

# BENCHMARKING EFFORT OF VIRTUAL MACHINES ON MULTICORE MACHINES

Aparna Venkatraman, Vinay Pandey, Beth Plale, and Shing-Shong Shei  
Computer Science Dept., Indiana University  
{apvenkat, vkpandey, plale, shei} @cs.indiana.edu

IUCS TR 654

## ABSTRACT

*Virtualization software is often run on server machines for reasons of load balancing, flexibility and lower hardware cost. We conducted a performance evaluation of two comparable dual core server machines, one from Dell and another from HP, both running VMWare ESX V3.0. The performance is measured using a series of independent benchmarks, testing the performance of a database server, java server and web server. This multi-workload benchmark is used to evaluate the two server machines and study how the different server workloads interact to utilize the hardware resources.*

## KEYWORDS

Virtualization, server benchmark, benchmarking, performance

## 1.0 INTRODUCTION

Benchmarking is a useful tool in determining the performance of a particular piece of technology. In this paper we describe a benchmarking study to evaluate the performance of several benchmark applications running in virtual machine hosted on a multi-core processor machine. This study is carried out at the offering of our sponsor, the Office of Vice President of Information Technology, Enterprise Infrastructure at Indiana University. Enterprise Infrastructure has deployed virtual machines in production services for several years, in particular, VMware's ESX Server [9] for purposes of running numerous servers on the same physical machine. Our study examines two comparable platforms and analyzes workload combinations and the utilization of the machines.

Virtualization provides a flexible environment for running applications. On a single physical machine, there can be multiple operating systems running independently while sharing the resources of the physical machine in harmony. Virtual machine software can run either on the bare hardware or on top of a host operating system. On top of that, virtual machines (VMs) can host different operating systems and applications. The number of VMs per physical machine depends on the virtualization software and the hardware capabilities but can grow. This is an advantage because when the need for another server arises, one can just add another VM to a machine, install the appropriate software and the system is ready, as opposed to purchasing a new machine to handle each new server.

Virtualization allows the user to change the allocation of resources through various tools provided by the software vendor. This feature is very important to find the optimum configuration of applications and VMs. For our experiment we compare the performance of two servers, one from Dell and the other from Hewlett Packard, running five VMs each. Three of the five VMs run a database benchmark, a java JVM benchmark, and web server benchmark. and one VM is

used as load generating client for the web server, while the other VM is not being used. The virtualization software, ESX 3.0.1, is provided by VMware, a commercial vendor that has a long history of enterprise virtualization products. We based our server and benchmark server choices on the needs of our client (UIS) and previous virtual machine benchmarks [6–8, 17].

The remainder of the paper is as follows. Section 2 compares how our work relates to previous work on benchmarking VMs. Section 3 provides a thorough description of the experimental platform. Section 4 describes each of the benchmarks we chose to use and why. Section 5 explains the workloads we benchmarked and discusses the results. Section 6 describes the interesting observations and Section 7 addresses the directions for future work and conclusions.

## 2.0 RELATED WORK

Today’s computers are sufficiently powerful to run hundreds of processes at the same time. Therefore, it is now a waste to acquire a new machine for each server process. Virtualization is an important technique for subdividing the resources of a modern computer. The benefits of virtualization include [14]

- improves the utilization of machine resources
- provides secure, isolated sandboxes for running untrusted applications
- provides resource limit constraints and in some cases, resource guarantees
- easy migration of systems

One of VMware’s technical report [21] is similar to the goals of this study. However, the report compares performance of two machines at one fixed configuration and does not utilize the machine heavily. In this paper, we began with a basic configuration and then varied the resource allocation for a VM based on the usage trend and compared the performance of the normal and the altered configurations. VMware [9] is a full system virtualization tool for x86 architecture [13]. Virtualization benchmarking using VMware has been performed by various hardware vendors, such as Dell [11, 12], and HP [3] and VMware [16, 21]. Xen is an open source virtualization tool that runs on Linux platform. The difference between Xen and VMware is that while VMware employed the fully virtualization approach where the exposed functionality of virtual hardware is identical to that of the underlying machine, Xen uses an approach known as paravirtualization where the virtual machine abstraction is similar but not identical to the underlying hardware [8, 13]. Virtualization benchmarking using Xen has been covered in Clark et al [6], and Barham et al [8]. Disco is a full virtualization technique for running commodity operating systems on ccNUMA architectures [7]. Denali is a paravirtualization technique for hosting vast numbers of virtualized OS instances [17].

## 3.0 EXPERIMENT DESCRIPTION

The experimental setup consists of a Dell Inc. PowerEdge 6950, four 2.6 GHz Dual-Core AMD Opteron Processors and 24 GB of memory and a HP Proliant DL585 G2 model, four 2.6 GHz Dual-Core AMD Opteron Processors and 32 GB of memory. Each of these machines is running VMware’s ESX Server 3.0.1 as the demonstration platform for the virtual machines. The 5 disks connected to the HP are configured in RAID Level 5. However, the 5 disks that are connected to Dell are not configured in any RAID Level. We created 5 virtual machines (four running Red Hat Enterprise Linux 4 and one running Windows Server 2003), each with 2 Virtual CPUs and 4 GB of memory. We decided to run only one application per VM as this would represent the majority of the customer usage environment. All measurements shown in the paper were done under simultaneous workload on all the three active VMs. We observed that there was not any significant difference in the performance of the machine or the resource utilization between running each benchmark individually and running it all in parallel. We feel that the VMs do a thorough job in protecting resource usage and ensure that the VM partitions are as good as having a separate physical machine. We have analyzed this in more detail in Section 5.0.

### **3.1 ARCHITECTURE DESCRIPTION - AMD OPTERON PROCESSORS**

Both the machines are powered by 4 AMD Opteron 8218 processors. These processors work on the AMD technology with direct connect architecture and eliminate the bottleneck that is present in traditional front-side bus architecture. An exclusive feature of this architecture is the HyperTransport Technology [20]. HyperTransport technology is a high-speed, bi-directional, low latency, point-to-point communication link that provides a scalable bandwidth interconnect between computing cores, I/O subsystems, and other chipsets. AMD Opteron processors support up to three coherent HyperTransport links, yielding up to 24.0 GB/s peak bandwidth per processor.

The system hardware supports Non-Uniform Memory Architecture (NUMA). Each processor has its own memory controller and local memory for reduced access times, but it can also access memory from another processor. This architecture improves system performance if an operating system is installed that supports this feature [18].

### **4.0 BENCHMARKS**

In Spring 2007 a team of students from a graduate level course in distributed computing at Indiana University began the virtual machine performance evaluation effort of the two dual quad-core server machines. This benchmark included a Database server, Mail server, File server, Java server and Web server, and was designed based on the virtual machine benchmark results described in [21]. Following completion of the class project, the sponsor expressed interest in a deeper performance comparison of the database, java JVM and web server benchmarks. In this paper we present the results of the more comprehensive study.

#### **4.1 DATABASE SERVER**

Swingbench is a freely available load generator (and benchmark) designed to stress test an Oracle Database 10g. It models users repeatedly executing a predefined mix of transactions. Swingbench has two benchmarks OrderEntry and CallingCircle. OrderEntry uses the “oe” schema that ships with Oracle10g. It introduces heavy contention on a small number of tables and is designed to stress interconnects and memory.

Our choice of Oracle as the database platform was guided by a preference from our sponsor. The decision to use Swingbench was on the basis of much literature review and also the fact that it has been designed specifically for Oracle Database. Swingbench runs mainly in two different modes, specifically Swingbench, a GUI Interface for generating extensive network load, and CharBench which allows passing the same parameters in command line. The parameters to the benchmark are easily configurable via a single XML file. Swingbench is also scalable in that it can be extended to stress test using multiple load generators using a coordinator. We choose to have the Process Orders (Select/Update) transactions as 5.5% of total transactions load and New Customer Registration (Select/Insert) transactions as 11%. The transactions Browse Products (Select), Browse Orders (Select) and Order Products (Select/Insert/Update) each contributed around 27.8% of the total transactions. These parameters were decided according to the contemporary databases write and read trends with consensus by our sponsor. The user think time between transactions was kept between 10 and 500 milliseconds.

