

WHITE PAPER

Virtualized SAP Performance with VMware vSphere™ 4



Table of Contents

Introduction	3
Experiment Configuration and Methodology	3
Workload Description	3
Experimental Setup	3
SAP Server	3
Load Generator	4
Storage	4
Results	5
Hardware Nested Paging	5
Transaction Processing (Windows)	6
Memory Throughput Load (Linux)	7
Scale-Up	8
Overcommit Fairness	9
Large Pages	10
Recommendations	11
Conclusion	11
Acknowledgments	12
About the Author	12
References	12

Introduction

SAP provides a range of enterprise software applications and business solutions to manage and run the business processes of an entire company. The mission critical role of this software requires reliability, manageability, high performance, and scalability.

VMware vSphere™ 4 can help customers manage a smarter SAP application-based landscape via template-based virtual machine cloning; distributed resource scheduling and power management (VMware DRS and DPM); and improved uptimes through VMotion, VMware HA, and Fault Tolerance. However, these management and reliability features would be less appealing if performance were to suffer. This paper demonstrates that vSphere supports virtualized SAP Enterprise Resource Planning software with excellent performance and scalability. By supporting physical CPU features such as *large pages*, *AMD Rapid Virtualization Indexing*, and *Intel Extended Page Tables*, vSphere exhibits low overhead when running SAP with even its most MMU-intensive memory models. VMware Virtual SMP support allows SAP software to scale-up (utilize more CPUs) with up to 95 percent of physical machine performance.

This paper includes results of several experiments using vSphere and SAP software with both the Microsoft Windows Server 2008 and SUSE Linux 10.2 operating systems. It presents performance gains due to vSphere's support for hardware nested page tables, performance data from scale-up scenarios, and the efficiency of vSphere's resource management. You will also find best practice recommendations based on these experiments.

Experiment Configuration and Methodology

Many performance and sizing studies were conducted in VMware's labs to measure, analyze, and understand the performance of SAP in virtual environments. The following sections describe the workloads used for the majority of the tests and present the hardware and software setup.

Workload Description

The performance of SAP on vSphere was studied with a popular online transaction processing workload used to size SAP deployments. The workload simulates sales and distribution scenarios with concurrent users creating orders, entering delivery information, posting a goods issue, and performing other typical sales and distribution tasks. There are 10 seconds of "think time" between each transaction.

SAP deployments may be configured in a two-tier or three-tier configuration. All experiments in this paper used a two-tier configuration wherein the database and application server share the same physical host or virtual machine. An automated load generator served as the presentation tier. The load generator slowly ramps up the number of concurrent users until a preset maximum is reached. After ramp-up is complete, response times were measured for 10-15 minutes. The number of concurrent users that can be served while maintaining an average response time below two seconds was recorded. The experiments were conducted in good faith and run to completion without errors or warnings, but the results in this paper have not been certified by SAP.

Experimental Setup

The two-tier setup consisted of 1) a database and application server (SAP Server) and 2) a presentation server (Load Generator). The following sections describe the hardware and software stacks for each server including the storage attached network (SAN) system configuration.

SAP Server

In addition to tests with virtual machines, tests were run in a physical environment to provide context for the virtual results. For an "apples-to-apples" comparison, identical hardware was used for both physical and virtual tests. The server was rebooted to enter either the Windows environment for physical tests or the vSphere environment for virtual tests.

- Server: Dell PowerEdge 2970
- Processors: 2 AMD Opteron 2356 processors at 2.3GHz (8 cores total)
Third-generation AMD Opteron processors implement two technologies designed to assist virtualization. Instruction set extensions simplify execution virtualization, and Rapid Virtualization Indexing (RVI) provides capabilities for nested memory translation [1].
- Memory: 32GB
- Virtualization Software: VMware vSphere 4 (Release Candidate)
- Operating System: Windows Server 2008 Enterprise x64 Edition (Build 6001 SP1)

- Database: Microsoft SQL Server 2005 x64 SP2
- SAP: ECC 6.0 / NetWeaver 7.0 SR2 with 64 bit Unicode kernel (7.00 PL146)
- The Server's local storage had two partitions:
 - NTFS: Physical system disk (Windows, VMware Tools, SAP, and SQL Server)
 - vmfs3: Virtual Machine system disk: (Windows, SAP, and SQL Server)

Load Generator

A physical machine was used as the load generator. It emulated multiple users interacting with the SAP Server.

