Performance Comparison of SGI Altix 4700 and SGI Altix 3700 Bx2

Subhash Saini, Dennis C. Jespersen, Dale Talcott,
Jahed Djomehri and Timothy Sandstrom

NASA Advanced Supercomputing Division, NASA Ames Research Center, Moffett Field, California, 94035-1000, USA
{Subhash.Saini, Dennis.C.Jespersen, Dale.R.Talcott, Mohammad.J.Djomehri@nasa.gov, Timothy.A.Sandstrom}@nasa.gov

Abstract

Suitability of the next generation of high-performance computing systems for petascale simulations will depend on a balance between factors such as processor performance, memory performance, local and global network performance, and Input/Output (I/O) performance. As the supercomputing industry develops new technologies for these subsystems, achieving system balance becomes challenging. In this paper, we evaluate the performance of a newly introduced dual-core-based SGI Altix 4700 system and we compare its performance with that of a single-core-based SGI Altix 3700 Bx2 system. We used the High-Performance Computing Challenge (HPCC) benchmarks and five real-world applications, three from computational fluid dynamics, one from climate modeling and one from nanotechnology. Our study shows that the SGI Altix 4700 performs slightly better than the SGI Altix 3700 Bx2 up to 128 processors, while the performance of the systems is almost the same beyond 128 processors, when the communication time dominates the compute time.

1. Introduction

Developing petascale scientific and engineering simulations for difficult large-scale problems is a challenging task for the supercomputing community. Suitability of the next generation of high-performance computing technology for these simulations will depend on a balance between several factors, such as processor performance, memory performance, local and global network performance, and Input/Output (I/O) performance. As new technologies are developed for these subsystems, achieving a balanced system becomes difficult. In light of this, we present an evaluation of a newly introduced SGI Altix 4700 computing system. We use the High-Performance Computing Challenge (HPCC) micro-benchmarks to develop a controlled understanding of individual subsystems and then use this information to analyze and interpret the performance of five real-world applications of interest to NASA. As a baseline, we compare the performance of the SGI Altix 4700 with a previous generation SGI Altix 3700 Bx2 system.

The remainder of this paper is organized as follows: Section 2 details the architectures of the two selected computing systems. Section 3 describes the suite of HPCC benchmarks [1] and the five real-world applications. In Section 4, we present and analyze the results of the benchmarking study. Section 5 summarizes the analysis.

978-1-4244-1694-3/08/$25.00 ©2008 IEEE

2. High-End Computing Platforms

In this section, we describe the SGI Altix Bx2 and SGI Altix 4700 systems.

2.1 SGI Altix 3700 Bx2

The single-core-based system studied is an SGI Altix 3700 Bx2 (hereafter called “Bx2”) [2]. The Bx2 system has global shared memory and is characterized as a cache-coherent Non-Uniform Memory Access (NUMA) computer. It is a single-system image (SSI) machine, where a single memory address space is visible to all of the computing system resources. SSI is achieved through NUMAlink4, a Non-Uniform Memory Access Flexible (NUMAflex) memory interconnect. The Scalable Hub (SHUB) chip implements the global cache coherency protocol. In the SGI Bx2 system, eight Intel Itanium 2 processors and four SHUB application-specific integrated circuits (ASICs) are grouped together into what is called a C-brick. The C-bricks are connected together by a NUMAlink4 interconnect. Each pair of processors shares a peak bandwidth of 6.4 gigabytes per second (GB/s) to local memory. Peak bandwidth between bricks is 1.6 GB/s.

2.2 SGI Altix 4700

The Altix 4700 system (hereafter called “4700”) uses individual rack units (IRUs) instead of C-bricks [3]. Each IRU holds eight processor blades. Each blade contains one dual-core Itanium 2 socket. Physically, the 4700 system consists of eight racks with four IRUs in each rack. Each IRU also contains four routers to connect to the NUMAlink4 network. The Bx2 and 4700 systems have the same clock rate, main memory size, L1 and L3 cache sizes, peak performance rate, and network bandwidth. However, they differ in the processor architecture (dual-core vs. single-core), the front side bus (FSB) frequency, the L2 cache size, and the type and number of the component modules (IRUs vs. C-bricks). Another difference is that the Bx2 C-bricks have eight cores sharing eight connections to the rest of the NUMAlink4 fabric, whereas the 4700 IRUs have sixteen cores sharing eight connections.

The SGI Altix 4700 system was installed at NASA Ames Research Center in January 2007. It consists of 256 dual-core Intel Itanium 2 p9000 series sockets. The 1.6 GHz processors of the 4700 system have 32 KB of L1 cache, 1 MB of L2 instruction cache, 256 KB of L2 data cache, and 9 MB of L3 cache for each core. The FSB, which transports data between the memory and the two cores, runs at 533 MHz for the 4700 system. The processors are interconnected via the NUMAlink4 network with a fat-tree topology and a peak bidirectional bandwidth of 6.4 GB/s. The peak performance of the Altix 4700 system is 3.3 Tflop/s. Characteristics of the two systems are shown in Table 1.

