VMmark: A Scalable Benchmark for Virtualized Systems

Vikram Makhija (VMware)
vmakhija@vmware.com

Bruce Herndon (VMware)
bherndon@vmware.com

Paula Smith (VMware)
psmith@vmware.com

Lisa Roderick (VMware)
lroderic@vmware.com

Eric Zamost (VMware)
ezamost@vmware.com

Jennifer Anderson (VMware)
jennifer@vmware.com

Technical Report VMware-TR-2006-002

September 25, 2006

VMmark: A Scalable Benchmark for Virtualized Systems

Vikram Makhija, Bruce Herndon, Paula Smith, Lisa Roderick, Eric Zamost and Jennifer Anderson

{vmakhija, bherndon, psmith, lroderic, ezamost, jennifer}@vmware.com

Technical Report VMware-TR-2006-002
September 2006
VMware, Inc.
Palo Alto, CA 94304

Abstract

The increasing speed of computing resources coupled with the rise of robust and flexible virtual machine technology creates opportunities for consolidating multiple, variably loaded systems onto a single physical server. Traditional server benchmarks, which focus on a single workload, do not capture the system behavior induced by multiple virtual machines. An appropriate virtual machine benchmark should employ realistic, diverse workloads running on multiple operating systems. The benchmark should generate an easily understandable metric that scales with underlying system capacity using a controlled strategy based on a combination of increased individual workload scores and running an increasing number of workloads.

This paper presents a tile-based benchmark consisting of several familiar workloads running simultaneously in separate virtual machines. Each workload component is based upon a single-system benchmark running at less than full utilization. This collection of different workloads is aggregated into a unit of work referred to as a *tile*. The performance of each workload is measured and used to form an aggregate score for the tile. The scores generated when running multiple tiles simultaneously may be summed to increase the overall benchmark score.

Key Words and Phrases: Benchmarking, virtual machines, performance.

VMware, Inc., 3145 Porter Drive, Palo Alto, CA 94304 USA
Tel 650-475-5000   Fax 650-475-5001   www.vmware.com

1. INTRODUCTION

Trends in computer hardware have led to the proliferation of powerful yet relatively inexpensive multiprocessor servers. In many cases, applications do not fully utilize these systems. As recent industry developments such as multi-core processors become commonplace, the degree of underutilization should increase. These realities have led to renewed interest in virtual machines for server consolidation. Virtual machine environments provide a software layer that enables users to create multiple independent virtual machines on the same physical server, as shown in Figure 1.

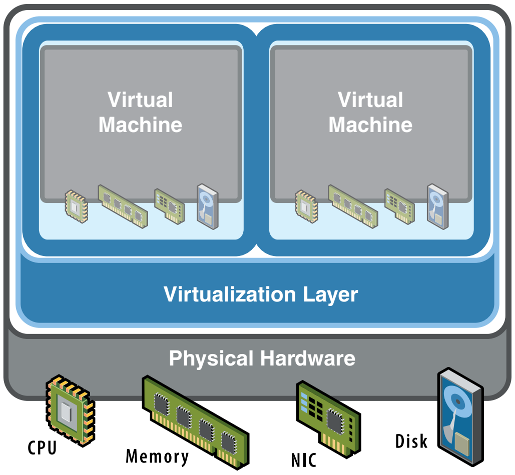

Figure 1 - Example Organization of a Virtual Environment

By running multiple virtual machines simultaneously, a physical server can be driven to much higher utilizations, albeit with some virtualization overhead. Although the underlying physical resources are shared, each virtual machine is fully isolated from other virtual machines and executes a separate operating system instance and separate application software.

The use of virtual machines dates back more than thirty years, with systems primarily used for functions such as software test and development, reliability, and security [1]. A notable example was the IBM VM/370 virtual machine system [2]. More recent work, such as Disco [3], has applied the concept of virtual machines to commodity operating systems running on modern shared-memory multiprocessors. Several vendors including VMware, Microsoft, and XenSource currently provide virtual machine software environments for running a wide range of commodity operating systems on x86 [4] hardware. These systems provide popular platforms for consolidation of server workloads.

