Hyper Threading Technology in Hardware Architecture for Processor Efficiency Enhancement

Neha Srivastava[*1], Kalyan Awasthi[2], and Sadaf Z. Rizvi[3]

ABSTRACT

Intel's Hyper-Threading Technology brings the concept of simultaneous multi-threading to the Intel architecture. Hyper-Threading Technology makes a single physical processor appear as two logical processors; the physical execution resources are shared and the architecture state is duplicated for the two logical processors. From a software or architecture perspective, this means operating systems and user programs can schedule processes or threads to logical processors as they would on multiple physical processors. From a micro architecture perspective, this means that instructions from both logical processors will persist and execute simultaneously on shared execution resources. This paper describes the Hyper-Threading Technology architecture of Intel's first implementation on the Intel Xeon processor family. Hyper-Threading Technology is an important addition to Intel's enterprise product line and will be integrated into a wide variety of products.

Keywords*: Hyper threading technology, Multi threading , Architecture, Single task and Multi task mode, Media application*

1. INTRODUCTION

Hyper-threading is a technology incorporated into Intel® Xeon™ processors and is the Intel implementation of an architectural technique called *simultaneous multi-threading* (SMT). SMT technology was conceived to exploit a program's instruction and thread-level parallelism through the simultaneous execution of multiple threads on a single physical processor. Several commercial processors recently have incorporated SMT technology.

To understand Hyper-Threading, the basic issues that provided the impetus to develop such a technology for Intel Architecture-(IA-) 32 processors must first be understood [1]. According to Intel, processor efficiency for today's typical server workloads is low, on average, only one-third of processor resources are utilized. Computer users running a Microsoft® Windows® operating system (OS) can monitor processor utilization with the Task Manager tool. Unfortunately, this tool provides only the high-level view of processor utilization. The Task Manager might show 100 percent processor utilization, but the actual utilization of internal, processor-execution resources is often much lower. This inefficient use of processor resources limits the system's overall performance. To

1*. Neha Srivastava, Lecturer, Department of Electronics & Comm. Engg., School of ManagementSciences, Technical Campus, Lucknow (U.P.) – India, e-mail:neha.bbdgec@gmail.com
2. Kalyan Awasthi, Lecturer, Department of Electronics & Comm. Engg., BBDNITM, Lucknow (U.P.)– India, email:awasthi.kalyan@gmail.com
3. Sadaf Z. Rizvi[3] Lecturer, Department of Electronics & Comm. Engg., School of ManagementSciences, Technical Campus, Lucknow (U.P.)– India, e-mail:rizvi.sadaf1010@gmail.com

satisfy the growing needs of computer users, several techniques have been developed over the years to overcome processor inefficiency and improve system performance. These performance strategies were primarily based on two techniques: software improvement and resource redundancy.

Software improvements range from algorithmic changes to code modifications (improved compilers, programming languages, parallel coding, and so forth) to multithreading. These innovations were intended to provide better mapping of an application architecture to the hardware architecture. A perfect mapping would mean that no hardware resource is left unused at any given time. Despite these efforts, a perfectly mapped scenario is yet to be achieved.

The second technique, resource redundancy, avoids any attempt to improve performance through better utilization. Rather, it takes an approach opposite to software improvement and actually degrades efficiency by duplicating resources. Although the duplication degrades efficiency, it helps improve overall performance. The Intel processors of the 1980s had only little execution unit resources and, consequently, could handle only a few instructions at any given time. Today, Intel Pentium® 4 processors (the Intel Xeon processor is derived from the Pentium 4 processor core) are seven-way superscalar machines with the ability to pipeline 126 instructions Multiprocessor systems with these Intel processors allow multiple threads to be scheduled for parallel execution, resulting in improved application performance. This redundancy within a processor as well as within the entire computing system addresses growing performance needs, but it does so at the cost of diminishing returns.

To improve system performance cost-effectively, inefficiencies in processor resource utilization should be alleviated or removed. Hyper-Threading technology provides an opportunity to achieve this objective. With less than 5 percent increase in die size, Hyper-Threading incorporates multiple logical processors in each physical processor package. The logical processors share most of the processor resources and can increase system performance when concurrently executing multithreaded or multitasked workloads [2-3].

