Volume 10 Issue 03 Published, August 10, 2006 ISSN 1535-864X DOI: 10.1535/itj.1003

Intel® Technology Journal

Intel® Virtualization Technology

Redefining Server Performance Characterization for Virtualization Benchmarking

More information, including current and past issues of Intel Technology Journal, can be found at:
http://developer.intel.com/technology/itj/index.htm

Redefining Server Performance Characterization for Virtualization Benchmarking

Jeffrey P. Casazza, Digital Enterprise Group, Intel Corporation
Michael Greenfield, Software and Solutions Group, Intel Corporation
Kan Shi, Digital Enterprise Group, Intel Corporation

Index words: virtualization, benchmark, server consolidation

ABSTRACT

Virtualization will dramatically change enterprise system deployments and is being driven by innovation across a broad set of platform technologies. All of this attention has created a broad interest in comparing different products. One challenge is that there are no established performance methodologies to measure virtualization performance. Today's existing server benchmarks cannot be easily used as-is by an end user to generate clear and relevant results. This paper presents a workload methodology to help the reader characterize the performance of servers exploiting virtualization technologies to consolidate multiple physical servers. It presents an example benchmark using applications often found in typical virtualization deployments. The existence and exploitation of a standard methodology is a key to accelerating continued improvement of virtualization technologies.

INTRODUCTION

Virtualization has been part of the datacenter since the 1960s when it was exploited across mainframe systems [1]. Virtualization is experiencing a renaissance as this technology is finding its way to high-volume servers. To the IT technologist, virtualization brings the promise of solving several datacenter problems. Virtualization can reduce costs by consolidating older servers. It helps organizations become more nimble through fast provisioning of virtual servers. It improves equipment utilization and the end-user experience by enabling dynamic load balancing and improved disaster recovery capabilities. These benefits provide a strong motivation for accelerating server virtualization deployments.

Virtualization provides an abstraction of hardware resources enabling multiple instantiations of operating systems (OSs) to run simultaneously on a single physical platform. The abstraction provides isolation between the separate running partitions to prevent individual faults from affecting the entire system. The virtualization of the hardware also means that different OSs can be supported on a single platform simultaneously—even older OSs. Consolidating several physical servers that have workloads with non-overlapping peak utilization requirements over time allows better hardware utilization than if these were carried out on separate systems. These benefits are attractive in environments with legacy servers that, though important to the business, cannot justify the porting and validation activity to a newer OS [2].

In this paper we present a methodology and an example for characterizing the performance of servers using virtualization to consolidate multiple physical servers. We provide a general overview of two key virtualization usage models. We also briefly look at how contemporary methods can be applied to virtualization. We discuss the challenges generated by the virtualization abstraction layer and consolidation, and we present a systematic approach to performance measurement. Finally. an example workload, called vConsolidate, is defined to further clarify the methodology.

ENTERPRISE VIRTUALIZATION USAGE MODELS

The value virtualization brings to datacenters depends on what problems it can solve for the IT technologist. These are defined to be virtualization's usage models. Virtualization usage models today are focused on legacy consolidation, flexible provisioning, test/development, dynamic load balancing, and disaster recovery. As the technology penetrates the mainstream datacenter, new usage models will likely emerge.

The performance discipline [1] presented in this paper provides a basic framework fitting multiple usage models. However, we focus specifically on two usage models due to their historic appeal to datacenter managers: legacy consolidation and flexible provisioning

Legacy consolidation refers to transferring application and OS stacks from multiple servers to a fewer number of (typically more powerful) servers. The old "legacy" servers are often running older OSs because upgrading to a newer version has some manpower impacts. For example, a legacy application may need some porting work, but the resources to do this cannot be justified. In this paper an old (legacy) server is classified to be 3+ years old. The utilization level of these legacy systems is often quite low, 5-15%. Such low utilization of an older platform means that a contemporary server could provide the processing capability for running many multiples of the legacy applications simultaneously. Market research indicates that legacy consolidation projects with virtualization tend to mix different application types on the new consolidated server; however, IT managers are avoiding mixing OS types [3]. The final consolidated server typically has some mixture of Web servers, e-mail servers, database servers, and/or other types of applications running simultaneously, but all are running the same type of OS (e.g., Linux* or Windows*) and often the same version of the OS [3].