- Server: Dell PowerEdge 1950
- Processors: 2 Intel Xeon 5160 processors at 3 GHz (four cores total)
- Memory: 8 GB RAM
- Operating system: 64-bit Windows Server 2008 Datacenter Edition SP1
- Disk: 1 x 10KRPM 146GB disk drive

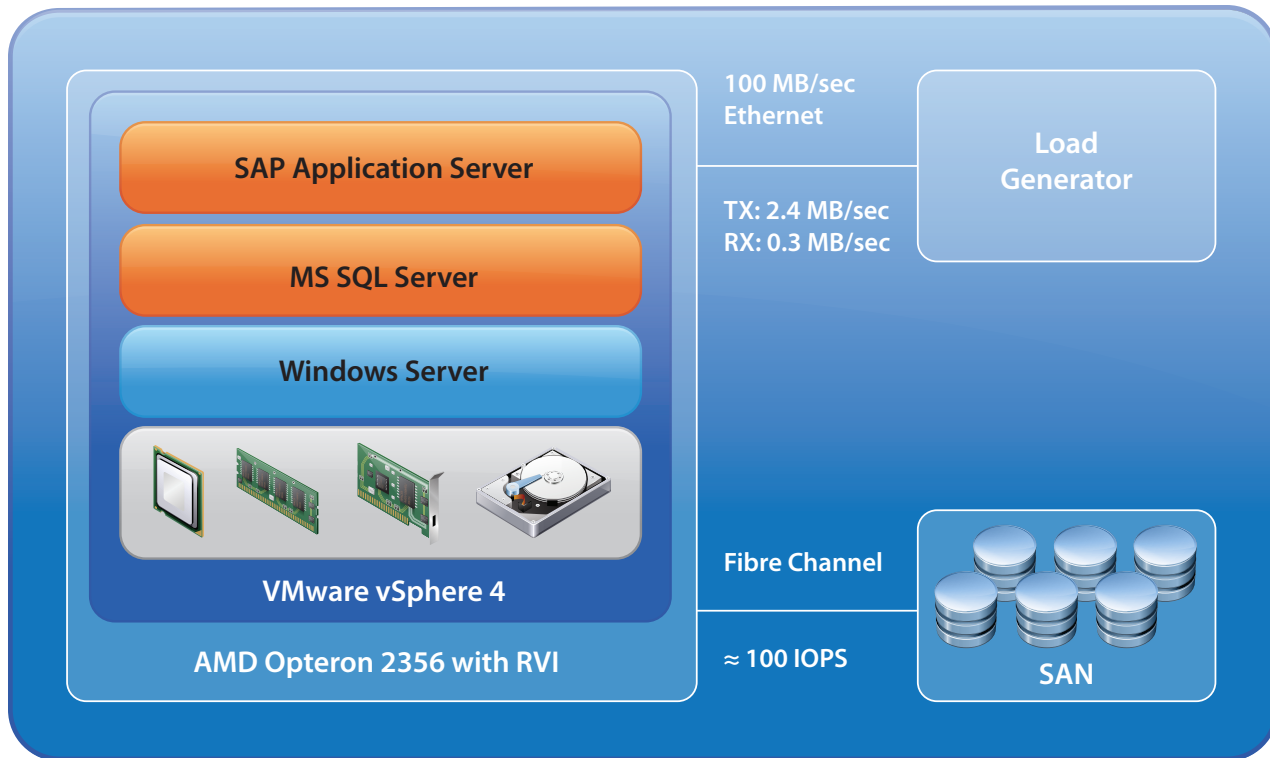
Storage

The data and log files of the database were deployed on an EMC CX3-40 SAN (Firmware version R4F0) with the configuration details shown in [Table 1](#). The database was stored on a logical unit formatted with NTFS and exposed to the virtual machine via raw device mappings so that the same disks and files could be used in both the physical and virtual environment.

Table 1: Storage array configuration

RAID Group	RAID Level	Disks <i>15K RPM 146 GB Fibre Channel (4Gbps)</i>	LUNs
0	0	15	Database data files
1	0	15	Database log files, Windows swap file

Figure 1: Two-tier SAP Application Load Test Architecture



The hardware and software were connected as shown in [Figure 1](#). Network and storage links are labeled with the observed traffic. The link capacity greatly exceeds the bandwidth generated by the workload.

Results

Using the testbed described above, the performance benefit of nested paging is highlighted. The next result focuses on the scale-up behavior of SAP in physical and virtual environments. Next, scheduler fairness is demonstrated when the host server is overcommitted. Finally, it is shown how the use of large pages can improve SAP performance.

Hardware Nested Paging

AMD Rapid Virtualization Indexing (RVI) and Intel Extended Page Tables (EPT) are hardware technologies that support the multiple layers of memory address space translation needed to run virtual machines. Prior to these technologies, vSphere translated memory addresses in software with a technique called shadow page tables. Shadow page tables may cause a time and space overhead as vSphere must monitor the page tables of a guest operating system to ensure coherence with the underlying mapping to the physical host's memory.

