2011

Service level agreements and virtual machines

Jordan Caminiti

Service Level Agreements and Virtual Machines

by

Jordan Caminiti

Committee Members

Charlie Border

Tae Oh

Luther Troell

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Networking and System Administration

Rochester Institute of Technology

B. Thomas Golisano College

of

Computing and Information Sciences

02/11/2011

Abstract

Virtual machines (VMs) have been used for some time now, but only now with the newer and faster hardware that is being developed, now it is possible to consolidate many machines down to a single one running multiple instances of operating systems each with their own purpose. For as long as there have been users to use the servers, there has been the need for service level agreements. Now with the virtualization of services comes the need for a new breed of service level agreements.

Service level agreements (SLAs) rarely exist for virtual servers unlike physical servers. Since this is the case, an effective SLA needs to be developed so the users of the virtual machines can know and be guaranteed their service like they were with the physical server. Questions such as, is it possible to take an existing service level agreement with the metrics from a physical server and transfer it directly to the virtual server? Are there changes that need to be made to the SLA in order for the SLA to be functional? These questions will be addressed in this study, along with a discussion on why changes would need to be made when moving a SLA to a virtual environment from a physical server.

It is indeed possible to take most of the preexisting service level agreement that was written for a physical server and with modifications to metrics of CPU usage and memory utilization use it in a virtual environment. Additional metrics need to be made to the SLA when it is transferred to the virtual environment in order for the SLA to work effectively. These new metrics were found, tested and shown to be necessary in a virtual environment.

Table of Contents

1. Introduction

In light of the current economy, companies are consolidating physical servers, using virtual machines, to reduce energy and equipment costs. Parallels and VMware are two of the more popular hypervisor packages available. Qualities such as ease of use among different operating systems, and a large user-base resulting in high quality support of the software make the aforementioned packages logistical selections when compared to lesser known products. Selecting the type of virtualization to be used is another component of consolidating physical servers using virtual machines (VM). Two of the more recognized styles of virtualization are para-virtualization and full virtualization. Full virtualization will be used in this study and will be discussed briefly later in this paper.

All software and hardware implementations in order to be properly managed and used should be governed by an service level agreement (SLA). Physical machines have certain metrics that are able to be accurately analyzed, and thus, are incorporated into a service level agreement. When a server is virtualized memory usage, CPU usage, hard drive utilization are all split among different virtual servers the integrity of the metrics that were once accurate representations of the server's efficiency are now compromised. This poses the question: How are the metrics of virtual servers to be assessed, and incorporated into SLAs relevant to the current technology? Does one look from the inside of the virtual server outward to receive feedback from the virtual system about its usage? Would the metrics need to be examined on the server performing the hosting functions for the virtual servers? Could the data be collected externally, rather than internally?

Currently for virtual servers, the problem exists that the method of collecting the metrics that was once used is no longer reliable, and thus “by and large SLAs are not available today” [4]. The providers of virtual products do not know what standards VMs need to meet, there is no requirement for uptime, or realistic expectations regarding the time involved to service the VM, should it fall below acceptable operating status. VMs are simply expected to work uninterrupted. This is an unrealistic goal, and the creators of service level agreements now have to determine new methods to measure performance, resulting in new standards of quality for their products. If a server starts to respond slowly or not at all, who is to say that the level of service is unacceptable. The previous concerns mandate the development of a virtual service level agreement to insure that the service providers, and the users, are aware of what performance can be expected from the hardware to which they are converting. Once a baseline has been set up for the VM SLA it can then be adapted to meet whatever the requirements are for the specific service that is going to be virtualized.

2. Related works

Three topics of research will be discussed, in order to achieve the ultimate goal of this thesis. First, research that was completed in the realm of VMs, including metrics, performance, and other components that are directly related to the creation of a VM service level agreement. Second how the creation, and management, of SLAs for physical machines can be utilized to create, and manage, SLAs for VMs. Third, optimizing, and monitoring, the new SLA format to make the SLA a more useful tool for all parties involved in the contract.

Begnum, Disney, Frisch and Mevåg, discussed three ways to look at how virtual machines are organized in a network, which created new methods to analyze the implementation of virtual machines. These three methods are: "server redundancy level, resource conflict matrix and location conflict table" [1] Begnum, et al. devised these methods because they observed a lack of "general and vendor-independent quantitative criteria/metrics" [1]. Server redundancy level, according to the authors, is a notation of server capacity, R/S, where R is the number of servers currently in use, and S is the number of servers that can be eliminated. This concept is based upon the availability of space to reconfigure the VM's for the case of, as the notation is called, redundancy. This concept could be adapted for use in a VM service level agreement. In order to insure the proper uptime, a good server redundancy level needs to be maintained. Resource conflict matrix, which was another method mentioned in the paper, involves graphing out potential conflicts in resources that the VM's use. The basic premise indicates that the fewer conflicts evident, the better the VM's will perform. This matrix can determine the most efficient way to redistribute the VM's amongst servers, by determining where the least amount of conflicts will occur. While this concept may be useful to the system administrator, it would not likely be included in a SLA. The third method by which to analyze VM's, the location conflict table, is related to the previous resource conflict matrix. Similar to the resource conflict matrix, the location conflict table is a graph which provides the system administrator an easier view of resource conflicts located in the VM network. While applicable when deciding where, and how, to reposition VM's, it too is not necessarily an item that would be useful in a VM SLA implementation.

The article from Computer World addresses a valid point about virtualizing, and consolidating, servers. It states "In some cases, end users object to virtualization because they're

concerned that virtual machines lack the security and performance of dedicated servers"[2] The goal when creating a SLA for VMs should be to alleviate the stress and concerns that the end users have about the new virtualized servers. If the users have contract documents, and security, that guarantee a certain level of performance then it will lessen most of their concerns about a move to virtualization. With system administrators using phrases such as "I guarantee Service-level agreements"[2], shows the urgency for creation of new SLA's for VM's. Also, with companies starting to purchase larger more powerful servers just to accommodate the future virtualization, like SAIC, "replacing 300 physical servers with 20 servers hosting virtual machines"[2], it is becoming imperative that a service level agreement specifically designed for a virtual server is established.

Throughout A Performance Evaluation Methodology in Virtual Environments, the authors propose a new metric which could be used when creating a SLA for VM's. This further proves the point regarding the difference in the metrics that are used to measure the performance of VM's and non virtualized servers. The article goes on to define the new metric as the normalized waste rate of the CPU. The author gives the example: "Then, the utilization of *(VM host)* is 50% after the consolidation, and the consolidated system *(VM host)* wastes only 40% of capacity, instead of 75%"[3] This concept of waste rate is novel for system administrators, or managers who have been creating SLA's. When drafting a SLA to include this new metric, a fine balance will need to be established between decreasing the waste rate of the VM host CPU, while avoiding the point where performance is affected. The article tests server consolidation, and measures throughput, on consolidation of two, and four servers. The data collected indicates that an increase in systems consolidated decreases the waste rate of the CPU. This evidence

provides another reason why the SLA's that exist for current physical servers need to be revised if they are to be relevant to this new generation of virtual servers.

Zhang et al. delves into the ability of VMs by studying the I/O performance of the virtual machine. Studies like this prove a difference exists between the way a VM works versus an OS hosted on a server. The prior statement exemplifies that current SLA's, written for physical machines, need to be modified in order to make them relevant for the new virtual world in which they exist. Zhang and his colleagues, state that "OS in VM owns a more complicated storage hierarchy, which contains the OS cache, VM cache and disk buffer."[5] This thesis will examine if a SLA that has been developed for physical hardware is still applicable in a virtual environment. Portions of current SLAs are irrelevant, and this poses a problem for everyone that relies on the SLA.

Cluster management in a virtualized server Environment [6] explains a way to increase the performance of servers using VM. The concept of cluster management exists in the physical world, however porting this knowledge to the virtual world is a newer idea. Park et al. presents three concepts about virtualization, virtualizing up, virtualizing down, and virtualizing up and down. Virtualizing up involves taking two or more servers and combining them using a VM to create a more powerful server than either was prior. Virtualizing down is similar to server consolidation as it takes multiple servers and places them on separate VMs, but all on the same server. The best example of virtualizing up and down is in the article: "there are three different physical servers available. But two applications that need to be hosted require more capacity than one server's, but less than twice . . . using virtualizing up only, one application is over-provisioned and the other is under-provisioned, but the combination of the two types of virtualization should provide proper provision for the requirements."[6] Each method has their

own place in the network, and all lead to lower server costs. Virtualizing down is currently the most popular management style, of the three, as it decreases physical cost and energy use. Managing the VMs is just one part of the equation though; the second half is the ability to provide services with these managed VMs. The research being conducted will confirm the proper protocol for VM management, thus resulting in more effective drafting of SLAs. The Advanced Communication Technology article investigates creation and management of VM clusters, while lacking a standardized consistency of services, as a result of virtualizing up, down or both. The idea of being able to provide consistent service is something that is left by the wayside.

Coordinating the elements of an SLA to provide maximum effectiveness is challenging, however the study that was done by Bouman, Trienekens and Van der Zwan [7] clarifies the task. The study reveals seven different lessons that were learned that can be applied to any SLA that is being developed. The first lesson discussed was: "Decide at an early stage what the serviced objects are, seen from the eyes of the user, and determine how this will be reflected in the document structure of the SLA". [7] This first lesson mentions the serviced object, which in this case of SLAs for VMs are the individual metrics that surround the VM, its performance and availability, as well as any special metrics that will only be applicable to a VM.

The second lesson in the article is "Try to untangle components of the needed service(s) and focus on these parts, rather than on the whole service". [7] This particular lesson can be applied to the topic at hand because this type of SLA has not had much coverage. The ability to break down the individual metrics, and look at them on a case by case basis will aid in the SLA being well planned, accepted and followed. If the service is observed as a whole, then vital portions of the SLA may not be covered as thoroughly. The lack of attention toward components

creates the potential for an SLA to be designed, yet useless, and ultimately forgotten. This leads into the third lesson; “Create a readable and easy to adapt document, by including descriptions of taken decisions on both document structures and services.” [7] This lesson reflects the first notion that the SLA needs to be understandable by everyone; it needs to be written in as plain of English as possible. It needs to be understood by non-technical people, and avoiding jargon will make a document, such as a new SLA, easier to implement than one that is written in confusing tech-speak. These are the lessons presented in the study that were relevant to the creation of any SLA, and subsequently relevant material to this thesis.

In the paper titled “Precise Service Level Agreements”[11], Skene, Lamanna and Emmerich introduce a streamlined approach to precisely defining a SLA. They have created an outline for a new language SLAng. This modeling language has the potential to revolutionize the way that SLA’s are created, monitored and kept current. The authors differentiate themselves from others that have attempted to design a SLA modeling language by using a slightly different approach to the language.

First, “in contrast with other languages that focus on web services exclusively, it defines SLA vocabulary for a spectrum of Internet services, including Application Service Provision (ASP), Internet Service Provision (ISP), Storage Service Provision (SSP) and component hosting.”[11] The authors of SLAng decided against verbiage that is very narrow in scope, in liew of a broad language that could be applied by the general public. SLAng has the potential to bridge the technological gap and use an SLA that was developed for a physical server with a VM. The ability to make changes to the SLA, without having to completely rewrite it, could save time and meet the needs of both the user and provider.

