2013-12

# Feasibility of virtual machine and cloud computing technologies for high performance computing

Hutchins, Richard Chad

Monterey, California. Naval Postgraduate School

# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**FEASIBILITY OF VIRTUAL MACHINE AND CLOUD COMPUTING TECHNOLOGIES FOR HIGH PERFORMANCE COMPUTING**

by

Richard Chad Hutchins

December 2013

Thesis Co-Advisors: Albert “Buddy” Barreto
James Hansen

**Approved for public release; distribution is unlimited**

***Reissued May 2014 with additions to the acknowledgments***

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | *Form Approved OMB No. 0704-0188* |
|---|---|



| **1. AGENCY USE ONLY *(Leave blank)*** | **2. REPORT DATE**<br>December 2013 | **3. REPORT TYPE AND DATES COVERED**<br>Master's Thesis |
|---|---|---|

| | |
|---|---|
| **4. TITLE AND SUBTITLE**<br>FEASIBILITY OF VIRTUAL MACHINE AND CLOUD COMPUTING TECHNOLOGIES FOR HIGH PERFORMANCE COMPUTING | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Richard Chad Hutchins | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>N/A | **10. SPONSORING/MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB protocol number ____N/A____.

| **12a. DISTRIBUTION / AVAILABILITY STATEMENT**<br>Approved for public release; distribution is unlimited | **12b. DISTRIBUTION CODE**<br>A |
|---|---|

**13. ABSTRACT (maximum 200 words)**

Knowing the future weather on the battlefield with high certainty can result in a higher advantage over the adversary. To create this advantage for the United States, the U.S. Navy utilizes the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) to create high spatial resolution, regional, numerical weather prediction (NWP) forecasts. To compute a forecast, COAMPS runs on high performance computing (HPC) systems. These HPC systems are large, dedicated supercomputers with little ability to scale or move. This makes these systems vulnerable to outages without a costly, equally powerful secondary system. Recent advancements in cloud computing and virtualization technologies provide a method for high mobility and scalability without sacrificing performance. This research used standard benchmarks in order to quantitatively compare a virtual machine (VM) to a native HPC cluster. The benchmark tests showed that the VM was feasible platform for executing HPC applications. Then we ran the COAMPS NWP on a VM within a cloud infrastructure to prove the ability to run a HPC application in a virtualized environment. The VM COAMPS model run performed better than the native HPC machine model run. These results show that VM and cloud computing technologies can be used to run HPC applications for the Department of Defense

| | |
|---|---|
| **14. SUBJECT TERMS** Virtual machines, cloud computing, numerical weather prediction, high performance computing, benchmarks, cloud infrastructure, infrastructure as a service, COAMPS, COAMPS-OS, continuity of operations | **15. NUMBER OF PAGES**<br>55 |
| | **16. PRICE CODE** |

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500 — Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

# FEASIBILITY OF VIRTUAL MACHINE AND CLOUD COMPUTING TECHNOLOGIES FOR HIGH PERFORMANCE COMPUTING

Richard Chad Hutchins
Civilian, Department of the Navy
B.S., University of North Carolina at Asheville, 2008

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL**
**December 2013**

Author: Richard Chad Hutchins

Approved by: Albert "Buddy" Barreto
Thesis Co-Advisor

James Hansen
Thesis Co-Advisor

Dan Boger
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Knowing the future weather on the battlefield with high certainty can result in a higher advantage over the adversary. To create this advantage for the United States, the U.S. Navy utilizes the Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) to create high spatial resolution, regional, numerical weather prediction (NWP) forecasts. To compute a forecast, COAMPS runs on high performance computing (HPC) systems. These HPC systems are large, dedicated supercomputers with little ability to scale or move. This makes these systems vulnerable to outages without a costly, equally powerful secondary system. Recent advancements in cloud computing and virtualization technologies provide a method for high mobility and scalability without sacrificing performance. This research used standard benchmarks in order to quantitatively compare a virtual machine (VM) to a native HPC cluster. The benchmark tests showed that the VM was feasible platform for executing HPC applications. Then we ran the COAMPS NWP on a VM within a cloud infrastructure to prove the ability to run a HPC application in a virtualized environment. The VM COAMPS model run performed better than the native HPC machine model run. These results show that VM and cloud computing technologies can be used to run HPC applications for the Department of Defense.

THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| CentOS | Community Enterprise Operating System |
| COAMPS | Coupled Ocean/Atmosphere Mesoscale Prediction System |
| COAMPS-OS | Coupled Ocean/Atmosphere Mesoscale Prediction System-On Scene |
| COOP | continuity of operations |
| CPU | central processing unit |
| CSFV | cubed-sphere-finite-volume |
| DoD | Department of Defense |
| DoD CIO | Department of Defense Chief Information Officer |
| D-VTM | distributed virtual task machine |
| FNMOC | Fleet Numerical Meteorology and Oceanography Center |
| HPC | high performance computing |
| I/O | input/output |
| IaaS | infrastructure as a service |
| iSCSI | internet small computer system interface |
| MPI | message passing interface |
| NAS | NASA advanced supercomputing |
| NASA | National Aeronautics and Space Administration |
| NAVGEM | Navy global environmental model |
| NCSA | National Center for Supercomputing Applications |
| NIST | National Institute of Standards and Technology |
| NOGAPS | Navy Operational Global Atmospheric Prediction System |
| NPB | NAS parallel benchmark |
| NPS | Naval Postgraduate School |
| NPS-CCL | Naval Postgraduate School—Cloud Computing Laboratory |
| NRL | Naval Research Laboratory |
| NWP | numerical weather prediction |
| OMB | OSU Micro-Benchmarks |
| OS | operating system |
| OSU | Ohio State University |

| | |
|---|---|
| PaaS | platform as a service |
| REHL | Red Hat Enterprise Linux |
| SaaS | software as a service |
| VM | virtual machine |
| vNUMA | virtual non-uniform memory access |
| WRF | weather research and forecasting model |

# ACKNOWLEDGMENTS

First, thank you to the Naval Research Laboratory (NRL), Marine Meteorology Division for allowing me the opportunity to attend the Naval Postgraduate School and conduct this research at NRL. Specifically I would like to thank my supervisor John Cook for understanding the academic responsibilities involved and helping me balance them with my work priorities. I would also like to thank the Office of Naval Research (PE 0602235N) for sponsoring this research.

I would like to thank my co-advisors Buddy Barreto and Jim Hansen. Thank you Buddy for taking me on as your student, letting me borrow the keys to your computing castle, and for all your guidance and expertise. Thank you Jim for advising me even though you the busiest person I know! And even though you are busy, you were always there. Thanks for your constant advice and pushing me to keep going.

I would also like to thank my co-workers at NRL specifically: Dan Martinez for putting up with my school schedule, Dan Geiszler for your invaluable technical support and help with COAMPS-OS, Bill Green for your IT support, and Tim Whitcomb for FORTRAN support.

I also would like to thank my parents and family for always supporting me. And lastly my wife, Megan, saying thank you for always supporting me would be an understatement; you helped me accomplish my goal of completing this degree. Without you, I would not have completed this. Thank you for your love and knowledge. Now we finally have our weekends back!

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. BACKGROUND

The Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS) [1] is the U.S. Navy's mesoscale scale (regional) numerical weather model. COAMPS is run on High Performance Computing (HPC) systems. The purpose behind the U.S. Navy running regional weather models is to predict the weather in tactical environments around the world on an on-demand basis. Presently, the Fleet Numerical Meteorology and Oceanography Center (FNMOC) in Monterey, California (CA) are responsible for running COAMPS and providing its output to the Fleet. At FNMOC, COAMPS is maintained on a 24/7 operations watch floor environment [2]. In order to properly accomplish the task of running and managing COAMPS, FNMOC utilizes dedicated, large scale HPC cluster systems to run COAMPS. In addition to this, FNMOC maintains the Navy global environmental model (NAVGEM), which initializes each and every COAMPS run [2].

While there are many advantages for running COAMPS at FNMOC on its HPC systems, there are a number of disadvantages that should be addressed. First, the HPC systems at FNMOC are limited in its ability to scale. This is mainly due to the dedicated system architecture's support for scaling to the physical requirements where the system resides. Another disadvantage to be considered is the fact that all NWP operations in the U.S. Navy are currently run at only one facility, which is FNMOC. This greatly increases the risk of downtime during a major crisis, which could be an act of war or a natural disaster. The Monterey Peninsula is located in a tsunami zone as well as being located on the San Andreas Fault. Taking these issues into regard, one might think that having the ability to rapidly deploy a NWP model to another large computing cluster resource or to a mobile computing center would be of great value in the event that FNMOC were to experience a casualty. At this time, this capability does not exist.

A virtual machine (VM) can be defined as the abstraction of a computing system, generally its operating system (OS), from its hardware through the use of software. Utilizing VM technology can provide high mobility and high scalability needed to maintain continuity of operations (COOP) [3]. VMs can also provide many management benefits, and the ability to have a customized OS [4]. VMs are often provided by or deployed to the “cloud” as part of an infrastructure as a service (IaaS). The cloud is a colloquial term for providing computing resources from either a private or public provider.

The Department of Defense (DoD) has begun to realize the potential value of utilizing cloud computing for operational purposes when the DoD Chief Information Officer (DoD CIO) released the *Department of Defense Cloud Computing Strategy* in 2012. The strategy seeks to move the department away from the current “state of duplicative, cumbersome, and costly set of application silos to an end state that is agile, secure, and cost effective environment that can rapidly respond to changing mission needs” [5]. After considering what we know about VM technology along with the DoD’s newfound interest in cloud computing, we felt that the utilization of cloud computing with virtualization for HPC applications could potentially remove single use supercomputers and consolidate the processing to the cloud of computing resources.

