2009

Priority Based Power Management and Reduced Downtime in Data Centers

Barath Kuppuswamy

San Jose State University

Priority Based Power Management and Reduced Downtime in Data Centers

A Project
Presented to
The Faculty of the Department of Computer Science
San José State University
In Partial Fulfillment
of the Requirements for the Degree
Master of Science
by
Barath Kuppuswamy
May 2009

SAN JOSÉ STATE UNIVERSITY
The Undersigned Project Committee Approves the Project Titled
Priority Based Power Management and Reduced Downtime in Data Centers
by
Barath Kuppuswamy
APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Chris Pollett, Department of Computer Science Date

Dr. Sami Khuri, Department of Computer Science Date

Dr. Teng Moh, Department of Computer Science Date
APPROVED FOR THE UNIVERSITY

Associate Dean Office of Graduate Studies and Research Date

ABSTRACT

Priority Based Power Management and Reduced Downtime in Data Centers

by Barath Kuppuswamy

The project deals successfully with software that performs priority based power management and reduced downtime for virtual machines running in data centers. The software deals with power management only at the processor level. The software automatically performs load distribution among servers in data centers to save power. In addition, the software also lets administrator of data centers to mark certain virtual machines, which run user applications, as critical to minimize downtimes for these virtual machines.

The software reveals that energy consumption can be minimized while maintaining high runtime availability for the mission critical applications. The software operates in Green mode and in regular mode while maintaining high runtime availability. The experimental results show that Green mode minimizes energy usage by as much as 35%.

ACKNOWLEDGMENTS

I like to thank my project advisor Dr. Teng Moh of Computer Science department at San Jose State University for providing me with guidance and feedbacks on this project.

I also like to thank Symantec Corporation and its employees for providing me with machines that I used to analyze my software.

Finally, I like to thank my parents, Geetha Kuppuswamy and Chitta Kuppuswamy, and my sister, Sharanya Kuppuswamy, for their support while I worked on this project.

Table of Contents

List of Figures

Problem

A data center is a place where there are servers. According to Ricardo Bianchini and Ram Rajamony of Computer Science department at Rutgers University and IBM Austin Research lab respectively, “data centers typically host clusters of hundreds, sometimes thousands, of servers [1].” These servers run users’ applications. Sometimes, some of these applications are mission critical, and only few of the servers in data centers are running these critical applications while others are just used as backups. In case of a server failure, products such as Veritas Cluster Server and Sun Cluster will do the transfer of mission critical applications from failed server to one of the backup servers. The current problem is that even the backup servers are running at full speed even though only few of them do useful work. The majority of backup servers are just idle and waiting to be used in case of a failure. These idle backup servers are consuming power too. Hence, lot of energy is wasted in data centers these days.

Due to rapid increase in energy cost, data centers are left with severe economic stress while hosting critical applications in their servers. According to San Jose Mercury newspaper, Subodh Bapat, the vice president of Sun Microsystem’s energy efficiency department, stated that the “cost of powering data centers worldwide could grow from $18.5 billion in 2005 to $250 billion by 2012 [5].” Furthermore, David Filani, an engineer at Intel’s Digital Enterprise Group, stated that “the cost of power and cooling [of servers in data centers] has increased 400% [2].” In addition, data centers are leaving large carbon footprints on our environment and contribute to global warming as a result of data centers wasting energy by letting majority of their backup servers running in full speed even though only few of them do useful work. According to San Jose Mercury newspaper, Bapat believed that people are leaving carbon foot prints whenever they use online applications hosted by servers in the data centers [5].

Since Bapat indicated that data centers are leaving large carbon foot prints on our planet and will face severe economic stress in the future, it is important that data centers tackle this issue of power wastage. This issue must be tackled while ensuring that hosted critical applications still have high runtime availability with the help of backup servers coming to the rescue in case of server failures.

Project Description

Products like Veritas Cluster Server and Sun Cluster that ensure high runtime availability of applications do not have the capacity to manage or optimize power consumption of individual servers in the data centers today. If a power management software tool is available which can minimize power consumption in the main and backup servers in the data centers without compromising the guarantee that the mission critical applications have minimum downtime, then data centers can not only ensure that the mission critical applications have reduced downtime, but also conserve energy. This will result in reduced economic strain on the data centers and help save our planet in future.

In order to make data centers more energy efficient, we must identify the components of servers where energy usage can be minimized. Power consumption can be minimized in a server at various places like CPU, memory, disk etc. But, according to Pat Bohrer of IBM Research, CPU is the most power consuming element of a server [6]. Hence, I decided to develop, as part of my master’s project at SJSU, a CPU-level Priority Based Power Management and Reduced Downtime tool for the data centers.

System Architecture

Initial Architecture

There was already a power-management tool for virtual machines developed by Jan Stoess, Christian Lang, and Frank Bellosa of System Architecture Group at University of Karlsruhe in Germany [8]. I initially decided to make use of this tool and add priority based power management and high runtime availability features to it. The following was my initial architecture of my software.

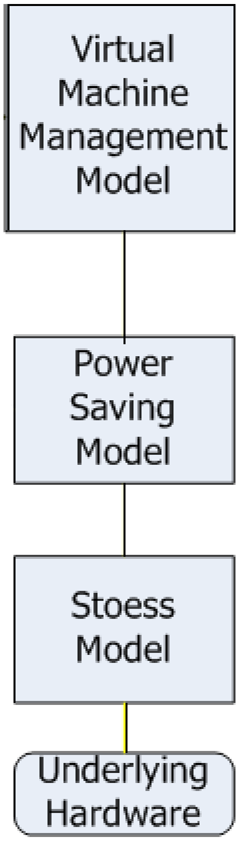


Figure 1: Initial Architecture

Problems with initial architecture

The initial architecture had following shortcomings.

1. Stoess' model was geared specifically towards XEN Virtualization [8]. As a result, my initial architecture would only work with XEN virtualization platform.
2. In addition, there is no technical support or documentation available for Stoess' model. Hence, it would be harder to extend my initial architecture to support multiple virtualization platforms if my architecture continues to rely on Stoess' model.
3. Finally, it was mentioned that Stoess' model is a "prototype architecture [8]." Therefore, I do not want to build my complete architecture on top of this "prototype" [8] entity.

New Architecture

Due to the reasons stated above, I decided to design a new architecture that does not rely on Stoess' model. The following is the architecture of my new Priority based Power Management and Reduced Downtime software where I did not use Stoess' model. The architecture consists of six layers.

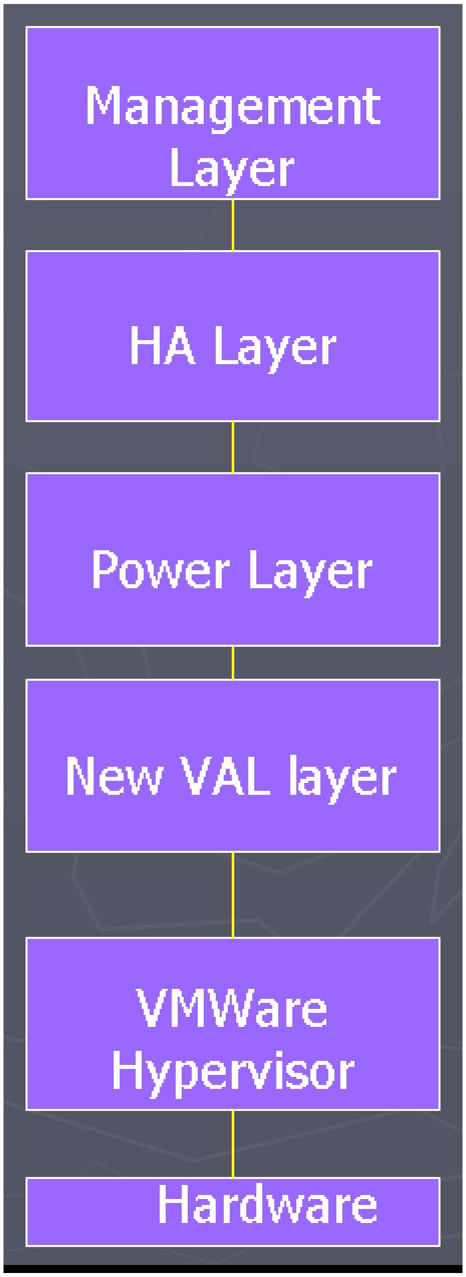


Figure 2: New Architecture

Priority Based Power Management & Reduced Downtime Layers

Hardware

The Hardware layer in the architecture represents the CPU of the servers in the Data Centers. Since I focused only on the servers' processors for the scope of this project, my hardware layer just deals with the server's processor. My software works with any kind of CPU although CPU needs to be a Virtualization Technology (VT) enabled processor in order to run any virtualization software such as VMWare, XEN, or Microsoft's HyperV. In addition to the energy savings from my new software, more power can be saved if the underlying processor adjusts its operating voltage or frequency based on its load to save power according to E.N. Elnozahy at IBM research [3]. Currently, there are only very few processor that has this capability [3].

Hypervisor

My new Priority Based Power Management & Reduced Downtime tool uses virtualization technique to optimize power. As a result, a hypervisor is needed. Based on Wikipedia's definition of hypervisor as of May 3, 2009, a hypervisor needs to sit on top of the system hardware directly without an operating system in between [4]. Moreover Wikipedia's definition states that the hypervisor itself is an operating system that communicates with the underlying hardware, and as a result, a hypervisor has direct access to the hardware enabling the hypervisor to control the hardware [4]. A hypervisor has APIs to control the CPU shares for a particular application or virtual machine running on top of the hypervisor. Furthermore, a hypervisor can fetch amount of CPU cycles consumed by a particular virtual machine application running on top of the hypervisor.

The hypervisor receives instruction from my VAL layer regarding how much cycles should be allocated to each virtual machine running on top of the hypervisor. The hypervisor simply processes the request of VAL layer. There are different vendors who provide different hypervisors. For this project, I used only VMware's hypervisor although my software is designed flexible enough to deal with other vendors' hypervisors without any code changes to the software.

VAL Layer

I developed this layer, and it acts as an interface between my Power Management layer and the underlying hypervisor. The hypervisor could be provided by vendors like Xen or VMWare. This interface is generic enough to support any vendor's hypervisor. This layer receives a request from Power Management layer to minimize or maximize the energy consumption for a particular virtual machine. The VAL layer then processes the request by first figuring out the type of underlying hypervisor and then issuing appropriate commands to the hypervisor. The VAL layer also fetches and calculates statistical information regarding the processor's performance from the hypervisor and transfers the calculated statistics to the GUI, HA and Power Management layer.

Power Saving Module

Power Saving Module is the intelligence of my Priority Based Power Management and Reduced Downtime tool. I developed this module in such a manner that it looks at the current virtual machines present in different servers and their respective priorities and analyzes how to distribute the virtual machines among available servers in a manner that minimizes power consumption. This module performs priority based load balancing among available hosts or servers.

The priority based load balancing mechanisms takes into account the Coordinated Voltage Scaling Policy from the following list of CPU power management schemes designed by E.N. Elnozahy at IBM research [3].