Second SLAng differentiates itself from other SLA modeling languages in that it is "derived from industrial requirements" [11]. This enhances SLAng as a useful tool when developing a SLA, because it already understands the needs of the users and providers. The use of SLAng could unify all SLA's under a common language, in order to increase the understanding of content disclosed within the SLA. Having a common language such as SLAng would help all parties involved in the SLA, by using consistant terms throughout all SLA's. The end user will better understand what they are agreeing to, and the provider of the service will receive fewer customer complaints, when the terms of the contract are clearly stated. While SLAng is not a fully functional language at present, as stated on their website [12] it could lead to an invaluable tool in the creation and modification of SLA's.

Proper monitoring of the requirements in an SLA should be addressed with a customer seeking service. The authors of "The Monitorability of Service-Level Agreements for Application-Service Provision" [13] discuss monitoring SLA's, whether written for a VM or a physical server. Being able to properly monitor the requirements of the SLA is a very important topic and is one that when creating the SLA with the customer it should be addressed. The paper discusses a specific type of monitoring that can be put into the SLA. According to Skene et al, "Monitorability of an SLA is classified as unmonitorable, monitorable by one party, mutually monitorable by both parties, or arbitratable"[13]. Components in an SLA that can be classified as unmonitorable are not desired as a component that cannot be monitored would leave the provider unable to substantiate any claims against that specific component. Components that are monitorable by one party are equally undesirable as unmonitorable components should a dispute arise regarding the implementation of the SLA. Mutually monitorable, and arbitratable components, are two successful approaches for monitoring the agreed upon conditions. The

information presented in the paper [13] is easily applicable to any SLA which makes it applicable regardless of the medium. Monitorability is a key element to consider when attempting to create a new breed of SLA, particularly for a virtual server, because the rules of what and where to monitor have changed and they need to be considered when writing this new SLA.

Raimondi, Skene and Emmerich continute to probe options to monitor SLAs that are covering web services. They address the topic of online, versus offline, monitoring of SLA's. They claim that online monitoring of an SLA is superior to that of its offline counterpart for a variety of reasons. "Firstly, it requires the storage of possibly large volumes of data about the service provision. Secondly, for triggering software architecture reconfigurations or resource allocation decisions in data centres it is necessary to know about service quality violations as soon as they happen."[14] They determine that because of the aforementioned reasons, they will, for the sake of that study, only be looking at online monitoring of SLA's. It is logical to use online monitoring of the web services, and online monitoring will be used exclusively in this thesis as well. The authors of "Efficient online monitoring of web-service SLAs" [14] used an Eclipse plug in and Apache AXIS handlers to monitor, in real time the SLA's in their case study. This thesis will vary, and Nagios will be used to monitor the different performance metrics in the SLA's.

The Raimondi et al. study goes on to describe the requirements necessary order to effectively monitor, and insure completion, of the duties enumerated in the SLA. Latency requirements, reliability requirements, and throughput requirements are all major components that should be considered when forming any service-oriented SLA. All three of these

requirements will be necessary in this thesis in order to determine if changes are essential to an SLA when moving to a VM hosted service.

Raimondi et al. defines the above terms in the following manner. Latency requirements are defined as “the response of the service must follow the request within t seconds” [14]. A metric clearly stating on acceptable response latency will insure customer satisfaction and implement guidelines for provider accountability. Latency is an important monitoring requirement as it is a measureable unit of time that can be mutually monitored by both the user, and the provider, of the service. Reliability requirements are defined as “Another set of requirements for services includes constraints on the number of acceptable errors. Errors are defined as violations of the latency requirements or as other kinds of timeouts.”[14]. A reliability metric allows concrete numbers to be used when trying to claim a breach of the contract. It is quantifiable, and mutually monitorable, thus making it an effective metric for use in monitoring an SLA. The third requirement of an SLA, in order to be useful according to the authors of the article, is throughput requirements. The authors of the paper define it as: “restrictions on the number of requests that a client is allowed to submit in a given time window.”[14]. This portion of the SLA is used by the provider to insure the customer a specified level of quality of service that is mutually monitorable, and quantifiable. These three requirements will be crucial sections of to the SLA’s that will be developed for this thesis. Revision of the afore mentioned metrics and requirements may occur in order to create an effective SLA for a service that is provided by a VM.

3. Virtualization

As stated previously, various options exist to virtualize OS, or individual application software. Full virtualization implementation was selected from the various possibilities, to evaluate the SLA devised for this thesis. As with any virtualization technique there are assets as well as detriments and full virtualization strikes a balance; leading it to be a recognizable technique when server consolidation is discussed.

3.1 Full Virtualization

Full virtualization exists when is a complete operating system is installed on the physical server, and a program is installed and run on that operating system which contains the virtualized system. For example, the base server is a Windows 2003 server, and once windows fully boots up a program such as VMware is run. VMware is now the program on which the guest operating system will run. Once VMware is started, the virtualized system is booted up and once booted acts as its own separate system even though it is sharing all of the physical hardware with the host operating system. As with any piece of software, full virtualization has its advantages, as well as its disadvantages. Its main asset is also one of its primary shortcomings, which is that the VM is seen to the host OS as a program. Any calls that need to be made to any of the hardware are completed through the host operating system. This is a limitation for full virtualization because there is greater performance over-head due to passing of information from the VM, through the hypervisor, then through the host, to the hardware. This represents a simultaneous benefit as the VM is nothing more than a program, not an entire OS, that is running on a server;

therefore the VM is able to be moved from one server to another without being shut down. This allows for physical server updates to occur without the need to bring down the hosted service, which provides for greater uptime and flexibility.

3.2 Parts of full Virtualization

The key components to a full virtualization implementation are the virtual machine, the virtual machine manager, the host operating system and the hardware that runs the aforementioned elements. This section will provide a brief overview of each of these components.

3.2.1 VM

The virtual machine is a software implementation of an operating system. It is installed and run through a virtual machine manager (VMM) or hypervisor. Depending on the type of virtualization that is running, the VM may or may not know that it is a virtual system. In a full virtualization implementation the VM is not aware that it is a virtual system, and operates independently of its host OS. Any system calls that the VM needs to complete for any system resources are delivered to the VMM, passed to the host computers OS, and finally received at the hardware, the results are sent back in reverse. The VM perceives the interaction as a direct connection to the system hardware.

3.2.2 Hypervisor

The ability of the VM to operate seamlessly with the host OS is due to the virtual machine manager, or hypervisor. In the case of this thesis, VMware will be used for full virtualization. VMware uses binary translation and direct system calls to accomplish full virtualization. The VMM "translates kernel code to replace nonvirtualizable instructions with new sequences of instructions that have the intended effect on the virtual hardware" [8]. The VMM is the essential for virtualization and is present in various implementations of virtualization. In addition to passing on all of the information that the VM needs to the host OS, the VMM "is the software responsible for hosting and managing all virtual machines" [8]

3.2.3 Host Operating System & Hardware

The host operating system is not uniquely different when hosting a virtualized system versus a standalone server. The host operating system of a virtualized server(s) does not recognize that there are any changes to its system. The only change that may occur in the hardware of the host system would be an increase in system resources in order to better server multiple machines running on a single set of hardware. Otherwise the hardware is unchanged, no specialized hardware is needed.

4. Service Level Agreements

Service level agreements are contracts established between the provider of a service, which can be the IT department of a company or an internet service provider, and the end user of

the provided service. Assisting the IT department in managing their resources, and protecting the users of the service are two primary reasons SLAs are established. The SLA should be written in as plain of English as possible both the provider and the user of the service can comprehend the terms in the agreement. A superior SLA is written sans loopholes for either the providers, or the user, to use to undermine the effectiveness of the SLA.

4.1 Protect the provider

This contract enables the provider of the services to be more efficient in the ways their servers run particular services. The agreed upon SLA provides guidelines and metrics that need to be adhered to in order for the contract to remain valid. Creating and abiding by these rules improves the providers awareness of their machines performance, as well as any problems with their implementations. In order to know when the servers are or are not meeting the agreement, monitoring needs to be established to track the status of the resources. This monitoring can be manual, where a systems administrator assesses the metrics by hand, or it could be performed automatically by software which has been configured to evaluate the individual metrics defined by the SLA. Proper monitoring allows the system administrators the time to find, and correct, the resources not meeting the standards contained in the SLA before a potential server crash, or system loss. Monitoring is part of an SLA to maintain appropriate system function, however, an additional benefit occurs when due to proper monitoring, a provider can refute a breach of contract claim by a user.

4.2 Protect the user

User of the services benefit when a contract is established to govern daily technology function. By agreeing to a set level of service the users are able to determine whether they are receiving the level of service for which payment is rendered. When an error occurs they are able to reference the SLA and determine what the course of action should be and who is responsible. In order to properly protect users of the SLA, the monitoring system that is established should be easily accessible to the user.

4.3 Components

SLAs can differ in components. A typical service level agreement contains: services delivered, performance management, problem management, customer Duties, warranties & remedies, security, disaster recovery, and termination. [9] For this thesis, performance management will be examined exclusively. Performance management can be divided into two sub categories, performance metrics and monitoring. The following sections will go into brief detail about metrics and monitoring.

4.3.1 Performance and Availability Metrics

These metrics are created in order to determine whether if a service is performing up to standards. According to Network World, “Loosely all metrics can be grouped under the headings of either availability or performance.” [10] For the sake of this thesis, this portion of the

SLA will be the main concentration of the testing. The metrics devised will be of a general nature to be applicable to all types of servers, regardless of the services that are hosted on them.

4.3.2 Performance Monitoring

Without proper monitoring, the metrics defined in the SLA have no real value. The second piece of developing an SLA is effective monitoring. A provider should create the metrics that meet both parties needs, then determine a way to substantiate that the metrics are being accomplished. The monitoring must be able to be accessed by both parties, the provider and the user, and modified by neither party. If either party is able to modify the data, the findings reported become irrelevant. The trust established in the formation of the SLA contract would be irradiated by the tampered data, and the SLA voided.

5. Physical Server SLA Development

The development of SLA metrics to assess in the virtual environment required an SLA for the physical server to be created first. This was accomplished by connecting the two test machines, each running a packet generator which sent traffic to both the web service as well as the FTP service. In addition to the packet generator, Microsoft Windows Media Server Load Simulator was used to evaluate the Windows Media Server (WMS). In order to monitor server performance, an additional computer with Nagios monitoring software installed, was connected to the server. The software would report an OK state, a warning state, or a critical state based upon the values in the SLA. The traffic generators were initially set to simulate the following number of users: 100 users for FTP, 100 users for web traffic, and 100 users for WMS, to create

a baseline for the SLA. After the traffic generators ran using the initial parameters, none of the services would respond from within Windows, or Nagios. The CPU was reporting to have 100% utilization from within Windows and >99% from within Nagios. The number of users was lowered by 10 for one service at a time until the server became viable. The final average number of users for each service was then established at 10 for FTP, 40 for web traffic, and 40 for WMS. These numbers would become the guaranteed numbers for the physical server SLA, allowing for a 50% increase over the guaranteed number before the server would exceed capacity and stop responding. Nagios would report each service in an OK state until a 25% increase would trigger a warning, and an increase of 50% would trigger a critical state.

