UNIVERSITATEA POLITEHNICA BUCUREŞTI
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

PROIECT DE DIPLOMĂ

Analiza sistemelor cu maşini virtuale multiple, cu aplicaţie in cloud

Coordonator ştiinţific:
SL. Mihail Ionescu

Absolvent:
Cristian Chilipirea

BUCUREŞTI

2012

POLITECHNICA UNIVERSITY OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE DEPARTMENT

DIPLOMA PROJECT

Multiple Virtual Machine Systems Analysis, For Cloud Computing

Thesis supervisor:
SL. Mihail Ionescu

Author:
Cristian Chilipirea

BUCHAREST

2012

Table of Contents

Notations and Abbreviations

OS - Operating system
CPU - Central Processing Unit
GPU - Graphical Processing Unit
VM -Virtual Machine
VMM - Virtual Machine Manager
ISP - Internet Service Provider
SP - Service Pack
DSL - Damn Small Linux
LTS - Long Term Support
GUI - Graphical User Interface
USB - Universal Serial Bus
CD - Compact Disk
RAM - Random Access Memory
FPGA - Field Programmable Gate Array
DSL - Damn Small Linux
PC - Personal Computer

Abstract

Cloud computing represents a powerful tool we have at our disposal today. Based on virtual machine technology cloud providers seem to act like a "Swiss Army Knife", and choosing the right tool for the job becomes an increasingly difficult task.

There isn't, and probably never will be, a one fits all alternative when it comes to computing and with more and more companies moving to the cloud, the necessity for better ways to determine the correct configuration rises.

Keywords: cloud, optimization, virtual machine, hypervisor

1. Introduction

This paper discusses the advantages of using certain virtualization environments on a cloud computer infrastructure and tries to find out which Operating System might be preferred for the end user.

Cloud providers tend to grow their numbers every day and most of them use different Virtualization methods and different operating systems on different hardware with different tools provided, and of course all of them come at different prices. The user is left with a huge pool of options and no real way to determine the best solution for their needs. And all these options can become a burden, "we have too many choices, too many decisions, too little time to do what is really important" [1] (Paradox Of Choice: Why More is Less – Barry Schwartz – Eccon 2004).

The problem of virtualization itself is tackled in different ways by different tools, from simple desktop applications to Enterprise Hypervisors. And each tool has its own advantages. VMware ESXi tends to be a bit more light considering it doesn't require a Host Operating System, Hyper-V has a better integration with Windows Tools, and Xen offers a free Open Source alternative. And these are just a few examples of all the virtualization methods at our disposal today.

1.1 The hardware

Just like in a simple environment, hardware represents the base of everything we do. At first look things seem simple: the better the hardware the better the performance, and even though usually true in a Virtual Environment, this tends to be a harder choice. CPU frequency cannot be the sole determining factor. Today the number of cores is more relevant, in Cloud Computing even more so, considering one can theoretically give a CPU core to every VM and never have them share.

Equally important advantages are brought by actual virtualization features included in the CPUs like AMD-V or Intel VT-x, which offer a huge boost to guest machines.

We find it also important to mention CPU cache as a important factor in VM efficiency and the CPU-Memory Interconnect Bus.

Depending of the user's needs, other important benefits could be given by the storage system used and by the Network Connection offered. The Network Connection can be influenced by a lot of factors like the hardware used, the network architecture, and even the bandwidth the cloud provider has with other ISPs.

Of course the end user could also have special needs, for example a need for high performance GPU resources. This special needs are starting to be considered by the cloud providers, like Amazon who now offers Cluster GPU Instances.

1.2 Virtual Machine Managers

We can now find a large number of Virtual Machine Mangers, each having different needs and advantages. In this paper, we were interested in Xen, VMware ESXi and Hyper-V, because they power some of the biggest Cloud Providers in the world.

Xen powers Amazon EC2 and Linode.

Hyper-V similar with Microsoft Azure.

VMware powers Dell vCloud and HP Cloud System.

Of course there are a lot of other cloud providers we haven't explicitly mentioned and are probably equally important as the ones we did. And there are also other Hypervisors and Virtual

Machine Systems that could be used. We would like to mention systems like Qemu which don't only offer a virtual machine support but they do this by emulating a different CPU then the one available. This kind of technology could later bring x86 operating systems to a cloud made only by ARM processors or the other way around.

1.3 Guest Operating Systems

There are currently many choices for Operating Systems that one could run inside a Virtual Machine, there is Windows, there is Linux with the many flavors it has, and there are also other products like BSD, Android and iOS.

In this paper we have concentrated on a small number of Linux Distributions. We choose Linux because it is free, we have access to the source code to see exactly what features it brings and what it doesn't. Also all the major cloud providers we mentioned have some kind of Linux support, usually in more distributions.

Because Linux comes in so many flavors choosing the right one is a hustle in itself. One might need a specific GUI or a specific feature that only a certain distribution provides. There are also efficiency issues that one has to consider in case one needs the machine for a high performance application.

We have decided to use the following Linux Distributions: Ubuntu 12.04 Server Amd64 because it is probably the most popular Linux Distribution at this moment, CentOS 6.2 x86_64 because it is an enterprise distribution that promises to bring enhanced security and stability, and Damn Small Linux 4.4.10 because we wanted to see how a very light distribution and kernel would compare to the others. It is important to mention that we found no cloud provider that actually offers explicit support for Damn Small Linux but we still consider it important as it might provide an idea of how a mini-OS could bring improvements to the cloud.

1.4 Other important considerations

Probably the most important factor in deciding what system to use is not its efficiency or what features it offers but the actual needs of the project and the tools one already has. It is rather hard to choose Windows if all the tools are already configured to run smoothly on Linux, or the other way around.

Another important factor is the experience the people have with the system; forcing different tools, even if they are more efficient in the long run, can bring delays big enough so that the project does not matter anymore.

This is probably the reason why Clouds offer such a large pool of options, to fit the needs of any possible client they might have, and making it simple for the client to migrate to the cloud without changing important parts of the project. Ultimately clouds need to be as accessible as possible.

Stability and proximity are other important features but these can only be measured per individual clouds and they require a really long term analysis of the entire infrastructure. We know now, that no matter how stable a system seems a small bug can bring down the entire cloud. This was made obvious when Azure was taken down for 12 hours because of a leap year bug.

Cloud security is an issue that still needs to be studied and considering no system is ever truly secure the option of migrating to the cloud can be a non-existing one for some.

2. Hardware

Hardware is an important part of obtaining efficiency in any computational system, but it's also the most expensive and difficult to replace. Unlike the software used hardware needs to be picked correctly from the start. Luckily when one is using a cloud the difficult task of choosing or replacing hardware is left at the hand of the cloud provider. Getting rid of the hassle of managing hardware could actually be a key factor in the migration towards the cloud.

Hardware could also influence one's decision when choosing a service in the cloud. The CPU frequency, the number of cores, available memory, storage capacity and bandwidth are important factors, but they can all be changed, according to one's, needs at any time. These factors are also the main parts that influence the price of the virtual machine one gets in the cloud. Another factor is the operating system used in case the OS license is not free.

2.1 Hardware used in the tests

All the tests were run on the following PC:
AMD Phenom II X2 545 Processor 3.00 GHz
MSI 770-C45 Motherboard
4GB DDR3 RAM
USB Stick (VMware ESXi System)
Maxtor 200GB PATA133 Hard Drive (Windows Server 2008 + Hyper-V)

The networking:
Realtek PCIe GBF (On the main PC)
Lynksys WRT54GL Wireless Router
Both PCs were connected throw Ethernet wire to the router
All the networking tests were run with iperf between the 2 PCs

Storage System for VMware:

VMware ESXi requires a SCSI hard driver for storage, as we could not provide one we used a separate PC with a NFS server. In the tests we ran we did not test boot up speed or anything that could be affected by slow storage.

We used a secondary PC with an external ASUS LEATHER II 500GB hard drive. This PC was running Debian from a USB stick and had an NFS servers.

2.2 Advantages introduced by AMD-V

The processor we used, AMD Phenom 545, comes with AMD-V support and this is one of the main reasons we used this PC to run the Virtual Machines and the Hypervisors. AMD-V is the virtualization solution offered by AMD for a large number of their available processors. It offers hardware support for Virtual Machines and it fills an important role in any system that requires virtualization. "And like any other type of hardware-assisted virtualization technology, AMD-V is also capable to make the operating system (OS) have direct access to necessary resources without any OS adjustments pr emulation" [2] (Virtualization 100 Success Secrets 100 Most asked questions on Server and Desktop Virtualization, Thinapp Software, SAN, Windows and Vista Applications – Michael James - Lulu.com, 2008 - Pg. 134)

AMD-V is supported in Xen, VMware ESXi and Hyper-V and Citrix, VMware, Microsoft are all partners with AMD Virtualization.

The features introduced with AMD-V can be seen in Table 1. (All the data is taken directly from the amd.com website.)

Table 1

AMD-V

Features	Benefits
Virtualization extensions to the x86 instruction set	Enables software to more efficiently create virtual machines so that multiple operating systems and their applications can run simultaneously on the same computer.
Tagged TLB	Hardware features that facilitate efficient switching between virtual machines for better application responsiveness.
Rapid Virtualization Indexing (RVI)	Helps accelerate the performance of many virtualized applications by enabling hardware-based virtual machine memory management.
AMD-V™ Extended Migration	Helps virtualization software with live migrations of virtual machines between all available AMD Opteron processor generations.
I/O Virtualization	Enables direct device access by a virtual machine, bypassing the hypervisor for improved application performance and improved isolation of virtual machines for increased integrity and security.

Of course, other processors also have hardware-supported virtualization, like the Intel VT-x. These are used, for example, by Linode, which offer Intel Xeon L5520 CPUs.

3. Virtual Machine Systems

Having the required hardware to run our virtual machines is not by itself sufficient: we still need software to manage the virtual machines and decide in a fair way which VM gets access what resource and when. We need a hypervisor and when it comes to hypervisors we have a few from which to pick.

We have 2 main types of hypervisors, native and hosted. The native one runs directly on the Hardware while the hosted runs on top of a OS that runs on the hardware. There can be hybrids, like in the case of Xen, which requires one Guest to act as Domain 0 so it can run its administration routines on it. The difference between the 2 types of Hypervisors is made clear in Fig. 1.

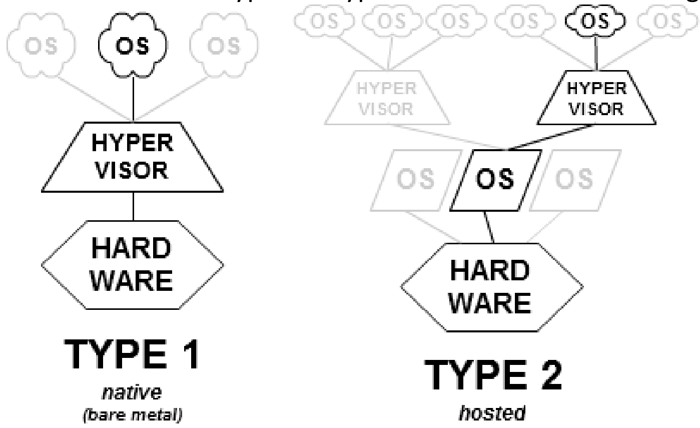

Fig. 1. Hypervisor (source Wikipedia)

3.1 VMware ESXi

VMware ESXi is a powerful hypervisor that supplements the array of Virtualization solutions from VMware. It is constructed especially to be used in a cloud environment and it is probably the most native Hypervisor one can find. Unlike the others ESXi doesn't require any other OS to run, it just installs and runs directly on the hardware. This gives it a bit of efficiency edge compared to the other products available.

It also integrates with VMware Tools which offers a variety of drivers to increase efficiency and pass messages from the guest OS to the hypervisor, execute clean shut down and restart commands, handles the mouse cursor, synchronize time and a lot of other functions that help the Guest system be simpler to use and more efficient. "VMware Tools is a suite of utilities that enhances the performance of the virtual machine's guest operating system and improves management of the virtual machine." (VMware.com website)

One can see that ESXi is an enterprise solution by its system requirements, for storage it needs a hard drive connected to it via SCSI or Fiber-Wire, or an NFS drive somewhere on the network. This and the fact that it offers no interface on the host machine other than to check logs or restart/shutdown the machine makes it almost impossible to run with under 3 computers. One for the VMware ESXi machine, one for a data store and one to remotely connect to the host and manage it and the virtual machines throw a software called vSphere Client.