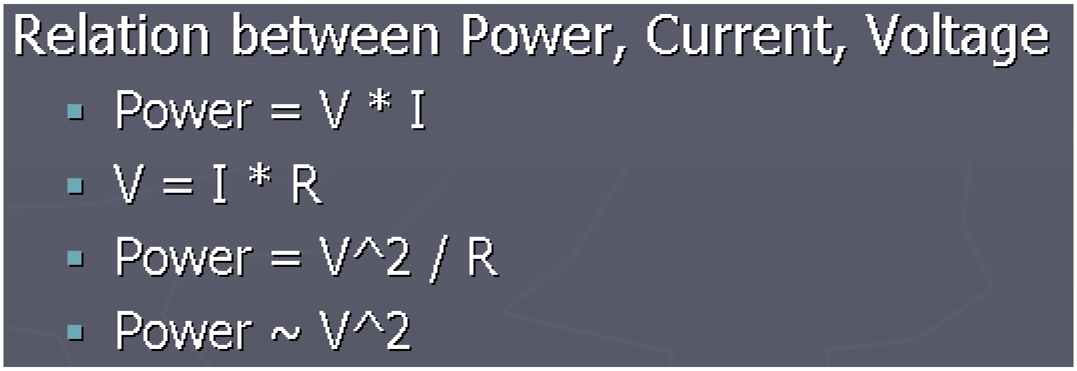
Different CPU Power Management Schemes

Independent Voltage Scaling

In this policy, the CPU automatically adjusts its operating voltage or frequency based on its load to save power [3]. Only few processors are capable of adjusting itself [3].

Coordinated Voltage Scaling

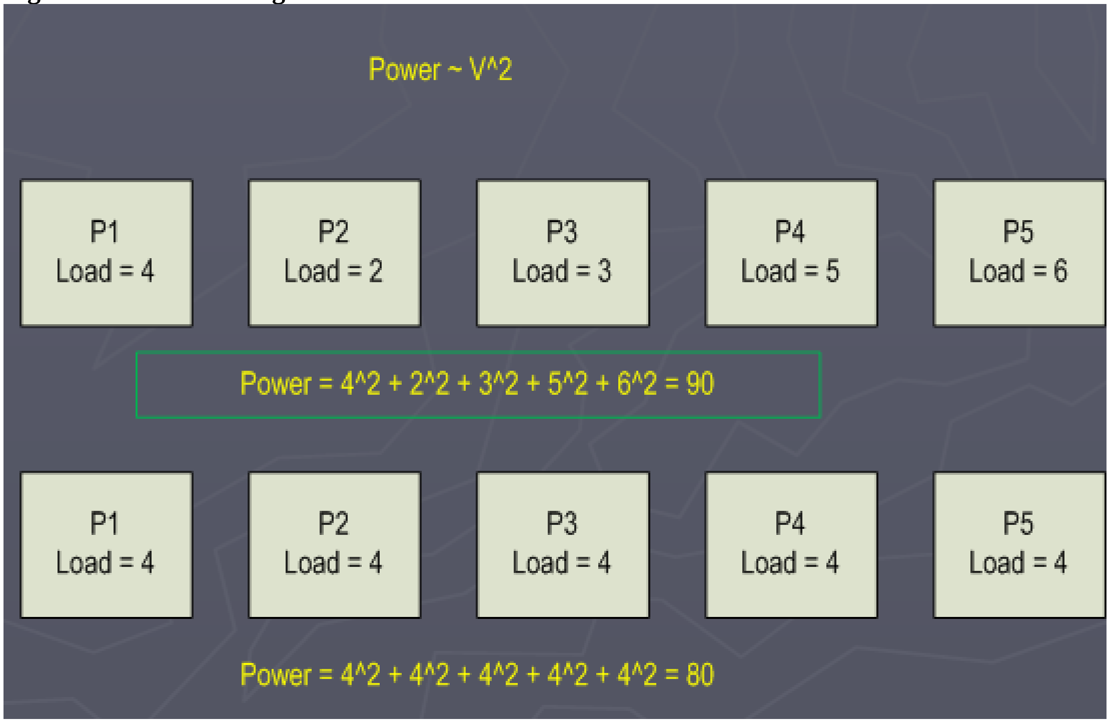
In this policy, all processors are operated at similar frequency to save power [3]. According to Vivek Sharma and Zhijian Lu of Computer Science department and Electrical and Computer Engineering department at University of Virginia respectively, “the energy saving are maximized when load is exactly balanced among the back-end machines [servers] [7].” They assert that their claim is based on the “nonlinear power voltage relation and the fact that the sum of squares (or higher order functions) of numbers that add up to the same total is minimized when these numbers are equal [7].” Therefore, I came up with following simple example to illustrate Sharma’s and Lu’s claim.



Relation between Power, Current, Voltage

- Power = V * I
- V = I * R
- Power = V^2 / R
- Power ~ V^2

Figure 3: Power - Voltage relation



Power ~ V^2

P1 Load = 4

P2 Load = 2

P3 Load = 3

P4 Load = 5

P5 Load = 6

Power = 4^2 + 2^2 + 3^2 + 5^2 + 6^2 = 90

P1 Load = 4

P2 Load = 4

P3 Load = 4

P4 Load = 4

P5 Load = 4

Power = 4^2 + 4^2 + 4^2 + 4^2 + 4^2 = 80

Figure 4: Load - Power relation

From my above example, one can notice that the power is directly proportional to square of the voltage. The processor’s voltage increases with the load. Hence, the power consumed approximately increases proportional to square of the load present in the processor. The example illustrates that if the load is distributed evenly among the processors, then the total power consumed is minimized as compared to total power consumed when the load is unevenly distributed among the processors. Finally, Sharma and Lu presented the following chart that showed “improvement” in energy efficiency when voltage scaling is used versus when the voltage scaling is not used [7].

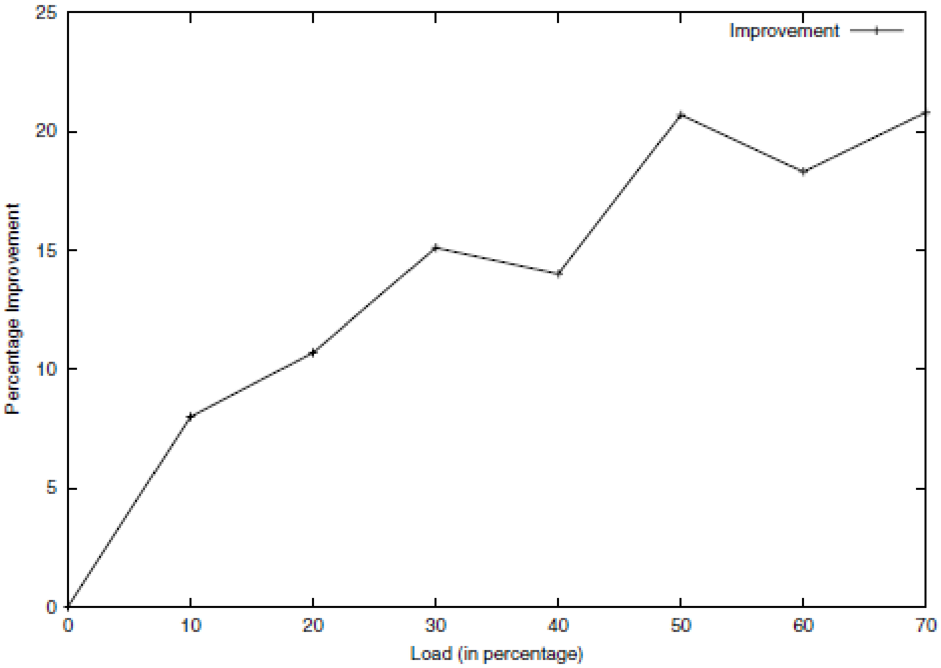


Figure 5: Increase in energy efficiency when voltage scaling is adopted [7]

From the above chart, I interpreted that the energy savings is maximized when the processor’s load is around 50%. As a result, in my software, I tried to maintain a uniform load around 50% throughout the servers.

Vary-on Vary Off (VOVO)

In this policy, the machines or servers, where the CPUs are idle, are switched off completely to save power [3]. If the load increases, then the machines are switched on to distribute the load [3]. This policy requires that the machine can be turned on or off remotely. Not all machines are capable of this. I didn’t implement this policy in my software.

Combined Policy (VOVO + IVS)

This is a combination of VOVO and IVS policies [3]. In this policy, the servers are switched off completely if the processors are idle [3]. Also, the processor, which is not idle, adjusts its operating frequency to match the load [3]. I didn’t implement this policy in my software.

Coordinated Policy (VOVO + CVS)

This is a combination of VOVO and CVS policies [3]. In this policy, the servers are switched off completely if the processors are idle [3]. In addition, an observer constantly examines the average operating frequency across all processors and broadcast this average to all processors so that the every processor can try to operate around this average frequency [3]. My software doesn’t implement this policy.

Reduced Downtime Management or HA Layer

This is the layer that provides high runtime availability or minimum downtime to the virtual machine applications running on the servers. I developed this layer, and it constantly monitors those virtual machines marked as critical by the user. If the machine or server where one or more critical virtual machines are running goes down due to disk crash, network failures, or something similar, then this layer detects such scenarios and automatically transfers the critical virtual machines to another machine or server that is running. The transfer is made possible by virtualization techniques provided by products such as VMWare.

The mission critical virtual machines will be unavailable or unresponsive for the time it takes to transfer these critical applications from the failed machine to the machine that is running. But, this time is minimal (in minutes) and trivial when compared to efforts needed to bring these applications up manually if this layer is not present.

The Reduced Downtime layer works independently of Power Management layer. That’s the Reduced Downtime layer can work even if my software doesn’t have a power management layer or operate in Green mode. In case of server or network failures, the Reduced Downtime takes precedence and suspends any Power Management monitor if present. Then, it transfers only the critical virtual machines to another working servers or hosts before bringing the Power Management monitor to running state if present.

Virtual Machine Management

This is the GUI of the Power Management tool. I developed this GUI such that the user sees all the servers in the data centers. In addition, the user can see all the Virtual Machines running inside servers.

The virtual machine runs a single application. Thus, the load of the server is determined by how many virtual machines are running on the system and the load of application running inside the virtual machine. If the load of a server is too much, then one or more virtual machines are transferred to another server where there is a lighter load.

Through this GUI, users can set priorities to virtual machines so that high priority virtual machine can always kick out a low priority virtual machine if the load of a system is heavy. In addition, in case of server or machine failure, the high priority virtual machines of the failed machine are automatically transferred to another running server or machine. If the running machine has any low priority virtual machines, then they will be ejected in order to make room for the incoming high priority virtual machines.

Initial Vs New Architectures

The following are the differences between the initial architecture using the Stoess' model and the new architecture.

1. The new architecture is not geared towards any specific virtualization platform. As a result, even if the VMWare hypervisor layer is removed from the new architecture diagram and replaced with a XEN hypervisor layer, the rest of the layers work seamlessly. This flexibility is significant since there are lots of virtualization vendors.
2. The new architecture welcomes support to new virtualization platforms gracefully. That is, even if a new virtualization platform is born tomorrow, the new architecture can still communicate with the new platform without any code modifications. This adaptability is important since we live in a world where changes happen too frequently. The explanation on how this adaptability was achieved will be explained in a later topic.

