



Impact of Enhanced vMotion Compatibility on Application Performance

Performance Study

TECHNICAL WHITE PAPER

Table of Contents

Executive Summary3

Introduction.....3

Enhanced vMotion Compatibility.....4

 Scalability4

 How Does EVC Work?.....5

 EVC Baseline.....5

 EVC Requirements.....6

Test Environment.....6

 Workloads Studied6

 Hardware and Software7

Database.....7

Java Applications8

Encryption9

Multimedia.....9

String Processing.....10

Best Practices.....10

Conclusion11

References12

Executive Summary

VMware® Enhanced vMotion Compatibility (EVC) enhances the scope of VMware vSphere® vMotion® by making VMware® ESXi™ hosts with different CPU technologies compatible for vMotion. It does this by making available a common CPU feature set through the use of a baseline. With a baseline in place for older processors, application performance becomes important. Do the applications running in the virtual machines with an older CPU presented perform as well as they do on virtual machines that have access to feature sets available in newer generation processors? In this paper, we quantify the performance impact of EVC mode on a diverse set of applications. We study workloads from database, Java, multimedia, and encryption categories and report the results.

Test results show that almost all workloads perform well even when the virtual machine presents an EVC mode that corresponds to an older processor generation. The EVC mode setting had varying impact on workload performance based on the ESXi hosts' CPU instruction set features made available and their relevance to the workloads. One workload, AES-Encryption, didn't fare as well due to a dependence on special-purpose instruction sets only available in younger processor generations.

Introduction

VMware vMotion [1] plays a critical role in data center management; virtual machine migration helps in load balancing, resource management, and preventive maintenance. Clusters in a typical datacenter usually have a mix of processors belonging to different generations, if not different vendors. Processor vendors continue to offer special purpose enhancements targeting individual market segments with each new generation. In light of this heterogeneous nature of a cluster and attendant hardware dependencies, vMotion is forced to limit the possible destinations to which a virtual machine could migrate.

In order to ease this restriction, VMware supports EVC mode [2] [3] through the use of Intel FlexMigration and AMD-V Extended Migration technologies. EVC mode can be specified at the cluster level. This sets a baseline processor generation enabling wider migration choices for a virtual machine, typically the oldest or the least capable processor becomes the determining EVC mode for all the hosts in that cluster. Subsequently, any virtual machine running in that cluster can be migrated to any other ESXi host, regardless of CPU, within the cluster.

Despite the obvious advantages of EVC mode, administrators need to factor in the costs associated with this feature to help in the decision making process. Some applications could potentially lose performance due to certain advanced CPU features not being made available to the guest even though the underlying host supports them. This has been a concern for VMware customers, partly due to the lack of information on the extent of performance loss and the class of applications that get affected.

In this study, we aim to quantify the performance impact of EVC mode for a set of applications covering a spectrum of application domains.