As one can see in figure 2, VMware ESXi comes with a kernel that offers support for managing and monitoring the entire system. The management interface itself is done through vSphere Client and is rather simple and user friendly. The client also makes it possible to connect to any guest OS and work with it as if it were the computer one is actually using. In the client interface, you can manage anything from the virtual machines, to setting up different data stores, to keeping a close eye on the resources being used.

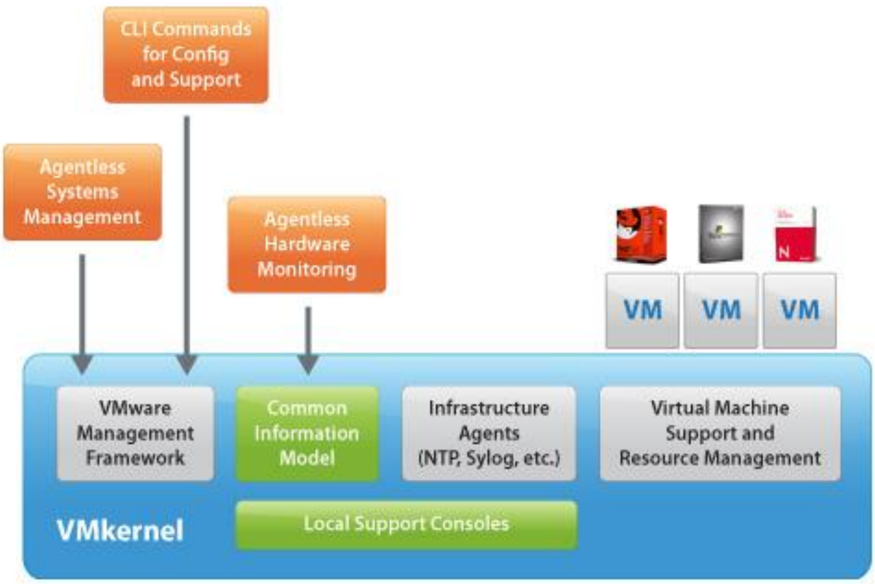

Fig. 2 VMware ESXi architecture (vmware.com website)

3.3 Hyper-V

Hyper-V is a hypervisor offered by Microsoft that runs as a role for Windows Server. It is also a native hypervisor even thou it requires Windows Server to work. In our tests we used it with Windows Server 2008 R2 SP1. Just like VMware ESXi it offers a simple user friendly interface that is integrated directly into the Windows Manager. It makes it simple to create virtual machines and control all of their details.

Like ESXi, Hyper-V offers support to fine tune the virtual machines, it can set up what percentage of processor each guest can use and what amount of memory. It can also set up the memory to be delivered dynamically, as needed.

The architecture of the system is in figure 3. In it one can see clearly why Hyper-V is a native hypervisor and how it requires a main OS to work. The Hypervisor manages the memory and has a scheduler for the CPU and deals with interrupts. It also handles hypercalls which is very important because without WinHV – Windows Hypervisor Interface Library - and LinuxHV – Linux Hypervisor Interface Library - a guest cannot run on Hyper-V. The lack of LinuxHV support is what prevented us from running DSL on Hyper-V. This need for a Hypervisor Interface Library limits the number of OSs one can use with Hyper-V. On the other hand Windows XP, or newer, and Linux kernels newer then 2.6 can run as a guest for Hyper-V without any problems.

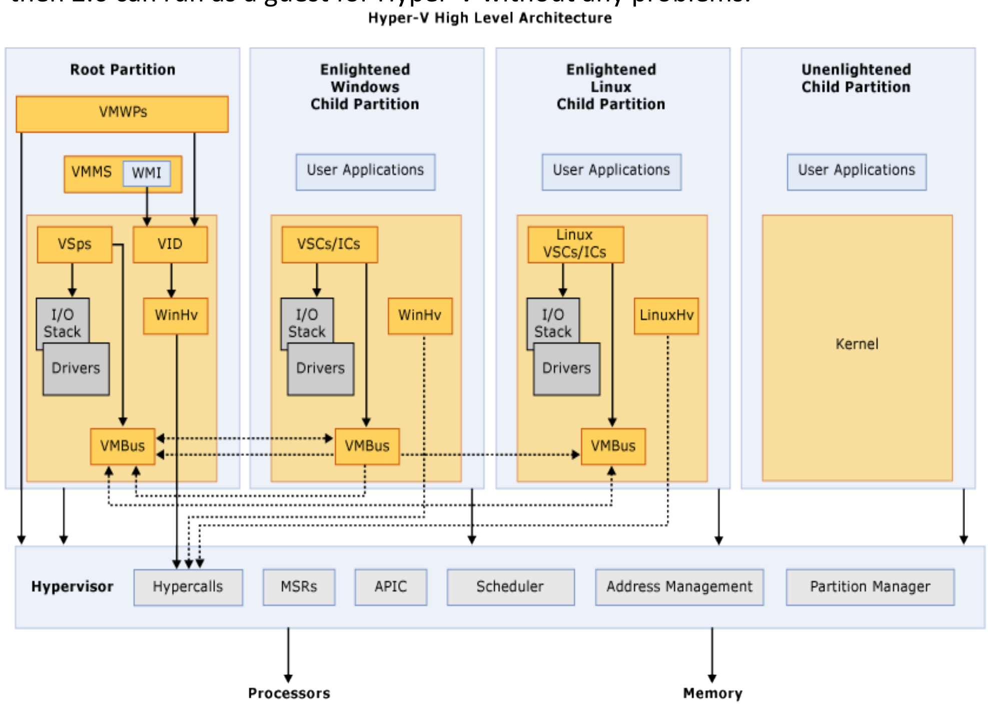

Fig. 3 Hyper-V architecture (msdn.microsoft.com website)

Regarding security there is something worth mentioning about Hyper-V: It has "**No third-party-code.** The hypervisor included as part of Hyper-V doesn't include any third-party code. Other hypervisors include device drivers or other drivers that were submitted by third parties." [6] (Windows Server 2008 Hyper-V Insiders guide to Microsoft's Hypervisor – John Kelbley – Ed. John Wiley & Sons, 2011 – Chapter 5) This ensures that all the software connected to Hyper-V is controlled by Microsoft.

3.3 Xen

Xen is an Open Source Hypervisor and we feel it worth mentioning because it is used by top cloud providers like Amazon and Linode. Xen has a similar architecture to Hyper-V, the only difference is that it uses a Linux Kernel as a main OS instead of Windows. There have been attempts at using Windows as a host for Xen but the results were not made public.

A lot of Cloud systems are constructed around Xen like XenServer from Citrix or Oracle VM – even Amazon probably uses a modified version. It is important to mention that Qemu integrates itself with Xen offering Xen the power to support a wider range of OSs for different physical architectures.

Xen offers 2 ways of creating a virtual system:

Hardware Assisted Virtualization, which allows the system to run unmodified guests just like VMware ESXi does, but to do so it requires that the CPU provides hardware support like AMD-V or Intel VT-x. This way Xen can run anything from Linux to Windows, and if integrated with Qemu even others.

Paravirtualization. "There is no hardware emulation. The operating system that runs on a guest needs to be a modified version that is aware of the fact that it is running inside a hypervisor." [7] (Xen Virtualization – Prabhkar Chaganti - Packt Publishing Ltd, 2007 – Chapter 1). This has its benefits because the modified systems are optimized for this particular use case, but it has the downside of only being able to run a small number of Linux Distributions.

The Xen architecture seems rather similar to Hyper-Vs, as we can see in Fig. 4

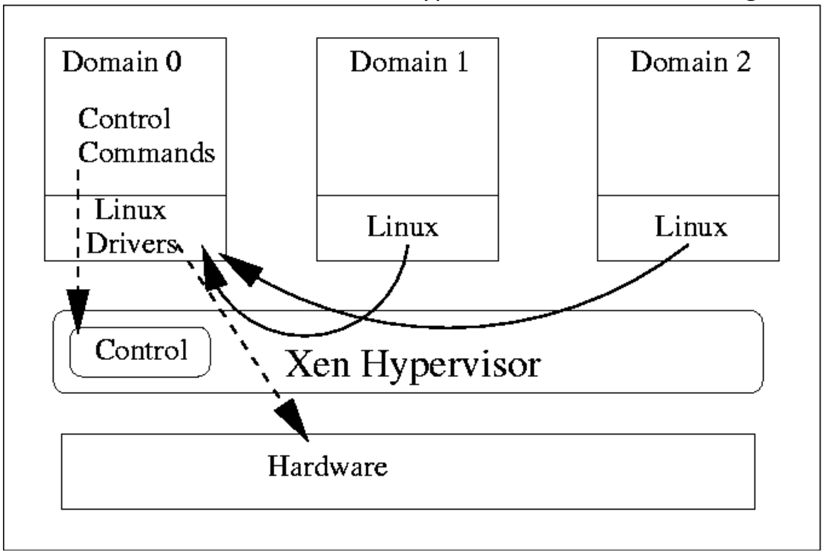

Fig. 4 Xen Architecture (veillard.com website)

Unfortunately, because Xen Hypervisor is tightly connected with the Domain 0 OS, we were unable to properly install and run it on our hardware, due to some driver issues. After trying a number of versions including Xen on Debian and on Ubuntu, and even Citrix's Xen Server, we were unsuccessful at starting any guest OSs.

We did test Xen on the Linode cloud infrastructure where we ran some extensive tests with both Ubuntu and CentOS. Both OSs being provided by the Linode team specifically configured to run best on their servers.

4. Guest Operating Systems

"It is hard to pin down what an operating system is other than saying it is the software that runs in kernel mode – and even that is not always true. Part of the problem is that the operating systems perform two basically unrelated functions: providing application programmers (and application programs, naturally) a clean abstract set of resources instead of a messy hardware ones and managing these hardware resources." [9] (Modern Operating Systems – Andrew S. Tanenbaum – 3rd Edition – Pg. 3)

Looking at OSs from the perspective given to us in "Modern Operating Systems", we notice that Virtual Machines Managers are to Operating Systems as Operating Systems are for programs. Of course they serve a different purpose; While the OS tries to offer a unified, simple way to access the hardware, the VMM tries to just split the hardware in a fair way.

Choosing the guest operating system is probably the most critical decision. A bad choice will severely impact the productivity of the end user. Previous experience with the OS is far more important than a set of features. In the tests we run we decided to use 3 Linux based OSs. We made this choice because they were free and openly available.

Choosing Linux as a kernel is not enough, we had to choose a distribution. Ubuntu was an obvious choice because of how widely used it is. Popularity brings a few advantages, it makes it more probable that new cloud users already know how to use this OS, and because more people use it and more people develop for it, it has better and more up to date modules and programs that it supports.

Even though Ubuntu is popular, we also wanted to test a distribution better suited for enterprise environment. We found that in an enterprise environment CentOS is similarly popular to Ubuntu, this made it an obvious choice for us.

Our last option was given by the idea of using a light OS in a cloud environment. The most stable and simple to use distribution we found was DSL.

4.1 Ubuntu

Ubuntu is a Linux Distribution developed by Canonical. It is probably the most popular Linux Distribution, being used by over 20 Million people every day.

Ubuntu started in 2004 when "Mark Shuttleworth gathered a small team of developers from one of the most established Linux projects – Debian - and set out to create an easy-to-use Linux desktop, Ubuntu." (Ubuntu.com).

From the start Ubuntu concentrated in being user friendly and by doing so it managed to put itself as a strong first step when moving to Linux. In fact, Ubuntu easily became popular enough that it got a stripped down server edition and today it is popular on both as desktop and server OS.

In our tests we used Ubuntu Server 12.04 LTS 64 bit. We chose the server edition because this is the one supported by the major cloud providers and it doesn't have any kind of GUI in it, keeping the OS itself to a minimal.

Ubuntu uses Linux Kernel 3.2.