2. THREADING TECHNOLOGY

Hyper-threading (HT) is a term used related with computer processor. It is one of the features built-in with most Intel made CPUs. Intel introduced this technology when it releases the first 3GHz Pentium 4 processor.HT technology turns or simulates a single processor into two virtual processors so that it can handle two sets of instructions simultaneously. It is meant to increase the performance of a processor by utilizing the idle or non-used part of a processor. This enables a processor to perform tasks faster (usually 25% - 40% speed increase) than non-HT enabled processor. Generally a single core processor appears as one processor for the OS and executes only one set of instruction at time. But, HT enabled single core processor appears as two processors to the OS and executes two application threads as a result. For example, a dual core processor that supports HT will appear four core processors to the OS. A quad core processor that supports HT will appear as 8-core processor for the OS… etc. Some of Intel family processors that support HT technology are Intel Atom, core processors, Xeon, Core i-series, Pentium 4 and Pentium mobile processors. Hyper-Threading Technology is a groundbreaking innovation that significantly improves processor performance. Pioneered by Intel on the Intel® Xeon™ processor family for servers, Hyper-Threading Technology has enabled greater productivity and enhanced the user experience. View Cast introduced Hyper-Threading on the Niagara Power Stream systems in April 2003.

Hyper-Threading Technology is now supported on the Intel® Pentium® 4 Processor with HT Technology. Hyper-Threading provides a significant performance boost that is particularly suited to today's computing climate, applications, and operating systems.

2.1. How Hyper Threading Works

Faster clock speeds are an important way to deliver more computing power. But clock speed is only half the story. The other route to higher performance is to accomplish more work on each clock cycle, and that's where Hyper-Threading Technology comes in. A single processor supporting Hyper-Threading Technology presents itself to modern operating systems and applications as two virtual processors. The processor can work on two sets of tasks simultaneously, use resources that otherwise would sit idle, and get more work done in the same amount of time [4-8].

HT Technology takes advantage of the multithreading capability that's built in to Windows XP and many advanced applications. Multithreaded software divides its workloads into processes and threads that can be independently scheduled and dispatched. In a multiprocessor system, those threads execute on different processors.

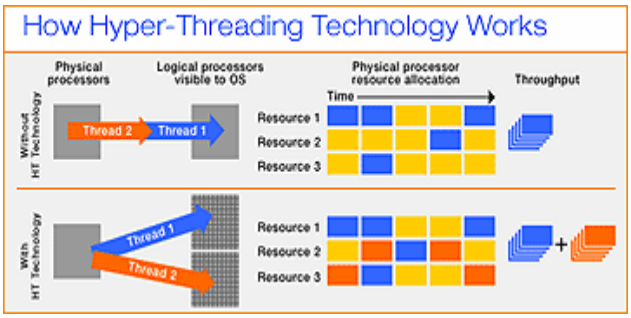

Fig. 1: Working principle of hyper-threading

3. HYPER-THREADING TECHNOLOGY ARCHITECTURE

Hyper-Threading Technology makes a single physical processor appear as multiple logical processors. To do this, there is one copy of the architecture state for each logical processor, and the logical processors share a single set of physical execution resources. From a software or architecture perspective, this means operating systems and user programs can schedule processes or threads to logical processors as they would on conventional physical processors in a multiprocessor system. From a micro architecture perspective, this means that instructions from logical processors will persist and execute simultaneously on shared execution resources.

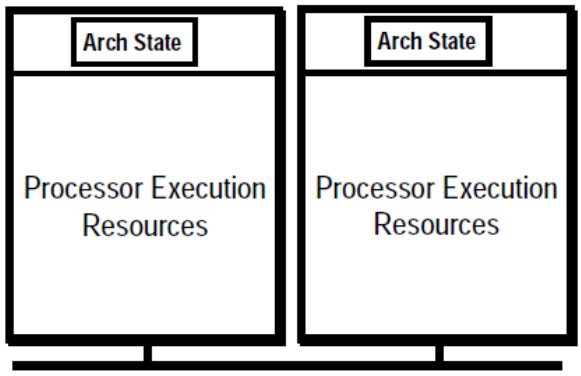

Fig. 2: Processors without Hyper-Threading Architecture

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries. Xeon is a trademark of Intel Corporation or its subsidiaries in the United States and other countries. As an example, Figure 2 shows a multiprocessor system with two physical processors that are not Hyper-Threading Technology-capable. Figure 3 shows a multiprocessor system with two physical processors that are Hyper-Threading Technology-capable. With two copies of the architectural state on each physical processor, the system appears to have four logical processors.