Tools Used

- VMWare virtualization software
- Java/Struts for business logic
- Adobe Flex for UI
- VT-enabled processor
- Tomcat server

Description of Deliverables

- **New system consists of following modules that I authored**:
  - **Virtualization Layer**: Interface that can communicate with 3rd party virtualization techniques such as XEN, Microsoft Virtual PC or VMware.

- **Power Saving Layer**: A module that implements power management schemes and performs priority based distribution of virtual machines based on load to optimize power consumption.
- **Reduced Downtime Layer**: A module that provides reduced downtime to mission critical virtual machines.
- **Virtual Machine Management Layer**: A management layer that gives data center's administrator a complete picture about how much power is being by consumed by different virtual machines. An administrator can set priorities for different virtual machines.

Software Architecture

The following are the list of classes, interfaces, and entities used in the Priority-Based Power Management and Reduced Downtime system. Every item in this list is developed by me except the VMWare SDK, which is owned by VMWare, Inc.

1. PowerManagement
2. HighAvailability
3. GenericDataCenter
4. DataCenter
5. SimulatedDataCenter
6. VMWareDataCenter
7. VMWare SDK
8. ActionClass
9. GraphicalUserInterface

Adaptable Architecture

The following architecture diagram illustrates the relationship between the classes listed above. In this architecture, a design pattern called Strategy pattern is followed. This is because the PowerManagement and HighAvailability classes only deal with the GenericDataCenter interface. As a result, these classes can deal with any classes that implement this interface. Hence, PowerManagement and HighAvailability classes can deal with DataCenter, SimulatedDataCenter or VMWareDataCenter classes. If there is a new virtualization vendor, then there will be a class or a SDK from this vendor. In order to make this new vendor's class intractable with PowerManagement and HighAvailability classes, the vendor’s class must implement the GenericDataCenter interface. This is the reason why the architecture is very adaptive to new virtualization vendors.

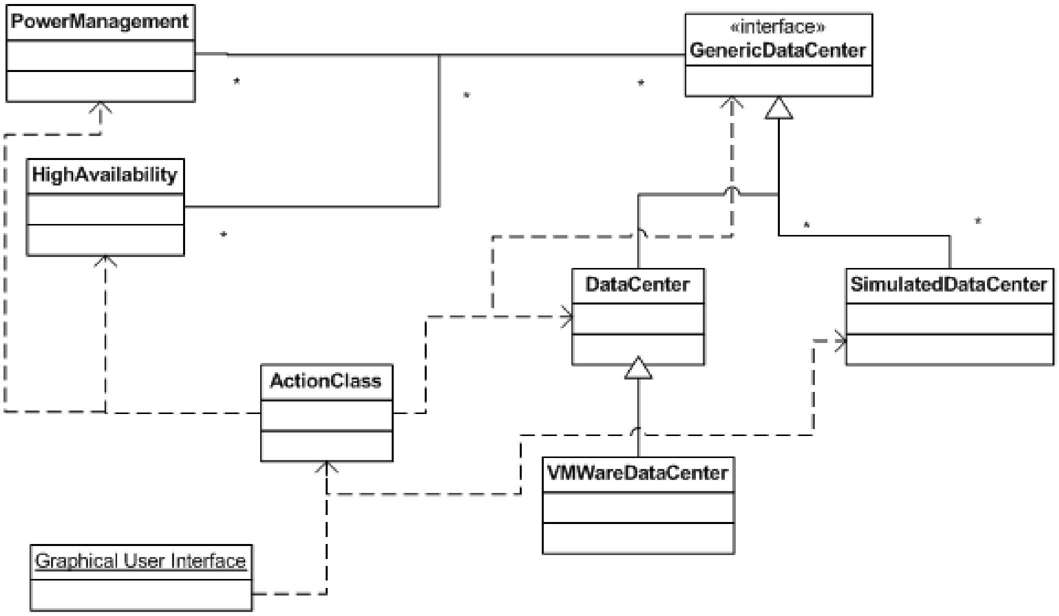


Figure 6: UML Class Diagram

PowerManagement Class

This is the class that deals with Priority-Based Power Management. This entity runs as a separate thread and monitors the generic data center constantly. It reads from the data center the virtual machines, virtual machines’ CPU usage, hosts, hosts’ CPU usage, and the priorities of each individual virtual machine. Based on these, this entity determines the best way to distribute the load among the available hosts or machines and powering-off the low-priority virtual machines for certain time such that the CPU load on each available host or server is close to 50%. Furthermore, if the CPU load on a host falls below 40%, then this entity tries to power-on any powered-off virtual machines based on priority until the CPU load on the host reaches close to 50%. This entity performs these activities periodically to main the CPU load on all the hosts around 50%.

HighAvailability Class

This is the class that keeps monitoring if any of the running machines in the data centers becomes faulted or not. If one or more hosts get faulted in the data centers, then this entity suspends any active Priority-Based Power Management and transfers all the virtual machines from the failed hosts to any available running hosts based on priority. Thus, the high priority or mission critical virtual machines always get migrated first to another running host to minimize their down time. This is the entity that ensures that mission critical virtual machines are available all the time with little or no down time.

GenericDataCenter Interface

This is an interface that generalizes the common functionalities of a data center. Functions such as starting a virtual machine, powering off a virtual machine, migrating a virtual machine are located here. The PowerManagement and HighAvailability classes use only this interface when dealing with data centers. As a result, my Power Management and Reduced Downtime features are applicable for any type of data center such as VMWareData Center or SimulatedData Center as long as they implement the functions mentioned in this GenericDataCenter.

DataCenter Class

This class deals with the Data Center. This class is generic enough to address data centers with different types of virtualization techniques such as VMWare or XEN implemented. This class implements the functionalities mentioned in the GenericDataCenterInferface.

SimulatedDataCenter Class

This is the class that aids in simulating the users' behavior of using the virtual machines in the Data Centers. This class implements the GenericDataCenter. As a result, it contains all the functionalities such as starting, stopping, and migrating a virtual machine.

Furthermore, a configuration file specifies the maximum operating frequency for each of virtual machine's processor located inside the GenericDataCenter. The configuration file is fed into the SimulatedDataCenter where the entity User Simulator extracts the maximum operating frequencies for each and every virtual machine specified in the configuration file. Then, for each extracted maximum operating frequency of a virtual machine, the User Simulator picks a random value between 0 and the maximum operating frequency and sets this random value as the operating frequency of corresponding virtual machine's processor.

The User Simulator performs the above procedure every 10 seconds to simulate the real random usage of virtual machines in a data center. The Power Management and Reduced Downtime modules are completely unaware of the fact that the data center they are dealing with is a simulated data center. The Power Management and Reduced

Downtime modules work independent of type of data center used on the other side. The following diagram illustrates the contents of SimulatedDataCenter class.

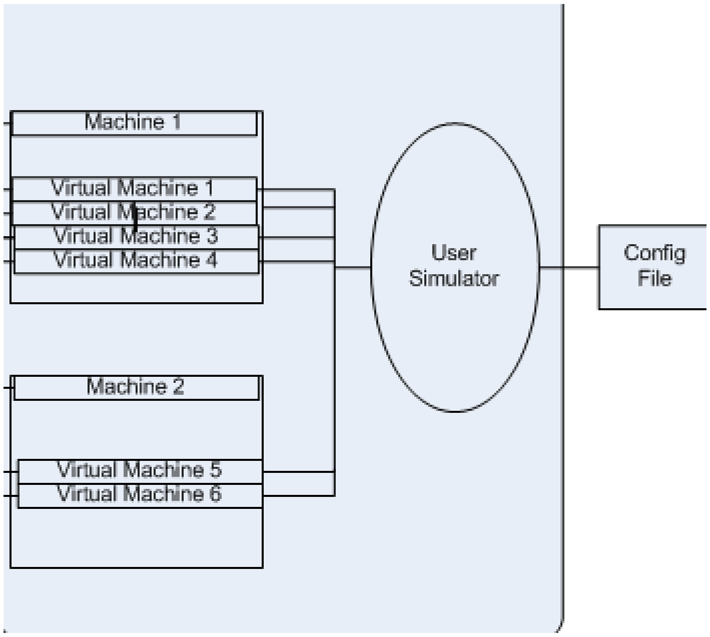


Figure 7: Simulator Setup

VMWareDataCenter Class

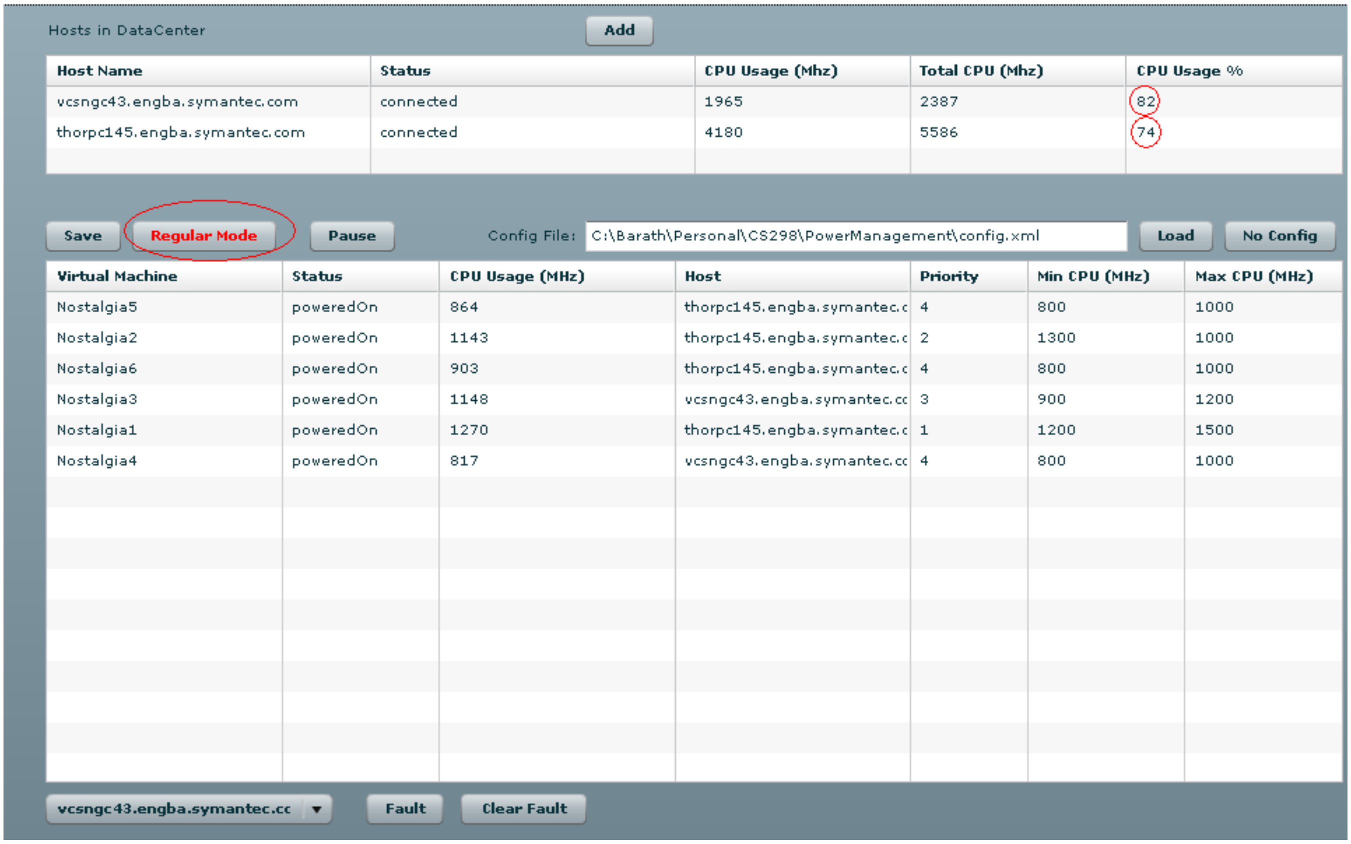
This is the class that represents the data centers containing hosts or servers that run on VMWare ESX Hypervisor operating systems. This class extends from DataCenter class, and thus, it contains all the functionalities such as starting, stopping, and migrating a virtual machine. In addition, this class implements these functionalities in a manner that is specific to the VMWare ESX Hypervisor operating system [9]. This class makes use of the Software Development Kit and APIs provided by VMWare [9] to implement its functionalities.