4.2 CentOS

CentOS is also known as *the* community enterprise operating system. It is specifically built for enterprise use. It is a more stable Operating System then other Linux distributions and it offers long term support for the individuals and organizations who use it.

"CentOS exists to provide a free enterprise class computing platform to anyone who wishes to use it. CentOS 2, 3, and 4 are built from publically available open source SRPMS provided by a prominent North American Enterprise Linux vendor. CentOS conforms fully with the upstream vendors redistribution policies and aims to be 100% binary compatible. (CentOS mainly changes packages to remove upstream vendor branding and artwork.). CentOS is designed for people who need an enterprise class OS without the cost or support of the prominent North American Enterprise Linux vendor." (CentOS.org website)

The CentOS team is a small one but it is slowly growing and has already gathered a strong community around it.

We decided to use CentOS 6.2 64 bit. We installed it without a GUI so that it would be smaller and run faster.

CentOS uses Linux Kernel 2.6.

4.3 Damn Small Linux

Damn Small Linux is a very light OS, it can run from a USB stick or a CD and it can stay completely in RAM.

Being this small DSL can feel very fast on old machines. The impressive thing about it is that it manages to stay small and still run a GUI and even Firefox. It's size actually makes it competitive with large distributions because of the speed advantage it offers. Having a small distribution means having more resources available for the application one is interested in, where the resources in question refer to RAM, CPU cache or actual storage space.

Unfortunately the distribution does not offer as many options as more popular distributions like Ubuntu or CentOS, but supporting gcc means that most applications can be compiled and executed on it. More worryingly the last version of DSL was released in 2008 and support seems to have ended. With all that being said DSL is still an excellent representative of the light OSs and probably the simplest to use at the moment. Alternatives like Puppy Linux were considered, they were found to be a bit less user-friendly and not as well made.

"Damn Small Linux is a very versatile 50MB mini desktop oriented Linux distribution." (damnsmalllinux.org website)

We installed Damn Small Linux on the hard drive so it wouldn't have to hold the entire kernel and programs in memory. It installs with a GUI but before we ran the tests we stopped it and exited to prompt.

It is important to mention that DSL does not have native SCSI support, which demanded the creation IDE disks in the virtual machine management console.

Damn Small Linux runs Kernel 2.4 and it never benefited from the improvements it came with the newer kernel versions.

We believe that a mini-OS like DSL can prove to be an advantage on the cloud because of the small space it uses in memory and the speed increases it could potentially provide. We are not sure that DSL offers the answer lacking modern kernel features but it is sufficiently representative.

5. Tests

When it comes to a system as complex as the cloud, there are a lot of things one needs to consider. In the cloud, even the smallest increase in efficiency can bring about major improvements in service quality. One strives for the best CPUs, the best RAM, the best Storage, all of it at the smallest price and least power consumption possible. It is even important to consider the amount of heat the entire computer farm gives out because cooling such vast systems is increasingly difficult. This are problems the cloud administrators and the companies that deal with them need to handle every day.

But when one uses the cloud as a consumer, there are other things that matter. Of course, for some the cloud can represent just a machine that needs to run 24/7. But when one is interested in High Performance Computing, getting everything from the cloud at the smallest price possible is vital. That's where the details start to matter – having sse4 support on the CPU could make the entire execution time of a large batch operation drop by days. Unfortunately the hardware used is picked by the cloud company and the end user has little influence on it other than finding a cloud provider that offers the hardware one needs. We can take for instance the need for a GPU cluster, in this case one would need to find someone like Amazon that provides GPU support.

Although the user does not have control over the details of the hardware used – he can't decide if the CPU is an Intel or AMD, for instance – one does have control of the amount of resources one purchases. The amount of resources needs to be calculated based on the project and it is doubtful that a formula that fits all projects will ever be found.

We decided to pick the most pressing resources and test only those. We tested the CPU, the memory throughput and the Network using the OSs and the VMMs that we mentioned in the previous chapters.

Because in a cloud environment there is a high probability that one's virtual machine would run with several others on the same hardware we also considered important to test the effects of other similar VMs have on the efficiency one gets and if the VMMs schedule the machines in a fair way.

The tests were run repeatedly with a combination of VM settings.

We used Hyper-V and on top of it we run 1, 2, 4 and 8 VMs with Ubuntu and CentOS. And we had an external Physical machine with which we tested the bandwidth.

We used VMware ESXi and on top of it we run 1, 2, 4 and 8 VMs with Ubuntu, CentOS and DSL. And we had an external Physical machine with which we tested the bandwidth.

5.1 The Central Processing Unit

With the wide range of CPUs available today comparing one to another is increasingly difficult and is based more on the task the CPU is being used for. If some tasks work best on a large number of general purpose CPUs, that does not mean there are not tasks that would work best on a single CPU specially designed for it, using a FPGA for instance.

We believe that a cloud user determines that the task at hand fits the cloud and that it does not require any special hardware other then what a cloud provider can offer. So we tested CPU in a simple matter, we gave it a somewhat trivial mathematical task and let it compute it in a loop for a given number of steps then compared the time it took until it finished.

We used the square root as the mathematical function because it uses a small amount of variables and because of this, a small amount of memory. Using as little memory as possible was vital, assuring us that only the cache would be used and that it would give a clear image of the CPU alone, without any external interference.

5.2 The Memory

Even though the processor is the work-horse of the CPU, memory can massively determine the efficiency of a system and in some cases it a small upgrade to memory can bring with it bigger improvements than changing the CPU.

In a cloud environment the amount of memory given to a VM is also the first to be controlled by the user. In the case of Linode for instance one can only control the amount of memory, storage and network traffic. The CPU is given and it is an Intel Xeon L5520 at 2.27 GHz with 4 cores and the remaining hardware is also fixed.

The amount of memory used is extremely variable depending on the task. On the other hand the speed of the memory is important, and this is what we decided to test.

We tested the memory allocating space, writing random data to it, moving it around then freeing all the space. All of this was done in a loop and run several times in all our tests. This way the calculations that have to be done in the CPU are minimal and most of the time is spent writing to memory.

5.3 The Network

In most applications calculating the data is not enough, one also needs to move it around and in some high performance applications this seems to be the biggest bottleneck. If the communication between the VMs is important the cloud might not even be suited for that application as one cannot control the bandwidth between two VMs, and one cannot even guarantee that the VMs will constantly be in the same physical location. Results can vary a lot if the machines happen to be on the same physical unit, in the same building or even on different continents as it can be the case with large cloud providers that have farms spread all over the world.

Most of the time in the cloud the bandwidth needed goes towards the internet and some external users of the service provided by the cloud guest. This is extremely hard to measure because getting small latency to ones client and high bandwidth can even be a problem that is not even controlled by the cloud provider but by the entire internet.

In our tests we were more interested to see if the VMs can use all the bandwidth provided to the physical machine and in case there are multiple machines using it, if it would be shared in a fair manner.

To test the bandwidth we used an application called iperf, it creates a server on one end with “iperf –s” and one can connect to it using “iperf –c <IP>”. CentOS and DSL did not provide package support for it but iperf has the entire source code available and we managed to compile it using “./configure && make && make install”. It worked very well and it gave us the data we needed. It was a bit difficult to start it at the same time for 8 clients but we used a script that would run it and throw it in the background then run the next one.

5.4 Encountered Problems

During the project we encountered a multitude of problems that made it difficult to run the complete tests we wanted.

Probably the biggest problem we encountered was caused by the Xen hypervisor. Xen requires a special Linux Kernel for Domain 0 to work and this kernel is not as simple to install as it seems.

First we wanted to try Xen on an Ubuntu server. The first problem we got was installing Ubuntu itself. The problems with installing Ubuntu were mostly concentrated on the Grub. Here we had a ton of issues. Grub refused to run because the hard drive we were using was too large. So we had to create a special partition just for "/boot". Once we did this Grub installed with no issues and everything worked fine. It is important to mention that the version of Grub Ubuntu 9.0 comes with was stable and worked from the start, but Ubuntu 9 is no longer supported especially considering we moved to LTS 12. Another interesting thing about the Ubuntu install was that it offers to install Grub to the "first disk" without informing the user which disk that actually is. Considering we installed Ubuntu from a USB stick we were very surprised when we discovered that Grub was installed on the USB Stick we used for install instead of the actual hard drive.

After we managed to install Ubuntu we encountered more problems. Theoretically Ubuntu 12 has full support for Xen and one can just use aptitude to install it and then a restart gets one to the Xen Kernel. At this point we needed to connect a manager to the Xen hypervisor but to no success.

Debian was our next option, also fully supporting Xen, but it had problems with the network card drivers. We used a Realtek network card that had proprietary drivers which we couldn't find support for.

After Debian we chose to move to XenServer provided by Citrix. This one comes as a full operating system and requires nothing else to run. Unfortunately it refused to run as we got a Kernel Panic right after boot.

We believe that the problems we had with Xen were all caused by trying to use it on unsupported hardware as we had the Realtek network card and an Nvidia north bridge. On the other hand we did test Xen on the Linode cloud service where we got some interesting results.

VMware ESXi also proved to be a bit of a challenge. It comes as an .iso and installs directly on the physical machine. It is important to mention that the installation itself is rather smooth and we encountered no issues with it. The only issues we had with VMware ESXi was its need for a data store and the fact that it only accepts a handful of storage options for it. We could not provide it with a SCSI hard drive or a fiber wire one, but we did manage to set up a NFS server using Debian. The NFS server proved challenging in itself because NFS cannot run over any file system type so we installed an external hard drive and gave it an ext3 partition on which to run on.

Hyper-V requires Windows Server to run. We tried to install Windows Server on an external hard drive but it refused. We did manage to get an internal IDE hard drive and installed Windows Server on that. After this the Hyper-V installation was just a few clicks away, we just selected the role and Windows did everything for us, then a Hyper-V administration console appeared in the server management window.

We would like to mention that Hyper-V was the simplest to set up and use as VMware ESXi forced us to use 3 computers for the task. On the other hand for desktop virtualization there are simpler to use tools that do not offer such a high efficiency.

As we mentioned before when we discussed the Hyper-V architecture, Hyper-V requires special kernels from the guest and even thou most new kernels support it we did not manage to run DSL on it.

It is important to mention that all this hypervisors are meant for an enterprise environment and expect a certain type of hardware to function properly. There is no USB support for instance and most of them expect high performance storage systems.

All these needs make it a bit difficult to test the hypervisors completely and properly. Also the problems we encountered with storage made us take the decision to not test the storage capabilities in any way and write our testing software in a way that it would not be dependent of storage in any way. We knew that the storage we used would probably put all the hypervisors in a bad light and probably make an unfair difference between them.

6. Results

We will break the results in the same categories we run the tests in and analyze them separately trying to explain what caused the difference between them.

We will start with the CPU then move to memory and finally networking trying to discuss both the impact the VMM had and the impact the OS had.

6.1 The Central Processing Unit

To see what the CPU efficiency is we will take the tests we run individually. We will take all combinations of Operating Systems and Virtual Machine Managers and treat them independently then we will try to make a larger comparison between them.

We will first start with the VMware with Ubuntu, CentOS and DSL then we will move to Hyper-V with Ubuntu then CentOS. After that we will try to make a comparison between the Operating systems and one between the VMM providers.

6.1.1 VMware ESXi

The following tests are all run in VMware ESXi using 1, 2, 4 and 8 Virtual Machines, all of them with one CPU with 512 MB of RAM.