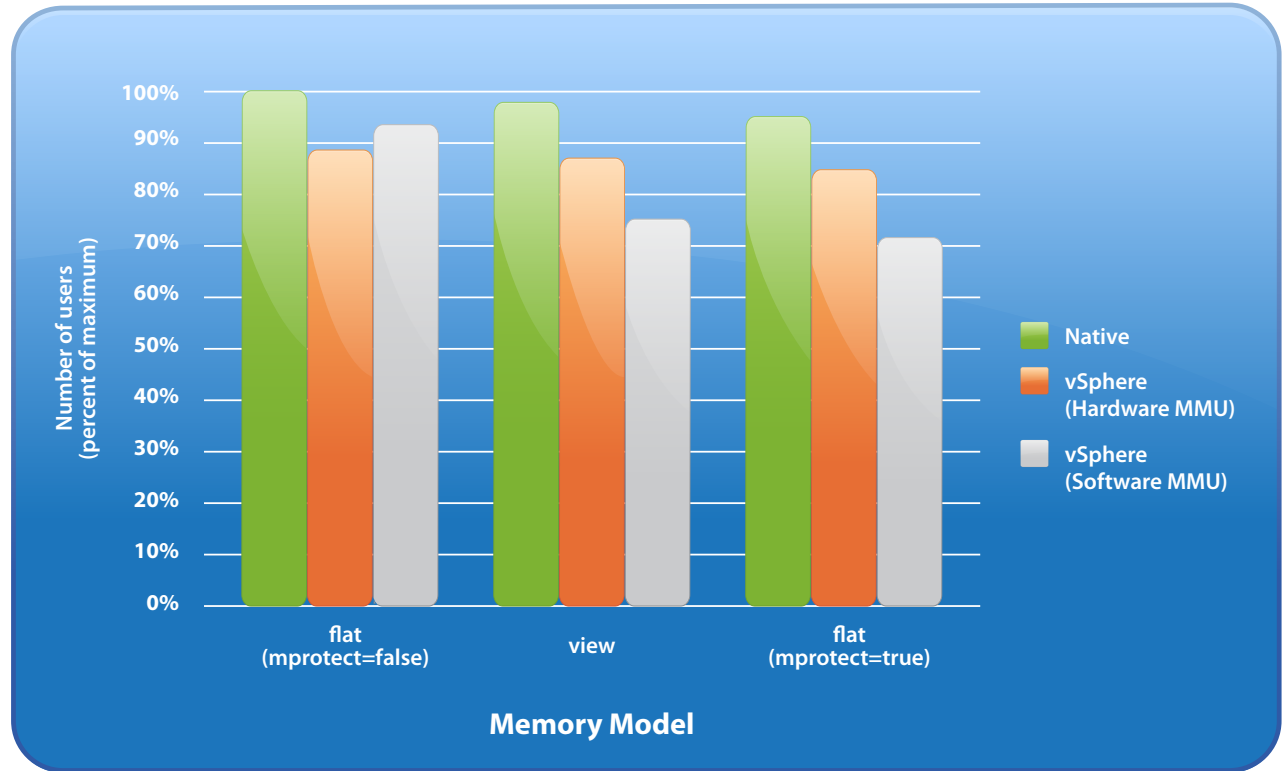
As RVI and EPT move virtual machine page table management from software to hardware, VMware often refers to the support for this technology as Hardware MMU support. A Hardware MMU typically offers higher performance for workloads that heavily manipulate the guest operating system's page tables. If these workloads also exhibit significant translation lookaside buffer (TLB) misses, however, performance may suffer. This can be mitigated by the use of large pages as discussed in [Large Pages](#). A more complete discussion of the trade-offs between a Hardware MMU and traditional Software MMUs on various workloads may be found in VMware white papers [2, 3].

This section focuses on the effect of a Hardware MMU on SAP performance in a virtual machine.

Transaction Processing (Windows)

SAP can be configured to use two memory models in Windows: View and Flat. A memory protection (mprotect) option for the Flat mode results in three effective modes, but SAP recommends either the View or default Flat mode (mprotect enabled) for production use [4,5]. To show the impact of hardware support for MMU virtualization, the MMU virtualization technique is manually toggled using the Virtual Infrastructure Client GUI¹. Figure 2 shows the number of users achieved with each memory model in a 2-CPU experiment. The results are normalized to the highest user count achieved in a physical (native) environment.

Figure 2: vSphere's Hardware MMU support minimizes overhead for SAP's production memory models on Windows (View and Flat with mprotect).



vSphere's ability to support hardware nested page tables (RVI in this case) markedly improved performance in the two production memory models. A two-way virtual machine achieved 89 percent and 90 percent of native performance for the View and Flat (mprotect=true) modes, respectively. This represents a 15 percent and 20 percent improvement over the software MMU performance.