The benchmark setup consisted of an Oracle 10g (10.2.0.1) Database Server installed on Red Hat Enterprise Linux 4 and a Load Generator Client. Initially, 2 CPU's and 4 GB memory was allocated. The Load Generator Client requires 1.5 JVM, Swingbench and Oracle Client and was run from a separate physical machine. The database instance was approximately 8GB. Application Data files consumed approximately 7 GB and log files and dictionary views accounted for the remaining space. The Load Generator's average load was maintained below 70 percent. Also, the ratio of one Load Generator CPU to two Database CPU's was also maintained

to give reliable results. The Oracle's SGA (in-memory data cache) was configured to be 512 MB. The duration of the benchmark run was one hour. We restored the cold backup of database before each benchmark run. This provided consistent data across all runs.

Number of Users	Dell Average Number of Transactions Per Second	HP Average Number of Transactions Per Second	Dell CPU Utilization ( % )	HP CPU Utilization ( % )	Dell Memory Utilization ( % )	HP Memory Utilization ( % )
100	55.98	56.05	64.40	62.63	32.50	32.70
200	70.43	75.04	96.65	96.90	48.49	49.05
225	42.17	72.17	63.22	97.90	42.01	50.90
250	40.44	64.00	60.10	93.69	42.35	49.26
275	39.47	63.95	65.53	92.08	46.93	47.60
300	38.82	55.07	66.81	94.71	49.67	55.33
325	38.27	50.12	70.77	94.89	50.55	55.50
350	37.14	47.41	73.54	95.52	53.96	58.07
375	36.45	45.76	76.14	95.56	56.80	60.67
400	34.86	41.65	80.68	96.03	59.36	60.81
800	35.02	39.25	90.96	93.54	75.19	78.06

**Table 1: Measure of Transactions per Second (TPS) for an increasing number of users. The measurements were taken using 1 VM configured with 2 CPUs and 4GB of memory.**

In an examination of TPS under an increasing workload, we scaled the number of users from 100 to 800 for each platform while measuring TPS, CPU Utilization, and Memory Utilization. The runs were conducted for the duration of one hour and the results were averaged. These average results are given in Table 1 and plotted for a subset (Figure 1). As can be seen from Figure 1, TPS peaks at 200 users. To understand this further, we took a closer look at the Swingbench Benchmark, in particular, the way in which it handles failed transactions.

Number of Users	Dell Failure rate as Fraction of Total	HP Failure rate as Fraction of Total
100	0.00000	0.00000
200	0.00004	0.00000
225	0.00089	0.00008
250	0.00406	0.00012
275	0.01060	0.00007
300	0.02002	0.00096
325	0.03211	0.00550
350	0.04734	0.01394
375	0.06146	0.02461
400	0.08204	0.04589
800	0.20922	0.20836

**Table 2: Failed Transactions as a fraction of the whole. Note that at a workload of 800 users both systems yield a transaction failure rate of 20%.**

Swingbench considers a transaction as failed when a rollback happens in a database. The Total Number of Failed transactions is recorded in the xml file generated from the run. A deeper analysis of failed transactions focused on workloads between 100 and 400 is plotted in Figure 1. The peak TPS achieved for both machines at 200 users is about 70-75. At that point the number of failed transactions is at zero. Adding 25 more users results in a decrease of TPS for both machines but an increase of failed transactions for the Dell for Dell. The drop for the Dell Machine is substantially larger than the drop for the HP Machine. The explanation we offer for the difference is the way in which the Operating System on each platform handles CPU scheduling when utilization is high. Under 200 user

workloads, performance under both platforms has reached 97%, see Figure 2. Adding more users to the Dell Box results in utilizations dramatically adjusted downward. As a result, things are slowed down, and transactions start to fail. The HP box carries the CPU utilization at a high

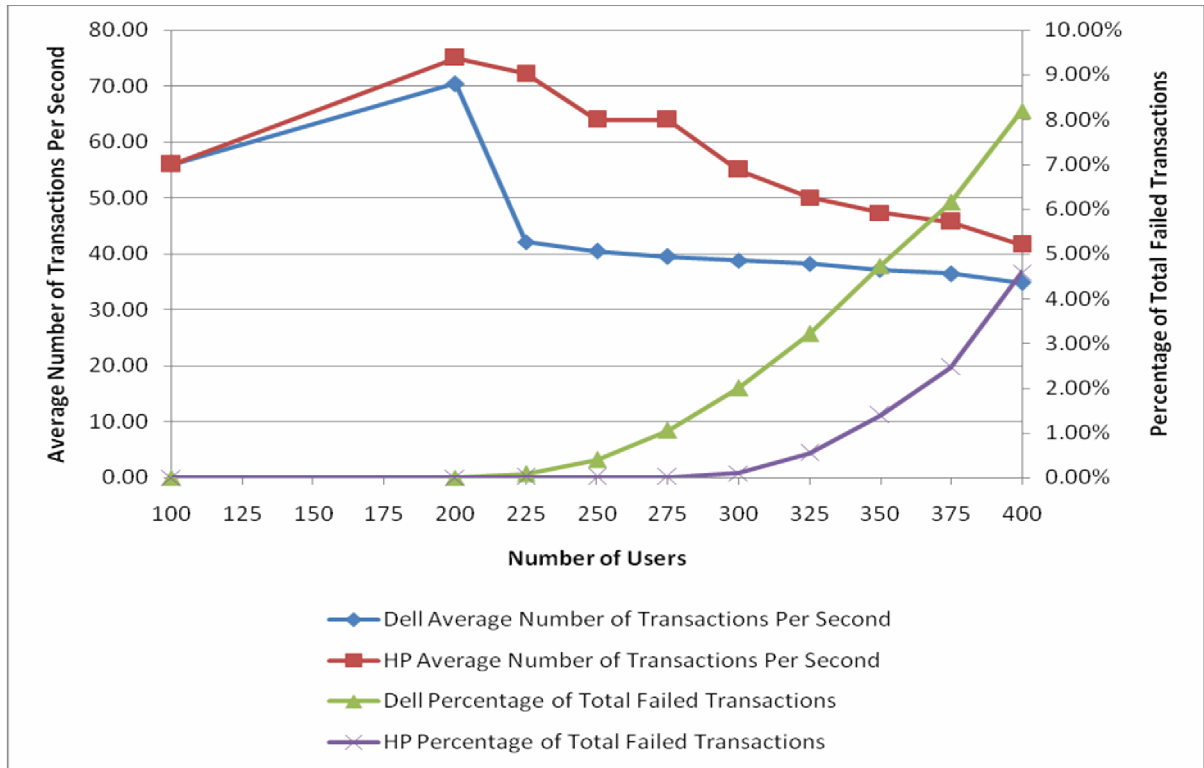
volume, so TPS drops more gradually and failed transactions don't set in until the number of concurrent user workload reaches 300.

As evident from Figure 2, we can see that CPU utilization is consistently high for HP machine. To evaluate the performance with increased CPU we doubled the number of CPUs from 2 to 4. The results are tabulated in Table 3. The speedup obtained for Dell machine is 1.49 while that of HP machine is 1.41.

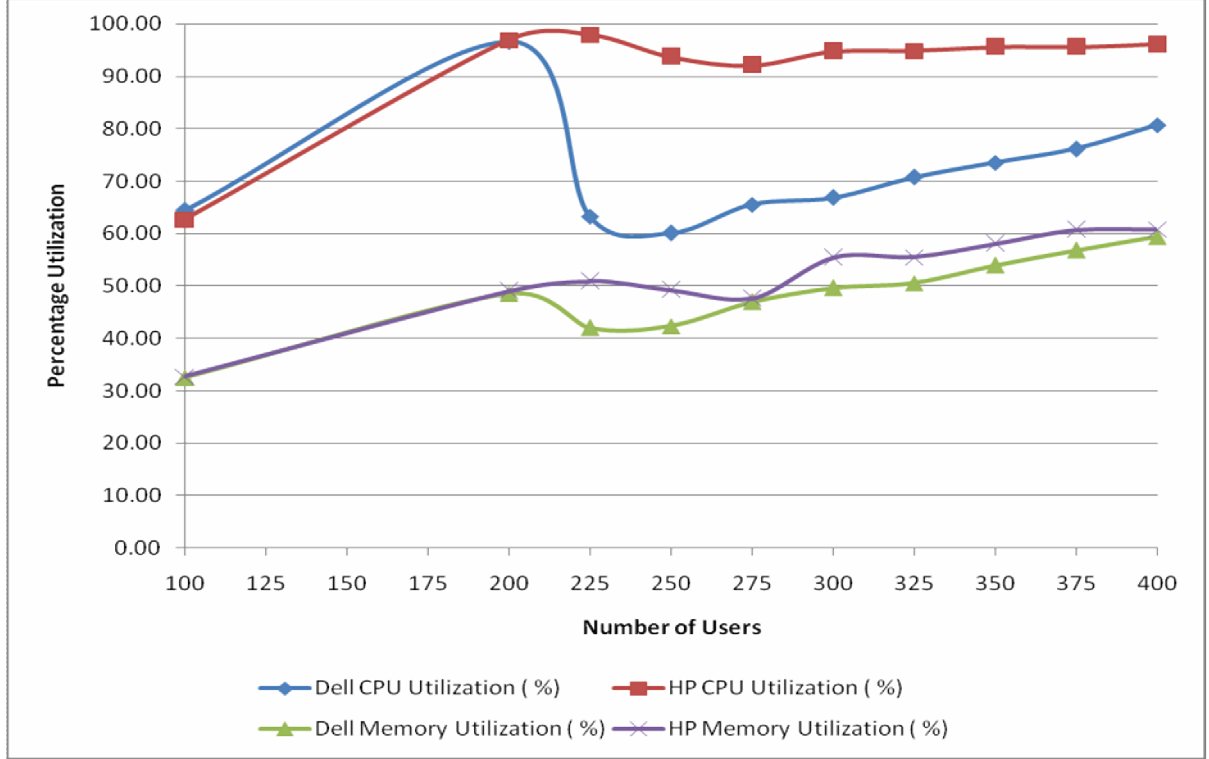
Number of CPUs	Dell Average Number of Transactions Per Second	HP Average Number of Transactions Per Second	Dell CPU Utilization (%)	HP CPU Utilization (%)	Dell Memory Utilization (%)	HP Memory Utilization (%)
2	70.43	75.04	96.65	96.90	48.49	49.05
4	104.95	105.59	86.96	87.07	51.61	52.08

