Characterization & Analysis of a Server Consolidation Benchmark

Padma Apparao, Ravi Iyer, Xiaomin Zhang, Don Newell, Tom Adelmeyer

Intel Corporation

Contact: padmashree.k.apparao@intel.com

Abstract

Virtualization is already becoming ubiquitous in data centers for the consolidation of multiple workloads on a single platform. However, there are very few performance studies of server consolidation workloads in the literature. In this paper, our goal is to analyze the performance characteristics of a representative server consolidation workload. To address this goal, we have carried out extensive measurement and profiling experiments of a newly proposed consolidation workload (vConsolidate). vConsolidate consists of a compute intensive workload, a web server, a mail server and a database application running simultaneously on a single platform. We start by studying the performance slowdown of each workload due to consolidation on a contemporary multi-core dual-processor Intel platform. We then look at architectural characteristics such as CPI (cycles per instruction) and L2 MP (L2 misses per instruction) I, and analyze the benefits of larger caches for such a consolidated workload. We estimate the virtualization overheads for events such as context switches, interrupts and page faults and show how these impact the performance of the workload in consolidation. Finally, we also present the execution profile of the server consolidation workload and illustrate the life of each VM in the consolidated environment. We conclude by presenting an approach to developing a preliminary performance model based on the performance characteristics collected for the consolidation workload.

Categories and Subject Descriptors C.4 Performance of Systems, Measurement techniques; Performance attributes

General Terms Measurement, Performance, Experimentation.

Keywords Virtualization; Xen, Data center consolidation; vCon; Cache Scaling; virtualization overheads

1. Introduction and Motivation

The use of virtualization for consolidation of multiple workloads on to a single platform is growing rapidly in datacenter environments. Virtualization offers the opportunity for better manageability, provisioning and cost. Virtual machine monitors or hypervisor from VMware [19], XenSource [21], Microsoft[13], and others manage the virtual machines running on a single platform and ensure that they are functionally isolated from one another. However, from a performance standpoint, it is expected that the performance of each of the virtual machines can be significantly affected by the other virtual machines running on the same platform. Since each of the virtual machines can be running entirely different operating systems and workloads, the overall performance behavior of consolidated scenarios will be significantly different from traditional commercial server workloads that run in dedicated mode on a platform. As virtualization becomes ubiquitous, it is imperative that future platforms are architected with characteristics of consolidated workloads in mind so that they offer maximum performance.

In this paper, our goal is to characterize and analyze the performance of a representative server consolidation workload in order to guide future platform architectures. To our knowledge, a detailed characterization and analysis of a server consolidation workload has not been accomplished in the past. Existing studies [2][11][14][18] of the performance impact of virtualization have been focused on comparing multiple virtual machines for a particular workload or studying specific aspects of virtualization such as I/O or memory virtualization. One of the key challenges for studying a server consolidation scenario is the lack of industry-standard benchmarks on virtualization and server consolidation. To represent a server consolidation scenario, we chose the vConsolidate benchmark [16] that consists of a web server VM, a database server VM, a java server VM and a mail server VM running simultaneously on a server platform. Based on vConsolidate, we perform detailed performance analysis on Intel's latest Core 2 Duo based dual-processor server platform running Xen [21]. We start by examining the performance slowdown that each virtual machine experiences due to consolidation and then delve deeper into the architecture characteristics of the workload (cycles per instruction, misses per instruction, virtualization event counts, etc) based on hardware performance counters. We then study the execution profile of the consolidated server workload and highlight the life of each virtual machine. This allows us to determine how virtual machines are often paired with one another during the execution and how they interfere with each other in terms of shared resource consumption (such as cache and memory). Last, but not least, the purpose of the characterization and analysis is to build a virtualized workload modeling framework that can be used to predict the performance of future platforms and configurations. In this paper, we show how the data collected so far can help towards developing such a consolidated server modeling framework and what the next steps are to achieve this.

VEE'08 March 5–7, 2008, Seattle, Washington, USA.

The rest of this paper is organized as follows. Section 2 provides some more background and motivation on consolidated servers and describes related work on virtualization performance analysis. Section 3 describes the vConsolidate workload, the system configurations and the measurement and analysis methodology used in this paper. Section 4 describes the results and analyzes the implications. Section 5 shows a path to developing a consolidated server modeling framework based on the data collected and the next steps required. Section 6 concludes the paper with a summary of contributions and future work.

2. Analyzing Consolidation Performance

Due to the complexity of the problem and the lack of well-defined industry-standard benchmarks in this space, analysis of consolidated server environments is still in its infancy. However, the need for consolidated server analysis continues to grow in the following different vectors:

(a) Performance Feedback for IT Administrators: Virtualization is seen as a means to meet the growing demands for IT by providing flexibility in provisioning and transitioning to new more powerful systems. These two form the basis of consolidation. In typical data centers, we have a consolidation of various applications running together. When a new IT customer demands resources to run their application, virtualization addresses the problem by providing a VM with virtual resources that runs on a large mutli-processor platform. The IT customer is however unaware of any virtualization layer. As more and more demands are made the IT administrator flexibly deploys these applications in isolated VMs that run on this large powerful platform.

The job of IT is not done just by deploying the customer applications or consolidating the legacy application on the faster and better systems. Performance is a very important criterion that has to be evaluated. Earlier without the consolidation the applications exhibited a certain performance. The customers expect much better performance on the newer systems without any concern for the fact that the applications are running in a virtualized environment. Hence the task of the IT managers is not done. They have to ensure that all applications running in the data center are performant and get their “fair” share of the resources.