Additional metrics were monitored during the aforementioned testing, in pursuit of a thorough examination of SLA components. The metrics observed by the monitoring software included: CPU utilization, drive space and individual folder size, memory usage (physical, virtual and paged), network utilization (input and output), packets per second (input and output) and the response time in seconds of the web service and FTP service, and finally, if the WMS service was still in a running state on the server. In order to generate an average for these metrics, the traffic generators were run as before, setting the number of users to that which was established above. The data collected would be the guaranteed numbers entered into the SLA, thus becoming the OK level for Nagios. Similarly to the user test, a report 25% above the OK level would deliver a warning to the Nagios system, and 50% above the OK level would report a critical level.

5.1 Physical Server SLA

The baseline data outcomes, derived from testing in the physical server SLA development section above, resulted in the metrics that are detailed below. The physical server SLA will be used as the basis for the virtual server SLA, thus making the development of a cohesive, and comprehensive, set of metrics imperative to this study.

The machine at IP address 10.100.100.1 will be able to perform up to the following metrics, which will be monitored by the Nagios server at address 10.100.100.4. All parties involved in this service level agreement will have access to the Nagios service summary page in order to track individual components of the server. In addition to manually checking the services within Nagios, if any of the services detailed below reach a critical level, both customer and the provider will be automatically emailed by the Nagios server, and corrective action taken.

For use in this service level agreement, the term service will be defined as the individual instance of the monitoring software that shows the state of the object being monitored. The service can be in only one of three states at any one point in time; OK, Warning, or Critical.

1. Drive space

The server will posses multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 HTTP: The folder containing the data for this service will be capped at 3 Gigabytes, and a warning will be activated on the monitoring software when the data level reaches 2 Gigabytes. The service will convey a critical state when the data level reaches 2.5 Gigabytes.

1.2 FTP and File storage: Data in the folder for these services will be capped at 10 Gigabytes. A warning will be activated on the monitoring software when the data level reaches 8 Gigabytes. The service will relay a critical state when the data level reaches 9.5 Gigabytes.

1.3 Windows Media Server: The folder storing the data for this service will be capped at 5 Gigabytes. A warning will be activated on the monitoring software when the data level reaches 3.5 Gigabytes. The service will be marked as critical when the folder containing the data exceeds 4.5 Gigabytes.

1.5 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been utilized. The service will enter a critical state when the drive reaches 90% capacity.

2. CPU

The maximum guaranteed CPU usage is 75%. Nagios will activate a warning when the CPU exceeds 75% utilization. The service will enter a critical state when the CPU exceeds 90% utilization.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the Virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

5. Services Provided:

The server will be providing the customer four different functions: FTP server, File server, Web server, and a Windows Media Server. Each of these services tasks their own metrics detailed below.

5.1 FTP and File server:

5.1.1 Response time: The maximum guaranteed response time is .5 seconds. A warning will be activated on the monitoring software when the response time exceeds .5 seconds. The service will enter a critical state when the response time exceeds 1 second.

5.1.2 Connected users: The maximum guaranteed number of connected users is 10. A warning will be activated on the monitoring software when the number of currently connected users exceeds 10. The service will enter a critical state when more than 15 users are connected.

5.2 Web server

5.2.1 Response time: The maximum guaranteed response time is .5 seconds. A warning will be activated on the monitoring software when the response time exceeds .5 seconds. The service will enter a critical state when the response time exceeds 1 second.

5.2.2 Connected users: The maximum number of connected users is 40. A warning will be activated on the monitoring software when the number of currently connected users exceeds 50. The service will enter a critical state when more than 60 users are connected.

5.3 Windows media server

5.3.1 Service running: The service will be accessible 99% of the time that the server is operating. A warning will be activated on the monitoring software when the service cannot be contacted for .5% of the time the server is running. The monitoring system will enter a critical state when the service cannot be contacted, or the time that the service has been down, exceeds 1% of the uptime of the server.

5.3.2 Connected users: The maximum number of connected users is 40. A warning will be activated on the monitoring software when the number of currently connected users exceeds 50. The service will enter a critical state when more than 60 users are connected.

5.2 Virtual Server SLA Development

Testing the feasibility of the metrics retaining from the physical server to the virtual servers was the goal of this thesis; thus, the metrics from the physical server SLA needed to be used in the creation of a SLA to impose on VMs. The physical server SLA contains information about three different services, (FTP, web and WMS) which needed to be separated into their components in the virtual SLA. Each of the services that were provided by the physical server would now be hosted by its own virtual server. The VM SLA would need to address four servers, each performing independent functions, yet still use identical metrics found in the physical server SLA. The VM SLA was broken down into four main sections, each section labeled by the server

IP address and function. The common metrics, from the physical server SLA were inserted to the corresponding sections of the VM SLA; drive space, CPU utilization, memory usage, and network utilization. The metrics specific to one particular service, on the physical server were copied only to the section whose server it belonged to in the virtual environment.

5.3 Virtual Machine Service Level Agreement

The machines at IP addresses 10.100.100.2(VM Host), 10.100.100.10(Windows Media Server), 10.100.100.11(HTTP Server) and 10.100.100.12(FTP and File Server) will be able to perform up to the following metrics, which will be monitored by the Nagios server at address 10.100.100.4. All parties involved in this service level agreement will have access to the Nagios service summary page in order to track individual components of the server. In addition to manually checking the services within Nagios, if any of the services detailed below reach a critical level, both customer and provider will be automatically emailed by the Nagios server, and corrective action taken.

For use in this service level agreement, the term service will be defined as the individual instance of the monitoring software that shows the state of the object being monitored. The service can be in only one of three states at any one point in time; OK, Warning, or Critical.

10.100.100.2(VM Host)

1. Drive space

The server will posses multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been utilized. The service will enter a critical state when the drive reaches 90% capacity.

2. CPU

The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU exceeds 75% utilization. The service will enter a critical state when the CPU exceeds 90% utilization.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches a 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches a 60% usage. The service will enter a critical state when the physical memory reaches a 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

10.100.100.10(Windows Media Server)

1. Drive space

The server will posses multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been used. The service will enter a critical state when the drive reaches 90% capacity.

1.2 Windows Media Server: Data in the folder for this service will be capped at 5 Gigabytes. A warning will be activated on the monitoring software when the data level reaches 3.5 Gigabytes. The service will be marked as critical when the folder exceeds 4.5 Gigabytes.

2. CPU

The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU usage exceeds 75%. The service will enter a critical state when the CPU usage exceeds 90%.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

5. Service Provided:

The server at 10.100.100.10 is a Windows Media Server, whose metrics are detailed below.

5.1 Service running: The service will be accessible 99% of the time that the server is operating. A warning will be activated on the monitoring software when the service cannot be

contacted for .5% of the time the server is running. The monitoring system will enter a critical state when the service cannot be contacted, or the time that the service has been down, exceeds 1% of the uptime of the server.

5.2 Connected users: The maximum number of connected users is 40. A warning will be activated on the monitoring software when the number of currently connected users exceeds 40. The service will enter a critical state when more than 60 users are connected.

10.100.100.11(Web Server)

1. Drive space

The server will display multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 HTTP: The folder containing the data for this service will be capped at 3 Gigabytes. A warning will be activated on the monitoring software when the data level reaches 2 Gigabytes, and the service will be marked as critical when the data level reaches 2.5 Gigabytes.

1.1 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been used. The service will enter a critical state when the drive reaches 90% capacity.

2. CPU

The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU usage exceeds 75%. The service will enter a critical state when the CPU exceeds 90% utilization.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

5. Service Provided:

The server at 10.100.100.10 is a Webserver, whose metrics are detailed below.

5.1 Response time: The maximum guaranteed response time is .5 seconds. A warning will be activated on the monitoring software when the response time exceeds .5 seconds. The service will enter a critical state when the response time exceeds 1 second.

5.2 Connected users: The maximum number of connected users is 40. A warning will be activated on the monitoring software when the number of currently connected users exceeds 40. The service will enter a critical state when more than 60 users are connected.

10.100.100.12 (FTP and File server)

1. Drive space

The server will present multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 FTP and File storage: The folder containing the data for these two services will be capped at 10 Gigabytes. A warning will be activated on the monitoring software when the data level reaches 8 Gigabytes. The service will reach a critical state when the data level reaches 9.5 Gigabytes.

1.2 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been used. The service will enter a critical state when the drive reaches 90% capacity.

2. CPU

The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU usage exceeds 75%. The service will enter a critical state when the CPU exceeds 90% utilization.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

5. Service Provided:

The server at 10.100.100.12 is a FTP and File server, whose metrics are detailed below.

5.1 Response time: The maximum guaranteed response time is .5 seconds. A warning will be activated on the monitoring software when the response time exceeds .5 seconds. The service will enter a critical state when the response time exceeds 1 second.

5.2 Connected users: The maximum guaranteed number of connected users is 10. A warning will be activated on the monitoring software when the number of currently connected users exceeds 10. The service will enter a critical state when more than 15 users are connected.

6. Discussion of SLA's in a Virtualized Environment versus a Physical Environment.

While there are differences in the way that a virtual server and a physical server operate, when migrating a SLA from a physical server to a virtual server there are similarities which can make the conversion easier on both the client and the provider. A server whether physical or virtual, is assigned a role to provide a specific service. The quality of the service provided should remain constant regardless of the medium in which the server resides. Due to similarities in server expectations, the metrics written for a physical server SLA can be transferred to the new virtualized SLA. Monitoring can be performed, as before, allowing the customer and the provider confidence that they are obtaining the expected results from the server.

The creation of a SLA for a physical server involves metrics for tangible hardware exclusively. The usage of the CPU and amount of free disk space are both examples of metrics, directly correlated to concrete hardware that would likely be measured in a SLA for a physical server. Quantifying such metrics requires a query to the hardware, which returns data that either meets the guaranteed metric or it does not. This is the primary difference between the physical environment and a virtualized environment. The hardware used by the virtual server is still

tangible, however it is the way that the virtual environment integrates the hardware which differs. Metrics measured in the physical environment can succeed in the virtual environment with modifications to the values implemented.

The differences setting the virtual environment apart from the physical environment are subtle, yet critical when creating the metrics for a SLA. The VM has little direct contact with the host machine's hardware; it requires the use of an intermediary in the form of the virtual machine monitor (VMM). The VMM provides contact with the physical hardware that the guest's operating system uses. A query to the host's operating system may report utilization that is greater, or less, than the system is truly utilizing through the VMM. A majority of functions are performed through the VMM, and thus needs to be considered when creating a SLA for a virtualized machine. The new virtualized server may configured with identical specifications as the physical server that it is replacing, but the host machine on which the virtual server runs needs to be more powerful in order to accommodate the needs of the virtualized environment. Virtualizing the CPU calls and memory overhead are two such examples where additional power will be required. Two types of memory overhead, according to VMware, are "The additional time to access memory within a virtual machine." and "The extra space needed by the ESX Server host for its own code and data structures, beyond the memory allocated to each virtual machine."[15] Memory overhead prompts monitoring the memory of the host machine. The memory usage within the VM may be acceptable according to the SLA; however the memory usage plus any overhead may cause the VM to fall out of compliance with the SLA if left unconsidered. While particular processes of the VM are able to run directly on the CPU without the VMM; a large amount of processes necessitate the VMM causing an increase in CPU usage. If CPU usage were not considered when creating the SLA for the virtualized environment,

similar to neglecting memory overhead, the SLA may breach compliance. In order to track increased CPU usage, it may be necessary to monitor the CPU usage of the VM from within the guest operating system and the VMM; gaining a clearer picture as to the exact CPU usage.