**Table 3: Impact on Performance of adding CPUs to the virtual machine. The experiment is done with 1 VM having 4GB of memory, and holding the user workload at 200 users.**

Hence, given an infinitely powerful machine, the relationship between the number of users and TPS can be linear to an extent. However, Concurrency also plays a part in determining this relation. As we increase the user count the more concurrency will affect the TPS.



**Figure 1: Comparison of Transactions Per Second and Failed Transactions**



**Figure 2: Comparison of CPU and Memory Utilization**

## 4.2 JAVA SERVER

SPECjbb2005 is a software benchmark developed by Standard Performance Evaluation Corporate (SPEC) [4] for evaluating the performance of server side Java. The software industry uses a 3-tier architecture model where the user interface, application logic and data storage are conceptually separate and maintained as separate modules [15]. Particularly, in a Web server, the three tiers correspond to the front end web server, a middle level application and a backend database. SPECjbb2005 emulates this 3-tier system with an emphasis on the middle tier.

Client input (the first tier) is handled in SPECjbb2005 by means of random input selection. The middle tier or business logic is completely represented by SPECjbb2005. The third tier is represented by tables of objects, implemented by Java Collections, rather than a separate database. Using Java collections removes the variable of database performance from the performance picture. SPECjbb2005 is self contained and self driven (with respect to data). It is completely memory resident and supports workload scalability.

Benchmarks like RUBiS and VolanoMark have also been used to benchmark Java servers. While RUBiS is an auction site prototype [22] and is usually used to evaluate application server's performance and scalability, VolanoMark is a server-side java benchmarking tool [23] for assessing the performance and stability of any java virtual machine. We chose SPECjbb2005 in our study because it emulates a 3-tier system, which is the most common type of server-side Java application in the real world. In SPECjbb2005, a *warehouse* is a unit of stored data. It contains roughly 25MB of data stored in many objects in several Collections (HashMaps, TreeMap). An active user posting transaction requests within a warehouse is called a thread. There is a one-to-one mapping of threads to warehouses (that is, a thread will access only one warehouse) plus a few threads for SPECjbb2005 main and various JVM functions.

In our study, we configure the benchmark for one JVM instance having one warehouse and we extend up to eight warehouses. The benchmark increments by one warehouse after a set of transactions is completed. As more threads begin to post requests, we capture the throughput and the memory usage. We also measure the throughput while running 2 instances of JVM. In this case, the instances are run concurrently with the final measurement being the sum of the

individual instances. The JVM is assigned a score measured in terms of bops (business operations per second), obtained by averaging the total transaction rate in a SPECjbb2005 run from the expected peak number of warehouses, to twice the peak number of warehouses.

#### 4.2.1. TEST SETUP

We test the performance of the JVM server under different configurations, by altering the number of CPUs allocated and the Heap size of the JVM. The physical memory allocated to the VM is constantly maintained at 4 GB for all the tests. The setup details and metric calculation are as follows:

*Configuration 1: 2 CPUs, JVM Heap Size = 256 MB*

SPECjbb2005 needs a minimum of 256 MB of heap to support 8 warehouses. For the 1 JVM instance, metrics under 2, 3 and 4 warehouses is averaged to give the overall throughput (in bops). For the 2 JVM instance, metrics under 1 and 2 warehouses is averaged to give the overall throughput (in bops).

*Configuration 2: 2 CPUs, JVM Heap Size = 1640 MB*

As the real world use of Java server is memory intensive, we test the machines at the maximum allowable heap size of 1640 MB<sup>+</sup>. The metric calculation is the same as in Configuration 1.

*Configuration 3: 4 CPUs, JVM Heap Size = 256 MB*

In the above test setup, we observe that the JVM consumed almost 100% of the CPU and hence evaluate the machine when the CPU allocation is doubled to 4 CPUs. The metric calculation is based on the additional warehouses as the peak warehouse is now 4. For the 1 JVM instance, metrics under 4,5,6,7 and 8 warehouses is averaged to give the overall throughput (in bops). For the 2 JVM instance, metrics under 2,3,4 warehouses is averaged to give the overall throughput (in bops). From the results, it can be observed that the bops increase when more CPUs are allocated, confirming that the CPU has been a bottleneck in the previous cases.

*Configuration 4: 4 CPUs, JVM Heap Size = 1640 MB*

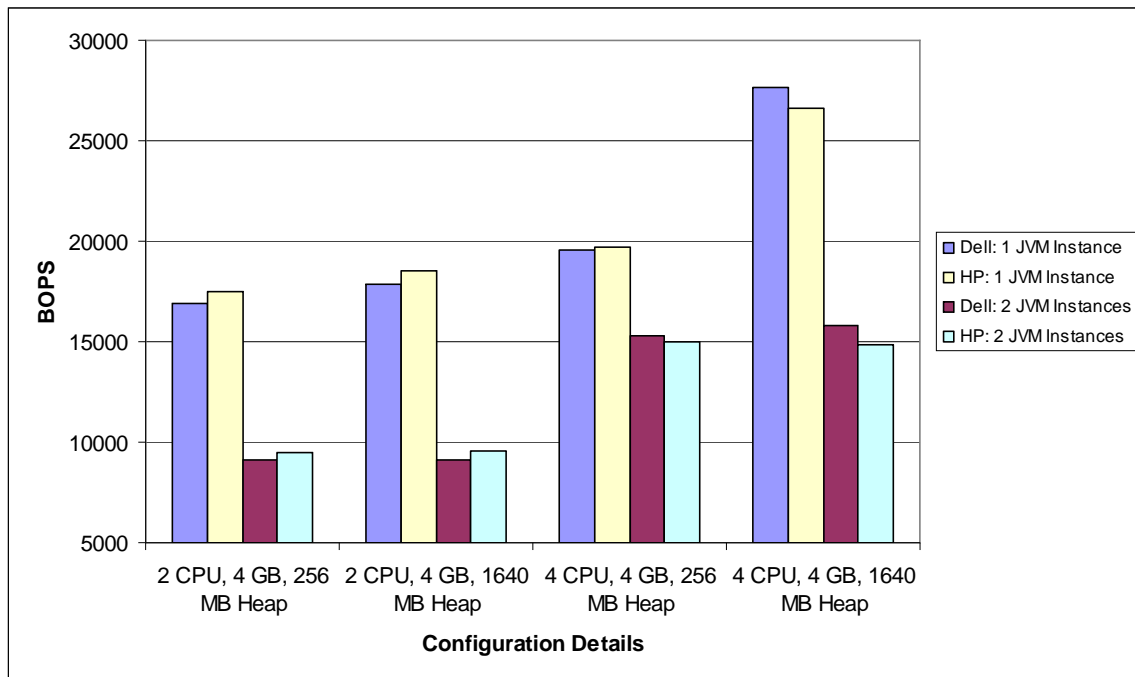
This configuration tests the effect on maximum CPUs with the maximum available Heap size. The metric calculation is the same as in Configuration 3.