Action Class

This is the class that handles the request from Graphical User Interface, and it determines how to process the GUI’s request. It gets a refresh request from GUI every 10 seconds. In addition to that, GUI also requests this class for actions such as establishing connection with the data center, faulting hosts, clearing the faults on the hosts, asking to operate in simulator mode etc.

Graphical User Interface Entity

This is the entity that is visible to the administrator of a data center. Through this entity, the administrator of a data center can establish connection with a data center, see all the hosts and virtual machines in a data center, the CPU usage of each host and virtual machines present in a data center. In addition, an administrator can set priority for each virtual machine to let the Power Management and Reduced Downtime module know the priorities of the virtual machines. Furthermore, an administrator can choose to operate a data center in Regular mode or in Green Mode. The administrator toggles between the two modes of operating by clicking on the Green Mode or Regular Mode toggle button as shown in the following two screen shots.



Hosts in DataCenter

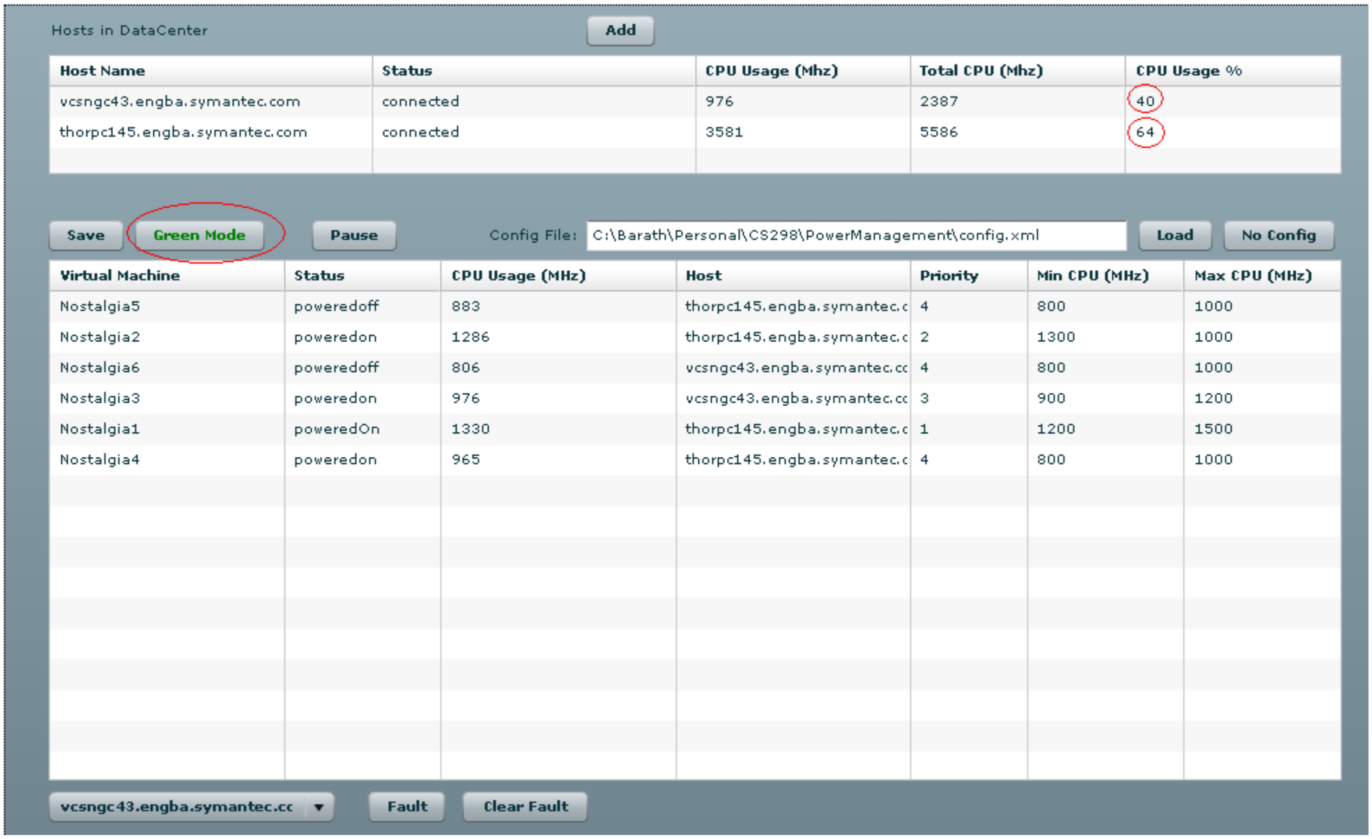
Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	1965	2387	82
thorpc145.engba.symantec.com	connected	4180	5586	74

Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredOn	864	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredOn	1143	thorpc145.engba.symantec.c	2	1300	1000
Nostalgia6	poweredOn	903	thorpc145.engba.symantec.c	4	800	1000
Nostalgia3	poweredOn	1148	vcsngc43.engba.symantec.cc	3	900	1200
Nostalgia1	poweredOn	1270	thorpc145.engba.symantec.c	1	1200	1500
Nostalgia4	poweredOn	817	vcsngc43.engba.symantec.cc	4	800	1000

vcsngc43.engba.symantec.cc ▼

Figure 8: Regular Mode



Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	976	2387	40
thorpc145.engba.symantec.com	connected	3581	5586	64

Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

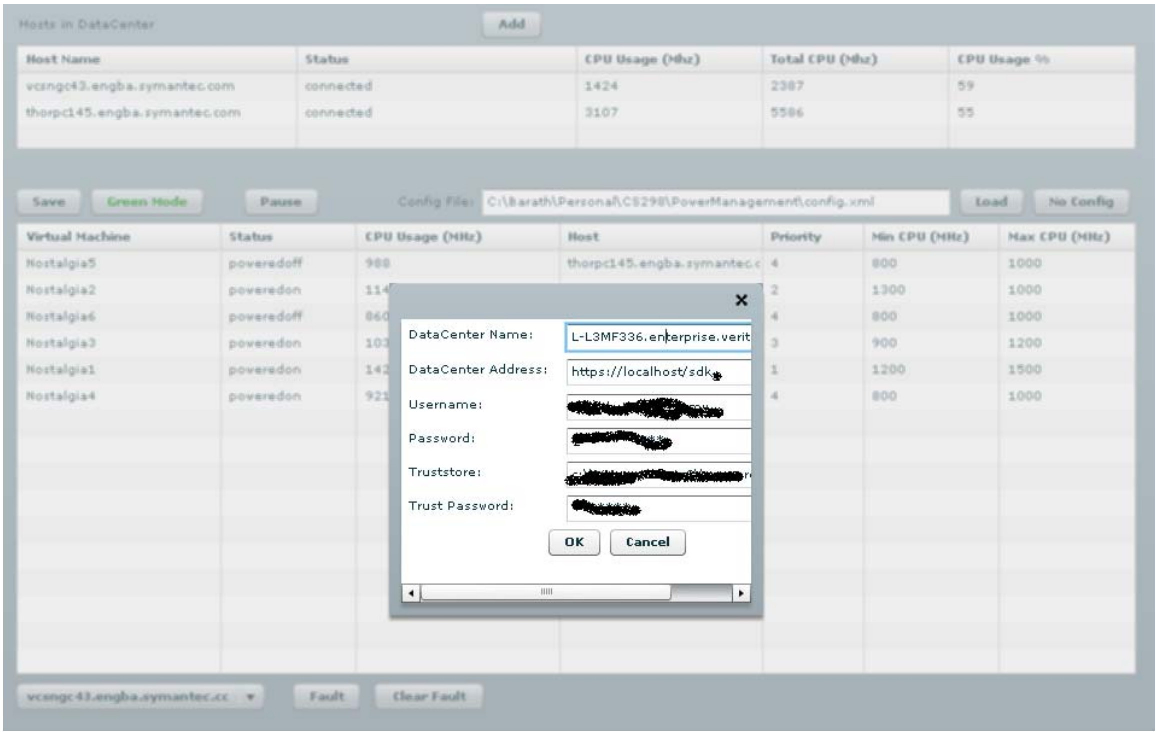
Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredoff	883	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredon	1286	thorpc145.engba.symantec.c	2	1300	1000
Nostalgia6	poweredoff	806	vcsngc43.engba.symantec.cc	4	800	1000
Nostalgia3	poweredon	976	vcsngc43.engba.symantec.cc	3	900	1200
Nostalgia1	poweredOn	1330	thorpc145.engba.symantec.c	1	1200	1500
Nostalgia4	poweredon	965	thorpc145.engba.symantec.c	4	800	1000

Figure 9: Green Mode

Furthermore, through GUI, one can specify the configuration file to make the Priority-Based Power Management and Reduced Downtime system to operate in a simulation mode by entering the complete path to the configuration file and clicking on the Load button. To resume normal mode or to exit the simulation mode, the user has to click on the No Config button.

Software Usability Manual

An administrator of a data center opens a browser and goes to the URL http://localhost:8080/gui to access the user interface for the Priority-Based Power Management and Reduced Downtime system. In that page, the user clicks on the Add button to bring up a pop-up which prompts for administrator's credentials to login to the data center. The pop-up is shown below.