3. Benchmarks and Applications Used

We used the HPCC benchmarks suite [1] and five real-world applications as described below:

3.1 HPC Challenge Benchmarks

The HPCC benchmarks provide multi-faceted, and comprehensive insight into the performance of modern high-end computing systems [1]. These benchmarks stress the processors, the memory subsystem, and system interconnects. They provide a good indication of how a high-end computing system will perform across a wide spectrum of real-world applications. Although application performance is the ultimate measure of system capability, understanding how an application interacts with a computing system requires a detailed performance evaluation of the system components. Four HPCC benchmarks: HPL, PTRANS, STREAM, and FFT, capture important performance characteristics of most real-world applications.

Hostname	**Columbia 20**	**Columbia 21**
Model	Altix 3700 Bx2	Altix 4700
Type of core	Single	Dual
Number of sockets	1	2
Number of sockets used	1	1
Type of core	Itanium-2 (Madison)	Itanium-2 (Montecito)
Core clock frequency (GHz)	1.600	1.594
L1 cache size (KB)	32	32
L2 cache size (KB)	256 (I + D)	1024 (I)+256 (D)
L3 cache size (MB)	9	9
Total memory (GB)	1026.466	979,996
Frequency of FSB (MHz)	400	533
Transfer rate of FSB (GB/s)	6.4	8.5
Building blocks (module)	C-bricks	IRU
Processors/cores per module	8	16
Number of modules	64	32
Maximum number of hops	6	7
Number of links	912	848
Interconnect	NUMAlink4	NUMAlink4
Network topology	Fat tree	Fat tree
Number of routers	224	176
SHub I.D.	1.2	2.0
Installation date	October 2004	January 2007
Installation place	NASA, California	NASA, California

Table 1: System characteristics of Bx2 and 4700 systems.

3.2 Scientific and Engineering Applications

We used the following five real-world applications in our study.

3.2.1 OVERFLOW-2

OVERFLOW-2 is a general purpose Navier-Stokes solver for computational fluid dynamics problems [4]. It is a Fortran90 application, and the MPI version has 130,000 lines of code. The code uses an overset grid methodology to perform high-fidelity, viscous simulations around realistic aerospace configurations. The main computational logic of the sequential code consists of a time loop and a nested grid loop. Within the time loop, solutions to the flow equations are obtained on the individual grids with imposed boundary conditions. Overlapping boundary points, or inter-grid data, are updated from the previous time step using an overset grid interpolation procedure. The code uses finite differences in space with implicit time-stepping. It uses overset-structured grids to accommodate arbitrarily complex moving geometries. The data set used is DLRF6, with 23 zones and 36 million grid points. The input data set is 1.6 GB in size, and the solution file is 2 GB in size.

3.2.2 CART3D

CART3D is a high-fidelity, inviscid application that solves the Euler equations of fluid dynamics [5]. Phenomena like boundary layers, wakes, and other viscous terms are not explicitly accounted for. CART3D includes a solver called *Flowcart*, which uses a second-order, cell-centered, finite volume upwind spatial discretization scheme, in conjunction with a multigrid accelerated Runge-Kutta method for steady-state cases. It is available in both OpenMP and MPI versions. *Flowcart* uses a multigrid method for convergence acceleration, and a domain-decomposition scheme for sub-dividing the global solutions for the governing Euler equations among the processors. The mesh coarsener and mesh partitioner use hierarchical nesting of adaptively refined Cartesian meshes. In this study we used the geometry of the Space Shuttle Launch Vehicle (SSLV) for the CART3D simulations. The SSLV uses 24 million cells for computation. We used both MPI and OpenMP version of the code in the present study.

3.2.3 USM3D

USM3D is a 3D unstructured tetrahedral, cell-centered, finite volume Euler and Navier-Stokes flow solver [6]. Spatial discretization is accomplished using an analytical reconstruction process for computing solution gradients within tetrahedral cells. The solution is advanced in time to a steady state condition by an implicit Euler time-stepping scheme. A single-block, tetrahedral, unstructured grid is partitioned into a user-specified number of contiguous partitions, each containing nearly the same number of grid cells. Grid partitioning is accomplished by the graph partitioning software Metis [10].

Communication among partitions is accomplished by suitably embedded MPI calls to the solver. The test case used a mesh with 10 million tetrahedrons, requiring about 16 GB of memory and 10 GB of disk space.

3.2.4 ECCO

Estimating the Circulation and Climate of the Ocean (ECCO) is a global ocean simulation model solving the fluid equations of motion using the hydrostatic approximation [7]. The model employs a structured, rectilinear, 3D, latitude- and longitude-based mesh. It uses a 2D decomposition in the horizontal direction for parallel implementation. ECCO heavily stresses processor performance, input and output (I/O), and scalability of the interconnect. ECCO performs a large number of short message global operations using the *MPI_Allreduce* function. The ECCO test case uses 50 million grid points, and requires 32 GB of system memory and 20 GB of disk to run. It writes 8 GB of data using Fortran I/O.

3.2.5 NAMD

NAnoscale Molecular Dynamics (NAMD) is an MPI-based parallel molecular dynamics application designed for high-performance simulation of large, complex, bio-molecular systems [8]. NAMD is based on CHARM [9]. NAMD's parallel strategy combines spatial decomposition with force decomposition to enhance scalability. The parallel efficiency of the NAMD program is highly dependent on efficient load distribution. Two types of load balancing are used: initial load balancing and dynamic load balancing. The program uses an irregular data structure for I/O data consisting of particle coordinates and velocities. The test case used consists of 475,202 atoms and is characterized as a compute-intensive application. The input file is 200 MB in size.