From Figure 3, we can see that the HP machine outperforms the Dell machine for 2 CPUs. Despite similar usage of the CPU on either machine (as shown in CPU utilization graphs), HP is able to get a higher throughput. A higher heap size gives a better performance than the minimum heap size across all configurations. We can also observe that an increase in CPU has increased the bops. This confirms that the CPU is the bottleneck in the performance of the JVM server. However, the speedup obtained does not seem to have a linear relationship with the number of CPUs. This can be observed from Table 4.

Heap size	1 JVM Instance Speedup ratio		2 JVM Instances Speedup ratio	
	Dell	HP	Dell	HP
256 MB	1.153519	1.126729	1.672835	1.584989
1640 MB	1.543366	1.434984	1.738221	1.553407

**Table 4: Speedup ratio upon doubling the number of CPUs**

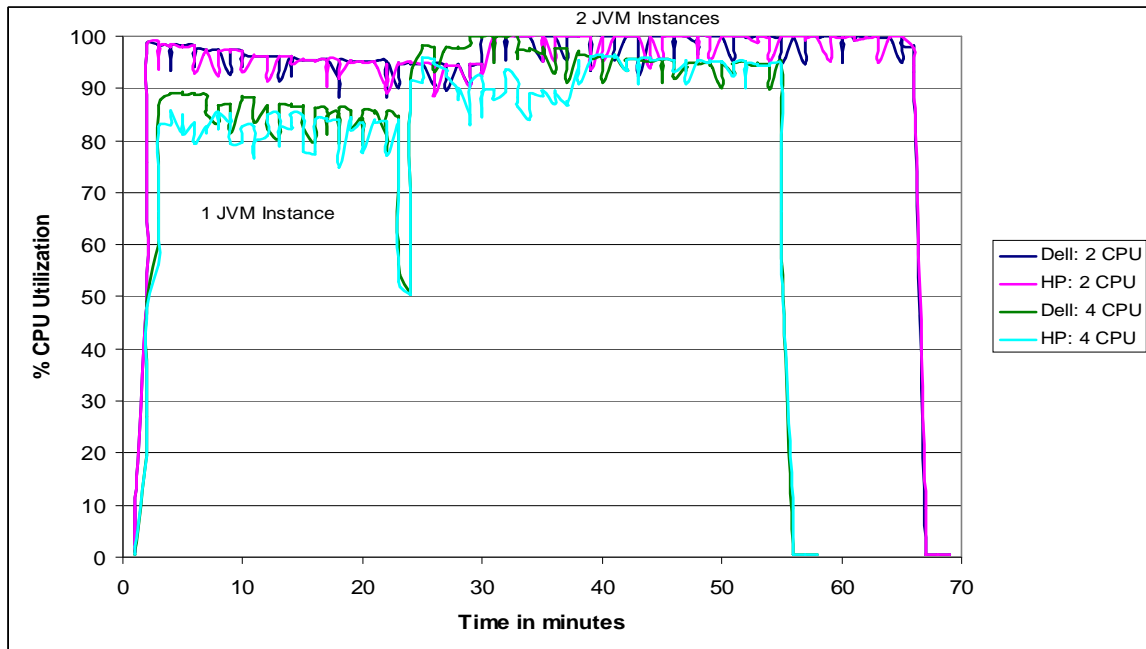
<sup>+</sup> The Microsoft Windows OS that has 4 GB of physical memory can provide applications with 3 GB flat virtual address space, and the kernel and executive use only 1 GB. The system DLLs map themselves to an address in the middle of the 3 GB address space available for the application. The JRE needs the object heap to be a continuous address range, so it can only use the range from the bottom of the address space to the middle of the address space. 1640 MB seems to be the maximum possible allocation without rebasing system DLLs.[19].



**Figure 3: Throughput measures on Dell and HP machines on various configurations**

The Dell machine shows a better speedup ratio and this could be the reason that it outperforms the HP machine when 4 CPUs are allocated.

From the CPU utilization graphs with 2 CPUs and 4 CPUs, it is evident that the CPU has been a bottleneck on the machine performance. The CPU utilization drops down considerably when 4 CPUs are allocated.



**Figure 4: CPU Utilization on Dell and HP machines (2 CPU vs. 4 CPU)**



From Figure 4, it can also be observed that the test iterations took a shorter duration when more CPUs were allocated, in spite of providing higher bops.

### 4.3 WEB SERVER

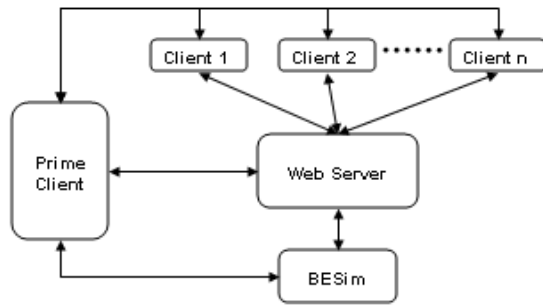
SPECweb2005 is a software benchmark developed by Standard Performance Evaluation Corporate (SPEC) [5] designed to measure a system capability as a web server. The SPECweb2005 measures the unencrypted and SSL encrypted request/response performance of the web server through the following three workloads.

- SPECweb2005 Banking - purely SSL encrypted workload.
- SPECweb2005 Ecommerce - partly SSL encrypted workload
- SPECweb2005 Support - purely unencrypted workload. Most of the requests and traffic in SPECweb2005 Support is HTTP downloading of files of various sizes.

All page requests of the three workloads are dynamic, enabling customization of the size of the page requests. In this benchmarking study, however, we have chosen to adhere to the default settings. The architecture of SPECweb2005 consists of the following components (as depicted in Figure 5)

- (1) Clients: The benchmark clients run a program that sends HTTP requests to the server and receive HTTP responses from the server.
- (2) Prime Client: The prime client initializes the clients and the back-end simulator. It also collects and stores the results of the benchmarking run.
- (3) Web Server: The web server is the system to which the clients send their HTTP requests.
- (4) Back end Simulator: The back-end simulator emulates a back-end database server from which the web server must communicate to get dynamic data.

Each workload in SPECweb2005 has a metric, which is the number of simultaneous user session that the web server can service while meeting the QOS requirements of the benchmark. The metric is expressed as the geometric mean of the ratio of each workload



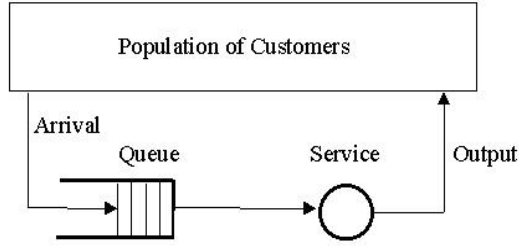
sub metric scores by their respective workload reference scores. For example, a system with a SPECweb2005 score of 100 is considered equivalent to the reference system, while one that has a score of 200 is considered to be twice as fast as the reference system. In our study, the prime client and the BESim server were collocated on the same machine and connected to the **Figure 5: SPECweb2005 architecture diagram** web server via a network connection.

We use 4 clients including the prime client that share 290 simultaneous sessions between them.

#### Calculating the number of simultaneous sessions\*

Each simultaneous session represents a customer who is repeatedly sending requests after a Think-time (TT) in between requests. We model this behavior using notions from the closed queuing model. Conceptually, the simplest queuing model is the single queuing system illustrated in Figure 6. The system models the flow of customers as they arrive, wait in the queue if the server is busy serving another customer, receive service, and eventually leave the system. With regard to SPECweb2005, the arrival rate is number of simultaneous sessions/TT (approximately). The request rate for each type of page is based on the pre-defined workload mix specified in the

\* With inputs from SPECweb2005 support team



**Figure 6: Single Queuing model.**

benchmark. The system models the flow of customers as they arrive, wait in the queue if the server is busy serving another customer, receive service, and eventually leave the system. With regard to SPECweb2005, the arrival rate is number of simultaneous sessions/TT (approximately). The request rate for each type of page is based on the pre-defined workload mix specified in the benchmark.

Typically, each user session would start with a single thread requesting a dynamically created file or page. Following the receipt of this file and the need to request multiple embedded files within the page, two threads corresponding to that user session actively make connections and request files on these connections.

The number of threads making requests (maximum request rate) on behalf of a given user session is limited to two, in order to comply with the HTTP 1.1 recommendations. The processing time here is the response time which is based on the page requested.