Hosts in DataCenter

Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	1424	2387	59
thorpc145.engba.symantec.com	connected	3107	5586	55

Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredoff	988	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredon	114		2	1300	1000
Nostalgia6	poweredoff	860		4	800	1000
Nostalgia3	poweredon	103		3	900	1200
Nostalgia1	poweredon	142		1	1200	1500
Nostalgia4	poweredon	921		4	800	1000

DataCenter Name: L-L3MF336.enterprise.verit

DataCenter Address: https://localhost/sdk

Username: [redacted]

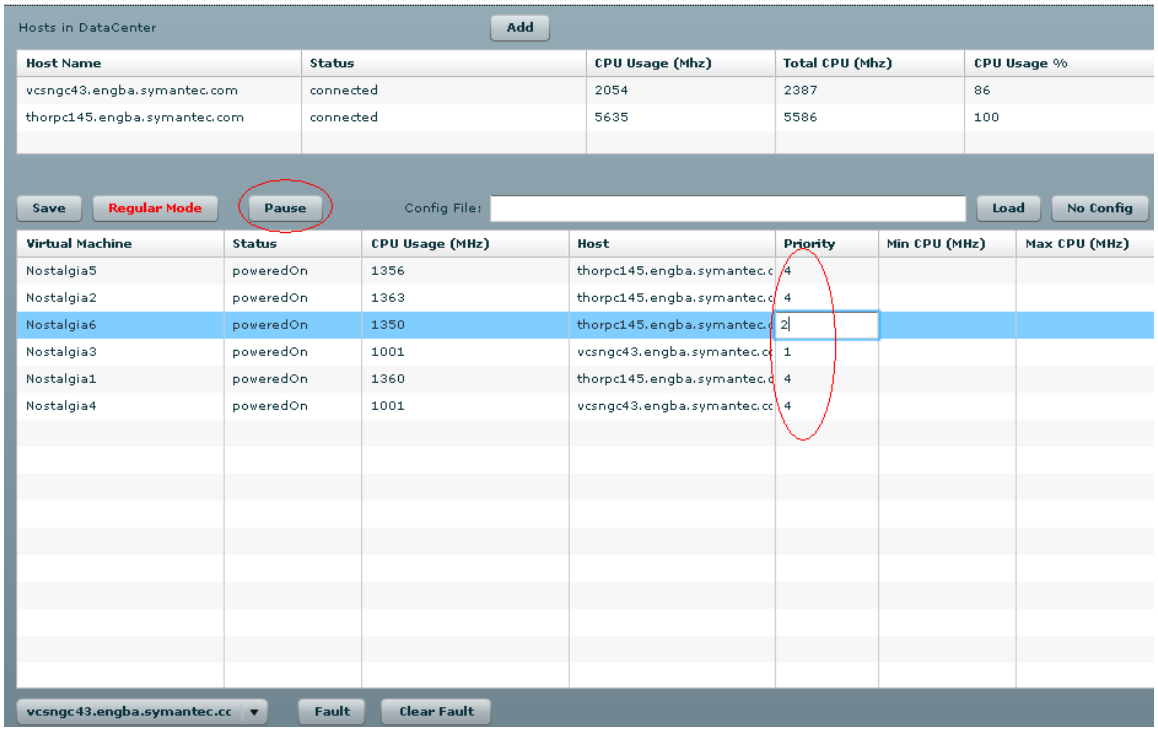
Password: [redacted]

Truststore: [redacted]

Trust Password: [redacted]

Figure 10: Credential Popup

Once the administrator's credentials are verified, then the administrator is shown with complete details of the data center such as the hosts and virtual machines present in the data center, and their CPU usages. The administrator then clicks on the Pause button and assigns priorities as shown in the screenshot below.



Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	2054	2387	86
thorpc145.engba.symantec.com	connected	5635	5586	100

Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredOn	1356	thorpc145.engba.symantec.c	4		
Nostalgia2	poweredOn	1363	thorpc145.engba.symantec.c	4		
Nostalgia6	poweredOn	1350	thorpc145.engba.symantec.c	2		
Nostalgia3	poweredOn	1001	vcsngc43.engba.symantec.cc	1		
Nostalgia1	poweredOn	1360	thorpc145.engba.symantec.c	4		
Nostalgia4	poweredOn	1001	vcsngc43.engba.symantec.cc	4		

Figure 11: Assign Priority

Once an administrator has finished assigning priorities to the virtual machines, the administrator clicks on the Save button to save his or her changes. Then, the administrator clicks on the toggle button “Regular Mode” to operate on “Green mode” where Priority-Based Power Management scheme takes into effect. The user can again click on this toggle button to come back in regular mode.

Experiment

Setup

A series of experiments were performed in order to determine the effectives of my proposed Priority-Based power management and Reduced Downtime system. The aim of my experiments is to determine whether my new software tool can make all the servers in the data center operate close to 50 % CPU load range, recommended by Sharma and Lu to conserve energy usage [7], by ejecting or powering-off any low priority virtual machines without compromising on the promise that the high priority virtual machines have minimum downtime. The setup shown below is the ideal setup since the virtualization layer that I created directly interacts with a real data center. This virtualization layer has the capacity to fully interact with a live data center and perform both power management and reduced downtime for virtual machines in the data center. However, for testing purposes, the setup shown below has few problems.

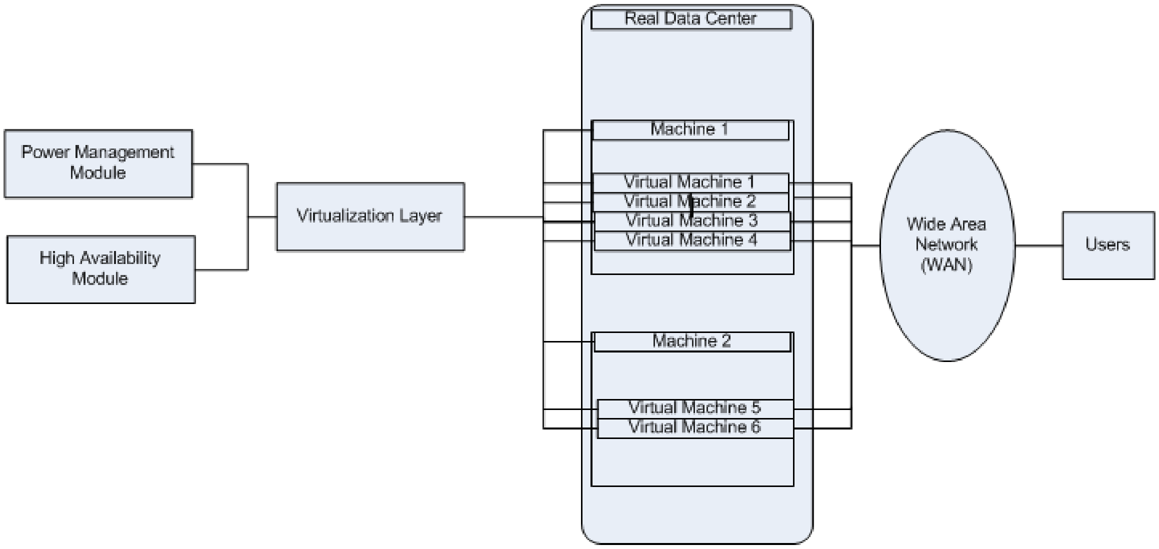


Figure 12: Ideal Setup

The following are the problems implementing the above setup for testing purpose:

1. Users are not present. Hence the virtual machines do not get used and CPU usage of these virtual machines remains steady as a result.
2. We cannot force the real virtual machines to operate in such a manner that it consumes specific amount of CPU load. There are no SDKs or APIs to accomplish this.
3. Virtual Machines are independent entities. Once we configure the virtual machine with Maximum CPU limit, the virtual machine operates on its own thinking that it

has its own processor with capacity equal to Maximum CPU limit. The virtual machine may use its allocated processor’s fullest capacity or none at all. It depends on how the rigorously the virtual machines are being used by the users. All we know from the management’s perspective is that virtual machine’s CPU usage varies from 0 to Maximum CPU limit. Therefore, we cannot control the virtual machine’s CPU usage to a specific value located between 0 and Maximum CPU limit.

As a result, the experiments were conducted using simulator for data center rather than actual Data Center since it is impossible to set the CPU usage pattern for different virtual machines present in the data center without actual users using the virtual machines. The simulator set up used for experiment is shown below:

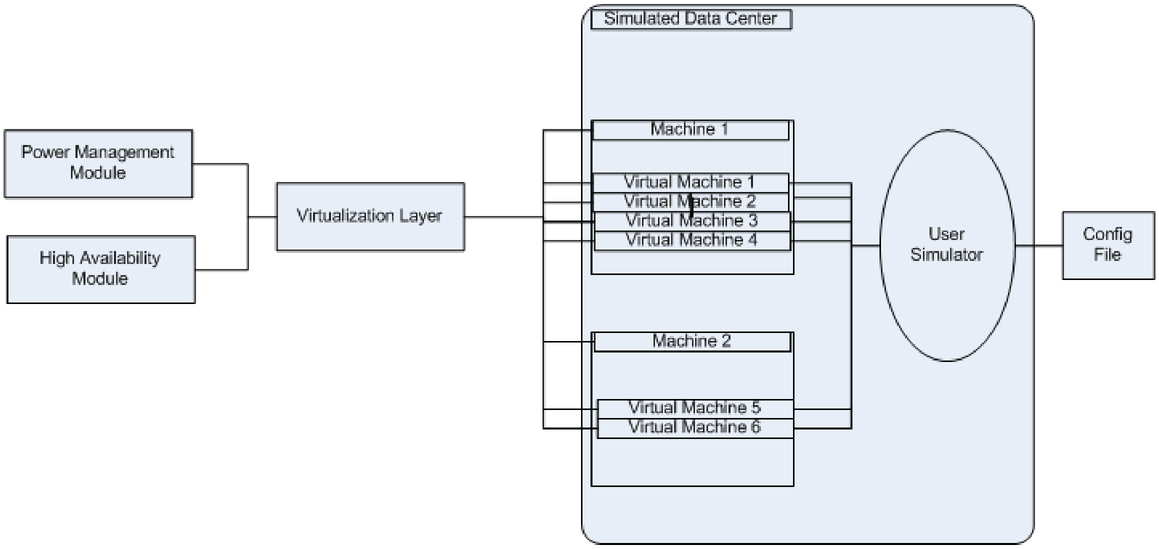


Figure 13: Simulated Setup

In the above setup, a configuration file specifies the maximum operating frequency for each of virtual machines’ processors. The configuration file is fed into the

Simulated Data Center where the entity User Simulator extracts the maximum operating frequencies for each and every virtual machine specified in the configuration file. Then, for each extracted maximum operating frequency of a virtual machine, the User Simulator picks a random value between 0 and the maximum operating frequency and sets this random value as the operating frequency of corresponding virtual machine's processor.

The User Simulator performs the above procedure every 10 seconds to simulate the real random usage of virtual machines in the data center. The Power Management and Reduced Downtime modules are completely unaware of the fact that the data center they are dealing with is a simulated data center. The Power Management and Reduced Downtime modules work independent of type of data center used on the other side.

The test environment contains the following

1. Host or Server 1 : vcsngc43.engba.symantec.com
2. Host or Server 2 : thorpc145.engba.symantec.com

Host 1 contains following virtual machines

1. Nostalgia3
2. Nostalgia 4

Host 2 contains following virtual machines

1. Nostalgia 1
2. Nostalgia 2
3. Nostalgia 5
4. Nostalgia 6

All the virtual machines specified above are simple game applications downloaded from VMware's website [10].

