

Early Performance Evaluation of a “Nehalem” Cluster Using Scientific and Engineering Applications

Subhash Saini,¹ Andrey Narakin,² Rupak Biswas,¹ David Barkai,³ and Timothy Sandstrom¹

¹ NASA Ames Research
Moffett Field, CA 94035 USA
{[Subhash.Saini](mailto:Subhash.Saini@nasa.gov), [Rupak.Biswas](mailto:Rupak.Biswas@nasa.gov),
[Timothy.A.Sandstrom](mailto:Timothy.A.Sandstrom@nasa.gov)}@nasa.gov

² Intel Corporation
30 Turgenev str
Nizhny Novgorod, Russia
Andrey.Narakin@intel.com

³ Intel Corporation
2111 NE 25th Avenue
Hillsboro, OR 97124 USA
David.Barkai@intel.com

ABSTRACT

In this paper, we present an early performance evaluation of a 624-core cluster based on the Intel® Xeon® Processor 5560 (code named “Nehalem-EP”, and referred to as Xeon 5560 in this paper)—the third-generation quad-core architecture from Intel. This is the first processor from Intel with a non-uniform memory access (NUMA) architecture managed by on-chip integrated memory controller. It employs a point-to-point interconnect called the Intel® QuickPath Interconnect (QPI) between processors and to the input/output (I/O) hub. It also introduces to a quad-core architecture both Intel’s hyper-threading technology (or simultaneous multi-threading, “SMT”) and Intel® Turbo Boost Technology (“Turbo mode”) that automatically allow processor cores to run faster than the base operating frequency if the processor is operating below rated power, temperature, and current specification limits. It can be engaged with any number of cores or logical processors enabled and active. We critically evaluate these features using the High Performance Computing Challenge (HPCC) benchmarks, NAS Parallel Benchmarks (NPB), and four full-scale scientific applications. We compare and contrast the results of a cluster based on the Xeon 5560 with an SGI® Altix® ICE 8200EX cluster of quad-core Intel® Xeon® 5472 Processor (“Xeon 5472” from here on) and another cluster of Intel® Xeon® 5462 Processor (“Xeon 5462”; the Xeon 5400 Series Processors are previous generation quad-core Intel processors and were code named Harpertown).

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – benchmarking, evaluation/methodology.

General Terms

Measurement, Performance, and Experimentation.

1. INTRODUCTION

Even when run on supercomputers, many applications are unable to achieve sustained performance of more than 10 percent of peak. The primary obstacle is memory bandwidth. The “memory wall” bottleneck prevents memory from feeding the cores commensurate with the floating-point power of the cores.

(c) 2009 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

SC09 November 14-20, 2009, Portland, Oregon, USA
(c) 2009 ACM 978-1-60558-744-8/09/11... \$10.00

Problems with memory latency, memory bandwidth, and smaller caches per core will intensify in future supercomputers that use much more complex memory hierarchies. Long-term research is being done to address the “memory wall” problem [1]. The Xeon 5560 (Nehalem-EP) offers some important initial steps toward ameliorating the memory bandwidth problem. This processor has been used to build a 624-core cluster with Quadruple Data Rate (QDR) InfiniBand (IB) in a fat-tree topology [2–3]. The architecture has overcome problems associated with the sharing of the front side bus (FSB) in previous processor generations by integrating an on-chip memory controller and by connecting the two processors through the Intel® QuickPath Interconnect (QPI). The result is more than three times greater sustained-memory bandwidth per core than the previous-generation dual-socket architecture [4].

The present study uses low-level High Performance Computing Challenge (HPCC) benchmarks to measure processor, memory, and network performance at the subsystem level on an HPC cluster based on the Xeon 5560. The results are used to gain insight into the performance of the NAS Parallel Benchmarks (NPB) and four production quality applications (OVERFLOW-2, ECCO, USM3D, and CART3D). Engineers at NASA and the aerospace industry use these applications extensively.

To the best of our knowledge, this is first paper to conduct:

(a) Critical and extensive performance evaluation and characterization of a cluster based on the Xeon 5560, hereafter called “Discovery”, using HPCC suite, NPB, and four real-world production-quality scientific and engineering applications taken from the existing workload of NASA and U.S. aerospace industry. However, Barker et al. [5] has conducted performance evaluation of the desktop version of Xeon 5500 processor (when still referred to as Nehalem-EP) using U.S. Department of Energy (DoE) workload;

(b) Detailed comparison of Discovery (the Xeon 5560–based cluster) with two other clusters: One based on the Xeon 5472, an SGI ICE 8200EX with IB-connected hypercube topology (hereafter called “ICE”), and another, a Xeon 5462 cluster with IB-connected fat tree topology (hereafter called “Endeavor”). Note however, that Saini et al. [6] conducted the performance evaluation of the SGI Altix ICE 8200, an earlier generation of SGI ICE 8200EX;

(c) Performance analysis of comparison between Double Data Rate (DDR) IB with QDR IB in a Xeon 5560–based cluster;

- (d) Performance comparisons between hypercube and fat-tree topologies of two similar clusters using a real workload;
- (e) Performance evaluation of hyper-threading (or simultaneous multi-threading, “SMT”) using NPB and full-scale applications;
- (f) Performance evaluation of Turbo mode in the Xeon 5560’s architecture using HPCC and NPB;
- (g) Performance comparison of DDR3-1333 and DDR3-1066 memory; and
- (h) Performance analysis of multi-core effects in half-subscribed mode (using two cores of each socket) and full-subscribed mode (using all the cores) for SGI Altix 8200EX and Intel clusters.

The remainder of this paper is organized as follows: Section 2 details the architectures of the SGI Altix ICE 8200EX cluster, Intel Xeon 5462 cluster, and Intel Xeon 5560 cluster. Section 3 describes the suite of HPCC benchmarks, the NPB, three real-world production-quality computational fluid dynamics (CFD) applications (OVERFLOW-2, USM3D, and CART3D), and one full-scale climate-modeling application (ECCO). Section 4 presents and analyzes results from running these benchmarks and applications on the various clusters. Section 5 contains a summary and conclusions of the study. Sections 6 and 7 contain acknowledgements and references respectively.

2. High-End Computing Platforms

2.1 Altix ICE 8200EX Cluster

The ICE cluster uses the Xeon 5472 [7]. A node based on this architecture has two processors with four cores each. Each of the two processors is clocked at 3.0 GHz, with a peak performance of 48 Gflop/s per chip. Peak performance of the node is therefore 96 Gflop/s. Key features include 32 KB L1 instruction cache and 32 KB L1 data cache per core, and 6 MB shared L2 cache per die (12 MB total L2 cache per chip). Each socket of the compute node has 400 MHz quad-pumped bus and 8 GB fully buffered (FB) dual in-line memory module (DIMM) (double data rate 2) DDR2-800MHz memory. This configuration can produce 12.8 GB/s peak-memory bandwidth per socket and twice that per node. The ICE system uses high-speed 4 x DDR IB interconnects [8]. Each Individual Rack Unit (IRU) includes two switchblades, eliminating external switches altogether. The fabric connects the service nodes, leader nodes, and compute nodes. There are two IB fabrics on the ICE system—one for MPI (ib0), the other for I/O (ib1). Tests were run with the vendor MPI library (MPT) [9]. The nodes are connected in a hypercube topology using IB and use the Linux operating system. Altix ICE systems are ranked 4th, 17th, and 20th in June 2009 TOP500 list [10].

2.2 The Endeavor Cluster

The Endeavor cluster uses a Xeon 5462 processor [11]. A node based on this architecture has two processors with four cores each. Each of the two processors is clocked at 2.8 GHz, with a peak performance of 44.8 Gflop/s per chip. Peak performance of the node is therefore 89.6 Gflop/s. Each socket of the compute node has 400 MHz quad-pumped bus and 16 GB FBDIMM DDR2-667MHz memory. This configuration produces 10.67 GB/s peak-memory bandwidth per socket and twice of that per node. Unlike ICE, the Endeavor nodes are connected in a fat-tree topology using DDR IB. The cluster uses the Linux operating system (Red

Hat EL4) and a single switch, Cisco SFS 7024D DDR, with 288 ports.

2.3 The Discovery Cluster

Discovery, the Xeon 5500 processor cluster, is the first server implementation of a new 64-bit microarchitecture [3]. A node based on this architecture has two processors with four cores each. Each of the two processors is clocked at 2.8 GHz, with a peak performance of 44.8 Gflop/s per chip. Peak performance of the node is therefore 89.6 Gflop/s. Figure 1 shows, schematically a Intel Xeon 5560 processor. The processor has two parts: core and uncore. The core part has four cores plus L1 and L2 caches. Uncore has an L3 cache, integrated memory controller, and Quick Path Interconnect (QPI). The processor has four cores, each with 64 KB of L1 cache (32 KB data and 32 KB instruction). Each core has 256 KB of L2 cache. All four cores share 8 MB of L3 cache. The architecture has an on-chip memory controller, which supports three DDR3 memory channels. The node has a total memory of 18 GB DDR3-1066 MHz: 2 DIMMS (2 GB + 1 GB) per each channel; the 2 GB DIMM is closer to the processor. Later in the study the nodes were upgraded to DDR3-1333 MHz (6x4GB DIMMs per node). Peak-memory bandwidth per socket is 25.584 GB/s and 31.992 GB/s for DDR3-1066 and DDR3-1333, respectively, and twice of that per node. Each processor chip has two interconnect links called QPI. One QPI link connects the two processors of the node to form a non-uniform-memory access (NUMA architecture), the other connects to the IO hub [4]. The QPI link runs at 6.4GT/s (“T” for transactions), at which rate 2 bytes can be transferred in each direction – for a rate of 12.8 GB/s in each direction per QPI link.