It is crucial for users to have meaningful and precise metrics in order to effectively compare the suitability and performance of different hardware platforms for virtual environments. Traditional performance and scalability benchmarks, such as those available through SPEC [5] or TPC [6], were developed with neither virtual machines nor server consolidation in mind.

These metrics focus on achieving maximum system performance for a single workload by driving at least one of the underlying hardware resources into saturation.

These types of benchmarks do not provide sufficient insight into the scalability of virtual environments supporting multiple simultaneous workloads.

Nevertheless, single-workload benchmarks have been useful for quantifying virtualization overheads within a single virtual machine [7][8] and can be used for tuning portions of the virtualization layer. The scalability of virtual environments has been measured by running multiple virtual machines, each executing the same benchmark workload [8][9]. Although simple scaling tests do measure some aspects of system performance, they do not stress the physical resources and virtualization layer sufficiently to fully capture the complexity of running multiple different workloads within a server consolidation framework. IBM has recently published a virtualization benchmark based upon mixed workloads running in virtual machines [10]. In this study, the number of virtual machines remains fixed while the individual workloads are increased to measure performance. This benchmark lacks the notion of increasing the number of active virtual machines to match the underlying hardware resources as is common in a consolidation context.

Clearly, a more sophisticated approach is required to quantify a virtualization environment's performance and ability to run an increasing number of diverse virtual machines as physical resources increase. First, all relevant hardware subsystems should be exercised as they would in an actual datacenter. The individual virtual machines should also operate at less than full utilization to mimic consolidation within a datacenter environment. The benchmark must scale in a controlled fashion to make comparisons between systems meaningful. Small fluctuations in the performance of individual virtual machines can be used to discern minor differences between similar systems. Larger gaps in performance can be measured by increasing the number of active virtual machines. The benchmark must also exhibit stable, reproducible performance.

This paper presents a benchmark, VMmark, to address these goals. The paper is structured as follows:

- “Workload Tiling” on page 2 introduces the concept of a multi-workload tile that encapsulates several diverse workloads.
- “Workloads” on page 3 describes the individual workloads.
- The scoring algorithm is presented in “Scoring Methodology” on page 6.
- Scores from a 2-CPU server are presented in “Experimental Results” on page 7.
- Finally, conclusions and future work are presented in “Conclusion and Future Work” on page 8.

2. WORKLOAD TILING

The ultimate goal of VMmark is to create a meaningful measurement of virtualization performance across a wide range of hardware platforms. Server consolidation typically collects several diverse workloads onto a single physical server. This approach ensures that all system resources such as CPU, network, and disk are more efficiently utilized. In fact, virtual environments tend to function more smoothly when demands are balanced across physical resources.

The unit of work for a benchmark of virtualized consolidation environments can be naturally defined as a collection of virtual machines executing a set of diverse workloads. The VMmark benchmark refers to this unit of work as a tile. The total number of tiles that a physical system and virtualization layer can accommodate gives a coarsegrain measure of that system's consolidation capacity. This concept is similar to some server benchmarks, such as TPC-C, that scale the workload in a step-wise fashion to increase the system load.

Tiles are relatively heavyweight objects that cannot by themselves capture small variations in system performance. To address this, the overall VMmark benchmark score is determined by both the number of tiles and the performance of each individual workload as described in Scoring Methodology on page 7.

Each workload within a VMmark tile is constrained to execute at less than full utilization of its virtual machine. However, the performance of each workload can vary to a degree with the speed and capabilities of the underlying system. For instance, disk-centric workloads might respond to the addition of a fast disk array with a more favorable score. These variations can capture system improvements that do not warrant the addition of another tile. However, the workload throttling will force the use of additional tiles for large jumps in system performance. When the number of tiles is increased, workloads in existing tiles might measure lower performance. However, if the system has not been overcommitted, the aggregate score, including the new tile, should increase. The result is a flexible benchmark metric that provides a relative measure of the number of workloads that can be supported by a particular system as well as the overall performance level within the virtual machines.