Virtualizing the server changes the fundamental operation of the machines, however, a resolution can be achieved by adapting the physical server SLA. Additional metrics would need to be added to the virtual server SLA for continued effectiveness upon transferring from a physical schema. For example, in a non-virtualized environment, concepts such as memory overhead and virtualizing CPU calls are irrelevant due to the absence of the VMM. In a virtual server the VMM presents as a central element with supplementary metric requirements. Metrics for these components are lacking when migrating a physical server SLA to a virtual machine. Monitoring of specialized processes, in their respective locations, becomes essential when converting metrics designed for use with a physical server SLA into one customized for use in a virtualized environment. The CPU usage of the physical server is a metric from the physical server that needs to change when migrating to the virtual environment; this change has been determined by the testing that was performed. In the virtual machine SLA, the document must state the maximum guaranteed CPU utilization as before, but it must be documented to gather the metric from inside of the guest's operating system as well as from the host, or the VMM, depending on the particular implementation of virtualization. The physical memory from the physical server SLA also needs to be modified in order to be effective in the virtual environment. Memory overhead is the main culprit of the failure of the metric measuring physical memory. Measuring physical memory from within the virtual machine would yield similar results to before the server had been virtualized. In actuality, the server is using more memory in the virtual realm, as compared to the physical state, because of memory that is being occupied by the

VMM. The physical memory requires monitoring Similar to the metrics for the CPU usage, which needs to detail separate documentation for utilization inside the guest operating system, and either the VMM, or the host operating system. Specific changes to the metrics that address memory usage would accommodate the increased memory load, as well as the change in memory usage. The aforementioned changes to the SLA will optimize how a SLA functions in the virtual environment, benefiting consolidation and virtualizing servers by providing more accurate representation of information.

6.1 Revised VM SLA Discussion

After examining at how a VM communicates with the underlying hardware it is clear that some alterations are required for the initial virtual machine SLA in order for relevance to be achieved. Two new metrics were added to the SLA, as mentioned in the discussion above. The additional metrics will be measuring the memory and CPU utilization of the VMM. These novel metrics will be placed in the SLA section designated for the VM host as they can only be monitored from outside of the VM itself. Memory and CPU Utilization metrics will allow the monitoring software to more accurately provide relevant data in regards to server performance, indicating if one of the VM's are monopolizing the CPU, or the memory.

6.2 Revised Virtual Machine Service Level Agreement

The machines at IP addresses 10.100.100.2(VM Host), 10.100.100.10(Windows Media Server), 10.100.100.11(HTTP Server) and 10.100.100.12(FTP and File Server) will be able to perform up to the following metrics, which will be monitored by the Nagios server at address 10.100.100.4. All parties involved in this service level agreement will have access to the Nagios service summary page in order to track individual components of the server. In addition to manually checking the services within Nagios, if any of the services detailed below reach a critical level, both customer and provider will be automatically emailed by the Nagios server, and corrective action taken.

For use in this service level agreement, the term service will be defined as the individual instance of the monitoring software that shows the state of the object that is being monitored. The service can be in only one of three states at any one point in time; OK, Warning, or Critical.

10.100.100.2(VM Host)

1. Drive space

The server will display multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been utilized. The service will enter a critical state when the drive reaches 90% capacity.

2. CPU

2.1 The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU exceeds 75% utilization. The service will enter a critical state when the CPU exceeds 90% utilization

2.2 The maximum guaranteed CPU usage for the FTP and File Server VM process is 25% of the total CPU usage. The CPU usage service will activate a warning when the usage exceeds 25%. The service will enter a critical state when the CPU usage exceeds 33%.

2.3 The maximum guaranteed CPU usage for the Windows Media Server VM process is 25% of the total CPU usage. The CPU usage service will activate a warning when the usage exceeds 25%. The service will enter a critical state when the CPU usage exceeds 33%.

2.4 The maximum guaranteed CPU usage for the HTTP Server VM process is 25% of the total CPU usage. The CPU usage service will activate a warning when the usage exceeds 25%. The service will enter a critical state when the CPU usage exceeds 33%.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

3.4 FTP and File Server VM Process: The guaranteed maximum usage of the physical memory is 201MB. Usage between 201MB and 301MB will activate a warning in the monitoring system. The service will enter a critical state when usage exceeds 301MB.

3.5 Windows Media Server VM Process: The guaranteed maximum usage of the physical memory is 201MB. Usage between 201MB and 301MB will activate a warning in the monitoring system. The service will enter a critical state when usage exceeds 301MB.

3.6 HTTP Media Server VM Process: The guaranteed maximum usage of the physical memory is 201MB. Usage between 201MB and 301MB will activate a warning in the monitoring system. The service will enter a critical state when usage exceeds 301MB.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

10.100.100.10(Windows Media Server)

1. Drive space

The server will display multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been utilized. The service will enter a critical state when the drive reaches 90% capacity.

1.2 Windows Media Server: The folder containing the data for this service will be capped at 5 Gigabytes. A warning will be activated on the monitoring software when the data level reaches 3.5 Gigabytes. The service will be marked as critical when the folder containing the data exceeds 4.5 Gigabytes.

2. CPU

The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU exceeds 75% utilization. The service will enter a critical state when the CPU exceeds 90% utilization.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

5. Service Provided:

The server at 10.100.100.10 is a Windows Media Server, whose metrics are detailed below.

5.1 Service running: The service will be accessible 99% of the time that the server is operating. A warning will be activated on the monitoring software when the service cannot be contacted for .5% of the time the server is running. The service will enter a critical state when the service cannot be contacted, or the time that the service has been down, exceeds 1% of the uptime of the server.

5.2 Connected users: The maximum number of connected users is 20. A warning will be activated on the monitoring software when the number of currently connected users exceeds 20. The service will enter a critical state when more than 30 users are connected.

10.100.100.11(Web Server)

1. Drive space

The server will present multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 HTTP: Data in the folder for this service will be capped at 3 Gigabytes, a warning will be activated on the monitoring software when the data level reaches 2 Gigabytes. The service will be marked as critical when the data level reaches 2.5 Gigabytes.

1.1 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been utilized. The service will enter a critical state when the drive reaches 90% capacity.

2. CPU

The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU exceeds 75% utilization. The service will enter a critical state when the CPU exceeds 90% utilization

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

5. Service Provided:

The server at 10.100.100.11 is a webserver, whose metrics are detailed below.

5.1 Response time: The maximum guaranteed response time is .5 seconds. A warning will be activated on the monitoring software when the response time exceeds .5 seconds. The service will enter a critical state when the response time exceeds 1 second.

5.2 Connected users: The maximum number of connected users is 20. A warning will be activated on the monitoring software when the number of currently connected users exceeds 20. The service will enter a critical state when more than 30 users are connected.

10.100.100.12 FTP and File server

1. Drive space

The server will present multiple folders containing the data for the associated services. Each folder will be monitored separately, and each will have their own warning and critical states.

1.1 FTP and File storage: The folder containing the data for these two services will be capped at 10 Gigabytes. A warning will be activated on the monitoring software when the data level reaches 8 Gigabytes. The service will be marked as critical when the data level reaches 9.5 Gigabytes.

1.2 Total Drive Space: The maximum guaranteed total drive usage is 75%. A warning will be activated when 75% of the drive has been utilized. The service will enter a critical state when the drive reaches 90% capacity.

2. CPU

The maximum guaranteed CPU usage is 75%. The CPU usage service will activate a warning when the CPU exceeds 75% utilization. The service will enter a critical state when the CPU exceeds 90% utilization.

3. Memory Usage

3.1 Page File: The guaranteed maximum usage of the page file is 60%. A warning will be activated on the monitoring software when the page file reaches 60% usage. The service will enter a critical state when the page file reaches 90% usage.

3.2 Virtual Memory: The guaranteed maximum usage of the virtual memory is 60%. A warning will be activated on the monitoring software when the virtual memory reaches 60% usage. The service will enter a critical state when the virtual memory reaches 90% usage.

3.3 Physical Memory: The guaranteed maximum usage of the physical memory is 60%. A warning will be activated on the monitoring software when the physical memory reaches 60% usage. The service will enter a critical state when the physical memory reaches 90% usage.

4. Network Utilization

4.1 Number of packets received per second: The maximum guaranteed number of packets received per second is 6000. A warning will be activated on the monitoring software when the number of packets per second exceeds 6000. The service will enter a critical state when the number of packets per second exceeds 9000.

4.2 Number of packets transmitted per second: The maximum guaranteed number of packets transmitted per second is 9,000. A warning will be activated on the monitoring software when the number of packets per second exceeds 9,000. The service will enter a critical state when the number of packets per second exceeds 13,500.

4.3 Network Utilization input: The maximum guaranteed network utilization input is 50%. A warning will be activated on the monitoring software when the network utilization input level exceeds 50%. The service will enter a critical state when the network utilization input level exceeds 90%.

4.4 Network Utilization output: The maximum guaranteed network utilization output is 50%. A warning will be activated on the monitoring software when the network utilization output level exceeds 50%. The service will enter a critical state when the network output level exceeds 90%.

5. Service Provided:

The server at 10.100.100.12 is a FTP and File server, whose metrics are detailed below.