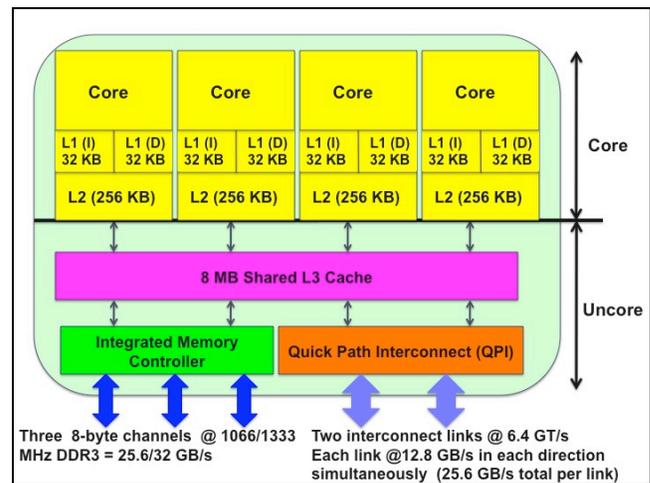


Figure 1: Intel Xeon quad-core Xeon 5560 node.

The Xeon 5560 includes several new elements that help to improve application performance [3]. **Turbo mode** provides a frequency-stepping mode that enables the processor frequency to be increased in increments of 133 MHz. The amount of Turbo boost available varies with processor bin. The Intel Xeon 5560 processor can turbo up to three frequency increments in less than half-subscribed mode—that is, for less than two cores per chip busy, the frequency can go up by 3 x 133 MHz and by two bin splits in half-subscribed to fully subscribed mode (2 x 133 MHz). The frequency is stepped up within the power, current, and thermal constraints of the processor. **Intel’s hyper-threading technology** enables two threads to execute on each core to hide latencies related to data access. Two threads can execute

simultaneously, filling each other's unused stages in the functional unit pipelines. The new micro-architecture also includes new instructions and improved performance of unaligned loads, which are available through Intel software development tools such as the Intel's C and Fortran compilers and the Intel's Math Kernel Library (MKL).

Figure 2 depicts the fat-tree topology of 78 nodes (624 cores) based on the Xeon 5560 connected using QDR IB. The cluster has a two-tier structure consisting of nine 36-port switches: six leafs and three spine switches. Eighteen compute nodes are connected to each of five leaf switches. The other ports of these leafs are used to connect to spine switches. There are six links from each leaf to each spine. The Discovery cluster uses a Lustre file system. Eight parallel file system nodes are connected to the sixth leaf switch. Like the other five leafs, this one has six links to each spine switch.

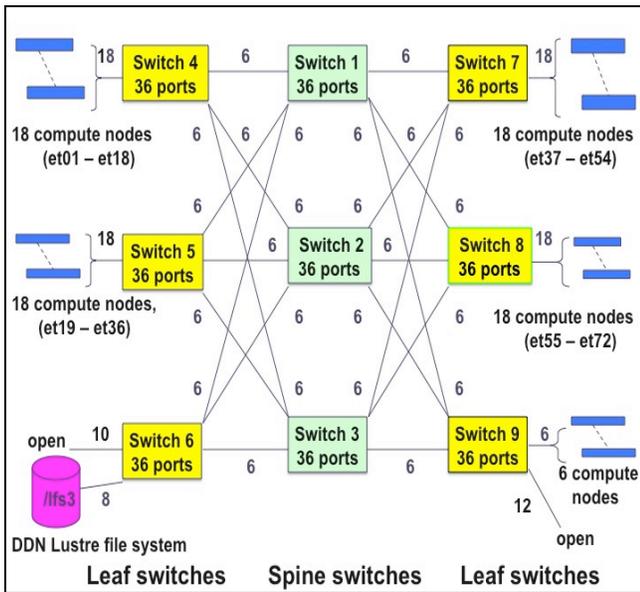


Figure 2: Fat-tree topology of the Discovery cluster using QDR InfiniBand.

System characteristics of the clusters used are given in Table I.

TABLE I. SYSTEM CHARACTERISTICS OF THE CLUSTERS USED.

Model	SGI Altix ICE 8200EX	Intel Endeavor Cluster	Intel Discovery Cluster
Processor type	Intel Xeon processor 5400 series	Intel Xeon processor 5400 series	Intel Xeon processor 5500 series
Processor generation	Second	Second	Third
Processor description	Quad-core Intel Xeon E5472	Quad-core Intel Xeon E5462	Quad-Core Intel Xeon X5560
Technology (nm)	45	45	45
Intel® Turbo Boost technology	no	no	yes
Sockets per node	2	2	2
Cores per socket	4	4	4

No. of cores/node	8	8	8
Core clock frequency (GHz)	3.0	2.8	2.8
Floating point/clock/core	4	4	4
Peak performance/node (Gflop/s)	96	89.6	89.6
L1 cache size	32 KB I + 32 KB D on chip per core	32 KB I + 32 KB D on chip per core	32 KB I + 32KB D on chip per core
L2 cache size	12 MB I+D, 6 MB shared/2 cores	12 MB I+D, 6 MB shared/2 cores	256 KB/core (I+D)
L3 cache size	N/A	N/A	8.192 MB (I+D)
Local memory/node (GB)	8	16	18/24
Total memory on 64 nodes (GB)	512	1024	1152/1536
Memory type	2 channels FBDIMM	2 channels FBDIMM	3 channels DDR3 2 DIMMS per channel
Memory speed (MHz)	800	667	1066/1333
Memory controllers	1	1	2
Memory cont. on chip	no	no	yes
Peak memory transfer rate (GB/s) per socket	12.8	10.67	25.584/32
Front side bus (FSB)	yes	yes	no
Hyper-threading (SMT)	no	no	yes
Number of threads/core	1	1	2
Intel QuickPath Interconnect	no	no	yes
Intel's Turbo Boost Technology	no	no	yes
Interconnect type	DDR IB ConnectX	DDR IB ConnectX	QDR IB ConnectX
Network topology	Hypercube	Fat tree	Fat tree
Operating system	Linux SLES 10	Red Hat EL4 Update 4	Red Hat EL 5.2, kernel 2.6.18-53
Intel Fortran and C compiler	11.0.083	Mult. versions	Mult. versions
System manufacturer	SGI	Intel	Intel
Interconnect manufacturer	Mellanox	Mellanox	Mellanox
MKL Library	10.0.011	Mult. versions	Mult. versions
MPI	SGI mpt-1.23	Intel® MPI 3.2	Intel MPI 3.2
Page sizes	4 KB	4 KB	4 KB
File system	Lustre	Lustre, Panasas, NFS	DDN Lustre, Panasas, NFS
System name	ICE	Endeavor	Discovery

3. Benchmarks and Applications Used

Our evaluation approach recognizes that application performance is the ultimate measure of system capability. However, understanding an application's interaction with a computing system requires a detailed comprehension of individual component performance of the system. Keeping this in mind, we used low-level HPCC benchmarks that measure processor, memory, and network performance of the architectures at the subsystem level. We then use the insights gained from the HPCC

benchmarks to guide and interpret performance analysis of the NPB and four full-scale applications.

3.1 HPC Challenge Benchmarks

The HPC Challenge Benchmarks are intended to test a variety of attributes that can provide insight into the performance of high-end computing systems. These benchmarks examine not only the processors but also the memory subsystem and system interconnects [12].

EP-DGEMM: The embarrassingly parallel (EP) DGEMM measures the floating-point rate of execution of double-precision real matrix-matrix multiplication performed by DGEMM subroutine from Basic Linear Algebra Subroutines (BLAS). All cores execute the benchmark simultaneously. It measures contention in the memory subsystem for floating-point intensive computations.

EP-Stream: The embarrassingly parallel STREAM benchmark is a synthetic program that measures sustainable memory bandwidth. All computational cores execute the benchmark simultaneously, and the arithmetic average is reported. This benchmark measures performance of a memory subsystem.

G-HPL: The High-Performance LINPACK benchmark measures system performance when solving a dense linear equation system. LINPACK is the basis of the Top500 list [10].

G-PTRANS: The parallel-matrix transpose benchmark measures the rate of transfer for large arrays of data from a multiprocessor's memory over the network. It exchanges messages simultaneously between pairs of cores, and is a useful test for measuring total communication capacity of system interconnects. Its performance strongly depends on configuration of the process grid and to lesser extent on memory bandwidth.