COOP and datacenter consolidation are some of the main drivers for cloud computing and virtualization [5]. The ability to forward deploy a NWP model or HPC system in a communications adverse environment is also of interest [6]. The cloud computing laboratory’s footprint at the Naval Postgraduate School can be used to mimic shipboard or small networks [7], [8]. This research could determine the feasibility of a forward deployed NWP model or HPC system. This research will create, test, and evaluate the ability to run a military grade NWP model within a VM in a private cloud computing infrastructure.

## B. DEFINITIONS AND PRIOR RESEARCH

### 1. Virtualization

Virtualization is the logical abstraction of a hardware or software system. Virtualization's roots dates back to the late 1960s when IBM developed virtualization technology to increase the shared usage of computer resources among a large group of users [9]. Today, there are two types of virtualization, application and hardware (Figure 1).

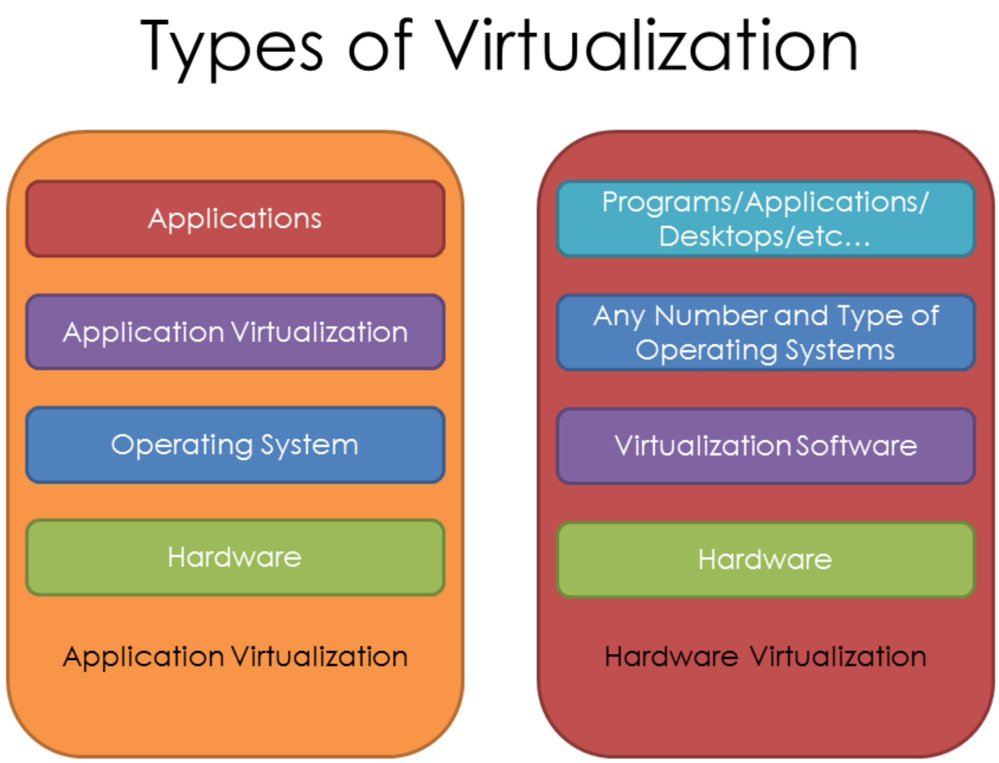



Figure 1. Diagram of the major types of virtualization. There are two main types of virtualization: application and hardware. Our study focuses on hardware virtualization and its use by the U.S. Navy to run weather models more efficiently.

The first main type of virtualization is application virtualization. Application virtualization is defined when the communication between an application and the underlining OS is virtualized (Figure 1). An example of application virtualization is the lightweight, virtual framework called a Distributed Virtual Task Machine (D-VTM). D-VTM provides resource factories, resource managers, and abstract resources for distributed systems [10]. In relation to cloud computing, application virtualization is typically provided by platform as a service (PaaS) companies where they control the physical infrastructure and provide programming language support, services, and other tools to deploy applications [11].

The second main type of virtualization is hardware virtualization. Hardware virtualization is defined when communication between the OS and the hardware is virtualized through the use of virtualization software (Figure 1). Many commercial and open source hardware virtualization solutions exist, including, but are not limited to, VMware and Xe respectively. Hardware virtualization is highly desirable when organizations have the need for many operating systems to reside on one computing resource. Hardware virtualization solutions typically provide many features including customized OS, security, management features, performance isolation and more [4].

### 2. Cloud Computing

Cloud computing as defined by the National Institute of Standards and Technology (NIST):

> is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models. [11].

Our study utilizes the Cloud Computing Laboratory at the Naval Postgraduate School (NPS-CCL) to host our VM for the purpose of running COAMPS HPC application.

### *a. Service Models*

The cloud computing model is composed of three service models: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). These service models can be expressed as layers of abstraction of the cloud infrastructure as shown in Figure 2.

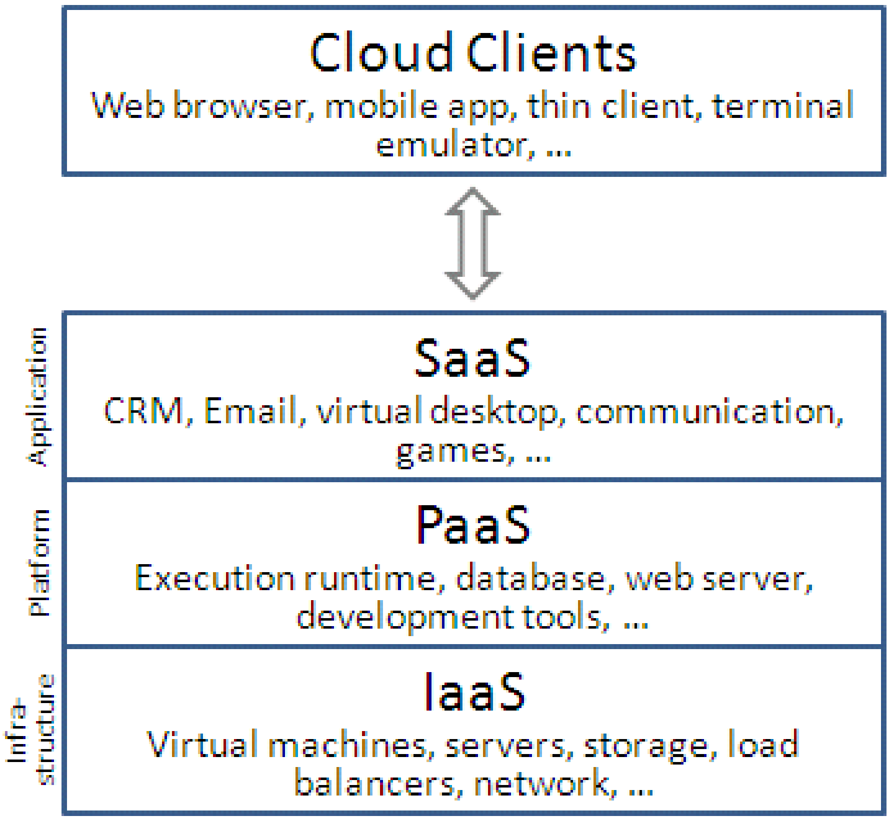



Figure 2. Schematic diagram that displays the three different types of cloud computing layers. The IaaS is shown at the bottom of the diagram to demonstrate how it is the basis the three layers (from [12]).

IaaS provides the core infrastructure in a cloud service. As defined by the NIST, IaaS should provide the ability to provision processing, storage, networks (virtual or logical), and other resources for a user to deploy and run software, including an OS and/or applications [11].

### 3. High Performance Computing

High performance computing (HPC) is defined as the clustering of computing power to accomplish high performance tasks. HPC differs from other high computational system terms like High Throughput Computing in that HPC "brings enormous amounts of computing power to bear over relatively short periods of time" [13].

### 4. Prior Research

A number of prior research studies have investigated the ability and performance of running HPC applications in virtual machines.

In 2006, Huang *et al.* conducted a case study analysis of HPC computations with VMs. Huang's analysis concluded that "HPC applications can achieve almost the same performance as those running in a native, non-virtualized environment" [4]. Huang *et al.* achieved this by addressing two challenges they concluded were the reasons why VM technologies have not been adopted in HPC: virtualization overhead and management efficiency [4].

Huang *et al.* developed a framework that bypassed the hypervisor (called virtual machine monitor bypass I/O) and in addition provided a scalable VM management system [4]. This addressed the virtualization overhead and management efficiency respectively. To test their framework, Huang *et al.* conducted performance evaluations on an eight node, 3.0Ghz Intel Xeon CPU with 2GB of RAM with an InfiniBand interconnect. InfiniBand is a high bandwidth and low latency network communication between compute nodes typically used in native computational clusters [4]. The evaluation of abilities of computation clusters typically involves performing message passing interface (MPI) latency and bandwidth tests [4], [14], [15]. Huang performed a MPI latency test (Figure 3), which showed very little difference between their Xen VM and a native computer [4].