(b) Performance Feedback for Platform Architects: Characterization and analysis of non-virtualized commercial server bench-marks has been an important domain of research for architecture evaluation. As virtualization grows rapidly and becomes ubiquitous, it is important that the consolidated server usage model is taken into account when defining future platform architectures. Evaluation of consolidated commercial server workloads is all the more challenging since it not only requires the understanding of each individual server application, but also the behavior of how these applications are affected by virtualization overheads, how they are affected when they run in an interleaved fashion with other workloads, how they share resources within the platform and finally how they scale in terms of future platform requirements.

As an example, let us consider the current multi-core architectures with several cores sharing a last level cache on the die. While sharing the last level cache makes most efficient use of the cache space, it will also allow virtual machines to interfere with another depending on their memory access pattern and resource consumption behavior. A deeper understanding of consolidated server environments is important to appropriately estimate the extent of this problem and determine appropriate solutions to addressing it.

(c) Performance Feedback for VMM/hypervisor developers: Today, the virtual machine monitor effectively manages the functional isolation of virtual machines and arbitrates their consumption of visible resources (such as processor cores and I/O devices). Without a detailed analysis of many virtual machine environments, we expect that the scheduling heuristics currently used in virtual machine monitors are likely to be very sub-optimal. To provide appropriate feedback to improve VM schedulers and help manage the resources in a consolidated environment efficiently, it is important to characterize and analyze the execution profiles of server consolidation scenarios.

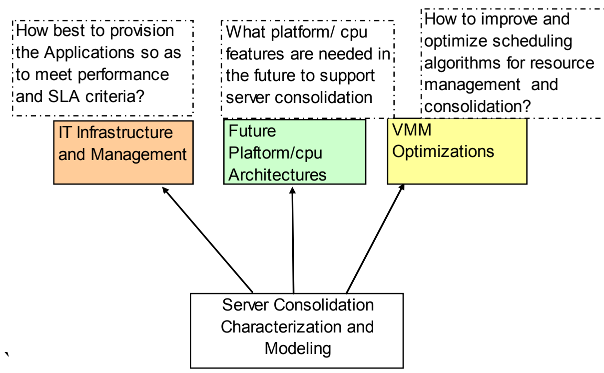

Figure 1: Need for Server Consolidation Analysis

Our work stems from the above need for better overall modeling of server consolidation scenarios. The final goal is to build a server consolidation model based on detailed characterization and analysis of a variety of server consolidation scenarios. This paper is a first step towards that goal. In this paper, we picked on server consolidation scenario, as specified in the vConsolidate benchmark [[16]], and perform a detailed characterization of this server consolidation scenario along the following vectors:

(a) What is the impact of consolidation on individual virtual machine performance? What are the key contributors to performance slowdown?

(b) What are the architectural characteristics of a consolidated workload? How does the workload scale in terms of architectural resources such as cache?

(c) What is the execution profile of consolidated workloads and what is the life of a virtual machine? How sensitive is the performance to scheduling changes (such as affinitization of a virtual machine)?

(d) How do we go about collecting characterization data to build a server consolidation modeling framework for future architecture evaluation?

Before we delve into answering these questions, we will next go over the related work in this domain on virtualization analysis.

2.1 Related work

With virtualization rapidly regaining relevance in the last few years, there have been a number of recent papers [2][5] on the performance overheads of virtualization and the analysis of specific aspects of virtualization. Monitoring tools for virtualized environments have been investigated by [1] [5]. Both Xenoprof and Xenmon provide important instrumentation capability that allows us to overcome hurdles in performance analysis of consolidation environments. I/O virtualization analysis has been presented in several papers including [14][18]. These papers describe the overheads of networking in Xen and the architectural effects of cache and TLB performance on virtualized network I/O performance. Memory virtualization has been investigated by Waldsperger in [20]. Waldsperger describes several innovative techniques for VMware ESX server memory management including ballooning, idle memory tax, content-based page sharing and hot I/O page remapping.

Some recent architectural studies [4] have also started evaluating the effects of consolidation on architectural alternatives for caches and directories in CMP servers. While these are a great start, the analysis framework used in these papers actually do not model virtualization directly, but instead just runs multiple workloads simultaneously to take the consolidation effects into account. To our knowledge, there are no published studies that describe a detailed characterization and analysis in the context of a server consolidation benchmark. As a result, we expect that insights presented in this paper are useful and will be valuable to IT administrators, architects and VMM developers.

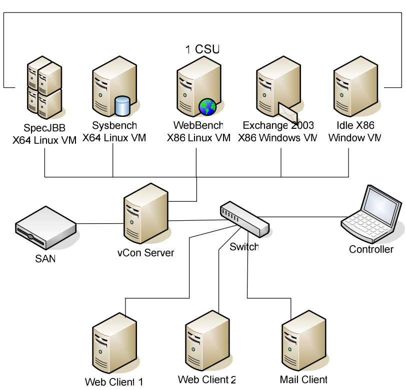

Figure 2: vConsolidate 1 CSU configuration

3. vConsolidate: Server Consolidation Workload

The vConsolidate (vCon) benchmark consists of a compute intensive workload/application, a database workload, a Web server workload, and a mail server workload. Each of these workloads runs in its own VM. To emulate a real world environment an idle VM is added to the mix since datacenters are not fully utilized all the time. The compute intensive VM runs SPECjbb. Typically SPECjbb is a CPU intensive workload that consumes as much CPU as it possibly can. However, in this environment, namely vCon, SPECjbb has been modified to consume roughly 75% of the CPU or so, by inserting in random sleeps every few millisecs. This is to represent workloads that are more realistic. The database VM runs Sysbench; an OLTP workload running transactions against a mysql database. The Webserver VM runs Webbench which uses Apache Webserver. The Mail VM is a Microsoft Exchange workload that runs transactions on Outlook with 500 users logged in simultaneously. A configuration as described above with 5VMs running the different workloads comprises a Consolidated Stack Unit or known as CSU. The diagram in Figure 1 represents a 1CSU configuration.