4.0 Results

In this section we present results of selected HPC Challenge benchmarks and our application benchmarks.

4.1 HPC Challenge Benchmarks:

In Figure 1 we plot the performance of the compute-intensive global high-performance LINPACK (G-HPL) benchmark for the two systems**.** The performance is nearly the same on both systems as they have the same processor. Both systems have 512 processors, but users can use only 508 of them since four are used as boot processors. A nearly square, 506-processor grid (23 x 22) was used for optimal performance. The performance of G-HPL is about 80% of the theoretical peak performance, which is 3.24 Tflop/s for 506 processors.

In Figure 2 we plot memory bandwidth using the EP-STREAM benchmark. The measured memory bandwidth for the Bx2 is 2 GB/s for 4 to 506 processors, whereas memory bandwidth for the 4700 system is almost a constant 2.66 GB/s up to 128 processors. The 33% difference in performance is consistent with the differing FSB frequencies, which are 400 MHz and 533 MHz for Bx2 and 4700 systems respectively. Both the systems can load two 64 bit words (16 bytes) per FSB clock, giving peak theoretical bandwidths of 6.4 GB/s (400 MHz x 16 bytes) and 8.5 GB/s (533 MHz x 16 bytes) for the Bx2 and 4700 systems respectively. In the 4700 system, the memory bandwidth starts increasing at 256 processors and becomes 3.1 GB/s at 506 processors. We don't understand this increase.

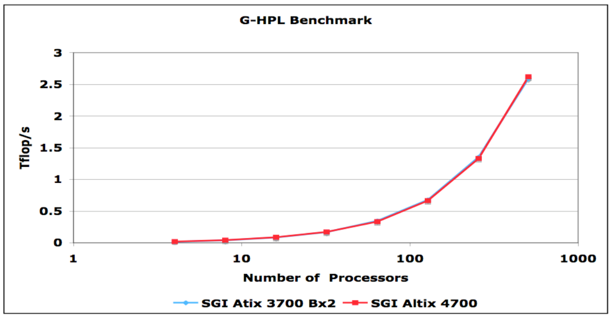

Figure 1: Performance of G-HPL on Bx2 and 4700 systems.

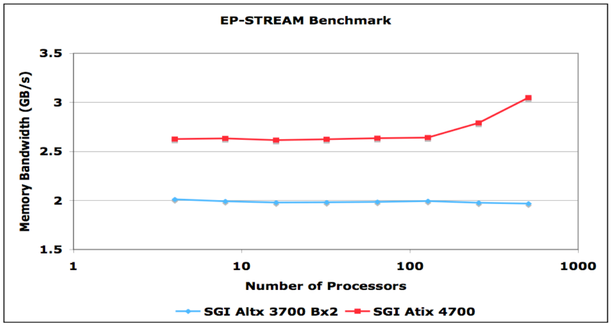

Figure 2: Performance of EP-STREAM on Bx2 and 4700 systems.

In Figure 3 we plot the random-ordered ring latency for 4 to 506 processors. In both systems the latency increases as the number of processors increases. The higher latency in the 4700 system is puzzling. While it is true that the higher core-to-interconnect link ratio of the IRUs compared to the C-bricks should cause more contention in the 4700 than in the Bx2, single packets should not be affected by this. Perhaps getting to the first router is more expensive in a blade architecture than in the C-brick arrangement. In the 4700 system, we don't fully understand the cause of the variation in latency between 16 and 32 processors. One possible reason could be disjoint/discontiguous allocation of processors by the Portable Batch System (PBS), which would entail more network hops.

In Figure 4 we show the random-ordered ring bandwidth for the two systems. For up to eight processors, the bandwidth for the Bx2 system is higher by 36% (at 4 processors) and by 65% (at eight processors) than the 4700 system. For eight processors, communication in the Bx2 system is within a single brick, while communication is across four blades in the 4700, which involves more hardware/software overhead. In the Bx2 system, the bandwidth decreases significantly at 16 processors and then remains constant up to 128 processors, since the communication is then across separate modules. Then the bandwidth again drops significantly for between 128 and 256 processors, where communication is across two racks, and then remains constant up to 506 processors. The 512-processor system comprises four 128-processor double cabinets. Within the cabinets, addresses are de-referenced using the complete pointer. More distant addresses are de-referenced using *coarse mode,* which drops the last few bits of the address. On average, this results in slightly slower communication when addressing more distant memory. In the 4700 system, there is variation in the bandwidth. In both systems, the average bandwidth decreases as the number of processors increases, until it becomes almost the same at 506 processors.

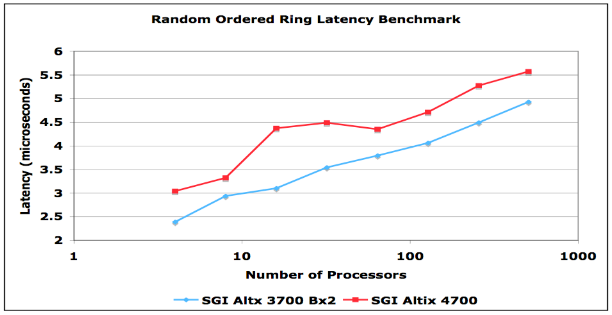

Figure 3: Performance of random-ordered ring latency for Bx2 and 4700 systems.