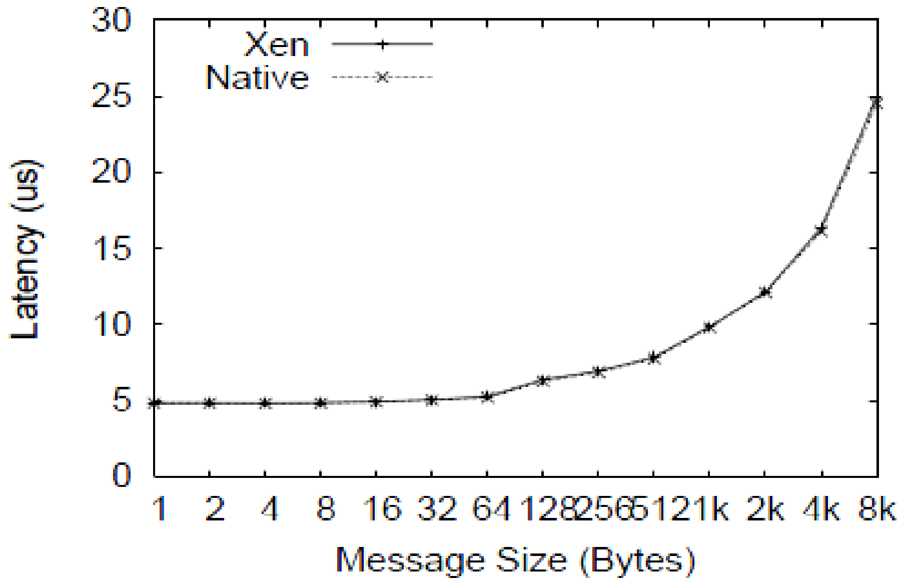



Figure 3. Results of the MPI latency test performed by Huang. This figure shows the Xen VM and the native machine perform nearly identical in the MPI latency benchmark test (from [4]).

Huang and colleagues' results are important in the further investigation of HPC applications in VMs because they show that it is possible to communicate between nodes as fast as a native machine. While the time between messages is important for HPC applications, one must also consider the importance of the size of the message, which is defined as bandwidth. Figure 4 shows the results of Huang *et al.* MPI bandwidth test, which shows "almost no difference between Xen and native environments" [4].

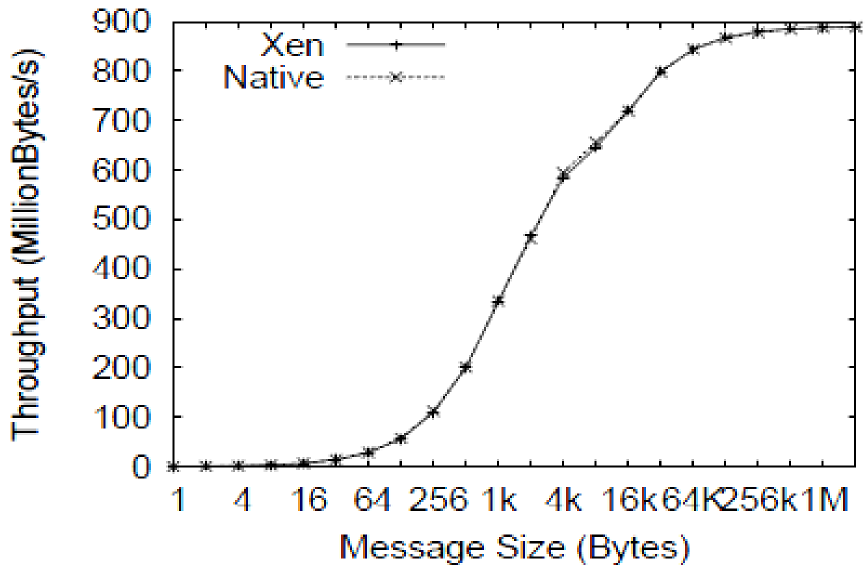



Figure 4. Results of the MPI bandwidth test performed by Huang. This figure shows very little difference between the Xen VM and the native environment. Units are missions of bytes per second (from [4]).

With the knowledge that VMs can perform at computationally similar speeds to native machines, we now need to know if it is possible to run HPC applications in a VM cloud environment. He *et al.* conducted a case study of running HPC applications in public clouds [14]. In their study, they found that virtualization (VMs) added little performance overhead and that most current (2010) public clouds are not designed for HPC applications due to network capabilities [14]. As did [4] in order to conduct the MPI benchmarks, [14] used the NAS Parallel Benchmark (NPB) from NASA, and the High Performance LINPACK. He *et al.* also added in an HPC application called the Cubed-Sphere-Finite-Volume (CSFV), which is a climate and NWP model [14].

He *et al.* chose three public clouds for their tests, the Amazon EC2 cloud with “dual-socket quad-core Intel Xeon processors E5545@2.33GHz” [14], the GoGrid Cloud eight socket quad-core Intel Xeon processors E5459@3GHz [14], and the IBM Cloud Intel Nehalem processors X5570@2.93GHz with 32 bit OS

[14]. These cloud systems were compared to the native machine benchmark results published by [16], on a native National Center for Supercomputing Applications (NCSA) dual-socket, quad-core 2.33GHz with Intel Xeon.

In He *et al.*'s first test, they tested for VM overhead by running the NPB on a single cloud server instance and compared the results to [16] as shown in Figure 5. The results show that "virtualization technology does not add significant overhead to HPC performance" [14]. This result is very important for the use of VMs in HPC applications as research by [4], [14], and [10] all write about the perceived notion of performance overhead of VMs being a driving factor in the limited use of VMs for HPC applications.

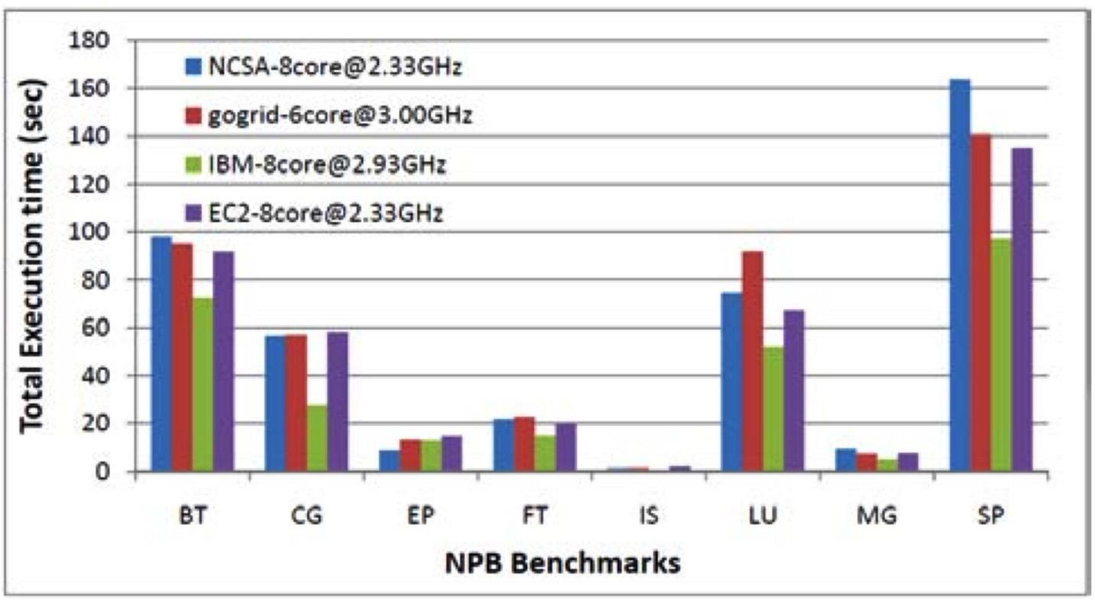



Figure 5. This figure shows the NPB benchmarks for the 3 cloud server providers from [14] against the native NCSA machine from [16] (from [14]).

He *et al.* also conducted the MPI latency test, the same as [4], for the three cloud computing services, which were compared to the results published by [16] shown in Figure 6. The tests show that the three cloud providers significantly lag behind the NCSA native environment, which [14] postulates that is due to slow networks. The variation of the MPI test for the Amazon EC2 is caused by application level message passing share the same characteristics of network level messages [14].

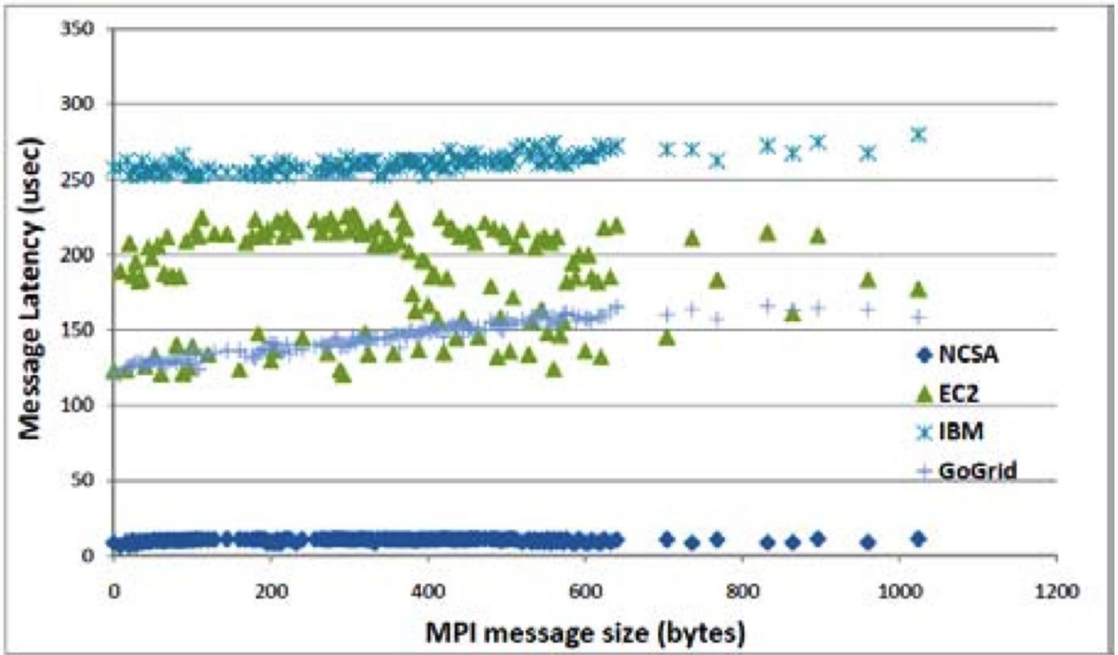



Figure 6. Results of the MPI latency test performed by [14] as compared to the results from [16]. Note the large lag between the 3 cloud providers to the NCSA native machine (from [14]).