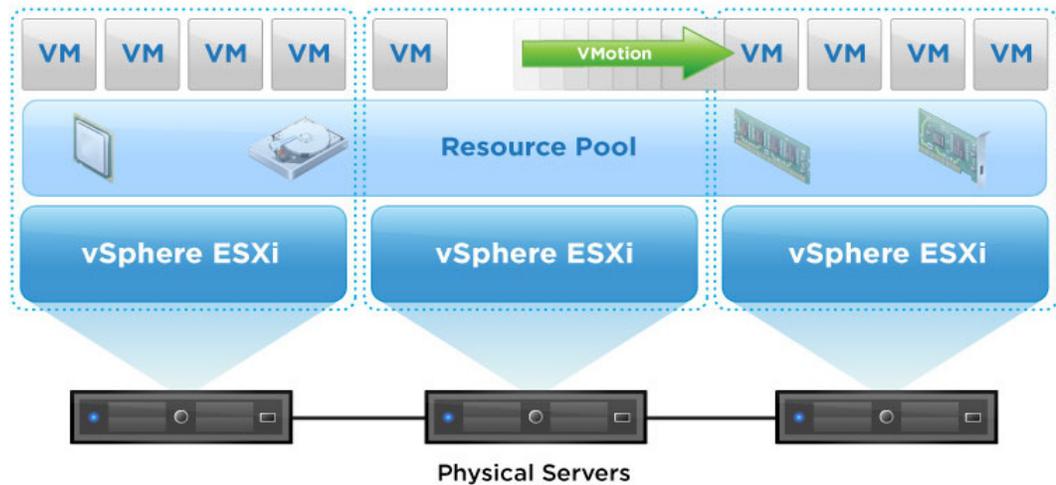


Figure 1. Live migration of machines through VMware's vMotion

Enhanced vMotion Compatibility

EVC mode allows migration of virtual machines between different generations of CPUs, making it possible to aggregate older and newer server hardware generations in a single cluster. This flexibility provides scalability of the virtual infrastructure by offering the ability of adding new hardware into an existing infrastructure while extending the value of existing hosts.

Scalability

EVC allows IT organizations to scale out (expand) their existing infrastructures by increasing the number of ESXi hosts available for vMotion. Instead of having to wait for a purchasing window to buy servers in bulk to ensure a homogenous cluster that meets vMotion requirements, IT professionals can use existing hardware with mixed CPU technology to increase the ESXi hosts in their clusters and still maintain vMotion compatibility.

EVC is complimentary to the popular design methodology of the building block architecture. Building blocks extend the concept of a framework and outline a pre-defined set of items or modules that allow for scalability while maintaining standardization. Cluster configurations are often treated as a single building block structure, which can limit an organization's procurement strategy of server hardware. This is especially true in environments where migration using vMotion is necessary but the ESXi hosts have different CPU generations and are incompatible for vMotion. In this case, the best approach has been to aggregate server hardware. Aggregation can be difficult, however, because the release cycle of hardware generations are typically shorter than financial purchasing windows. Bulk procurement of machines during a financial purchasing window can lead to an oversized, underutilized cluster configuration in the early stages of its lifecycle. By enabling EVC, the requirement of identical hardware to provide maximum portability of virtual machines within the cluster is removed. Enabling EVC allows for expanding clusters gradually and allows for future expansion of clusters while still aligning with building block architectures.

How Does EVC Work?

EVC creates a baseline that allows all the hosts in the cluster to advertise the same CPU feature set. The EVC baseline does not disable the features within a CPU, but indicates to a virtual machine that specific features are not available. EVC only focuses on CPU features specific to CPU generations, such as SIMD (SSE) or AMD-now instructions. **EVC hides these CPU features from software running inside virtual machines by not advertising these features. This means that the features are still available and active, but they are not “publically broadcasted.”** When enabling EVC, a CPU baseline must be selected. This baseline represents a feature set of the selected CPU generation and exposes specific CPU generation features. If a virtual machine powers-on, this baseline will be attached to the virtual machine until it powers off.

Note: The EVC baseline is attached to the virtual machine until it powers off, even if the virtual machine is migrated to another EVC cluster.

EVC Baseline

When an ESXi host with a newer generation CPU joins the cluster, the baseline will automatically “hide” the CPU features that are new and unique to that CPU generation. As an example, suppose an administrator has a cluster containing ESXi hosts configured with Intel® Xeon® Core™ i7 CPUs, commonly known as Intel “Nehalem.” The baseline selected – *Intel® “Nehalem” Generation* – presents the cumulative features of the *Intel® “Merom” Generation*, *Intel® “Penryn” Generation* and the *Intel® “Nehalem” Generation* to the virtual machine. This has the net effect of providing the standard *Intel® “Merom” Generation* features plus SSE4.1, SSE4.2, Popcount and RDTSCP features available to all the virtual machines. When an ESXi host with a Westmere (32nm) CPU joins the cluster, the additional CPU instruction sets like AES/AESNI and PCLMULQDQ will be suppressed automatically.

For our testing, we chose baselines of Intel processor generations “Westmere,” “Nehalem,” “Penryn,” and “Merom.”