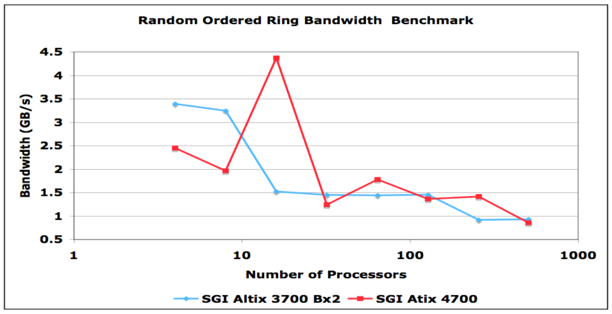

Figure 4: Performance of random-ordered ring bandwidth for Bx2 and 4700 systems.

In Figure 5 we plot the performance of the Random Access benchmark as Giga UPdates per second (GUPS) for 4 to 506 processors. GUPS measures the rate at which a computing system can update the elements of a table spread across global system memory.GUPS profiles the memory architecture of a system and is a measure of performance similar to GFLOPS. In Figure 5 we see that the benchmark scales very well for both the Bx2 and 4700 systems and the performance is the same for both. The poor performance of this benchmark is due to the poor parallel implementation provided by HPCC, which typically performs poorly on distributed-memory systems like Bx2 and 4700 because the updates require numerous, small, point-to-point messages between processors.

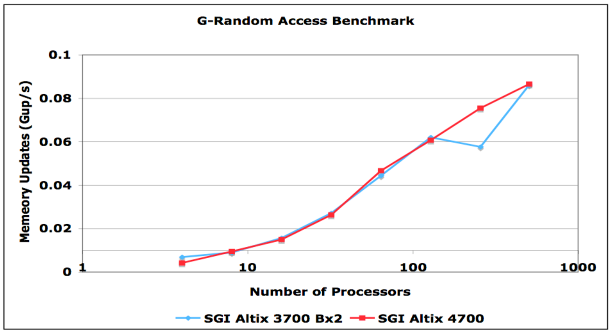

Figure 5: Performance of RandomAccess benchmark for Bx2 and 4700 systems.

Figure 6 shows the performance of the parallel matrix transpose (PTRANS) benchmark. PTRANS exchanges large messages simultaneously between pairs of processors. This benchmark is a useful test to measure the total communications capacity of the system interconnects. The performance of the 4700 is much better than the Bx2 system at 128 processors, even though it has fewer routers (176 compared to 224). For both systems, however, peak aggregate interconnect bandwidth is the same (6.4 GB/s). It should be noted that the performance of PTRANS strongly depends on the configuration of the processes grid. Performance is best when the numbers of communicating pairs are minimum. For example, for a matrix of 9x9, 3x3 processes grid has 3 communicating pairs (2-4, 3-7 and 6-8). However, a 1x9 processes grid has 36 communicating pairs (1-2, 1-3, 1-4, 1-5, 1-6, 1-7, 1-8, 1-9, 2-3, …, 8-9). According to HPCC benchmark rules, only one configuration of a processes grid should be used for the entire benchmarks suite. We used the one, which gives the best performance of G-HPL benchmark. Beyond 128 processors, performance degrades for the 4700 system as network latencies start increasing and network bandwidth starts decreasing. In fact, at 506 processors performance of the Bx2 is better than the 4700. This benchmark uses “all-to-all” communication and therefore stresses the global network.

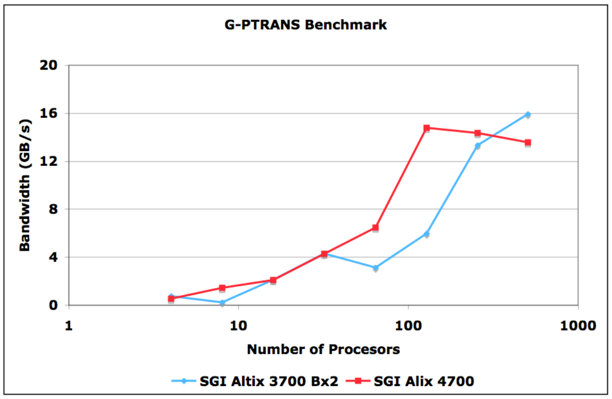

Figure 6: Performance of PTRANS benchmark for Bx2 and 4700 systems.

Figure 7 shows the performance of the G-FFTE benchmark on the Bx2 and 4700 systems for 4 to 506 processors. The G-FFTE benchmark measures the floating-point execution rate of a double-precision, complex, one-dimensional Discrete Fourier Transform. In G-FFTE, since cyclic distribution is used, all-to-all communication takes place only once. The benchmark stresses inter-processor communication of large messages. Both G-FFTE and PTRANS are strongly influenced by the memory bandwidth benchmark (EP STREAM) and the inter-process bandwidth benchmark (random ordered ring). Like PTRANS, G-FFTE also performs parallel two-dimensional transpose of a matrix involving all-to-all communication that stresses the global network. For this reason, the qualitative performance of PTRANS and G-FFT benchmarks is quite similar. Performance of the benchmark up to 64 processors is similar on both systems, since the number of links and routers in each is adequate to handle the all-to-all communication (for 64 processors there are 64x63 pairs of communicating processors). At 506 processors, performance on the 4700 system decreases as interconnect latency increases and interconnect bandwidth decreases.