While the results from [14] in comparison to [16] is disappointing for HPC applications in cloud computing environments, recall [4] who was able to use a virtual framework in their Xen VM to perform to par with a native machine with the high throughput interconnect InfiniBand.

The past two research efforts, [4] and [14] used modified hardware virtualization, either with Xen VM software or the cloud server VM instance provided respectively. Research conducted by Duran-Limon *et al.* conducted a study in 2011, which showed that application virtualization (sometimes called lightweight) outperformed hardware (sometimes called heavyweight) virtualization solutions such as VMware [10]. Duran-Limon *et al.* presented a D-

VTM [10], an application level framework previously mentioned in Chapter I Section B1 of this paper, which is used to run the weather research and forecasting (WRF) model in their experiments against a VMware virtualization solution of WRF. The experiment consisted of running WRF in a standard configuration between the D-VTM and VMware while measuring execution time and running those configurations with and without other processes in their 48 cores, six-node, Intel Xeon 5500 2.0GHz processor, with 12GB memory cluster [10].

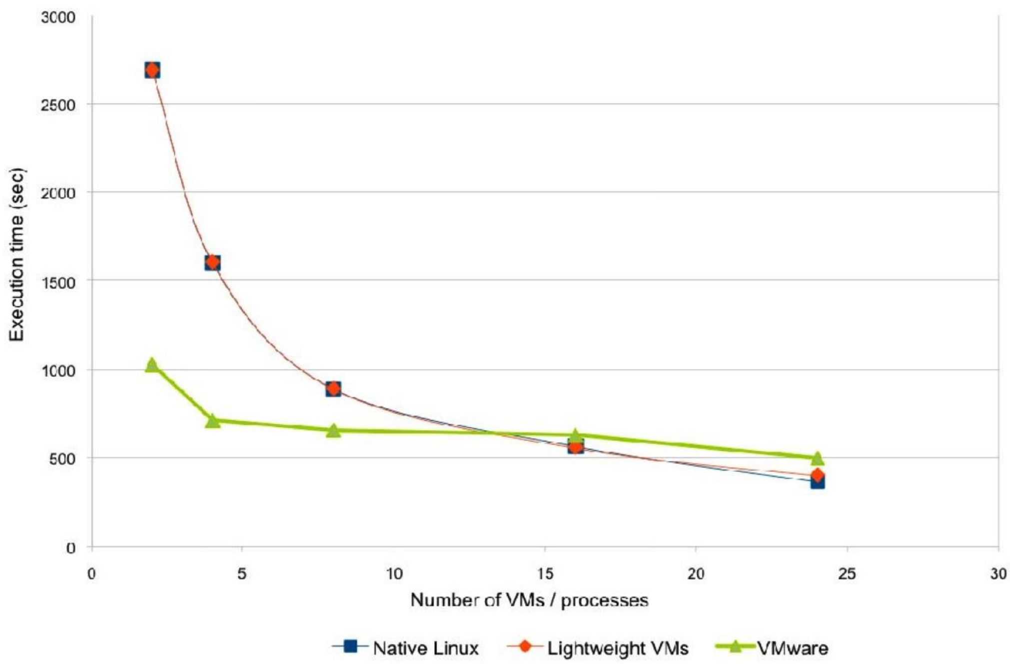
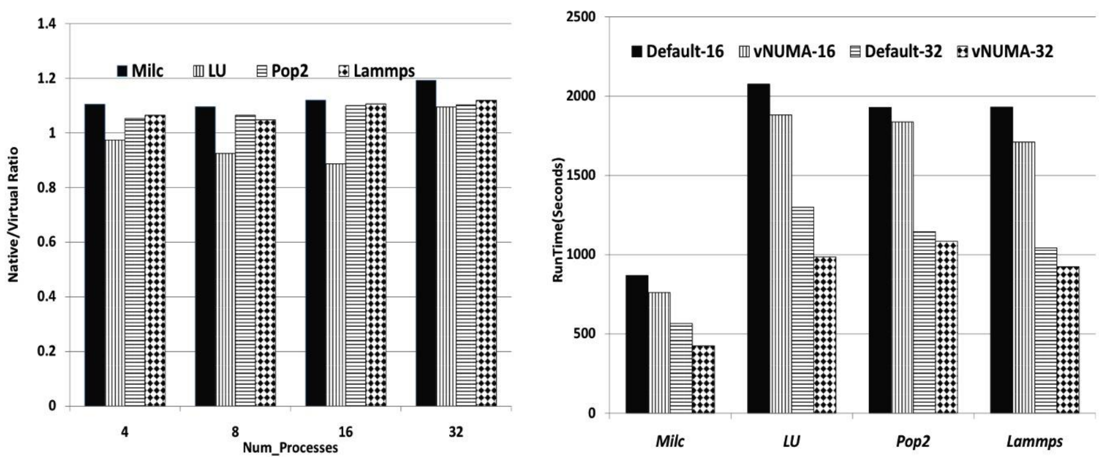



Figure 7. Shows the results from [10] experiment. As the number of VMs increased the D-VTM outperformed the VMware solution. The D-VTM performed at their native Linux comparison (from [10]).

Figure 7 shows the results from the single execution of the WRF job with no other jobs consuming the system in comparison to native Linux and VMware [10]. It should be of note that VMware initially outperforms both native and D-VTM. The research from [10] notes that this is due to the VMware instances given more CPU resources than the D-VTM and native systems.

The framework from [10] presents a good case to use an application level virtualization to overcome overhead performance. However, [4], [14], and [10] all battled performance overhead for HPC applications with modified frameworks. Is it possible to use out of the box “heavyweight” hardware virtualization for HPC workloads? A recent experiment by VMware was able to use VMware’s ESXi server to “achieve close to native performance (in some cases even 20 percent better than native) with applications from SPEC MPI and SPEC OMP benchmarks” [15]. The SPEC MPI and SPEC OMP benchmarks are the same benchmarks used by [4] and [14].




(a) Native to Virtual Runtime Ratio (b) Default vs vNUMA Runtime

Figure 8. MPI results from [15] experiment. (a) Shows overhead performance while (b) shows vNUMA advantages (from [15]).

The MPI results from [15] in Figure 8 (a) shows that “virtualization is adding little or no overhead” [15] in the experiment where 32 virtual CPUs were used on a Dell PowerEdge R910, running Red Hat Linux 5 and with 258 total GB of memory [15]. Part (b) of Figure 8 shows the advantages of exposing virtual non-uniform memory access (vNUMA) in comparison to not (default) [15]. The results from Ali *et al.* demonstrate that HPC workloads can reach native performance on “heavyweight” hardware virtualization solutions like VMware.

## C. RESEARCH SCOPE

In this study, we researched, created, tested, and evaluated the ability to run a military grade NWP model within a VM. This study focused on answering the following research questions:

### 1. Research Questions

1. How does VM performance compare to native machine performance using standard benchmark tests?
2. Can the COAMPS NWP model be run in an out of the box hardware virtualization environment?
3. What is the performance of the COAMPS NWP model in a VM in comparison to a native machine?

### 2. Thesis Organization

To answer these research questions, we developed a framework of steps that evaluates the feasibility of VM and cloud computing technologies for the use in HPC.

Chapter II provides: (a) the selection of the HPC systems, namely the VM and native machine, (b) analysis methods that were used, detailing the experimental setup, analysis, and measures of success, and (c) a summary of the methods used. Chapter III provides: (a) results from the benchmarks, (b) results from the COAMPS model run, and (c) a summary of the results. Chapter IV provides: (a) a conclusion of the key results, (b) how the results are applicable to the DoD, and topics for further research.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. HARDWARE AND METHODS

## A. SELECTION OF THE HPC SYSTEMS

### 1. Virtual Machine and Cloud Environment

The VM and cloud computing environment for this research were housed at the Naval Postgraduate School's Cloud Computing Laboratory (NPS-CCL).The NPS-CCL's resources have recently been used in studies for cloud technologies in afloat networks [7] and VM technologies in hastily formed, forward deployed networks [8].

Hardware consisted of Dell M620 dual quad-core 2.4 GHz Intel CPU, with 96 GB of RAM. The network infrastructure consisted of 10Gbps network on six hardware switches. The cloud computing software consisted of VMware vSphere ESXi 5 server. Available cloud storage consisted of one Dell EqualLogic iSCSI (Internet Small Computer System Interface) unit with multipathing enabled and one AoE unit totaling 14 TB within the 10 GBps network. The VM OS was a CentOS version 6.4. The CentOS is based on Red Hat Enterprise Linux (REHL), which was used in the native machine.

### 2. Native Machine

The native machine for this research is a REHL 6 cluster housed at the Naval Research Laboratory’s Marine Meteorology Division (NRL). This computing system is actively used for research and development of the COAMPS model.

Hardware consists of two, dual-core 2.2 GHz AMD 6174 Opteron processor login nodes with 128 GB of RAM, one Management node that has the same specifications as the login nodes except with 64 GB of RAM, 44 Computational nodes with dual-core 2.6 GHz AMD Opteron processors, and eight GB of RAM.

## B. ANALYSIS METHODS

### 1. Experimental Setup

The experiment was broken up into three parts, with each part seeking to answer the research questions raised in Chapter I, Section C1.