For completeness, the figure includes data showing performance using the unsupported Flat memory model with mprotect disabled. This mode is often used during benchmarks by server vendors to assess peak performance. There are fewer page table manipulations in this mode, so the Hardware MMU provides little benefit. In fact, the higher cost of a TLB miss with the Hardware MMU is enough to slightly reduce the performance compared to running with the traditional Software MMU. TLB miss costs may change in future hardware generations, altering the hardware-software gap.

Another important take-away from Figure 2 is that the difference between the maximum and minimum user counts is less than 6 percent in native and in optimally configured virtual environments. This is a much better result than tests done on older hardware with Windows 2003. Those tests showed higher virtualization overhead in the View mode. They also saw a dramatic decrease in absolute performance on both native and virtual machines when mprotect was set to true, though the relative performance was similar to current results. The Hardware MMU has addressed the higher-than-desired View virtualization overhead, and changes in Windows have greatly improved the absolute performance with mprotect enabled.

¹ Edit Settings... Options Tab / Advanced:CPU/MMU Virtualization

Memory Throughput Load (Linux)

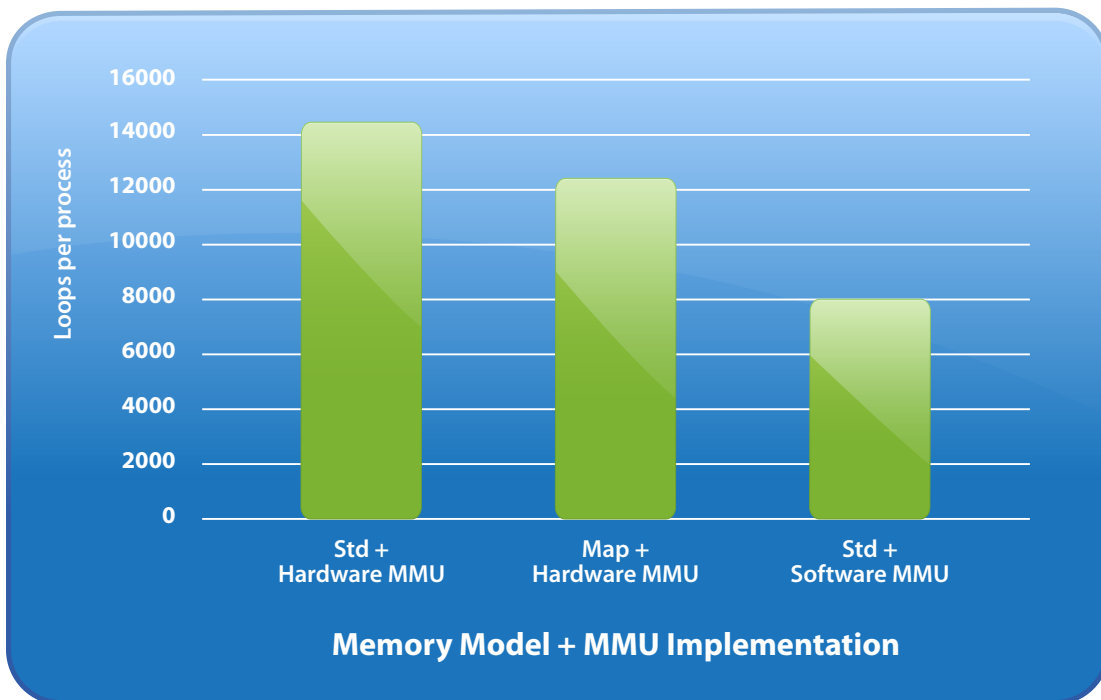
The impact of vSphere’s Hardware MMU support was also tested using the EPT implementation from Intel. The hardware and software used for these tests differs from that described above.

- Server: SuperMicro X8DTN
- Processors: 2 Intel Xeon X5570 processors at 2.93GHz (8 cores total)
This processor implements Intel VT-x to assist virtualization of the CPU and Extended Page Tables (EPT) to provide capabilities for nested memory translation.
- Memory: 72GB
- Virtualization Software: VMware vSphere 4 (Release Candidate)
- Operating System: x64 SUSE Linux 10.2
- Database: MAXDB 7.7.04
- SAP: ECC 6.0 / NetWeaver 7.0 SR2 with 64-bit Unicode kernel
- Storage: iSCSI filer with 24 1TB disks in a RAID 5 array. Ten 300GB LUNs

Instead of the transaction processing workload, the `memory_load` script was run from the SAP Linux Certification Suite (SLCS), a “stress test” that heavily loads an SAP system. According to documentation supplied with the SLCS, `memory_load` generates a 300MB internal ABAP table and writes random data into this table. It then loops over this table for 900 seconds. During each loop it reads one data set randomly. The overall number of loops per second is reported. For every available CPU in the system, the script configures three dialog work processes. Each process runs one report. For example, when the server has two CPUs, the script configures a total of six work processes, allocating approximately 1800MB [6]. The results have a ±3% margin of error.

