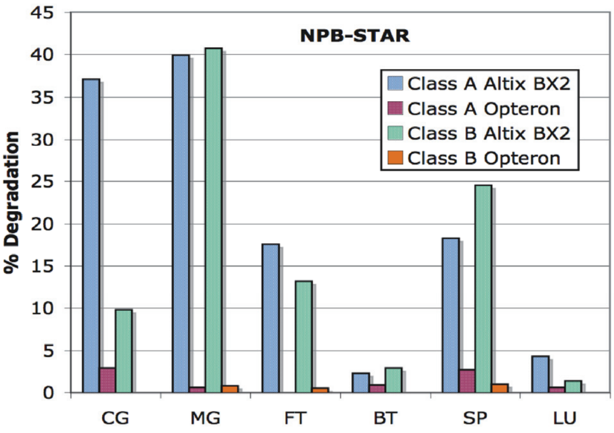


Fig. 1. Percentage of degradation of NPB-STAR Classes A and B on BX2 and Cray Opteron Cluster

In contrast to the performance degradation of NPB-STAR on the BX2, there is no significant performance degradation on the Cray Opteron Cluster, except for about three percent in Class A of CG and SP. The performance degradation for Class B on the BX2 shows the same trend as for Class A, except for CG which shows a much smaller penalty. In comparison to Class A, the Class B benchmarks show much less memory performance degradation on the Cray Opteron Cluster.

4.2. NPB on SGI Altix BX2

In Figure 2 we plot the performance (in Gflop/s) on a BX2 of the MG benchmark for Classes B, C, and D. For 16 and 32 processors, performance for all three classes is the same, because data does not fit in cache. For 64 and 128 processors, performance of Class B is better than Classes C and D, as data for the smaller B class fits into the cache. For 256 processors, MG Class B suffers from a higher communication-to-computation ratio, resulting in worse performance than the other two classes. From 64 to 256 processors, performance of Class D is better than Class C, as the Class D computation-to-communication ratio is high.

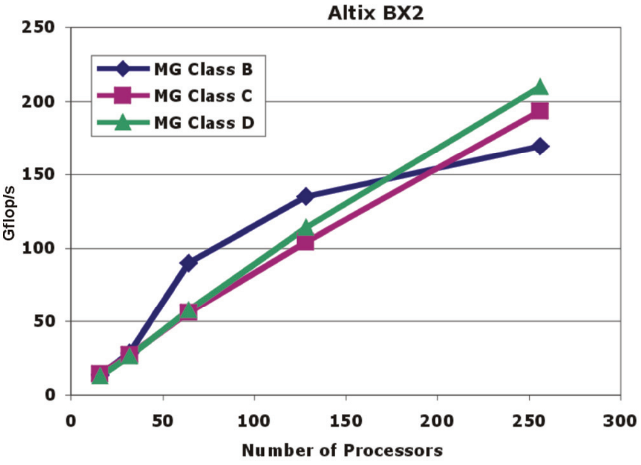


Fig. 2. MG Classes B, C, and D on BX2

In Figure 3 we plot the performance of CG, Classes B, C, and D. At 16 processors, all three classes have a similar performance. For 32 and 64 processors, the performance of Classes C and D is almost the same, whereas the performance of Class B is almost double. For 128 processors and higher, Class D performed the best. For CG, per-processor performance is very poor – ranging from 80 to 220 Mflop/s, which translates to only 1-3 percent of the peak performance of the Intel Itanium 2 processor. In this benchmark, most of the work is done in the sparse BLAS 1 (sparse matrix times vector) kernel and, as such, involves indirect addressing. The advantage of more cache as the number of processors increases does not help because noncontiguous memory accesses keeps the cache miss rate high. The larger problem size, Class D, did not result in better performance, which, like FT, (see below) is the opposite of the other benchmarks.

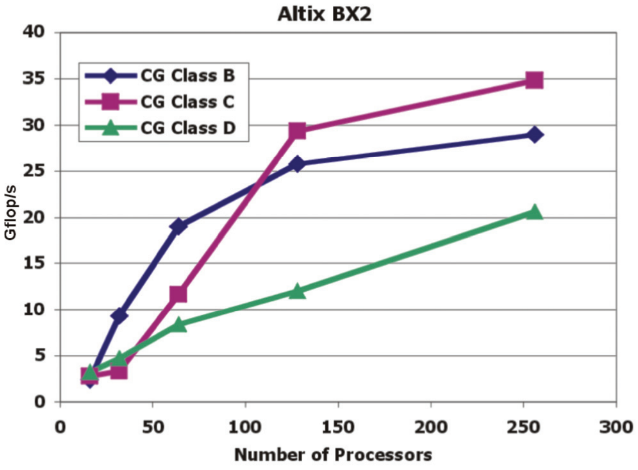


Fig. 3. CG Classes B, C, and D on BX2

In Figure 4 we plot FT's performance on Classes B, C, and D. For 16 and 32 processors, the performance of Classes B and C are almost the same, but Class B outperforms the other two Classes as the number of processors increases. Like CG, per-processor performance of FT is also very poor, ranging from 400 to 500 Mflop/s, which translates to only 6-8 percent of the peak performance. In this benchmark, most of the work done is in transposing the matrix, which involves all-to-all communication. It stresses the entire network of the system. Similar to CG, the larger problem size, Class D of FT had the worst performance.

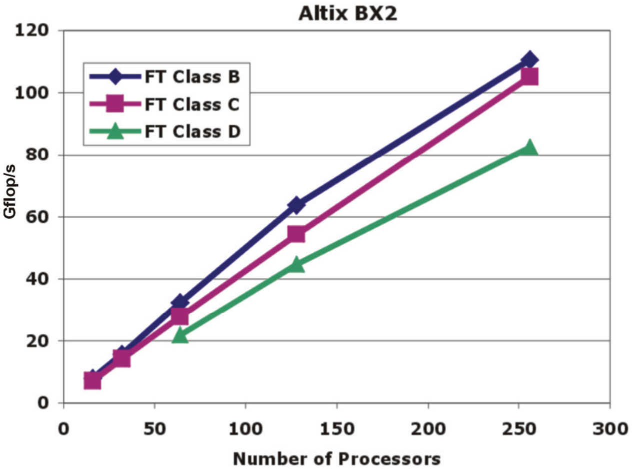


Fig. 4. FT Classes B, C, and D on BX2

In Figure 5 we plot the performance of classes B and C of the IS. Note that a Class D problem for IS has not been specified in NPB3.2. Class B scales up to 64 processors and then plateaus. Class C scales up to 128 processors, but the performance drops dramatically at 256 processors.