G-FFTE: It measures the floating-point rate of execution of double-precision complex one-dimensional Discrete Fourier Transform (DFT). It performs the FFT operation across the entire system by distributing the input vector in block fashion across all nodes.

G-Random Access: Giga-Updates per second (GUP/s) measures the rate at which a system can update individual elements of a table spread across the global system memory. GUP/s profiles the memory architecture of a system and is a measure of performance similar to Gflop/s. This benchmark also measures system performance and uses at least half of the total memory.

Random Order Ring Bandwidth: The Random Ordered Ring Bandwidth benchmark reports bandwidth achieved per core in a ring communication pattern. Communicating nodes are ordered randomly in the ring. The result (in GB/s) per core is averaged over various random assignments of rings; that is, various permutations of the sequence of all cores in the communicator. It measures contention in the network.

Random Ring Latency: The Random Ordered Ring Latency benchmark reports latency (in microseconds) in a ring communication pattern. The communicating nodes are ordered randomly in the ring and the result is averaged over various random rings.

3.2 NAS Parallel Benchmarks

The NPB suite contains eight benchmarks comprising five kernels (CG, FT, EP, MG, and IS) and three compact applications (BT, LU, and SP). We used NPB version 3.3 Class C in our study [13].

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 3-D partial differential equation is solved using Fast Fourier Transforms (FFTs). MG calculates the solution to a 3-D discrete Poisson equation using the V-cycle multigrid method. The MG benchmark has highly structured short- and long-distance communications.

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 by the NASA CFD code INS3D. 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 tri-diagonal equations with a 5x5 block size. Both SP and BT are typified at NASA by the ARC3D CFD code.

3.3 Science and Engineering Applications

For this study, we used four production applications that were taken from NASA's workload.

3.3.1 OVERFLOW-2

OVERFLOW-2 is a general-purpose Navier-Stokes solver for CFD problems [14]. The MPI version, a Fortran90 application, 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. The code uses an overset grid methodology to perform high-fidelity viscous simulations around realistic aerospace configurations. The code uses finite differences in space with implicit time stepping. It uses overset-structured grids to accommodate arbitrarily complex moving geometries. The dataset used is a wing-body-nacelle-pylon geometry (DLRF6), with 23 zones and 36 million grid points. The input dataset is 1.6 GB in size, and the solution file is 2 GB.

3.3.2 CART3D

CART3D is a high-fidelity, inviscid CFD application that solves the Euler equations of fluid dynamics [15]. It includes a solver called Flowcart, which uses a second-order, cell-centered, finite-volume upwind spatial discretization scheme, in conjunction with a multi-grid accelerated Runge-Kutta method for steady-state cases. In this study, we used the geometry of the Space Shuttle Launch Vehicle (SSLV) for the simulations. The SSLV uses 24 million cells for computation, and the input dataset is 1.8 GB. The application (in this case, the MPI version) requires 16 GB of memory to run.

3.3.3 USM3D

USM3D is a 3-D unstructured tetrahedral, cell-centered, finite-volume Euler and Navier-Stokes flow solver [16]. 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 [17]. The test case used 10 million tetrahedral meshes, requiring about 16 GB of memory and 10 GB of disk space.

3.3.4 ECCO

Estimating the Circulation and Climate of the Ocean (ECCO) is a global ocean simulation model for solving the fluid equations of motion using the hydrostatic approximation [18]. ECCO heavily stresses processor performance, I/O, and interconnect scalability. 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. The test case is a 1/4 degree global ocean simulation with a simulated elapsed time of two days.

4. Results

In this section, we present performance results of selected HPCC benchmarks, NPB version 3.3, and application codes. We used the Intel MPI Library [19] for Endeavor and Discovery and MPT for ICE [9].

4.1 HPCC Benchmarks

In this section we present results for HPCC benchmarks for three systems. In Figure 3, we plot performance of the compute-intensive embarrassingly parallel (EP) DGEMM (matrix-matrix multiplication) for the three systems. ICE has the highest theoretical one-core peak of 12.0 Gflop/s. The Endeavor and Discovery clusters have theoretical peak performance of 11.2 Gflop/s. When using Turbo mode on the Discovery cluster, the processor core frequency can be increased by up to two 133 MHz increments, raising its peak to 12.24 Gflop/s—slightly higher than that of ICE. Discovery has a memory subsystem fast enough to enable performance almost independent of the number of cores. For the ICE and Endeavor systems, however, performance for one core and then for four cores was higher than that of fully subscribed mode. For four core runs on ICE and Endeavor, one core from each die is used, effectively doubling the memory bandwidth available for each process. Overall, the Discovery performance with Turbo On was the best, followed by ICE, Discovery with Turbo Off, and Endeavor. The achieved performance was 92, 91, and 95 percent of the peak on ICE, Endeavor, and Discovery with Turbo Off, respectively. For Discovery with Turbo On, the efficiency computation is difficult to define since the core frequency varies over time.

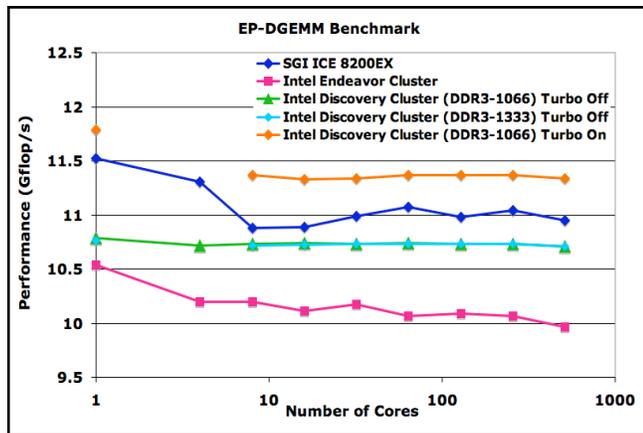


Figure. 3 Performance of EP-DGEMM on ICE, Endeavor, and Discovery.

In Figure 4, we plot performance of the compute-intensive global high-performance LINPACK (G-HPL) benchmark. For Discovery with Turbo On, we give the efficiency for its base frequency of 2.8 GHz, even though it is clear that the frequency was higher during the runs. Similarly to EP-DGEMM, the efficiency of this cluster was higher due to faster memory subsystem. ICE showed the worst efficiency primarily due to the smaller amount of memory per core.

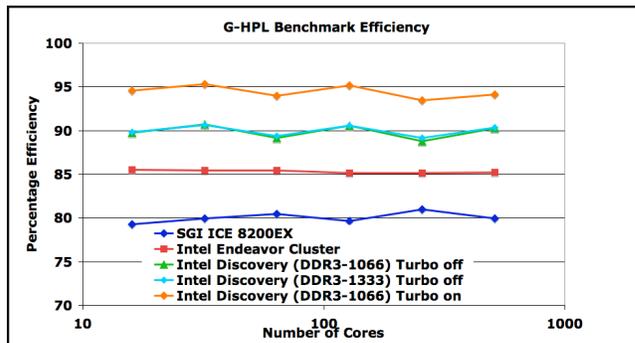


Figure 4. Performance of G-HPL on ICE, Endeavor, and Discovery.

In Figure 5, we show memory bandwidth for each system using the EP-Stream Triad benchmark. For a single core, the measured bandwidths were 4.8 GB/s, 4.3 GB/s, and 11.5 GB/s for ICE, Endeavor, and Discovery, respectively. For four cores, these values decreased to 2.5 GB/s (factor of 1.9 decrease), 2.3 GB/s (factor of 1.9), and 7.7 GB/s (factor of 1.5) due to memory contention. In fully subscribed mode (8 to 512 cores), the average measured memory bandwidth for ICE and Endeavor was 1.23 GB/s (factors of 3.9 and 3.5 decrease respectively), 4.2 GB/s (factor of 2.7) for Discovery with DDR3-1066, and 4.725 GB/s with DDR3-1333. The aggregate node level bandwidth in fully subscribed mode was then $1.23 \times 8 = 9.84$ GB/s for ICE and Endeavor, 33.6 GB/s (DDR3-1066) and 37.8 GB/s (DDR3-1333) for Discovery. This translates into 38, 46, 66, and 59 percent of peak-memory bandwidth for these four cases. The integrated memory controller and QPI enable Discovery to deliver both higher peak-memory bandwidth and efficiency, producing significant advantages for memory-intensive codes.

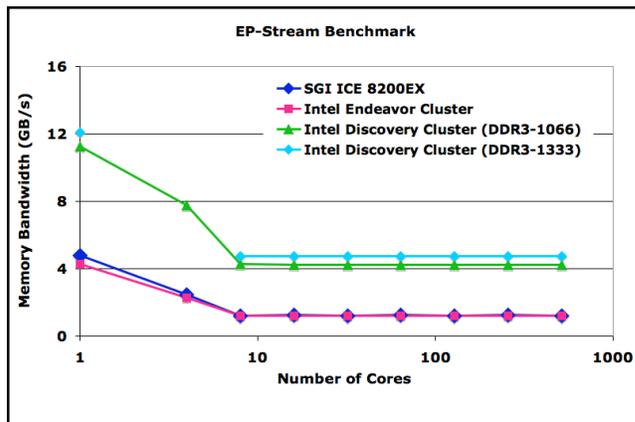


Figure 5. Performance of EP-STREAM on ICE, Endeavor, and Discovery.