The tests were performed in two modes. First, the test was performed in regular mode where the Power Management module was turned off. This was performed to determine how much power is consumed without any Priority-Based Power Management. The same configuration for virtual machines is used in both regular and Priority-Based Power Management modes or Green mode. The following are the contents of the configuration file:

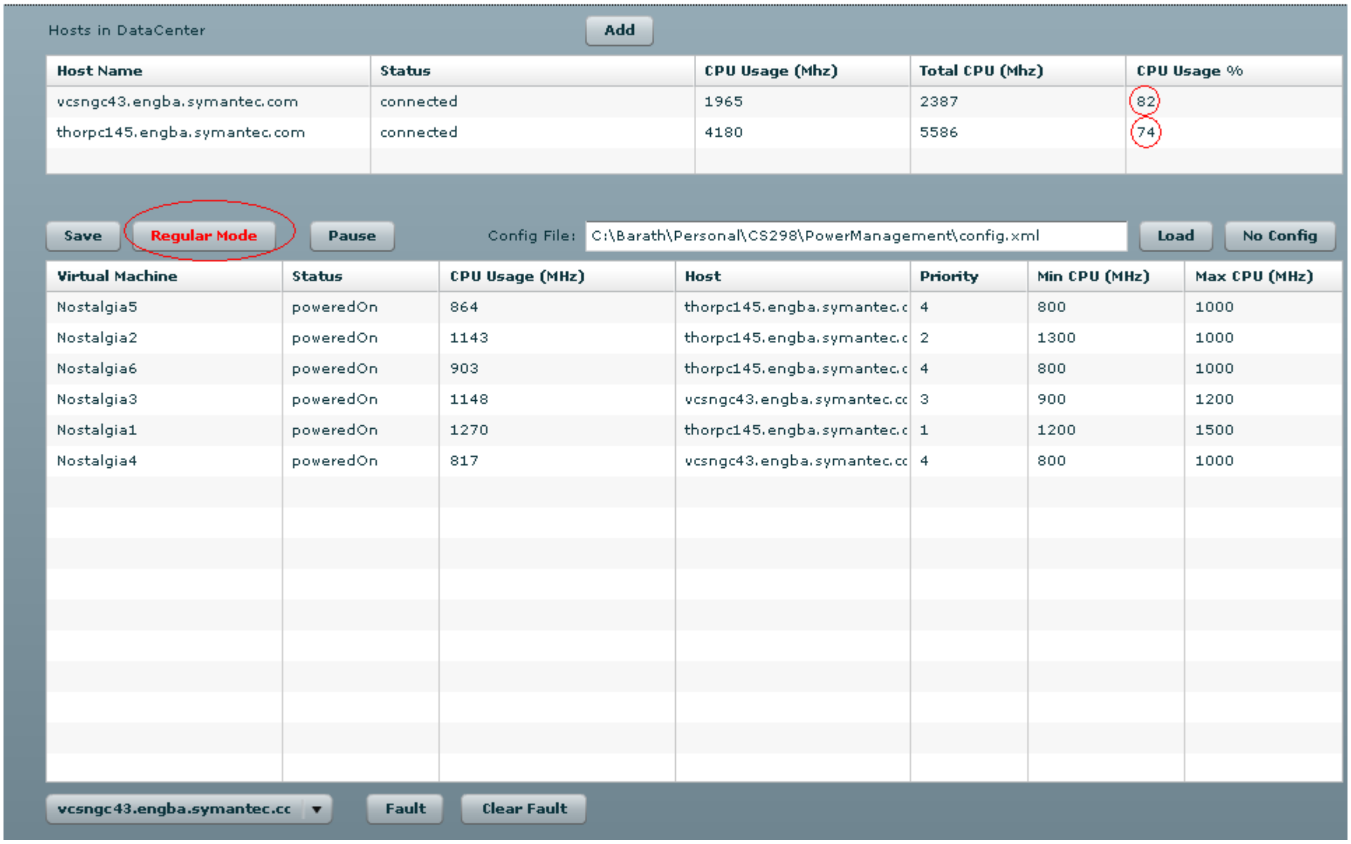
```
<datacenter>
        <virtualmachines>
                <virtualmachine name="Nostalgia5" max="1000" priority="4" min="800"/>
                <virtualmachine name="Nostalgia4" max="1000" priority="4" min="800"/>
                <virtualmachine name="Nostalgia1" max="1500" priority="1" min="1200"/>
                <virtualmachine name="Nostalgia3" max="1200" priority="3" min="900"/>
                <virtualmachine name="Nostalgia6" max="1000" priority="4" min="800"/>
                <virtualmachine name="Nostalgia2" max="1000" priority="2" min="1300"/>
        </virtualmachines>
</datacenter>
```

From the above configuration file, we can notice that virtual machine Nostalgia1 gets the highest priority of 1. In addition, this virtual machine has a maximum CPU limit of 1500 Mhz. This states that we’re allocating a processor to this virtual machine that has a range of 0 Mhz to 1500 Mhz. The virtual machine will start to think that it has its own processor that has operating range between 0 Mhz to 1500 Mhz. The virtual machine then operates in such a manner that its processor usage varies between 0 Mhz to 1500 Mhz. The min value in the configuration file specifies the minimum processor speed that this virtual machine requires. This information is solely used by the hosts or machines to determine if it has enough speed left in its processor to host this virtual machine. This min value has no bearing on the way virtual machine operates.

Results

With No Priority-Based Power Management

This test is conducted to determine the CPU load exerted by all the virtual machines with configurations specified in the configuration file on their hosts without any Priority-Based Power Management over a period of time. The following screenshot illustrates running the test without Priority-Based Power Management.



Hosts in DataCenter

Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	1965	2387	82
thorpc145.engba.symantec.com	connected	4180	5586	74

Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredOn	864	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredOn	1143	thorpc145.engba.symantec.c	2	1300	1000
Nostalgia6	poweredOn	903	thorpc145.engba.symantec.c	4	800	1000
Nostalgia3	poweredOn	1148	vcsngc43.engba.symantec.cc	3	900	1200
Nostalgia1	poweredOn	1270	thorpc145.engba.symantec.c	1	1200	1500
Nostalgia4	poweredOn	817	vcsngc43.engba.symantec.cc	4	800	1000

Figure 14: Operating in Regular Mode

The time duration during which this test was conducted is 100 seconds. Every 10 seconds, the CPU load of the hosts changes since the configuration file loads new CPU usage pattern for each virtual machine present in the configuration file every 10 seconds. The graph shown below summarizes the results of this test.

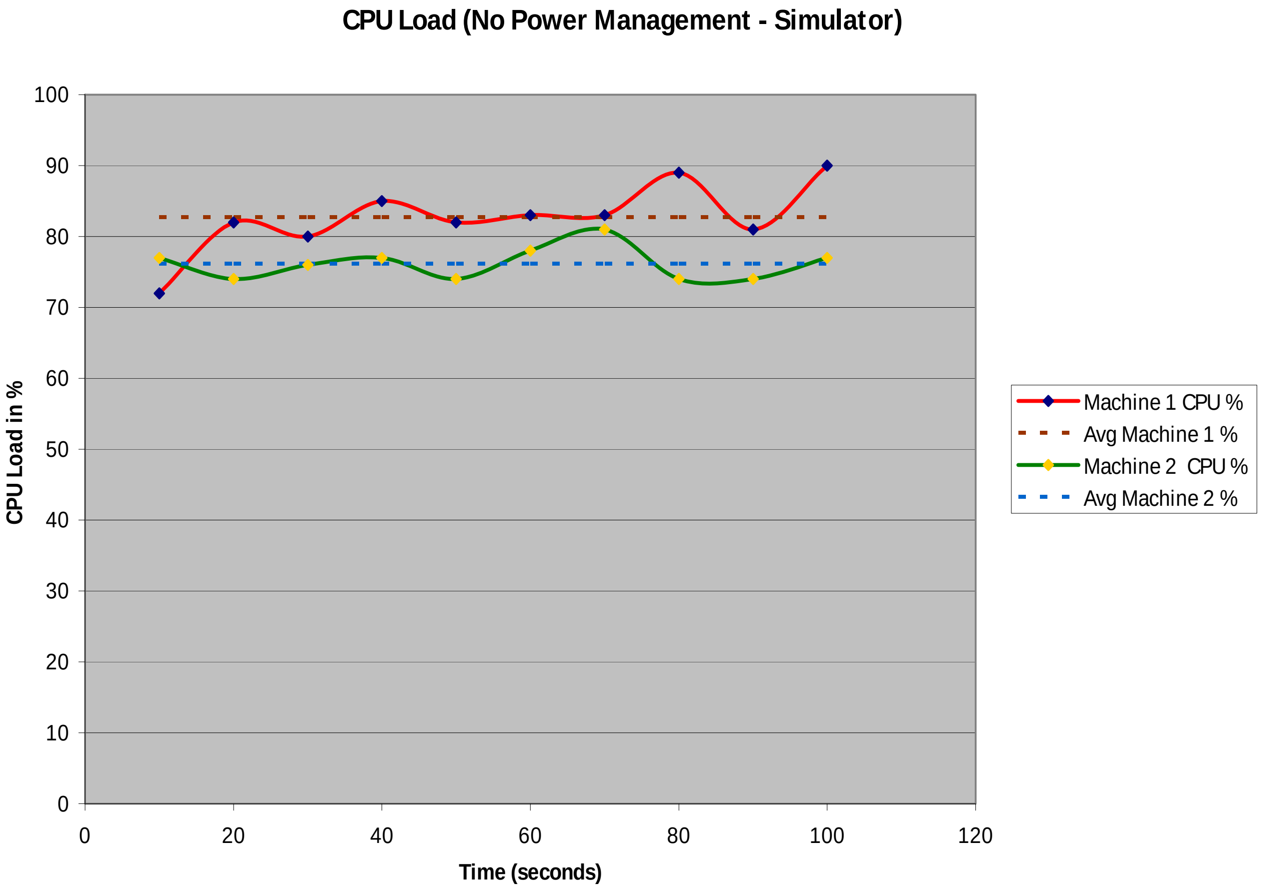


Figure 15: CPU Load in Regular Mode

As per the above graph, the CPU load of machine 1 varies between 70% and 90%. Similarly, the CPU load of machine 2 varies between 70% and 80%. Thus, the average CPU load on machine 1 is close to 82%, and the average CPU load on machine 2 is close to 76% without any active Priority-Based Power Management for the virtual machines with configurations specified in the configuration file.

We can notice from the graph that the average CPU load on both the machines are way beyond the average CPU load of 50% recommended by Sharma and Lu in order to minimize energy consumption [7]. Hence, in this regular setup, the energy usage is not optimized based on Sharma's and Lu's recommendations.

