# Evaluating Hyperthreading and VMware

Vuk Marojevic

Department of Computer Architecture
Polytechnic University of Catalonia
marojevic@tsc.upc.es

**Abstract.** The Hyperthreading technology has been introduced to enhance CPU resource utilization so as to increase overall system performance. We briefly present this technology and evaluate its impact on CPU performance by means of a small test series. On the software side, VMware is becoming increasingly of interest in the private and professional sectors, allowing the installation of various operating systems on the same physical disk partition. Practically this means that you can change from one operating system to another as with applications. We shortly present VMware and discuss performance measurements in different operating environments.

## 1 Introduction

### 1.1 Hyperthreading

Hyperthreading (HT) emerged as a technique to optimize CPU resource utilisation and is included in the latest Intel processors. It is a technology, also known as simultaneous multithreading (SMT), enabling the simultaneous processing of multiple threads of different processes. There is still only one physical processor, but from an operating system (OS) and user perspective, a simultaneously multithreaded processor is split into two logical processors, and threads can be scheduled to execute on any of the logical processors just as they would on either processor of a symmetric multiprocessor (SMP) system.

The HT technology was introduced to overcome the waste of resources of microprocessor units. For instance, as the instruction level parallelism (ILP) that can be extracted from most code is statistically around 2.5 [1], a microprocessor with resources to handle 4 instructions in parallel will mostly deal with 1 to 3 at a time and rarely reach 100% of the correspondent resource utilization. In relation to that, empty pipeline stages of execution core's functional units, called pipeline bubbles, remain. A microprocessor supporting HT, on the other hand, can interleave two or more threads of different programs and thus, decrease the number of missed opportunities for useful work [1], [2].

There are and have been many discussions on the actual benefits of HT. Resuming them in one sentence, applications may or may not benefit from this technology depending on a multitude of factors including application peculiarities as well as system (software and hardware) characteristics. Even a performance decrease is documented in certain scenarios. Having heard so many different declarations

regarding HT from different interest groups motivated us to schedule a small practical study. We expose here our results and conclusions, while also mentioning the problems associated with this technology and the proposed solutions.

### 1.2 VMware

VMware was originally founded in 1998 to bring mainframe-class virtual machine technology to industry-standard computers [4]. Among its products are server and workstation solutions, the latter one being of interest for this study. Launched in 1999, VMware Workstation works by enabling multiple operating systems (OSs) and their applications to run concurrently on a single physical machine. These operating systems and applications are isolated in secure virtual machines that co-exist on a single piece of hardware. The VMware virtualization layer maps the physical hardware resources to the virtual machine's resources, so each virtual machine has its own CPU, memory, disks, I/O devices, etc [3]. The OSs above and beneath the virtualization layer are called *guest* and *host* operating systems, respectively. This so-called hosted architecture (figure 1) relies on the host OS and, therefore, does not have to handle all the hardware abstractions. Once VMware is installed and executed, it runs like an ordinary application. More information on VMware and downloads can be found in [3].

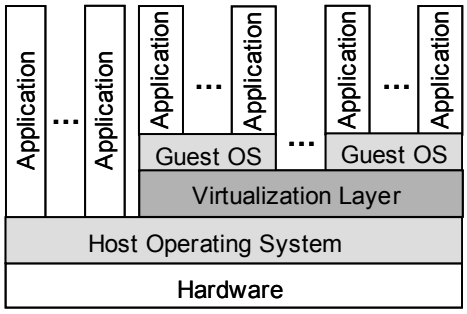


**Fig. 1.** Hosted architecture

The motivations for studying VMware are apparent when studying operating environments for parallel application resource management. VMware manages resources for the concurrent execution of various operating systems which can be considered as one step beyond the resource management of parallel applications. Due to its advocates it can be a very useful tool in many different scenarios ranging from the private sector to the professional software development area. In this study we are focused on the private and professional sectors, where VMware is mainly used to change operating systems without repartitioning disks or rebooting so as to run applications concurrently in both worlds. Then the question arises how efficient it is working in a guest operating system world. This, of course, is application and environment dependent and a general answer cannot be found. Instead, what we want

to explore here is how it behaves in some particular scenarios and try to give an, as far as possible, generalised outlook.