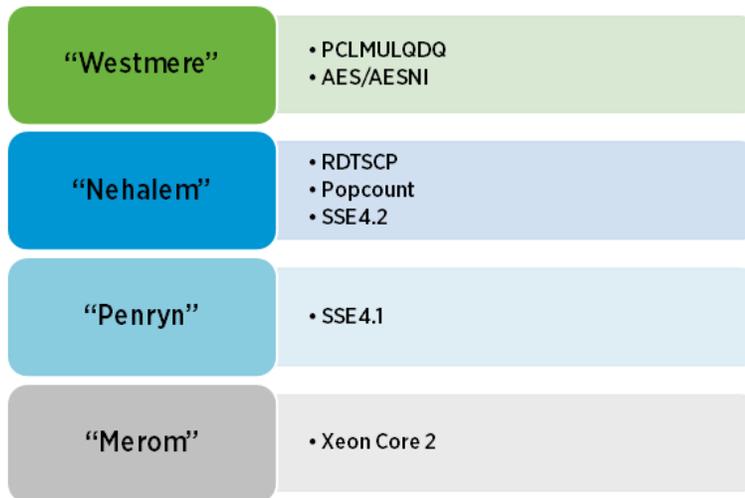


Figure 2. Intel processor generations and corresponding features

Note: The figure above provides some examples, it does not list all of the differences.

EVC Requirements

To enable EVC on a cluster, the cluster must meet the following requirements:

- All hosts in the cluster must have CPUs from a single vendor, either AMD or Intel.
- All hosts in the cluster must have advanced CPU features, such as hardware virtualization support (AMD-V or Intel VT) and AMD No eXecute (NX) or Intel eXecute Disable (XD) and must be enabled in the BIOS.
- All hosts in the cluster should be configured for vMotion.
- All hosts in the cluster must be connected to the same vCenter Server system.

In addition, all hosts in the cluster must have CPUs that support the EVC mode you want to enable. To check EVC support for a specific processor or server model, see the *VMware Compatibility Guide* [4]. Interaction of EVC and Hardware Virtualization Support

VMware's hypervisor is unique in that it supports a variety of execution modes depending on the capabilities of the hardware. The VMkernel automatically selects the best hypervisor execution mode that will deliver the best virtual machine performance given the capabilities of the hardware and type of guest operating system. These virtualization acceleration features such as VT-x/AMD-V and RVI/EPT are available for use by the hypervisor independent of EVC or the EVC baseline, and the VMkernel switches on the fly to whatever mode is the best performing one for the guest as the virtual machine is migrated around the cluster.

Test Environment

Our goal is to replicate the scenario of an administrator assigning a server being downgraded in terms of processor generation due to the presence of older generation nodes in a cluster so that all the hosts can be available for vMotion. Accordingly, we created a cluster, with EVC-mode enabled, in a datacenter. We added a host based on the Intel Xeon "Westmere" processor to the cluster. We created several guest virtual machines to run several workloads with different EVC modes ranging from Intel "Merom" to Intel "Westmere."

Workloads Studied

We selected several workloads to represent different classes of popular applications running in the enterprise and compared the performance of each when the virtual machine was set with an EVC mode presenting different processor generations. This test was done to determine if virtual machines with lower processor generation EVC settings performed at the same level as those with higher processor generation EVC settings. The workloads chosen are shown in Table 1.

APPLICATION CLASS	WORKLOAD
Database	Oracle SwingBench
Java	SPECjbb2005
Encryption	OpenSSL (version 1.0.0)
Multimedia	H264 Video Encoding (X264 0.120.X)

Table 1. Workloads selected to represent different types of popular applications

These applications and their performance with EVC are described in the following sections.

Hardware and Software

We used the following hardware and software in the guest and the host.