3.1 vConsolidate Profiles

vConsolidate has predefined VM configurations that are termed as profiles. These profiles describe the configuration of each VM in terms of virtual CPUs, memory, the OS and the benchmark to run within. The profiles one can use depend on the Server under Test and the underlying platform (Linux/Windows), the VMM capability such as (a) Can the VMM support 64 bit guests (b) Can the VMM support Windows guests? (c) Can the VMM support PAE etc. The profiles defined by vConsolidate are shown in Table 1.

Workload	Profile # 1				Profile # 2			
	vCPUs	vMemory	OS	App	vCPUs	vMemory	OS	App
Web Webbench	1	1.0 GB	Windows 32-bit	IIS	2	1.5 GB	Windows 32-bit	IIS
Mail Loadsim	1	1.0 GB	Windows 32-bit	Exchange	1	1.5 GB	Windows 32-bit	Exchange
Database Sysbench	1	1.0 GB	Windows 32-bit	MS SQL	2	1.5 GB	Windows 64-bit	MS SQL
Java SPECjbb	1	1.7 GB	Windows 32-bit	BEA JVM	2	2.0 GB	Windows 64-bit	BEA JVM
Idle	1	0.4 GB	Windows 32-bit		1	0.4 GB	Windows 32-bit	

Workload	Profile # 3				Profile # 4			
	vCPUs	vMemory	OS	App	vCPUs	vMemory	OS	App
Web Webbench	2	1.5 GB	Linux 32-bit	Apache	2	2.0 GB	Windows 32-bit	IIS
Mail Loadsim	1	1.5 GB	Windows 32-bit	Exchange	2	2.0 GB	Windows 32-bit	Exchange
Database Sysbench	2	1.5 GB	Linux 64-bit	MySQL	4	2.0 GB	Windows 64-bit	MS SQL
Java SPECjbb	2	2.0 GB	Linux 64-bit	BEA JVM	4	2.0 GB	Windows 64-bit	BEA JVM
Idle	1	0.4 GB	Windows 32-bit		1	0.4 GB	Windows 32-bit	

Table 1: vConsolidate Profiles

3.2 Our Experimental Configuration

Our experiments have focused on a single CSU configuration with all the 5 VMs running on the Xen hypervisor. We have used profile 3 from above for our experimentation where the SPECjbb VM has 2 vCPUs and 2GB memory, Sysbench VM has 2 vCPUs and 1.5GB memory, Webbench is assigned 2 vCPUs and 1.5GB memory and Mail VM has 1vCPU and 1GB memory. The idle VM is given 1 vCPU and 0.4GB memory. Some of the Vms are 64 bit and others are 32 bit and as observed there is a mix of Linux and Windows guest VMs. The entire configuration is on a private switched subnet so that the corporate traffic does not affect the benchmark. Two clients generate traffic for the Webserver workload and one client for the Exchange/Mail workload. All VMs run on platforms with Intel Virtualization technology enabled [17].

3.3 System Configuration and Tools

The server under test is an Intel Core2-Duo machine with two sockets each with Intel Core-2 Duo™ [9] processors running at 3 GHz. with a 4MB second level cache. The system has 16GB of physical memory that is distributed between the VMs. The

Hypervisor in our experiment is the 64 bit Xen 3.1 (cs 13100) version This version of Xen supports Intel VT technology and can support both windows and Linux guests.

The tools used mostly were generic tools such as xentop, sar, iostat, xentrace and xenperf available in Xen. Xentop gives the CPU utilization per domain (VM or virtual machine) and other IO characteristics. We typically run xentop over a period of time during the benchmark run and take an average of the data we see. Care is taken to make sure that each of the VMs has reached its steady state before any data is gathered.

For using xentrace and xenperf we have heavily instrumented the Xen hypervisor code to give us the statistics we need. For example in the scheduling study discussed later, we have instrumented the Xen scheduler to dump out information about which VM and which vCPU is running a physical core whenever a scheduling event occurs. Each processor on the system has its own buffer in which that data is written. Periodically the Xen hypervisor gathers the data from these buffers and combines them into a single buffer. We use offline tools to post process the data and recreate a chronological sequence of events. Similarly we have used xentrace to gather virtualization events such as context switches, page faults, interrupts etc. We have also used Xentrace to give us the cost of these events. However, as these costs are proprietary to Intel Virtualization technology, they are not discussed in the paper.

Finally, we have also used a tool developed by Intel that reads processor counters such as cycles, instructions retired etc. This tool along with tools like Xenoprof [1] has been used for the architectural data.

4. Consolidated Server: Results and Analysis

We have done extensive measurements to characterize the consolidated benchmark under virtualization. Starting from looking at the raw performance, we have performed experiments to understand the degradation of performance going from a dedicated to a consolidated environment. Insights into the architectural characteristics have shown that cache interference is primarily the cause for the performance loss. In later sections we quantify these in-sights and also show how the interference changes with cache size. We have also performed other experiments to understand how scheduling the individual workloads and the effect of this on the cache characteristics. Scheduling is considered from two different perspectives; first looking at how the VMs get scheduled across different processors and how much time each VM runs with any other VM during a full consolidated run. The second aspect of scheduling is what effect a VM has on any other VM and this is done by looking at the individual VMs pair wise and performing affinitization of the VMs and their vCPUs. In the following subsections we will look at all the different experiments and their results. All these results will be used as input to a performance model that will be outlined in the last section.