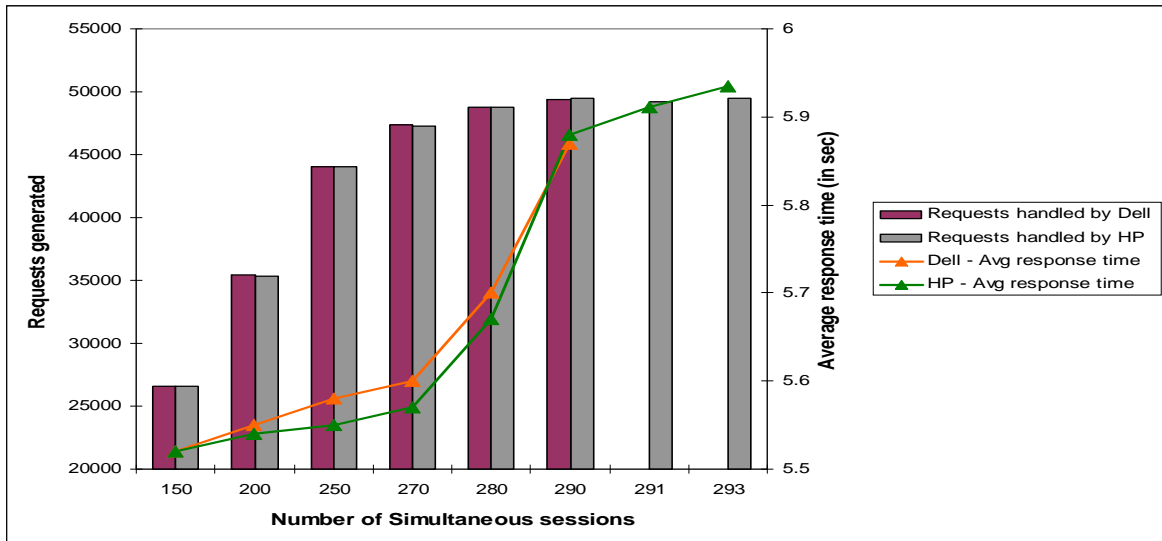
The benchmark defines QOS (Quality of Service) in terms of two parameters, Time\_Good and Time\_Tolerable. For each page, 95% of the page requests are expected to be returned within Time\_Good and 99% of the requests within Time\_Tolerable. There is another validation requirement that is defined as SWPD (sum of the per page Weighted Percentage Differences) checks. This ensures that server and clients maintain a minimum request rate.

While benchmarking the Dell machine, we encountered SWPD errors when the number of simultaneous sessions was 291. Upon calculation of the metrics<sup>+</sup>, it was found that one of the iterations did not meet the expected (minimum) request rate. In spite of all requests satisfying the Time\_Good parameter, the benchmark run was declared invalid. In the case of HP, the benchmark was successful for up to 293 sessions. When the number of sessions was 294, the benchmark failed as it could not satisfy the Time\_Good parameter (though adequate number of requests was made). To compare the 2 machines at the same level, we set the number of sessions to be 290 and captured the resource utilization behavior.

## Results

Configuration details: 2 CPU, 4 GB.

Workload type: Support workload



**Figure 7: Page requests and response time measures for Dell and HP machine.**

<sup>+</sup> Detailed calculations in Appendix section

From Figure 7 it can be observed that the Dell and HP machines processed a similar number of requests until the number of user sessions was 290. The response time was slightly lower on the HP machine compared to the Dell. Also, in both the machines, there is a steep increase in the average response time when the number of simultaneous sessions exceeds 270. The response time would scale to enable fetching more pages and support more requests. To ensure that the response time does not increase to an unacceptable value, the Time\_Good parameter constantly compares the actual response time to the expected response time and validates the run. When the numbers of sessions reached 291, the Dell machine is unable to generate enough requests and produces an invalid run thereafter. In the case of the HP machine, the threshold is reached at 294 sessions.

The difference in the machine performance at the threshold level and above the threshold is shown in the table(s) below.

Simultaneous user sessions	Test iteration	No. of requests	Avg. response time	Aggregate QOS Compliance			Errors
				Good	Tolerable	Fail	
290	1	49284	5.870	98.2%	99.9%	0.1%	None
	2	49310	5.866	98.8%	100.0%	0.0%	None
	3	49457	5.879	99.2%	100.0%	0.0%	None

Simultaneous user sessions	Test iteration	No. of requests	Avg. response time	Aggregate QOS Compliance			Errors
				Good	Tolerable	Fail	
291	1	49531	5.937	98.0%	100.0%	0.0%	None
	2	49413	5.828	97.7%	100.0%	0.0%	Too few requests
	3	49430	5.918	97.3%	100.0%	0.0%	None

**Table 4: Deeper examination of Dell machine at the point where the threshold is reached.**

Simultaneous user sessions	Test iteration	No. of requests	Avg. response (second)	Aggregate QOS Compliance			Errors
				Good	Tolerable	Fail	
293	1	49444	5.934	96.4%	99.9%	0.1%	None
	2	49460	5.936	96.1%	99.9%	0.1%	None
	3	49432	5.936	96.6%	99.9%	0.1%	None

Simultaneous user sessions	Test iteration	No. of requests	Avg. response (second)	Aggregate QOS Compliance			Errors
				Good	Tolerable	Fail	
294	1	49144	6.018	94.5%	100.0%	0.0%	Time_Good < 95%
	2	49158	6.046	94.2%	99.9%	0.1%	Time_Good < 95%
	3	49338	6.019	96.1%	100.0%	0.0%	None

**Table 5: Deeper examination of HP machine at the point where the threshold is reached.**

The HP machine clearly outperforms the Dell machine in both the aspects of supporting more simultaneous user sessions (and requests) and also having a lower average response time.

## 5.0 ROLE OF VM IN THE PERFORMANCE OF THE MACHINE

One of the significant features of a VM is that it supports partitioning of one machine into many virtual machines, each of which can function effectively. To check if high utilization of one VM affected the utilization on another, we ran the three benchmarks in parallel. We observe that the performance and resource utilization within a VM is not affected by the activity in other VMs. This is evident from the scores we obtained across each benchmark.

<b>Benchmark Details</b> Each VM has 2 CPU, 4 GB memory	<b>Individual run</b>		<b>Parallel run</b>	
	<b>Dell</b>	<b>HP</b>	<b>Dell</b>	<b>HP</b>
Swingbench (100 users, metric: TPS)	54.82	55.15	55.48	55.47
SPECjbb2005 ( Heap size: 256 MB, 1 JVM Instance, metric: bops)	16910	17498	16936	17494
SPECweb2005( 290 simultaneous sessions, metric: average response time)	5.873	5.879	5.871	5.88

**Table 6: Comparing throughput between individual and parallel benchmark runs.**

With the throughput scores varying by less than 0.2% between the individual and parallel run, we conclude that each VM is a separate entity (as it must be) and is not affected by the utilization of other VMs on the same physical machine. The CPU and memory utilization also do not vary much between the two cases.

## 5.1 DETAILS ON CPU AND MEMORY UTILIZATION

The workload consists of the overall set of processes being run by each benchmark, and is characterized by the amount of system resources used during this overall process. Each virtual machine was initially configured with two virtual CPUs, and 4GB of memory. While 3 VMs were actively used for running the benchmark, 1 VM on each machine was used as the client for the Web Server benchmark and one VM was left idle.

**DURATION** - The database benchmark can run for an infinite amount of time, measuring the transactions per minute and logging the data. The java server benchmark has two different workloads. One with a single JVM instance (runs for approximately 27-33 minutes) and another with 2 JVM instances in parallel (runs for approximately 35-40 minutes). SPECweb2005 uses 3 iterations of 40 minutes each to measure the performance of the web server. During the parallel run with all the benchmarks, we executed all the scripts for 140 minutes and captured the CPU and memory usage in that duration.

All graphs below are of the type XY (Scatter). A moving average line trend line has been drawn to connect the points and show the trend more clearly.