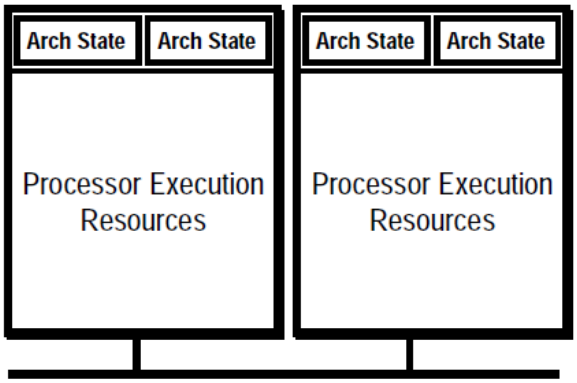

Fig. 3: Processors with Hyper-Threading Technology

The first implementation of Hyper-Threading Technology is being made available on the Intel Xeon processor family for dual and multiprocessor servers, with two logical processors per physical processor. By more efficiently using existing processor resources, the Intel Xeon processor family can significantly improve performance at virtually the same system cost. This implementation of Hyper-Threading Technology added less than 5% to the relative chip size and maximum power requirements, but can provide performance benefits much greater. Each logical processor maintains a complete set of the architecture state. The architecture state consists of registers including the general-purpose registers, the control registers, the advanced programmable interrupt controller (APIC) registers, and some machine state registers. From a software perspective, once the architecture state is duplicated, the processor appears to be two processors. The number of transistors to store the architecture state is an extremely small fraction of the total. Logical processors share nearly all other resources on the physical processor, such as caches, execution units, branch predictors, control logic, and Buses. Each logical processor has its own interrupt controller or APIC. Interrupts sent to a specific logical processor are handled only by that logical processor.

4. SINGLE-TASK AND MULTI-TASK MODES

To optimize performance when there is one software thread to execute, there are two modes of operation Referred to as single-task (ST) or multi-task (MT). In MT-mode, there are two active logical processors and some of the resources are partitioned as described earlier[9-11]. There are two flavors of ST-mode: single-task logical processor 0 (ST0) and single-task logical processor 1 (ST1). In ST0- or ST1-mode, only one logical processor is active, and resources that were partitioned in MT-mode are re-combined to give the single active logical processor use of all of the resources. The IA-32 Intel Architecture has an instruction called HALT that stops processor execution and normally allows the processor to go into a lower power mode. HALT is a privileged instruction, meaning that only the operating system or other ring-0 processes may execute this instruction. User-level applications cannot execute HALT. On a processor with Hyper-Threading Technology, executing HALT transitions the processor from MT mode to ST0- or ST1-mode, depending on which logical processor HALT is executed. For example, if logical processor 0 executes HALT, only logical processor 1would is active; the physical processor would be in ST1-mode and partitioned resources would be re-combined giving logical processor 1 full use of all processor resources. If the remaining active logical processor also executes HALT, the physical processor would then be able to go to a lower-power mode. In ST0- or ST1-modes, an interrupt sent to the halted processor would cause a transition to MT-mode. The operating system is responsible for managing MT-mode transitions.

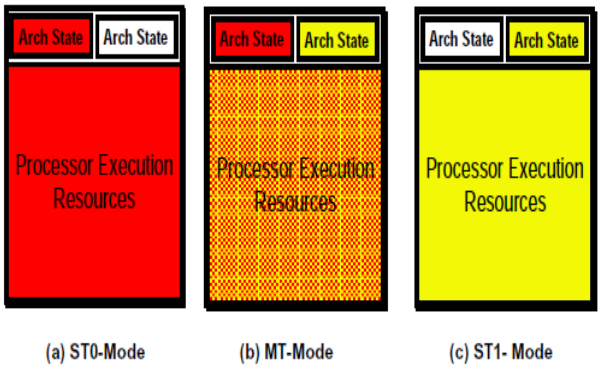

Fig. 4: Resource Allocation

Figure 4 summarizes this discussion. On a processor with Hyper-Threading Technology, resources are allocated to a single logical processor if the processor is in ST0- or ST1-mode. On the MT-mode, resources are shared between the two logical processors.