4.1 Performance Impact of Consolidation

First off we have looked at the raw performance of the consolidated workload. The consolidated performance is a measure of how the individual workloads behave under consolidation. The benchmark does report a consolidated score, but for the purposes of our work, we are interested more in the performance of the individual workloads and how that compares to the performance of the workload when running alone. To measure the dedicated performance we run the workload in a VM, (just as we would it when running in the consolidated environment), but here we do not have any other VMs.

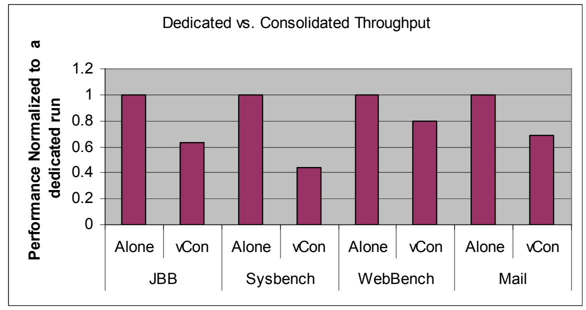

Figure 3: Performance Impact of Consolidation

Figure 3 shows the performance of the workloads running in dedicated vs. consolidated mode. We observe that SPECjbb loses about 37% when running with the other workloads while Sysbench loses 58% performance and Webbench loses 20% and mail loses about 32% performance in consolidation. The reason for this loss in performance is either due to the contention of the platform resources such as core, cache, memory and network or due to overheads of virtualization. SPECjbb being a compute intensive resource loses it performance due to core and cache interference only while Webbench and Mail perform network IO and hence their performance loss stems from the network contention and of course some from the cache and core contentions. We will look at this in more detail later in this section where we look at the architectural characteristics and also in the section on modeling.

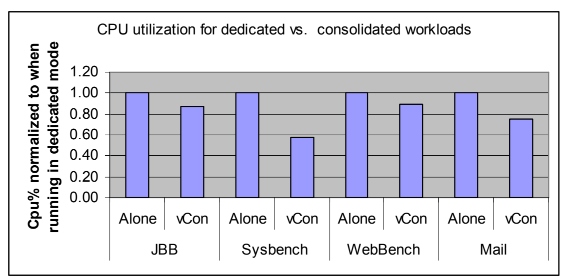

Figure 4: Consolidation impact to CPU utilization

Figure 4 shows the CPU utilization of the workloads and how that changes when they run in consolidated mode. We observe that SPECjbb and Webbench get most of their needed CPU resource reaching almost 87-89% of their original requirement. The most affected is Sysbench which barely gets 58% of its CPU requirement. This limitation on the CPU resource is due to core contention. Not only are the workloads contending for the cores but dom0 is also contending for the core as it does IO processing on behalf of the individual workloads such as Sysbench, Webbench and Mail. We will use this data in the section on modeling to quantify how much effect core contention has on the performance.

4.2 Architectural Characteristics: Detailed Look

A deeper dive into the architectural characteristics is necessary to see where the performance loss is coming from; we see that the CPI (cycles per instruction) increases almost inversely proportionately to the performance. For example as seen in Figure 5 the CPI increase for SPECjbb is about 37%, which is a significant fraction of the loss in performance for SPECjbb. Most of the increase in the CPI (expensive instructions) is due to the cache misses (L2 MPI) which increase due to the cache pollution caused by working with other workloads in the consolidated environment. We observe similar behavior with the other workloads as is shown in the graph [Figure 6] below. Again we will use this cache pollution/interference to develop a performance model.

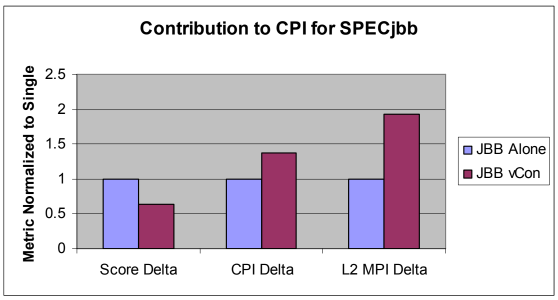

Figure 5: Architectural impact to SPECjbb in consolidation

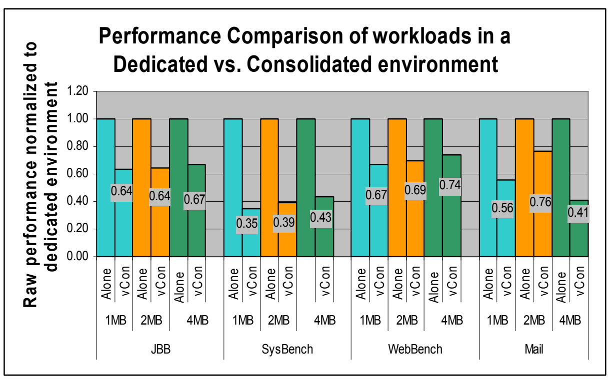

Figure 6: Workloads: Performance, CPI and MPI

4.2.1 Cache Scaling characteristics

Typically workloads will benefit if their working set size fits in the 2nd level cache. The working set size and characteristics' of the data access pattern determine how much a workload will gain by having bigger caches. Thus far, there has not been any work looking at cache scaling characteristics of consolidated workloads under virtualization. We have experimented with different cache sizes and shown how the individual VMs benefit with large caches.