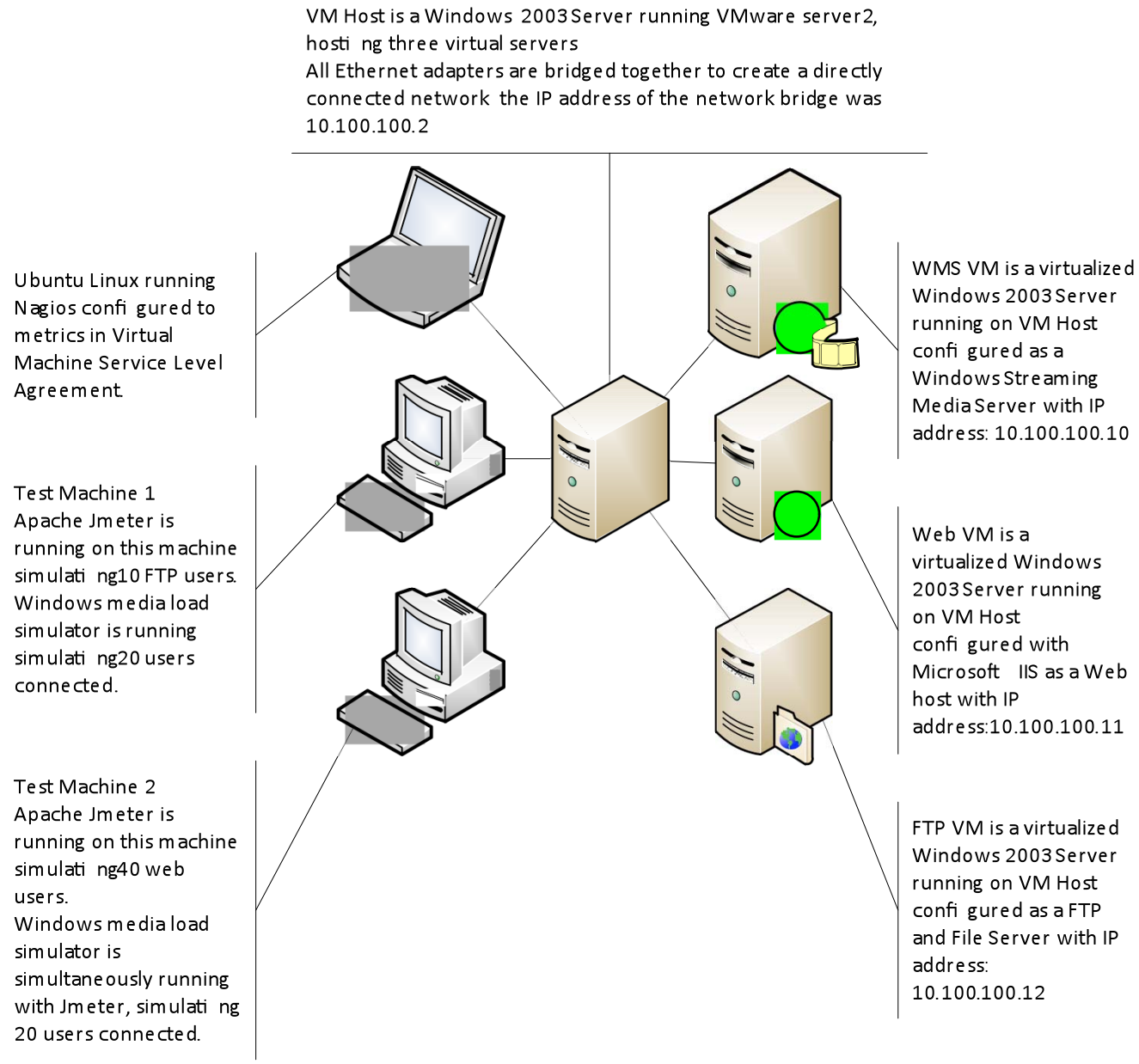
5.1 Response time: The maximum guaranteed response time is .5 seconds. A warning will be activated on the monitoring software when the response time exceeds .5 seconds. The service will enter a critical state when the response time exceeds 1 second.

5.2 Connected users: The maximum guaranteed number of connected users is 5. A warning will be activated on the monitoring software when the number of currently connected users exceeds 5. The service will enter a critical state when more than 7.5 users are connected.

7. Assessment Of The Physical Server SLA In A Virtual Environment (Tests 1 and 2)

Evaluating the compatibility of the physical server SLA with the virtual machines required testing to ascertain how well the VM's respond to the SLA that was developed for them. The first test gathered data on the machines while they were at idle. The purpose was to find out the lowest possible results if all of the VM's were on, but no one was actively trying to utilize them. The rationale for this procedure was that, if the VM's did not perform well with the absence of a load, then under a heavy load they would not be able to perform, as guaranteed in the SLA. The next test would trial the SLA employing the guaranteed number of connected users as outlined in the VM SLA, created from the physical server SLA. This analysis would reveal how the virtual servers perform under the same load as the physical server was capable of handling. If while this test was running under the virtual environment, the same results were gathered as from the examination of the physical server, then no changes would need to be made to the VM SLA. A detailed diagram depicting the aforementioned environment is presented below.

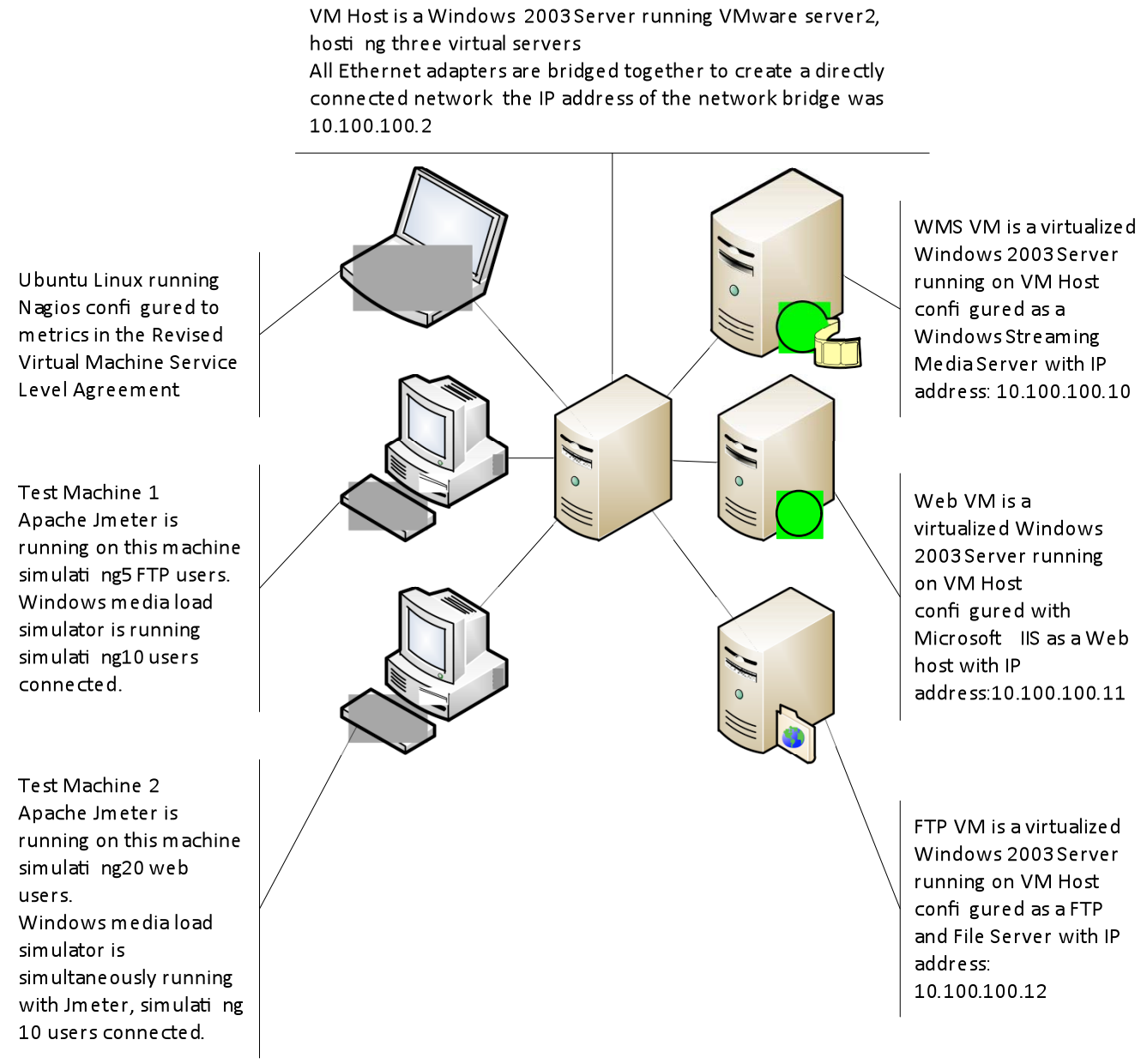
Diagram of second test



7.1 Assessment Of The Physical Server SLA In A Virtual Environment (Test 3)

The third test that was conducted applied half the number of guaranteed connected users as were in the original VM SLA. In addition to halving the users, the metrics that were found relevant in the discussion were added to the revised virtual server SLA and were monitored during this test. This investigation would establish if the server improved its performance, and response time, by more than 50% if half of the guaranteed number of users are allowed to connect; thus, showing that a change in the SLA would be needed in order to make the SLA applicable to the VM. A detailed diagram depicting the aforementioned environment is presented below, showing the changes that were made to the number of connected users.

Diagram of third test



8. Results

Data was collected from inside the Host server, from VMware, and from Nagios which was monitoring all of the servers. This section will detail three of the main metrics (CPU usage, network traffic, memory usage) that were gathered from the testing. The complete chart depicting the data that was collected in each of the three evaluations, along with the data that was recorded from the physical server analysis, can be found in Appendix A.

8.1 CPU

Data was collected while the host server was not experiencing any load except running all three virtual machines at idle. The CPU usage of the host machine, gathered via Nagios, was at 45%, the FTP VM was at 7%, the Web VM was 8%, and the WMS VM was 9%. The CPU load of the VM machines according to VMware was: 1% for the FTP server, 1% for the Web server, and 1% for the WM server. The CPU usage according to windows task manager was 2% for the FTP VM process, 3% for the Web VM process, 2% for the WMS VM process, and a total CPU usage for the VM host was 10%.

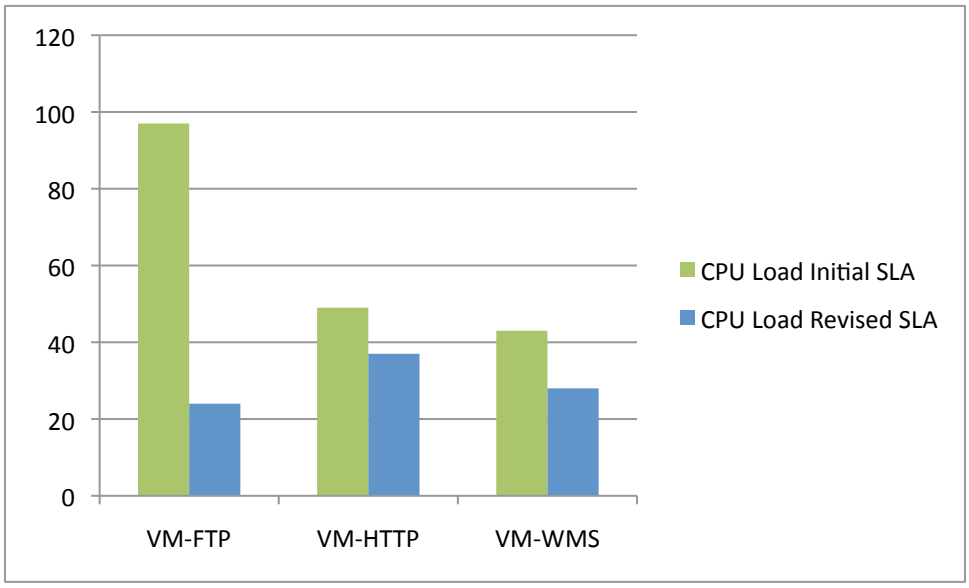
Following the first test, with the guaranteed number of users outlined in the physical server SLA, the following information was amassed. The CPU usage of the Host machine, according to Nagios, was in a critical state with usage of greater than 99%. The CPU usage of the FTP and File server VM was also in a critical state with usage of greater than 97%. In contrast to the FTP server, both the Windows Media server (WMS) and the Web server were at 43%, and 49%, utilization respectively. Nagios conveyed CPU usage of the processes from the task manager of the VM's within Windows. The results are: 58% total CPU usage for the FTP VM process, 25% total CPU usage for the Web VM process, 5% total CPU usage for the WMS

VM process, and 100% total CPU usage for the VM host machine. Additionally the CPU usage within VMware itself, according to VMware, reported the FTP server was using 42% of the total CPU, the Web server was using 29% of the total CPU, and the WMS was using 13.5% of the total CPU.