By dividing GB/s by Gflop/s, we can determine how many bytes of memory bandwidth are available for each floating-point

operation, as measured by EP-Stream Triad and EP-DGEMM. For ICE, Endeavor, and Discovery, the bytes/flops value is 0.11, 0.12, 0.40 (DDR3-1066), and 0.44 (DDR3-1333), respectively. From this perspective, Discovery is a more balanced system.

In Figure 6, we plot the random-ordered ring (ROR) latency for 4–512 cores for the three systems. Within a node (eight cores), latency of all three systems was about 1 μ s (1.08 μ s for ICE and 0.7 μ s for Endeavor and Discovery). Those values increased when the communication went through the IB layer. For ICE, the rate of increase was faster because of the overhead incurred in going from one IRU (128 cores) to another over the hypercube topology. At 512 cores, latency was 9.1 μ s for ICE while only 6.2 μ s for fat-tree-based Endeavor and Discovery.

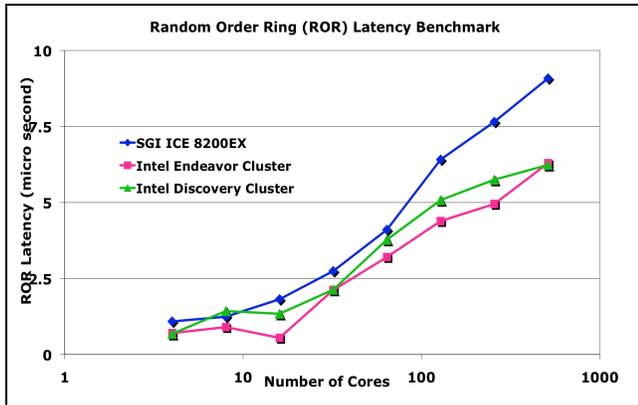


Figure 6. Performance of ROR latency on ICE, Endeavor, and Discovery.

In Figure 7, we show the ROR bandwidth for the three systems. Up to 64 cores, ROR bandwidth of both ICE and Endeavor was almost same. But that bandwidth was much lower than that of Discovery. This system can deliver higher bandwidth primarily because it uses a QDR communication network as opposed to the DDR used by the other two systems. Beyond 64 cores, the bandwidth gap between ICE and the other two systems became constant, whereas it decreases drastically in a hypercube topology, as is indicated from 64 to 512 cores. At 512 cores, ROR bandwidth for ICE, Endeavor, and Discovery was 270 MB/s, 105 MB/s, and 40 MB/s, respectively. The bandwidth on Endeavor was better by a factor of 2.6 than ICE due to topology, while Discovery was ahead of Endeavor by more than two times due to QDR IB.

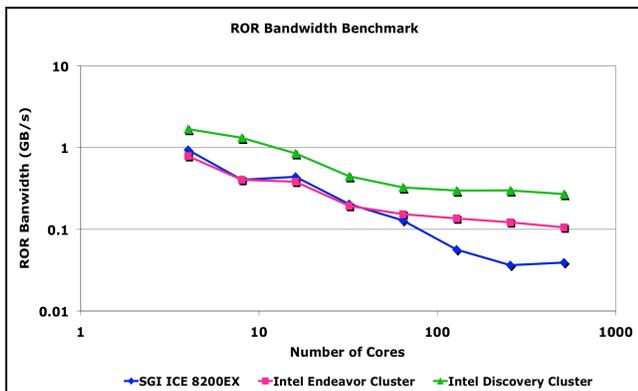


Figure 7. Performance of ROR bandwidth on ICE, Endeavor, and Discovery.

In Figure 8, we plot performance of the Random Access benchmark using SANDIA_OPT2 algorithm as Giga Updates per second (GUP/s) for 4–512 cores for all three systems. Up to 64 cores, performance on ICE and Endeavor was almost identical. However, the performance gap appeared thereafter, and at 512 cores, it was 0.81 GUP/s for ICE and 1.21 GUP/s (that is, 33 percent better on Endeavor due to the network topology). Performance was much better on Discovery than on ICE or Endeavor. The superior performance is due to the QDR IB and higher memory bandwidth. At 512 cores, the result was 2.8 GUP/s. Scaling is very good on Endeavor and Discovery because of the constant bisection bandwidth of the fat-tree topology used in these two systems.

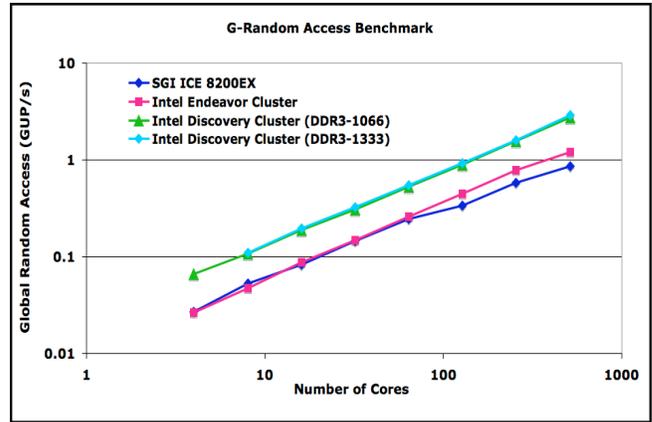


Figure 8. Performance of GUP on ICE, Endeavor, and Discovery.

In Figure 9, we plot performance of the PTRANS benchmark for all three systems. The benchmark performance primarily depends on the network and to a lesser extent on memory bandwidth. Like Random Access performance, the performance of PTRANS on ICE and Endeavor was almost the same up to 64 cores, and then ICE lagged. At 512 cores, it was 14 GB/s for ICE and 36 GB/s (2.6 times better) on Endeavor. Performance was much better on Discovery than on ICE or Endeavor due to the use of QDR IB and higher sustained-memory bandwidth. Use of DDR3-1333 on Discovery delivered an additional 3–12 percent improvement depending on core count. Scaling of the benchmark was very good on Endeavor and Discovery because of the constant bisection bandwidth on these two systems. At 512 cores, bandwidth was 94 GB/s on Discovery.

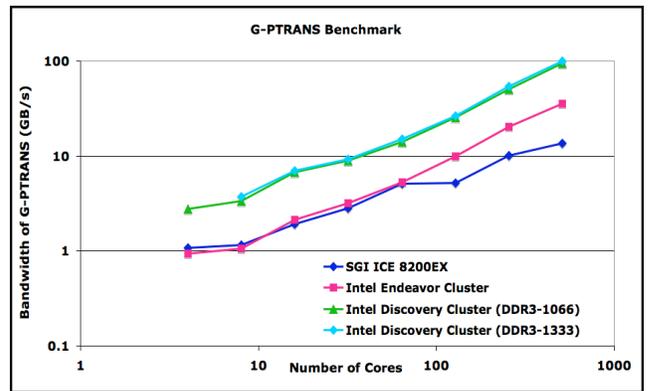


Figure 9. Performance of PTRANS on ICE, Endeavor, and Discovery.

In Figure 10, we plot performance of the G-FFT benchmark on the ICE, Endeavor, and Discovery. The latter two systems used MKL's FFTW2.1.5 interfaces [20-21], while both (HPCC default) FFTE and MKL were used on ICE. The difference due to MKL FFTs was significant—for example, on ICE at 64 cores, it resulted in 1.9 times improvement (18.5 Gflop/s vs. 9.7 Gflop/s). The benchmark's performance depends on a combination of flops, memory, and network bandwidth. The QDR IB and higher sustained-memory bandwidth enable Discovery to outperform Endeavor. Scaling was good on both fat-tree-based systems. However, for ICE (hypercube topology), scaling was good only up to 64 cores (one IRU) and then worsened (especially for FFTE) because of degrading bisection bandwidth, a typical characteristic of hypercube topology. At 512 cores, performance was 22.7, 82.5, 93.7, and 270.2 Gflop/s on ICE with FFTE, ICE with MKL, Endeavor, and Discovery, respectively. Use of DDR3-1333 delivered an additional 3–20 percent improvement, depending on core count.

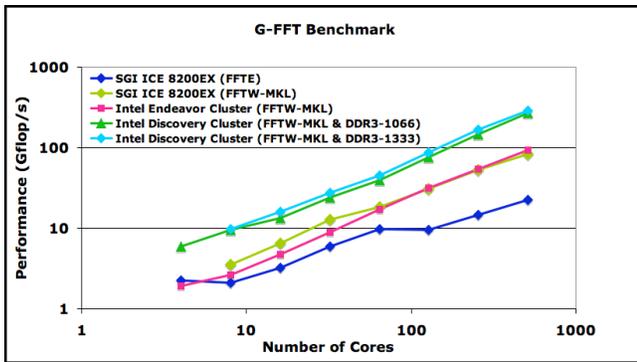


Figure 10. Performance of G-FFT on ICE, Endeavor, and Discovery.

