A Study of Hyper-Threading in High-Performance Computing Clusters

By Tau Leng, Ph.D.; Rizwan Ali; Jenwei Hsieh, Ph.D.; and Christopher Stanton

The effects of the Intel® Xeon™ processor's Hyper-Threading technology on server performance vary according to the type of applications the server is running. Hyper-Threading affects high-performance computing (HPC) clusters similarly. The type of application run on a cluster will determine whether Hyper-Threading will help or hinder performance. In addition, the size of the cluster (the number of nodes it contains) and its configuration (particularly, the number of processors used in each machine) will also affect Hyper-Threading technology's impact on performance.

A feature of the Intel® Xeon™ processor, Hyper-Threading technology makes a single physical processor appear as two logical processors to the operating system,[1] thereby allowing a processor to execute two instructions from different threads in parallel rather than in serial. This capability can improve the performance of highly parallel applications and can lead to better processor utilization.

Previous studies have shown that Hyper-Threading technology improves multithreaded applications' performance by 10 to 30 percent, depending on the characteristics of the applications.[2] These studies also suggest that the potential gain is obtained only if the application is multithreaded by any parallelization technique.

In high-performance computing (HPC) clusters, applications are commonly implemented by using standard message-passing systems such as Message Passing Interface (MPI) or Parallel Virtual Machine (PVM). Applications developed from a message-passing programming model often use a mechanism, `mpirun` for example, to spawn multiple processes and map them to processors in the system. Parallelism is achieved through the message-passing system, which coordinates the parallel tasks among processes. Unlike multithreaded programs in which the values of application variables are shared by all the threads, a message-passing application runs as a collective of autonomous processes, each with its own local memory.

This type of application can also benefit from the Hyper-Threading technology incorporated in Intel Xeon processors—the number of processes spawned can be doubled and the parallel tasks can potentially execute faster. Applying Hyper-Threading and doubling the number of processes that simultaneously run on the cluster will increase the utilization rate of the processors' execution resources; therefore, performance can be improved. On the other hand, overhead might be introduced in the following ways:

- More processes running on the same node may create additional memory contention.

[1] For more information on Hyper-Threading technology, see "An Introduction to Hyper-Threading Technology in the Intel Xeon Processor Architecture" by Humayun Khalid in Dell *Power Solutions*, August 2002.
[2] William Magro, Paul Petersen, and Sanjiv Shah, "Hyper-Threading Technology: Impact on Compute-Intensive Workloads," *Intel Technology Journal,* vol. 6, no. 1 (February 2002), http://www.intel.com/technology/itj/2002/volume06issue01/art06_computeintensive/p01_abstract.htm.

- Logical processes may compete for access to the cache and thus generate more cache-miss situations.
- More processes on each node increase the communication traffic (message passing) between nodes, which can saturate the shared memory, the I/O bus, or the communication capacity of the network interface adapter and thus create performance bottlenecks.

Whether the performance benefits of Hyper-Threading—particularly, better resource utilization—can nullify these overhead conditions depends on an application's characteristics. This article will discuss how the Dell™ HPC cluster team used various MPI benchmark programs to demonstrate the impact of Hyper-Threading technology on a Linux®-based HPC cluster and the adaptability of this new technology for improving performance in HPC clusters.

Establishing the test environment

The test environment was a cluster consisting of 32 Dell PowerEdge™ 2650 servers interconnected with Myricom™ Myrinet™ networking components. Each PowerEdge 2650 had two Intel Xeon processors at 2.4 GHz, 512 KB level 2 (L2) cache, 2 GB of double data rate (DDR) RAM, and a 400 MHz frontside bus. The PowerEdge 2650 uses the ServerWorks™ Grand Champion™ LE chipset, which accommodates up to 12 registered DDR 200 (PC1600) dual in-line memory modules (DIMMs), each with capacities of 128 MB up to 1 GB with a two-way interleaved memory architecture. Each of the two Peripheral Component Interconnect Extended (PCI-X) controllers on the PowerEdge 2650 had its own dedicated 1.6 GB/sec full-duplex connection to the North Bridge to accommodate the peak traffic generated by the PCI-X buses.