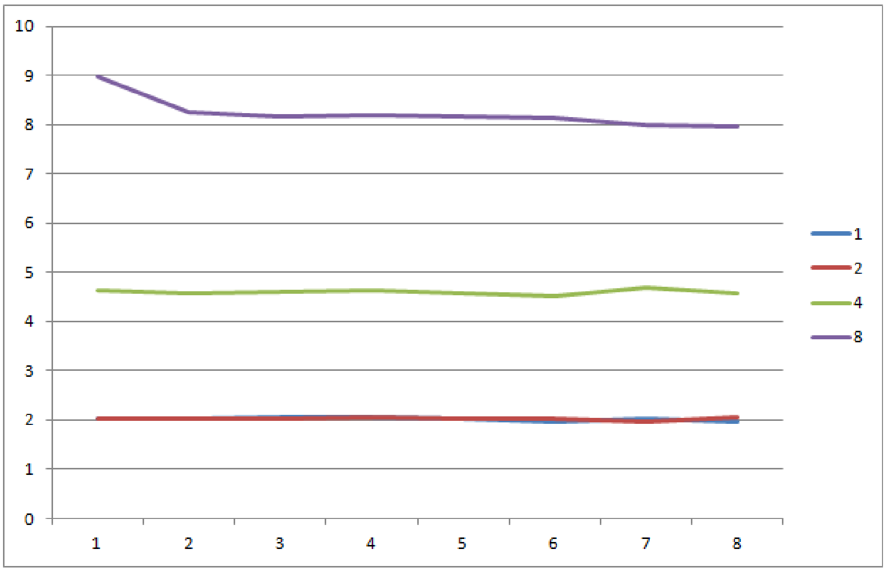

Fig. 5 VMware ESXi – Ubuntu – CPU Test

In figure 5 we have the tests run with 1, 2, 4 and 8 Virtual Machines started with Ubuntu. The horizontal axis represents the number of the test and the vertical axis represents the time it took until the program ended.

It is easy to observe that with the number of VM raising the execution time grows proportionally. This does not happen between 1 and 2 Virtual Machines started because we have 2 available CPU cores and each Virtual Machine uses one of them. When the Number of VMs grows bigger than the number of available CPUs then the execution time grows with it. This is to be expected and it shows that the VMM splits the CPUs correctly.

Next we will analyze the CentOS and DSL system expecting similar results and comparing them at the end.

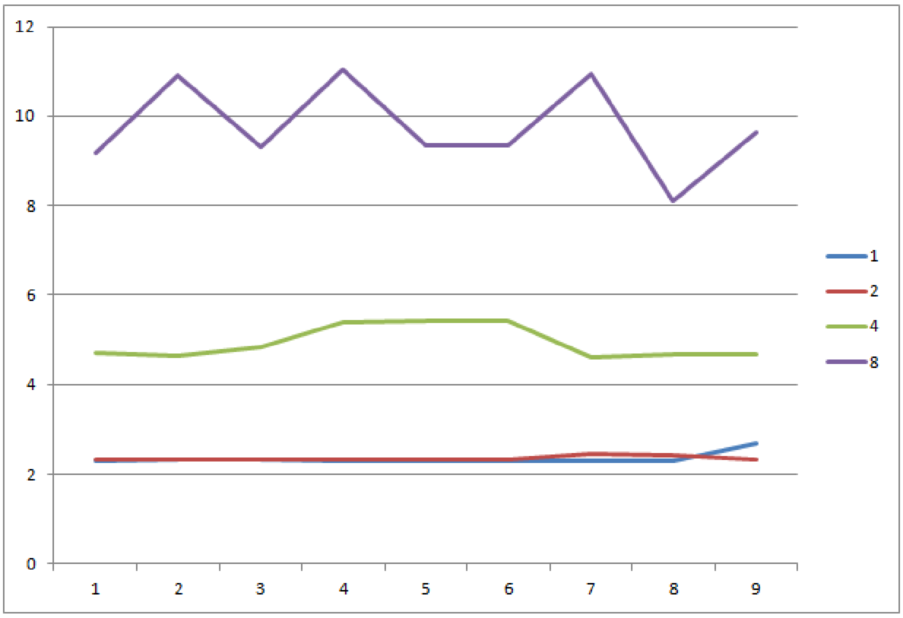

Fig. 6 VMware ESXi – CentOS – CPU Test

In the case of CentOS we can notice the same pattern growing just like the case with Ubuntu, the system work as expected with 1 and 2 Virtual Machines but the execution time grows the more Virtual Machines we start.

It is important to notice how the heavy load in the case of CentOS makes the results be a lot less linear and more unpredictable. Still they are marginally in the same area the execution time growing proportionally to the number of virtual machines started.

The way the entire system becomes unpredictable in the case of CentOS should probably be analyzed further. It could be caused by bad communication between the Guest OS and the VMM or something wrong with the kernel. It is important to note that the kernel is older than the one Ubuntu uses, 2.6 instead of 3.2 and this can be noticed in a speed decrease even with just one virtual machine.

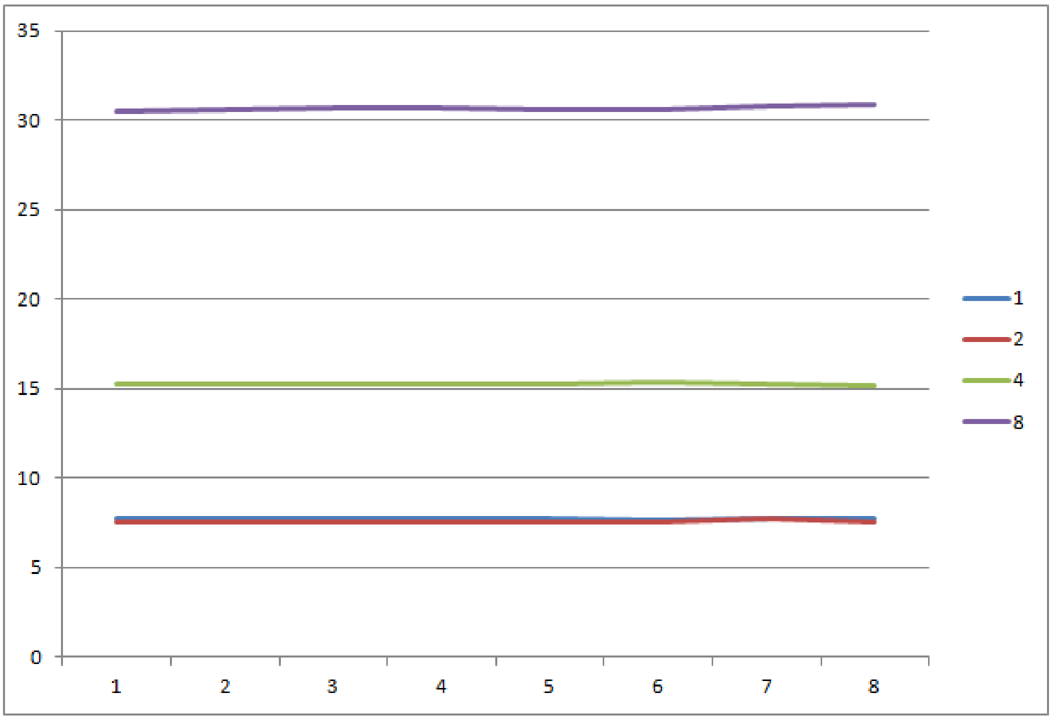

Fig. 7 VMware ESXi – DSL – CPU Test

In the case with DSL just as we expected we can notice the same pattern. The system finishes to execute in mostly the same time with one and 2 Virtual Machines started and the time grows proportional to the number of virtual machines provided. Just like in the other cases this is because we have 2 physical processors available and the VMM has no problem assigning a full processor to each virtual machine.

The execution time seems more linear then the case of CentOS but this does not mean it is, if we look closely at the numbers we can see that the execution time in the case of 1 virtual machine started grew from almost 2 to about 6 or 7. This is caused by the fact that DSL uses an old Kernel which is no longer supported, Linux Kernel 2.4, and the mathematical library is probably not as good. It is clear that the fault is the library it provides to calculate the square root because the time remains proportional in the case of more virtual machines. We cannot say if the results are more linear than in the case of CentOS because the execution time growing this much can hide and irregularity.

Now that we have determined the VMM works as expected, regardless of the Operating System used, and that the execution time grows linearly we should also check how fair the scheduler for the VMM is. We will do this comparing the results from 8 VM running Ubuntu.

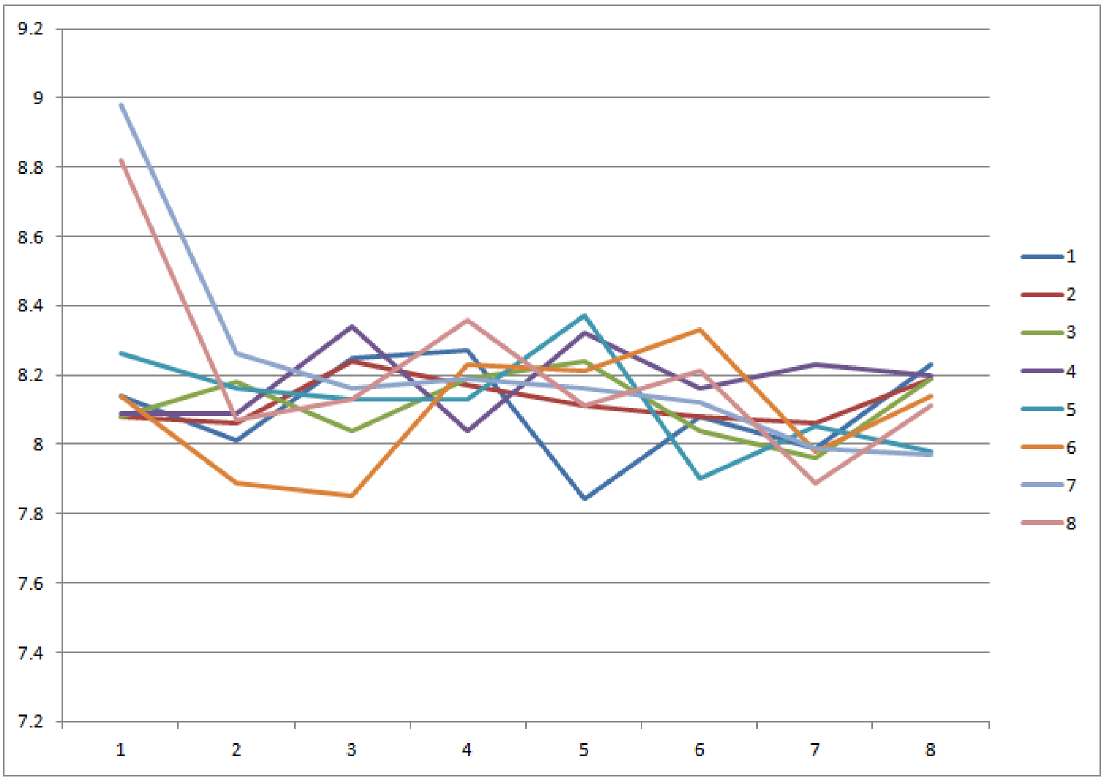

Fig. 7 VMware – Ubuntu – 8 machines

In figure 7 we can see how the execution time changed in each of the 8 virtual machines during the tests we run. The resources are not shared in a totally equal and fair manner, some get a bit more, some a bit less, but they mainly stick around 8.2 seconds as execution time. This is to be expected because no matter how well written a scheduler is it cannot offer fair time parts for every machine unless it manages to split the time in infinitely small parts and this is not possible because every CPU instruction has an amount of time it needs to finish executing.

We believe that given the circumstances and the fact that the execution time for all machines stays around the same value proves that the VMM splits the CPU in an fair and equal manner.

This also happens in the case of CentOS and DSL, the graphs being very similar to this one.

