Characterization of Multithreaded Application Workloads on Intel Processors

M.Shanthi[1], George T. Manohar[2]

ABSTRACT

This paper describes how to speed up a commonly used application by taking advantage of multithreading on Hyper-Threading Technology base Intel® Pentium® 4 processors. The techniques shown here can be applied to any application where parallelization is targeted. The performance gains will, of course, vary with the applications as well as system configuration. Multithreading is suitable for all applications where optimization is the subject for program performance. This paper shows a few simple applications for the threaded scenario. We will also present performance numbers for comparing the application with and without threading concept. All performance data presented was collected on a 3.2 GHz Intel® Pentium® 4 processor with Hyper-Threading Technology.

Keywords : Hyper-Threading, Multithreading, Performance, Threading

1. INTRODUCTION

Performance of a computer system is a complex, difficult and important task. Time and rates are usually the basic measures of system performance. Having more realistic data about system performance will be necessary in the design and development of integrated information systems and systems for reuse. The main objective of this work is to measure the performance of various Web-based computer systems and non Web based. A thread is a single line of execution within the process of application. That means that inside the application, more than one thing can happen at the same time virtually. All threads can have direct access to the data in application. Here are some typical situations where threads are used: To move a time-consuming initialization task out of the main thread, so that the GUI comes up faster. Examples of time-consuming tasks include making extensive calculations and blocking for network or disk I/O (loading images, for example).

Multithreading is the technology through which we can develop applications that perform concurrent tasks. Multithreading is effective on both single-processor and multiprocessor systems. Today's traditional single-core processors can only process one thread at a time, spending a majority of time waiting for data from memory.

Multithreaded and distributed computing is gaining a wide popularity in the area of high performance computing. Availability of high performance computer networks and sophisticated software environments are allowing performing parallel/concurrent computing on commodity hardware. Recently, threads have become powerful entities to express parallelism on this shared memory multiprocessor (SMPs) and multi computer (MPPs/ Clusters) systems. The Power of Multithreading determining how the code should run, but it could be important for licensed software where the number of physical processors determines the cost of the application,

[1]Deptartment of Computer Application, Velammal Engineering College,Chennai, , Tamilnadu,India-600 066. e-mail: sha_karnan@yahoo.com

[2]Dept. of Electrical Engg., IIT Madras,Chennai, India-600 036, e-mail : sha_karnan@yahoo.com

such as the operating system seat licenses. There isn't a significant performance difference between a multithreaded application and multiple single threaded processes as far as HT Technology is concerned, so long as the CPU utilization isn't bunched up within a single process.

2.Previous work

Paper[3] discussed that Http servers can be programmed easier by using multithreading, multithreading can be used to hide the long latency of a remote memory access. Paper [4] stated to increase processor core execution efficiency by executing instructions from two or more tasks simultaneously in the functional units in order to increase the execution rate of instructions per cycle (IPC). These processors implement simultaneous multi-threading (SMT), which increases processor efficiency through thread-level parallelism, [11] High performance computing (HPC) environments, where applications can benefit from the Java multithreading support for performing parallel calculations, or e-business environments, where multithreaded Java application servers (i.e. following the J2EE specification) can take profit of Java multithreading facilities to handle concurrently a large number of requests. In the second thesis[13],the results indicate that processors can substantially benefit from multithreading, even in systems with small caches, provided sufficient network bandwidth exists. Increased network contention due to multithreading has a major effect on performance. their experimental results show that by taking advantage of SMT technology achieve a 30% to 70% improvement in throughput over single threaded implementations on in-memory database operations.These threads execute tasks according to their structure with a simple scheduling. The results of paper[17] suggest that multithreading with a small number (3-5) of active threads can significantly improve the performance of such commercial environments.

3. Methodology

This paper presents a threading methodology for design, implementation and debugging threads, which has been successfully used and refined on many large applications. The threading methodology consists of three phases. The first phase focuses on performance analysis in order to identify performance optimization opportunities with the goal of optimizing the serial version of the application. The second phase involves writing application without threads.. The third phase includes threading concepts. Finally, the fourth phase looks for performance improvements.