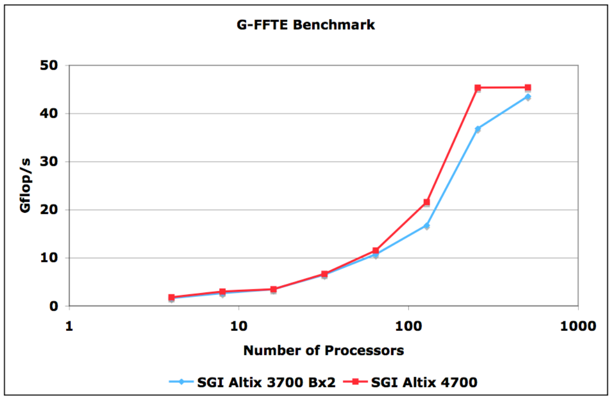

Figure 7: Performance of G-FFTE benchmark for Bx2 and 4700 systems.

4.2 Scientific and Engineering Applications

In the following we present the results and analysis of the five real-world applications on both the Bx2 and 4700 systems. The results and analysis for application OVERFLOW-2 are presented in more detail than the others due to page limits for the paper.

4.2.1 OVERFLOW-2

Figure 8 shows wall-clock time for 8 to 256 processors for the application OVERFLOW-2. The performance of the 4700 system is better than that of the Bx2 system up to 64 processors, while the two systems show similar performance for more processors. The OVERFLOW-2 code is memory-bound and performs better on the 4700 system since its memory bandwidth is 33% better than the Bx2 (2.66 GB/s versus 2 GB/s). To confirm this, we plot the compute time, instead of wall-clock time, as a function of the number of processors in Figure 9.

Qualitatively, Figures 8 and 9 are the same except that the times Fig. 9 are lower than those in Fig. 8, which include both compute time and communication time. Fig. 9 shows that performance of the 4700 system is better than that of Bx2 from 8 to 64 processors, after which the performance becomes almost the same. The reason is that, until 64 processors are used, the data does not fit into the L3 cache and must be fetched from the memory. As a result, the 4700 performance is better than Bx2 since the 4700 memory bandwidth is 33% better. For 128 and 256 processors, the compute time on the two systems is the same since the data fits into the L3 cache.

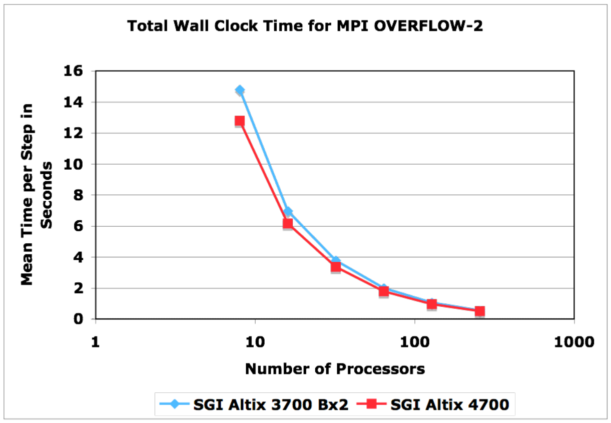

Figure 8: Wall-clock time (compute time + communication time) of the OVERFLOW-2 application for Bx2 and 4700 systems.

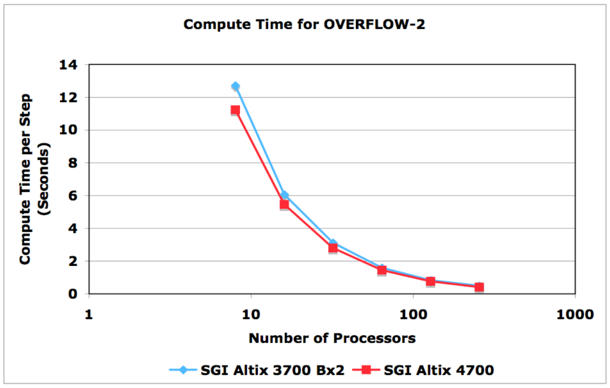

Figure 9: Compute time of the OVERFLOW-2 application for Bx2 and 4700 systems.

Figure 10 shows the communication time for a range of processors. Communication time is lower on the 4700 system than on the Bx2. The difference is largest for 8 processors and decreases as the number of processors increases. For 256 processors, the communication time on the two systems is the same. The explanation for this is that the network bandwidth decreases and network latency increases as the number of processors increases for the 4700 system.

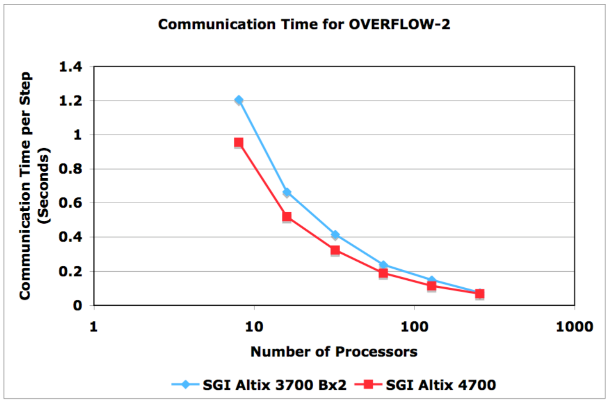

Figure 10: Communication time of the OVERFLOW-2 application for Bx2 and 4700 systems.