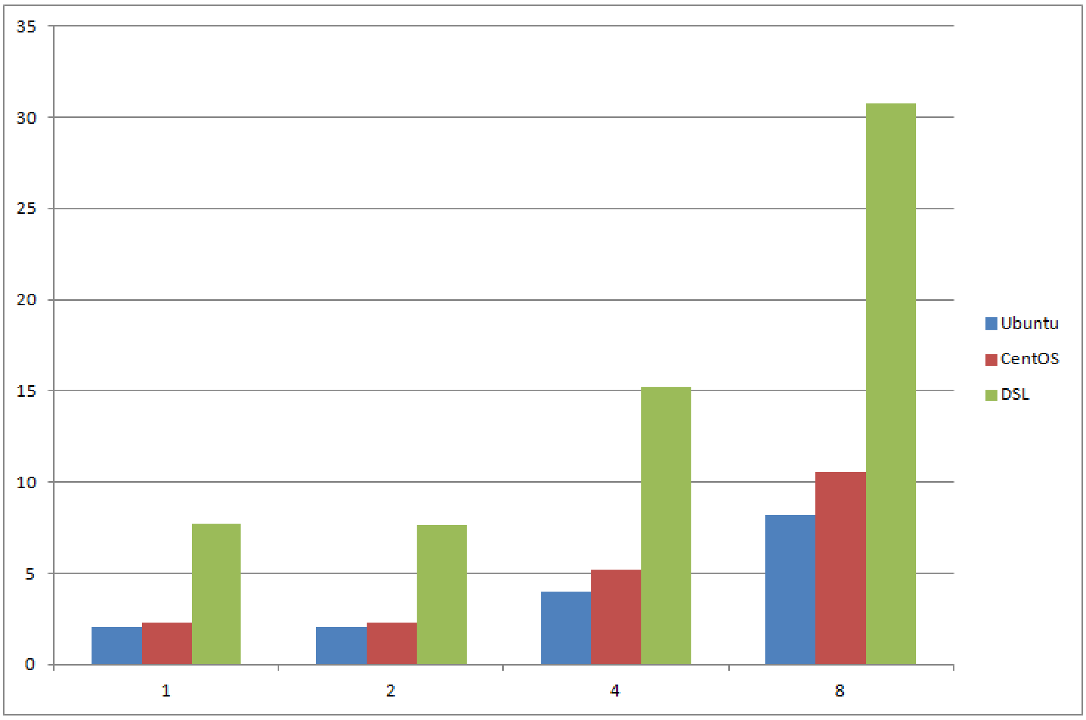

Fig. 8 VMware ESXi – CPU Test – OS comparison

In figure 8 we can see exactly how the Operating Systems compare to each other when used in a Virtual Machine environment. The horizontal axis represents the number of virtual machines started with each OS and the vertical one represents the execution time. Just like before smaller is better.

We can see here a bit clearer how no matter what OS we use the execution time for 1 and 2 virtual machines running is almost the same because each machine runs on a separate CPU and that with more machines added the execution time grows proportionally.

It is also very clear how the improvements in the Linux Kernel over the years directly influence the execution time. And even thou at first glance we expected DSL to finish in an amount of time comparable or better than the other OSs the older kernel proved to cause exactly the opposite and made DSL offer much worse results.

We have to say that for a task that is CPU intensive that needs to start running as fast as possible in or outside of a cloud environment using a popular Linux distribution seems to be the best option, not only because it is simpler to use or more efficient but the amount of people working on it and constantly testing it makes it more stable and gives it a better chance to have bugs and security issues already patched.

On the other hand if long term support or backward compatibility are an issue using CentOS is the best alternative, with CPU execution time almost as good as Ubuntu, it is a stable and powerful release that can provide a good alternative for ones projects.

6.1.2 Microsoft Hyper-V

The following tests were all run on the system we described in the hardware section using one CPU and 256 MB. We wanted to give every machine 512 MB like we did in the case of VMware

ESXi but Hyper-V needs all the RAM for the Virtual Machines to be real, physical RAM and it takes into account the RAM the Windows Server itself is using (about 1.3 GB). Both OSs had no problem running with just 256 MB even thou installing CentOS required 1 GB of RAM.

The tests are done with 1, 2, 4 and 8 virtual machines and the results are given in execution time, so just like in the case of VMware ESXi smaller is better. First we will analyze Ubuntu and then CentOS. Again DSL could not be run on Hyper-V without changing the kernel so there are no results for it.

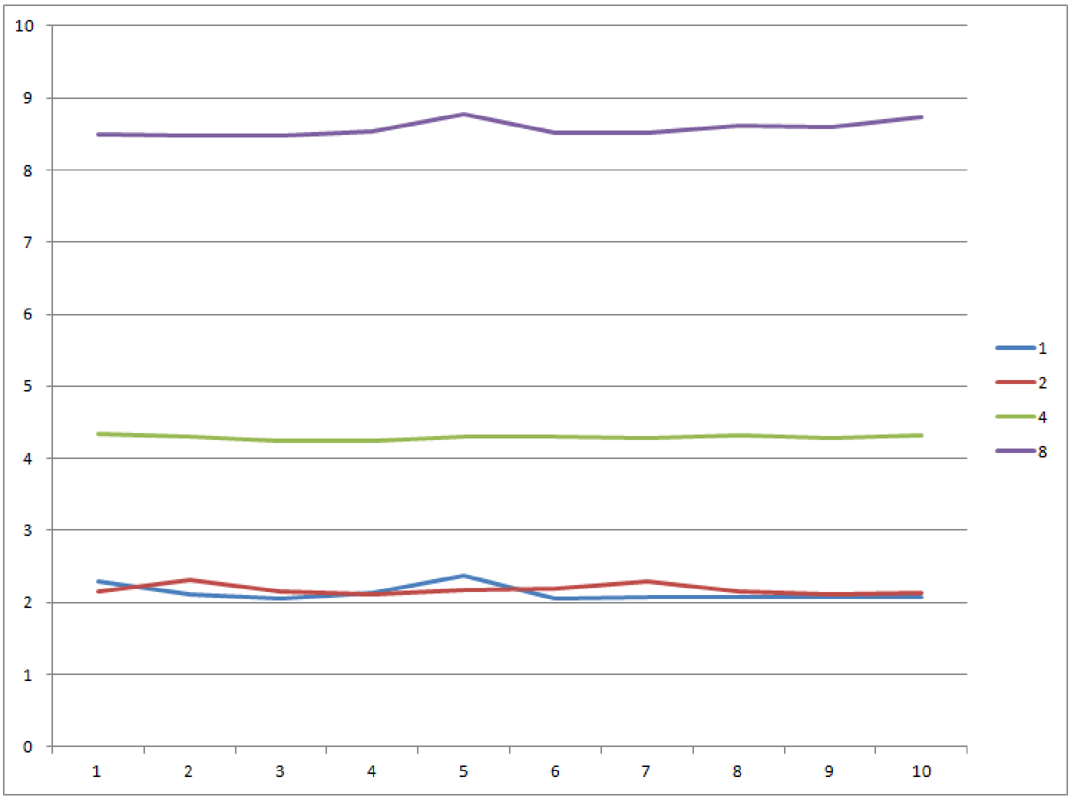

Fig. 9 Hyper-V – Ubuntu – CPU Test

In figure 9 we can observe how the execution time remains the same for 1 and 2 virtual machines running and grows if we add more. This is very similar to what we noticed on the 3 graphs (figure 7, figure 6 and figure 5) we have done for VMware ESXi. The CPU execution time seems to have a linear pattern with the more tests we run showing that the scheduler is fair.

It is important to mention that along with the virtual machines in the case of Hyper-V there is also the Windows Server Operating System running with a full graphical interface and the administration tools are on the same physical computer with the virtual machines unlike it was the case with VMware ESXi. This can affect the results in a meaningful way considering a lot of the available resources probably go to the Windows Operating System.

It is impressive that system stability is comparable with the case of VMware ESXi considering the vast differences between the 2 VMM. And if we compare Fig. 9 with Fig. 5 we will notice that the execution times themselves are rather similar.

We have to keep in mind that the Linux Kernel for both Ubuntu and CentOS both have LinuxHV which helps Hyper-V manage the Virtual machines and probably also gives them an efficiency boost.

This is also important considering Azure is starting to offer Linux support so that users that already have Linux applications find it more simple to move to the cloud.

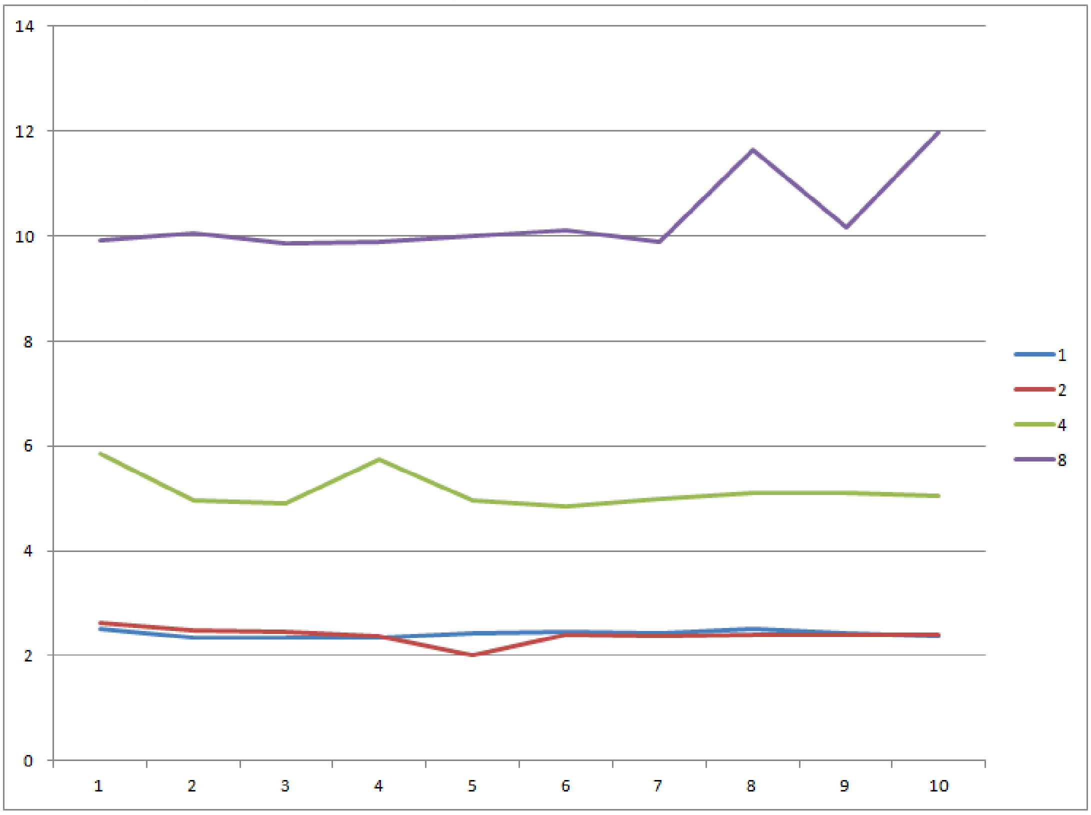

Fig. 10 Hyper-V – CentOS – CPU Test

Just like it was the case with VMware we can see in figure 10 that even thou CentOS keeps having similar execution times they also seem a bit less linear. This confirms that this is not a VMM issue but an OS one. Luckily this lack of linearity is not that relevant considering the values tend to stay around a central one.

We also need to check the fairness of the VMM and this time we decided to use CentOS to determine it to make sure it is not just a Distribution that gets to share the resources in a fair way.

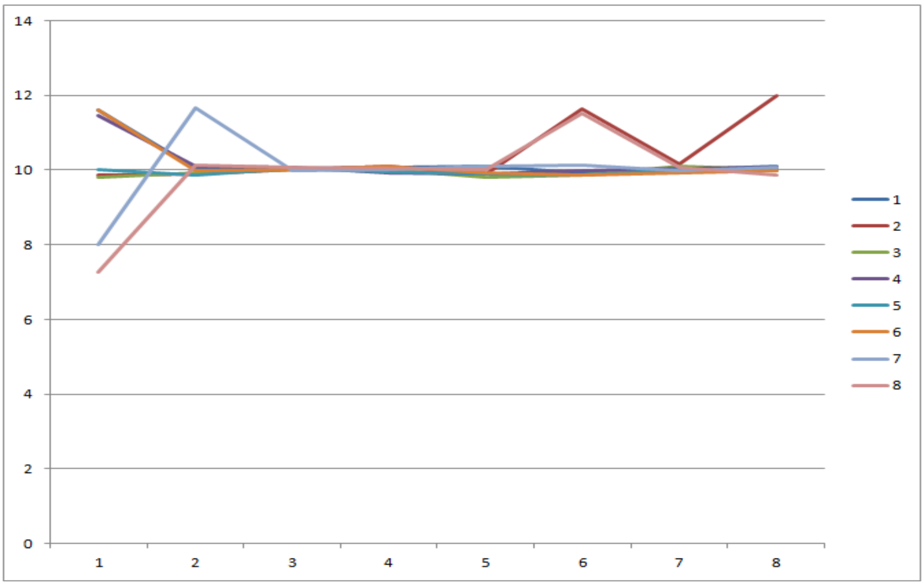

Fig. 11 Hyper-V – CentOS – 8 Machines

It is clear that all the VMs have an execution time around the same value and they all share the processor in a fair manner just like was the case with VMware ESXi. There are differences at the start because not all processes started at exactly the same time and the scheduler got a bigger request for resources when it didn't really expect it.