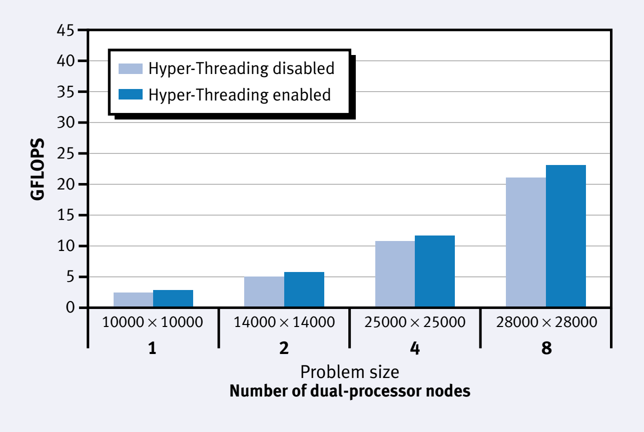

Figure 1. HPL performance results for the cluster (ranging from one node to eight nodes) when using only one CPU on each node

The Dell team used two benchmark suites to test the performance of Hyper-Threading in HPC clusters: High-Performance Linpack (HPL), a benchmark commonly used for HPC environments, and the NAS (NASA Advanced Supercomputing) Parallel Benchmarks. Linpack uses several linear algebra routines to measure the time required to solve dense linear equations in double-precision (64-bit) arithmetic using the Gaussian elimination method. The measurement obtained from Linpack is the number of floating-point operations per second (FLOPS). The NAS benchmark suite comprises five kernels and three pseudo-applications and is designed to gauge parallel computing performance.[3] Its results are measured in mega operations per second (MOPS).

Evaluating cluster performance with Linpack

Linpack primarily exercises the floating-point calculation capability of a system. However, a system's communication bandwidth also significantly influences the overall Linpack performance. Dual processors and high-speed networking connections, such as Myrinet, can help a cluster reach almost 60 percent of its theoretical performance, but a slower interconnect such as Fast Ethernet could bring actual performance to less than 30 percent of the theoretical performance.[4]

When running Linpack, the more memory used, or the larger the problem size for executing the program, the better the system performance. But to avoid a swapping situation that will decrease performance significantly, the problem size or memory usage should not exceed 85 percent of the total memory.

The impact of Hyper-Threading

Using different combinations of compute nodes and CPU configurations, the Dell team ran HPL on the test cluster and then compared the results of Hyper-Threading disabled versus enabled to determine how Hyper-Threading affected the cluster's performance. Figure 1 shows the performance results of clusters with one physical CPU per machine; Figure 2 shows the performance results of clusters with two physical CPUs per machine. Cluster configurations ranged from one node to eight nodes. The problem size for each configuration was determined by the amount of memory available—1 GB RAM per physical processor.

The results showed that the performance gains from Hyper-Threading were larger on the single-CPU configurations than on

[3] NASA Advanced Supercomputing Division (formerly the Numerical Aerospace Simulation Division), *NAS Parallel Benchmarks home page*, http://www.nas.nasa.gov/Software/NPB.

[4] Netlib, "HPL—A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers," *Netlib Repository at UTK and ORNL*, http://www.netlib.org/benchmark/hpl.

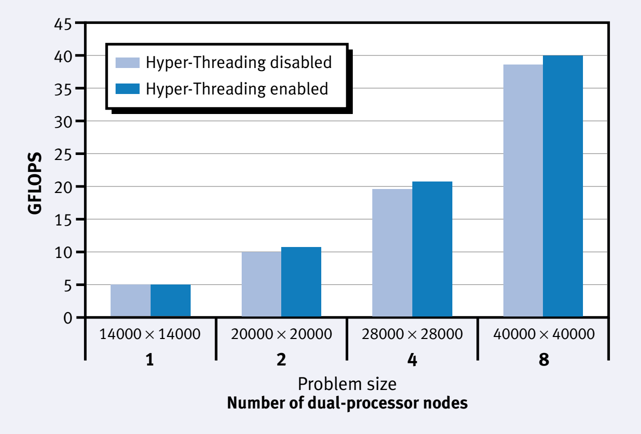

Figure 2. HPL performance results for the cluster (ranging from one node to eight nodes) when using two CPUs on each node

the dual-CPU configurations, approximately 10 percent versus 5 percent. This difference occurred because the overhead conditions mentioned earlier were more severe for the configurations that had four processes on each node, which occurs on dual-processor machines with Hyper-Threading enabled.