Figure 11 shows the number of grid points as a function of the number of processors. The original grid has 35.9 million grid points and the number of grid points increases as the number of processors increases. This increase is due to "ghost" points that are added from splitting zones for load balancing. At 256 processors, the number of grid points has increased 33% to 47.8 million. This suggests that, although wall-clock time is the metric the end user wants to see, wall-clock time per grid point would be a truer measure of the performance of the systems and the application. Therefore, we take the time per step and translate it to nanoseconds per grid point per time step by using the number of grid points actually being used.

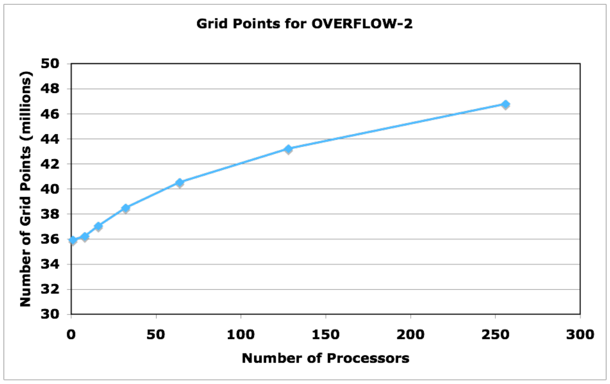

Figure 11: Number of grid points of the OVERFLOW-2 application for Bx2 and 4700 systems.

In Figure 12 we plot nanoseconds per grid point per time step for 8 to 256 processors. Up to 64 processors the performance on Bx2 system is better than 4700 system and beyond 64 processors the performance on both the systems is almost same.

Figure 13 shows the scaling of OVERFLOW-2 for the two systems. Scaling is good over the entire range of processors and is better for a 4700 system because it is a more balanced system than the Bx2. This is because the 4700 has higher memory bandwidth, while processor and network performance are the same.

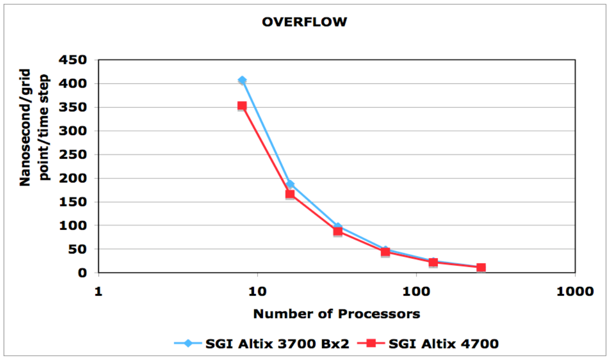

Figure 12: Wall-clock time (compute time + communication time) of the OVERFLOW-2 application for Bx2 and 4700 systems.

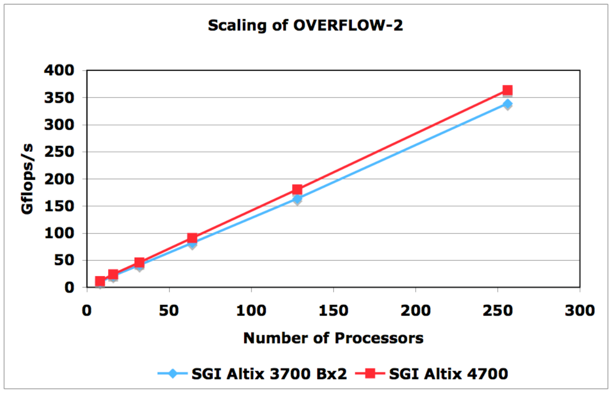

Figure 13: Scaling of the OVERFLOW-2 application for Bx2 and 4700 systems.

Figure 14 shows the sustained percentage of peak performance of OVERFLOW-2. This quantity is about 22% for the 4700 system and 20% for the Bx2. The improvement on the 4700 is due to the better memory bandwidth and the larger L2 cache.

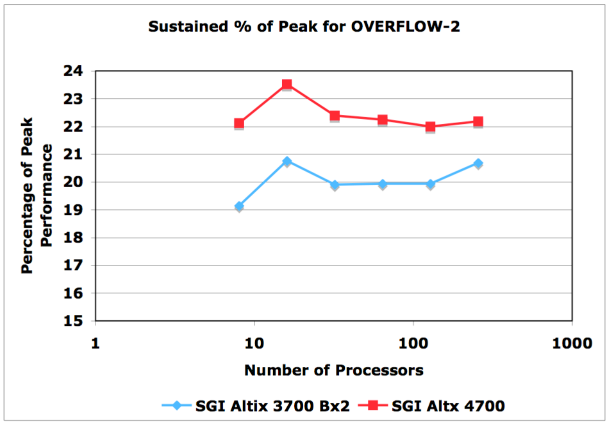

Figure 14: Sustained percentage of peak for OVERFLOW-2 application for Bx2 and 4700 systems

4.2.2 CART3D

In this subsection we present results and analysis of the CART3D application on the Bx2 and 4700 systems. Figure 15 shows time per step for 16 to 506 processors for the MPI version of CART3D.