The second test occurred by halving the number of users simulated in the traffic generators, but keeping the rest of the metrics identical to the previous investigation. The CPU usage of the VM host machine, according to Nagios, was 61% utilization. This was a decrease of 39% utilization, and brought the state of the service on Nagios from critical to OK. The CPU usage of the FTP was 24%, a decrease of 73% utilization, bringing the service from a critical state to OK state in Nagios. The CPU usage of WMS was 28%, demonstrating a 19% utilization decrease. The Web server CPU usage was 37%, presenting a decrease of 12% utilization. Both WMS and web CPU services in Nagios remained in the OK state. The CPU usage via Windows task manager indicated CPU usages of the VM processes as follows: 32% for FTP VM Process, 8% for the Web server VM Process, and 10% for the WMS VM process. The results equate to a 26% utilization decrease for the FTP VM Process, a 17% utilization decrease for the web VM process, and a 5% utilization increase for the WMS VM process. The total CPU usage of the host machine from within the task manager was 65%, a 35% utilization improvement. The CPU usage from within VMware for the FTP server was 24% (down 18% utilization), for the Web server it was 10% (down 19% utilization) and for the WMS it was 5% (down 8.5% utilization).

The three charts below detail the change in CPU usage from the initial virtual machine SLA to the revised SLA, when measuring CPU usage from inside the VM, within VMware, and from the host machine's task manager, which shows the CPU usage of the VM server process. The first chart shows a significant drop in CPU usage of the VM-FTP server when the number of users was reduced by half, however the VM-HTTP and VM-WMS do not show this drastic variation of CPU usage when the users were halved. It could be hypothesized that the VM-FTP was utilizing almost the entire host machine's CPU resources, but without further data to support the findings this can not specifically be determined.

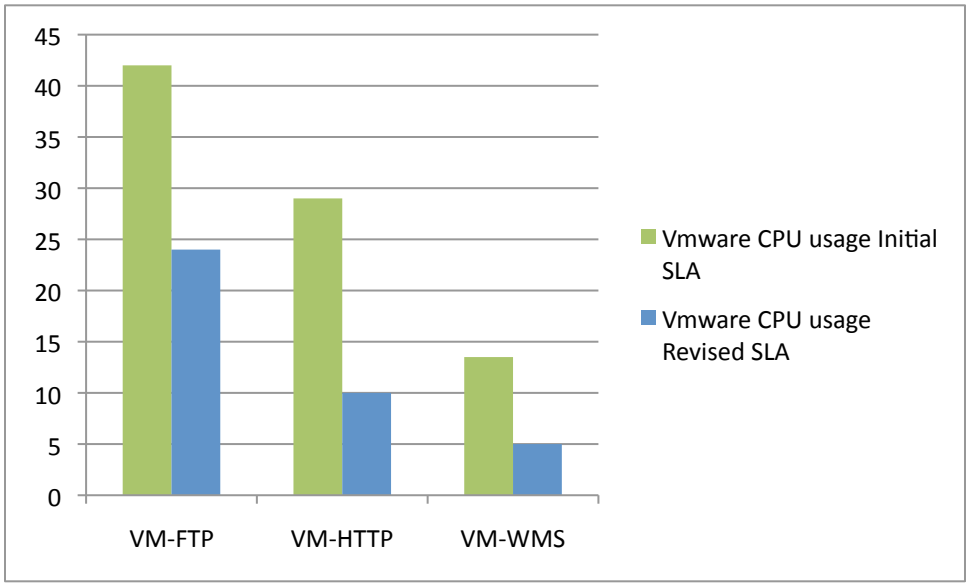
CPU Chart 1



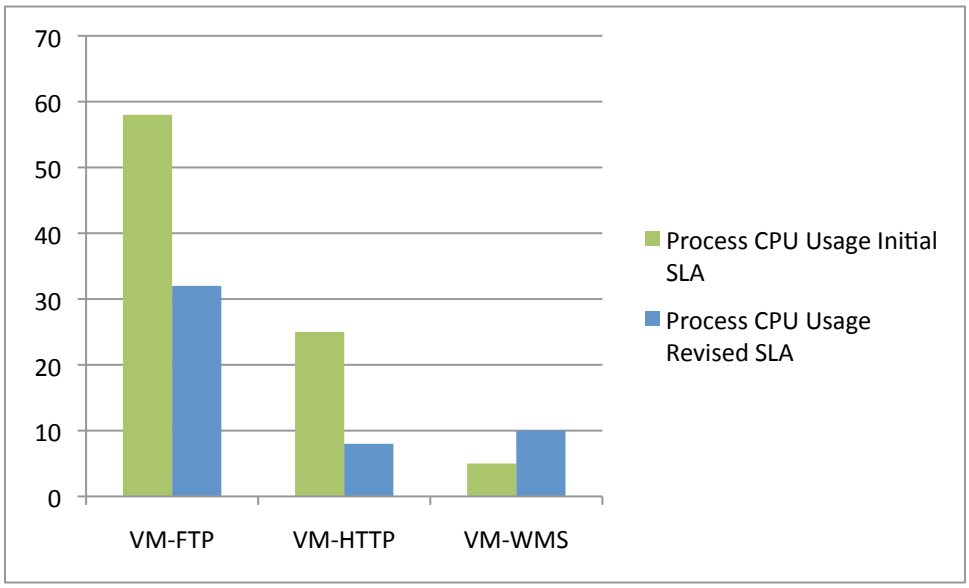
These subsequent CPU charts show the CPU utilization from outside of the VM illustrating the complete picture of CPU function. During the first test with the initial virtual server SLA, the VM-FTP was averaging 50% of the total CPU utilization, leaving the other half of the CPU to be split between the two other VMs and the host machine. Thus, the VM-FTP was dominating the CPU and the other entities were not performing to their full potential. This can be seen in CPU Chart 3 where the process CPU usage of the VM-WMS increases after a reduction in the number

of users. Since the VM-FTP server was no longer controlling the CPU, the other two servers were able to prosper.

CPU Chart 2



CPU Chart 3



8.2 Network Traffic

Measurements of the network traffic at idle were broken down into four categories; network utilization inbound, network utilization outbound, packets received per second, and packets transmitted per second, to improve the reporting accuracy from Nagios. The network utilization inbound and outbound was 0% for the FTP VM, WMS VM, the Web VM and the VM host. The packets received per second, averaged over a 5 minute period, was .42 for the VM host, .27 for the FTP VM, .15 for the Web VM, and .16 for the WMS VM. The packets transmitted per second, averaged over a 5 minute period was .375 for the VM host, .24 for the FTP VM, .13 for the Web VM, and .13 for the WMS VM. The data collected during idle assessment represented an OK state in the Nagios software.

Once the data for idle was captured, the first test gathered the data utilizing the metrics from the VM SLA. The network utilization inbound for the VM host was 0%, for the FTP VM it was .05%, it was .11% for the web VM, and the WMS VM was .006%. The network utilization outbound for the VM host was 0%, for the FTP VM it was .96%, the web VM was reporting .17%, and the WMS VM had an outbound network utilization of 1.24%. The number of packets received per second, averaged over 5 minutes, was 6.48 for the VM host, 1039.66 for the FTP VM, 1335.54 for the web VM, and 48.98 for the WMS VM. The number of packets transmitted per second, averaged over 5 minutes, was 4.18 for the VM Host, 1387.4 for the FTP VM, 925.11 for the web VM, 1783.96 for the WMS VM. As in the idle data collection, all of the Nagios services were in the OK state during this evaluation.

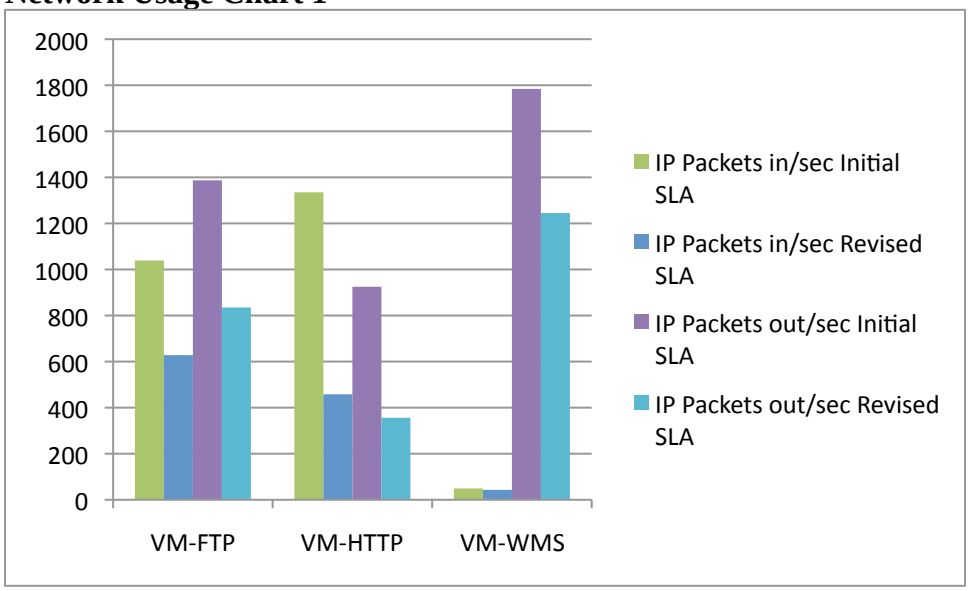
Using half the number of users stated in the initial SLA, the following data for the network traffic could be procured. The network utilization inbound to the VM host was 0%, for

the FTP VM it was .02% (a decrease of .03%), it was .04% for the web VM (a decrease of .07%), and for the WMS VM the inbound network utilization was .007% (an increase of .001%). The network utilization outbound for the VM host was 0%, for the FTP VM it was .51% (a decrease of .45%), for the web VM it was .06% (a decrease .11%) and for the WMS VM it was 1.06% (a decrease of .18%). The number of packets received per second for the VM host was .62/sec (a decrease of 5.86/sec), and 628.2/sec for the FTP VM (a decrease of 628.86/sec). The number of packets received was 458.18/sec for the web VM, a decrease of 877.36/sec. The number of packets received per second for the WMS VM was 42.83, a decrease of 6.15. The number of packets transmitted per second was .57 for the VM host, a decrease of 3.61. The FTP VM saw a decrease of 551.8/sec to a transmitted rate of 835.6/sec. The web VM transmitted an average of 355.74/sec for the web VM a decrease of 569.37/sec. The WMS VM transmitted 1245.22 packets per second, a decrease of 538.74. As with the previous two data collections, all of the services within Nagios monitoring the network traffic presented an ok state throughout this test.