SAP on Linux may be run in two memory modes: Standard (Std) and Map, with Std being the default for 64-bit Linux installations. The difference between the first and third bars in Figure 3 shows the impact of the Hardware MMU (EPT) on this memory-intensive workload. Specifically, running a virtual machine with a Hardware MMU increased the loops-per-process in Std mode by 82 percent of the amount achieved with the Std mode using a Software MMU. Note that VMware’s support for CPUs with a Hardware MMU means that the Map mode is not recommended when a Hardware MMU is available.

Figure 3: EPT performance on Linux



Scale-Up

To measure scale-up performance, the sales and distribution workload described earlier was run with an increasing number of CPUs in both a native and virtual environment. The baseline consists of 1-, 2-, 4-, and 8-way experiments conducted in the native environment. The `bcdedit` tool was used to force Windows to boot into the configurations shown in Table 2. In the virtual environment, a virtual machine was configured to use corresponding processor counts and memory.

Table 2: Physical machine boot configurations

Number of CPUs	Memory (MB)
1	5,500
2	9,500
4	15,000
8	28,000

To allow the SAP software to take advantage of multi-core CPUs, it can be configured with multiple *instances*. One instance was used for each of the *n* cores in the configuration: a central instance and *n* - 1 dialog instances. The central instance was comprised of two dialog processes, one update process, and one enqueue process (along with a batch and spool process which were mostly idle). Each dialog instance had three dialog processes and one update process. Affinity was set in the SAP instance profiles so that each instance ran on its own processor. The database ran on the same CPU as the central instance. This configuration is similar to that used by certified SAP 2-Tier benchmarks on similar hardware [7]. SAP's Flat memory model with `mprotect` disabled was used for peak performance. The VMs were configured to use the Software MMU as it provided the best performance for this memory model.

Figure 4 shows the performance of each configuration normalized to the number of users supported in a native uniprocessor system. It shows that the native SAP system scaled almost linearly from 1-4 instances and the virtual machine performed almost as well. Native and virtual performance did not scale as well to eight CPUs, but still exhibited sizable performance gains.

Figure 4: SAP scale-up in physical and virtual environments

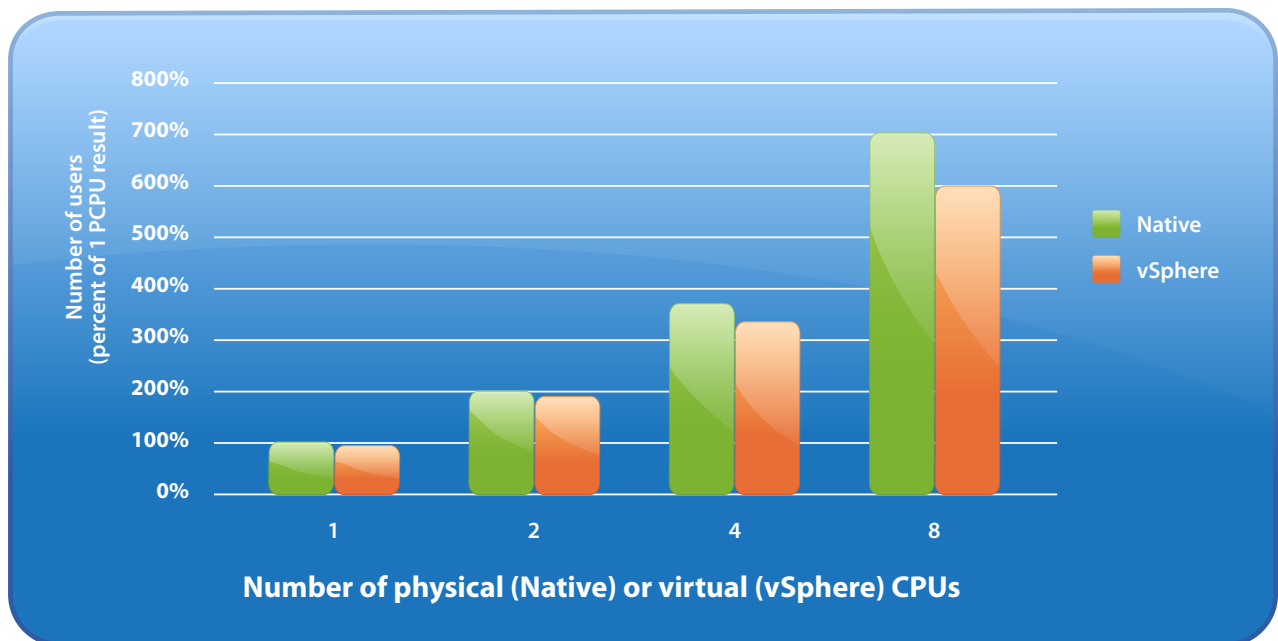


Table 3: Ratio of supported users (Virtual ÷ Native)