### DATABASE SERVER WITH 200 USERS:

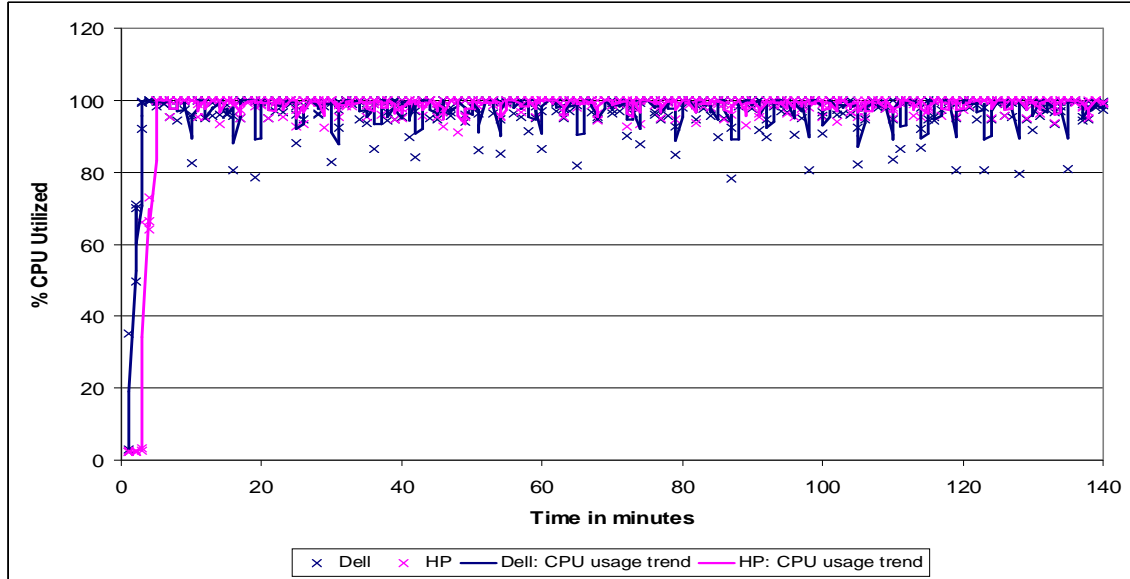


Figure8: CPU Usage on the database server

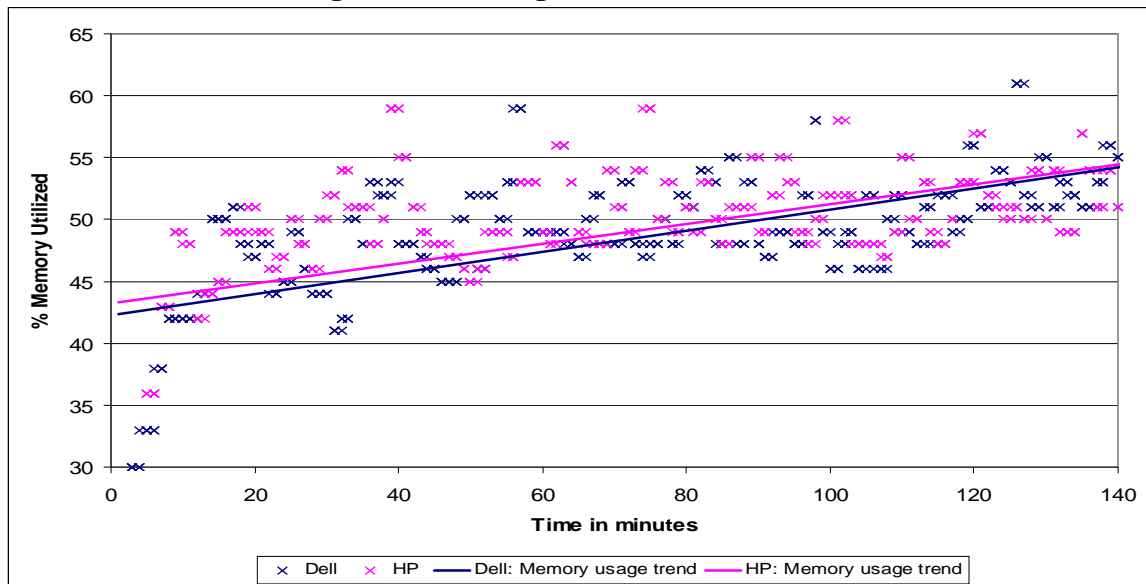
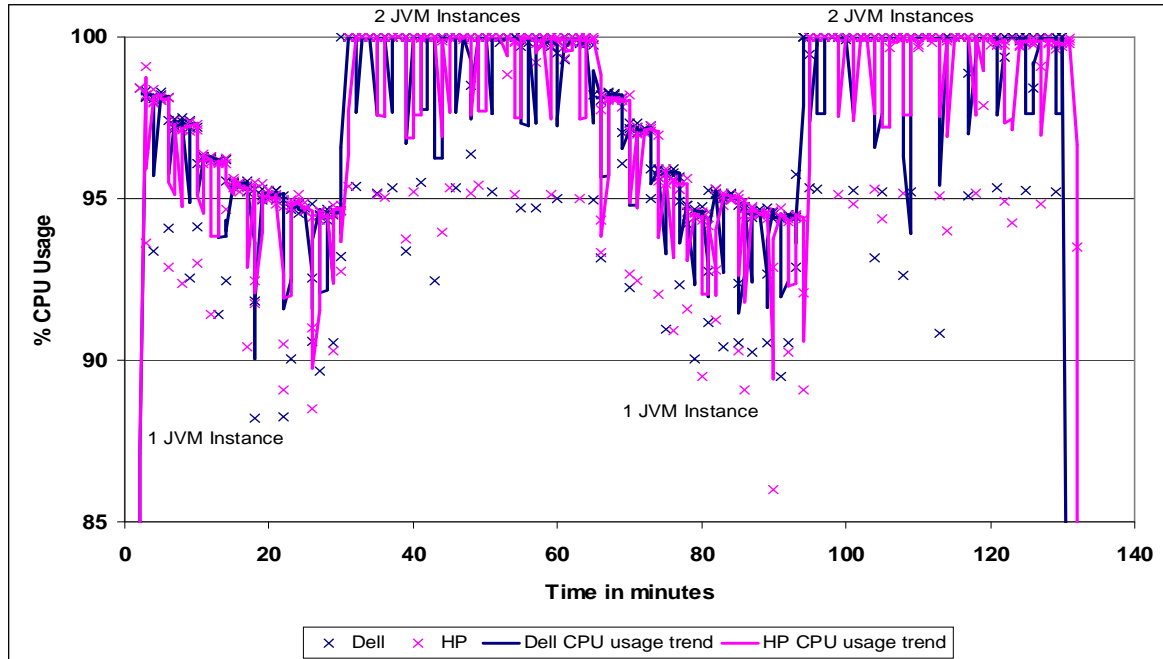


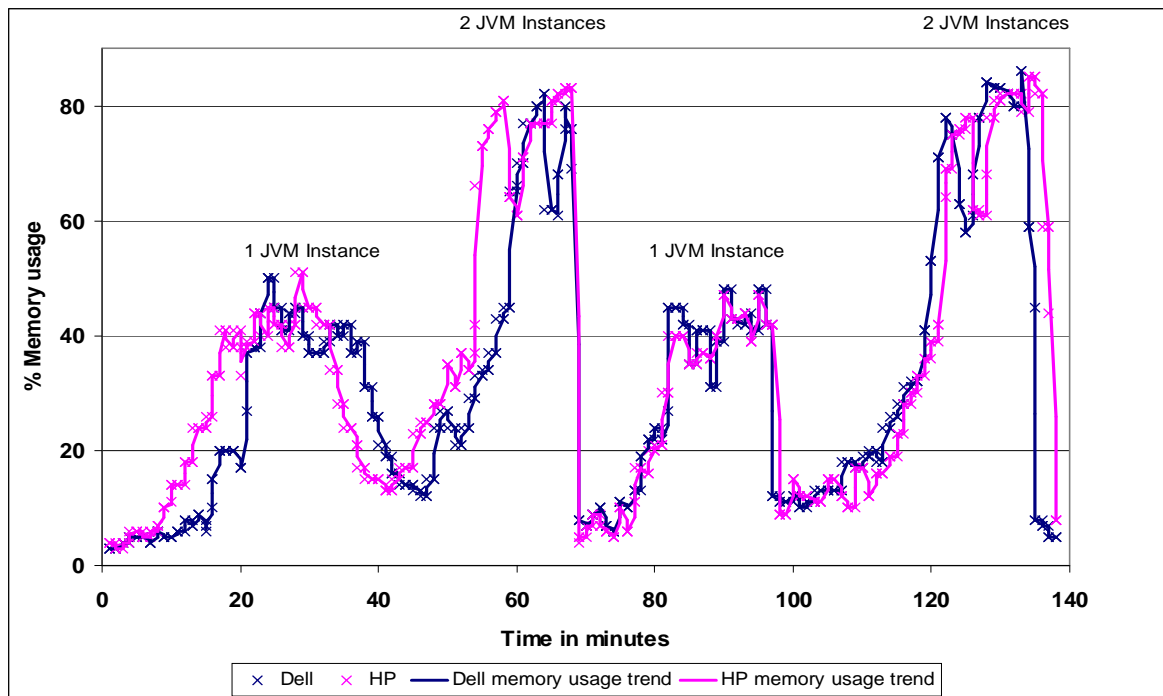
Figure 9: Memory Usage on the database server

### JVM SERVER:

Comparing the Dell and HP's CPU utilization, we can see that they had very similar utilization levels when they each had 2 CPUs(91.1% and 91.9%). But this similarity did not translate into the machine performance, measured in terms of business operations per second. HP outperformed Dell in this case.