4.2 DDR IB versus QDR IB

In this subsection, we compare performance of DDR IB and QDR IB for Discovery to investigate parameters such as interconnect latency and bandwidth and other HPCC subtests, which depend on those. DDR IB measurements were made when 16 Discovery nodes were connected to the Endeavor network switch and used the same HCAs as other Endeavor nodes.

Figure 11 captures the ping-pong, natural order ring (NOR), and ROR latencies using DDR IB and QDR IB. QDR ping-pong and NOR latencies were much lower than corresponding DDR latencies from 16 to 128 cores. However, at 128 cores, ROR latency for both DDR and QDR was the same.

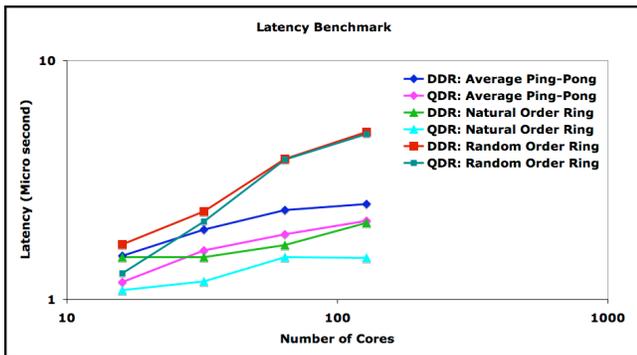


Figure 11. Performance of ping-pong, NOR, and ROR latencies on Discovery.

Figure 12 shows the ping-pong, NOR, and ROR bandwidths for DDR IB and QDR IB. All three bandwidths were much higher for QDR than DDR. At 128 cores, ROR bandwidth using DDR was 131 MB/s and 310 MB/s using QDR, a gain by a factor of 2.37.

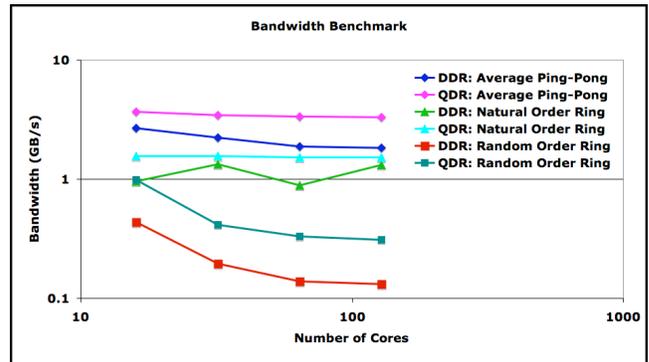


Figure 12. Performance of ping-pong, NOR, and ROR bandwidth on Discovery.

Figure 13 shows the relative performance of QDR IB over DDR IB for G-HPL, PTRANS, G-FFT and Global Random Access (GRA) on 128 cores (16 nodes) of Xeon 5560. G-HPL results were about the same for QDR and DDR, while PTRANS, G-FFT, and GRA were better with QDR by factors of 1.87, 1.48 and 1.35, respectively. Most of real life applications gain visibly less from network bandwidth improvements than these three HPCC kernels, which substantially depend on the interconnect bandwidth.

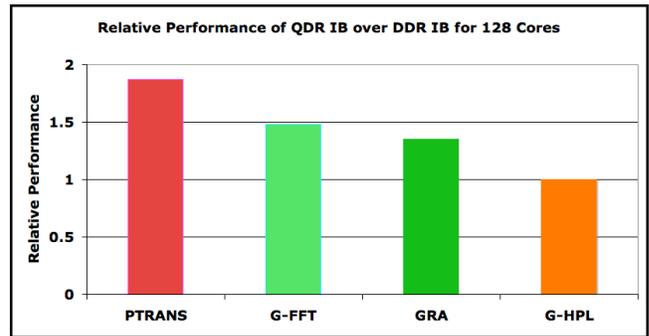


Figure 13. Relative performance of PTRANS, G-FFT, GRA and G-HPL on 128 cores of Discovery.

4.3 NPB MPI

In this subsection, we present results for six (MG, CG, FT, BT, LU, and SP) of the MPI NPB [11]. Figure 14 displays the single-node (8-core) performance ratios for CG, MG, FT, and LU benchmarks. We do not show results for the BT and SP benchmarks as they run on a square grid only. CG is the most memory-intensive benchmark and cannot reuse the cache, as it involves indirect addressing and has to fetch data from memory. As a result, Discovery performance advantage for the CG benchmark over both ICE and Endeavor was a factor of 3.3. Next to CG, the most memory-intensive benchmark is MG, and its performance on Discovery was higher by factor of 2.82 and 2.90 than on ICE and Endeavor, respectively. For somewhat less memory-intensive benchmarks (FT and LU), the performance advantage of Discovery was a factor of 2.1 and 1.7 over ICE, and 2.4 and 1.8 over Endeavor. For all four benchmarks, performance of ICE within a node was slightly higher than Endeavor because of the faster clock (3.0 GHz vs. 2.8 GHz) and memory (800 MHz

vs. 667 MHz). Within a node, the performance advantage of Discovery over ICE and Endeavor is due to its higher sustained-memory bandwidth and micro-architecture improvements (that is, faster unaligned loads making automatic vectorization easier and more effective). Within a node, in order of performance, the three systems were Discovery, ICE, and Endeavor.

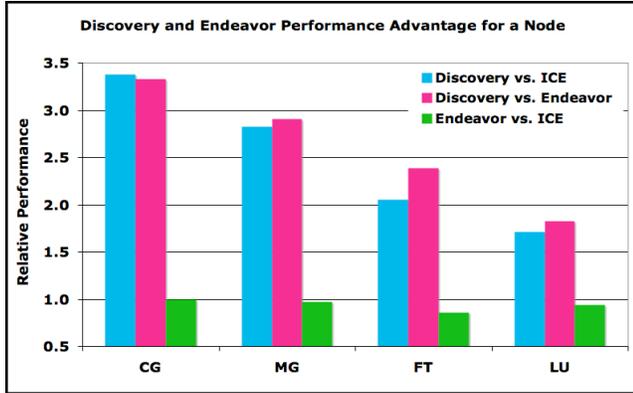


Figure 14. Performance advantage of Discovery and Endeavor on one node.

Figure 15 shows the performance ratios on 256 cores for the CG, MG, FT, LU BT, and LU benchmarks. Discovery held the performance advantage for the same reasons as for single node. However, the advantage was somewhat less pronounced at higher core counts primarily due to communication overhead. The performance advantage of CG on Discovery over ICE fell from a factor of 3.3 to 2.3, and of MG, from 2.82 to 2.5. For FT on 256 cores, Discovery had the highest performance advantage by a factor of 2.7 over ICE, and by a factor 2.0 over Endeavor. The reason that Discovery performance advantage for FT was more than that for MG is that at 256 cores the communication time (interconnect latency and bandwidth—especially bisection bandwidth) has a bigger impact than memory bandwidth. Bisection bandwidth of Discovery was highest due to the QDR interconnect and fat-tree topology. Bisection bandwidth for Endeavor and ICE were lower. On a single node (8 cores), the performance of CG, MG, FT, and LU was better on ICE than on Endeavor. However, on 256 cores, performance of all benchmarks except BT was higher on Endeavor than on ICE. The performance of BT was better than on Endeavor because BT is compute-intensive and has smaller communication overhead than others.

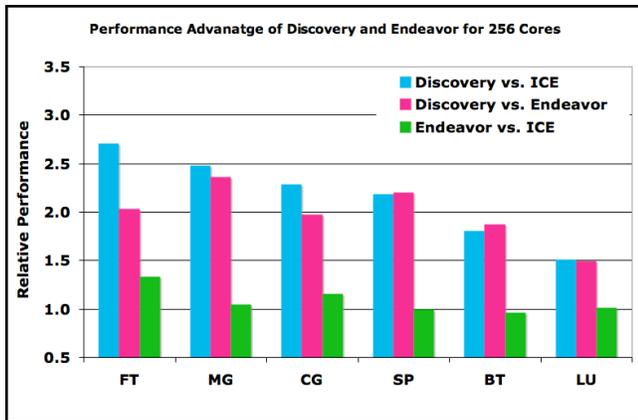


Figure 15. Performance advantage of Discovery on 256 cores.

Figure 16 shows the performance advantage of Discovery and Endeavor on 512 cores for the CG, MG, FT, and LU benchmarks. Unlike when running on 256 cores, benchmark performance was better on Endeavor than on ICE, underscoring once again the difference made by topology. The performance advantage for FT on Discovery over the other two systems was the same as for 256 cores.