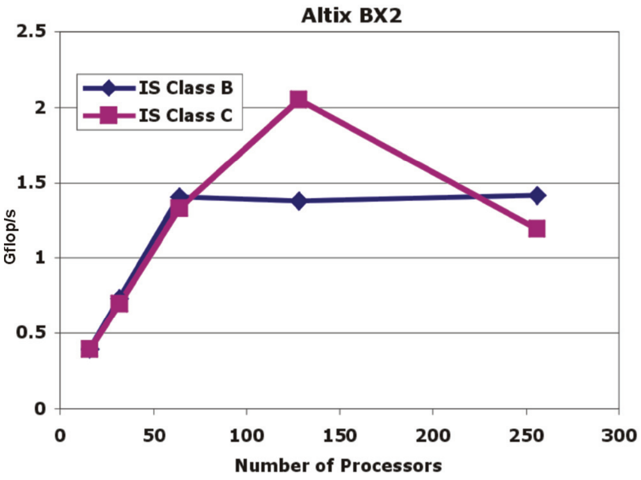


Fig. 5. IS Classes B and C on BX2

In Figure 6 we show the performance of Classes B, C, and D of the LU benchmark. For 16, 32, and 64 processors, the performance of all three Classes are the same, but at higher processor counts, Class D scales better than Classes B and C. The smaller problem size, Class B, suffers the most from the higher communication-to-computation ratio of LU.

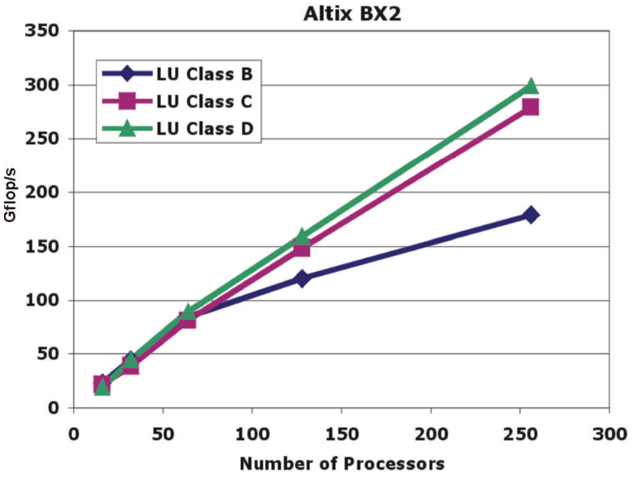


Fig. 6. LU Classes B, C, and D on BX2

In Figure 7 we plot the performance of Classes B, C, and D of the SP benchmark. For 16 and 25 processors, performance of Classes B and C are almost the same, whereas the performance of Class D is lower than either B or C. From 64 to 121 processors, the performance of all three classes is almost the same, Class B having slightly better performance than the other two. Beyond 121 processors, larger problem sizes illustrated better performance as a result of an improved computation-to-communication ratio. For 484 processors, per-processor performance (Mflop/s) is about 200 for Class B, 280 for Class C, and 380 for Class D, which are about 3-6 percent of peak.

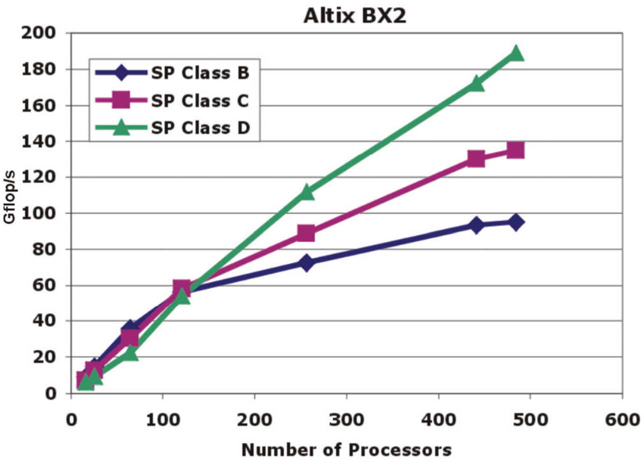


Fig. 7. SP Classes B, C, and D on BX2

The results for Classes B, C, and D of the BT benchmark are plotted in Figure 8. Except for absolute performance, which has more than doubled, the trend is very similar to that of the SP benchmark shown in Figure 7. At 484 processors, per-processor performance of Class D is 930 Mflop/s, which is about 15 percent of peak

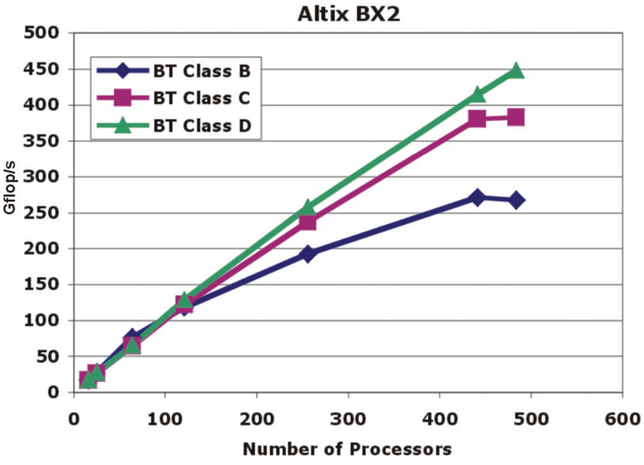


Fig. 8. BT Classes B, C, and D on BX2

4.3. NPB-MZ on SGI Altix BX2

Classes B, C, and D of the NPB multi-zone versions were run on an SGI Altix BX2. In this section, we present the results of those runs.

Figure 9 shows the BT-MZ results. Class B of BT-MZ has 64 zones. As the MPI parallelization in the multi-zone benchmarks exploits only zonal parallelism, and the zones for BT-MZ are different sizes, it is not possible to load-balance the Class B problem with more than 16 MPI processes. Additional scaling beyond 16 processors requires the use of OpenMP threads. Performance starts to degrade on 32 processors for Class B and 256 processors for Class C when more OpenMP threads are invoked. Class D shows close to 680 Gflop/s at 504 processors, which is about 22 percent of the peak performance.

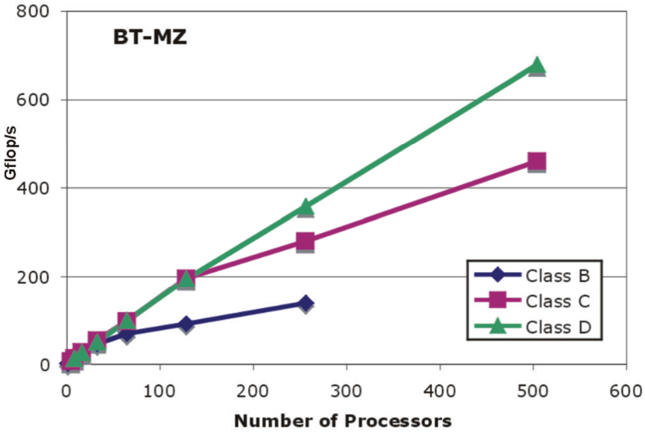


Fig. 9. BT-MZ Classes B, C, and D on BX2

Figure 10 shows the SP-MZ results. Load for this benchmark is perfectly balanced if the number of zones for a given problem class (64 for Class B, 256 for Class C, and 1024 for Class D) is divisible by the number of processors. Performance of Class B is the best up to 64 processors, and then degrades when OpenMP threads have to be used. The Class C problem has perfect scaling up to 256 processors, and then has a sudden drop at 504 processors, mainly due to load imbalance. The Class D problem does not fit into cache even at 256 processors, which translates into worse performance than Class C.