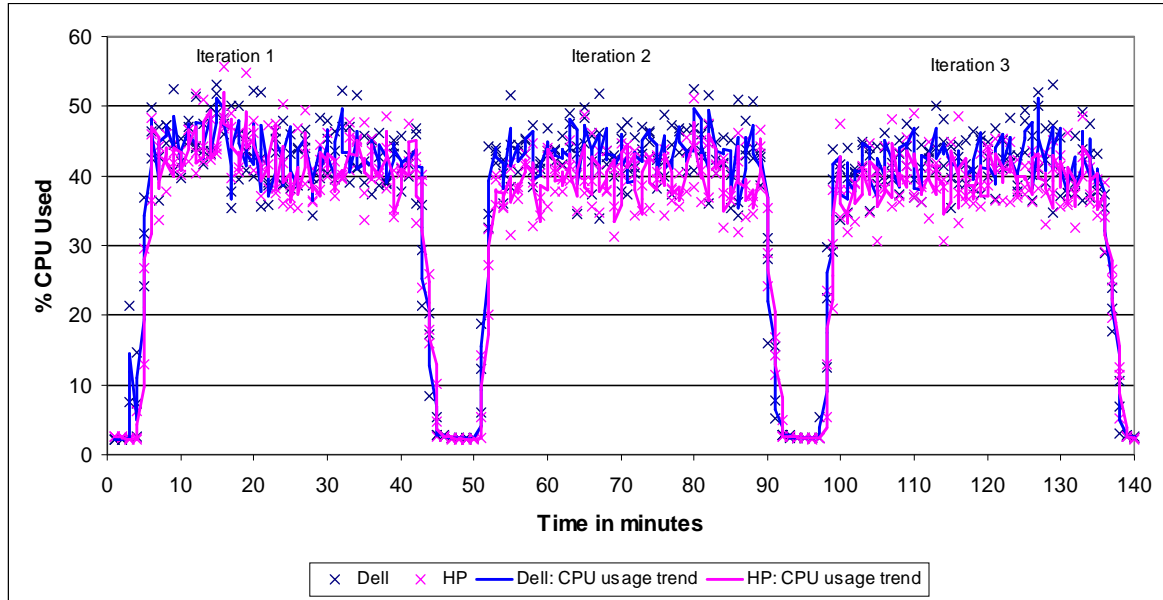


**Figure 10: CPU Usage on JVM server (2 CPUs, 4 GB Memory, 1640 MB Heap space)**

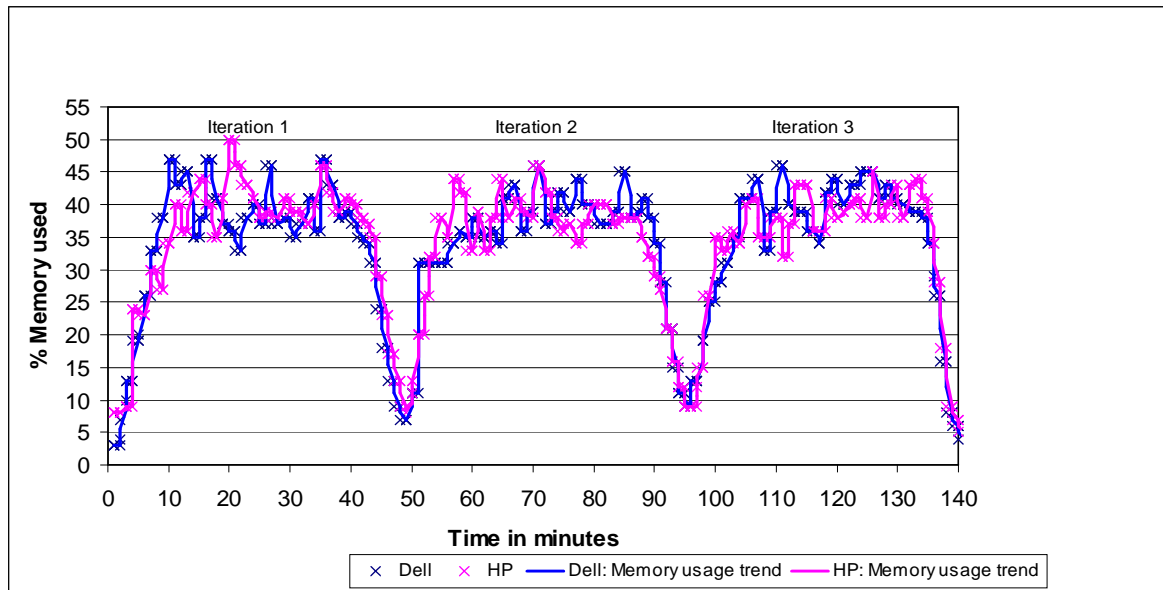


**Figure 11: Memory Usage on JVM server (2 CPUs, 4 GB Memory, 1640 MB Heap space)**

### WEB SERVER WITH 290 SESSIONS:



**Figure 12: CPU Usage on Web server**



**Figure 13: Memory Usage on Web server**

These statistics were captured while the benchmark measured the requests generated with 290 sessions on each machine. Even though the utilization trends are very similar, the HP machine processed slightly higher number of requests and yet had a comparable response time. HP's better performance could also be attributed to the fact that Dell machine was working on its threshold level while the HP was working on a level lower than its own threshold.

## 6.0 INTERESTING OBSERVATIONS

### LOAD SHARING BETWEEN THE CPUS:

Each VM has 2 CPUs and while the Database and the JVM show equal usage of the individual CPUs, the Web Server allocates 2/3 of the load to the first CPU and 1/3 of the load to the second CPU. We concluded that this was not a platform dependent issue but was dictated by the benchmark, as the Database server runs on the RHEL, and the JVM on a Windows platform and have equal load sharing between the two CPUs. We believe that the SPECjbb2005 emulates the master-slave system and hence utilizes the 2nd CPU in different manner. The fact that Dell and HP machines have the same behavior on load sharing reinforces our load sharing is determined by the benchmark.

### MEMORY ALLOCATION:

Each VM has a total of 4 GB of physical memory and the memory allocation trend after the machine boot up on the VM that runs the Windows OS is distinctly different from the VM that runs on the RHEL OS. To illustrate this, we left the VMs idle for about 30 minutes and then, rebooted it. The instant the VM was powered on, memory was granted. We observed that while the RHEL grants memory conservatively, as needed by the processes the Windows OS begins by granting the maximum amount of memory available. Also, once the memory was granted on the Windows machine, the metric was constant and never varied if a new process was started or stopped, while the RHEL allocation was directly related to the processes running at that instant.