Just like VMware ESXi Microsoft Hyper-V shares the processor in a dynamic way but there are ways to set it so that it would guarantee a certain percentage of the processor to a specific virtual machines. This is done per machine unlike in the case of VMware in which this is done by creating a resource pool.

Next we will compare the Operating Systems between them to see which if any is better.

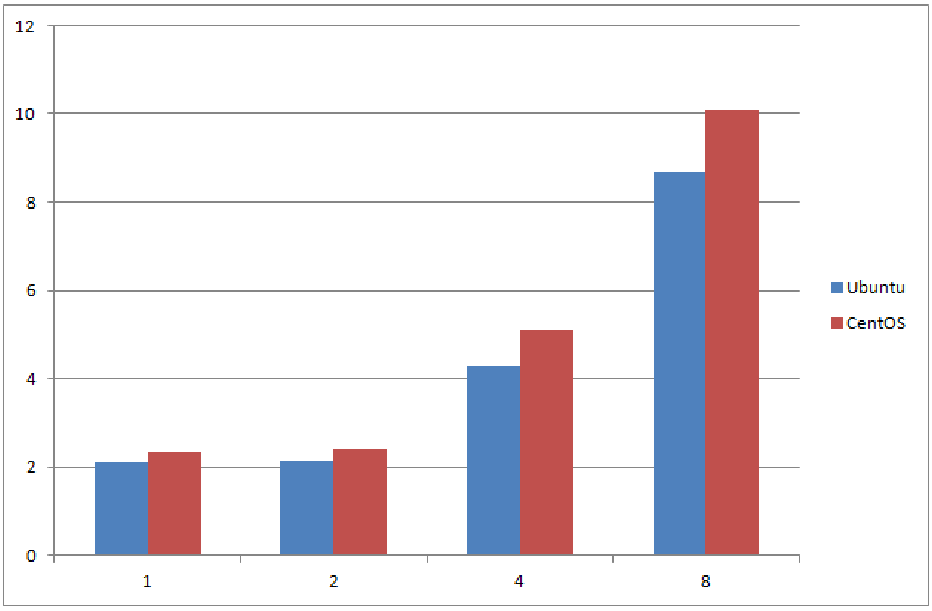

Fig. 12 Hyper-V – CPU Test – OS Comparison

Just like was the case with VMware we can see clearly that Ubuntu is just a bit faster than CentOS. This makes it clear that between versions 2.6 and 3.2 the Linux Kernel got some improvements that slightly increased its performance when it comes to CPU. This slight advantage Ubuntu has can mean a lot for someone interested in high performance and on the long run it might add up. This does not make CentOS a far worse option considering it offers other advantages and it is not far behind.

6.1.2 VMware ESXi versus Microsoft Hyper-V

Now that we discussed in detail the advantages the OS gives when it comes to CPU efficiency it is time we discuss the advantages actually introduced by the Hypervisors themselves. We will discuss Xen in a different section as the results are not directly comparable with these.

Outside of the CPU difference it is important to keep in mind the difference between this 2 VMMs. VMware ESXi is lighter and can easily be managed remotely. It is also compatible with other virtualization software offered by VMware. The most important part of it and its biggest advantage is that it supports any Guest OS without any kernel modification.

Hyper-V on the other side is more compatible with Windows products and with using Windows as a guest, which we did not test and if someone requires Windows guests it is probably the best option considering a Windows user can easily learn to use it.

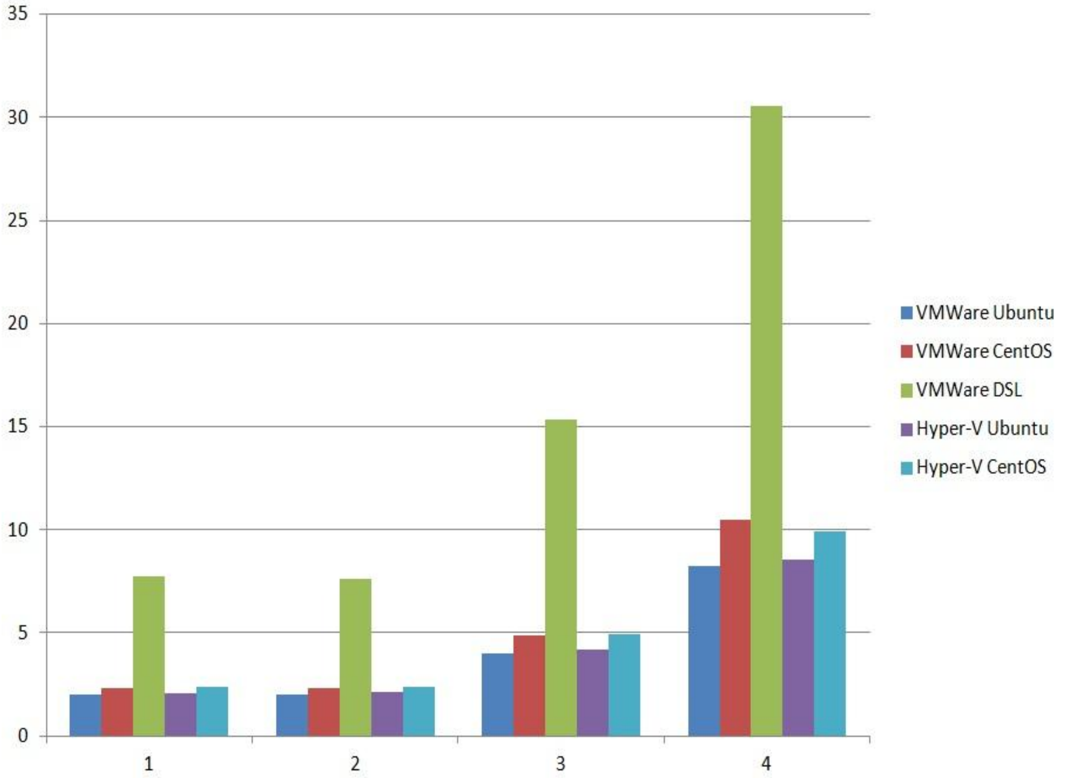

Fig. 13 CPU Test – Hypervisor/OS Comparison

If we compare the CPU results from both Hypervisors like we did in figure 13 we can notice that Hyper-V is just a bit slower then VMware. It is important to remember that the Hyper-V VMM also run Windows Server 2008 along with the machines and even though this OS is left mostly Idle during the tests running the GUI and the control panel for the VMs takes some processor. It would be interesting, if at all possible, to test Hyper-V without the additional Windows Server running in the background as the results may prove to be rather different. Of course VMware ESXi also uses an amount of processor for the scheduler and to send needed data to the remote control panel.

Even if it's slightly slower Hyper-V can prove to be a good choice for a lot of users and in the case of using Windows as a Guest probably even a best choice. Still our tests show that the best option at the moment to get the best CPU efficiency is to use Ubuntu on top of VMware ESXi.

6.2 The Memory

Even if the results for CPU show us which distribution to choose this does not mean the overall system performance is dictated only by it. The other important factor in any PC is the memory. We will analyze the results much in the same way we did with the CPU starting with VMware with the 3 OSs and then moving to Hyper-V with Ubuntu and CentOS.

The quantity of memory can easily be changed in a Cloud environment, of course at a cost, so our tests only show memory speed and how the number of Virtual Machines influence it.

We are interested which OS is better at handling memory and which Hypervisor shares it in a more efficient and fair way. For this we will show similar graphs with what we showed in the CPU section. Again the vertical axis represents execution time and the smaller the value is the better.

We would like to remind the reader that Hyper-V only accepts to share actual physical RAM and would refuse to start a new VM if there is no RAM available and that tests on Hyper-V were run only with 256 MB.

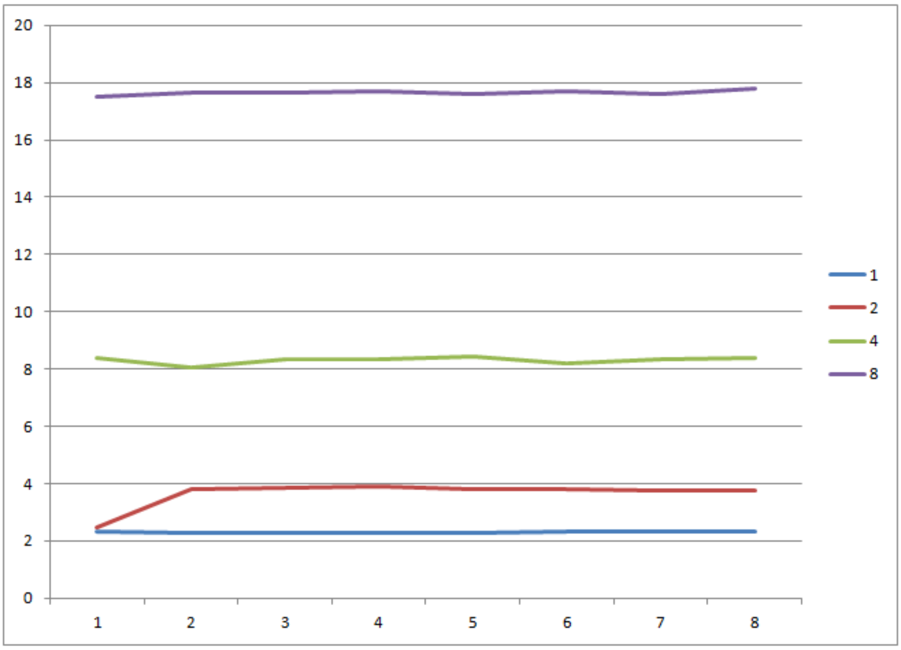

Fig. 14 VMware ESXi – Ubuntu – Memory Test

We can see in figure 14 that unlike the case with CPU the execution time grows linearly with the number of VMs from the start. This is because even if there are 2 CPUs to process the data the PC offers only one bus to communicate with the memory and a single CPU can easily fill up that bus.

The execution time growing in a linear way shows that the hypervisor has a somewhat fair way of sharing memory and the memory bus and that managing multiple VMs doesn't affect the VMM too much.

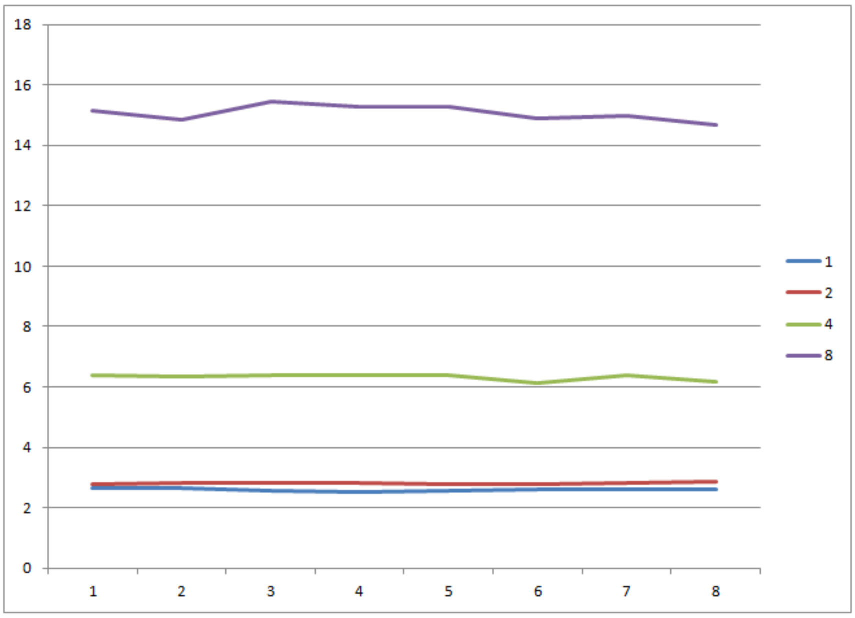

Fig. 15 VMware ESXi – CentOS – Memory Test

We can see in figure 15 that CentOS works almost in the same way as Ubuntu when it comes to memory, just like was the case with CPU but the tests for 1 and 2 VMs seem to be very similar, pointing towards the idea that CentOS either has a better memory management system or that it fails to use the entire bus from the start.