3. WORKLOADS

A meaningful tiled consolidation benchmark should be based upon a set of relevant datacenter workloads. A survey of datacenter applications led to the inclusion of the following workloads:

- Mail server
- Java server
- Standby server
- Web server
- Database server
- File server

Rather than develop workloads from scratch, existing benchmarks were used wherever possible. This reduces implementation effort and provides a well-understood foundation upon which to build. However, the run rules of the various benchmarks occasionally conflict with the design goals of VMmark. This required some modifications to the benchmarks to make them suitable for multi-VM benchmarking. The following sections discuss each workload and any necessary modifications.

3.1 Mail Server

Most businesses today provide employees with email as a means for communicating. For this reason, mail servers are important workloads in modern data centers, not just for their often-large resource requirements, but for their strict response-time requirements. Perhaps the most common mail server is Microsoft Exchange. We therefore chose Microsoft Exchange 2003 to represent the mail server workload in VMmark.

Microsoft provides a well-known load generation utility called LoadSim that simulates users of the Exchange mail server. For consistency with VMmark design a few changes were made in this utility’s implementation. In its default configuration, LoadSim requires a large amount of initial static disk space. This was reduced to make the workload more manageable.

Also, in contrast to recommended LoadSim methodology of increasing the load on the Exchange mail server until some resource is exhausted, the load is fixed by limiting the configuration to 1000 MMB3 users. This is a typical load for a medium-sized business, and it does not cause the Exchange virtual machine to be bottlenecked on any resource. VMmark design also requires a periodic rate metric for each workload. We can not use the number of users as a metric, as is commonly done, because we have fixed that number. We therefore periodically measure the number of transactions executed by the mail server and use that as our metric.

The Microsoft Exchange 2003 workload was run under Microsoft Windows Server 2003, Enterprise Edition on a virtual machine with two virtual CPUs and 2GB of memory.

3.2 Java Server

Java performance is crucial in many modern multi-tiered applications. A modified version of the popular SPECjbb2005 [15] benchmark was included in the VMmark Benchmark as a measure of Java workload performance. SPECjbb2005 is a transactional workload based upon the TPC-C database benchmark. However, SPECjbb2005 is designed to be entirely standalone and requires no external client to generate system load. The database layer, middle tier, and clients are all simulated within the same Java virtual machine (JVM) executing on the server.

Although the benchmark collects transaction rates and response times throughout each benchmark test, it is designed to report results only at the completion of a test. The benchmark reporting was modified to allow periodic performance snapshots to be collected. As designed, SPECjbb2005 does multiple short runs over an increasing database size. To generate steady load and simulate a long-running application, the database size was set to the maximum (eight warehouses) and a single long run was performed.

Finally, the resource consumption of SPECjbb2005 was throttled by introducing a non-standard think time between transactions. Finer-grained think times than originally existed were added to the benchmark code to allow more precise control of the workload.

The SPECjbb2005 workload was run under Microsoft Windows Server 2003, Enterprise Edition on a virtual machine with two virtual CPUs and 2GB of memory. The JVM used was BEA JRockit 5.0.

3.3 Standby Server

Many computing environments contain standby servers ready for new workloads, or for workloads with bursty behavior. These extremely lightly loaded (essentially idle) systems are attractive targets when consolidating servers. Even when idle, however, these systems still place resource demands upon the virtualization layer and can impact the performance of other virtual machines. For this reason, one standby server virtual machine was included in the benchmark tile.

The standby server does not produce a metric that affects the benchmark score. However, it is required to answer a periodic heartbeat for the VMmark test to be considered valid.