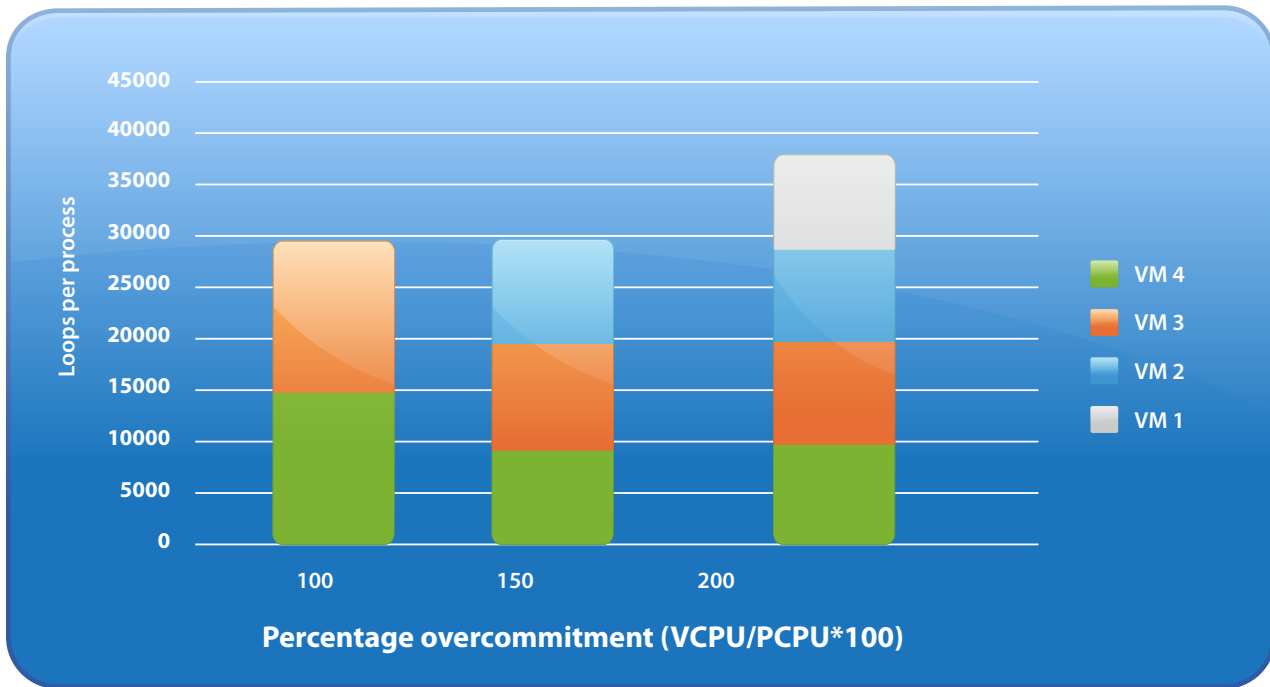
(V)CPUs	Ratio to Native
1	94%
2	94%
4	95%
8	85%

Table 3 indicates the near-native performance of SAP in a virtual machine. Note that this table is slightly different from Figure 4. The table compares the number of users supported on configurations of similar size, while the figure normalizes each result by the number of users in a native uniprocessor configuration. The high ratio of virtual to physical performance, especially in 1-4 VCPU virtual machines, presents the administrator with many configuration possibilities. In future work, VMware plans to examine a scale-out scenario in which several of the smaller, most efficient virtual machines work together on this sales and distribution workload.

Overcommit Fairness

When multiple identical virtual machines are running, one expects identical performance from each. To verify this property, multiple 4-way, 16GB virtual machines were instantiated on an 8-way server to create CPU overcommit scenarios of 100 percent, 150 percent, and 200 percent; memory was not overcommitted. Then, the `memory_load` test was run in each virtual machine and the throughput, measured in loops-per-process, was observed. Figure 5 shows the effect of overcommitting the CPU resources of the machine using the hardware and software from [Memory Throughput Load \(Linux\)](#). The total height of each bar represents aggregate throughput while the components of the bar show the contribution from each virtual machine. The fairness properties of the vSphere scheduler are clearly seen: when equal shares are specified, no virtual machine received substantially more CPU time than its peers.