Guest:

- Hardware: 1 virtual CPU, 8GB RAM, 32GB hard disk
- Operating System: Red Hat Enterprise Linux 6.1 (Kernel Version 2.6.32)

Host:

- Hardware: Dual 6-core Intel® Xeon® Processor X5680 @3.324GHz, 144GB RAM
- Operating System: ESXi 5.1

Database

To meaningfully understand the performance variations among database workloads across processor generations, we selected a popular benchmark named Oracle SwingBench [5]. Oracle SwingBench is a load generator for Oracle database, wherein the response times of various user transactions could be measured. We made use of one of the benchmarks supplied with SwingBench, named OrderEntry.

The setup consisted of a server virtual machine and a client virtual machine. The client virtual machine was hosted on a different node in a separate cluster, connected by a 10Gb Ethernet link. Our focus is on the server virtual machine, which houses the database and responds to transactions initiated by the client. In a typical SwingBench setup, the server performance almost exclusively determines overall transaction rate. Thus, we created a virtual machine with RHEL 6.1 and installed the SwingBench server component on it and paired it with the client virtual machine over the network.

We varied the EVC mode of the cluster in which the SwingBench server virtual machine was hosted and measured the transaction rate reported by the client. The results are presented in Figure 3.

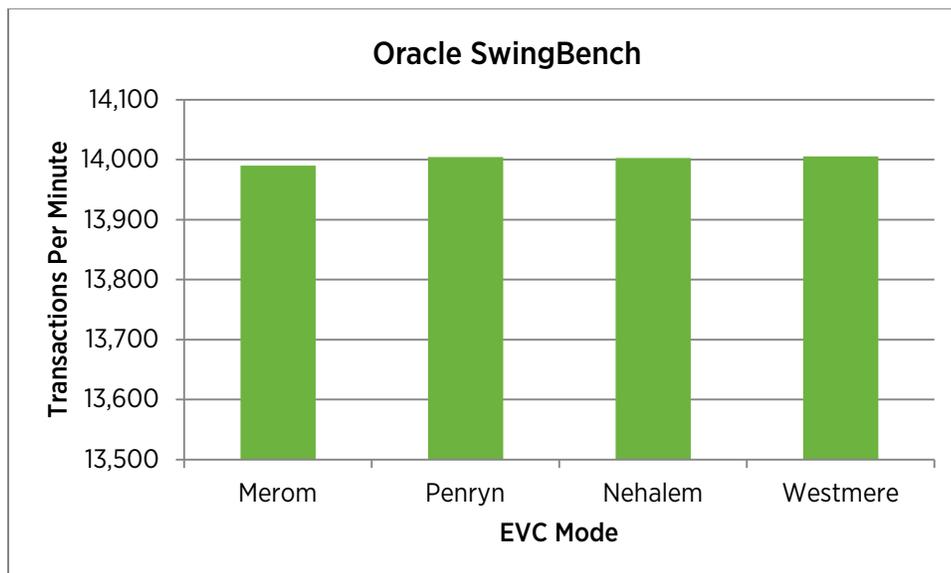


Figure 3. Transaction processing rate of Oracle SwingBench with different EVC modes

The figure shows that there is no significant variation in the transaction processing rate with EVC modes. A database server virtual machine will maintain its performance on an ESXi server with an Intel processor “Westmere” even when the older processor capabilities of “Nehalem,” “Penryn,” and “Merom” are presented through EVC.

Java Applications

We ran EVC mode experiments on the industry-standard, server-side Java benchmark SPECjbb2005[®] [6]. We used a non-compliant test configuration of one JVM with a heap size of 256MB and ran the benchmark with the warehouse sequence of {1, 2}, which starts the server-side Java application with one warehouse workload and then increases to two warehouses. The guest was a Red Hat Enterprise Linux 6.1 with 8GB RAM and 32GB hard disk virtual machine with one virtual CPU. The ESXi host is described in “[Hardware and Software](#)” on page 7. We ran SPECjbb with different EVC modes and present the reported scores, which reflect testing published on August 26, 2012.