The standby server workload ran Windows Server 2003, Enterprise Edition on a virtual machine with one virtual CPU and 256MB of memory.

3.4 Web Server

Web servers are pervasive in modern data centers and are often strong candidates for consolidation. In particular, the Apache web server (see http://httpd.apache.org/) running under the Linux operating system is an extremely common solution.

While many web server benchmarks exist and have achieved various degrees of success, the SPEC organization has produced some of the most widely accepted web server benchmarks. A modified version of the most recent of these, SPECweb2005 [12], was included as a workload in the VMmark Benchmark. SPECweb2005 supports multiple workload profiles. The E-commerce profile was used for these tests.

A single long iteration of the benchmark was run, rather than the three specified in the run rules, to avoid multiple ramp-up and ramp-down periods within the VMmark benchmark window. The client think time was reduced from ten seconds to two seconds to generate the desired load with fewer concurrent sessions, thereby reducing demands on the client system. The web server access logs were turned off.

The internal polling feature of the SPECweb client was used to determine the number of page accesses, and 100 sessions were used in all tests. Otherwise, the default configuration was used. The benchmark score can fluctuate based upon the response times of the web server.

SPECweb2005 was run under Red Hat Enterprise Linux 4, Update 1 on a virtual machine with two virtual CPUs and 512MB of memory running Apache 2.0.54 and PHP 4.4.0. SPECweb2005's backend simulator (BeSim) service was also run under Apache 2.0.54 using FastCGI 2.4.2. Although it is recommended that BeSim be run on a system separate from the web server, VMmark runs both on the same virtual machine to simplify administration and keep the overall workload tile size down.

3.5 Database Server

Databases running transactional workloads support a wide array of applications. Although very large databases requiring dedicated systems do exist, a significant fraction of databases are smaller and can easily run in a virtual environment. Regardless of size, databases tend to be resource intensive and exercise most system components, especially CPU and storage. In many cases, database systems also face strict response-time demands.

Swingbench [11], Oracle's freely-available online transaction processing application, was used as the database workload. Swingbench models users repeatedly executing a pre-defined mix of transactions against a database. For VMmark, 100 users multiplexed over 20 pooled connections were run. The user think time between transactions was defined to be 3 seconds.

Oracle 10g (10.1.0.3) using the Oracle-supplied JDBC driver for OCI was used as the underlying database. The database instance was approximately 8GB. Application data files consumed roughly 5GB and redo log files accounted for the remaining space. Oracle's SGA (in-memory data cache) was configured to be 516MB.

The performance metric for this workload is the number of database commits per second. This information was collected at regular intervals during benchmark runs by a remote application that queried the database directly. Despite the fixed number of database users and the fixed think time, commits per second can vary up to a fixed maximum based upon the response time of the Oracle database.

The Oracle workload was run in Red Hat Enterprise Linux 4, Update 1 on a virtual machine configured with two virtual CPUs and 2GB of memory.

3.6 File Server

The ability of a file server to service requests from Windows 95 clients is measured by the dbench application. The dbench application [14] is derived from the industry-standard NetBench benchmark, which requires a large number of client systems to generate the system load. The load for dbench is instead comprised of access pattern traces in NetBench. Dbench was chosen for its ease of setup and management.

A small application that allocates and mlocks a large block of memory was run concurrently with dbench to reduce the size of the system buffer cache and force more operations to access the physical disk. Without this additional memory utilization, dbench becomes a CPU-intensive workload where most disk reads and writes hit in the buffer cache.

Dbench has a relatively short running time, so the benchmark was modified to run repeatedly for the duration of each VMmark run. The dbench benchmark was further modified to connect to an external program via TCP/IP. This external program, which runs on a client system, keeps track of the benchmark's progress for scoring purposes and provides a pacing mechanism to control dbench’s resource utilization, thus ensuring that each tile provides a predictable load until the physical hardware becomes fully saturated. A throughput average was computed at the conclusion of each benchmark run. The throughput can vary based upon the speed of the underlying disk subsystem.