#### *a. Step One*

Step one seeks to answer research question one, which asks how a VM would perform in comparison to a native machine when running standard benchmark tests. We addressed this problem by performing two benchmark suites on the VM at the NPS-CCL and a native non-virtualized system at NRL. This step is broken up into two parts.

Part one of step one of the experiment will consist of running the NASA NAS Parallel Benchmarks (NPB) [17] HPC benchmark application on the VM and native machine. The NPB is “a small set of programs designed to help evaluate the performance of parallel supercomputers” [17]. NPB includes five kernels and three pseudo applications detailed in Table 1 [17]. The practice of using NPB to compare HPC systems to VMs has been completed by many researchers [4], [14], [15].

| Five Kernels | |
|---|---|
| **IS** | Integer Sort, random memory access |
| **EP** | Embarrassingly Parallel |
| **CG** | Conjugate Gradient, irregular memory access and communication |
| **MG** | Multi-Grid on a sequence of meshes, long- and short- distance communication, memory intensive |
| **FT** | Discrete 3D fast Fourier Transformation, all-to-all communication |
| **Pseudo Applications** | |
| **BT** | Block Tri-diagonal solver |
| **SP** | Scalar Penta-diagonal solver |
| **LU** | Lower-Upper Gauss-Seidel solver |

Table 1. Table contains the definitions of the five benchmark kernels and three pseudo applications used in the NPB (from [17]).

NPB breaks up the benchmarks into eight different classes (A-F, S, and W) [17]. Following the precedent set by [4] and [14], this research uses class B, standard test problems [17]. The VMware license at the NPS-CCL is limited to eight CPUs per VM. Results for the VM and native will be restricted to using four CPUs. Using four CPUs will allow all eight classes to be run as BT and SP require the number of processors to be a square number.

Part two consists of running a MPI micro-benchmark suite to examine the communication latency. The micro-benchmark program to be used is The Ohio State University (OSU) Micro-Bechmarks (OMB) version 4.2 [18]. The OMB benchmarks are similar to the tests conducted by [4]. Table 2 shows the point-to-point tests to be conducted in this research.

| OMB Point-to-Point Tests | | |
|---|---|---|
| **Executable** | **Name** | **Description** |
| osu_latency | Latency Test | Carried out in a ping-pong fashion, the sender sends a message with a certain size to the receiver. The receiver in turn sends a reply of the same size. Many iterations of the test are completed with an average one-way latency reported. |
| osu_bw | Bandwidth test | Sender sends a fixed number of back-to-back messages to the receiver. Receiver replies only after receiving all messages. Test is repeated several times with the bandwidth being calculated based on elapsed time from the first message until the reply. |
| osu_bibw | Bidirectional Bandwidth Test | Similar to the bandwidth test, however in this test both nodes involved send a fixed number of back-to-back messages and wait for the reply. Measures the maximum sustainable aggregate bandwidth by two nodes. |
| osu_mbw_mr | Multiple Bandwidth / Message Rate Test | A multi-pair of bandwidth and message rate tests to measure the aggregate uni-directional bandwidth and message rate between multiple pairs of processes. |
| osu_latency_mt | Multi-threaded Latency Test | A single sender process with multiple threads on the receiving process. Similar to the latency test, a message with a given data size is sent to the receiver and waits for a reply from the receiver process. Average one-way latency data is collected. |

Table 2. The table details the five point-to-point MPI benchmark tests from the OMB. All information within the table is from [18].

#### *b. Step Two*

Research question two asks if the COAMPS NWP can be run in a virtualization environment. Step two will address this question by running the

COAMPS NWP model within the COAMPS-OS system on the VM. COAMPS-OS (COAMPS-On Scene) is the software that manages and runs the COAMPS model [19]. COAMPS will be spun up using a static set of initialization conditions provided by the global NWP model Navy Operational Global Atmospheric Prediction System (NOGAPS) [20], which recently preceded NAVGEM. This area consists of three nested grids, shown in Figure 9, at resolutions 45, 15, and 5 km This configuration is the standard grid nesting used operationally by FNMOC for COAMPS.

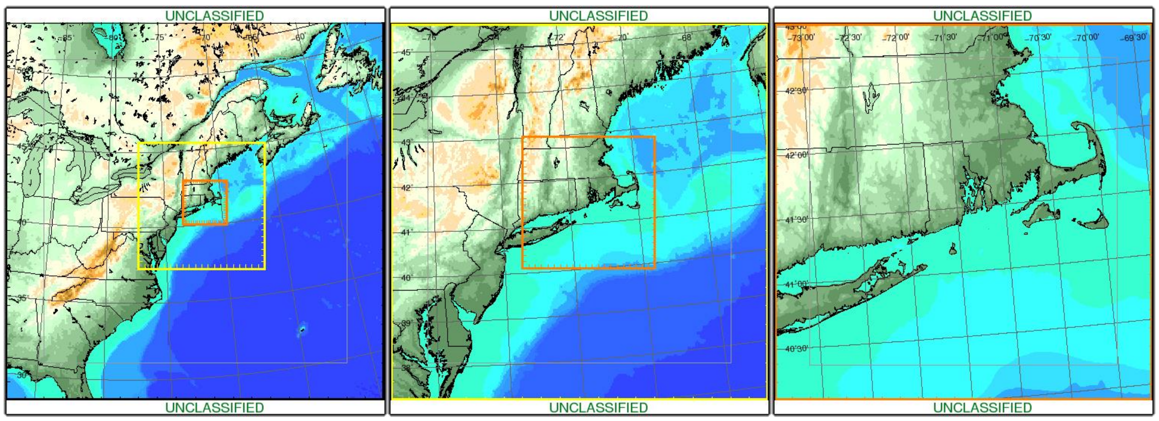



Figure 9. From left to right, 45 km, 15 km, and 5 km nest areas used by VM and native COAMPS-OS systems.

### *c. Step Three*

Step three will to answer research question three, which asks how a COAMPS NWP run would differ on a VM when compared to a native machine. In this step, we collected the model start and completion times from COAMPS-OS for both the VM and native machine. This information will be used to compare the effective run to completion time for the VM and native machine running a HPC application. Recall that the VMware license at the NPS-CCL is limited to eight CPUs per VM, both VM and native will be restricted to using six CPUs for the COAMPS model while one CPU will be reserved for post-processing and the last available CPU for logging. Our canned dataset includes enough data to run the model for two different base times, which will be done in this step.

### 2. Analysis

Statistical data from steps one and three will be collected and analyzed using Microsoft Excel.

For part one of step one, each NPB test produces an output file and within this file is “execution time in seconds,” which corresponds to the total time it took to complete the individual test. Execution times from the VM and native machine will be entered into Excel and a bar graph will be created. For this test, the larger (smaller) the bar corresponds to slower (faster) test completion times, which infer lower (higher) performance.

Part two of step one, the OMB produces an output file for each test where the output varies depending on the test run. Each of the output information will be entered into Excel and a line graph will be created for each of the five tests conducted in this research.

For part three, COAMPS-OS collects the start and completion times for the COAMPS model. For both model test runs, the total completion time will be collected and averaged for the VM and native. This information will be entered into Excel and presented as a table.

All of the datasets presented will have a *measure of success* calculated. An overall calculation will determine the successfulness of this research.

### 3. Measure of Success

For this research, the measure of success will be determined by the performance metrics in steps one and three, and the successful completion of step two.

Recalling research from Duran-Limon *et al.* where their lightweight virtualization produced a five percent overhead and “significantly” outperformed VMware [10]. If performance measurements from steps one and three perform within a five percent range between the native and VM, it will be considered *successful* for performance purposes.

## C. SUMMARY OF METHODS

This research was structured into three steps, each seeking to answer the three research questions. Step one consist of using standard benchmark programs NPB and OMB to quantitatively measure the performance of the VM and native machine. Step two involves the installation of COAMPS-OS on the VM and running the COAMPS model. Step three runs COAMPS-OS/COAMPS using a canned dataset on the VM and native machine for two base times. Data from steps one and three will be collected and analysis will be performed using Excel. A five percent measure of success range will be applied to each dataset where applicable. An overall measure of success will be calculated for the basis of determining overall research success.

THIS PAGE INTENTIONALLY LEFT BLANK

# III. RESULTS

## A. BENCHMARKS

The benchmark results from step one using NPB and OMB is organized in the following two subsections.

### 1. NPB Results

Figure 10 displays the results of the NPB test performed on the VM machine at the NPS-CCL and the native machine at NRL. Throughout all eight benchmark applications, the VM performs the tests faster than the native machine. Refer to Table 1 for details on the eight applications used in the NPB benchmarks.