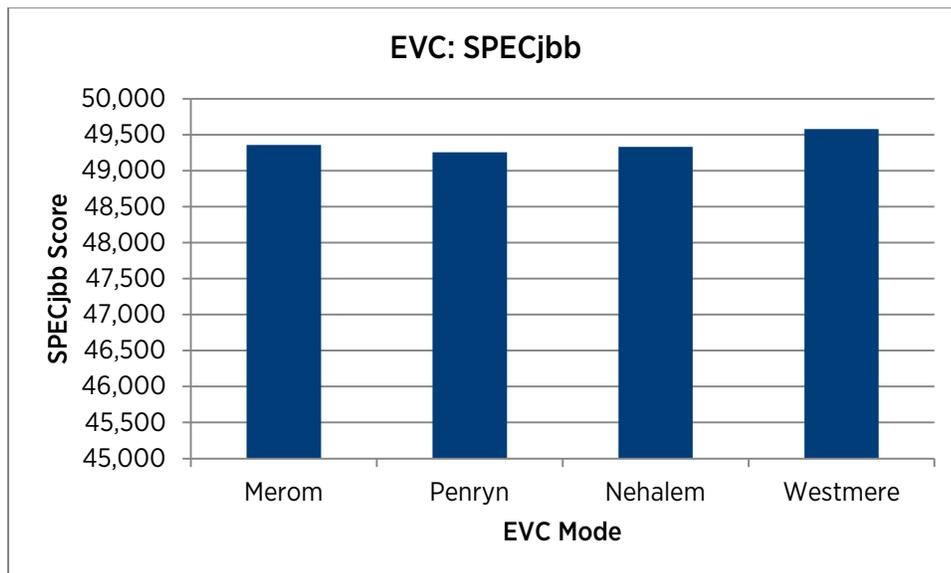


Figure 4. Scores from SPECjbb runs with different EVC modes

As shown in Figure 4, we observed negligible variation (0.007%) in the measured scores reported by these experiments across EVC modes. This means that a virtual machine running a Java-based server-side application maintains its performance on an ESXi host with processor capabilities as new as “Westmere” and as old as “Merom.”

Encryption

In this experiment, we used the open source encryption application OpenSSL (Version 1.0.0) [7] to measure the encryption speed using the Advanced Encryption Standard (AES) cipher with different key sizes. AES [8] has a key size of 128, 192, or 256 bits.

OpenSSL has a built-in performance test feature named *speed*, with which one can measure the operations performed in a given time. We chose to tabulate the results for a block size of 8192 bytes with different key sizes. In order to measure to the impact of EVC mode, we repeated this experiment with different Intel EVC modes: “Merom,” “Penryn,” “Nehalem,” and “Westmere.” The results are presented in the chart below.

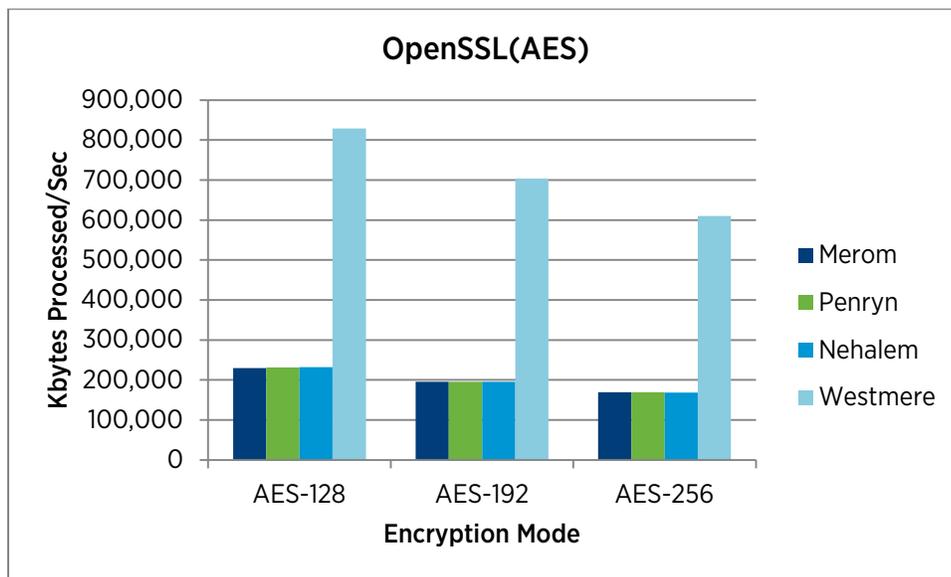


Figure 5. Encryption speed reported by OpenSSL (AES algorithm) with different EVC modes