Tables 2-5 shows data from the cache scaling measurements. The data shows that SPECjbb degrades by 33%-36% from running in consolidated mode irrespective of the cache size. The other workloads lose slightly more performance (5-7%) at smaller cache sizes. As can be seen from the tables, SPECjbb and Sysbench benefit from larger cache sizes while Webbench and mail do so moderately. The pathlength of these workloads remains almost the same across different cache sizes.

JBB in vCon	1MB	2MB	4MB
Jbb Score	1	1.31	1.78
Jbb CPI	1	0.77	0.57
Jbb L2 MPI	1	0.75	0.49

Table 2: Cache Scaling for SPECjbb (in vCon)

Sys in vCon	1MB-S	2MB-S	4MB-S
Sys Score	1	1.41	1.60
Sys CPI	1	0.83	0.76
Sys L2 MPI	1	0.70	0.57

Table 3: Cache Scaling for Sysbench (in vCon)

Web in vCon	1MB-S	2MB-S	4MB-S
Web Score	1	1.08	1.18
Web CPI	1	0.92	0.88
Web L2 MPI	1	0.84	0.69

Table 4: Cache Scaling for Webbench (in vCon)

Mail in vCon	1MB-S	2MB-S	4MB-S
Mail Score	1	1.15	1.09
Mail CPI	1	1.09	0.71
Mail L2 MPI	1	0.67	1.05

Table 5: Cache Scaling for Mail (in vCon)

4.3 Consolidation Execution Profile: Life of a VM

In this section, we look at how a VM behaves over time from a scheduling perspective. To understand the scheduling behavior we have instrumented the Xen scheduler to give us time profiles of which VM/vCPU is running on which physical CPU. This profiling is very extensive as the scheduler function is often invoked so we get extensive amounts of data for analysis. We have taken these dynamic profiles at various points in the benchmark run and analyzed them offline and present a statistical average of the various snapshots.

The first analysis here is to determine what the CPU utilization of each VM are using these profiles and compare that with those gathered using a standard tool like top/xentop. Our table below shows that the data from the scheduling closely matches that measured by xentop.

CPU% ----------- VM	Measured with Xentop	Computed from Scheduler Profile
dom0	30%	36%
JBB	122%	120%
Sys	116%	118%
Web	114%	112%
Mail	6%	8%

Table 6: Measured vs. Instrumented Data

Having established the validity of the data, we then determined the % of time spent by a VM on each physical CPU. This would give us an idea of how good the scheduler is at load balancing across the various VMs. As seen in the table below, the VMs run across all available physical CPUs. This is the representation of a single snapshot. Other snapshots have also shown similar distributions though the physical CPU is different.

Cpu% ---------- VM	Across all cpus	pCPU0	pCPU1	pCPU2	pCPU3
Dom0	100%	19%	33%	27%	8%
Jbb	100%	32%	28%	20%	25%
Sys	100%	26%	25%	28%	24%
Web	100%	18%	20%	27%	40%
Mail	100%	37%	23%	18%	2%

Table 7: CPU% distribution across physical CPUs

	Dom0	JBB	SYS	WEB	MAIL
% time came back to same cpu	95%	87%	92%	92%	97%
% Time went to another cpu	5%	13%	8%	8%	3%

Table 8: Affinity of VMs to physical CPUs

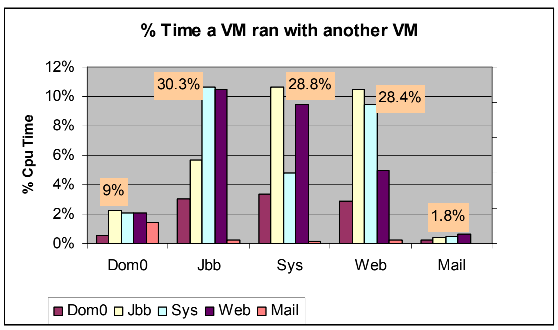

Figure 7: Percentage of time that two VMs run together on two cores of the same die

Table 8 shows the goodness of the scheduling algorithm, in how often a VM/vCPU comes back to the same physical CPU that it ran on previously. The VM may start on any CPU but once a VM has been scheduled on a CPU, even after it is preempted the scheduler attempts to place the VM on the same physical CPU, this is to improve cache behavior.

Now we will take a look at what the cache interference is likely to be in a consolidated environment and how one can estimate what the cache effects are. Since our experimental configuration is an oversubscription of the resources, namely we have 8 vCPUs (excluding dom0) and only 4 physical cpus, not all 8 vCPUs will be running at any given time. At some point of time we may have only 2 VMs, with 2 vCPUs from each VM, while at other points in time we may have 4 different VMs running, 1 vCPU from each. Given this situation it would be good to understand how often a VM runs with another VM. This will help us understand the interference impact because of consolidation on the workload. The interference effect can be estimated if one knows how much time is spent by the workload in question running with other workloads and what % of time. As seen in the Figure 7 for example of the 30.3% of the time spent in SPECjbb, 10.5% is with Sysbench, 10.2% with WebBench, 7.5% with JBB and about 3% with dom0 and 2% with Mail. We obtained this data by instrumenting the Xen scheduler to determine which vCPU/VM is running with which other VM/vCPU at any given point of time.

We also obtained an execution profile in time for the different VMs. The data in Figure 8 is a small snapshot (in time) of how the vCPUs run on which physical CPUs at any instant and thereby allows us to look at pairing of virtual machines within cores on the same socket, the time spent by each VM on the core before getting pre-empted, etc.