Figure 5: CPU overcommitment performance on Linux



Before interpreting the aggregate throughput result, it is important to note that the hardware platform has a non-uniform memory access (NUMA) topology. Memory access time is reduced when a processor is on the same NUMA node as the memory chip it is accessing. The server used for these tests has two NUMA nodes, each with four cores and 36GB of memory. A virtual machine with 4 VCPUs and 16GB of memory fits on a single NUMA node with memory to spare.

If one examines the aggregate performance shown in the figure, an interesting behavior is seen. Whether run on two virtual machines ($\approx 80\%$ host CPU utilization) or three ($\approx 100\%$ host CPU utilization), performance was roughly the same. Surprisingly, adding a fourth virtual machine increased aggregate performance. While one might expect the addition of the third virtual machine to have increased performance, the additional host CPU capacity was likely cancelled out by the costs of migration and remote memory access time. To detect such a problem, one can refer to the NUMA Statistics on esxtop's memory screen and the CPU Event Counts (migration rate) on esxtop's CPU screen.

With three 4-way virtual machines and two physical 4-way nodes, vSphere periodically migrated the virtual machines to ensure fairness. This introduced a memory access time penalty: local memory became remote when the virtual machine migrated to the other node. When a fourth virtual machine was added, no internode migration was needed to ensure fairness, and VMware® ESX™ allocated memory on the same node that runs the virtual machine. In this case, it's believed that the benefit of local memory access and lack of migrations in the four virtual machine case improved the aggregate performance – compared to the three-virtual machine case – even in the presence of overcommitted resources.

Local memory is preferred even in undercommitted situations due to its lower access time. Thus, when running on a NUMA system, one should strive to specify a virtual machine memory size that is less than the amount of memory on a NUMA node. Furthermore, choose a VCPU count that is less than or equal to the number of processors per node. This configuration allows vSphere 4 to employ NUMA optimizations for memory and CPU scheduling and ensures that all memory accesses will be satisfied by the memory closest to the processor.

Large Pages

Computers manage memory in groups of bytes called pages. The x86 platform supports several page sizes, the most common being 4KB (small) pages and 2MB (large) pages. When an application accesses a large memory region frequently or accesses a significant amount of memory within the large region, it can be advantageous to work with large pages. This allows circuits in the CPU to access the memory faster. In a virtual environment, hypervisor support is needed to fully realize this performance boost [8].

By default, ESX will use large pages for all memory that it manages when running on a platform with hardware nested page tables. This mitigates the extended TLB miss penalty that is possible on such hardware. Even when the hypervisor has backed a guest physical page with a large host machine page, configuring the guest to use large pages can offer further benefits and is always advised. Properly configuring the guest is essential for large page performance with hardware that does not implement RVI or EPT. For such servers, vSphere defers to the guest operating system for hints as to which pages should be backed large.

To see the value of vSphere's large page support, the virtual machine's operating system and applications were configured as described below prior to running the sales and distribution application load test¹. The number of supported users was compared against an execution using the same virtual machine configuration but with large pages disabled using vSphere advanced configuration settings².

- Windows: Use Group Policy Editor to give SAP process owner rights to "Lock pages in memory"
- MS SQL Server: Automatic if the virtual machine's memory size is ≥ 8 GB
- SAP: Profile should contain `em/largepages=yes`

1. Linux may also be configured for large pages as described in SAP Note 989935

2. `esxcfg-advcfg -s /LPage/LPageAlwaysTryForNPT 0`
`esxcfg-advcfg -s /Mem/AllocGuestLargePage 0`

(applicable in benchmarking configurations only)

As shown in [Figure 6](#), the SAP workload exhibits a marked performance benefit, 12 percent more users, when using vSphere's large page support.