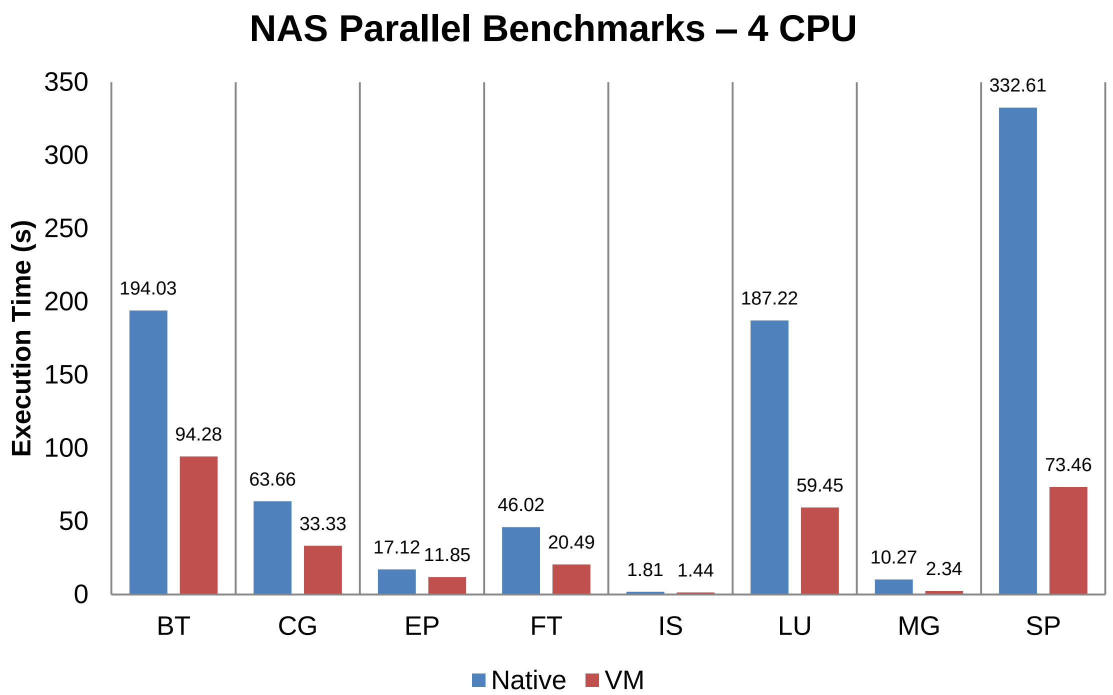



Figure 10. The results for the VM (in red) and native machine (in blue) from the NPB eight benchmark application. Refer to Table 1 for NPB application information. The higher the execution time in seconds, the slower performing the test the machine was.

### 2. OMB Results

This section contains the results from the OMB benchmark tests. Refer to Table 2 for details on the specific tests used in this research.

Results from the OMB latency test are shown in Figure 11. A key note is the change in latency that occurs at the 512 MB between the VM and native machine. At the 512 MB mark, the VM performs better than the native machine.

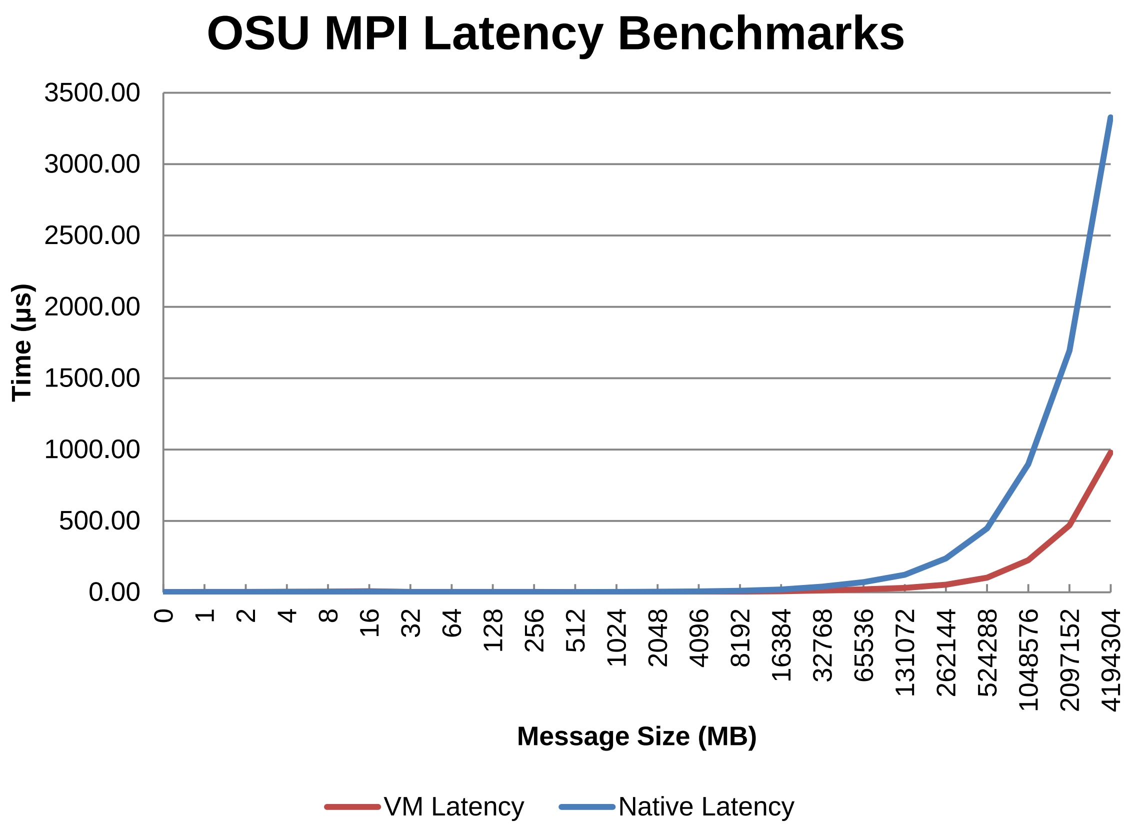



Figure 11. Chart shows the MPI latency in microseconds for the VM (in red) and native machine (in blue). Notice the change in latency at the 512 MB mark.

The results from the OMB bandwidth test are shown in Figure 12. In the beginning of the test with small message size, the VM does not fare well and falls outside the five percent threshold of the native machine. However, at 512 MB the

VM begins to fall within the five percent threshold. At 1024 MB the VM starts and continues to perform better in the latency test than the native machine.

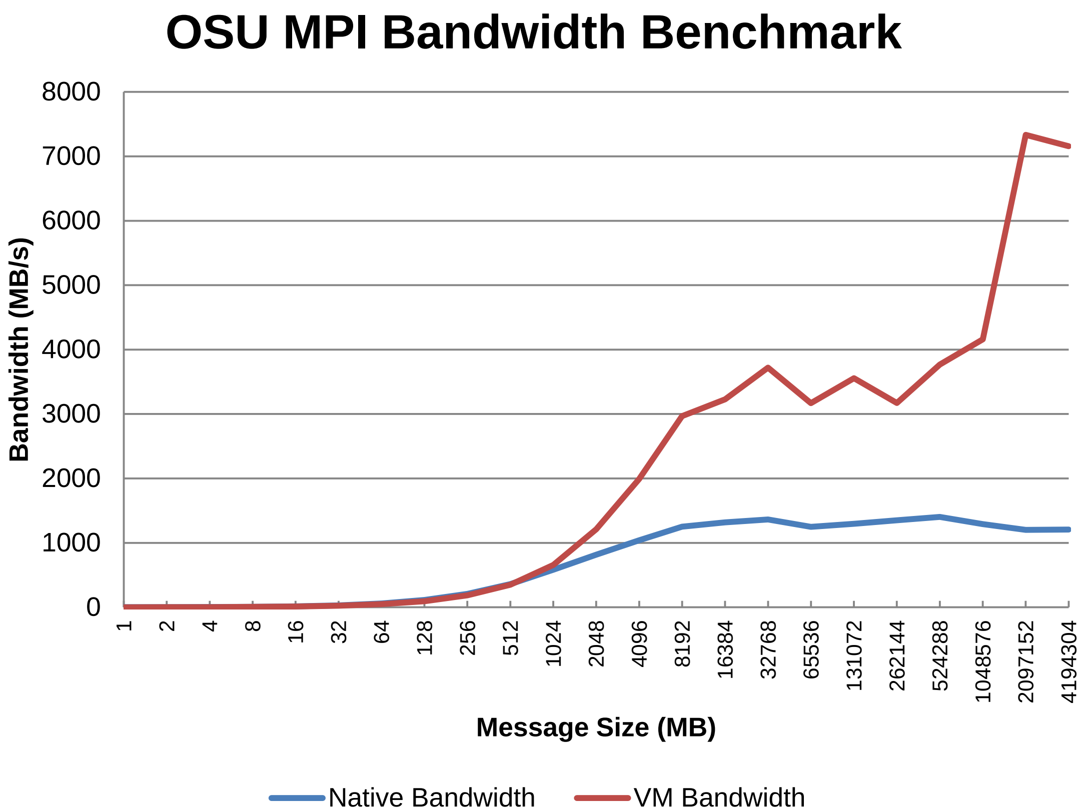



Figure 12. Chart shows the results from the OMB bandwidth benchmark. VM is in red while the native machine is in blue. Note the change in bandwidth at 512 MB.

The results from the OMB bandwidth benchmark test are shown in Figure 12. As seen in Figure 11, a change in performance occurs at the 512 MB message size. After the 512 MB mark, the VM begins performing better than the native machine.

The results from the bi-directional bandwidth benchmark test are shown in Figure 13. In this test the VM does eventually outperform the native machine but this occurs not much later than the previous tests at the 2048 MB message size.