Steps in Threading Methodology

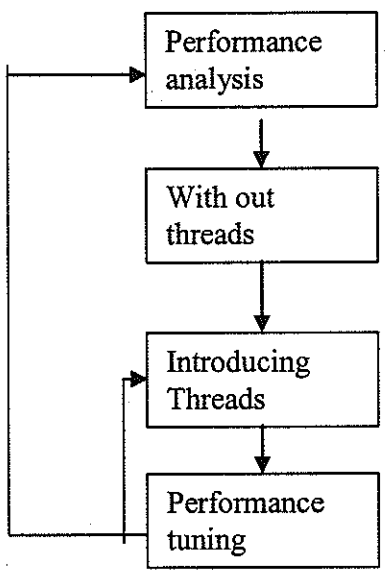


Figure 1

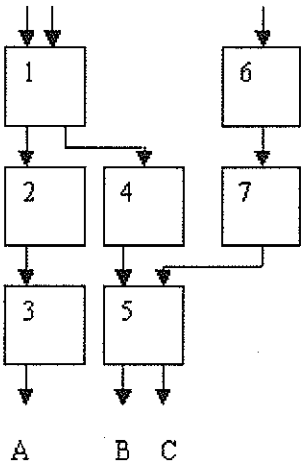


Figure 2

Consider fig 2 in which each block represents a task and each task has a unique number. As required by the application task2 can be executed only after task 1 is completed and task3 can be executed only after task1 and task2 are both completed. Thus the line through task 1,2,3 forms a thread A of execution. Two other threads B and C are also shown. The thread limits the execution of task to a specific serial manner. Although the task2 has to follow task1 and task3 has to follow task2, Task 4 can be executed in parallel with task2 or 3.similarly task 6 can be executed simultaneously with task 1 and so on . suppose the MIMD processor has three processors and if task takes the same amount of time to execute , the seven tasks shown in the figure can be executed in the following manner.

Table 1

Time slot	P1	P2	P3
1	Task1	Task 6	-
2	Task 2	Task 4	Task 7
3	Task 3	-	Task 5

Here a program is divided into many independent threads. Each thread has an independent stack space but shares a common global address space.

Pentium Processor

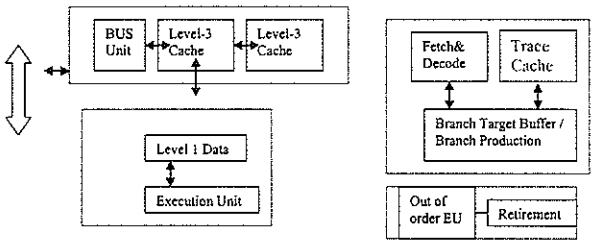


Figure 3

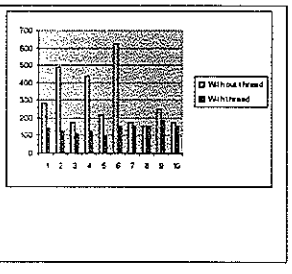
4. Experimental Study And Observations.

Threading in Banking Applications

Program is written in Java with and without threads for banking application. When we run the program, program with thread takes less time than the program with out thread. In this application we considered only two operations deposit and withdraw. Database contains 20 records. The time is varying based on the number of records and the amount deposit. For example the following table shows the data for deposit of Rs.1000. The program is run in Pentium processor with HT supported machine and database used ms access and java swing.

Table 2

S.No	With out thread	With thread
1	281	140
2	485	125
3	172	109
4	437	125
5	219	94
6	625	156
7	172	156
8	157	152
9	250	187
10	171	156



If average is considered 296ms for with out thread and 140 ms for with thread. From the above table it is

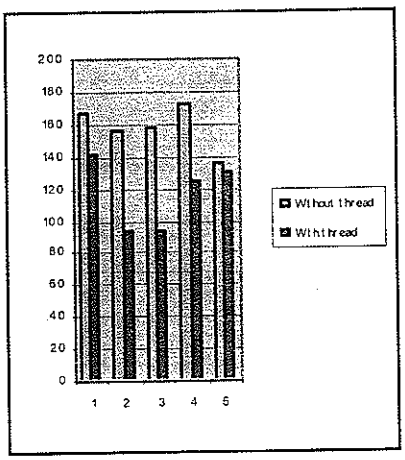
observed that threading is really advantage in all types of applications.

Threading In Client Server Applications

The evolution of the multi-threaded processor design is the trend for next generation desktop processors. To add a record from client side, a program is written and executed in Java with and without thread. The results are tabulated.

Table : 3

S.No	Without thread	With thread
1	167	141
2	156	94
3	157	93
4	172	125
5	135	130



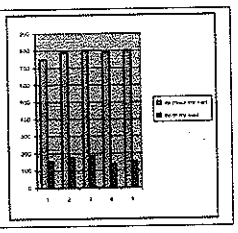
If we compare the average value of two, program with threading takes the advantage. So it is evident that threading in client server application is faster always irrespective of number of records.

Threading In Scientific Applications

The following are sample data which shows the results for random number generation up 3000 numbers. The program is written in Java with and without threads. The program is executed in Pentium HT machine and the time taken by the CPU in milliseconds is listed below. It is observed that threading takes the advantage.

Table 4

S.No	Without thread	With thread
1	750	156
2	796	172
3	797	187
4	796	141
5	797	156

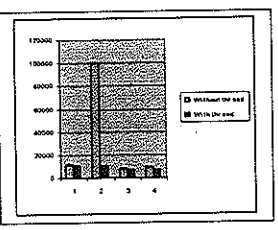


Threading In Producer Consumer Applications

Two application programs for producer consumer are written in Java. In the first program , Main itself a thread and main forks another thread .this thread deviates in to producer and consumer. The other program , Main is a thread which interleaves into two threads into producer and consumer. The first program takes 4125ms and the second takes 4890 ms.