Evaluating cluster performance with NAS

Although these results indicate that applications similar to Linpack can benefit from Hyper-Threading, mixed results occurred when running the NAS Parallel Benchmarks suite. For example, the Integer Sort (IS) benchmark, which is a communication-bound program, showed approximately 10 percent degradation in performance when applying Hyper-Threading on an 8×2 configuration (eight nodes with two CPUs on each node). Meanwhile, the Embarrassingly Parallel (EP) benchmark, a CPU-bound program, showed a 40 percent performance improvement. These two benchmark programs represent the opposite ends of the NAS suite: communication-intensive applications versus processor-intensive applications.

The IS and EP benchmarks

The IS benchmark is unique because it uses no floating-point arithmetic yet requires significant data communication. In Hyper-Threading-enabled configurations, doubling the number of IS processes running on each node (from two to four processes) exacerbates communication overhead among processes and memory contention inside the nodes. Furthermore, the CPU floating-point execution units remain underutilized and performance degrades (see Figure 3).

But when the cluster consists of 32 nodes, the IS benchmark will perform better with Hyper-Threading enabled and the processes doubled. This improvement finally occurs because the proportion of communication through the networking interconnect exceeds that which occurs through the shared memory, easing the memory contention and communication bottlenecks of four logical processors running four processes in each compute node. Therefore, performance should improve even for clusters larger than 32 nodes and running the IS benchmark with the same configuration.

In contrast, the EP benchmark primarily performs floating-point calculations and requires almost no communication during the runs. Hyper-Threading-enabled clusters outperform Hyper-Threading-disabled clusters because the EP benchmark can effectively utilize the cluster's CPU resources without causing the communication overhead. This improvement occurs regardless of the number of nodes or CPUs in the cluster. Figure 4 shows that the EP benchmark performance improved linearly, from 1 node to 32 nodes, and also shows the performance gained by enabling Hyper-Threading.

Other NAS benchmarks

Other programs in the NAS benchmark suite contain combined floating-point computation and data communication operations. These operations behave differently for each program. For example, varying the size of messages communicating between processes, the

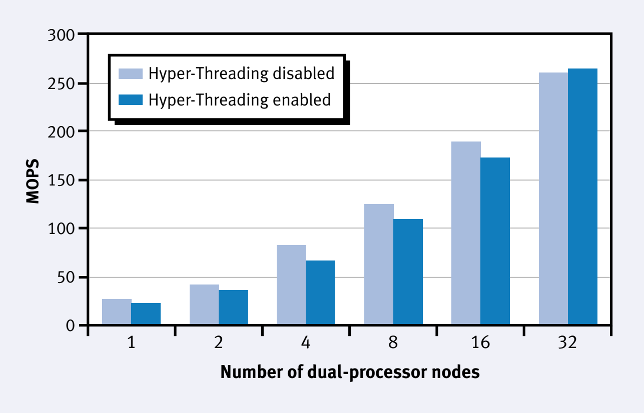

Figure 3. IS (Class B) benchmark results for Hyper-Threading-enabled and Hyper-Threading-disabled configurations

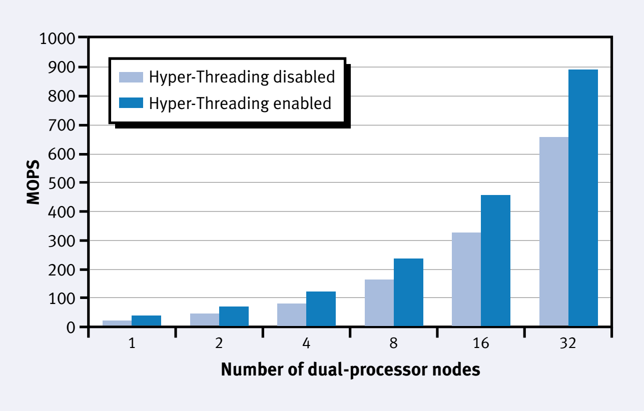

Figure 4. EP (Class B) benchmark results for Hyper-Threading-enabled and Hyper-Threading-disabled configurations

frequency of the communication, or the distances of the message passing among processes can affect a cluster's performance when Hyper-Threading is enabled.

The FT benchmark involves a three-dimensional, partial-differential equation, which is solved using fast Fourier transforms (FFTs). Figure 5 shows that, for applications similar to the FT benchmark, Hyper-Threading will not provide any performance gain but instead will degrade performance for all node counts.