Flexible provisioning is essentially forward-looking consolidation. As new servers are requested from the IT department, rather than deploying a new physical server for each request, the deployment is a virtual slice of an existing server. For example, if a two-processor server is needed to support a new Web server application, perhaps two virtual processors of a multi-processor server are deployed in support of the request. This drives up the utilization of the servers, thereby improving efficiency. The virtual provisioning is often much faster, as hardware may already exist to support the request, eliminating the procurement cycle.

These two usage models, legacy consolidation and flexible provisioning, constitute the main driver behind the adoption of virtualization in mainstream datacenters. The focus of this methodology is to measure the performance of virtual servers in a way that is related to these two usage models. Other usage models, like test and development, are also popular but not addressed in this paper.

Having the usage model defined is important for developing a relevant performance benchmark. The usage model provides boundary conditions to the testing. For example, the usage model indicates what types of applications should be used, whether the application and OS stacks are heterogeneous or homogeneous, what is the configuration of a typical system-under-test (SUT), and what is the appropriate size and number of the virtual machines (VM).

VIRTUALIZATION PERFORMANCE CHARACTERIZATION CHALLENGES

A question that may come to mind is "Why can't existing performance characterization methods be exploited?" Users already have many accepted methods and tools to characterize servers. Some of these include load generators (e.g., LoadRunner*) and a myriad of industry standard (e.g., SPEC*, TPC*) and proprietary workloads (e.g., SAP-SD 2 Tier*, MMB3*, R6iNotes*). There are several challenges presented in virtualization performance characterization including consolidation, virtualization, and implementation considerations. These limit the use of existing methods.

Consolidation Characterization Challenges

We need to differentiate between consolidation and virtualization challenges, as both introduce complexity into performance measurement and tuning. Virtualization facilitates creating multiple VMs on one physical machine. Consolidation relates to running multiple workloads on the system at the same time.

A challenge with consolidation characterization is the mixture of different workloads. If you consolidate a set of heterogeneous workload environments, consider that each will have a different set of requirements and metrics and that depending upon the users' specific requirements, the relative priority of each will vary across users, time, and other dimensions.

Another consolidation challenge relates to resource profiles. The non-steady state resource profile of the individual servers will look quite different from that of the consolidated system [4]. It is simplest to measure performance when all measurements are conducted in a time window after all workloads are in a steady state. While this may be nice for a benchmark, it fails to represent many real-world usage models. Consider the following examples:

- Most e-mail servers have distinct periods where the demands upon them vary a great deal. For example, the system may be idle until a wave of people arrive at work and log in, download their e-mail, and make other demands on the server. Conversely, the

---

[1] A structured and consistent set of methods and processes that are accurate and repeatable.

demands on the server would decrease as people finish up for the day.

- A Web store that supported a worldwide customer base could be busy 24x7 and reach a steady state as opposed to some service that was provided to people in one locale.
- Some workloads have seasonal variations, end-of-month closings, holiday duty cycles, and other modifications that may differ greatly from normal operations.

Consider two examples of a consolidated e-mail server: Web store server, and a customer relationship management (CRM) server. In the first scenario, Figure 1, we see that none of these is ever run in a steady state. If these are the actual profiles of the consolidated server, it would be prudent to examine peak resource requirements when superimposed at one instant of time to determine how well the overall system is performing.

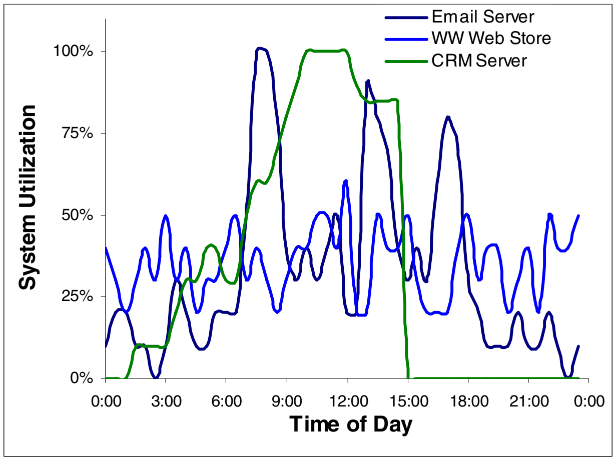

Figure 1: System resource profile for workloads that are not operating in a steady state

In the second example, Figure 2, we see three server utilization profiles that all reach a steady state. If we were to examine the performance of the consolidated system and did our study at some point after 15 hours of running, we would see a much simpler profile of the workloads in a steady state. While the second workload is easier to test and tune, it may not reflect the actual end-user resource profile.