The dbench workload was run in Red Hat Enterprise Linux 4, Update 1 on a virtual machine configured with one virtual CPUs and 256MB of memory.

4. SCORING METHODOLOGY

Once a VMmark test completes, each individual workload reports its relevant performance metric. The performance metrics collected are shown in Table 1. These metrics are collected at frequent intervals during the course of a run. A typical VMmark benchmark test is designed to run for at least three hours with workload metrics reported every 60 seconds. Once all workloads have reached steady state during a benchmark run, a two-hour measurement interval is taken. This steady-state interval is then divided into three 40-minute sections. For each of the 40-minute sections, the results for the tile are computed. The median score of the three sections is selected as the raw score for the tile. For multi-tile runs, the median of the sums of the per-tile scores would be used as the raw score.

After a benchmark run, the workload metrics for each tile are computed and aggregated into a score for that tile. This aggregation is performed by first normalizing the different performance metrics such as MB/s and database commits/s with respect to a reference system. Then, a geometric mean of the normalized scores is computed as the final score for the tile. The resulting per-tile scores are then summed to create the final metric.

For VMmark, normalization allows the integration of the different component metrics into an overall score. The use of a reference system for normalization is commonly employed when computing benchmark scores. For instance, SPEC CPU2000 [16] takes this approach and uses a Sun Ultra5_10 with a 300MHz processor as its reference platform. At present, the formal choice of reference system for VMmark is still under investigation with data from a wide range of systems being gathered.

Table 1 - Individual VMmark Workload Metrics.

Workload	Metric
Mail server	Actions/minute
Java server	New orders/second
Standby server	None
Web server	Accesses/second
Database server	Commits/second
File server	MB/second

5. EXPERIMENTAL RESULTS

VMmark was designed to measure performance across a wide range of systems, beginning with 2-CPU servers. Initial VMmark results were generated to verify that a fully-configured tile can be run successfully and maintain steady state for the length of a measurement interval on a small hardware platform. The measurement of larger systems will be presented in future work.

5.1 Experimental Setup

An HP ProLiant DL580 containing two 2.2 GHz Intel Xeon CPUs with hyper-threading support and running VMware's ESX Server 3.0 [17] was used as the demonstration platform. The system was configured with 16GB of memory. An EMC Clariion disk array with 5 disks configured in RAID5 connected via a fiber channel link provided storage. The system was connected to the load-generating client system over a single 100 Mbit network link. The load-generating client was an HP ProLiant DL385 with 4GB of memory running Microsoft Windows 2003.

5.2 Experimental Results

Multiple four-hour runs were performed. The results were quite consistent run-to-run. The system achieves full CPU utilization for the entire duration of the benchmark run, including the warmup period. The results of one representative run are presented below to demonstrate the stability of the benchmark during the scoring window and will be used to demonstrate the scoring methodology. Table 2 summarizes the raw scores for three 40-minute intervals within a two-hour window between minutes 110 and 230 in the benchmark. In addition, a 40-minute measurement was taken during the warmup interval between minutes 30 and 70. This measurement data is shown in lieu of a formal reference score and will be used to demonstrate the scoring algorithm.

Table 2 - Experimental Results (Raw Scores)

Workload	**Warmup Interval**	**Interval 1**	**Interval 2**	**Interval 3**
Mail server	660	891	917	935
Java server	13174	13285	13249	13310
Standby server	-	-	-	-
Web server	851	835	836	833
Database server	897	940	936	951
File server	7.23	6.93	6.84	6.79

Table 3 shows the performance of each workload normalized against the warmup measurement interval. Both the web server and file server workloads have slightly higher performance during the warmup period. These two workloads warm up relatively quickly and can benefit from the excess CPU cycles available until the other workloads reach steady state. Once the database server and mail server workloads fill their buffer caches and can consume additional CPU, that excess CPU capacity disappears and all workloads execute at their stead-state performance levels on this system. Table 3 also shows the geometric mean of the normalized scores for each benchmark interval. In this case, Interval 3 is the median value and would be reported as the benchmark result. If the system were not already fully utilized, an additional tile could be added to improve the overall benchmark score.