As shown in Figure 5, Intel-Westmere EVC mode outperforms other modes by more than three times. This improved performance is due to the encryption acceleration made possible by the introduction of the AESNI instruction set available on Intel processors “Westmere.” Administrators running encryption applications in a virtual machine on a “Westmere” capable ESXi host should expect a drop in application performance when the ESXi host is mixed with ESXi hosts that feature older generation processors.

Multimedia

CPU vendors have continually enhanced multimedia performance through various hardware improvements as well as instruction set extensions. The instruction set extensions have come in the form of SSE versions being introduced regularly: SSE1,2,3,4, SSE4.1, and SSE4.2. In the cross-section of EVC modes we evaluated, all the processor generations corresponding to the EVC modes incorporate all SSE versions prior to SSE4. SSE4 has two subsets named SSE4.1 and SSE4.2. SSE4.1 has extensions to the core SSE instruction set focused on multimedia, while SSE4.2 introduces new string and text instructions that are intended to accelerate string processing.

For evaluating multimedia workloads, we focus on the performance improvements resulting from the SSE4.1 instruction set. We selected a relatively performance-intensive video encoding task as our workload. We selected the popular open source implementation of H.264 standard video encoder implementation called X264 [9] in two pass fast encoding mode. The input to X264 is a raw HD video file of 1.3GB in size, which ensured that the encoder was able to recognize and use the SSE4.1 instruction set in applicable EVC modes. The performance is measured in terms of frames processed per second for different EVC modes and the results are presented in Figure 6.

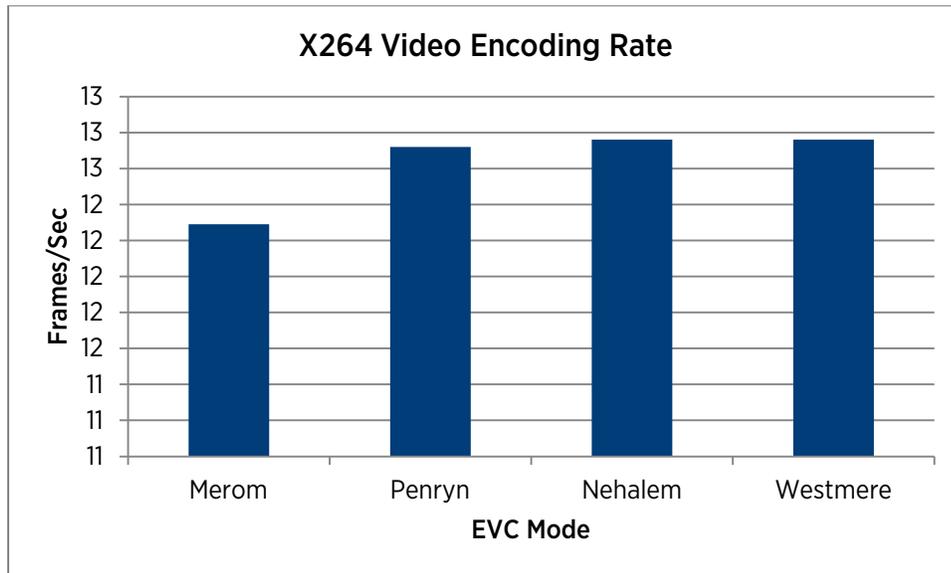


Figure 5. Video encoding rate of X264 with different EVC modes

The figure shows approximately 4% improvement in the encoding rate by going from “Merom” to “Penryn.” Beyond “Penryn,” the multimedia application maintained its performance. This is due to SSE4.1 being supported in all those EVC modes with no other multimedia-specific instruction set extension in play.