The next graph shows a summary of approximate power consumption during the test. The power is computed as follows:

Power ~ Voltage^2

Voltage ~ CPU Load

Hence, Power ~ CPU Load^2

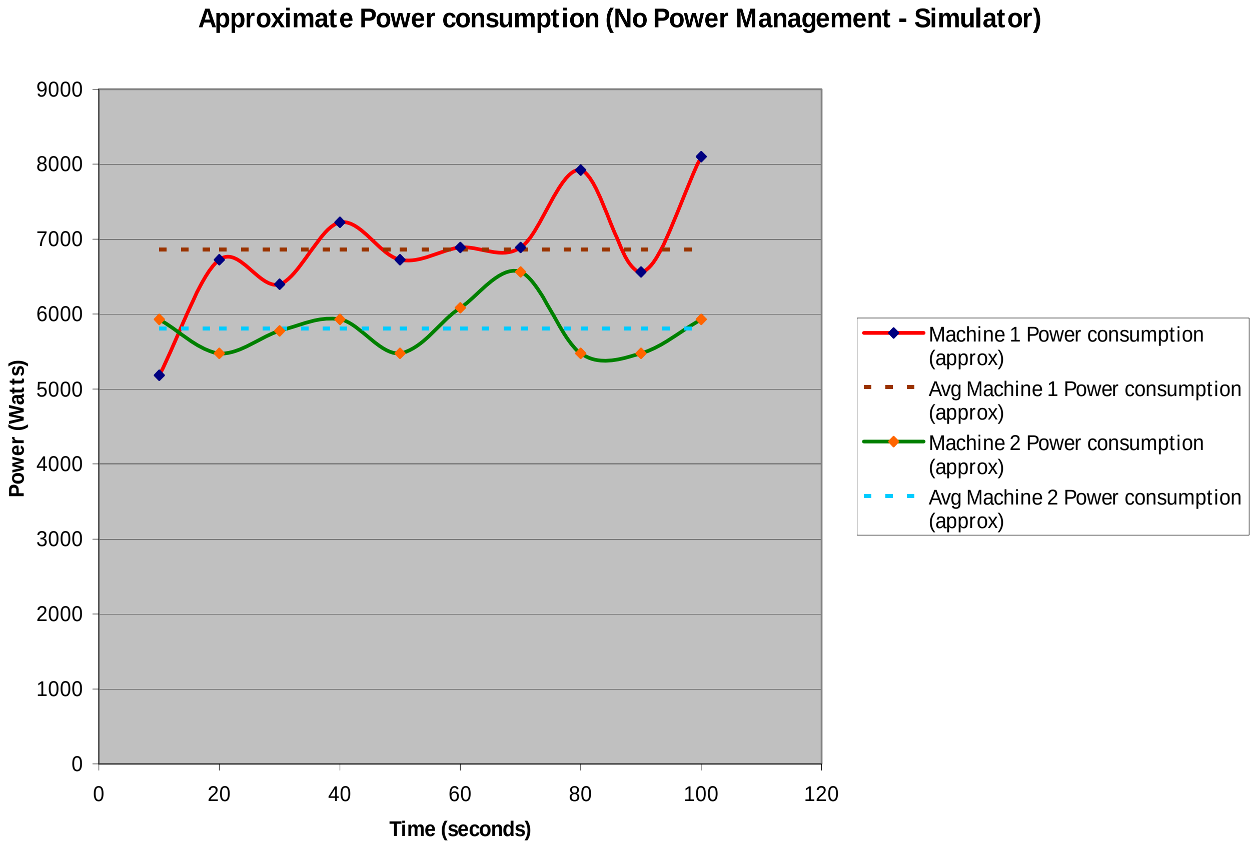
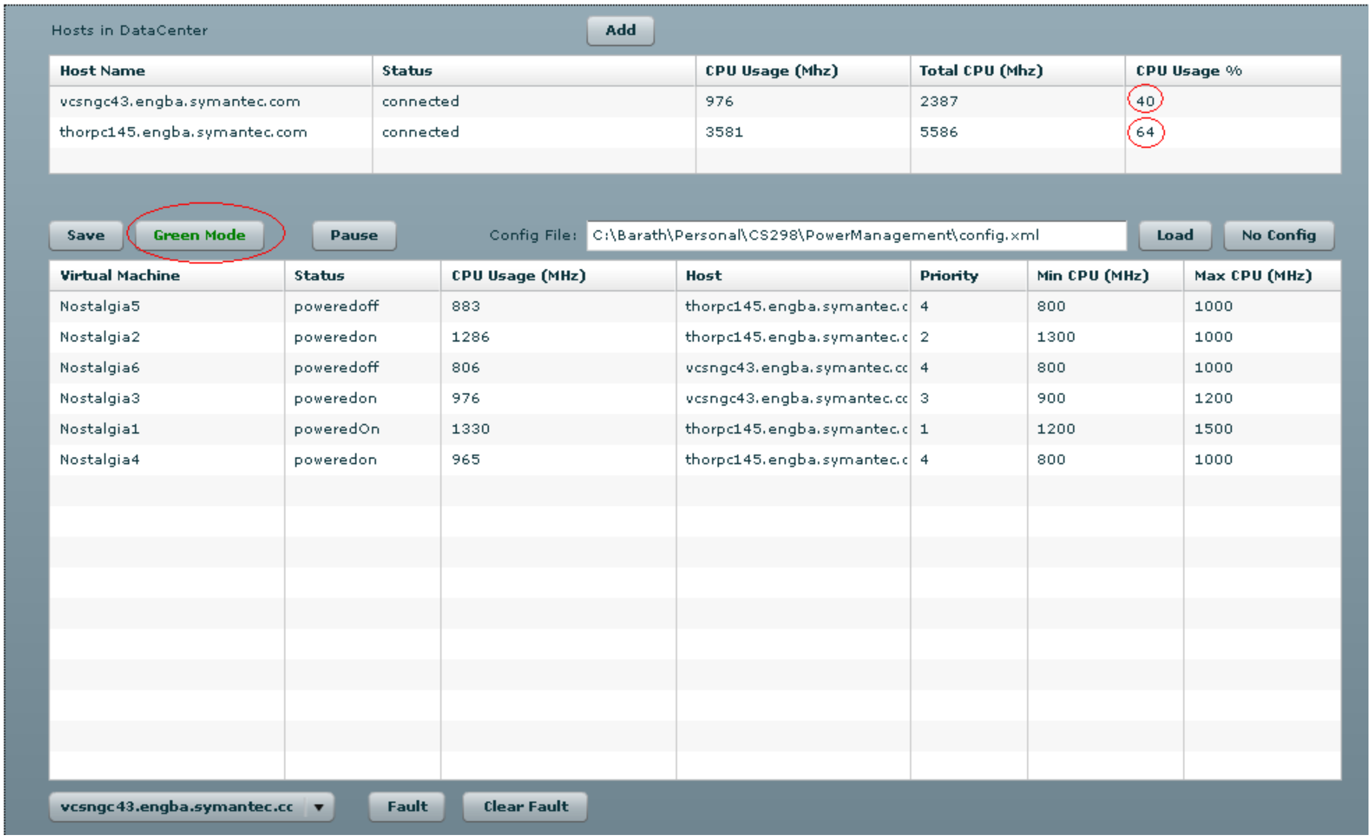


Figure 16: Approximate Power consumption in Regular Mode

As per the above graph, the power consumption of machine 1 varies between 5000 Watts and 8000 Watts. Similarly, the power consumption of machine 2 varies between 5000 Watts and 7000 Watts. Thus, the average power consumption of machine 1 is close to 7000 Watts, and the power consumption on machine 2 is close to 6000 Watts without any active Priority-Based Power Management for the virtual machines with configurations specified in the configuration file.

With Priority-Based Power Management

Next, a test is conducted to determine the CPU load exerted by all the virtual machines with the same configuration with Priority-Based Power Management over a period of time. The following screenshot illustrates running of this test with Priority-Based Power Management or in Green Mode.



Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	976	2387	40
thorpc145.engba.symantec.com	connected	3581	5586	64

Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredoff	883	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredon	1286	thorpc145.engba.symantec.c	2	1300	1000
Nostalgia6	poweredoff	806	vcsngc43.engba.symantec.cc	4	800	1000
Nostalgia3	poweredon	976	vcsngc43.engba.symantec.cc	3	900	1200
Nostalgia1	poweredOn	1330	thorpc145.engba.symantec.c	1	1200	1500
Nostalgia4	poweredon	965	thorpc145.engba.symantec.c	4	800	1000

Figure 17: Operating in Green Mode

The time duration during which this test was conducted is 100 seconds. Every 10 seconds, the CPU load of the hosts changes since the configuration file loads new CPU usage pattern for each virtual machine present in the configuration file every 10 seconds. The graph shown below summarizes the results of this test.