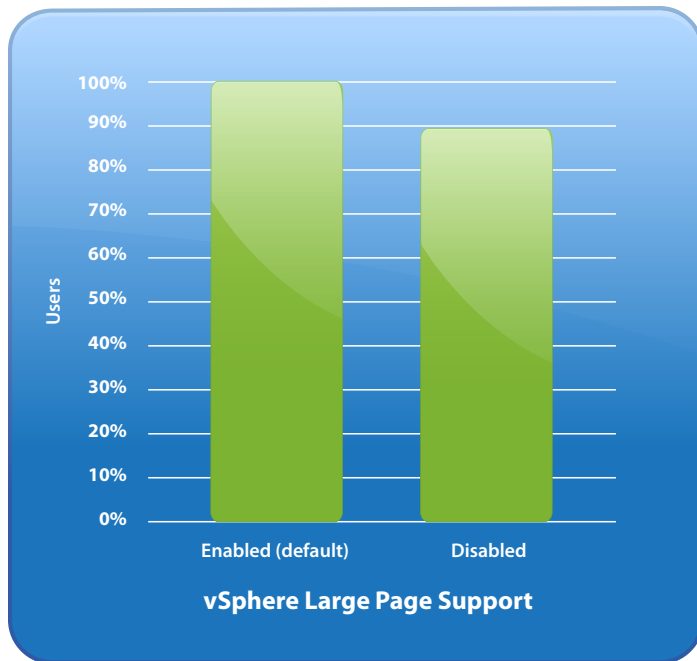


Figure 6: vSphere's support for large pages improves SAP performance by 12%

Recommendations

The results in this paper suggest that to run SAP in a virtual machine most efficiently, one should adopt the following best practices:

- Run with no more VCPUs than necessary.
- Use the newest hardware (e.g., "AMD Opteron 2300/8300 Series" or "Intel Xeon 5500 Series") to exploit vSphere's support of hardware nested page tables.
- Limit virtual machine size to fit within a NUMA node.
- Configure guest operating system and applications for large pages.
- If using a processor with hardware nested page tables (RVI or EPT) and Linux, choose the Std memory model.
- If using a processor with hardware nested page tables (RVI or EPT) and Windows 2008, convenience should dictate the choice of memory model as it has only a minor effect on performance.

Conclusion

The mission critical nature of SAP solution deployments requires a well-performing and scalable compute infrastructure to satisfy the service level agreement requirements of business users. The results in this document show how vSphere can help to meet the performance and scalability requirements of these SAP implementations.

This paper has demonstrated how CPUs that support nested paging in hardware can boost performance of an SAP workload in a vSphere virtual machine. Experiments that compare native and virtual performance have shown the excellent scale-up properties of SAP software on vSphere. The fairness of vSphere's scheduler during a CPU overcommitment scenario has been confirmed, and vSphere's support for large pages has been highlighted. By following the recommendations above, one can expect virtualized SAP environments to exhibit nearly the same performance as environments running on physical systems.

Acknowledgments

Tuyet Pham performed the experiments used to characterize performance on the Windows / RVI platform, and Michael Hesse measured performance of SAP on the Linux / EPT platform. Their significant contribution is greatly appreciated.

VMware thanks the SAP Benchmark team for its quick response whenever full disclosure information was requested.

VMware thanks Intel for lending the hardware used for the EPT and CPU overcommitment experiments.

Thanks to Scott Drummonds, Aravind Pavuluri, Vas Mitra, Rajit Kambo, Priti Mishra, and Michael Hesse for their reviews on early drafts of this paper.

About the Author

Kenneth Barr is a Staff Engineer in VMware's Core Performance group where he has studied virtualized SAP workloads, characterized Virtual Desktop Infrastructure (VDI) performance, and helped tune memory resource management features. At VMworld 2008, he spoke about his work at sessions detailing SAP and VDI performance results and best practices. He also helped present a general overview of ESX Performance Best Practices.

Ken received Masters and Ph.D. degrees from MIT CSAIL's Computer Architecture Group. At MIT, Ken performed research on Energy-Aware Lossless Data Compression and on techniques to accelerate simulation-based microprocessor design. Prior to his work at VMware, Ken's research appeared at various academic conferences including several papers at the International Symposium on Performance Analysis of Systems and Software. Ken received his BSE in Computer Engineering from the University of Michigan.

References

[1] *AMD Virtualization (AMD-V) Technology*.

http://www.amd.com/us-en/0,,3715_15781_15785,00.html.

Retrieved April 14, 2009.

[2] *Performance Evaluation of AMD RVI Hardware Assist*. Technical Report, VMware, 2009.

<http://www.vmware.com/resources/techresources/1079>

[3] *Performance Evaluation of Intel EPT Hardware Assist*. Technical report, VMware, 2009.

<http://www.vmware.com/resources/techresources/10006>

[4] *Flat Memory Model on Windows*. SAP Note Number 1002587.

[5] *Memory protection and performance*. SAP Note Number 807929.

[6] *H. Kuhnemund. Documentation for SLCS v2.3.1*. LinuxLab, SAP AG, Walldorf, May 2008. Release 0.7.

[7] *SAP SD Standard Application Benchmark Results, Two-Tier Internet Configuration*

<http://www.sap.com/solutions/benchmark/sd2tier.epx>. Retrieved May 15, 2009.

[8] *Large Page Performance*. Technical report, VMware, 2008.

<http://www.vmware.com/resources/techresources/1039>.



VMware, Inc. 3401 Hillview Ave Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com
Copyright © 2009 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. VMW_09Q2_WP_VSPHERE_SAP_P13_R1