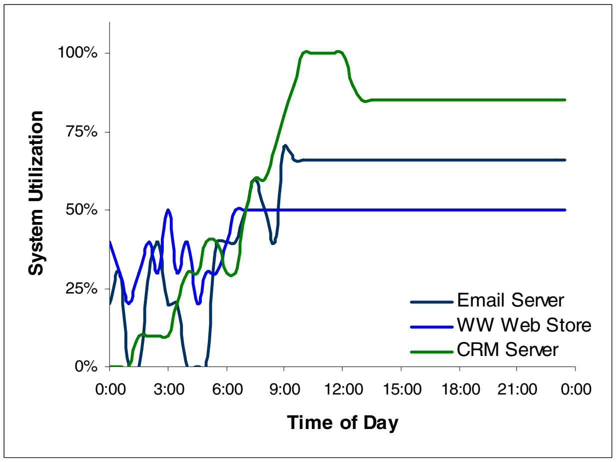

Figure 2: System profile for workloads that are operating in a steady state

Virtualization Characterization Challenges

Whatever performance tools, methods, and processes are used for the characterization, tuning and simulation of server-based workloads they are likely to continue to be relevant in a virtualized environment. As much as we would like to have a single benchmark (or a small set of benchmarks) to describe server performance, there is nothing as good as the actual end-user workload (what they do today and how that will change over time) to employ in developing a performance and projection discipline. This will also be true for virtualization performance, since no single workload will characterize all user requirements. Consider some different user requirements which may include the following:

- A threshold minimum throughput must be maintained over time.
- Some margin must be available for peak workload requirements or for future expansion.
- The server provides some service and the response time to any specific request or set of requests cannot exceed some specified quality-of-service threshold.

We can better understand some of the new challenges that are introduced in the context of virtualization when we consider the requirements for and how a system will be used. For our example, server environments are associated with the consolidation of existing (often legacy) systems and the virtual partitioning of an existing platform for new server deployments. The diversity of what is being consolidated requires that no one workload or environment can be used as a general proxy for (most) others. Consider the diversity of essential components (and how poorly one workload would serve as a proxy for another):

- One OS as a proxy for all others (e.g., Windows* to represent Linux*)
- One usage model or vertical to represent another (Linpack to represent MMB3)

Listed below are some of the other new challenges virtualization adds:

- There are many different options on how a platform will be partitioned and how resources can be allocated, each dramatically affecting the performance of each and how they interact with each other. These are further compounded depending upon what the goals are: for example, absolute performance, minimum performance thresholds, power consumption, TCO, or other optimization criteria.
- There are different strategies that can be used to evaluate a system, including response time, throughput, percentage utilization, and others. These may be exploited simultaneously across separate workloads running across different VMs in a single performance discipline.

Implementation Challenges

As different software stacks are combined inside a set of VMs, there are considerations that may affect precision and repeatability of the results. Often these are tied to the specific implementation of the virtualization abstraction layer and underlying platform. Though by no means an exhaustive list, such issues could include the following:

- *VM clock accuracy/precision*: Since there are several VMs running on a single platform, there is a variety of approaches to how the virtual clock is mapped to the physical platforms' clock, and any of these can cause clock skew. Since most benchmarks will compute a performance metric based upon the always assumed correct system clock, any changes in the clock behavior could lead to errors in computing the delivered performance. Such issues, as well as ways to minimize this possibility, are further explored in VMware [5].
- While an extensive set of system performance monitors are available under most native operating systems (OSs), most virtualization monitors provide only the most basic performance monitoring capabilities. This is sure to improve over time, but the combination of the environment getting more complicated from both consolidation and virtualization and the nascent state of performance monitoring conspire to increase the difficulty to comprehend and productively tune the system.
- All virtualization implementations introduce an additional level of abstraction and not unexpectedly, additional overhead. This makes appropriate system configuration even more important than it is for unvirtualized environments, since resource limitations usually drive up the context switching rates, perhaps at multiple levels of abstraction. Being more generous with memory and I/O capacity when setting up the system initial configuration in a virtualized environment can offer an even larger return in performance and price/performance than non-virtualized environments. As a simple example, a reduction in page fault activity after adding some RAM in a virtualized environment is likely to pay an even larger dividend than in the pre-virtualized environment.
- Many unvirtualized server benchmarks will have a range of observed performance. When multiple workloads are consolidated on a platform and hosted in VMs, this likely adds more variation, particularly if any of the constituent workloads can impact each other or are tested before they are running in a steady state. Readers are encouraged to run their experiments as many times as is necessary to understand the performance profile and variation from run to run.
- Obtaining consistent and predictable performance results assumes that scheduling across VMs is equitable and consistent. It is possible in a virtualization benchmark that the scheduler is not providing what appears to be an equitable distribution of compute and I/O resources across the VMs. For example, if you had N identical copies of a particular workload with the same virtualization monitor configurations, you would expect each to get 1/Nth of the resources available on the system. It is suggested that performance analysts inspect the system during benchmarking to ensure that expected resource profiles are observed.
- Some virtualization monitors will give you various options to map physical CPUs to virtual CPUs and to create affinity between certain sets or to allow a more general pool of resources to be shared amongst all VMs. Virtualization monitors may also permit the setting of weights or CPU percentage to each workload. The higher the workload's weight, the more it will be scheduled to use CPU resources. How to set these depends upon user requirements. For example, is it desirable to ensure that some CPUs are dedicated to certain workloads, or do you want the flexibility for the VM to allocate CPUs based upon dynamic workload changes in real-time? Is one of the

