



Characterization & Analysis of a Server Consolidation Benchmark on Xen

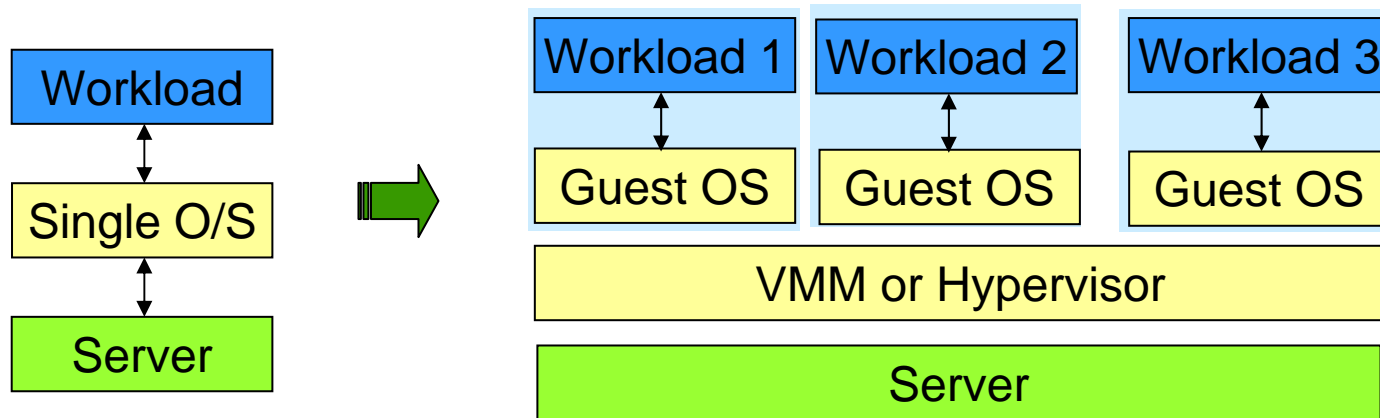
Padma Apparao

Ravi Iyer, Don Newell,
Xiaomin Zhang, Tom Adelmeyer

Intel Corporation
Nov 16th, 2007

Background

- Virtualization and consolidation are a growing trend in datacenters
 - > 40% of servers expected to run a consolidated workload by 2010



- Problem is there is no analysis methodology or performance studies in place for understanding consolidated workloads

Motivation

- Performance characterization is useful for
 - Providing feedback to IT administrators
 - Deployment with fair share of resources to end-users is challenging with virtualization
 - Providing feedback to Platform Architects
 - To project future platform performance
 - How do apps scale for future platforms
 - To optimize future architectures for consolidation
 - Architectural Effects on consolidation
 - Cache and other resource sharing effects
 - Virtualization overheads effects
 - Providing feedback to VMM developers
 - How are the platforms resources (cores/IO devices scheduled)
 - Scheduling heuristics may be suboptimal without an execution profile

A Consolidation Benchmark

- vConsolidate (vCon) is one of the proposed benchmarks for virtualization consolidation
 - Developed by Intel
 - VMM agnostic
- VMmark is another consolidation benchmark developed by VMware
- vSPEC is a virtualization benchmark being defined by the SPEC committee