Administrators can include ESXi hosts featuring Intel “Penryn,” “Nehalem,” and “Westmere” in a vMotion cluster with ESXi hosts as old as “Penryn” and expect similar performance in virtual machines running multimedia applications.

String Processing

As mentioned in the previous section, the SSE4 instruction set has two subsets named SSE4.1 and SSE4.2. SSE4.2 introduces new string and text instructions (STTNI) that are intended to speed up string and document processing in general and accelerate XML parsing in particular.

Unfortunately, there is no commercially available XML parsing engine that makes use of STTNI. We experimented with a popular XML processor, Apache Xerces, and found that EVC mode did not affect any measurable performance variation. We expect the performance of such applications to be maintained in virtual machines with an EVC mode of Intel “Merom” through Intel “Westmere.”

Best Practices

Based on our tests, we recommend administrators consider the types of workloads running in a cluster before enabling EVC mode. For a large class of mainstream workloads, including those of databases and Java applications, our studies showed no impact on performance by setting EVC mode to an older processor generation. We identified two workloads that didn’t fare as well: AES-encryption and video encoding. We observed only a slight degradation for video encoders on Intel “Merom” Generation EVC mode. We also observed performance loss for AES-encryption workloads on EVC modes prior to Intel “Westmere.” Administrators should consider what an acceptable performance loss is in these cases. For the multimedia case, make sure the oldest EVC baseline is Intel “Penryn” Generation to ensure application performance. For the encryption case, make sure the application performs well at the Intel “Merom” Generation setting. Otherwise, set up a homogenous cluster to ensure vMotion compatibility across ESXi hosts or warn users to expect performance loss for these applications.

Conclusion

In this paper, we examined a cross-section of workloads to understand and quantify the performance impact of EVC mode on real-life applications. We selected representative workloads from different domains including encryption, database, Java, and multimedia. We collected performance data in a cluster with EVC capability enabled and we presented the results.

We demonstrated that AES encryption workloads benefit from the presence of AESNI available in processor generations Intel “Westmere” and onwards, to the extent of about a three times gain. When it’s required to select an EVC mode to align with an older processor generation due to the presence of older generation nodes in a cluster, performance loss for some operations can occur. For example, selecting a pre-“Westmere” EVC mode can lead to performance loss on “Westmere” or post-“Westmere” systems due to a loss of AES functionality. The multimedia workload, video encoding, showed minor performance loss by going to the Intel “Merom” Generation (due to loss of SSE4.1). Barring these, other types of workloads showed no discernible performance loss by stepping down to older processor generations.

References

- [1] VMware, Inc., "VMware vMotion: Live Migration for Virtual Machines Without Service Interruption." 2009. <http://www.vmware.com/files/pdf/VMware-vMotion-DS-EN.pdf>.
- [2] VMware, Inc., "Enhanced vMotion Compatibility (EVC) processor support." VMware Knowledge Base, 13 June 2012. <http://kb.vmware.com/kb/1003212>.
- [3] VMware, Inc., "CPU Compatibility and EVC," ESXi and vCenter Server 5.0 Documentation. <http://pubs.vmware.com/vsphere-50/index.jsp>.
- [4] VMware, Inc., "VMware Compatibility Guide." <http://www.vmware.com/resources/compatibility/search.php>.
- [5] G. Dominic, "SwingBench 2.2 Reference and User Guide." 8 August 2005. <http://dominicgiles.com/swingbench/swingbench22.pdf>.
- [6] Standard Performance Evaluation Corporation, "SPECjbb2005." <http://www.spec.org/jbb2005>.
- [7] OpenSSL Project, "Open source toolkit for SSL/TLS." 10 May 2012. <http://www.openssl.org>.
- [8] C. Paar, J. Pelzl and B. Preneel, "The Advanced Encryption Standard," in *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer Berlin Heidelberg, 2009.
- [9] "x264 - a free h264/AVC encoder," VideoLAN Organization. <http://www.videolan.org/x264.html>.