On the other hand, the lower-upper diagonal (LU) benchmark solves a regular-sparse, block 5×5 lower and upper triangular system by using a symmetric successive over-relaxation (SSOR) numerical scheme. Because the algorithm implemented in this program is not highly parallelized, most of the MPI operations are in blocking mode. The performance bottleneck occurs primarily on the networking interconnect. For this type of application, enabling Hyper-Threading may reduce the volume of communication through the interconnect, and therefore could produce a small performance gain, as shown in Figure 6.

Strengthening HPC clusters with Hyper-Threading

Hyper-Threading can improve the performance of some MPI applications, but not all. Depending on the cluster configuration and, most importantly, the nature of the application running on the cluster, performance gains can vary or even be negative. The next step is to use performance tools to understand what areas contribute to performance gains and what areas contribute to performance degradation.

Tau Leng, Ph.D. *(tau_leng@dell.com) is the lead engineer for HPC clustering in the Scalable Systems Group at Dell. His current research interests are parallel processing, distributed computing systems, compiler optimization, and performance benchmarking. Tau has a B.S. in Mathematics from the Fu Jen Catholic University in Taiwan, an M.S. in Computer Science from Utah State University, and a Ph.D. in Computer Science from the University of Houston.*

Rizwan Ali *(rizwan_ali@dell.com) is a systems engineer working in the Scalable Systems Group at Dell. His current research interests are performance benchmarking and high-speed interconnects. Rizwan has a B.S. in Electrical Engineering from the University of Minnesota.*

Jenwei Hsieh, Ph.D. *(jenwei_hsieh@dell.com) is a member of the Scalable Systems Group at Dell. Jenwei is responsible for developing high-performance clusters. He has published more than 30 technical papers in the area of multimedia computing and communications, high-speed networking, serial storage interfaces, and distributed network computing. Jenwei has a Ph.D. in Computer Science from the University of Minnesota and a B.E. from Tamkang University in Taiwan.*

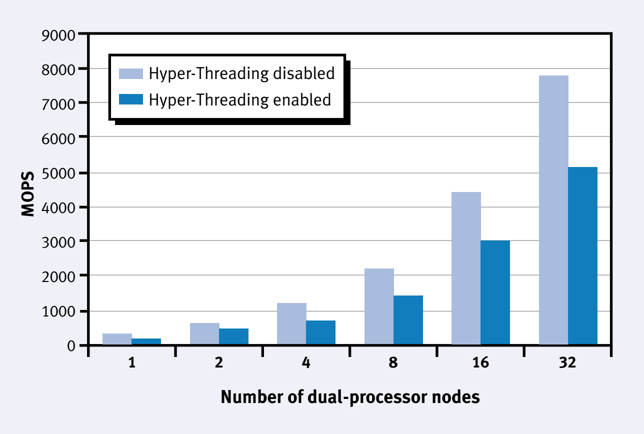

Figure 5. FT (Class B) benchmark results of Hyper-Threading-enabled and Hyper-Threading-disabled configurations

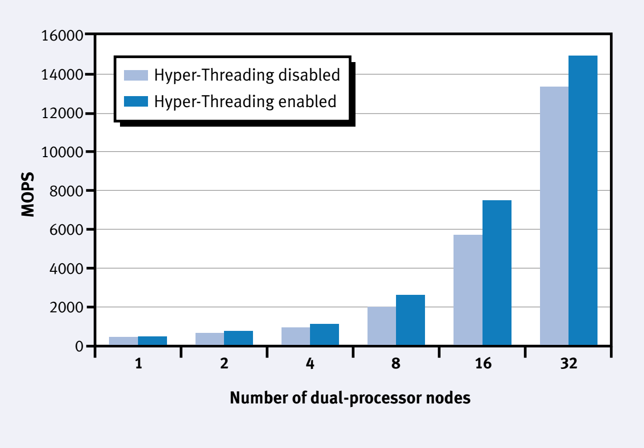

Figure 6. LU (Class B) benchmark results of Hyper-Threading-enabled and Hyper-Threading-disabled configurations

Christopher Stanton *(christopher_stanton@dell.com) is a senior systems engineer in the Scalable Systems Group at Dell. His HPC cluster-related interests include cluster installation, management, and performance benchmarking. Christopher graduated from the University of Texas at Austin with a B.S. and special honors in Computer Science.*

FOR MORE INFORMATION

High-Performance Linpack:
http://www.netlib.org/benchmark/hpl

NAS Parallel Benchmark suite:
http://www.nas.nasa.gov/Software/NPB