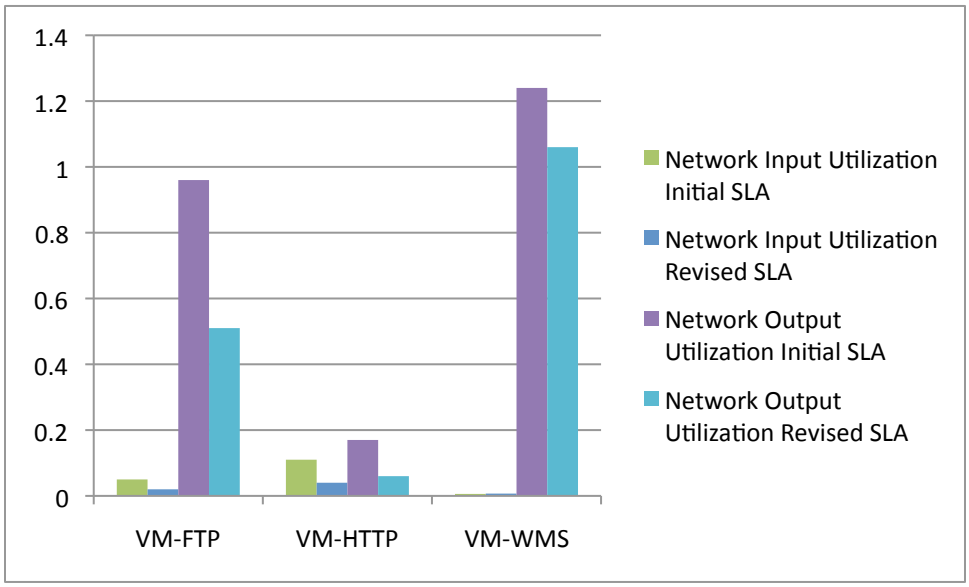
As seen in Network Usage Charts below, during the investigation of the initial virtual machine SLA, the number of packets per second, both received and transmitted for the VM-FTP and VM-WMS did not decrease by 50% when the number of users were halved. This decrease of indirect proportions can also be seen in Network Usage Charts 2 and 3. When testing the initial virtual machine SLA with the reduced number of users, the network usage of the VM-WMS scarcely decreased. This could be linked to the CPU usage of the VM-FTP server. Once the VM-FTP server's usage was lessened by decreasing the number of users as stated in the revised VM SLA, the VM-WMS was able to improve its output to peak performance. Without ascertaining why the server was unable to perform to its full abilities, it becomes difficult to

pinpoint a problem. One solution, in this case, is the ability to monitor the CPU usage from multiple points, as detailed in the revised virtual machine SLA. This data collection helps to better understand, and improve, the performance of all the servers being hosted by rapidly identifying the source of an error to be rectified by the provider.

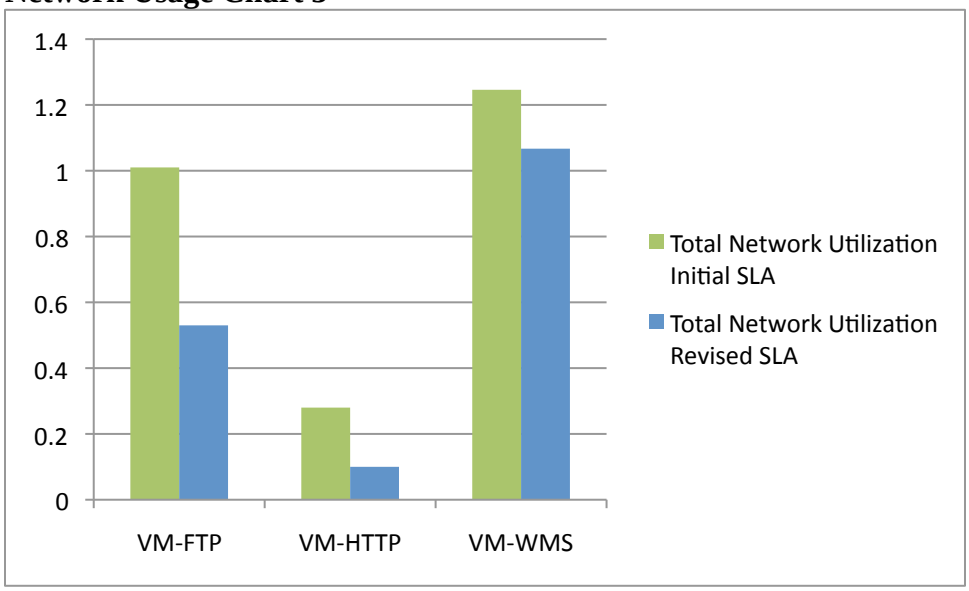
Network Usage Chart 1



Network Usage Chart 2



Network Usage Chart 3



8.3 Memory

As in the previous two sections, this section will be divided into three subsections; physical memory at idle, at the full number of users the SLA guarantees, and at half of the number of users that the initial SLA states. The physical memory at idle for the VM host was 1.36G, this is a 90% utilization of the memory and the Nagios service alerted a critical state. The FTP VM was using 119MB of physical memory, which is 44% of the total memory that is assigned to the VM. The web VM was using 95.7MB, which is 41% of the total memory allotted. The WMS VM was using 112MB, which is 56% of the assigned physical memory. According to Nagios the physical memory services of the three VMs were all in an OK state.

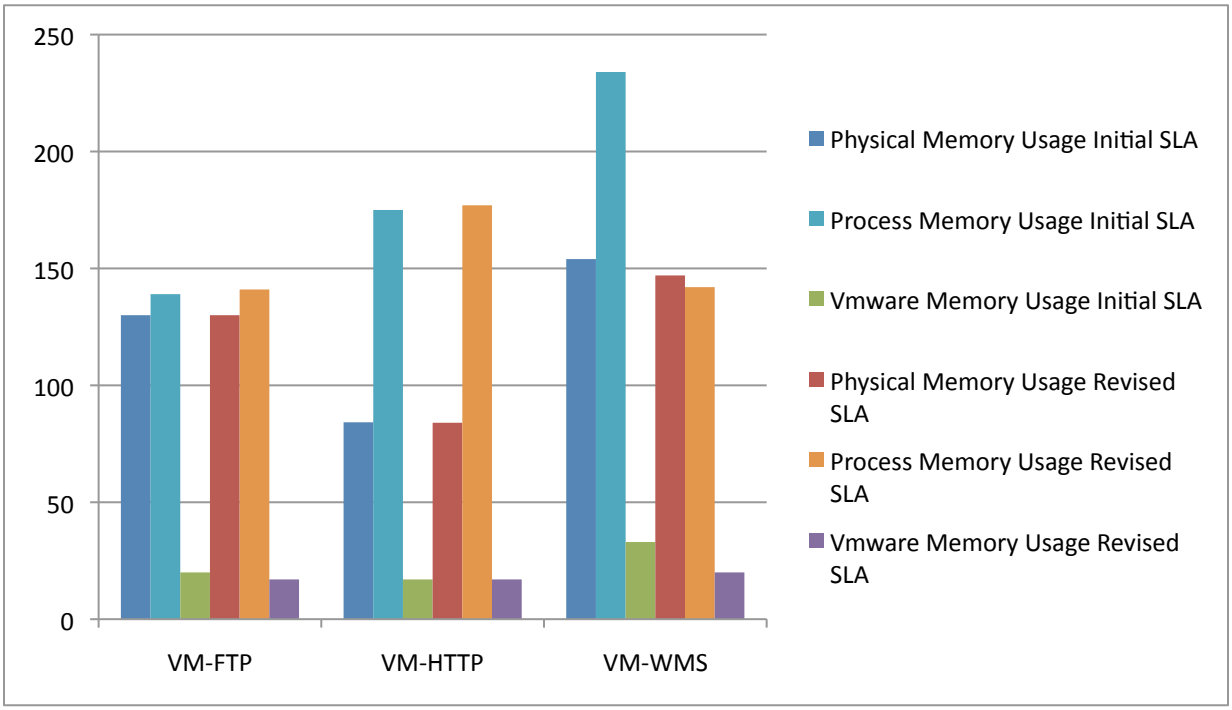
The data in this paragraph was obtained by administering the trials with the amount of users listed in both the physical server, and VM SLAs. The physical memory during this test for the VM host was 1.42G, which is 94% of the total memory used; this placed the service in a critical state in the Nagios server. The physical memory utilized on the FTP VM server was 51%, or 130MB, and the web VM was using 33%, or 84.2MB of its memory. Both of these

services were in the OK state in Nagios. The WMS VM was using 60%, or 154MB, of its allotted memory; thus the memory service triggered a warning state in Nagios.

Implementing the assessment with half the number of users from the initial SLA provided the results that follow. The physical memory being used by the VM host was 1.37G, which is 91% utilization, and retained a critical state in the Nagios service as in the first trial. The physical memory of the FTP VM server remained at 130MB, or 51% utilization, from the previous test and the Nagios monitoring service continued to document an OK state. The physical memory of the web VM went down slightly from 84.2MB utilization to 84MB utilization, which was a decrease of less than a percentage so the service remained at 33%, and registered an OK state in Nagios. The WMS VM reduced its memory usage by 3% from 154MB to 147MB, which brought the state of the Nagios service from a warning to an OK state.

When evaluating the physical memory usage chart below, there does not appear to be much difference between the two tests. However, when the amount of memory from within the VM (Physical Memory Usage) is compared with the amount of memory from outside the VM (Process Memory Usage), it is evident that the host identifies a greater memory usage than the VM. This data represents the memory overhead discussed above which also needs to be taken into account when creating a SLA for a VM. The chart below depicts the memory usage according to the VM, as a portion of the total memory usage. The difference between data communicated from the VM, and what can be measured from the host, demonstrates that the physical memory usage metric needs to be measured from multiple locations, similar to how the CPU usage is measured from multiple locales. Quantifying metrics from internal and external locations helps to assembles the information regarding server performance in its entirety.

Memory Usage Chart



9. Conclusion

With the use of virtualized servers on the rise, it has become imperative to examine the SLAs that are currently in practice. While it is assumed that a simple migration of metrics from a physical server SLA into a novel SLA, which covers the virtual machine, is the only necessary step in order to support a company's VM; this has been found to be erroneous. Virtualizing an enterprise server is complex and demands energy and resources for the virtual server to function properly. Companies need to extend the same amount of effort into the development of a SLA to ensure the executions new virtual server's functions, as they do into the tangible process of virtualizing. While the server may provide the same role as before, the manner in which the

server now functions has changed by virtue of it being virtualized. The difference is significant enough to warrant a redraft of the SLA that is governing the operation of the server.

Planning out a SLA for a physical server is simple task; determine the need the server is fulfilling, build the server with ample capacity to handle the demands, and configure the metrics of the SLA accordingly. The metrics for a well drafted SLA should be easy to understand and monitor by all parties involved. The same should be the case for developing an SLA regarding a virtual server; establish the need to be addressed, build accordingly, and configure the metrics of the SLA. The virtual server has caused frustration though, because when migrating to the virtual server the SLA is not being designed properly. SLA's copied over directly from their physical server counterparts have been the downfall in most cases.

Throughout the course of this study, it has been ascertained that SLA's do indeed need to change in order to work effectively in tandem with the server in a virtualized environment. While the standard metrics, from a physical server SLA, may be useful, they may not truly convey what is occurring in the server. It was discovered that if a physical server hosting multiple services was then migrated to multiple virtualized servers on a physical machine, changes to the SLA were required once the switch to virtualized server had been made.