Before heading forward we have to outline that this work basically entertains *two* studies to evaluate Hyperthreading and VMware independently. To date VMware does not support HT. Irrespectively of which OS runs beneath or above of VMware, each guest OS will see only one logical processor. This is also the case in SMP architectures. Nevertheless, a third study investigates the effect HT has on CPU performance while running VMware.

## 2 Objectives and Measurement Environments

### 2.1 CPU performance - HT

In this first study we want to find out if and how much HT increases or decreases CPU performance. Therefore, we created the environments that are shown in figure 2. We use some of the SiSoftware Sandra Benchmarks [5] - described as follows - and run the test with and without having HT enabled.

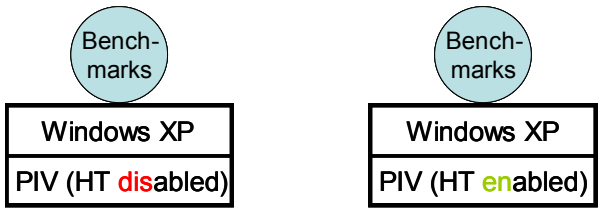


**Fig. 2.** Test environments I

The selected benchmarks evaluate the CPU performance. They all support multi-threading (MT) and, therefore, a performance increase enabling HT is a priori expected. We do five tests: The first, *Dhrystone ALU*, measures the time it takes to perform some sequences of instructions, mostly numerical ones that are used by applications. The result is given in million instructions per second (MIPS). The next two, *Whetstone iSSE2* and *Whetstone FPU*, measure the time it takes to perform some sequences of floating-point instructions, where iSSE2 supports double floats (64-bit). They are optimised for Intel Pentium 4 and return the result in million floating point instructions per second (MFLOPS). Finally, the numbers associated to the *Integer iSSE2* and *Float iSSE2* tests represent the number of iterations per second (it/s) while generating a picture (860x750 pixels) of the Mandelbrot fractal, using 255 iterations for each data pixel in 32 colours. *Integer iSSE2* is using integers to simulate floating point numbers. These tests, categorised as *CPU Multimedia* benchmarks, are real-life benchmarks rather than synthetic ones [5]. The five benchmark tests described above will be repeated throughout the whole work.

The underlying PC, which is the same for all tests, is a Pentium 4 (PIV) with the following characteristics: 2,8 GHz processor speed, HT support, 512 KB L2 Cache, Intel Chipset 875, 800 MHz System bus bandwidth, 512 MB RAM.

### 2.2 CPU performance - VMware

In the second part we take a look at VMware. VMware is gaining wide popularity as it allows running various OS without disk repartition and rebooting. This motivated us to check if it is really that useful and how much CPU performance loss is associated with working in the guest OS world. Therefore we run the same tests as in the first study in the following operating environments.

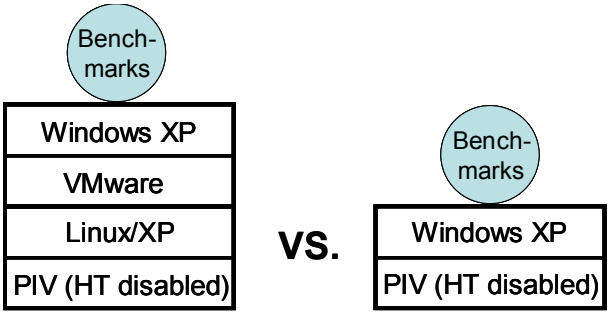


**Fig. 3.** Test environments II

HT is disabled (HT-) for this test series which is run in both the setups shown in figure 3 so as to compare the results obtained in the guest versus host OS worlds.

### 2.3 CPU performance – HT/ VMware

Finally, we study the effect of HT on CPU performance while working in the guest OS world and in both, guest and host worlds. In the host OS world we first run VMware only and then VMware together with some additional applications (see figure 4). Also, we enable and disable HT in each of the two environments. For all four scenarios the discussed benchmark test are scheduled in guest OS world. Later when comparing these results we might be able to make some additional conclusions on the performance of VMware as well as on HT. Mainly, we want to observe if VMware can take advantage of the HT technology, and if not, maybe with some additional active processes a performance gain can be obtained when having HT enabled (HT+).

As the additional applications we choose having two web browser windows open with more or less animated web pages (including radio transmission). Although they use only 0-3% of the CPU when running in the background, it is somehow a realistic and reproducible scenario. The task manager and system monitor, respectively, is also running in the host OS, while performing the benchmark tests in the guest OS.