workloads more important than others and therefore should a bigger weight be assigned to it?

When consolidating multiple workloads on a single physical platform, a number of physical devices need to be shared between VMs. Some platforms and virtualization monitors provide different options on how to map the physical devices to virtualized devices. Some physical devices can be assigned solely to a specific workload or just shared between a set of VMs. It depends on customer requirements to set the options. For example, customers can decide to assign a NIC to a Web-bound workload exclusively, and all other more compute bound workloads will share another NIC.

VIRTUALIZATION PERFORMANCE DISCIPLINE

The following is a performance discipline to develop a consolidated virtualization benchmark. The steps are serial in nature and provide a framework to help ensure a reasonable result.

Select workloads, performance metrics, thresholds, and weight factors.

We start with a complete definition of what work needs to be serviced, how we will measure how well the system is meeting the requirement, and if there are any minimum requirements. Weight factors are developed to ensure that resources are allocated in a manner that favors the most demanding workloads.

Sizing the platform and selecting the initial configuration.

Once the requirements are comprehended, initial selection of what platform, configuration, and virtualization technologies (hardware and/or software virtualization, choice of VM implementation, etc.) will be made and how the system will be set up will be decided. The initial platform capabilities can be approached in (at least) two ways. One approach is to start with a platform with clearly more resources (CPU, RAM, network, disk) in every dimension and iteratively reduce them until the right balance is achieved. Alternatively, you can start with a configuration that is too low and add resources until the desired balance is achieved. The former is much quicker and the latter is less expensive.

Define aggregation strategy (how the performance of the constituent workloads will map to an overall performance metric).

This step allows the user to assign and refine the priorities of the different workloads and results that the system-under-test (SUT) will deliver. By assigning weights that are most representative of the user-specific requirements, user perceived improvements can be achieved as opposed to random shifts of resources and workload impact that cannot be comprehended and exploited. It is also during this step that you must consider the repeatability for each workload and whether you will observe and measure it in a transient or a steady state and compensate for interactions between the constituent workloads.

Develop and iteratively refine the mapping of workloads-to-VMs and VMs-to-platform resources.

There are many resource decisions that will dramatically affect how well the system responds by workload. A baseline mapping of these resource decisions would include how to map workloads to VMs and how to relate VMs to system resources (cores, I/O, etc.). Some adjustments to the workloads may be motivated by some of the resource decisions made here.

Tune each workload as if it was not virtualized and establish a baseline, then iteratively refine it (optional step).

It is assumed that each workload is well understood in terms of having a performance discipline and that all tuning steps are fully exploited. This almost always is the best starting point for characterization and optimization of the consolidated virtualized environment. Where the two workloads that are being consolidated have mutually exclusive configuration requirements, it may be best to use either some blending of both or use the one that affects the most heavily weighted workload.

Measure the performance and aggregate it into a performance result.

A comparison to defined thresholds to see if the overall result is valid is conducted, and then the performance of each workload is assessed and aggregated into a single metric.

Identify bottlenecks and attempt to shift resources, augment configuration, and optimize components as needed to drive continuous improvement.