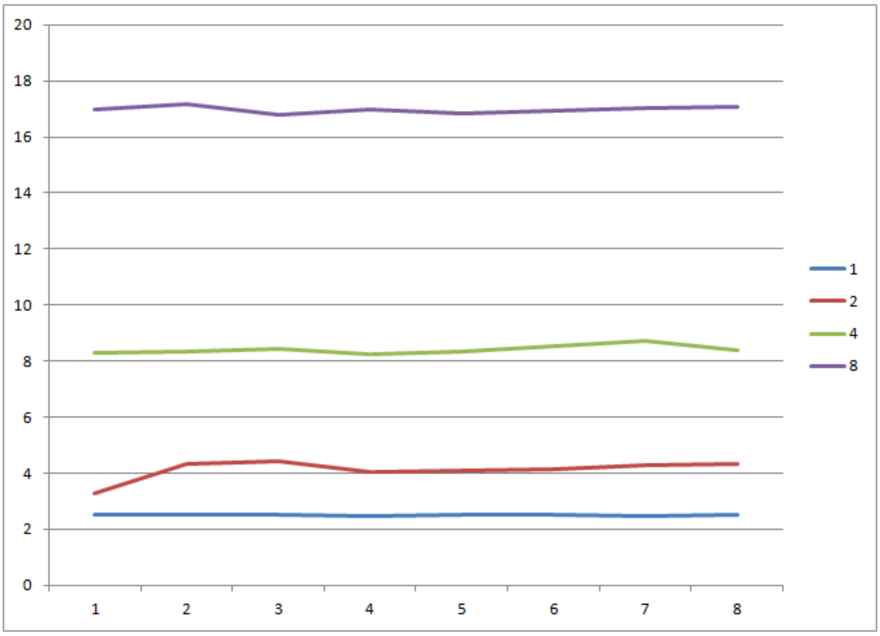

Fig. 16 – VMware ESXi – DSL – Memory Test

It is interesting to notice how figure 16 differs from figure 7. Here the results from DSL are far closer to the results we got for Ubuntu. The execution time grows in a similar manner and DSL has a good execution time from the start. This is because even thou there are changes, in the kernel, in the way memory is shared between applications and the way memory is handled the changes do not bring any efficiency in a way as visible as was the case with the CPU.

Here is where we most expected DSL to prove it's a good idea and it did. The results are comparable with the top Distributions even if the DSL distribution hasn't been updated in years. This makes us believe mini-distributions still have a huge potential in the Cloud environment.

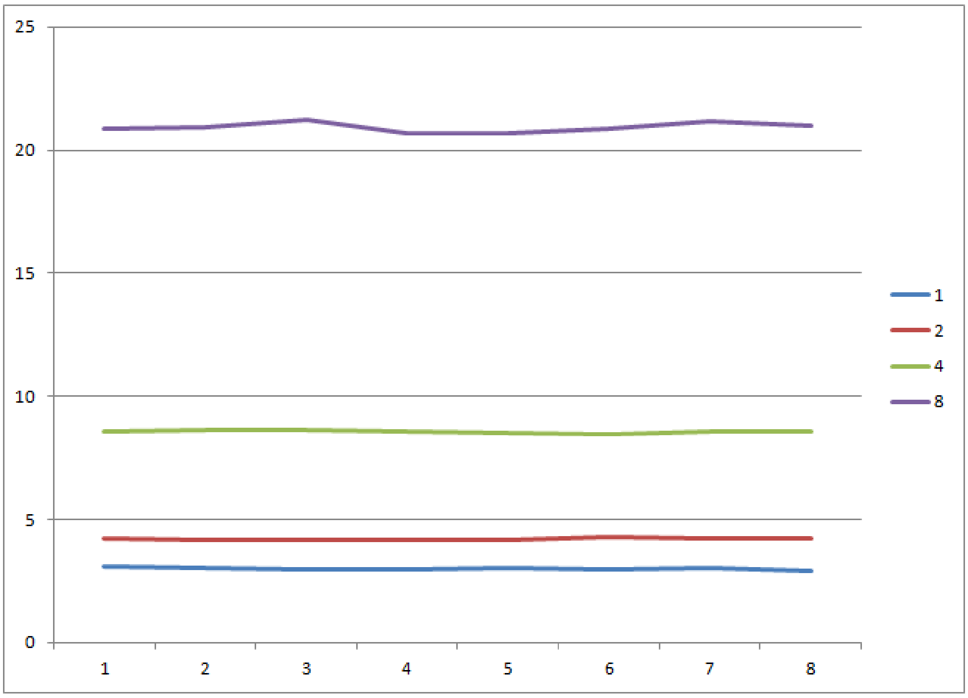

Fig. 17 – Hyper-V – Ubuntu – Memory Test

Moving to Hyper-V we can see in figure 17 that the same linear growth in execution time is present and that Ubuntu seems a bit slower in all the tests. This is caused by the fact that Hyper-V runs with Windows Server which uses a lot of memory. Because of this the cache gets overwritten a lot more often and the entire speed of the virtual machine drops a bit when memory is accessed.

The linearity remains showing that the systems is stable even with 8 VMs all accessing memory.

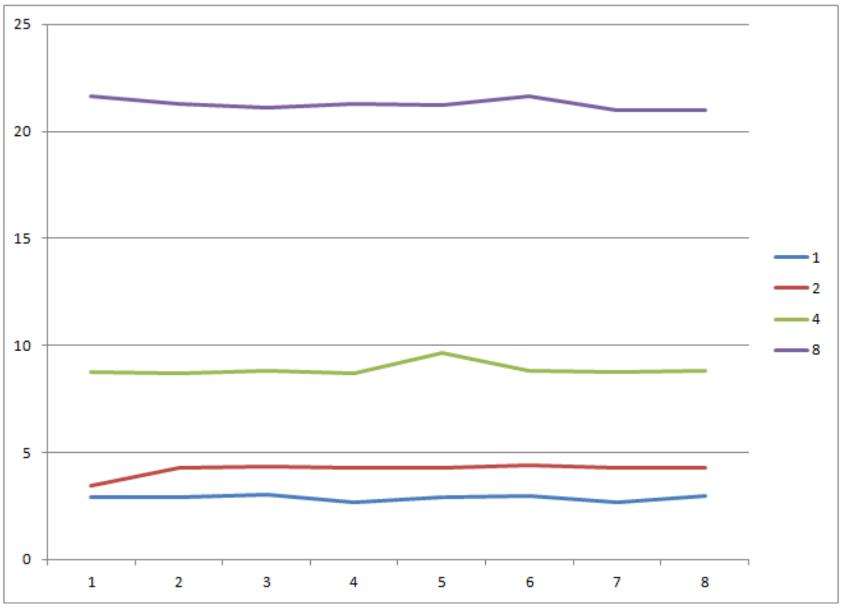

Fig. 18 Hyper-V – CentOS – Memory Test

The fairness of the scheduler and the fact that memory is correctly shared is also shown in figure 18 in the case of CentOS.

Just like was the case with VMware execution time in memory tests starts growing from the start, the number of available cores being unable to help the fact that there is only one available bus and that it is shared between all the Virtual Machines and the Host.

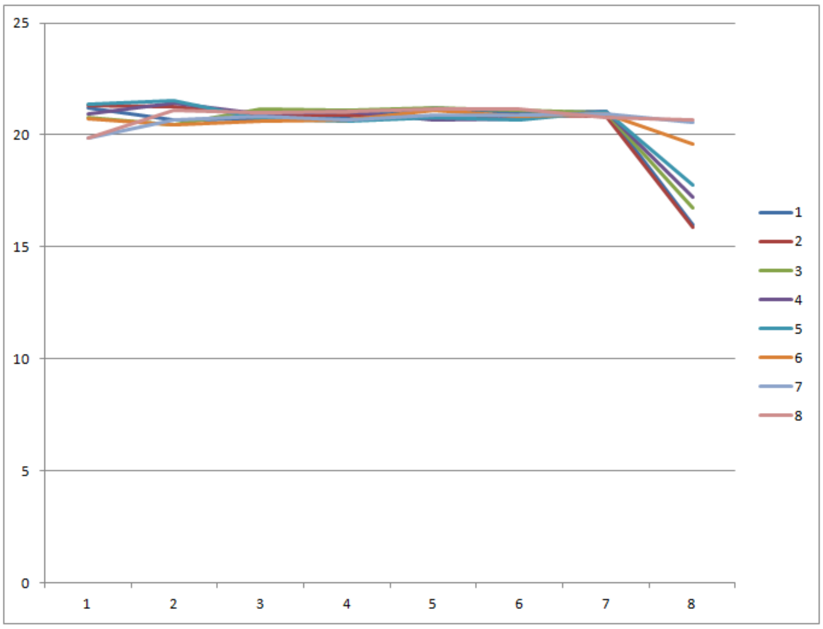

Fig.19 Hyper-V – Ubuntu – 8 machines

In figure 19 we can see that even in the case of memory system resources are equally shared by the VMM, all the VMs getting about the same execution time.

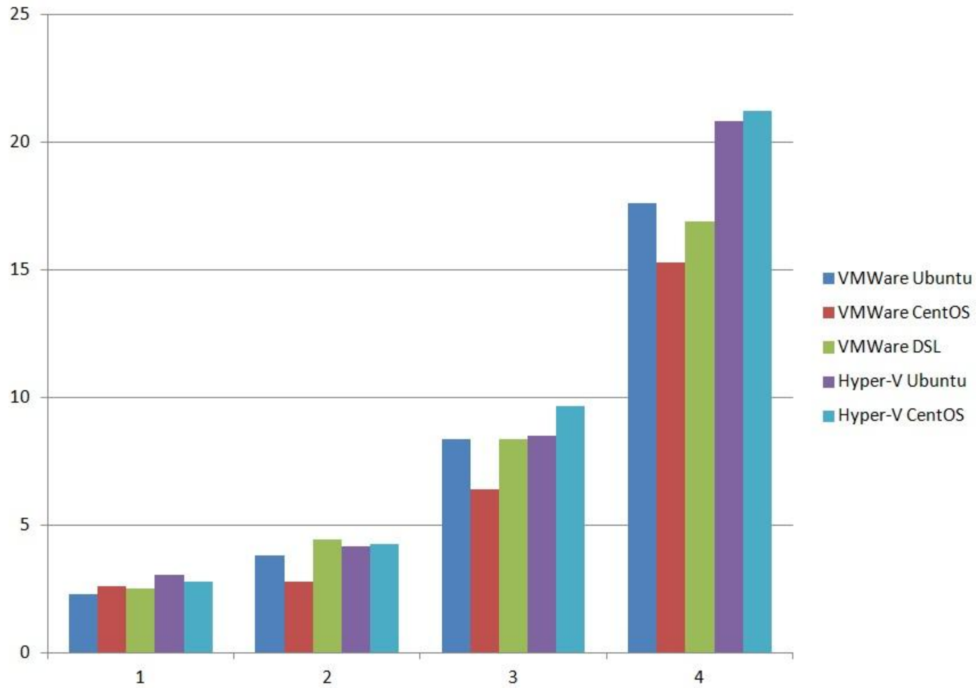

Fig. 20 Memory Test – Hypervisor/OS comparison

When we tested memory the results we got in all the tests were a lot more proportional compared to the big discrepancy that we got in the case of the CPU. This is mainly because the kernel efficiency when it comes to memory did not have such a huge impact over time so the distributions were a bit more equal.

We can see in figure 20 that there are still differences between the Hypervisors and between the Operating Systems. Hyper-V is a bit slower because of the Windows Server system it needs to run in parallel with the virtual machines.

Also we noticed here that DSL is just a bit faster than Ubuntu and this proves that a mini distribution could have an interesting impact on the cloud.

It is important to mention that the best results were given by CentOS under VMware. This is in part that it has a somewhat smaller memory footprint then Ubuntu, 185 MB rather than 195 MB but none compare to the under 20 MB that DSL manages to use. It could also be the way CentOS integrates with VMware and how it allocates memory in a way such that it provokes a smaller number of cache overwrites.