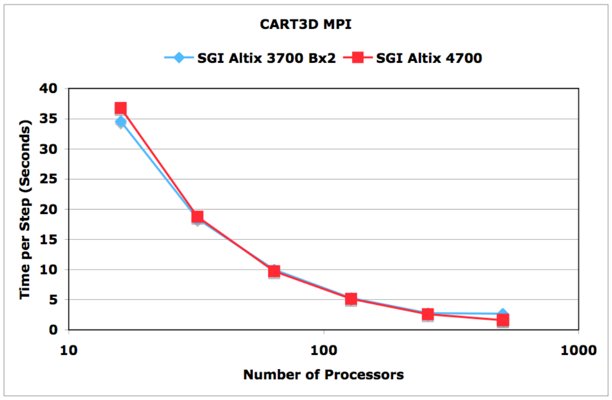

Figure 15: Time per step for application MPI version of CART3D for Bx2 and 4700 systems.

Figure 16 shows the time per step for 16 to 506 processors for the OpenMP version of CART3D. The performance of both versions of CART3D is almost same on both the systems except for 16 processors. The reason for this is that the 4700 has a latency increase for 16 processors (see Figure 3).

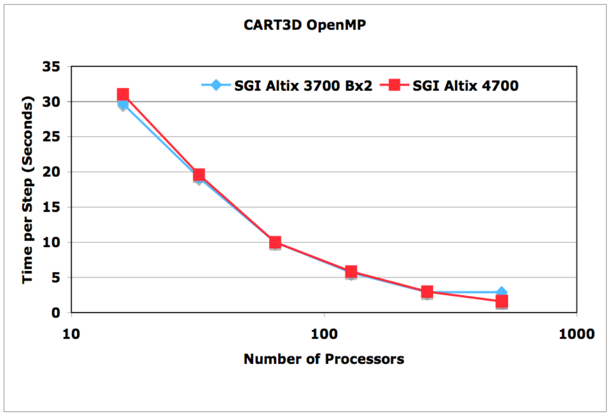

Figure 16: Time per step for application OpenMP version of CART3D for Bx2 and 4700 systems.

4.2.3 USM3D

In this subsection we present the results of the USM3D application. Figure 17 shows wall-clock time per step for USM3D for a range of processors on both the Bx2 and 4700 systems. The performance of USM3D is better on the 4700 than on the Bx2 for the entire range of processors. However, the code does not scale past 128 processors on either system. USM3D is an unstructured grid code, and indirect addressing (for all processor counts), network latency, and bandwidth (for large processor counts) limit its scalability.

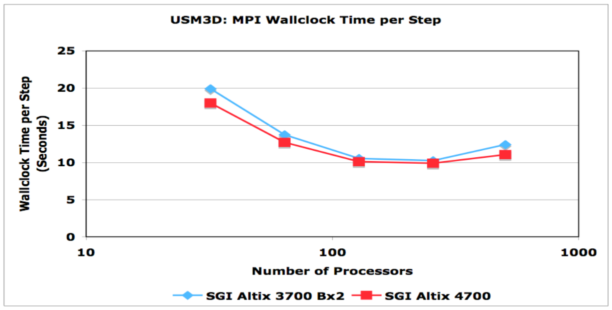

Figure 17: Wall-clock time for USM3D on Bx2 and 4700 systems.

To test the effect of the memory subsystem, we plot the compute time for a range of processors in Figure 18. Performance of USM3D is better on the 4700 system than on the Bx2. The reason for this is that Bx2 has a shared L2 cache (instruction and data) of 256 KB and a memory bandwidth of 2 GB/s, whereas the 4700 has separate L2 instruction and data caches of 1024 KB and 256 KB sizes. Beyond 128 processors, the performance of USM3D is the same on both systems but it does not scale for this test case.

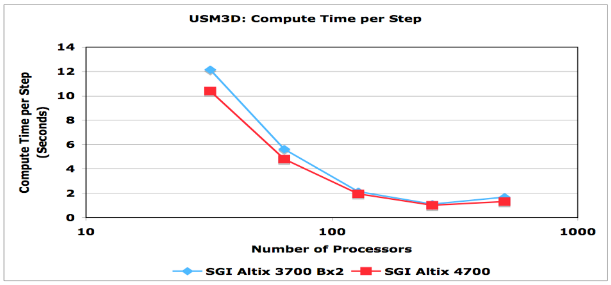

Figure 18: Compute time per step for USM3D on Bx2 and 4700 systems.

In Figure 19 we plot communication time per step for USM3D on both the systems. Communication time on a 4700 system is smaller than on the Bx2 system. As the number of processors increases, the gap between the two systems grows. This gap is expected because the network latency increases and network bandwidth decreases as the number of processors increases, as shown in Figures 3 and 4. This behavior is typical of unstructured codes like USM3D, which send many small messages.

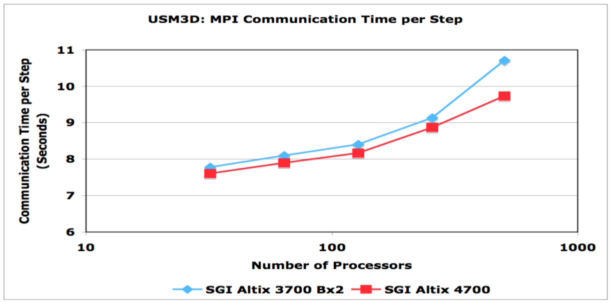

Figure 19: Communication time per step for USM3D on Bx2 and 4700 systems.