This step focuses on identifying bottlenecks, making configuration adjustments, and making other modifications that help drive improvements in the aggregated performance result for the entire system. It should be noted that there may be limitations in the performance monitoring tools available in a virtualized environment.

Iterate as needed.

This better reflects the tuning and optimization that are usually part of any measurement process, and the focus here is influenced by the difference between the requirement and measured result and the resources available.

VCONSOLIDATE EXAMPLE

While there is no one proxy benchmark for all virtualization deployments, it is highly desirable to have one since it provides a basis of comparison and a basis for a consistent approach to measure and continuously improve. We will arbitrarily define a benchmark called vConsolidate. There are as many possible workloads as there are servers. No two users (or usages) are exactly alike, so we need to make reasonable representative approximations. The purpose of this section is to illustrate how to build up a performance discipline in a virtualized environment with one viable example; and to show the reader how they can do this with component choices that are aligned with their respective environment. The reader should look at this as a basis that can be progressively refined.

In Figure 3, we present a high-level view of the system running vConsolidate. At the base, we have the physical platform, then a virtualization monitor, and some multiple number of VMs, each running a designated workload. We have also defined an aggregation strategy that helps us consolidate workloads, define the performance metric(s) and tell how measurements will be taken.

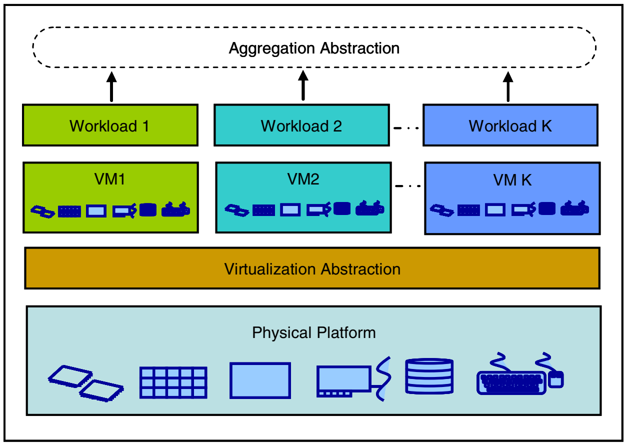

Figure 3: vConsolidate concept

We will construct a hypothetical example to illustrate the performance discipline discussed above. In this example, we select the following environments for consolidation: Web servers, e-mail servers, and database servers. For this example, we model the system by having some multiple of each running and designate the weights of each as Weight[i]. All the weights are fixed numbers pre-defined by the workload. Each workload can be run on the system before and after any virtualization layer is included, and we can compare the virtualized performance of each workload with a baseline measured without virtualization on a pre-defined standard machine. This serves as a useful tool for calibrating subsequent results. We replicate these workloads as needed based upon a set of usage model requirements. Then, we calculate the ratio of virtualized/baseline as the relative performance of each workload. The performance of the virtualized environment would be

N

Weight[i]*WorkloadPerf[i]

I=1

where WorkloadPerf[i] is the relative performance of the i'th workload

Here is an example:

- The consolidation workload consists of one Web server, one e-mail server, and one data base server.
- To mimic the real-world scenario, one idle VM is also running on the physical system. In that VM, no real workload is running. We do not take any score from this idle VM.
- The weight factors are 35% for the one Web server, 20% for the e-mail server, and 45% for the database server. This would correspond to a weight vector of (0.35, 0.20, 0.45).
- After testing each workload individually in a non-virtualized pre-defined standard machine with a specific configuration, we can get the baseline of each workload.
- At this stage, we define how the workloads were mapped into VMs, how the VMs were mapped to the underlying physical platform, and how resources were allocated amongst each. This is not defined by the workload. The user chooses the best VM configuration settings to do the measurements.
- Each of the workload components has a well-defined performance metric and a known, unvirtualized baseline result. The observed result for the virtualization of each of the component workloads is normalized to this known baseline. The resulting benchmark metric is the combination of all of the normalized workload component results. As an example, the performance results ratio vector WorkloadPerf[i] could look like (1.8, 1.5, 2.3)[2]. The rollup result for the performance of the virtualized system would be

---

[2] The reader is reminded that these are ratios to some predefined (different and likely older) standard system and are likely to be greater than 1.0.