- Our focus is vCon



vConsolidate Benchmark Configuration

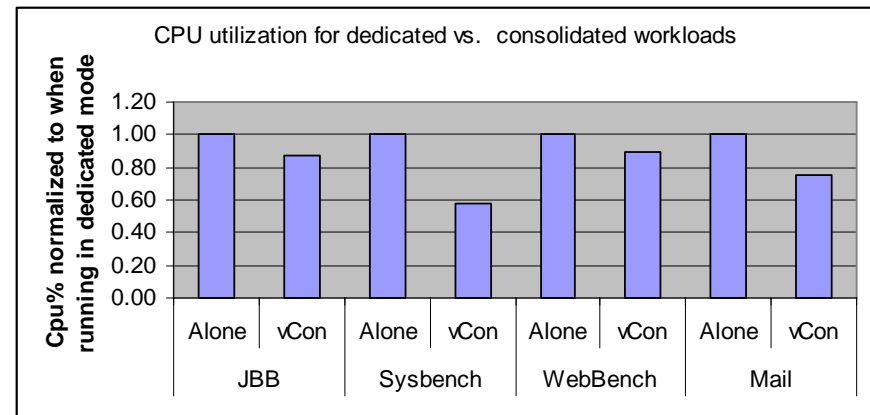
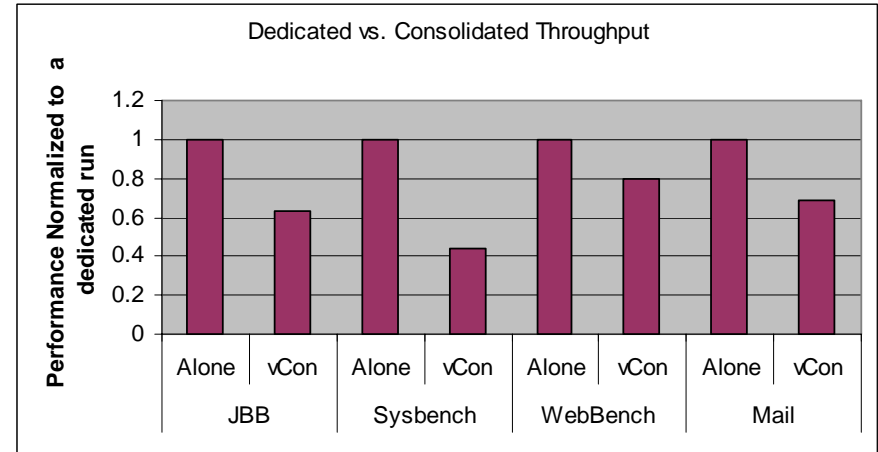
- Various profiles for vCon.
- We chose profile 3
- System config:
 - Intel dual socket core2-duo machine
 - Core2-duo processors at 3GHz/4MB second level cache
 - 16GB system memory
 - Intel VT technology
- Tools:
 - Xentop/sar
 - Virtual Emon developed by Intel
 - Xen code instrumentation

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	

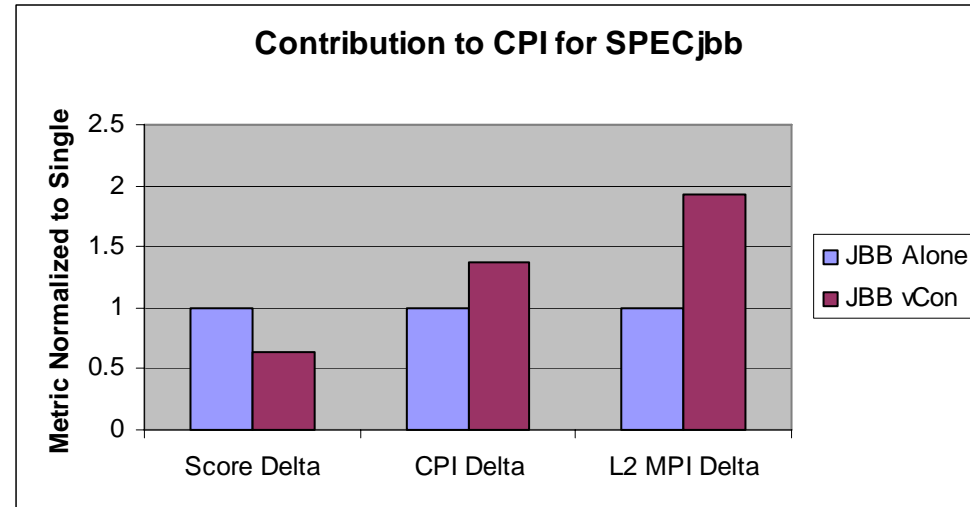
Results and Analysis: Performance Impact

- Workloads are run alone within a single VM for the dedicated measurements.
 - SPECjbb loses 37% in consolidation
 - Sysbench loses 58%
 - Webbench loses 20%
 - and Mail loses 32%
- Degradation likely to resources like core/cache/memory/IO/network contention and due to virtualization overheads
- Cpu utilization reduction is due to core contention