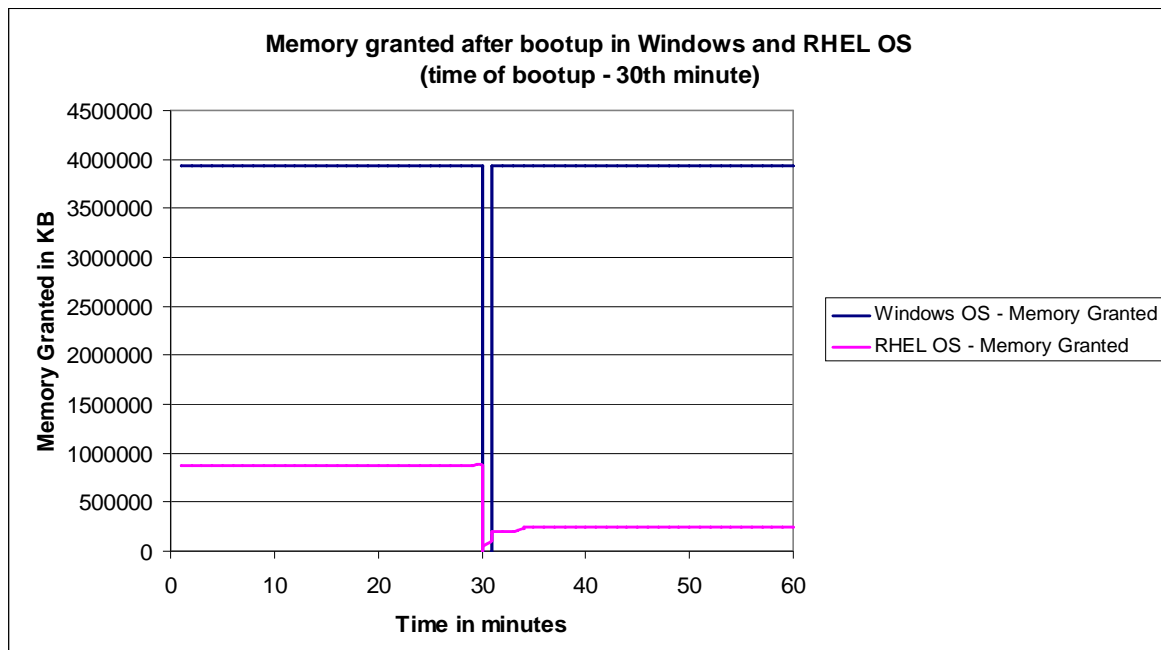


Figure 14: Memory Granted on Windows OS vs. RHEL OS

## 7.0 CONCLUSION AND FUTURE WORK

VMWare ESX is excellent at containing resource utilization and not allowing high levels of resource consumption in one VM affect the other VMs. However, the benchmarks analyzed the CPU and memory utilization and did not test the network traffic.

The two physical machines though configured with identical processors exhibit differences in performance. On the basic configuration with 2 CPUs and 4 GB memory per VM,



the HP machine in general outperforms the Dell machine on all the three benchmarks. The HP machine scales up efficiently to increased load and also delivers higher throughput in each setting. With regard to the memory and CPU utilization, the average CPU utilization on the HP machine is slightly higher than the Dell machine and the pattern is reversed in the case of memory utilization. Though, the differences in utilization are quite small (sometimes less than 5%), we believe that this could be an important consideration when choosing a machine for a specific workload.

With respect to the database benchmark and the web server benchmark, we can conclude that Dell lacks efficiency in utilizing the resources that was provided while HP was able to fully utilize the provided resources. This led to more number of transactions/requests being executed by the HP Virtual Machine.

With respect to the JVM benchmark, the results vary based on the number of CPU allocated. In the original setup, HP clearly outperformed the Dell machine. However, the reconfigured setup with 4 CPUs gives a slightly different performance. The Dell JVM gives slightly better performance than the HP JVM. The Dell machine seems to be constrained when it is allocated 2 CPUs. When the allocation increases, the performance gets better. Further analysis on these lines could be part of future work.

When the VM was set up initially, we failed to adhere to the recommendations for aligning the VMFS partitions. Upon reading the importance of this from one of VMWare's reports, we feel that the machine performance may improve if the partitions are aligned to reduce latency. It would be interesting to observe how the reduction in latency corresponds to the machine performance.

In regards to the database study, it would be interesting to see what the database on Dell Virtual Machine is waiting on. One can make use of Automatic Database Diagnostic Monitor (ADDM) technology provided in Oracle 10g to take a database snapshot and analyze the waiting events. This would help us better understand the underlying issue. In regards to the Java – JVM study, it would be interesting to study the effect of increase in CPUs on the performance of the machine. One could start with a machine of higher configuration and study the performance of the Java server in different settings and draw out a relationship between the number of CPUs, level of utilization and bops obtained. It would also be of interest to determine the reason for the difference in observed speedup between the two machines while the resource allocation is doubled.

## ACKNOWLEDGEMENTS

We sincerely thank Rob Henderson, Systems Staff of Department of Computer Science, Indiana University and Dominic Giles, author of SwingBench, for their valuable assistance and guidance throughout the study. We also thank Samantha Foley, Minh Tang and Felix Terkhorn for being part of the team during the class project that got the benchmark effort underway.

## REFERENCES

- [1] Answers.com. Computer Desktop Encyclopedia. Computer Language Company Inc., 2007. Answers.com, April 2007. <http://www.answers.com/topic/pc-magazine-benchmarks>
- [2] P. Chen and D. Patterson. A new approach to I/O performance evaluation. *ACM Transactions on Computer Systems*, 12:308–339, 1994.
- [3] Hewlett-Packard Development Company. Smartpeak WLM with an HP BladeSystem and VMware ESX environment, June 2006.
- [4] Standard Performance Evaluation Corporation. SPECjbb2005, <http://www.spec.org/jbb2005/>
- [5] Standard Performance Evaluation Corporation. SPECweb2005, <http://www.spec.org/web2005/>
- [6] B. Clark et al. Xen and the art of repeated research. *USENIX Annual Technical Conference*, pages 135–144, 2004.
- [7] K. Govil et al. Cellular disco: resource management using virtual clusters on shared-memory multiprocessors. *ACM Transactions on Computer Systems*, 18:229–262, 2000.

- [8] P. Barham et al. Xen and the art of virtualization. *In Proceedings of the nineteenth ACM symposium on Operating System principles*, pages 164–177, 2003.
- [9] VMware Inc. VMware. <http://www.vmware.com>.
- [10] S. King, G. Dunlap, and P. Chen. Operating system support for virtual machines. In 2003 *USENIX Annual Technical Conference*, pages 71–84, 2003.
- [11] D. Morse. Improved virtualization performance with 9th generation servers. Technical report, Dell Enterprise Systems, August 2006.
- [12] T. Muirhead and D. Jaffe. Advantages of Dell PowerEdge 2950 two socket servers over Hewlett-Packard Proliant DL 585 G2 four socket servers for virtualization. Technical report, Dell Enterprise Systems, December 2006.
- [13] R. Rose. Survey of system virtualization techniques. 2004.
- [14] Singh, A., An Introduction to virtualization. <http://www.kernelthread.com/publications/Virtualization/>
- [15] Three tier architecture [http://en.wikipedia.org/wiki/Multitier\\_architecture](http://en.wikipedia.org/wiki/Multitier_architecture)
- [16] VMware. A performance comparison of hypervisors, 2006.
- [17] A. Whitaker, M. Shaw, and S. Gribble. Denali: Lightweight virtual machines for distributed and networked applications. Technical report, University of Washington, 2001.
- [18] Dell System hardware details, <http://support.dell.com/support/edocs/systems/pe6950/en/hom/html/>
- [19] Maximum heap size, <http://forum.java.sun.com/thread.jspa?threadID=201718&forumID=37>
- [20] AMD HyperTransport Technology, [http://www.amd.com/us-en/Processors/DevelopWithAMD/0,,30\\_2252\\_2353,00.html](http://www.amd.com/us-en/Processors/DevelopWithAMD/0,,30_2252_2353,00.html)
- [21] VMware Technical report: VMmark: A Scalable Benchmark for Virtualized Systems <http://www.vmware.com/vmtn/resources/573>
- [22] Benchmarking tool: Rubis, <http://rubis.objectweb.org/>
- [23] Benchmarking tool: Volanomark, <http://www.volano.com/benchmarks.html>

## APPENDIX

### Setting the threshold on the number of sessions on the Dell machine:

QOS Criteria to be met: The sum of the per page Weighted Percentage Differences (SWPD) must be within +/- 1.5%. (i.e. the actual number of requests generated must be greater than or equal to the minimum number of requests)

Calculation:

Expected total # of requests (ETR) = (#Sessions \* RunTime) / (ThinkTime \* %RwTT + AvgRspTime)

RunTime, Thinktime and % RwTT are always constant for a specific workload.

RunTime = 1800 seconds; ThinkTime = 5 seconds; %RwTT = 92.08 %

SWPD = PageMix% of each page \* ETR

where PageMix% is the percentage requests for the given workload page.

For the Support workload,

SWPD = (PageMix% catalog + PageMix% download + PageMix% file + PageMix% file catalog + PageMix% product + PageMix% search ) \* ETR

SWPD = 1 \* ETR

SWPD = ETR

If the metric were to be satisfied, then  $(ETR - ANR)/ETR < 1.5$

$ETR = (291 * 1800) / (5 * 0.9208 + 5.828) = 50210.889$

ANR = 49413

$$(ETR - ANR)/ETR = (50210.889 - 49413)/50210.889 = 0.1589 = 1.589\% (> 1.5\%)$$

Hence the run is invalid for 291 sessions and the maximum number of sessions supported on the Dell machine is 290.