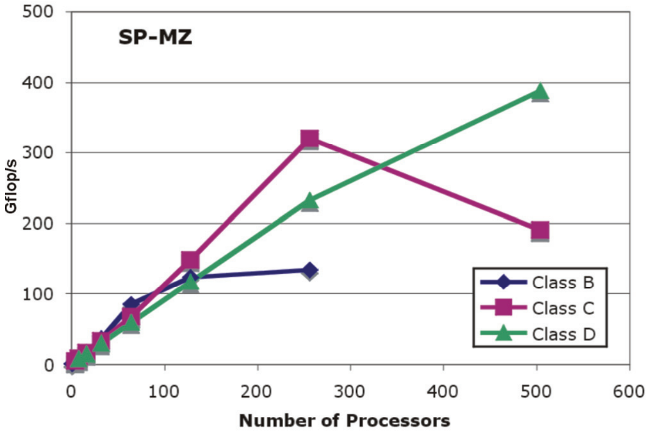


Fig. 10. SP-MZ Classes B, C, and D on BX2

Figure 11 shows the LU-MZ results. Because of a fixed number of zones (4x4), LU-MZ can only use up to 16 MPI processes. Additional scaling beyond 16 processors requires the use of OpenMP threads. At a small number of processors, both Class B and Class C show better performance than Class D, which can be attributed to better cache utilization. Beyond 32 processors, Class B scaled poorly, indicating the cost of shared data access from OpenMP diminished the performance gain from more processors. We observe a similar trend for Class C at 128 processors and beyond. Although Class D scaled up to 496 processors, it achieved only 9 percent of the peak performance.

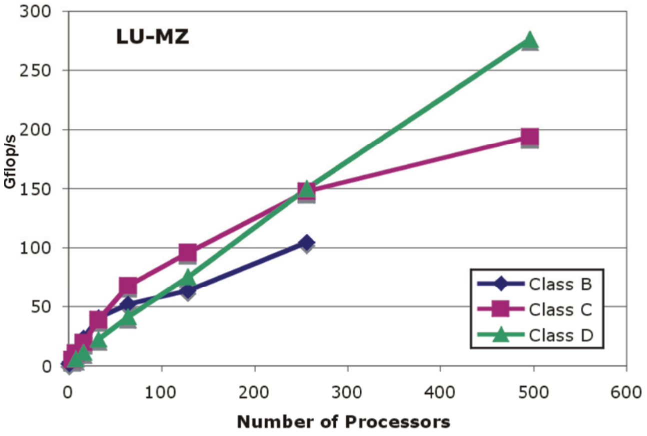


Fig. 11. LU-MZ Classes B, C, and D on BX2

4.4. Comparison to Cray Opteron Cluster

It is often instructive to compare performance results across different computing platforms. The limitations of one system can become more apparent when compared to another. In this section, we compare Columbia's benchmark results to those of a Cray Opteron cluster. Initially, we look at scalability within the Cray cluster, and then we do direct comparison of results between Columbia and the Cray.

In Figure 12 we plot the performance (in Gflop/s) of MG Classes B, C, and D for various numbers of processors on the Cray Opteron Cluster. For 16 and 32 processors, performance for Class C is higher than Class B and shows reasonable scalability. Because of limited memory for each processor, we were not able to run the Class D problem on 16 processors. (This was also the case with several other benchmarks.) On 32 processors, Class D and Class C had similar performance. However, on 64 processors, performance of Class D is worse than Class C, but better than that of Class B. This can be attributed to better cache utilization for Class C, and a larger communication-to-computation ratio for Class B. At 64 processors, performance per processor is about 400, 500, and 550 Mflop/s for Class B, D, and C, respectively.

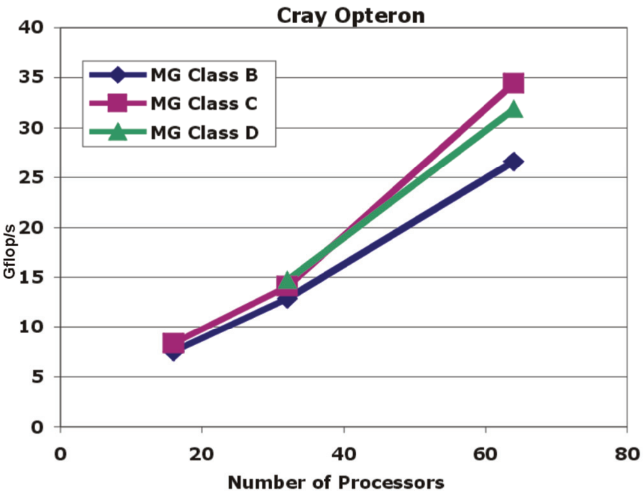


Fig. 12. MG Classes B, C, and D on Cray cluster

In Figure 13 we plot the performance of Classes B, C, and D of the CG benchmark. For 16, 32, and 64 processors, the performance of Classes B and C are almost same, with the exception of Class B being slightly higher at 32 processors. Performance of Class D is consistently poor as compared to Classes B and C because data in Class D is so large that it does not fit in the cache. For CG, per-processor performance is very poor, ranging from 45 to 70 Mflop/s, which translates to only 1-2 percent of the peak performance of the Opteron processor. In this benchmark, most of the work is done in the sparse BLAS 1 (sparse matrix times vectror) kernel, which involves.

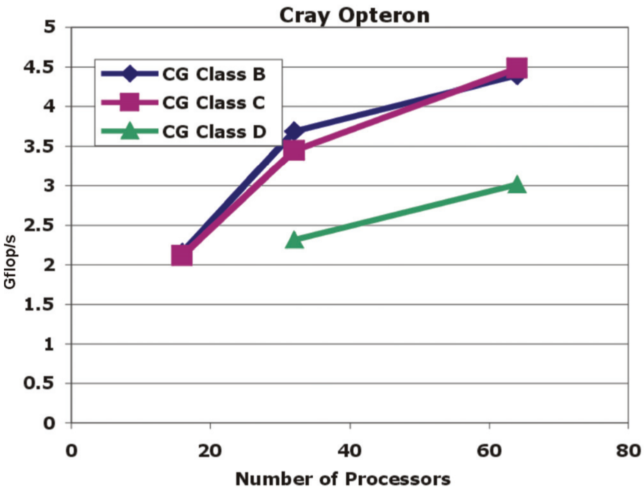


Fig. 13. CG Classes B, C, and D on Cray cluster

In Figure 14 we plot the performance of Classes B and C of the FT benchmark. Because there was insufficient memory, we were not able to run Class D. In this benchmark, most of the work is done in transposing the matrix, which involves all-to-all communication and stresses the system interconnect. For 16 and 32 processors, the performance of Class B is slightly better than Class C. For 64 processors, performance of both Classes B and C are similar, even though the larger-size problem, Class C, requires more communication bandwidth. Like CG, per-processor performance of FT is poor, ranging from 190 to 250 Mflop/s (which translates to only 5-6 percent of the peak performance of the AMD Opteron processor).