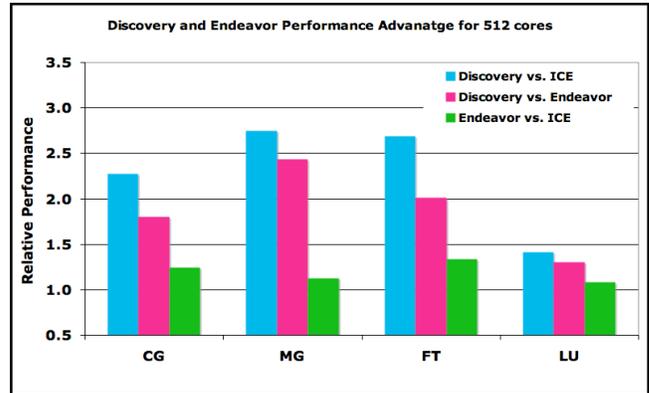


Figure 16. Performance advantage of Discovery and Endeavor on 512 cores.

4.4 NPB MPI Turbo Mode

In this subsection, we compare results for the MPI version of the NPB with Turbo mode On and Off. The maximal upside from Turbo mode for this specific 5500-processor model is 9.5 percent (two 133 MHz frequency increments) in fully subscribed mode. Figure 17 shows the measured performance advantage of Turbo mode. We ran six NPB (MG, SP, CG, FT, LU, and BT) for cores ranging from 16 to 512. We tabulated performance in Gflop/s in both modes and calculated the average for this relative performance. The performance gain was 1–5 percent. It was higher for more compute-bound benchmarks (for example, BT, LU) and lower for more memory-bound benchmarks (for example, MG).

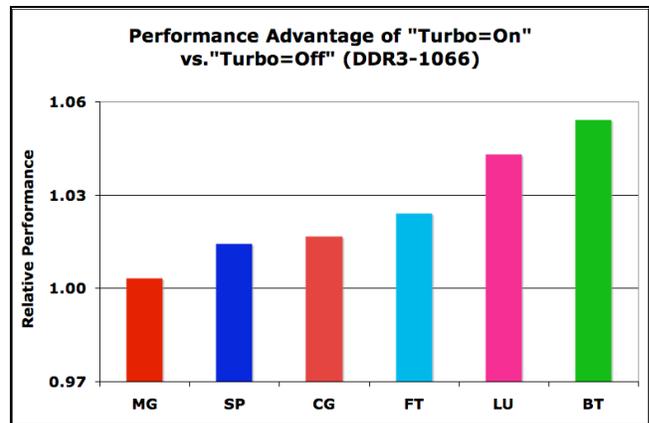


Figure 17. Performance advantage of Turbo On on Discovery.

4.5 NPB SMT On and Off Mode

In Figure 18, we show the relative performance of NPB in SMT mode. We used four Discovery nodes for our experiments. With SMT, the node can handle twice as many processes (16) as without SMT (8). With more processes per node, there is greater communication overhead. In other words, more processes

compete for the same host channel adapter (HCA) on the node. On the other hand, additional processes (or threads) can steal cycles in cases of communications imbalance or memory access stalls. The result is better overall performance. For one node, SMT mode did not provide an advantage for any NPB test but LU. There was a slight gain for LU. FT did not achieve any SMT benefit for any number of nodes. BT, SP, MG, and LU achieved the greatest benefit from SMT at 4 nodes: factors of 1.54, 1.43, 1.14, and 1.14, respectively.

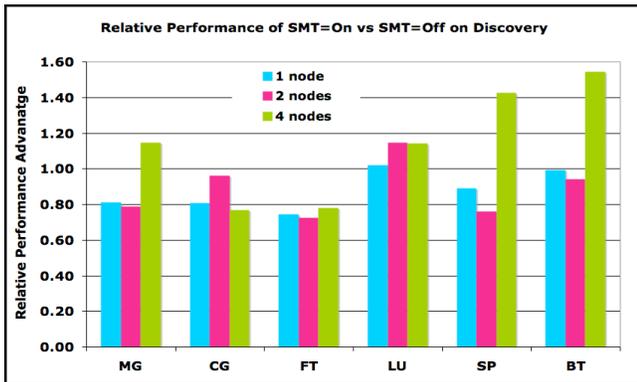


Figure 18. Relative performance of NPB with SMT mode.

4.6 NPB with DDR3-1333

In Figure 19, we compare the results of NPB with DDR3-1333 and DDR3-1066 memory. MG, CG, FT and LU results are for 4, 8, 16, 32 and 64 cores. However, SP and BT results are for 4, 9, 16, 25 and 64 cores as they run on a square grid only. All the NPB benchmarks benefited from faster memory for all the number of cores we tested: the gain was 2-11 percent. On the HPCC EP-Stream Triad test, the result for pure memory bandwidth was 12 percent. SP benefits the most from faster memory.

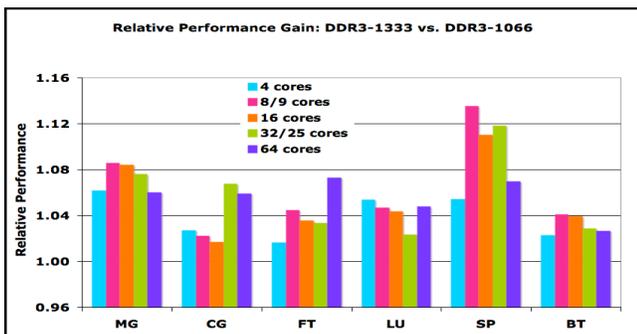


Figure 19. Relative performance of NPB benchmarks.

4.7 Scientific and Engineering Applications

In the following section, we present the results for four real-world applications on ICE, Endeavor, and Discovery.

4.7.1 OVERFLOW-2

In this subsection, we present and analyze results of the simulation using the CFD application OVERFLOW-2 [12] on the three systems. Figure 20 shows wall-clock time for 8–512 cores for OVERFLOW-2. Performance of OVERFLOW-2 on Discovery is much better than on the ICE system across the entire range of cores. OVERFLOW-2 is a cache-friendly, memory-intensive application and therefore performance was better on Discovery than on ICE because memory bandwidth of the former

is better (4.2 vs. 1.23 GB/s). The ICE system, despite having the highest floating-point operations per clock (12 vs. 11.2 Gflop/s), performed worse than Discovery. The ratio of GB/Gflop was 0.11 for ICE—memory bandwidth is inadequate to feed the floating-point units. For Discovery, by contrast, it was 0.41, greater by a factor of 3.7. However, ICE has an advantage, especially for large numbers of cores, as its L2 cache is 3 MB per core compared with 2 MB per core of L3 for Discovery. Overall Discovery performance was better than ICE by a factor of 2.

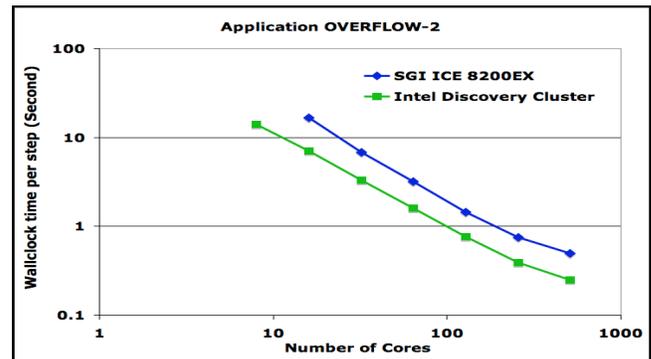


Figure 20. OVERFLOW-2 wall-clock time per step for ICE and Discovery.

4.7.2 CART3D

In this subsection, we present and analyze results of the simulation using the CFD application CART3D [13] on each of the three clusters. Figure 21 shows wall-clock time per step for 16–512 cores for CART3D. Performance of CART3D was best on Discovery and worst on the ICE system. Because CART3D is both memory- and compute-intensive, it benefits from a faster processor clock and better memory bandwidth. On ICE, CART3D runs only on 64 and 128 in fully subscribed mode (eight processes per node) because the program hangs on other core counts. In addition, it does not run on either Endeavor or Discovery when using all 8 cores of a node. However, it does run on all three systems when using only 4 cores of a node (half-subscribed mode).

We could not run CART3D at 256 and 512 cores on ICE when using eight cores per node due to lack of memory on the node that contains the MPI rank 0 process. On ICE, memory per core is one GB, and that memory was shared by the application, operating system, and kernel. Only 700 MB per core is available for user space. For this reason, we ran CART3D on four cores (half of a node) and eight cores (a full node) of ICE. We could not run CART3D for 256 and 512 cores with eight cores per node because with the MPI paradigm, memory usage increases when core counts increase. While monitoring CART3D’s memory usage when running on 128 cores and using eight cores per node, we found the node containing rank 0 was using 92 percent of its available memory for the user application, and that was using the default value. CART3D, which runs using a large number of iterations (as tested with 128 cores when using eight cores per node), occasionally hung on ICE because memory usage at runtime exceeded available memory [6]. CART3D ran fine when four rather than eight cores per node were used. To run CART3D successfully on the ICE system, two environmental variables must be taken into account: (a) MPI_BUFS_PER_HOST (MBPH) and (b) MPI_BUFS_PER_PROC (MBPP). The values of these variables control the number of the buffers used for message passing between nodes (MBPH) or within a single node (MBPP)

of a cluster. On ICE, default values for these variables are MBPH=32, MBPP=256. Corresponding variables in the Intel MPI Library on Endeavor and Discovery are: (a) RDMA (multi-node): I_MPI_RDMA_BUFFER_NUM=16, I_MPI_RDMA_BUFFER_SIZE=16 Kb and (b) shm (intranode): I_MPI_SHM_NUM_BUFFERS=16, I_MPI_SHM_BUFFER_SIZE=16 Kb. In the Intel MPI library, default values for the number of buffers are too small to run CART3D when 8 cores per node are used. CART3D runs fine when 4 cores per node were used

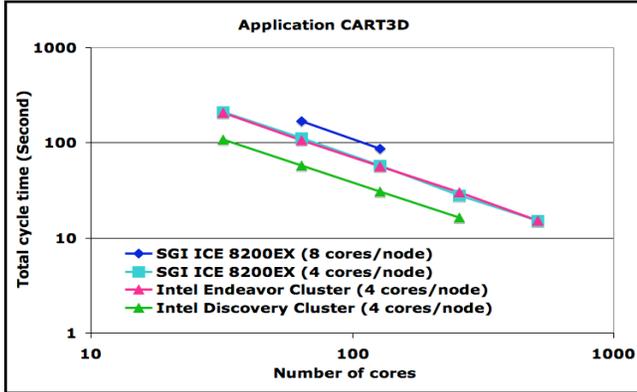


Figure 21. Wall-clock time for CART3D on ICE, Endeavor, and Discovery.