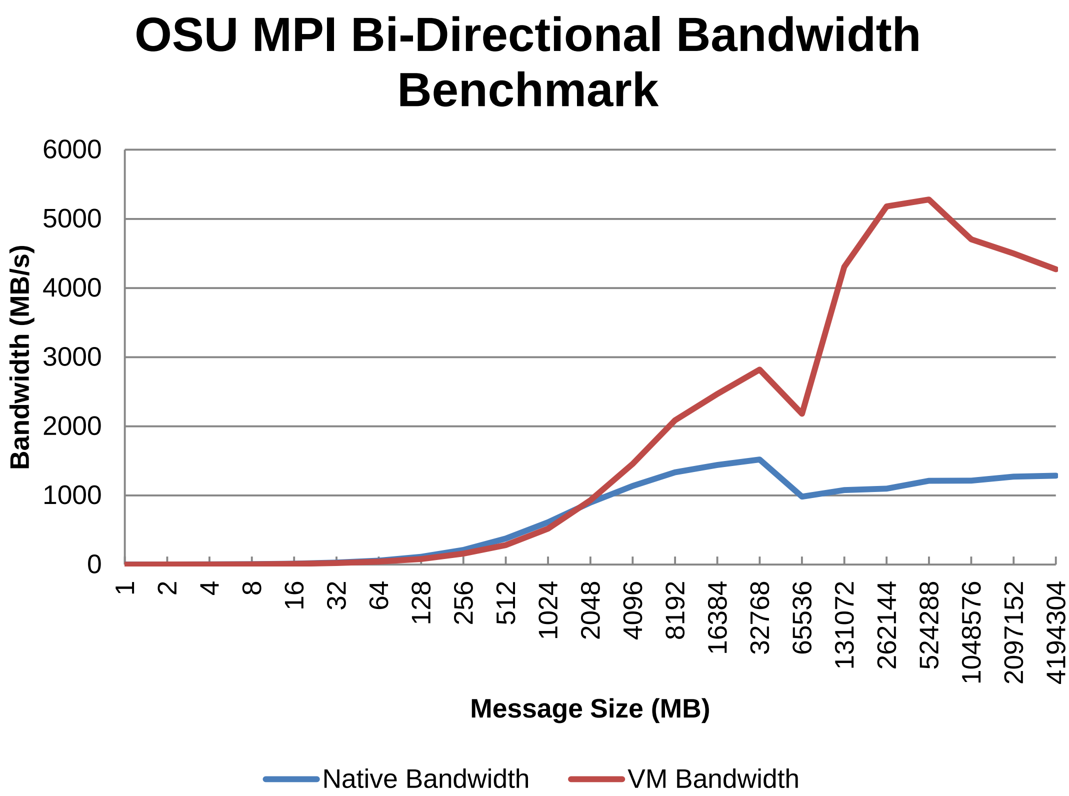



Figure 13. Chart shows the results from the OMB bi-directional bandwidth test. In this test the VM (in red) begins to outperform the native machine (in blue) at the 2048 MB message size.

Figure 14 shows the results from the multiple bandwidth/message rate OMB benchmark test. Unlike the previous benchmark tests where the VM trailed before outperforming the native machine, this test shows that the VM started out ahead in both measurements until after the 1024 MB message size mark. At the 2048 MB message size, the native machine outperforms the VM beyond the five percent threshold in the multiple bandwidth/message size tests.

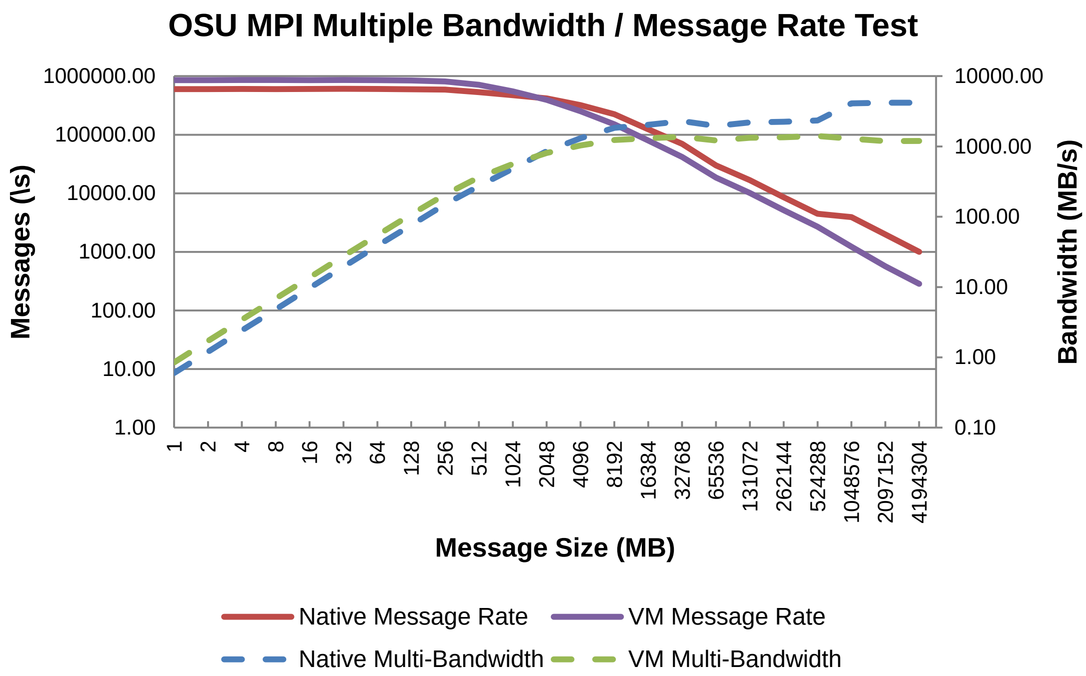



Figure 14. This chart shows the results from the OMB multiple bandwidth/message rate test. The dashed lines, blue for native and green for VM refer to the multiple bandwidth test whose units are MB/s. The solid lines, red for native and purple for VM refer to the multiple message rate test. Both messages per second and bandwidth are presented in logarithmic base 10.

Similar to the latency test shown earlier, the OMB multi-threaded latency test is shown in Figure 15. The output from this test is very similar to the latency test, including the change in latency at the 512 MB message size mark where the VM shows a lower latency than the native machine.

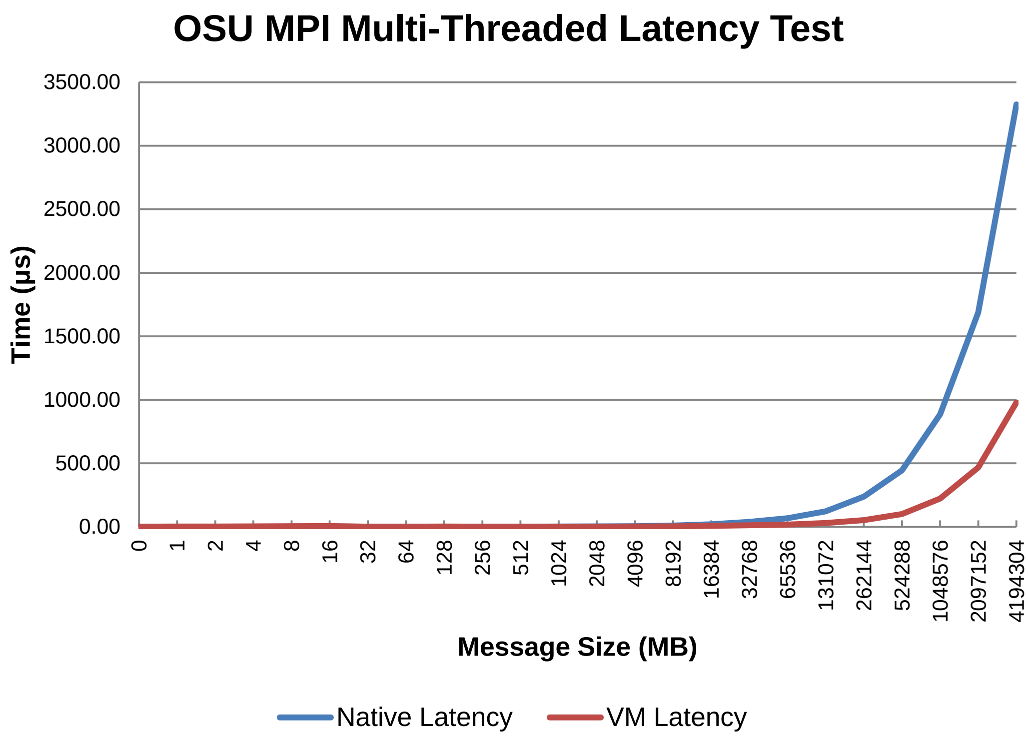



Figure 15. This chart is similar to Figure 11 except it is a multi-threaded latency test. The VM (in red) has a higher latency than the native (in blue) machine. At the 512 MB message size, the VM begins to have a lower latency than the native machine.

### 3. Summary

The overall results from both the NPB and OMB bandwidth tests show the VM tends to perform at the five percent threshold established by this research. All test runs had instances where the VM performed better than the native machine.

## B. COAMPS MODEL RUN

We successfully installed and setup COAMPS-OS (which runs the COAMPS model) over the nested domains in a VM. These domains were shown in Figure 9. After we installed COAMPS-OS, the two runs planned in Chapter II were initiated. The results from the two test runs are shown in Figure 16.

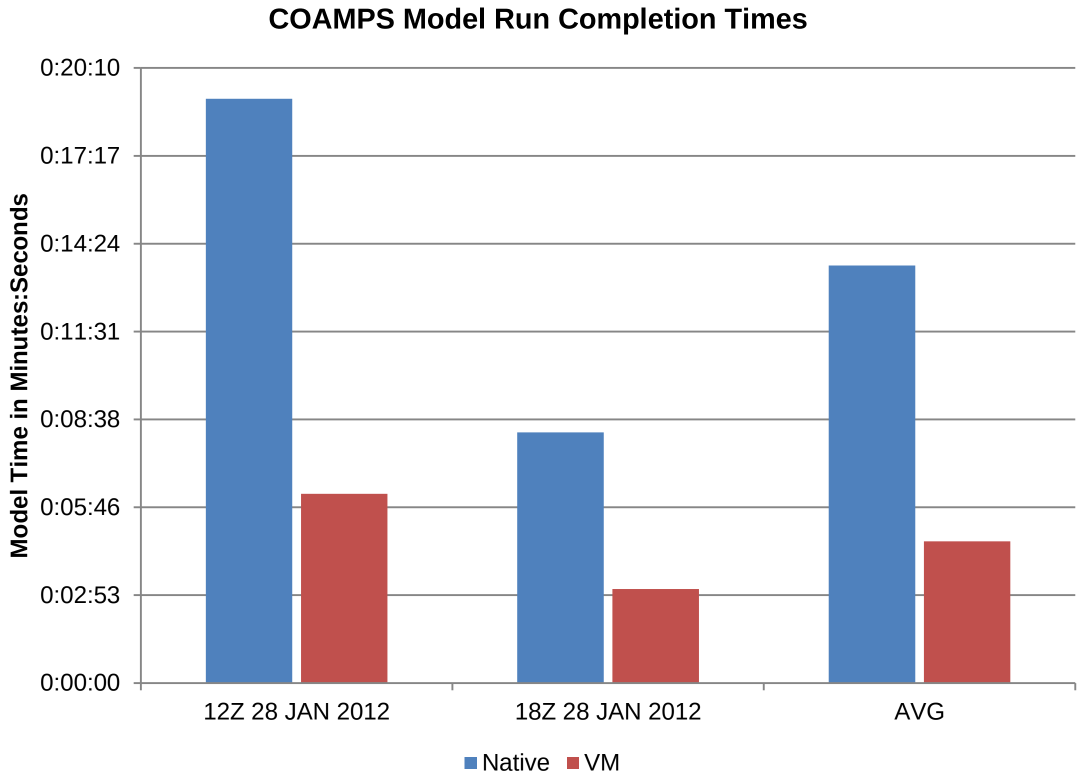



Figure 16. This bar graph shows the results from the two COAMPS runs on the native machine (in blue) and the VM (in red). The third column of bars shows the average run time for both base times.