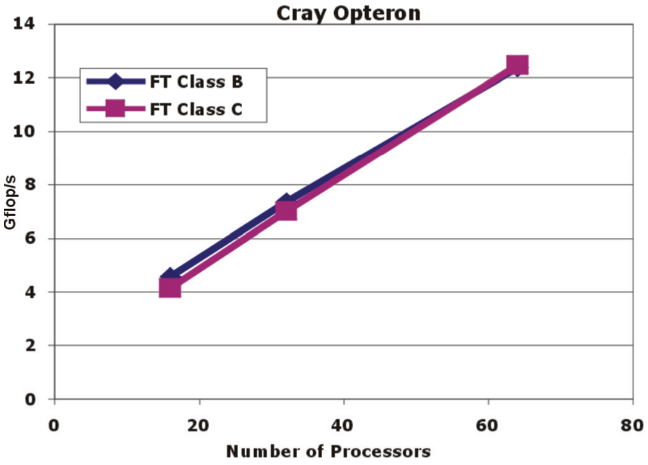


Fig. 14. MG Classes B and C on Cray cluster

In Figure 15 we show the performance of the IS benchmark for Classes B and C. On 16 processors, both classes have a similar performance. However, the Class B problem scaled better on larger processor counts. More

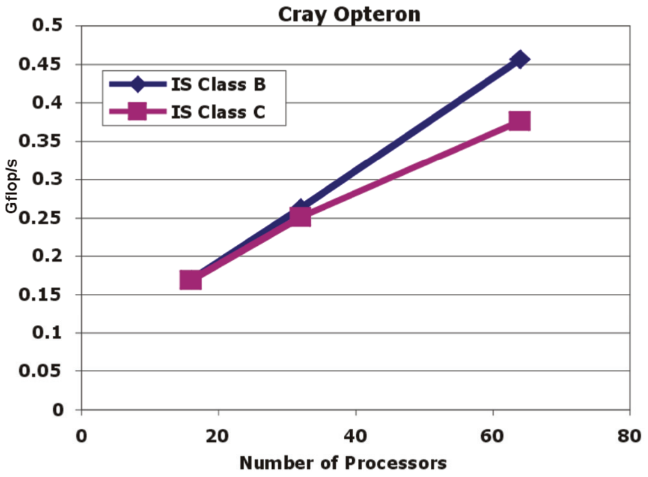


Fig. 15. IS Classes B and C on Cray cluster

data was communicated for the larger problem size, resulting in performance degradation.

In Figure 16 we plot the performance of the LU benchmark for classes B, C, and D. For 16 and 32 processors, the performance of Classes B and C are almost the same. The performance of both is better than that of Class D because its data does not fit into cache. At 64 processors, the performance of Class C is better than B, which in turn is better than Class D.

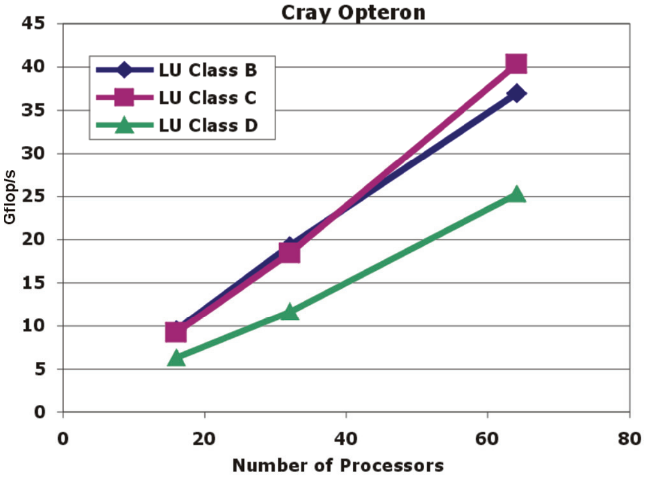


Fig. 16. LU Classes B, C, and D on Cray cluster

In Figure 17 we plot the performance of the SP benchmark for Classes B, C, and D. For 16 and 25 processors, the performance of Classes B and C are almost same. From 25 to 121 processors, the performance of Class D is better than Class C, which in turn is better than Class B. At 121 processors, per-processor performance is about 215, 270, and 290 Mflop/s for Class B, C, and D, respectively (which is about 5-7 percent of peak performance).

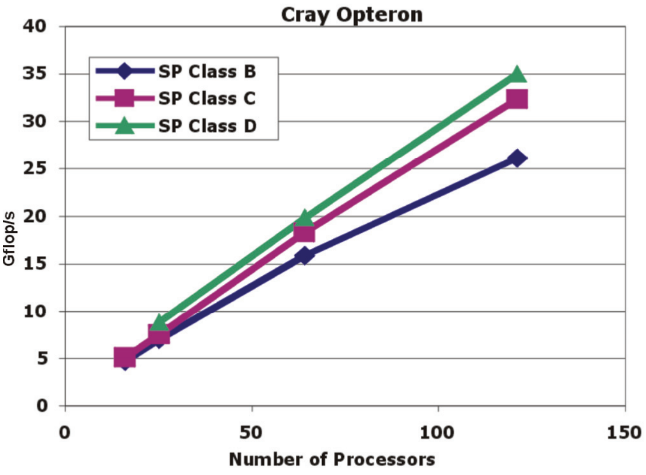


Fig. 17. SP Classes B, C, and D on Cray cluster

In Figure 18 we show the performance of the BT benchmark for Classes B, C, and D. Because of larger memory requirements for Class D, we could not run the problem at 16 or 25 processors. The scaling of BT is very similar to SP, as shown in Figure 14, although BT achieves about 15-20 percent of peak performance.

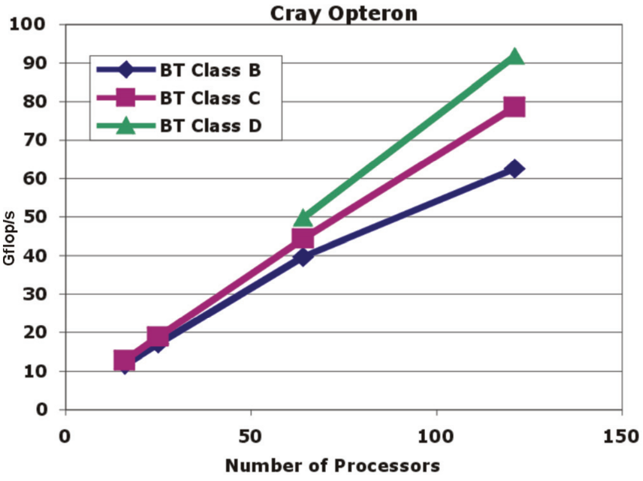


Fig. 18. BT Classes B, C, and D on Cray cluster

In Figure 19 we plot the performance of the MG Class C benchmark on both the SGI Altix BX2 and the Cray Opteron Cluster. The Altix performance is consistently better than that of the Cray Opteron Cluster for the entire range of processors – from 16 to 64. In addition, as the number of processors increases, the performance gap between the two systems increases. The smaller processor cache offsets better memory bandwidth of the AMD Opteron and the relatively poor network bandwidth of the Myrinet interconnect used in the Cray Opteron cluster. Overall, BX2 performance is about 40 percent better for a given processor count.