An observation noted during the testing of the original virtual server SLA revealed that when the total number of users were active, the FTP server claimed 100% of its CPU usage. This caused the host machine to do the same; however, the Windows Media Server(WMS) and web server were operating at the rate indicated in the first virtual server SLA. Ultimately, when the number of users was halved from the initial virtual server SLA, it was noted that the performance of all three servers did not decrease by half. The WMS and web server both

experienced a decrease in load of only 30%. With the FTP server no longer monopolizing the CPU, all three servers were able to function in an OK state, as per Nagios. The decline in users by 50% caused a 40% lesser total output demonstrating that the server was able to perform better with half the number of users. The necessary step of halving the number of users from the initial virtual SLA, which was derived from the physical server SLA, leads the reader to determine that the primary cause of multiple system failure was the physical server that hosted the virtual machines. While the physical server was qualified to run multiple virtual machines, it was not capable of handling the usage created by the users in the first virtual SLA.

Noting that the virtual servers were not capable of functioning to the standards as set in the initial virtual SLA produced novel ideas for metrics to be used in the revised virtual SLA that governed the virtual servers. New metrics, such as VMM utilization of CPU and memory, came about because of the testing that was performed on the original virtual server SLA. A discrepancy was perceived, during the first virtual SLA implementation, between data reported internally from the virtual machine and the data observed from the host machine. Monitoring the CPU usage of the VMM is an additional metric that should be used in the revised virtual server SLA. By monitoring the VMM CPU usage, as well as CPU usage of the virtual machine, the provider is able to more accurately determine the CPU usage of the virtual machine. This knowledge drives decisions regarding potentially moving the VM to a more appropriate host in order to compensate for the increase in CPU load. VMM memory utilization is also a metric that could be effectively added to the standard metrics used when creating a revised SLA for a virtual machine. This metric could help the provider more effectively manage the VM host machines. As with the VMM CPU load, if the VMM memory utilization became too elevated, the

monitoring system would alert the provider that the VM host may not be suitable for the current VM.

The metrics mentioned above are two of the observed metrics from the testing that would be beneficial if implemented in a SLA designed for a virtualized environment. This study indeed proves the point that while some of the metrics used for a physical server SLA can be transferred to the virtual environment; a SLA that was designed for a physical server can not be directly implemented on a virtual server. The aforementioned new metrics ought to commence further studies pertaining to potential metrics that could be implemented in a virtual environment, in order to properly protect all parties involved in the SLA.

10. Appendix – Server Statistics

Physical server statistics

	Idle	Full test
CPU Load	0%	62%
HD Free space	83%	83%
Memory Usage: Physical	23%	23%
Memory Usage: Virtual	44.1MB	44.1
Memory Usage: Paged	182MB	175MB
Network Input Utilization	0%	.1%
Network Output Utilization	0%	2.4%
IP Packets in/sec	.2	5160.9
IP Packets out/sec	.2	8474.67
FTP Response time	.002 Seconds	.006
FTP Users Connected	0	1.2
FTP Directory Size	43.6MB	43.6
HTTP Response time	.032 Seconds	.030
HTTP Users Connected	0	2.6
HTTP Directory Size	4.89MB	4.89MB
Windows Media Service	Service Running	Service Running
WMS Users Connected	0	32.6
WMS Directory Size	8.78MB	8.78MB

Virtual Server Statistics

	Idle	Full – SLA	Half - SLA	Percent Change
CPU Load	17%	>99%	61%	(38%)
HD Free Space	28%	28%	28%	0%
Memory Usage: Physical	1.36G	1.42G	1.37G	(3%)
Memory Usage: Virtual	46MB	51MB	55MB	7%
Memory Usage: Paged	1.12GB	1.23GB	1.3GB	5%
IP Packets in/sec	.42	6.48	.62	(90%)
IP Packets out/sec	.375	4.18	.57	(86%)
Network Input Utilization	0%	0%	0%	0%
Network Output Utilization	0%	0%	0%	0%
Total Task Manager CPU Usage %	11%	100%	65%	(35%)

FTP VM Server

	Idle	Full SLA	Half SLA	Percent Change
CPU Load	7%	>97%	24%	(75%)
HD Free Space	72%	72%	72%	0%
Memory Usage: Physical	119MB	130MB	130MB	0%
Memory Usage: Virtual	39.4MB	39.4MB	39.4MB	0%
Memory Usage: Paged	94.3MB	104MB	114MB	9%
IP Packets in/sec	.27	1039.66	628.2	(39%)
IP Packets out/sec	.24	1387.4	835.6	(39%)
Network Input Utilization	0%	.05	.02	(60%)
Network Output Utilization	0%	.96	.51	(46%)
FTP Response time	.004	.026	.035	34%
FTP Users Connected	0	8.8	.4	
FTP Directory Size	0B	0B	0B	
VMWare CPU Usage	.041GHz	1.159GHz	.661GHz	(42%)
VMWare CPU Utilization %	1%	42%	24%	(42%)
VMWare Memory Usage	15MB	20MB	17MB	(15%)
VMWare Memory Utilization %	5%	20%	6%	(70%)
Task Manager CPU Usage %	2%	58%	32%	(44%)
Task Manager Memory Usage	349MB	139MB	141MB	1%

VM HTTP Server

	Idle	Full – SLA	Half -SLA	Percent Change
CPU Load	8%	49%	37%	(24%)
HD Free Space	70%	70%	70%	0%
Memory Usage: Physical	95.7MB	84.2	84MB	(.2%)
Memory Usage: Virtual	39.4MB	41.5MB	41.5MB	0%
Memory Usage: Paged	104MB	133MB	131MB	(1%)
IP Packets in/sec	.15	1335.54	458.18	(65%)
IP Packets out/sec	.13	925.11	355.74	(61%)
Network Input Utilization	0%	.11%	.04%	(63%)
Network Output Utilization	0%	.17%	.06%	(64%)
HTTP Response time	.011	.12	.01	(91%)
HTTP Users Connected	0	7.4	6.6	(10%)
HTTP Directory Size	21.8K	21.8K	21.8K	0%
VMWare CPU Usage	.05GHz	.793GHz	.29GHz	(63%)
VMWare CPU Utilization %	1%	29%	10%	(65%)
VMWare Memory Usage	15MB	17MB	17MB	0%
VMWare Memory Utilization %	5%	6%	6%	0%
Task Manager CPU Usage %	2%	25%	8%	(68%)
Task Manager Memory Usage	360MB	175MB	177MB	1%

VM WMS Server

	Idle	Full - SLA	Half - SLA	Percent Change
CPU Load	9%	43%	28%	(34%)
HD Free Space	79%	79%	79%	0%
Memory Usage: Physical	112MB	154MB	147MB	(4%)
Memory Usage: Virtual	39.3MB	39.5MB	39.4MB	0%
Memory Usage: Paged	93.1MB	135MB	117MB	(13%)
IP Packets in/sec	.16	48.98	42.83	(12%)
IP Packets out/sec	.13	1783.96	1245.22	(30%)
Network Input Utilization	0%	.006%	.007%	16%
Network Output Utilization	0%	1.24%	1.06%	(14%)
WMS Service Running	Yes	Yes	Yes	No Change
WMS Users Connected	0	40	16.2	(59%)
WMS Directory Size	8.84MB	8.84MB	8.84MB	0%
VMWare CPU Usage	.052GHz	.369GHz	.155GHz	(57%)
VMWare CPU Utilization %	1%	13.5%	5%	(62%)
VMWare Memory Usage	7MB	33MB	20MB	(39%)
VMWare Memory Utilization %	2%	13%	7%	(46%)
Task Manager CPU Usage %	2%	5%	10%	50%
Task Manager Memory Usage	347MB	234MB	142MB	(39%)

11. References

[1] Begnum, K., Disney, M., Frisch, Æ., & Mevåg, I. (2007). Decision support for virtual machine re-provisioning in production environments. *Proceedings of* the *21st conference on Large Installation System Administration Conference,* article no. 8. Retrieved from http://portal.acm.org/citation.cfm?id=1349434

[2] Ambrosio, J. (2007, September 24). Virtual machines deployed on the sly. *ComputerWorld,* 14-15. Retrieved from http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=303210

[3] Jang, J., Han, S., Kim, J., Park, S., Bae, S., & Woo, Y. C. (2007). A Performance Evaluation Methodology in Virtual Environments. In *Seventh International Conference on Computer and Information Technology* (pp. 351-356). Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4385107&isnumber=4385041

[4] Creeger, M. (2008, December). CTO Virtualization Roundtable: Part II. *Communications of the ACM, 51*(12, Pt. 2), 43-49. Retrieved from http://portal.acm.org.ezproxy.rit.edu/citation.cfm?id=1409375

[5] Zhang, Y., Su, G., Hong, L., & Zheng, W. (2008). On Virtual-Machine-based Windows File Reads: a Performance Study. In *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application* (pp. 944-948). Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4756915

[6] Park, J.-G., Kim, J.-M., Ahn, C.-W., Woo, Y.-C., & Choi, H. (2008, February). Cluster Management in a Virtualized Server Environment. In *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on* (Vol. 3, pp. 2211-2214). Gangwon-Do. Retrieved from http://ieeexplore.ieee.org.ezproxy.rit.edu/xpl/freeabs_all.jsp?arnumber=4494229

[7] Bouman, J., Trienekens, J., & Van der Zwan, M. (1999, September). *Specification of service level agreements, clarifying concepts on the basis of practical research.* Retrieved from http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/6497/17345/00798790.pdf?arnumber=798790

[8] VMware. (2007). *Understanding full virtualization, paravirtualization, and hardware assist* [Brochure]. Retrieved from http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

[9] *Service Level Agreement Zone*. Retrieved February 20, 2009, from http://www.sla-zone.co.uk/index.htm

[10] Strum, R. (2001, October 31). SLA metrics . *Network World*. Retrieved from http://www.networkworld.com/newsletters/nsm/2001/01083536.html

[11] Skene, J., Lamanna, D. D., & Emmerich, W. (2004). Precise Service Level Agreements. *In Proceedings of the 26th International Conference on Software Engineering* (pp. 179-188). Retrieved May 6, 2009, from http://portal.acm.org/citation.cfm?id=998675.999422&coll=portal&dl=ACM&CFID=38520451&CFTOKEN=89481319

[12] The SLAng SLA Language. (n.d.). Retrieved May 9, 2009, from http://uclslang.sourceforge.net/

[13] Skene, J., Skene, A., Crampton, J., & Emmerich, W. (2007). The monitorability of service-level agreements for application-service provision. *In Proceedings of the 6th international workshop on Software and performance* (pp. 3-14). New York, NY: ACM. Retrieved May 9, 2009, from http://portal.acm.org/citation.cfm?id=1216993.1216997&coll=portal&dl=ACM&CFID=38520451&CFTOKEN=89481319

[14] Raimondi, F., Skene, J., & Emmerich, W. (2008). Efficient online monitoring of web-service SLAs. *In Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering* (pp. 170-180). New York, NY: ACM. Retrieved May 9, 2009, from http://portal.acm.org/citation.cfm?id=1453101.1453125&coll=portal&dl=ACM&CFID=38520451&CFTOKEN=89481319

[15] VMware. (2009). Advanced Resource Management. In Resource Management Guide. Retrieved January 2, 2011, from http://pubs.vmware.com/vi301/resmgmt/wwhelp/wwhimpl/js/html/wwhelp.htm?href=title_vc_cluster.1.1.html#1013453