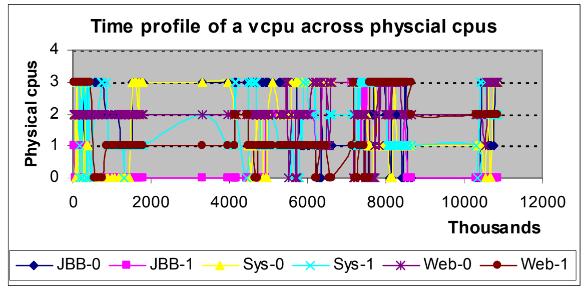

Figure 8: Execution Profile of vCPU across physial CPUs

4.3.1 Other Scheduling Experiments

In this section we look at different configurations of the VMs to understand better which configuration is best suited for a particular workload. We take each workload and experiment with three different scenarios; a) Affintize the vCPUs of the VM to 1 core of a socket and affinitize dom0 to the other three cores of the system and let all other workloads float. b) Affintize the vCPUs of the VM to a single socket, affinitize dom0 to the other socket while letting all other workloads float. c) Affintize the vCPUs of a VM to two cores of two different sockets, affinitize dom0 to the other cores of the different sockets while letting all other workloads float. The intent of configuration a) is to determine if the workload can perform on a single core alone. The intent of the second experiment is to see if the sharing of the cache between vCPUs of the same VM is sufficiently better than letting the work-load float around while the third c) is to remove the cache interference between the vCPUs of the same workload.

As can be seen from figures 9-13, none of the workloads perform well when they are affinitized to 1 core only. Reason being that one CPU is not sufficient to server the need if of the workload. However, that is not the only issue here: even if they were given 1 core, they were not able to hold on to it as they were preempted by the other workloads. So the performance impact we see is much worse than 50% of their normal performance (where they were allowed to float across any core).

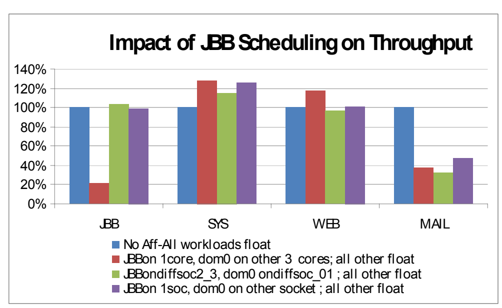

Figure 9: SPECjbb affinitzation impact on Throughput

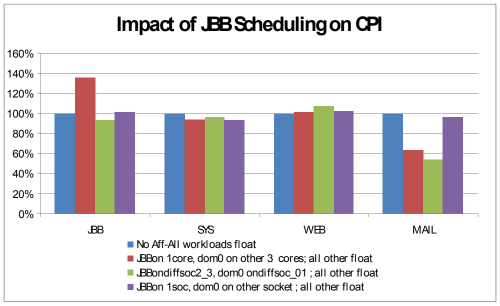

Figure 10: Impact of JBB scheduling on CPI

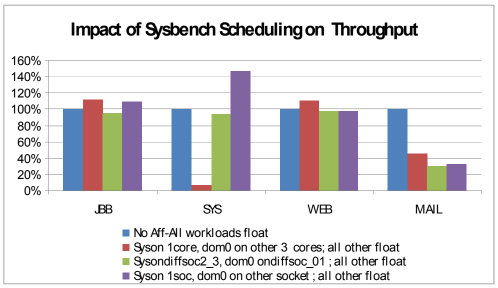

Figure 11: Sysbench Affinitzation impact

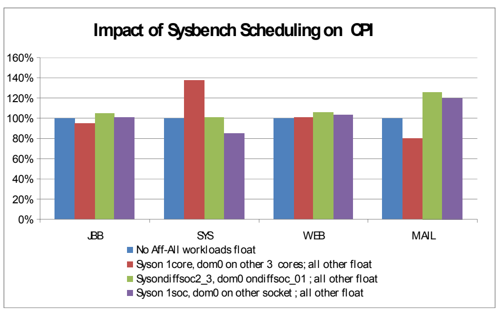

Figure 12: Sysbench affinization impact to CPI

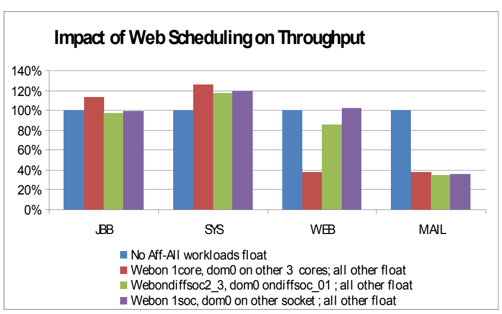

Figure 13: Webbench Affinitization impact

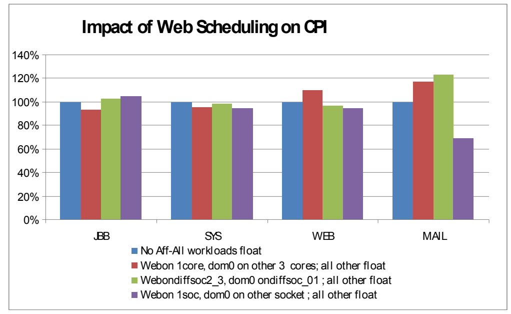

Figure 14: Impact of Webbench scheduling on CPI

In the second case where the two vCPUs of a VM were affintized to one socket, they did better than the normal run (where they were left floating). This is because of the cache sharing effects and the 2 vCPUs were able to take advantage to of the sharing. Sysbench really flourished in this situation with gaining 47% performance. Webbench performed nearly the same while JBB did almost equal to the normal run.