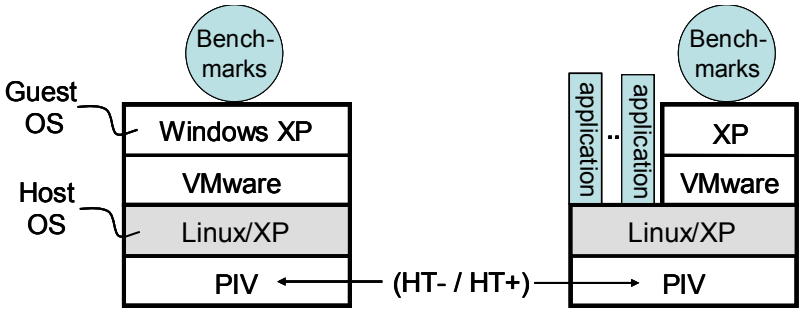


**Fig. 4.** Test environments III

# 3 Results

## 3.1 CPU performance – HT: Results

Figures 5 and 6 show the results obtained for the CPU and CPU Multimedia benchmarks for the setup described in part 2.1. In all five measurements we can observe significant CPU performance improvements when having HT enabled. This, however, is no that surprising as all the tests support multithreading (MT). There are differences in the improvements, though, as each test evaluates a different feature. For instance, the different percentage gains of the Whetstone (+65/72%) in respect to the Dhrystone (+18%) as well as the Float iSSE2 (+41%) in respect to Integer iSSE2 (+24%) benchmarks can be explained as follows: FPU units were underutilised in the original Pentium 4 and, thus, they get good improvement in SMT (~50%). The ALUs, on the other hand, were better utilised and gave great performance already and, therefore, they get some improvement only (~20%) [5].

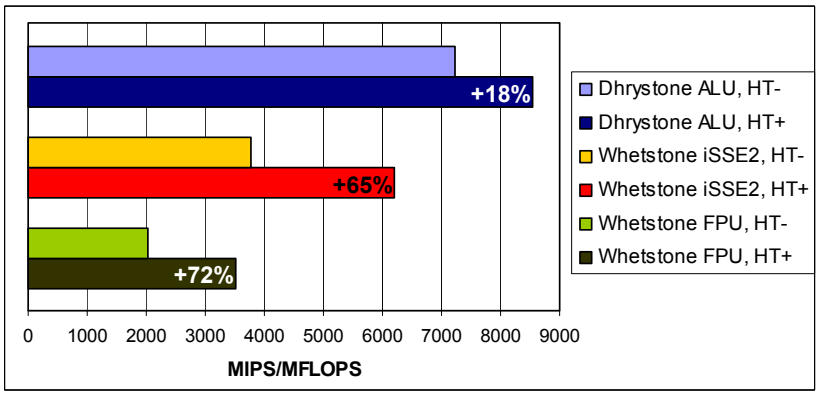


**Fig. 5.** CPU Arithmetic Benchmark, test environments I

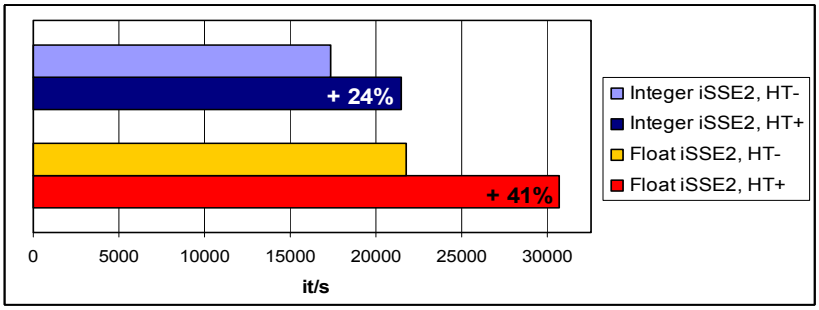


**Fig. 6.** CPU Multimedia Benchmark, test environments I

### 3.2 CPU performance – Vmware: Results

Figures 7 and 8 show the results obtained for the CPU and CPU Multimedia benchmarks for the setup described in part 2.2. We can observe a CPU performance loss of about 9-15% when working in the guest in respect to the host Os worlds for the particular test series. These measurements also unsheathe some OS dependence. In this sense the PIV-Windows-VMware-Windows environment leads to better results than PIV-Linux-VMware-Windows. Windows (WIN in the figures) is referring to *Windows XP Professional Edition*, while Linux represents *Red Hat 9* with the *2.4.20 kernel*. It would have been interesting to have the results with the latest Linux kernel. Due to problems with the configuration of VMware for the 2.6.5 kernel, we could not obtain these numbers before the paper submission deadline.