4.7.3 USM3D

Figure 22 shows the USM3D [14] cycle wall-clock time per step for a range of processors. The performance of USM3D was better on Discovery than on ICE. USM3D is an unstructured mesh-based application and memory-bound from indirect addressing. Consequently, it does not make good use of the L2/L3 caches—it depends exclusively on the memory bandwidth, which is highest for Discovery (4.2 GB/s) and lowest for ICE and Endeavor systems (1.23 GB/s). Beyond 256 cores, USM3D scaling was poor for this dataset, and performance became limited by communications.

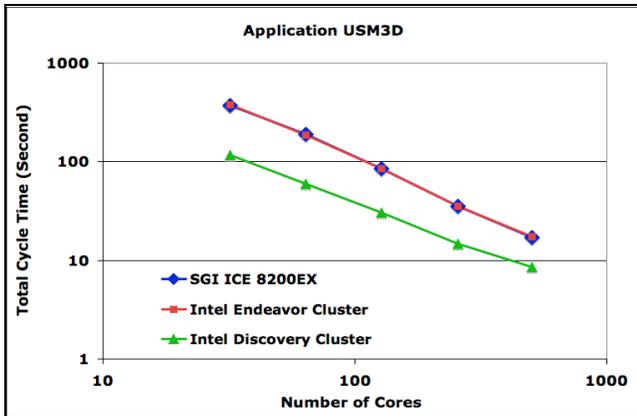


Figure 22. Wall-clock time for USM3D on ICE, Endeavor, and Discovery.

4.7.4 ECCO

In this subsection, we present and analyze results of the climate modeling application ECCO [16] on each of the three clusters. Figure 23 shows wall-clock time for ECCO [16]. This code is memory-bound for small processor counts. Since Discovery provides the highest memory bandwidth (4.2 GB/s), ECCO

performed much better on this system than on Endeavor or ICE. Beyond 256 cores, the code did not scale very well on either system, and the I/O time was becoming dominant (for example, for Discovery at 480 cores, the I/O took 30 seconds out of an overall 65 seconds of wall-time) as the number of metafiles opening and closing increased proportionately.

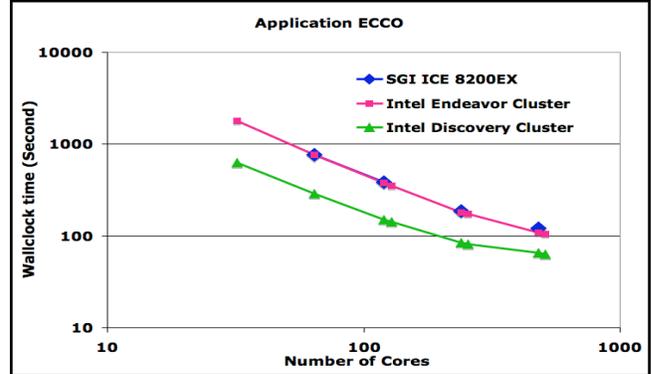


Figure 23. Wall-clock time ECCO on ICE, Endeavor, and Discovery.

Figure 24 shows the performance advantage from SMT and faster memory (DDR3-1333) for Discovery nodes (only 4 in case of SMT). Performance was 1.09-1.12 and 1.12 times better using DDR3-1333 instead of DDR3-1066 and with SMT On (16 processes per node) compared with SMT Off (8 processes per node), respectively.

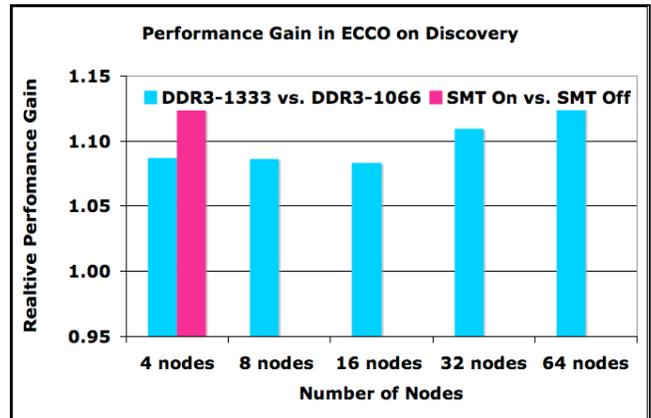


Figure 24: Performance gain from SMT and faster memory.

4.7.5 Performance Advantage of Discovery

In this subsection, we present analyses of the performance advantage of Discovery over ICE and Endeavor, and Endeavor over ICE. Figure 25 shows the relative performance of Discovery over ICE for all four applications. It is clear from this figure that the relative performance advantage of Discovery decreases as the number of cores increase from 64 to 512. At higher core counts, communication time (network latency and bandwidth) becomes dominant over compute time; therefore, the large memory bandwidth of Discovery has a somewhat smaller impact. USM3D is the most memory-intensive application since it cannot reuse L2/L3 cache—it uses indirect addressing because of the unstructured mesh used. Therefore, it had the highest relative performance advantage on Discovery—3.2 and 2.1 for 64 and 504 cores, respectively.

OVERFLOW-2 is another memory bandwidth-intensive application, but it is very cache-friendly with almost negligible communication overhead. For OVERFLOW-2, the performance advantage of Discovery over ICE was a factor of 2.0 and almost the same from 64 to 512 cores. ICE has a larger amount of last level cache (LLC, which is L2 for ICE and L3 for Discovery in this case)—there is 3MB of L2 per core on ICE and 2 MB of L3 per core on Discovery. As a result, ICE did not fall below 2.0.

ECCO is compute- and memory-intensive. It does not scale well beyond 256 cores due to I/O time impact for this dataset. The performance advantage of Discovery was 2.6 at 64 cores and 1.86 at 480 cores.

CART3D is both compute-bound and memory-bound. For CART3D, the performance advantage of Discovery over ICE was 1.9 and 1.72 for 64 and 256 cores.

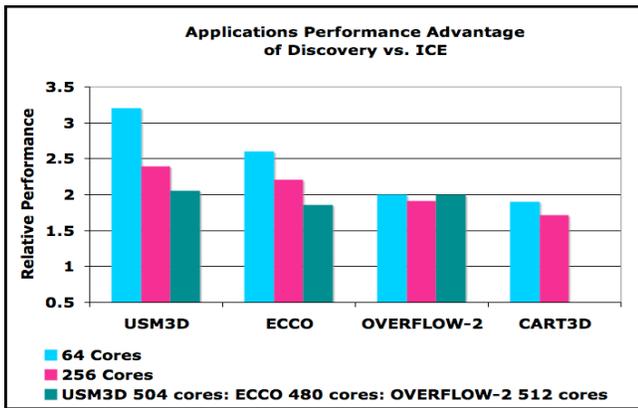


Figure 25. Performance advantage of Discovery over ICE.

Figure 26 shows the relative performance of Endeavor over ICE for three applications: USM3D, ECCO, and CART3D. Here, performance of both ICE and Endeavor was about the same for up to 256 cores. At higher core counts there were some differences. ECCO was faster on Endeavor at 480 cores since ECCO is network latency-sensitive at large numbers of cores and Endeavor has a single switch-based fat-tree network versus hypercube in ICE with multiple switches.

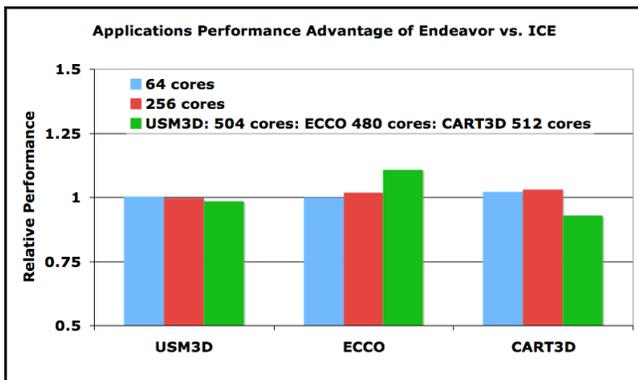


Figure 26. Performance advantage of Endeavor over ICE.