In the third scenario where we affinitized the cores to different socket cores, all of them did worse than normal expect for JBB which seemed to be slightly better than normal -4%. This is because JBB has no data sharing (so did not benefit being on the same socket) and so does not benefit either way. Affintizing to different sockets helped it a bit reducing any little cache interference it had with itself

Now, looking at the impact of affintizing one workload on another, both JBB and Sysbench did better when the other was affinitized to a complete socket, indicating that these two suffered from cache interference with each other.

5. From Characterization to Modeling

One primary motivation for collecting the characterization data is to help build a server consolidation performance that can be used to evaluate and guide the definition of future architecture options. In this paper, we will provide an overview of how we propose to build a performance model for virtualization workloads.

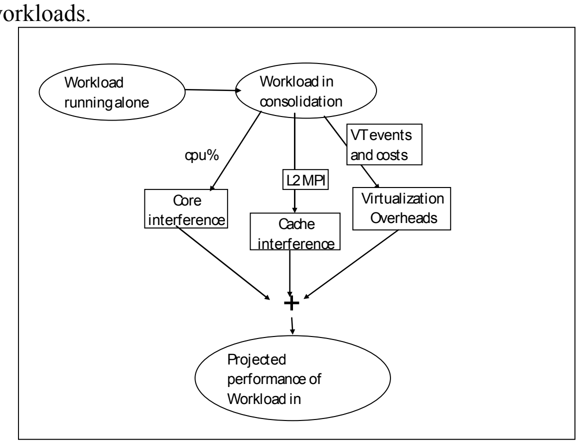

Figure 15: Towards building a performance model

In order to build a server consolidation model, the following are the key components that need to be developed:

1. A complete understanding of the performance behavior of each individual component workload that runs in a VM.

We assume that this information is available for the purposes of this paper.

2. An understanding of the performance behavior of each of the server workloads under virtualization, but running in dedicated mode. This allows for the modeling of virtualization overheads that will be incurred. In this paper, we start at this point. As shown in the measurements, we collected the base performance data with each workload running in isolation but within a virtual machine.

3. An understanding of the effects of consolidating a workload with several other workloads in a virtualized environment. In this paper, we focused on characterizing this aspect in more detail. Below, we will describe how this step can be further subdivided into several subcomponents that can be put together to build a consolidated server model.

For developing a server consolidation model (in particular for the third step (3) above), there are three important vectors to measure and analyze:

(a) **Core Interference:** Since the platform consists of a finite number of physical CPUs and each VM will demand some number of virtual CPUs, it is likely that the number of CPUs may be oversubscribed at any point in time. As a result the amount of CPU resources that are available to each virtual machine will be much lower and needs to be comprehended.

(b) **Cache Interference:** In a CMP platform, multiple cores share the last-level cache. Since different virtual machines will be running simultaneously on each of the cores sharing the cache, the effect of cache interference or sharing needs to be taken into account.

(c) **Additional Virtualization Overheads:** Since multiple workloads are running simultaneously in the platform, it is possible that additional virtualization overheads will be required. This could occur due to an increase in context switches, service VM instantiations, etc.

In the sections below, we will show an example of how go about capturing the behavior of steps (a) and (b) above. For the purposes of this illustration, we have chosen SPECjbb as the workload of choice.

5.1 Estimating core interference:

Having measured the performance of SPECjbb in virtualized mode but in isolation, we know that the SPECjbb workload uses 142% of the CPU resources in the platform (although it is configured with 2 virtual CPUs). Similarly, we know the CPU utilization of the other virtual machines that SPECjbb will be running with in consolidated mode. The total CPU consumption if we add up all of the virtual machine CPU consumption when running in isolation is about 453%. To project the CPU utilization of SPECjbb on a target platform with 4 CPUs, we need to normalize the utilization as follows: (142% / 453% * 4). This provides us an estimate of 125% which is fairly close to what we measured in consolidation (SPECjbb utilizes 122% of CPU utilization). This simple approach allows us to estimate that the slowdown that SPECjbb is likely to suffer due to core interference in consolidation as: (1 - 125% / 142%) = 12%.

5.2 Estimating cache interference:

In order to estimate cache interference for SPECjbb, it is important to know how often SPECjbb runs with each of the other VMs during consolidation. From our measurements (see Figure 7), we showed that a SPECjbb virtual machine VCPU runs with a Sysbench VCPU roughly 33% of the time, with Webbench roughly 33% of the time and with SPECjbb (the other virtual CPU) roughly 20% of the time. The remaining time is spent running with either DOM0 or with mail. We measured the performance loss of SPECjbb while running with Sysbench, Webbench and Mail separately. Our experimental setup to measure this is shown in the Figure 16. Our JBB VM has 2 vCPUs. We first performed an experiment with JBB alone cores of two separate sockets. Then we added Sysbench to the mix but affinitizing Sysbench, one to each core of the same socket where JBB was running. With the measure of the CPI, we found that SPECjbb lost 14% (Table 9) of it performance when running with Sysbench, 11% with Webbench and 3% with Mail and 16% with SPECjbb. Summing up these performance losses by weighting them against the fraction of time that they run together, we get about 14% performance losses to SPECjbb from cache interference.