N

Weight[i]*WorkloadPerf[i]

I=1

or

(0.35, 0.20, 0.45) * (1.8, 1.5, 2.3) = 1.965

This calculation is illustrated in Figure 4. The result becomes most useful when comparing different configurations. For example, the same consolidated set of workloads can be compared across different platforms or against two different virtualization monitors, or it can be used to compare two different sets of configuration settings.

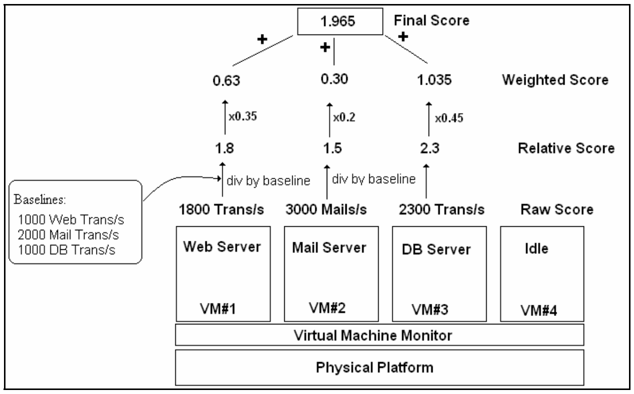

Figure 4: vConsolidate example virtualization benchmark results calculation

INDUSTRY-STANDARD VIRTUALIZATION BENCHMARKS

We have established a general methodology for developing a virtualization benchmark. The value of the benchmark will be amplified to the degree that its component workloads map to the actual work and metrics of the particular usage model. There is great value in having a proxy industry-standard benchmark since it permits the community to focus on some standard repeatable test that facilitates comparisons between configurations and platform technologies. Virtualization opportunities require that some abstraction on top of a set of workloads is developed, for example a throughput methodology (SPECthruput* and later SPECrate*) [6] applied on top of the well-known SPEC* CPU benchmarks.

Despite virtualization technologies and implementations dating back more than 30 years, no de-facto or industry-standard performance metric exists. As virtualization is exploited on commodity high-volume platforms across a range of server workloads, the community needs standards to compare virtualized performance (e.g., [7]). These are some of the motivations for an industry-standard benchmark:

- Motivate the industry to add a performance and optimization discipline to virtualization platforms, monitors, and tools.
- Help users considering virtualization to compare alternatives, particularly where they were unwilling or unable to develop, execute, and maintain their own benchmarks.
- Accelerate both the development and application of virtualization technologies.

What is a server virtualization performance benchmark and how could it be used? It is a performance discipline that includes consolidated server workloads running in virtualized partitions. It is most likely to be useful to compare platforms and virtualization technologies. Some examples on how it would be used are given:

- How does the performance, power efficiency, or price/performance of Platform A compare with Platform B?
- How does VM monitor A compare with VM monitor B (an upgrade, a competitive comparison, comparison of software and hardware accelerated implementations)?
- What is the effect of a configuration modification or a component substitution?

What is a good benchmark and how is this question modified by virtualization? We will first list what we believe is required of benchmarks and then what is desirable.

First, required attributes of benchmarks:

- They should be relevant to some well-defined user constituency like Web servers, application servers, database servers, and e-mail servers, that would be likely considered for consolidation.
- They should be timeless, in the sense they do not specify any platforms, technologies, or usage models that will not make sense over some reasonably long time.
- They should provide repeatable results from run to run and provide a means to ensure that valid results were obtained.
- They should be agnostic (in terms of platforms, virtualization monitors, and as many other parts of the stack) to the extent possible.

- They should be able to automate, repeat, scale, consolidate, and compare performance observations across a wide range of systems, times, and components.
- They should use a simple, single unit of measure for performance results (and associated component scores for those preferring more detail).
- They should be citable, which depends upon components that are freely accessible to the community.

Second, desirable attributes of benchmarks:

- They should not require infrastructure that will block most organizations from putting the system together. This rules out workloads that need a huge number of disks, a load-generation infrastructure, or an infrastructure that is otherwise encumbered.
- They should be easy to test. For example, load generation is considered a minor performance consumer and can be run within the test environment (no external load generators needed).
- They should be easy to obtain, set up, and test. For example, they should use relatively low-cost, easy-to-access and set up applications, tools, and utilities.

There are many practical challenges in constructing such a specification. Here are some examples.