S-JPSET : ISSN : 2229-7111, Vol. 3, Issue 1

5. MEDIA APPLICATION ON HYPER-THREADING TECHNOLOGY

To date, computational power has typically increased over time because of the evolution from simple pipelined designs to the complex speculation and out-of-order execution of many of today's deeply-pipelined superscalar designs[12-13]. While processors are now much faster than they used to be, the rapidly growing complexity of such designs also makes achieving significant additional gains more difficult. Consequently, processors/systems that can run multiple software threads have received increasing attention as a means of boosting overall performance. Our goal is to provide a better understanding of performance improvements in multimedia applications on processors with Hyper-Threading Technology. Figure 5 shows a high-level view of Hyper-threading Technology and compares it to a dual-processor system. In the first implementation of Hyper-Threading Technology, one physical processor exposes two logical processors. Similar to a dual-core or dual-processor system, a processor with Hyper-Threading Technology appears to an application as two processors. Two applications or threads can be executed in parallel. The major difference between systems that use Hyper-Threading Technology and dual-processor systems is the different amounts of duplicated resources. In today's Hyper-Threading Technology, only a small set of the micro architecture state is duplicated , while the front-end logic, execution units, out-of-order retirement engine, and memory hierarchy are shared. Thus, compared to processors without Hyper-Threading Technology, the die size is increased by less than 5%. While sharing some resources may increase the latency of some single threaded applications, the overall throughput is higher for multi-threaded or multi-process applications.

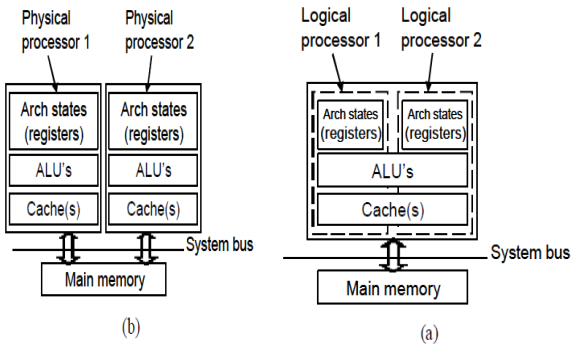

Fig. 5: High-level diagram of (a) a processor with Hyper-Threading Technology and (b) a dual-processor system

6. PERFORMANCE

The Intel Xeon processor family delivers the highest server system performance of any IA-32 Intel architecture processor introduced to date. Initial benchmark tests show up to a 65% performance increase on high-end server applications when compared to the previous-generation Pentium® III Xeon™ processor on 4-way server platforms. A significant portion of those gains can be attributed to Hyper-Threading Technology.

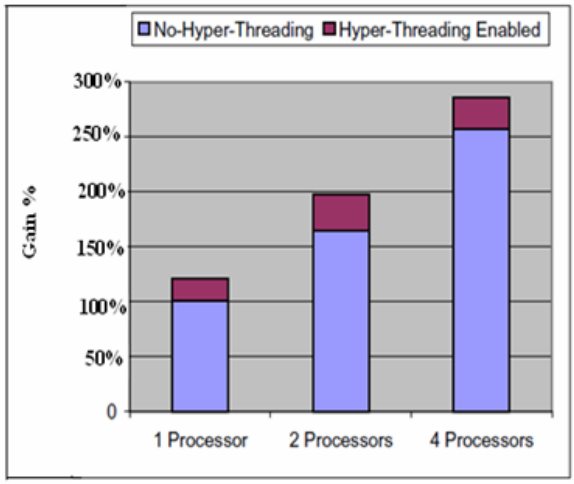

Fig.6: Performance increases from Hyper-threading technology

Figure 6 shows the online transaction processing performance, scaling from a single-processor configuration through to a 4-processor system with Hyper-Threading Technology enabled.

This graph is normalized to the performance of the single-processor system. It can be seen that there is a significant overall performance gain attributable to Hyper-Threading Technology, 21% in the cases of the single and dual processor systems.