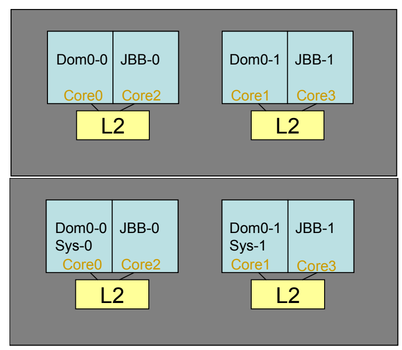

Figure 16: Experimental setup for estimating cache interference

	JBB	JBB with JBB	JBB with Sys	JBB with Web	JBB with Mail
Score	1.00	0.84	0.86	0.89	0.97
CPU %	1.00	1.06	0.99	0.99	0.99
CPI	1.00	1.25	1.11	1.13	1.04
L2MPI	1.00	2.72	1.51	1.33	1.11

Table 9: Interference Analysis for SPECjbb

Taking the core interference and the cache interference into account, we found that the performance loss for SPECjbb due to consolidation is roughly 26%. We expect that the additional performance loss due to virtualization is likely the third contributor to the overall loss of 37% that we observed in measurements (see Figure 3). The steps required to fully validate and complete the modeling framework is left to future work and is not within the scope of this paper.

6. Conclusion and Future work

In this paper we have focused on a detailed characterization and analysis of a server consolidation workload. We presented our findings using the vConsolidate benchmark that has been developed by Intel. Our experiments' have shown that the performance of any workload suffers considerable loss when it is run in a consolidated environment. This is to be expected but getting a deeper understanding of what the workload expects will help the IT administrators do a better job in making decision about provisioning the available server resources and help in deploying the applications. Most of the performance loss of the CPU intensive workload SPECjbb comes because of cache and core interference. SPECjbb suffers performance loss when run with Sysbench but that loss can be mitigated if we affinitized the vCPUs of Sysbench to one socket. That reduces the cache interference and hence performance degradation. Our cache scaling studies show that both SPECjbb and Sysbench benefit from large caches while the Webserver and Mail server workloads need more network bandwidth.

We performed several scheduling experiments to determine how often a given workload A runs with another workload B so as to understand the potential impact to performance of A when collocated with B. The analysis of this is helpful to VMM developers to design better scheduling algorithms that are tailored for consolidating workloads in the virtual environment. Finally we presented our initial proposal for creating a consolidated server performance model to project the performance of virtualized workloads on future platforms. We have taken the effects of cache and core interference in this preliminary model. For the future we need to look at the third aspect of performance impact, namely the additional virtualization overhead due to consolidation. To determine the virtualization overhead we need to measure the performance of the workload on a native platform (not running a hypervisor/VMM) and then compare the performance on a virtual machine. We need to measure the cost and number of events such as context switches, interrupts and page faults. These events occur in every environment but the cost of these events changes with virtualization. We will then need to factor-in these costs overhead in the model to predict the performance degradation due to virtualization. Another aspect of study that has not been addresses in this paper is the scaling, which is when we have multiple CSUs. Future work in this area is abundant since the performance analysis efforts on virtualization and consolidation are still in their infancy and need considerable work to bring to maturity.

References

[1] A. Menon, J. R. Santos:, *http://xenoprof.sourceforge.net/xenoprof_2.0.txt*

[2] A. Menon et al. *Diagnosing Performance: Overheads in the Xen Virtual Machine Environment*. In First ACM/USENIX Conference on Virtual Execution Environments (VEE'05), June 2005.

[3] A Singh. *An Introduction to Virtualization*. http://www.kernelthread.com/publications/virtualization

[4] N. Enright Jerger, D. Vantrease, M. H. Lipasti, *Evaluation of Server Consolidation Workloads for Multi-core* Designs University of Wisconsin – Madison, upcoming IISWC-2007

[5] D. Gupta, Gardner, Rob; Cherkasova, Ludmila *XenMon: QoS Monitoring and Performance Profiling Tool*, http://www.hpl.hp.com/techreports/2005/HPL-2005-187.html

[6] Intel Virtualization Technology Specification for the IA-32 Intel Architecture, April 2005.

[7] Intel Corporation, "Tera-Scale Computing," http://www.intel.com/research/platform/terascale/

[8] Intel Corporation. "Intel Dual-Core Processors," http://www.intel.com/technology/computing/dual-core/

[9] Intel Corporation, "World's first quad-core processors for desktop and mainstream processors," http://www.intel.com/quad-core/

[10] Intel Corporation, "Intel Vtune Performance Analyzer", http://www.intel.com/software/products/vtune/239144.htm

[11] L. Cherkasova and R. Gardner,"*Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor*," Proceedings of the USENIX Annual Technical Conference, April 2005.

[12] M. R. Marty , M.D. Hill *Virtual Hierarchies to Support Server Consolidation,* ISCA 2007

[13] Microsoft Corporation www.microsoft.com

[14] P. Apparao, S. Makineni, D. Newell, *Characterization of network processing overheads in Xen*, 2nd International Workshop on Virtualization Technology in Distributed Computing (VTDC 2007)

[15] R. Iyer, "*On Modeling and Analyzing Cache Hierarchies using CASPER,*" 11th MASCOTS, Oct. 2003

[16] J. P. Casazza, M. Greenfield, K. Shi, *Redefining Server Performance Characterization for Virtualization Benchmarking***,** Intel technology Journal, Volume 10, Issue 03

[17] R. Uhlig, et al., *"Intel Virtualization Technology*," IEEE Computer, 2005.

[18] V. Chadha, R. J. Figueiredo, R. Illikkal et al. *I/O Processing in a Virtualized Platform: A Simulation-Driven Approach, VEE'07* June 13–15, 2007, San Diego, California, USA

[19] VMware Inc, www.vmware.com

[20] C. A. Waldspurger *Memory Resource Management in VMware ESX Server* Proceedings of the 5th Symposium on Operating Systems Design and Implementation Boston, Massachusetts, USA December 9–11, 2002

[21] Xen: *The Xen Virtual Machine Monitor*. http://www.cl.cam.ac.uk/Research/SRG/netos/xen/architecture.html