- If you were to start with existing workloads, some of the most visible contemporary industry-standard server workloads may limit other components from participating in the benchmark or introduce complexities that cannot be easily managed across a range of systems and timeframes. One solution would be to develop new server workloads that were totally portable, but this would introduce a new challenge in that these would be unproven and unknown.
- Many benchmarks artificially attempt to saturate the CPU and marginally reflect best configuration practices.
- Some virtualization technologies may have different feature sets and in order to maximize those that could participate, we would have to limit the features that could be exercised, for example migration of VMs.
- Perhaps the most difficult set of decisions is to pick the constituent workloads and define an aggregation process—since there is no single right set. This will require compromises before it can be accepted by the community.

Driving a standardized benchmark is the best way to create a performance discipline, achieve continuous improvement at every layer of the platform stack, and ultimately develop an industry standard for measuring and optimizing virtualization performance.

CONCLUSION

In order to be agnostic to the underlying operating environments and implementation, we have taken a high-level approach that assumes a coherent performance discipline on each constituent workload component and have presented an approach on how to aggregate these into a unified metric for comparing systems. Since there are many different user requirements for assessing and projecting performance, there is no one right answer for what workloads should be included and how they should be combined, so we have provided readers with considerations for doing this in a manner that best addresses their needs. The vConsolidate benchmark was presented as an example implementation, highlighting the compromises required in workload selection, component definition, and metric aggregation. Industry standards for measuring the performance of virtualization environments will help accelerate the performance and deployment of same. The performance discipline presented outlined a basic framework that could be used to create industry-standard virtualization benchmarks.

ACKNOWLEDGMENTS

We thank Tom Adelmeyer, Andrew Anderson, Mark Brown, Clement Cole, Aaron Holzer, Ken Rule, and Jeff Smits for their technical contribution and review comments. Thanks to the paper's advisory sponsors: Naresh Sehgal, Paul Barr, Rich Uhlig, and Fernando Martins. A special thanks to the performance engineering team including Nan Bo, Vivi Huang, Steven Thomsen, Steve Yang, Mark Brown, Aamir Yunus, Ashok Emani, Anthony P. Bertapelli, and Priyalal Kulasinghe for evaluating many consolidation and virtualization test methodologies.

REFERENCES

[1] Creasy, R.J., "The Origin of the VM/370 Time-Sharing System," *IBM Journal of Research and Development*, vol. 25, no. 5, p. 483.

[2] Uhlig, R.; Neiger, G.; Rodgers, D.; Santoni, A.; Martins, F.; Anderson, A.; Bennett, S.; Kagi, A.; Leung, F.; and Smith, L., "Intel Virtualization Technology," *Computer*, vol. 38, no. 5, pp. 48–56.

[3] IDC. Server Virtualization 2005, June 2005, pp. 1–96.

[4] Parthasarathy Ranganathan and Norman Jouppi, "Enterprise IT Trends and Implications for

Architectural Research (2005)," in *Proceedings of the 11th International Symposium on High Performance Computer Architecture* (HPCA-11 2005), at http://www.hpcaconf.org/hpca11/papers/24_x_ranganathan-enterpriseittrends.pdf*

[5] VMware Whitepaper (2005), Timekeeping in VMware VMs, at http://www.vmware.com/pdf/vmware_timekeeping.pdf*

[6] Greenfield, Mike; Rayhona, Paul; and Bhandarkar, Dileep, "SPEC adopts new methodology for measuring system throughput. SPEC Newsletter," 2 (2), Spring 1990.

[7] Green, Diane (2006), Web Blog at http://www.vmware.com/vmtn/blog/diane/* advocating consistency and ease of making performance comparisons.

AUTHORS' BIOGRAPHIES

Jeffrey P. Casazza joined Intel in 1988 and has contributed to various projects including the MCS®96, i860®, Itanium® and Xeon® processor families. He holds a B.S.E.E. degree from Arizona State University. His e-mail is jeffrey.p.casazza at intel.com.

Michael Greenfield is a principal engineer at Intel. He joined Intel in 1997 and has held a variety of management and development roles in Workstations, High Performance Computing, and System Engineering. He holds an M.S.M.E. degree from Cornell University. His e-mail is michael.greenfield at intel.com.

Kan Shi obtained his B.S.E.E. degree from Shanhai Jiaotong University. He has been with Intel since 2003. Kan is currently a marketing engineer with Intel's Digital Enterprise Group responsible for server virtualization performance testing. His e-mail is kan.shi at intel.com.

THIS PAGE INTENTIONALLY LEFT BLANK

For further information visit:
developer.intel.com/technology/itj/index.htm