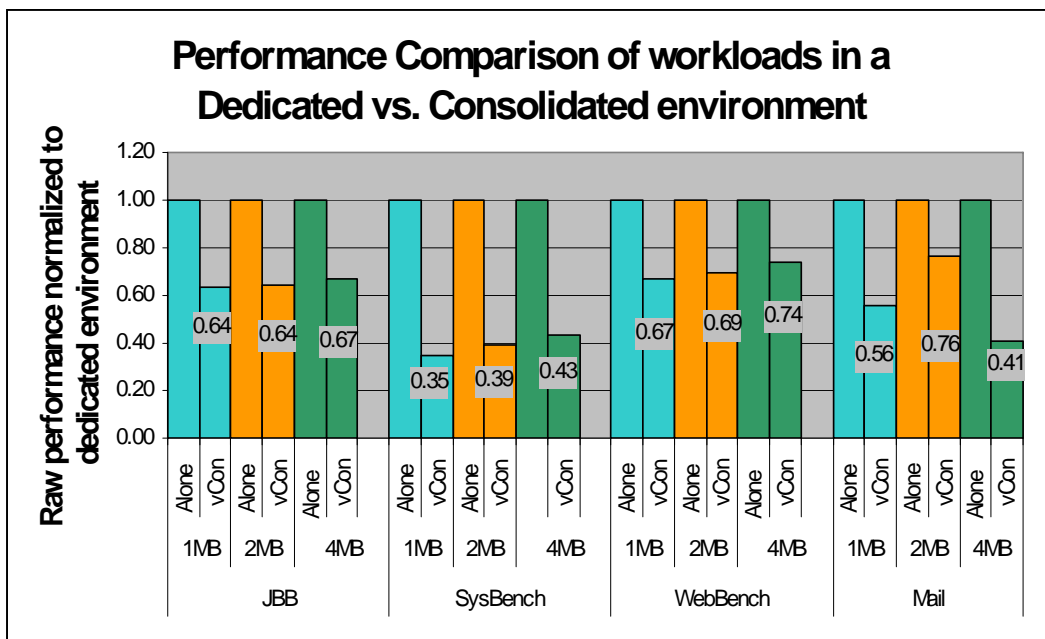
Results and Analysis: Architectural characterization

- Understand where the performance loss is coming from
 - jbb CPI (cycles per instruction) increases 37%
 - Most of CPI increase is due to L2 MPI (misses per instruction) increase
 - Due to cache pollution when running with other workloads
 - Cache interference
 - Similar behavior observed with other workloads too.



Results and Analysis: Cache Scaling

- Useful for understanding the benefit of larger caches to the workloads.
 - Jbb and Sysbench do well with larger caches.
- Helpful for platform architects in machine decision about cache sizes for future platforms



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

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

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

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

Cache Scaling for Mail (in vCon)



Results and Analysis: vCon Execution Profile – Life of a VM

- Understand how a VM behaves over time
- Instrumented the scheduler to give us data like on which pcpu the VM is running , how long, where it did migrate to, when did it come back and who ran while it had migrated
 - Help understand cache interference
 - Help understand the behavior of the scheduler
- We measured cpu utilization with xentop and using our instrumentation
- Data pretty close validating our methodology

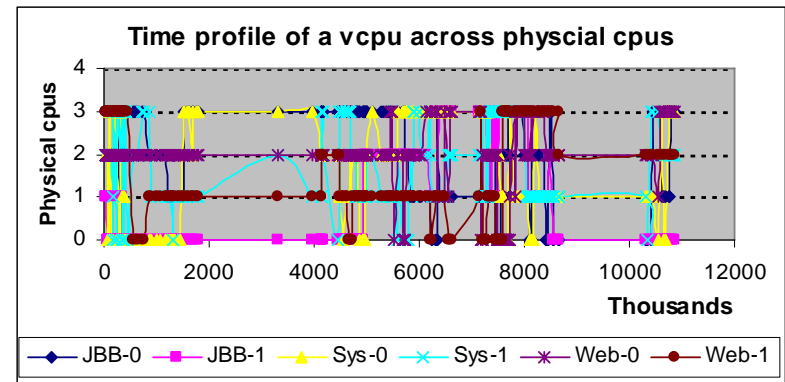
CPU% ----- VM	Measured with Xentop	Computed from Scheduler Profile
dom0	30%	36%
JBB	122%	120%
Sys	116%	118%
Web	114%	112%
Mail	6%	8%

Results and Analysis: vCon Execution Profile – Life of a VM

- A VM runs on all pcpus, no particular affinitization
 - Shows good dynamic load balancing by the scheduler.