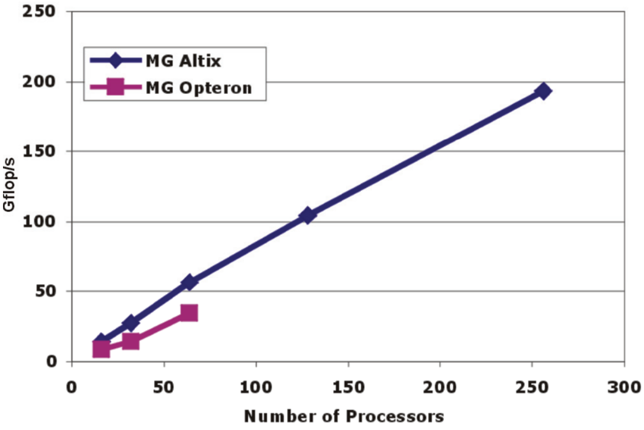


Fig. 19. MG Class C on BX2 and Cray Opteron cluster

In Figure 20 we compare the performance of the BX2 and the Opteron cluster on CG Class C. Again, Altix performance is consistently better than that of the Cray Opteron Cluster for the entire range of processors – from 16 to 64, except at 32 processors where the performance of two systems is the same. As the number of processors increases to 64, the performance gap between the two systems increases. At 64 processors, Altix BX2 performance is about 140 percent better.

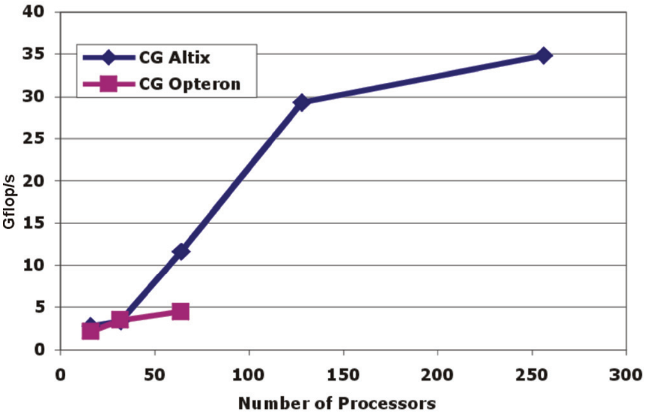


Fig. 20. Performance of CG Class C on SGI Altix BX2 and Cray Opteron cluster

In Figure 21 we plot the performance of the FT Class C benchmark for the BX2 and the Cray Opteron Cluster. Altix performance is consistently better in the entire range of processors – from 16 to 64. In addition, as the number of processors increases, the performance gap between the two systems also increases. Better memory bandwidth of the AMD Opteron is offset by the relatively poor network bandwidth of the Cray's Myrinet. The FT benchmark involves all-to-all communication and stresses the global network of the system and will perform better on a system with higher bisection bandwidth. For 64 processors, performance of the Altix is almost double as it has a higher bisection bandwidth than that of the Cray Opteron cluster.

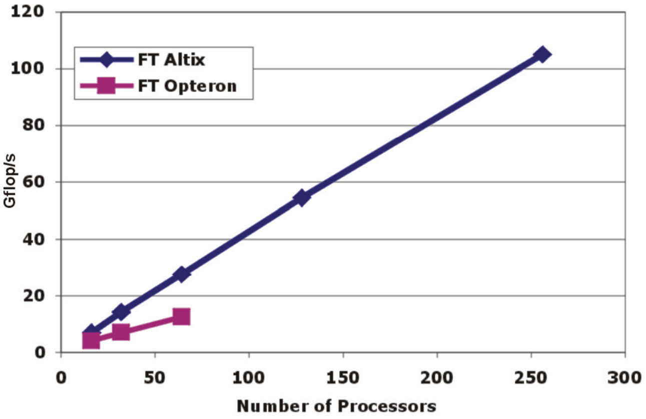


Fig. 21. FT Class C on BX2 and Opteron cluster

In Figure 22 we show the performance of the IS Class C benchmark on the BX2 and the Cray Opteron Cluster. Altix performance is consistently better by two to three times. Because of the small system size of the Opteron cluster, we are not able to confirm the performance drop at 256 processors that occurred on the Altix.

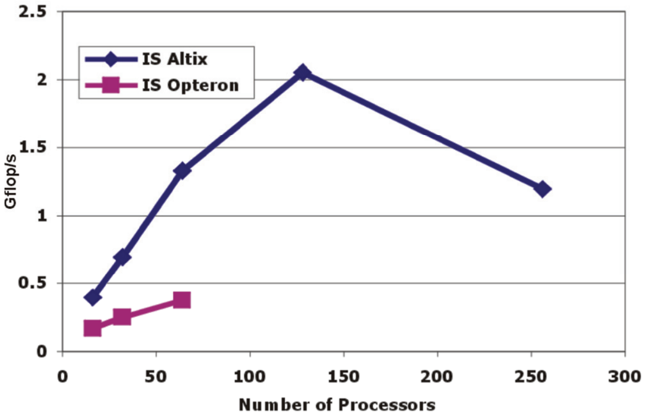


Fig. 22. IS Class C on BX2 and Opteron cluster

In Figure 23 we compare the performance of the LU Class C benchmark on the BX2 and the Cray. The Altix's performance is consistently better than the Cray's in the entire range of processors – from 16 to 64. In addition, as the number of processors increases, the performance gap between the two systems also increases. Better memory bandwidth of the AMD Opteron is offset by the relatively poor interconnect performance. At 64 processors, the performance of the Altix is almost double that of the Cray Opteron Cluster.

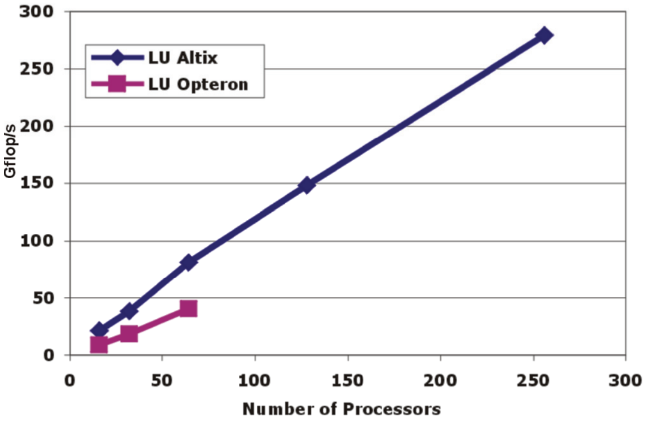


Fig. 23. LU Class C on BX2 and Opteron cluster

In Figures 24 and 25 we show the performance of the SP and BT Class C benchmarks, respectively. Again, the Altix's performance is consistently better than the Cray Opteron Cluster's—over the entire range of processors – from 16 to 121, except at 16 processors where the performance of the two systems is similar. In addition, as the number of processors increases from 25 to 121, the performance gap between the two systems also increases. At 121 processors, the BX2's performance is 80 percent and 50 percent better than Cray Opteron Cluster for SP and BT, respectively. The knee in the scaling curve at 121 processors for SP on the BX2 has yet to be observed on a larger Opteron cluster.