Table 3 - Experimental Results (Normalized Scores)

Workload	**Interval 1 ratio**	**Interval 2 ratio**	**Interval 3 ratio**
Mail server	1.35	1.39	1.42
Java server	1.01	1.01	1.01
Standby server	-	-	-
Web server	.98	.98	.98
Database server	1.05	1.04	1.06
File server	.96	.95	.94
Geometric Mean	1.08	1.06	1.07

6. Conclusion and Future Work

This paper presented VMmark, a novel benchmark for quantifying the performance of virtualized environments. As server virtualization becomes increasingly commonplace, it is important to have sound and representative benchmarks to evaluate the performance of hardware and software platforms for virtualization. VMmark is designed as a tile-based benchmark consisting of a diverse set of workloads commonly found in the datacenter, including database server, web server, and Java server. The workloads comprising each tile are run simultaneously in separate virtual machines at load levels that are typical of virtualized environments. The performance of each workload is measured and then combined with the other workloads to form the score for the individual tile. Multiple tiles can be run simultaneously to increase the overall score. This approach allows smaller increases in system performance to be reflected by increased scores in a single tile and larger gains in system capacity to be captured to adding additional tiles.

Future work will present data to demonstrate the ability of multiple tiles to measure performance of larger multiprocessor systems using a well-defined reference score.

7. REFERENCES

[1] R. P. Goldberg. Survey of Virtual Machine Research. *IEEE Computer*, 7(6) (June 1974), 34-45.

[2] R. J. Creasy. The Origin of the VM/370 Time-Sharing System. *IBM Journal of Research and Development,* 25(5), September 1981.

[3] E. Bugnion, S. Devine, K. Govil, and M. Rosenblum. Disco: Running Commodity Operating Systems on Scalable Multiprocessors. *ACM Transactions on Computer Systems*, 15(4), November 1997.

[4] Intel Corporation. *IA-32 Intel Architecture Software Developer's Manual. Volumes I, II, and III*, 2001.

[5] Standard Performance Evaluation Corporation (SPEC). *http://www.spec.org/*.

[6] Transaction Processing Performance Council (TPC). *http://www.tpc.org/*.

[7] I. Ahmad, J. Anderson, A. Holler, R. Kambo, and V. Makhija. An Analysis of Disk Performance in VMware ESX Server Virtual Machines. *Proceedings of the Sixth Workshop on Workload Characterization (WWC '03)*, October 2003.

[8] P. Barnum, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Nuegebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, January 2003.

[9] A. Whitaker, M. Shaw, and S. Gribble. Scale and Performance in the Denali Isolation Kernel. *Proceedings of the Fifth Symposium on Operating System Design and Implementation (OSDI '02)*, December 2002.

[10] IBM Corporation. Virtualization Grand Slam Benchmark. Available from *http://www-03.ibm.com/servers/eserver/iseries/hardware/virtualizationgrandslam/*.

[11] Oracle Corporation. *Swingbench 2.2 Reference and User's Guide*, August 2005.

[12] Standard Performance Evaluation Corporation (SPEC). *http://www.spec.org/web2005/*.

[13] The Apache Software Foundation*. http://www.apache.org/*.

[14] A. Trigdell. dbench benchmark. Available from *ftp://samba.org/pub/tridge/dbench/*.

[15] Standard Performance Evaluation Corporation (SPEC). *http://www.spec.org/jbb2005/*.

[16] Standard Performance Evaluation Corporation (SPEC). *http://www.spec.org/cpu2000/*.

[17] VMware, Inc. *VMware ESX Server User's Manual*, Palo Alto, CA, August 2006.