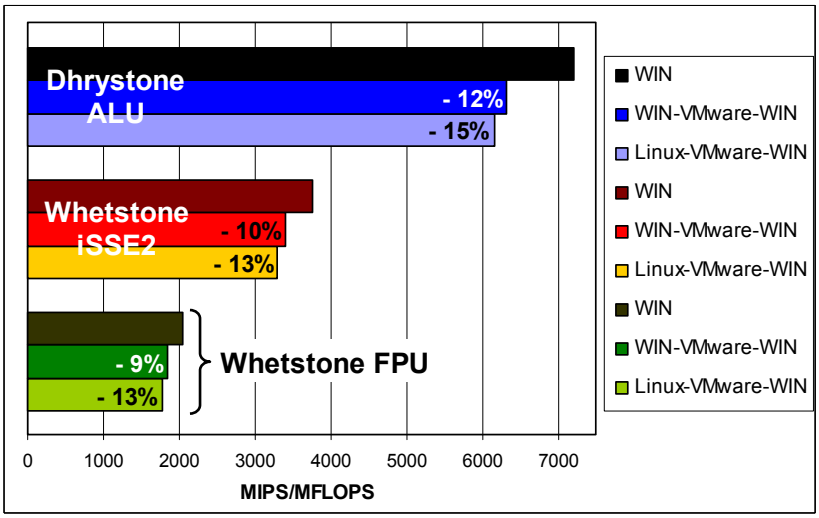


**Fig. 7.** CPU Arithmetic Benchmark, test environments II

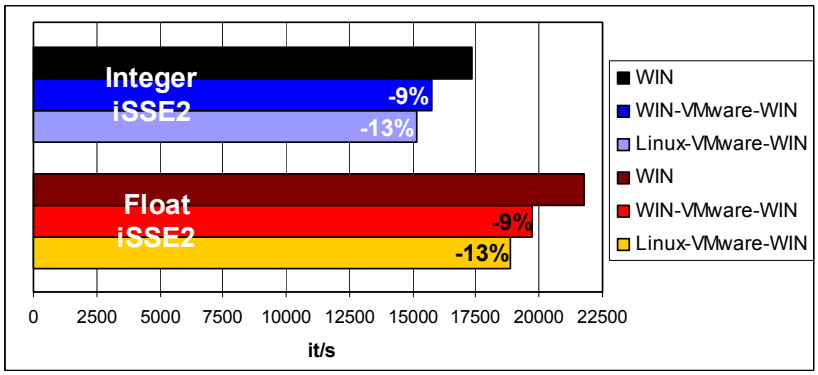


**Fig. 8.** CPU Multimedia Benchmark, test environments II

### 3.3 CPU performance – HT/ Vmware: Results

The following figures illustrate the results obtained for the CPU and CPU Multimedia benchmarks for one part of the setup described in part 2.3. More precisely, only the results for the right setup in figure 4 with Linux as the host OS are shown in the figures. Before taking a look at them, we present what we have obtained for the operating environment on the left of figure 4.

With Windows as the host OS, enabling and disabling HT has lead to negligible differences in the obtained results. With Linux as the host OS, enabling HT has yielded a performance increase of around 5-7%. For the operating environment on the right of figure 4, however, the things look different (see figures 9 and 10). There we have a performance increase of more 10%. Very similar results have been obtained with Windows as the host OS. From these results we conclude that HT does not result in a performance decrease when having one single application running that is not multithreaded or not optimized for multithreading but rather in no effect at all. On the other hand, having more than one application running, HT has some considerably positive effect on CPU performance, even when the applications are not supporting or are not optimized for multithreading. These last two statements, of course, apply to the specific test series only. A generalization cannot be made, although it is expected that there are many more scenarios where similar observations can be made. Finally, we dare to estimate even better results with the latest Linux kernel, as it has some more optimisations at OS level for HT supported CPUs.