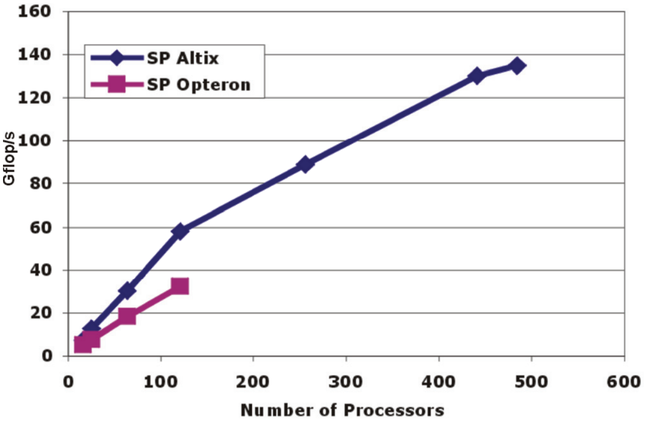


Fig. 24. SP Class C on BX2 and Opteron cluster

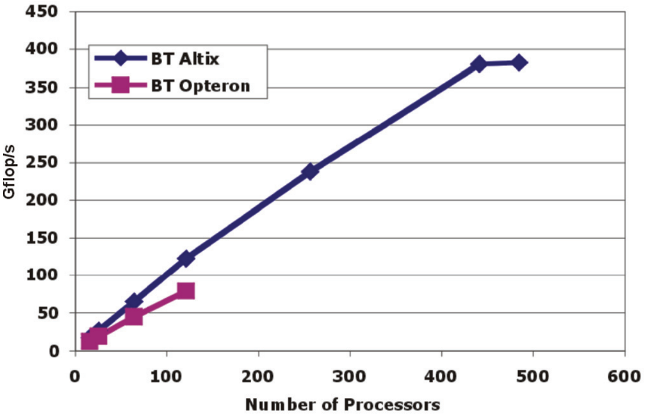


Fig. 25. BT Class C on BX2 and Opteron cluster

4.5. Performance Across Multiple Altix BX2 Nodes

We also wanted to investigate the performance of computations involving processors spread across multiple Altix nodes. All of Columbia's nodes are connected with IB, and four of the Altix BX2's are also connected via NL4 outside the nodes to form a 2,048-processor capability sub-cluster. We ran the Class C benchmarks across two and four Altix nodes in this Columbia 2,048 system using both IB and NL4, and compared the results to runs performed on one node. This section presents the results of those runs. For runs performed across nodes, an equal number of processors were used on each.

In Figure 26 we plot the performance of the MG Class C benchmark on the Columbia 2,048 system. Here, 1–host means a single SGI Altix BX2 node with 512 processors, connected by a fat-tree SGI proprietary NL4 interconnect (as discussed in section 2). Performance of the MG benchmark from 16 to 256 processors is almost the same for 1-host, 2-host (xpmem), and 4-host (xpmem). The "xpmem" designation refers to the software layer employed by communication across nodes using NL4. While the performance using NL4 is encouraging, cross-node performance using IB (the 2-host (IB) and 4-host (IB) results) is much lower than that of the NL4 interconnect.

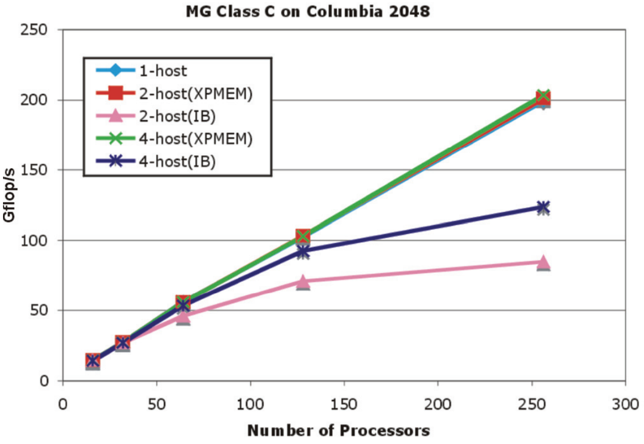


Fig. 26. MG Class C on the Columbia 2,048

In Figure 27 we show the performance of the CG benchmark on the Columbia 2,048 system. Up to 128 processors, the performance of 1-host, 2-host (xpmem), and 4-host (xpmem) is almost the same, and above 128 processors, performance of 2-host (xpmem) is better than 4-host (xpmem), which in turn is better than 1-host. However, performance of both 2-host (IB) and 4-host (IB) is much lower than the corresponding NL4 results. In addition, performance of 4-host IB is better than 2-host IB. This is due to the dot product operation in the sparse Basic Linear Algebra Subroutines 1 (BLAS 1), which is both latency and bandwidth sensitive, and the latency of the IB network is much higher than the corresponding latency of the NL4 network. Also, the bandwidth of the IB network is lower than the corresponding bandwidth of NL4. Performance across 4-hosts using IB is better than across 2-hosts, as the random ring bandwidth of 4-host IB is better than the bandwidth of 2–host IB [17] (which is due to the larger number of IB cards in 4-hosts than 2-hosts). From this, one can conclude that to get good performance for CG, a system with a low latency and high bandwidth network is needed.

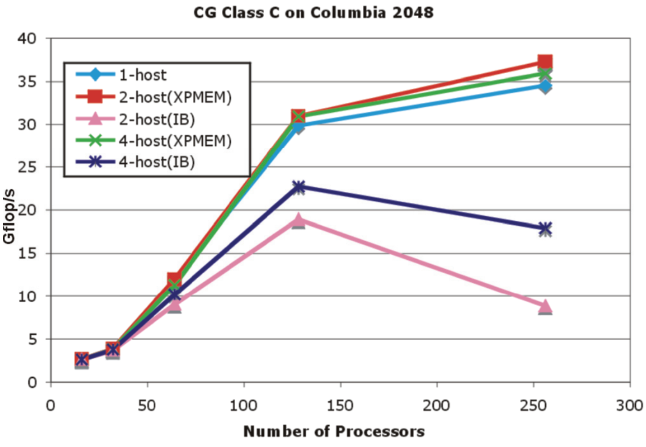


Fig. 27. CG Class C on the Columbia 2,048

In Figure 28 we plot the performance of the FT benchmark on the Columbia 2,048 system. Up to 128 processors, the performances of 1-host, 2-host (xpmem), and 4-host (xpmem) are almost the same, but beyond 128 processors, performance of the 4-host (xpmem) lags behind the 1- and 2-host (xpmem). However, performance results of both the 2-host (IB) and 4-host (IB) are much lower than that of the corresponding NL4 results. Again, performance of 4-host IB is better than the 2-host IB. Performance using NL4 is better than that of IB as a result of the all-to-all communication required in the parallel matrix transpose of the FT benchmark. Each processor sends messages to all other processors, and this stresses the global bandwidth of the system interconnect, which is higher for NL4 than IB.