Table 5

S.No	Without thread	With thread
1	11114	11058
2	99750	10625
3	7969	7406
4	8828	7400



Threading In Multimedia Applications

Threading is really advantages because it increases the processor time irrespective of the image format. This is observed by writing a multimedia program in Java for loading the images. The program is written with and with out threading concept and executed in Pentium IV Processor with HT enabled systems. Based on the tabulated values, it was evident that threading gives better performance.

Table 6

S.No	No of threads	Time in ms	CPI
1	Without thread	203	4.7
2	Two	188	3.4

5. Conclusion and Future Enhancements

This is only an introduction on how to get an easy performance gain using thread to take advantage of Hyper-Threading Technology on an Intel Pentium 4 processor. The coding techniques presented here are applicable to all applications . To gain real benefits from multi-Threading Technology many times, re-architecting the high level algorithm for load balancing is necessary. In future threading tools can be developed to take advantage of multithreading in all the applications like data mining ,multimedia and bio-medical applications.

References

[1] A.K Ray, K.M. Bhurchandi, *"Advanced Microprocessors and peripherals"*.

[2] Shivya, Taylor and Francis, *"Advanced Computer Architecture"*.

[3] J. Keller and O. Monien (Germany), *"Improving HTTP-Server Performance by Adapted Multithreading,"* July 2005.

[4] Nikola Vouk Buddy, *"Threading in Distributed Applications"*, on SMT-2005.

[5] Jordi Guitart Fernández, *"Performance Improvement of Multithreaded Java Applications Execution on Multiprocessor Systems,"*, 2005.

[6] Effects of HTT in the response time of business applications white paper 2004.

[7] D. Marr, F. Binns, D. L. Hill, G. Hinton, D. A.Koufaty, J. A. Miller and M. Upton, *"Hyper - Threading Technology Micro architecture and Performance"*, Intel Technology Journal, Q1 2002.

[8] E. Palmer, *"Hyper-Threading Characterization"*, private communications, Apr. 2002.

[9] H. Wang, P. Wang, R. D. Weldon, S. Ettinger, H.Saito, M. Girkar, S. Liao and J. Shen, *"Speculative Precomputation: Exploring the Use of Multithreading Technology for Latency"*, Intel Technology Journal, Q1 2002.

[10] Yen-Kuang Chen, Matthew Holliman and Eric DebesnI, *"Video Applications on Hyper-Threading Technology"*, EEE International Conference on Multimedia and Expo 2002, Lausanne, Switzerland, August 2002.

[11] J. Manson and W. Pugh, *"Core multithreaded semantics for Java"*, In Proceedings of the Joint ACM Java Grande - ISCOPE 2001 Conference, Stanford, CA, June 2001.

[12] Susan Eggers, Joel Emer, Henry Levy, Jack Lo Rebecca Stamm and Dean Tullsen, *"Simultaneous Multithreading : A Platform for Next-generation Processors"*, IEEE Micro, International Journal PP.516 - 524

[13]Pop, R, Kumar. S, Dept. of Electron. & Comput. Eng., Jonkoping Univ, On Performance Improvement of Concurrent Applications Using Simultaneous Multithreaded Processors as NoC Resources.

[14] Talla. D, John, L.K.Execution, *"characteristics of multimedia applications on a PentiumII processor*

Performance, Computing, and Communications", Conference, 2000. IPCCC apos;00. Conference Proceeding of the IEEE International Journal, PP.516 - 524, Feb 2000.

[15] R. Rugina and M. Rinard, *"Pointer analysis for multithreaded programs",* In Proceedings of the SIGPLAN '99 Conference on Program Language Design and Implementation, Atlanta, GA, May 1999.

[16] R. J. Eickemeyer, R. E. Johnson, S. R. Kunkel, M. S. Squillante and S. Liu, *"Evaluation of multithreaded uniprocessors for commercial application environments",* In Proceedings of the 23rd Annual International Symposium on Computer Architecture, PP. 203-212, May 1996.

[17] M. Gulati and N. Bagherzadeh, *"Performance study of a mulltithreaded superscalar microprocessor. 1996."*

Authors Biography

Mrs M.Shanthi, doing her Ph.D in Mother Teresa Women's University, Kodaikanal and working as a Assistant Professor , MSc (IT) Dept in Velammal Engineering College, Chennai. She is having an academic experience of about 16 years and presented papers in national and international conferences and published research papers and articles in Magazines and Journals. She has published solved question paper for MCA students. (Anna University). She was also a Chair person for MSc (CS&T) five years course in Madras University.

Dr. G.T. Manohar, Retired Professor in Department of Electrical and Electronics Dept, IIT, Chennai. He has presented papers in national and international conferences and published research papers and articles in Magazines and Journals.