4.2.4 ECCO

In Figure 20 we show wall-clock and I/O time for the application ECCO. This code is memory-bound for small processor counts, while its performance for large processor counts depends on the network latency. Since the 4700 system has 33% better memory bandwidth and has a larger, non-shared L2 cache, ECCO performs as well or better on the 4700 than on the Bx2. The performance is better for 32 to 120 processors. For 240 and 480 processors, the performance on the two systems is almost the same, since network latency on both systems increases with the number of processors (see Figure 3). The I/O is sequential, so its time is constant.

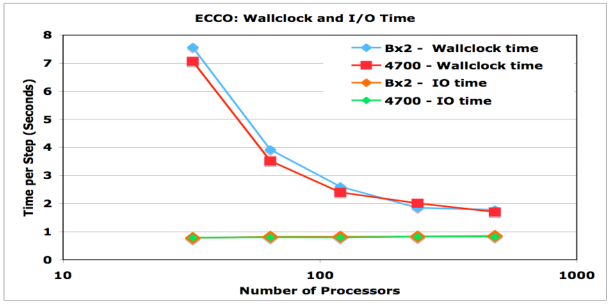

Figure 20: Wall-clock and I/O time for ECCO on Bx2 and 4700 systems.

Figure 21 shows the I/O write bandwidth for ECCO. Average write bandwidth is about 82 MB/s on both systems, being slightly higher on the 4700, and is about 4% of the 2 GB/s peak theoretical value.

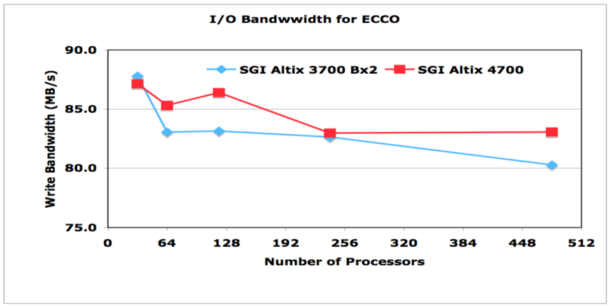

Figure 21: Write I/O bandwidth for ECCO on Bx2 and 4700 systems.

4.2.5 NAMD

In Figure 22 we show wall-clock time for the application NAMD for the Bx2 and 4700 systems. This application is compute-bound for low processor counts and is network latency-bound for higher processor counts. The performance of NAMD is almost the same on both of the systems for processor counts 4 through 64. The reason for this is that both systems have the same processor. A performance gap between the two systems appears at 128 processors and the gap widens for 256 and 508 processors, with performance being better on Bx2 system. The reason for this is that network latency in this range of processors is much higher on the 4700 than on the Bx2 system.

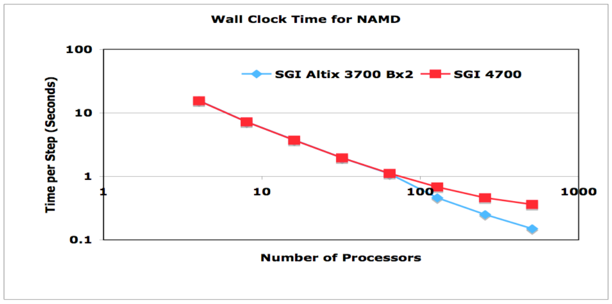

Figure 22: Wall-clock time for ECCO on Bx2 and 4700 systems

5. Conclusions

The measured memory bandwidth of the SGI Altix 4700 is better than that of the SGI Altix 3700 Bx2 (2.66 GB/s vs. 2 GB/s). Compute-bound and memory-bound applications, such as OVERFLOW-2 and CART3D running on small numbers of processors, perform better on SGI Altix 4700 due to its faster FSB (533 MHz vs 400 MHz) and larger, non-shared L2 cache. Interconnect latency-bound applications, such as ECCO and NAMD running on large numbers of processors, perform better on SGI Altix 3700 Bx2. The systems are architecturally different in memory bandwidth, L2 cache, and network latency and bandwidth. Overall the performance difference between the Bx2 and 4700 is marginal. I/O is a bottleneck in an application like ECCO because of Fortran I/O. Performance of OpenMP can be as good as MPI (e.g., CART3D). For consistently good performance on a wide range of processors, a balance between the performances of processor, memory subsystem, and interconnects (both latency and bandwidth) is needed. We plan to extend this study to POWER5+ clusters, the IBM Blue Gene/P, and the Cray XT4.

6. References

1. HPCC, HPC Challenge Benchmarks, URL: http://icl.cs.utk.edu/hpcc/
2. SGI Altix 3700 Bx2 Servers and Supercomputers, URL: http://www.sgi.com/pdfs/3709.pdf
3. SGI Altix 4700, URL: http://www.sgi.com/products/servers/altix/4000/
4. OVERFLOW-2, URL: http://aaac.larc.nasa.gov/~buning/
5. CART3D, URL: http://people.nas.nasa.gov/~aftosmis/cart3d/cart3Dhome.html
6. USM3D, URL: http://aaac.larc.nasa.gov/tsab/usm3d/usm3d_52_man.html
7. ECCO: Estimating the Circulation and Climate of the Ocean, URL: http://www.ecco-group.org/
8. NAMD, URL: http://www.ks.uiuc.edu/Research/namd/
9. CHARM: URL: http://charm.cs.uiuc.edu/
10. METIS: URL: http://glaros.dtc.umn.edu/gkhome/views/metis/