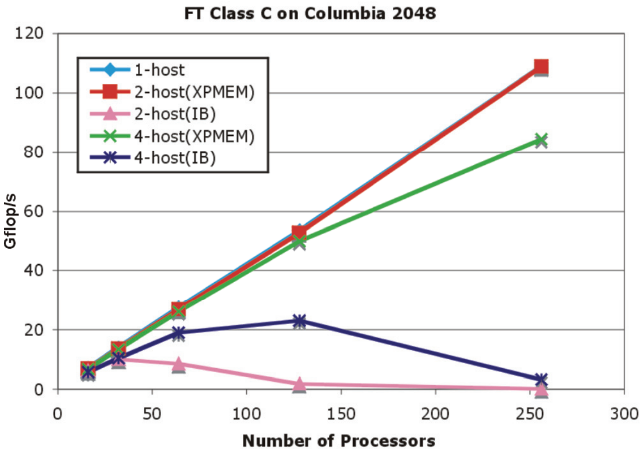


Fig. 28. FT Class C on the Columbia 2048

In Figure 29 we plot the performance of the LU benchmark on the Columbia 2,048 system. The cross-node results using NL4 are about the same as the 1-host results. The 16 and 32 processor 2-host and 4-host IB results are comparable to the NL4 results and only lag by 20 percent at 256 processors.

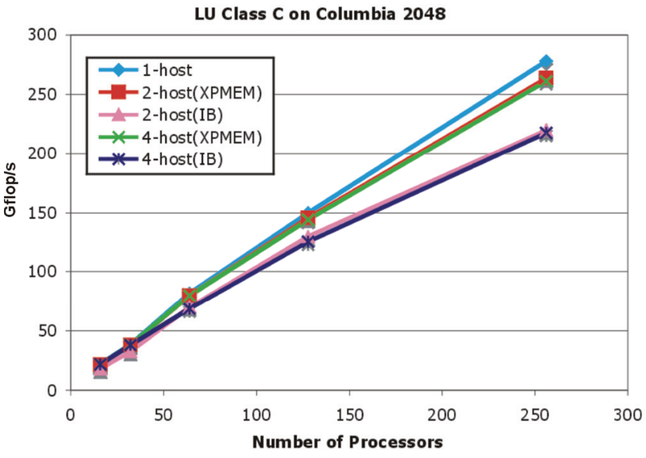


Fig. 29. LU Class C on the Columbia 2,048

In Figure 30 we show the performance of the SP benchmark on the Columbia 2,048 system. The results of running across nodes using NL4 and IB are comparable to that of 1-host. At 256 processors, the 2-host and 4-host NL4 results are actually better than the 1-host results. The 1-host and 4-host (IB) results are almost the same, and the 2-host (IB) results lag behind the others by about 20 percent or less. Once again, like the MG, CG, and FT benchmarks, performance of the SP benchmark on 4-host IB is better than on 2-host IB.

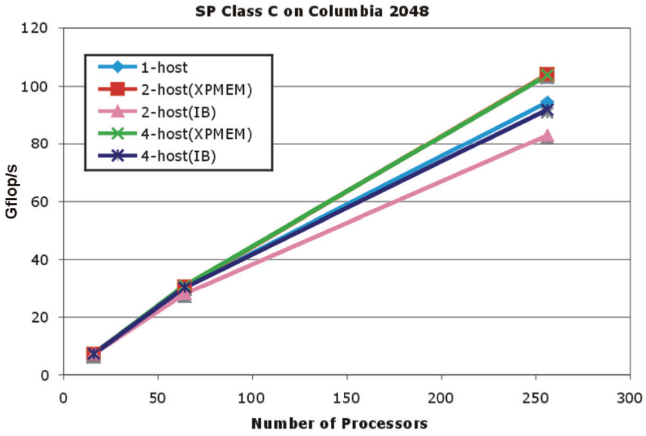


Fig. 30. SP Class C on the Columbia 2,048

In Figure 31 we plot the performance of the BT benchmark on the Columbia 2,048 system. Here, the results of running on 1-host or across nodes using either NL4 or IB are very comparable. This benchmark sees the least amount of performance degradation running across nodes with the IB interconnect among the six benchmarks tested.

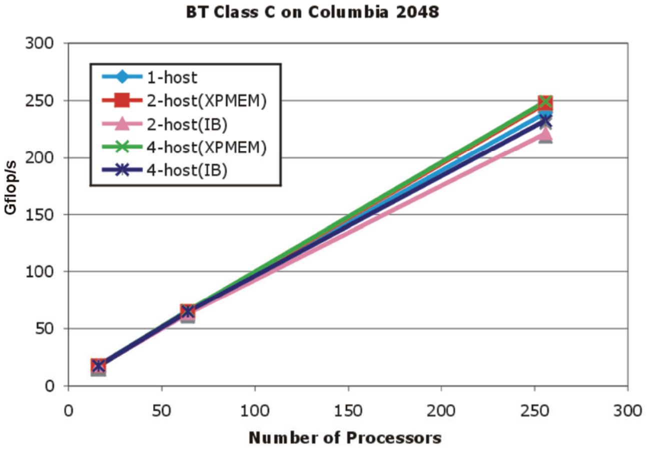


Fig. 31. BT Class C on the Columbia 2,048

5. CONCLUSIONS

By running several copies of a serial benchmark and comparing performance to that of a single copy, we investigated the effect of memory contention in the Altix and Cray Opteron Cluster. The MG and CG benchmarks showed that performance degradation due to memory contention could be quite severe for the SGI Altix BX2 because the two processors on each node in a C-Brick share the same memory bus. In contrast, memory contention is almost negligible in the Cray Opteron cluster. The AMD Opteron's integrated memory controller allows the processor to access local-RAM without using the HyperTransport bus. Each Opteron processor can access the main memory of another processor, transparent to the application scientist. The Opteron approach to multi-processing is not the same as standard symmetric multiprocessing—instead of having one bank of memory for all processors, each processor has its own memory. In contrast, in the BX2 system, two processors share a single common bus for both processor-processor and processor-memory communication, and contention for the shared bus causes computing efficiency to drop.

NPB Class D benchmarks showed the best performance and scalability with the exception of the CG and FT benchmarks where Class D was the worst. Performance of the NPB benchmarks on the BX2 is much better than on the Cray Opteron Cluster because the NL4 interconnect has lower latency and higher bandwidth than the Myrinet interconnect used in the Cray Opteron cluster.

When running benchmarks spanning multiple BX2 nodes of Columbia's 2,048 system, the performance of NPB and NPB-MZ benchmarks using NL4 was better than using IB. Our study emphasizes the importance of good memory bandwidth per processor and good interconnects in a high-end computing system. In the future, we plan to include parallel I/O benchmarks [18] and real applications, and extend the study to the Cray XT3, the IBM POWER5/6, and the NEC SX-8 [2, 5-8].