4.8 Multi-Core Effects

While increasing the number of processor cores provides numerous benefits for HPC, it raises some problems that also need to be addressed. Most notable of these are sustained-memory bandwidth per core, contention for network (that is, more cores on

the node using the same HCA), and some shared processor resources (for example, last level cache). To illustrate the combined impact of these factors, we ran a number of applications in half-subscribed mode (used 2 cores of each socket) and compared the performance with that of the fully subscribed mode. Some of those comparisons can be found in HPCC section above. For many full-scale applications (including but not limited to those considered in this paper), the performance difference was similar to what we show below for ECCO and USM3D, which we present as two examples.

Figure 27 shows the relative performance advantage for ECCO in half-subscribed mode. ECCO does not run on 32 cores of ICE due to memory footprint. All three systems showed performance improvement in half-subscribed mode. Discovery results showed smaller benefit from half-subscribed mode due to higher sustained-memory bandwidth. Still, for many applications (ECCO included) higher memory bandwidth per core in half-subscribed mode helped even in the case of Discovery. Memory bandwidth per core typically has a smaller impact with core count growth (especially for strong scaling cases), but contention for network resources then comes into play as communication overhead becomes more visible.

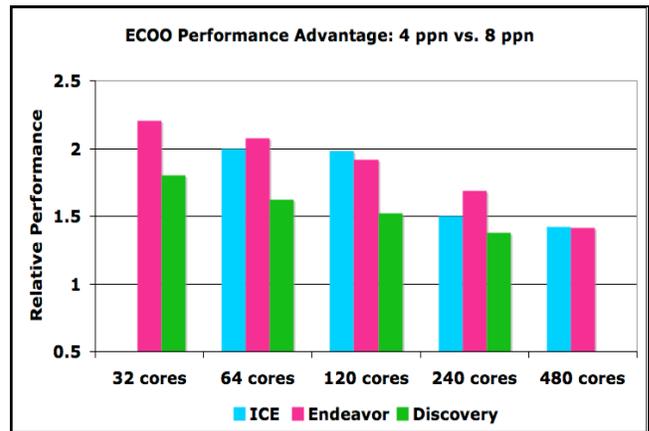


Figure 27. Relative performance for ECCO using 4 and 8 cores of a node.

Figure 28 shows the relative performance advantage for USM3D in half-subscribed mode. All three systems showed significant performance improvement in half-subscribed mode. Performance gain using four cores of a node increases with increasing number of cores and then decreases at higher number of cores when communication becomes dominant over computation.

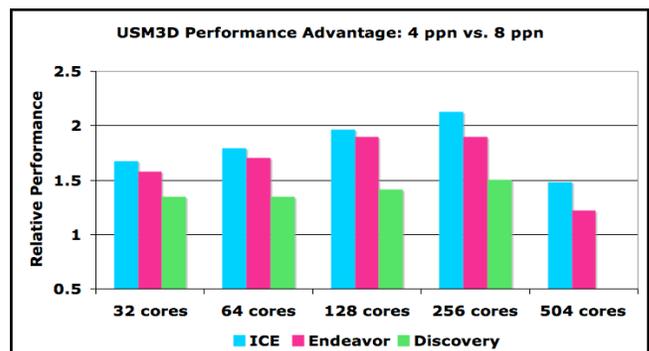


Figure 28. Performance gain of USM3D using 4 and 8 cores of a node.

5. Summary and Conclusions

This paper examined the performance characteristics of a cluster built from the newly launched Intel Xeon processor 5500 series (previously known as “Nehalem-EP”) for a range of core counts using both standard benchmarks (HPCC and NPB) and a selection of full-scale applications. Memory bandwidth is the most prominent improvement in this new architecture compared with previous architectures. Since memory bandwidth has been the performance limiter for so many of the HPC applications, we observe significant performance improvements even when running with clock frequencies not higher than those of previous processor generations. In other words, the Xeon 5500 has a much better bandwidth-to-compute balance for HPC. The sensitivity to memory bandwidth is evident when comparing measurements for memory-bound codes on different platforms as well as using different memory speeds on the same platform.

Among the architecture’s numerous new features, there are two in particular that will help enterprise applications. We examined both features in this paper. Intel Turbo Boost Technology (which allows higher frequencies) delivers some performance improvements to compute-intensive codes. And when it does, the contribution is proportional to frequency up-tick. The impact is smaller when the application is memory-bound. Intel Hyper-Threading Technology (SMT) is helpful in some cases, but for HPC applications this is not universal. Experimentation is recommended.

Some conclusions can be drawn from observing the performance profile when increasing the core or node count. As might be expected, for larger systems the interconnect bandwidth and topology have an increasing impact. In such cases, a fully connected fat-tree topology has a performance advantage over hypercube topology (though the same cannot be said for its cost and complexity especially as the node count grows). It is also clear that to maintain the per-node proportional performance on a large system, it is necessary to increase the interconnect fabric’s performance accordingly. The performance advantage of real-world applications on Discovery is between 1.9 and 2.1 over Endeavor for higher core counts and up to 3.2 for lower ones.

In summary, the Intel Xeon 5500 Processor provides a well-balanced high-performance building block for HPC platforms and systems.

6. ACKNOWLEDGMENTS

We gratefully acknowledge the help and support provided by staff of Intel Corporation (Dmitry Mishura, Sergey Shalnov, Dmitry Shkurko, Mack Stallcup, Michael Greenfield, and the CRT Datacenter team), Computer Sciences Corporation editors (Holly Amundson and Jill Dunbar), John Baron from SGI, Dennis Jespersen from NASA Ames Research Center and The TDA Group.

7. REFERENCES

[1] R. C. Murphy, P. M. Kogge, and A. Rodrigues, “The Characterization of Data Intensive Memory Workloads on Distributed PIM Systems,” *Intelligent Memory Systems* 2000: 85–103.

- [2] Intel White Paper, “First the Tick, Now the Tock: Next Generation Intel® Microarchitecture (Nehalem),” Intel publication 0408/VP/HBD/PDF 319724-001US, April 2008.
- [3] Intel® Microarchitecture (Nehalem), www.intel.com/technology/architecture-silicon/next-gen/.
- [4] “An Introduction to the Intel® QuickPath Interconnect,” Document Number: 320412, January 2009, www.intel.com/technology/quickpath/; www.intel.com/technology/quickpath/introduction.pdf.
- [5] K. J. Barker, K. Davis, A. Hoisie, D. J. Kerbyson, M. Lang, S. Pakin, J.C. Sancho, “A Performance Evaluation of the Nehalem Quad-core Processor for Scientific Computing,” *Parallel Processing Letters*, Vol. 18, No. 4 (2008) 453-469.
- [6] S. Saini, D. Talcott, D. Jespersen, J. Djomehri, H. Jin, and R. Biswas, “Scientific Application-based Performance Comparison of SGI Altix 4700, IBM Power5+, and SGI ICE 8200 Supercomputers,” *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, Austin, Texas, November 15-21, 2008.
- [7] SGI Altix ICE Integrated Blade Platform, www.sgi.com/pdfs/4008.pdf.
- [8] InfiniBand Trade Association, www.infinibandta.org/home.
- [9] Message Passing Toolkit (MPT) User’s Guide, <http://techpubs.sgi.com/library/manuals/3000/007-3773-003/pdf/007-3773-003.pdf>.
- [10] TOP500 Supercomputing Sites, www.top500.org/.
- [11] Intel® 5400 Chipset—Technical Documents, www.intel.com/Products/Server/Chipsets/5400/5400-technicaldocuments.htm.
- [12] HPC Challenge Benchmarks, <http://icl.cs.utk.edu/hpcc/>.
- [13] NAS Parallel Benchmarks, www.nas.nasa.gov/Resources/Software/npb.html.
- [14] OVERFLOW-2, <http://aaac.larc.nasa.gov/~buning/>.
- [15] D. J. Mavriplis, M. J. Aftosmis, and M. Berger, “High Resolution Aerospace Applications using the NASA Columbia Supercomputer,” *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, Seattle, Washington, Nov. 12–18, 2005.
- [16] USM3D, http://aaac.larc.nasa.gov/tsab/usm3d/usm3d_52_man.html.
- [17] METIS Family of Multilevel Partitioning Algorithms, www.cs.umn.edu/~metis.
- [18] ECCO: Estimating the Circulation and Climate of the Ocean, www.ecco-group.org/.
- [19] Intel® MPI Library 3.2 Support Resources, Intel® Software Network, www.intel.com/cd/software/products/asmo-na/eng/308292.htm.
- [20] Intel® Math Kernel Library 10.1 Overview, Intel® Software Network, www.intel.com/cd/software/products/asmo-na/eng/307757.htm.
- [21] FFTW, FFTW to Intel® Math Kernel Library Wrappers, Technical User Notes, www.fftw.org/.