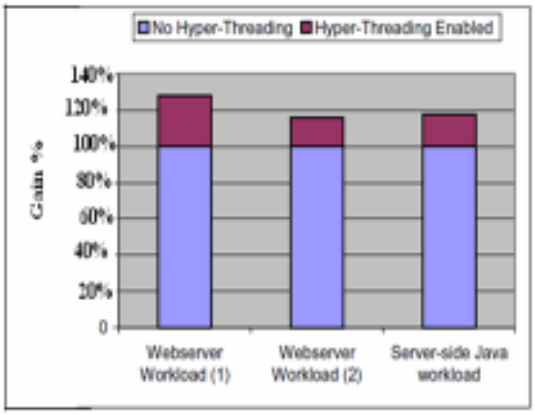

Fig. 7: Web server benchmark performance

Figure shows the benefit of Hyper-Threading Technology when executing other server-centric benchmarks. The workloads chosen were two different benchmarks that are designed to exercise data and Web server characteristics and a workload that focuses on exercising a server-side Java environment. In these cases the performance benefit ranged from 16 to 28%.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

7. CONCLUSION

Intel Xeon Hyper-Threading is definitely having a positive impact on Linux kernel and multithreaded applications.

Today with Hyper-Threading Technology, processor-level threading can be utilized which offers more efficient use of processor resources for greater parallelism and improved performance on today's multi-threaded software. The potential for Hyper-Threading Technology is tremendous; current implementation has only just begun to tap into this potential. Hyper-Threading Technology is expected to be viable from mobile processors to servers; its introduction into market segments other than servers is only gated by the availability and prevalence of threaded applications and workloads in those markets.

REFERENCES

[1] D. Marr, F. Binns, D. L. Hill, G. Hinton, D. A. Koufaty, J. A. Miller, and M. Upton, "Hyper-Threading Technology Microarchitecture and Architecture," *Intel Technology Journal*, Vol. 6, Q1, 2002.

[2] J.H. Chow, L. E. Lyon, and V. Sarkar, "Automatic Parallelization for Symmetric Shared-Memory Multiprocessors, in *Proc. Of CASCON*, pp. 76-89, Nov. 1996.

[3] M. J. Wolfe, *High Performance Compilers for Parallel Computers*, Addison-Wesley Publishing Company, Redwood City, CA, 1996.

[4] D. M. Tullsen, S. J. Eggers, and H. M. Levy, "Simultaneous Multithreading: Maximizing On-Chip Parallelism", In Proc. of *Int'l Symp. on Computer Architecture*, pp. 392-403, Jun. 1995

[5] A. Snavely, D. M. Tullsen, and G. Voelker, "Symbiotic Jobscheduling with Priorities for a Simultaneous Multithreading Processor", Proc. of International Conference on Measurement and Modeling of Computer Systems, June, 2002

[6] D. Tullsen, S. Eggers and H. Levy, "Simultane-ous Multithreading: Maximizing On-Chip Paral-lelism. In 22nd Annual International Symposium on Computer Architecture, pp. 392-403, June, 1995.

[7] L.A. Barroso et. al., "Piranha: A Scalable Architecture Based on Single-Chip Multiprocessing," in *Proceedings of the 27th Annual International Symposium on Computer Architecture*, Pages 282 - 293, June 2000.

[8] D. Tullsen, S. Eggers, and H. Levy, "Simultaneous Multithreading: Maximizing On-chip Parallelism," in *22nd Annual International Symposium on Computer Architecture*, June 1995.

[9] Guy G.F. Lemieux, "Hardware Performance Monitoring in Multiprocessors", Department of Electrical and Computer Engineering, University of Toronto,1996.

[10] L. Hammond, B. Nayfeh, and K. Olukotun, "A Single-Chip Multiprocessor," *Computer*, 30(9), 79 - 85, September 1997.

[11] B.J. Smith, "Architecture and Applications of the HEP Multiprocessor Computer System," in *SPIE Real Time Signal Processing IV*, Pages 2 241 - 248, 1981.

[12] Y.K. Chen, M. Holliman, E. Debes, S. Zheltov, A. Knyazev, S. Bratanov, R. Belenov, and I. Santos, "Media Applications on Hyper- Threading technology," *Intel Technology Journal*, Q1 2002.

[13] E. Su, X. Tian, M. Girkar, G. Haab, S. Shah, and P. Petersen, "Compiler Support for Workqueuing Execution Model for Intel SMP Architectures", in *Proc. of European Workshop on OpenMP(EWOMP*), Sep. 2002.

S-JPSET : ISSN : 2229-7111, Vol. 3, Issue 1