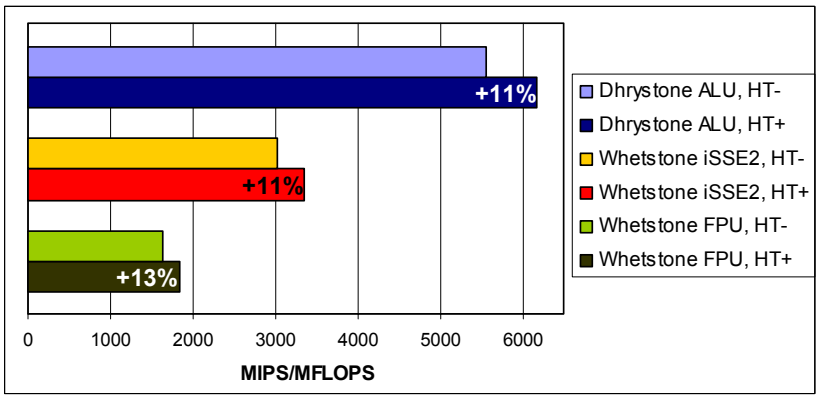


**Fig. 9.** CPU Arithmetic Benchmark, test environments III

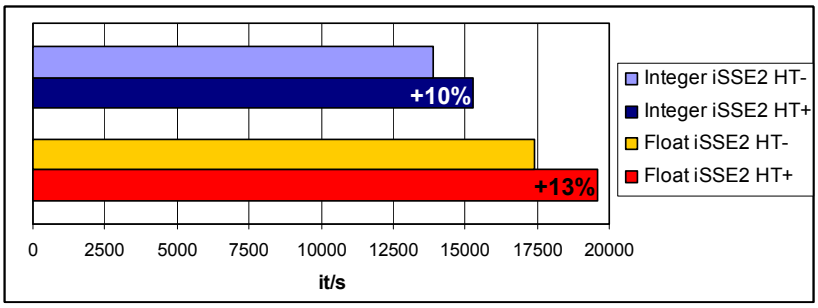


**Fig. 10.** CPU Multimedia Benchmark, test environments III

## 4 Conclusions

These particular benchmark results show that HT increases CPU performance when the application (in this case the benchmark) supports multithreading (MT). Having HT enabled and running a single application that does not support or is not optimized for MT, as is the case with VMware (which is running like an ordinary application inside the host OS world), generally does not have any effect at all. On the other hand, as shown in this work and also stated by Intel [2], having two or more applications running on a processor with HT enabled a performance increase can be expected, even when those applications do not support MT. A performance decrease is documented, although not observed in the scheduled tests, when running heavy-duty floating point operations, since there is still only one floating point unit. In this case, it is recommended to disable HT. Another problem that has been discussed in [6] is when executing applications that need lots of concurrent memory accesses, the performance can be bus bandwidth bound. The problems occurred there having system bus speeds of 400 MHz, whereas with 533 MHz these problems disappeared. We have run a benchmark that evaluates memory bandwidth and another that

evaluates cache and memory together on our PIV while enabling and disabling HT. The obtained results were identical, as we have expected having a system bus bandwidth of 800 MHz. In future, this might become a problem again when the processing speed grows unproportional to system bus speeds. Finally to conclude this part, while the benchmarks we have looked at here have shown improvements of the HT technology, it is important to do other testing with the applications used on a daily basis. HT certainly looks auspiciously but there are still situations where HT is going to negatively impact the performance of an application.

As regards VMware, our assessment is that it is indeed a useful tool when dealing with more than one operating system. VMware has the great advantage that applications can execute in concurrently in different OS on the same hardware. An OS switch is fulfilled immediately and the CPU performance loss is, from our point of view, tolerable. For the discussed tests a performance decrease of 9-15 % has been observed. This depends on the OS, the applications running in the guest and host OS worlds and many other factors, like the available physical memory. With insufficient memory or memory intensive tasks we recommend to not use VMware but rather make a new disk partition for each OS one wants to install. There may be more hardware/software limitations, which we will study in further work. As with HT, a general guideline cannot be given. Each user has to decide on behalf on his/her experience and needs whether to use VMware or not.

## References


1. J.H. Stoke: Introduction to Multithreading, Superthreading and Hyperthreading (2002)
2. www.intel.com
3. www.vmware.com
4. J. Sugaerman, G. Venkitachalam, B.-H. Lim: Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor, Proceeding of the 2001 USENIX Annual Technical Conference (2001)
5. www.sisoftware.net
6. www.2cpu.com