Cpu%	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%

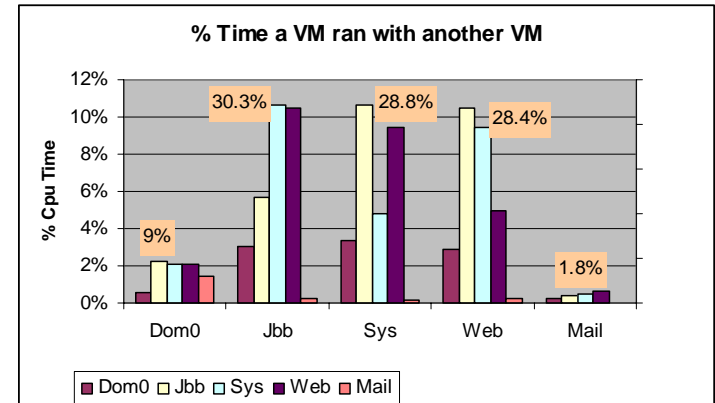
- A VM comes back to the same pcpu most of the time
 - Helps in reducing cache misses, if the cache was not polluted
 - What is missing is: what VMs ran during the interim that the VM had migrated → extent of cache pollution/ quantification of cache misses



	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%

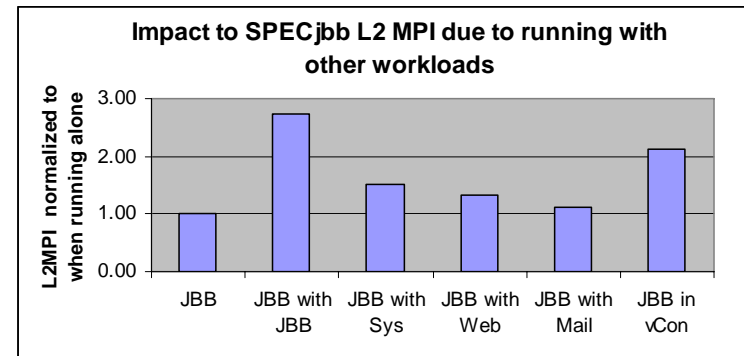
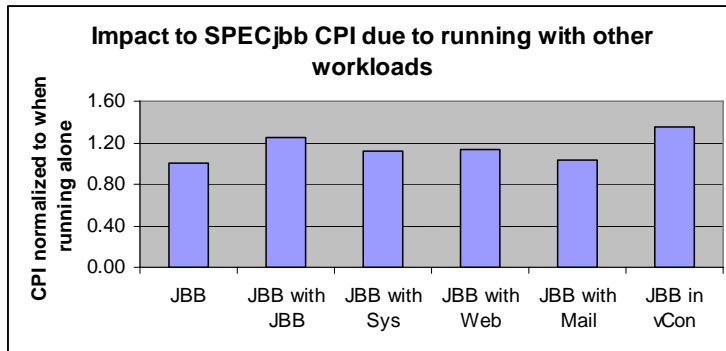
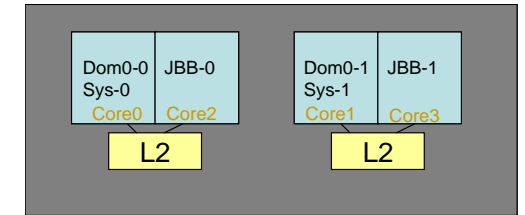
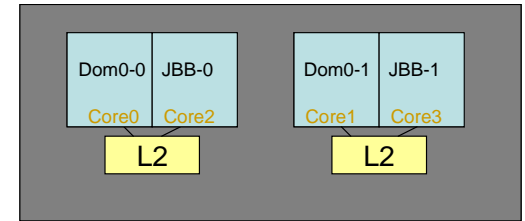
Results and Analysis: vCon Execution Profile – Cache Interference