The 12Z COAMPS model run includes more observations and is a cold start. A cold start is defined when a NWP model needs to be fully initialized before beginning a forecast. A warm start, which was the 18Z run in this test, uses the forecast fields from the previous model run (12Z in this case) to quickly spin up a forecast.

Figure 16 shows how the large difference in the completion time for running the 12Z COAMPS between the native machine and the VM. The native machine took 19 minutes and nine seconds to complete while the VM took six minutes and 12 seconds. The 18Z model run, a warm start, took a shorter amount of time to complete but the VM still outperformed the native machine. The native machine took eight minutes and 13 seconds while the VM took three minutes and five seconds.

## C. SUMMARY AND DISCUSSION OF RESULTS

The results presented in this chapter present a quantitatively look at the feasibility of VM and cloud computing technologies for NWP and possibly other HPC software applications. In step one of this research, we used standard benchmarking software suites to objectively compare a VM to a reasonably powerful native machine. In step three we ran the COAMPS NWP model on a VM and native machine using a static dataset and the exact model configuration between the two machines.

The results in step one shows that the VM can perform at the level of a native machine. In addition, all benchmark test cases at some point performed better than the five percent threshold set in this study and beyond.

Step three results show that the VM can run a HPC application under set conditions but that is also outperforms a native machine under the same conditions.

This results in this study show that a VM can be a viable environment to run a HPC application. Also the results show that this VM can reside in a cloud infrastructure and run HPC applications.

# IV. CONCLUSIONS

## A. KEY RESULTS

This research focused on quantitatively analyzing the feasibility of running an HPC application in a VM on a cloud computing infrastructure. The purpose of this research was to investigate the practicality of running a numerical weather model (i.e., COAMPS) in a VM rather than a native machine in order to determine if a VM could be a viable option.

Using standard benchmarks, we were able to objectively compare the VM with the native machine. The NPB test was chosen in this study because it was used in prior comparisons between VMs and native machines, as discussed in Chapter II. The OMB test was chosen because it is a maintained project at OSU, which tests the systems similar to past research. When we conducted the NPB test, we found the VM had a shorter execution time in all instances when compared to the native machine. When we conducted the OMB test, the VM performed at least five percent better than the native machine at some point in time. In all OMB tests, the VM at some point performed better than the native machine.

After we conducted the benchmark tests, we installed COAMPS-OS on the VM machine within the cloud computing infrastructure at the NPS-CCL. Using the COAMPS-OS software, we were able to run the COAMPS NWP on the VM and native machine. We found that the VM performed better than the native machine within controlled setting.

The results of both the benchmark tests and the COAMPS model run show that a VM in a cloud infrastructure is a practical runtime environment. While our results showed that a VM in a cloud infrastructure is a useful option, there were instances where the VM was not a better option. For example, the

OMB test had instances where the native machine performed better than the VM. The purpose of this study was to show that the COAMPS model can be run on a VM.

## B. APPLICABILITY TO DOD

COAMPS is a regional NWP that is run a FNMOC, which utilizes HPC clusters to run the model, as discussed in Chapter I. FNMOC is the only agency that manages and run COAMPS for the U.S. Navy for operational purposes. While FNMOC has many advantages, the fact that they are the only center that provides COAMPS output to the Fleet leaves them vulnerable to down time. Having the capability to run COAMPS on a VM provides high mobility and high scalability.

The DoD CIO has already begun to investigate the value of cloud computing for operational purposes, which was outlined in the DoD *Cloud Computing Strategy* in 2012 and discussed in Chapter I. There were many prior studies that already investigated the option of using VM and cloud computing technologies to run HPC applications (outlined in Chapter I).

The results of our study show that a VM in a cloud computing infrastructure can be better than a native machine. This shows the feasibility of the U.S. Navy using a VM to run COAMPS for the forward deployed on ships and as an option to maintain COOP during a major crisis.

## C. TOPICS FOR FURTHER RESEARCH

While this research shows the viability of using a VM to run COAMPS, additional research is needed in order to continue this development. Below are recommendations for further research.

1. Since the NPS-CCL VMware license restricts the amount of CPUs allocated to a VM, the performance of a VM to a native machine with a larger amount of CPUs should be further investigated.

2. FNMOC runs multiple COAMPS regions at the same time. Our research only examined one COAMPS region. Running multiple COAMPS regions in a VM at the same time should be further investigated.

3. In operations, FNMOC conducts data assimilation and post processing as part of the COAMPS run cycle. This research used a static data set in order to run COAMPS without including the rest of the NWP run cycle. Further research should be conducted to include the entire NWP cycle on a VM.

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


[1] R. M. Hodur, “The Naval Research Laboratory's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS),” *Monthly Weather Review,* vol. 125, no. 7, p. 1414–1430, 1996.

[2] United States Navy, “Fleet Numerical Meteorology & Oceanography Center Command Overview,” Fleet Numerical Meteorology & Oceanography Center, Monterey, 2012.

[3] White House, “National Continuity Policy,” White House, Washington, DC, 2007.

[4] W. Huang *et al*., “A case for high performance computing with virtual machines,” in *ICS '06 Proceedings of the 20th Annual International Conference on Supercomputing*, Queensland, Australia, 2006.

[5] Department of Defense, “Cloud Computing Strategy,” Chief Information Officer, Washington DC, 2012.

[6] J. Cook, J. McCarthy, and T. Tsui, “Nowcast for the Next Generation Navy,” in *Presentations of the Battlespace Atmospheric and Cloud Impacts on Military Operations Conference*, Fort Collins, 2000.

[7] S. E. Gillette, “Cloud computing and virtual desktop infrastructures in afloat environments,” M.S. thesis, Naval Postgraduate School, Monterey, 2012.

[8] A. Barreto, “Integration of virtual machine technologies into hastily formed networks in support of humanitarian relief and disaster recovery missions,” Naval Postgraduate School, Monterey, 2011.

[9] Oracle. (April 2013). “Oracle VM User's Guide for Release 3.2.1.” [Online]. Available: http://docs.oracle.com/cd/E35328_01/E35332/html/index.html. [Accessed 5 October 2013].

[10] H. A. Duran-Limon *et al*., “Using Lightweight Virtual Machines to Run High Performance Computing Application: The Case of the Weather Research and Forecasting Model,” *2011 Fourth IEEE International Conference Utility and Cloud Computing (UCC),* pp. 146–153, 2011.

[11] National Institute of Standards and Technology, “The NIST Definition of Cloud Computing,” U.S. Department of Commerce, Gaithersburg, VA, 2011.

[12] *Wikimedia Commons*. (August 2012). “Cloud computing layers.” [Online]. Available: https://commons.wikimedia.org/wiki/File:Cloud_computing_layers.png. [Accessed 16 06 2013].

[13] A. Black, “High Throughput Computing: An Interview with Miron Livny,” *HPCwire,* 27 June 1997.

[14] Q. He, S. Zhou, B. Kobler and T. McGlynn, “Case study for running HPC applications in public clouds,” in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, New York, 2010.

[15] Q. Ali, V. Kiriansky, J. Simons, and P. Zaroo, “Performance Evaluation of HPC Benchmarks on VMware’s ESXi Server,” in *5th Workshop on System-level Virtualization for High Performance Computing (HPCVirt 2011)*, Bordeaux, France, 2011.

[16] E. Walker, “Benchmarking Amazon EC2 for High-Performance Scientific Computing,” *;login:,* vol. 33, no. 5, pp. 18–23, 2008.

[17] National Aeronautics and Space Administration, “NAS Parallel Benchmarks,” 19 March 2012. [Online]. Available: http://www.nas.nasa.gov/publications/npb.html. [Accessed 2 November 2013].

[18] The Ohio State University. (August 2013). “Benchmarks,” The Ohio State University's Network-Based Computing Laboratory. [Online]. Available: http://mvapich.cse.ohio-state.edu/benchmarks/.

[19] J. Cook *et al*., “The U.S. Navy's on-demand, coupled, mesoscale, data assimilation and prediction system,” in *22nd Conference on Weather Analysis and Forecasting/18th Conference on Numerical Weather Prediction*, Park City, UT, 2007.

[20] T. F. Hogan and T. E. Rosmond, “The Description of the Navy Operational Global Atmospheric Prediction System's Spectral Forecast Model,” *Monthly Weather Review,* vol. 119, no. 8, pp. 1786–1815, 1991.

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California