The results force us to consider that the best solution is CentOS on VMware for any application that has high memory throw output. But one has to consider the results Ubuntu got for CPU and the possibility that a small distribution has the potential of being faster than both.

We also have to notice that even thou Hyper-V also runs Windows Server in parallel with the virtual machines it managed to get similar results. This forces us to appreciate the way Hyper-V manages the system and makes us wonder if some modifications to it and a better configuration could actually make Hyper-V the best choice.

The results from both CPU and Memory point towards the fact that a specifically built Linux Distribution for the task would be the best choice if this is at all possible, if the resources are available. This is something that needs to be further studied in the future. Luckily cloud providers like Linode offer the possibility to upload a personal distribution and test it.

6.3 The Network

In today's systems that grow more and more dependent of communication either between them or with the internet network and it's efficiency take a more important role. We decided to also test it and see how a Virtual Machine Managers share it and manage this resource.

Unlike the CPU and memory tests we used iperf to test memory and it gives its results in Megabits/Second so the bigger the value is the better.

The vertical axis will be in Mbits/sec and the Horizontal axis will represent the number of Virtual Machines started.

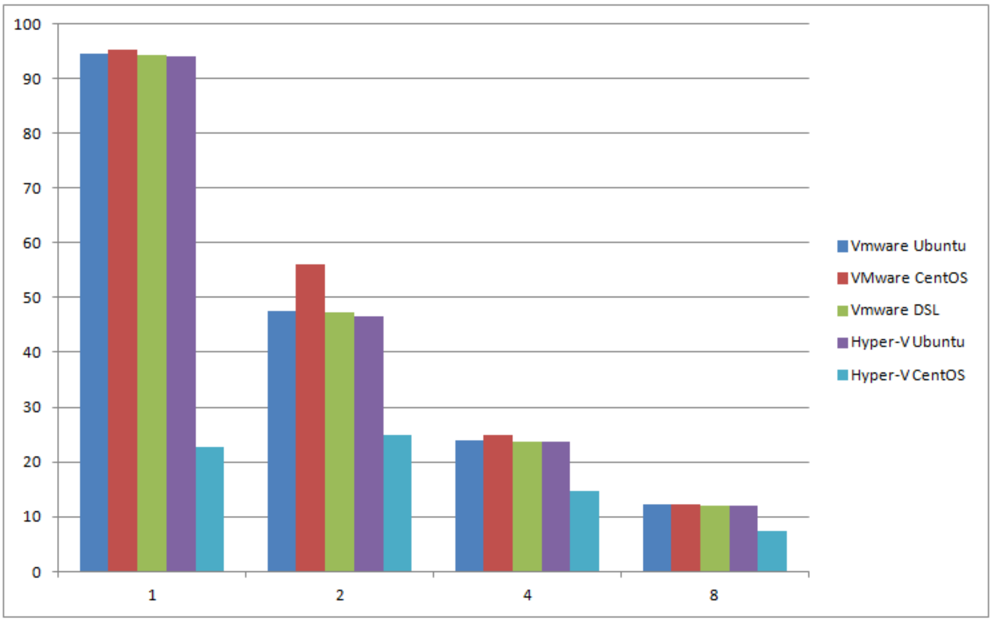

Fig. 21 Network Test – Hypervisor/OS Comparison

Figure 21 represents the bandwidth for all the configurations we have tested. It is easy to see that the drop in bandwidth is linear with the number of machines no matter what OS or hypervisor is used. CentOS on Hyper-V stands out because it uses a Legacy Virtual Network drive because CentOS lacks the drivers for the standard virtual network card which Hyper-V provides.

With the exception of CentOS on Hyper-V all the other configurations work almost the same way, the actual differences between them being negligible.

It is important to notice that all the time no matter how many VMs we start the system manages to use the entire bandwidth.

When it comes to bandwidth there is no clear winner. Neither the Hypervisors or the OSs seem to make any major difference on the end result. They all share the bandwidth in a fair manner and they all try to use the entire bandwidth made available.

This shows that a user that just wants to run a web server or anything that is not processor intensive but needs a strong internet connection and a stable one, can move to the cloud and use whatever Operating System the user is more comfortable with and freely choose any cloud provider regardless of the hypervisor they offer.

7. Cloud Results

We run the same tests we did on our hardware on two Linode Virtual Machines. As we said before Linode runs there Cloud on top of Xen. They offer packages with the following configuration:

Intel Xeon L5520 Quad Core 2.27 Ghz
512MB
200GB Transfer
20GB Hard Drive Space

Linode offers all the 4 Cores of the CPU to each VM but they share all the cores between multiple VMs. There are also other configurations available with more RAM, more transfer and more Hard Drive Space.

We tested Ubuntu and CentOS on the Linode Cloud as they offered both OSs already installed and configured. The tests were done on both 32 bit and 64 bit versions of both Operating Systems.

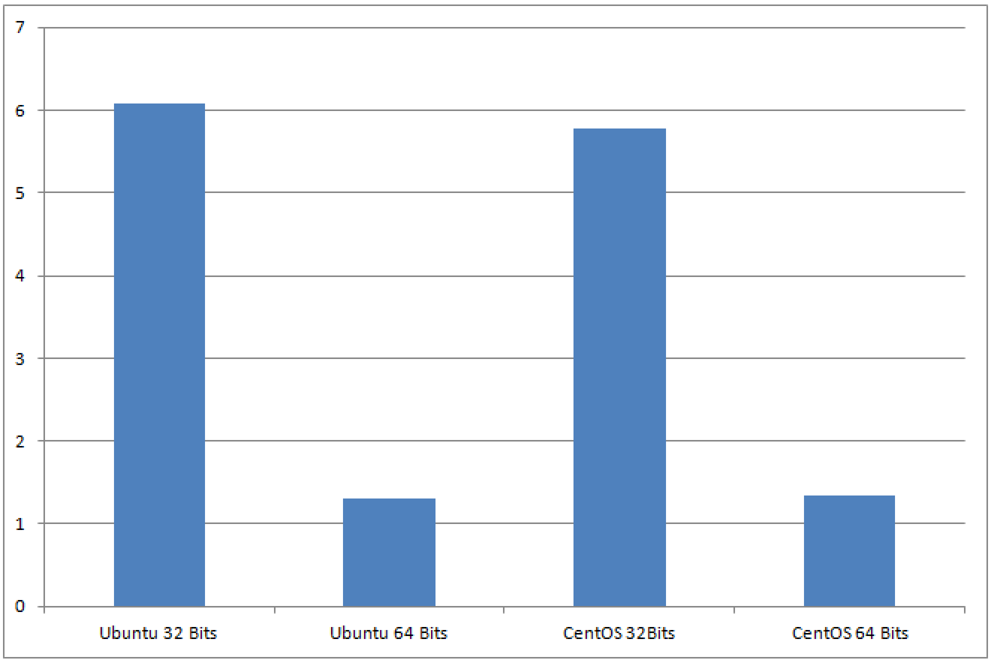

Fig. 22 Cloud – CPU Test

In figure 22 we have the results from the cloud tests. The results are in seconds and they represent execution time of the same program which we used to test on our local physical machine. The smaller the results are the better.

We can clearly see how even though both operating systems have basically the same results there is a clear difference between the 32 and 64 bits versions of both OSs. The 64 bit version has a clear advantage being over 3 times faster than the 32 bits version ones. Extrapolating this results we notice that this is the same reason why on our local tests DSL was a lot slower than Ubuntu and CentOS as we tested Ubuntu and CentOS with 64 bit versions and DSL only offers a 32 bit version.

The results we got for the memory tests were almost identical, this confirms to us that the difference in execution time for the CPU tests were made by the 32/64 bit kernels we were using and not by any special characteristics of the 2 distributions.

Keeping this in mind we have to consider how much DSL would improve if one would change its kernel to a 64 bit version, thus confirming our theory that a mini distribution can have an interesting effect on the cloud.

During our tests on the cloud we have left the tests running for 24 hours each in a try to see how other virtual machines would affect our results. After testing both memory and CPU on both Ubuntu and CentOS each for 24 hours we have noticed no difference in the execution time of our programs neither in the memory tests or the CPU tests.

We have also tested the network and we got a transfer rate of about 49 MBits/sec. Unfortunately we cannot interpret this result as we do not know what the network inside the Linode cluster looks like or how far, from a network perspective, the 2 Virtual Machines were from one another.

We do have to mention that Linode does not consider traffic inside their network to the cost. So renting more virtual machines and constructing a small virtual cluster inside their cloud is possible and the entire communication between the individual systems is going to be free.

They also let one use the processor all the way to 100% and try to accommodate any such special needs.

To end we must say that we were impressed with the simplicity and the speed with which we set up a Linode VM and how simple it was to use. This combined with popular distributions like Ubuntu makes it very simple for new inexperienced persons find it more comfortable to start using the cloud.

We consider that the offer given by cloud providers like Linode can please anyone from a standard user that needs just a VM for something like a web server to anyone interested in high performance computing.

One should always consider the price in its decision and renting space on the cloud is not something that can easily be consider as being minimal.

8. Future Plans

Next there are still a lot of things that still need doing. We find it important to find a physical system that would permit us to also use the Xen Hypervisor and test it the same way we tested VMware ESXi and Hyper-V.

We also find it important to study the effects of a small distribution on the cloud in much more detail. This might mean actually trying to build one based on the freely available Linux Kernel. Or try to continue the work started by the group of people who created DSL. Trying such systems should be done both on physical machines and on the cloud. This is made possible in systems like the one offered by Linode where one can upload its own Linux Distribution.

Tests should also be extended to see the efficiency impact on more special CPU features like SSE and special hardware like GPUs. This can be done both on available physical machines and on the cloud offered by Amazon.

We also believe that tests like the one we did on the cloud to analyze the long term effects caused by VM migration or by other VMs could give a better insight on cloud stability. This should probably be done on multiple clouds and in multiple system configuration. The tests should also be run for long periods of time, like weeks or months. In tests like this communication with the cloud provider should be a priority.

The difference between 32 bit and 64 bit kernels should also be analyzed in a bit more detail, the efficiency advantage given by 64 bit systems seeming a bit too impressive.

9. Conclusion

During our tests we made a bunch of interesting discoveries. We noticed the effect of the kernel has on CPU speed. We noticed how VMMs can share the CPU, the Memory and the Bandwidth in a fair and efficient way, no matter what Hypervisor we used.

We must say we were impressed with the results from Hyper-V which even thou it was somewhat slower also run Windows Server 2008 and still had comparable results with VMware ESXi. On the other hand we were pleased with VMware ESXi in the way that it had no difficulty in running any OS we provided.

When it comes to the Operating System we have to recommend Ubuntu, not just because of the results but because it is more popular, simpler to use and the popularity that comes with it makes it a more up to date system with more applications available than any other distribution.

It is important to keep in mind the power that could be brought by a possible mini distribution, not only in efficiency but even in cases where space reduction is a priority. We do not believe that DSL is the answer but it does represent an interesting idea, one that should be explored more in the future.

The results we got from the cloud also showed us how stable a cloud system can be and that other VMs executing on the same physical space is not as much an issue as one would think. Still when using cloud services one should always expect rise and drop in performance. As our tests show it is impossible to run more virtual machine on the same hardware and expect no drop in performance.

Both virtual machine hypervisors and Operating System still have a lot of parts that could use improving and any boost in any of these could help on the long term efficiency of systems like the cloud.

BIBLIOGRAPHY

[1] Paradox Of Choice: Why More is Less – Barry Schwartz – Eccon 2004)
[2] Virtualization 100 Success Secrets 100 Most asked questions on Server and Desktop Virtualization, Thinapp Software, SAN, Windows and Vista Applications – Michael James - Lulu.com, 2008 - Pg. 134
[3] amd.com
[4] wikipedia.org
[5] msdn.microsoft.com
[6] Windows Server 2008 Hyper-V Insiders guide to Microsoft's Hypervisor – John Kelbley – Ed. John Wiley & Sons, 2011 – Chapter 5
[7] Xen Virtualization – Prabhkar Chaganti - Packt Publishing Ltd, 2007 – Chapter 1
[8] veillard.com
[9] Modern Operating Systems – Andrew S. Tanenbaum – 3rd Edition – Pg. 3
[10] centos.org
[11] ubuntu.com
[12] damnsmalllinux.org