References

[1] Top500, http://www.top500.org

[2] S. Saini, Hot Chips and Hot Interconnects for High End Computing Systems, M4, IEEE/ACM SC 2004, Pittsburgh (2004).

[3] S. Saini, Performance Comparison of Columbia 2048 and IBM Blue Gene/L, SGIUG 2005 Technical Conference and Tutorials, June 13-16, 2005 Munich (2005).

[4] NAS Parallel Benchmarks, http://www.nas.nasa.gov/Resources/Software/npb.html (2006).

[5] S. Saini, R. Ciotti, T. N. Gunney, T. E. Spelce, A. Koniges, D. Dossa, P. Adamidis, R. Rabenseifner, S. R. Tiyyagura, M. Mueller, and Rod Fatoohi, Performance Evaluation of Supercomputers using HPCC and IMB Benchmarks IPDPS 2006, PMEO, April 25-29, Rhodes, Greece (2006).

[6] S. Saini, R. Fatoohi, and R. Ciotti, Interconnect Performance Evaluation of SGI Altix 3700 BX2 Cray X1, Cray Opteron Cluster, and Dell PowerEdge, IPDPS 2006, PMEO, April 25-29, Rhodes, Greece (2006).

[7] S. Saini, R. Ciotti, T. N. Gunney, T. E. Spelce, A. Koniges, D. Dossa, P. Adamidis, R. Rabenseifner, S. R. Tiyyagura, M. Mueller, and Rod Fatoohi, Performance Comparison of Cray X1 and Cray Opteron Cluster with Other Leading Platforms Using HPCC and IMB Benchmarks, CUG 2006, May 8-11, 2006 Lugano, Switzerland, (2006).

[8] S. Saini, P. Adamidis, R. Fatoohi,, J. Chang, and R. Ciotti, Performance Analysis of Cray X1 and Cray Opteron Cluster, CUG 2006, May 8-11, 2006 Lugano, Switzerland (2006).

[9] D. Lenoski, J. Laudon, K. Gharachorloo, A. Gupta and J. Hennessy, International Conference on Computer Architecture archive Proceedings of the 17th annual international symposium on Computer Architecture, Seattle, Washington, USA, 148-159 (1990).

[10] InfiniBand Trade Association, InfiniBand Architecture Specifications, Release 1.0 October 24, 2000, http://www.infinibandta.org/home/

[11] Advanced Micro Devices, http://www.amd.com/us-en/

[12] HyperTransport Consortium, http://www.hypertransport.org/

[13] Myricom, http://www.myri.com/

[14] H. Jin and R. Van de Wijngaart, Performance Characteristics of the Multi-zone NAS Parallel Benchmarks, Proceedings of International Parallel and Distributed Processing, Santa Fe, New Mexico, USA, (2004).

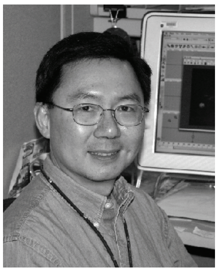
[15] The Blue Gene/L Team, IEEE/ACM Proceedings of SC 2002, Baltimore, Maryland, USA (2002).

[16] HPC Challenge Benchmark, http://icl.cs.utk.edu/hpcc/, (2006).

[17] R. Biswas, M. J. Djomehri, R. Hood, H. Jin, C. Kiris, and S. Saini, An Application-Based Performance Characterization of the Columbia Supercluster, IEEE/ACM SC 2005: 26 (2006).

[18] S. Saini, D. Talcott, H. Yeung, G. Myers, and R. Ciotti, A Scalability Study of SGI Clustered XFS Using HDF5 Based AMR Application, SGIUG 2006 Technical Conference and Tutorials, June 6-9, 2006 – Las Vegas, USA (2006).

SUBHASH SAINI received his Ph.D from the University of Southern California and has held positions at University of California at Los Angeles (UCLA), University of California at Berkeley (UCB), and Lawrence Livermore National Laboratory (LLNL). He has 11 years of teaching experience at graduate level. Since 1989, he is a senior scientist at the NASA Advanced Supecomputing (NAS) program at NASA Ames Research Center. He is a senior visiting scientist at LLNL under a participating guest program   He has been a highly rated tutorial speaker at SC 92, SC `94, SC `95, SC '96, SC '97 and 'SC 98. His SC '94, SC '95, SC '96, SC '97 and SC' 2004 tutorials on high end computing drew the highest number of attendees in any of the pre-conference tutorials. His research interests involves performance evaluation and modeling of new generation of highly parallel computers including next generation of petaflop class computers. He has published 139 technical papers and presented over 250 technical talks. He has won several awards for "Excellence in Teaching" including one from USC. In 1992, he was named the NAS employee of the year. In 2001, he was a co-author of a Best Technical Paper Award at SC 2001. Currently, he is a member of US High End Computing Revitalization Task Force (HECRTF) Interagency Working Group (HECIWG), DARPA HPCS team and its I/O Working Group.



JOHNNY CHANG – NASA Ames Research Center/CSC. Johnny is a member of the Application Performance and Productivity group at the NASA Advanced Supercomputing (NAS) Division located in Moffett Field, California. He is part of a group that provides consulting service to the 700+ users of the Columbia supercomputer – a luster of twenty 512p SGI Altix systems. His work includes code porting, debugging, tuning and optimization, and code scaling. Johnny received his PhD in Chemical Physics from the University of Texas at Austin, in 1985. He has published papers in multi-photon dynamics, quantum scattering, path-integral methods, quantum functional sensitivity analysis, and, most recently, weather modeling.

ROBERT T. HOOD. CSC, Inc. NASA Ames Research Center B.A. (1976), University of Virginia M.S. (1979), Ph.D. (1982), Cornell University. After completing his Ph.D., Robert Hood joined the faculty of Rice University. He participated in the R^n and ParaScope research programs, concentrating on debugging issues. After ten years on the Rice faculty, he took a position with Kubota Pacific Computer in 1992, serving for a time as the director of the languages group. In 1993 he joined the contract staff of the Numerical Aerodynamic Simulation (NAS) division at the NASA Ames Research Center and has been there since. Robert Hood's professional interests are in high performance computing systems, including benchmarking, programming tools, and advanced compilation systems. He led the effort to design and build p2d2, a portable, scalable debugger. In addition, he served on the steering committee of the Parallel Tools Consortium and was active in the High Performance Debugger Forum, which sought to standardize aspects of debuggers available on HPC platforms.

Haoqiang H. Jin obtained his Ph.D from the University of Tennessee in 1991. He currently is a senior member of the Applications and Tools Group in NAS Division at NASA Ames Research Center. His expertise includes the development of parallelization and optimization tools for scientific applications. His research interests include performance optimization of parallel applications, parallel programming paradigms, such as those beyond MPI and OpenMP, and parallel benchmarks for characterizing supercomputers.