- Cache interference impacts the performance of the workload
- Find out which Vm/vcpu shares the second level cache with another VM/vcpu and for how much time
- Of the 30.3% spent in jbb, 10.5% is spent with sysbench, 10.2 % with webbench, 7.5% with jbb (other vcpu) 2% with Mail
- Knowing the L2 MPI and CPI impact of a VM with another VM we can determine the cache interference



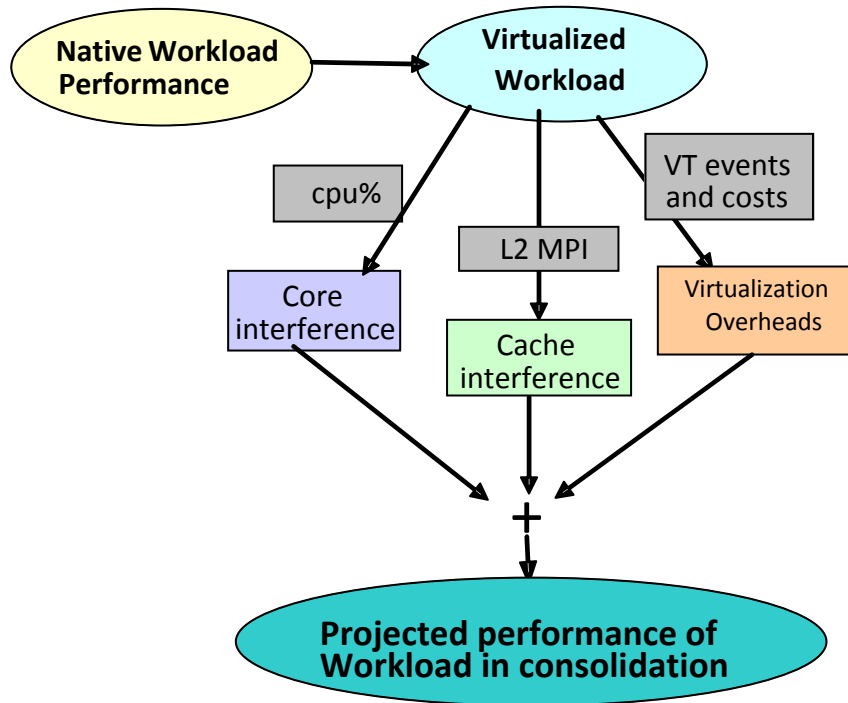
Results and Analysis: vCon Execution Profile – Cache Interference.....

- Affintize the vcpus to different cores
- Get cache MPI with each of the workloads
- Jbb loses 16% running with JBB, 14 with Sysbench, 11 with Webbench and 3% with Mail



Form characterization to Modeling

- How do we build a performance projection model



What kind of hooks would we like?

- How can we get the execution profile of a VM without instrumentation?
- Have some logging in the VMM that gives:
 - Where (which pcpu) did the VM start running, what other VMs were running on all other pcpus.
 - Where (which vcpu) did the VM/vcpu migrate to?
 - When did the vcpu come back to the same pcpu?
 - How long did it run on the pcpu before migration?
 - What was the longest duration a VM ran on any pcpu/with another VM
- While the VM was running, what events (context switches, interrupts (external and ipis) page faults, tlb flushes etc were generated, and what was the time taken by the VMM to service these events.

Summary

- Need for performance characterization of a consolidation benchmark
- Introduction of the vConsolidate benchmark developed by Intel
- Dedicated vs. Consolidated performance
 - Sysbench and SPECjbb lose the most performance
 - Degradation due to contention of resources
- Architectural characterization for cache interference
 - Scheduling studies to understand the life of a VM/vcpu
 - Affinitization studies to understand cache interference
- Overview of building a performance projection model
- Xen hooks needed



Thank you

Questions?