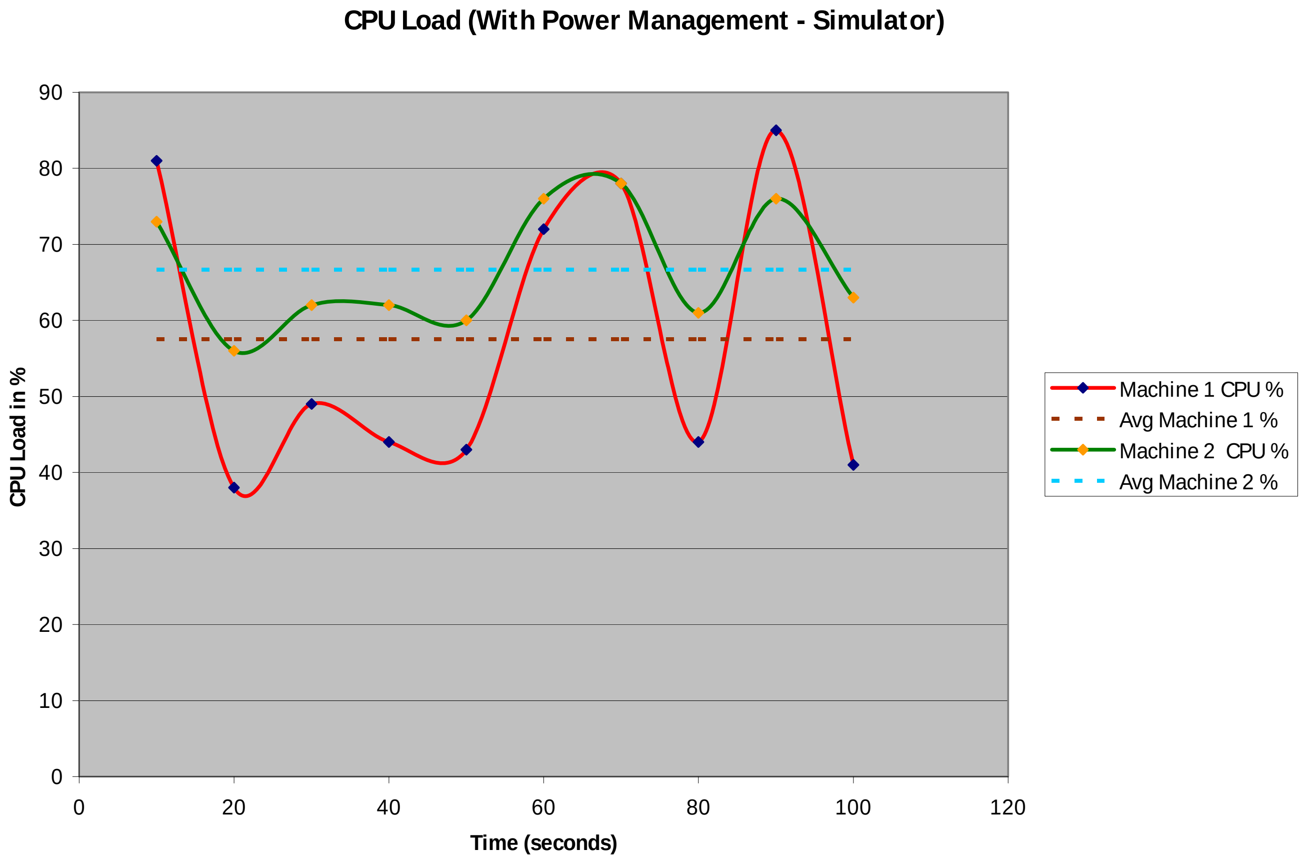


Figure 18: CPU Load in Green Mode

As per the above graph, the CPU load of machine 1 now varies between 50% and 80% which is a reduction from 70 % - 90 % range we noticed for the test conducted without Priority-Based Power Management. Similarly, the CPU load of machine 2 varies between 50% and 70% which is a reduction from 70 % - 80 % range we noticed for the test conducted without Priority-Based Power Management. Thus, the average CPU load on machine 1 is around 58%, and the average CPU load on machine 2 is close to 66% with active Priority-Based Power Management for the virtual machines with configurations specified in the configuration file. Hence, the CPU loads on the two machines are 58 % and 66 % respectively. These CPU loads closely resemble the ideal CPU load of 50% recommended by Sharma and Lu in order to minimize energy consumption [7].

Furthermore, the following graph illustrates that the approximate average power consumption of both the hosts are down as well with the new Priority-Based Power Management. Again, the approximate power consumption is computed using methods mentioned earlier.

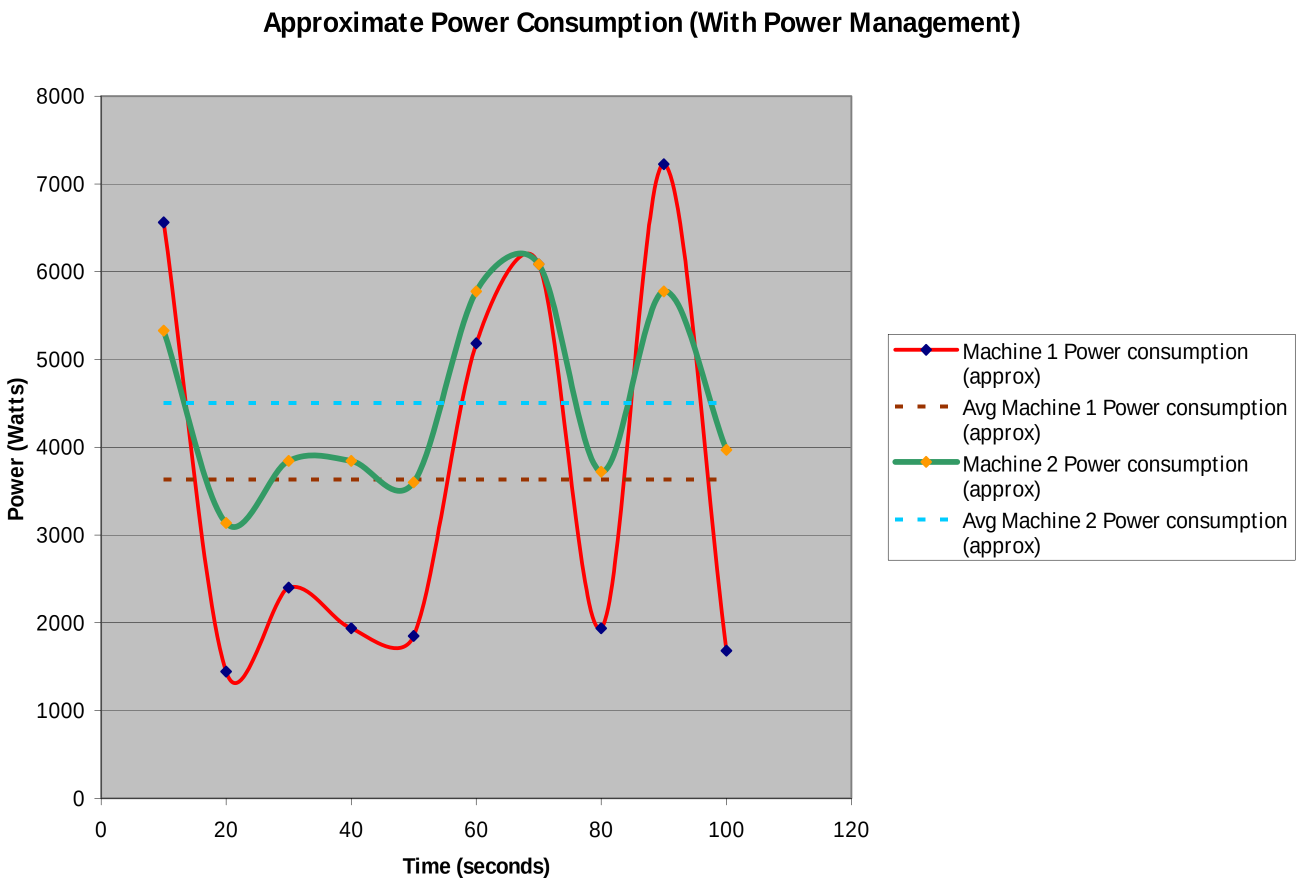


Figure 19: Approximate Power consumption in Green Mode

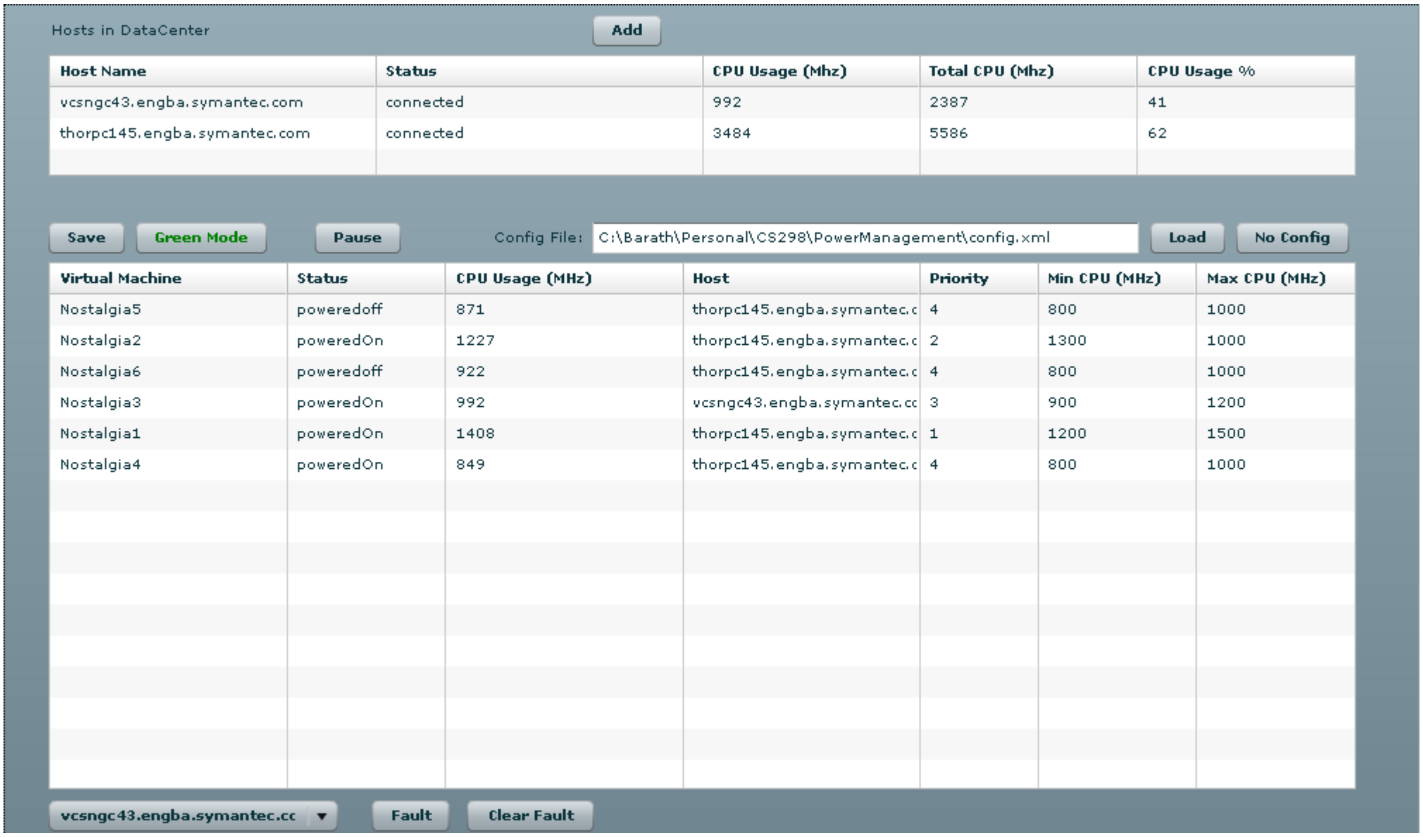
As per the above graph, the power consumption of machine 1 varies now between 1000 Watts and 7000 Watts. Similarly, the power consumption of machine 2 varies now between 3000 Watts and 6000 Watts. These ranges are way below the ranges we noticed when we ran the tests without any Priority-Based Power Management. Furthermore, the average power consumption of machine 1 now is close to 3800 Watts which is down from

7000 Watts we noticed when we ran the test without any Priority-Based Power Management. Similarly, the power consumption on machine 2 is close to 4500 Watts which is down from 6000 Watts we noticed when we ran the test without any Priority-Based Power Management. Thus, the average reduction in energy consumption in both the machines combined is close to 35%.

With Priority-Based Power Management, one can notice from the above graph that the power consumption alternate frequently between high and low. This is due to the fact that I ran my experiments with only 2 machines. As a result, there is constant relocation of virtual machines between the 2 machines by Priority-Based Power Management monitor. This resulted in CPU load and approximate power consumption to alternate frequently between high and low frequently. I believe that this power consumption will smooth out if the data centers contains many more servers or hosts since there will not be frequent relocation of virtual machines in such cases. Hence, we can conclude that the new Priority-Based Power Management does in fact reduce the average power consumption of hosts in data centers over a period of time.

Testing Reduced Downtime

The following screenshot show a data center being operated in a power-efficient manner using Priority-Based Power Management and Reduced Downtime system where the mission critical virtual machines Nostalgia1 and Nostalgia2 are running together in a single host.



Hosts in DataCenter

Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	992	2387	41
thorpc145.engba.symantec.com	connected	3484	5586	62

Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

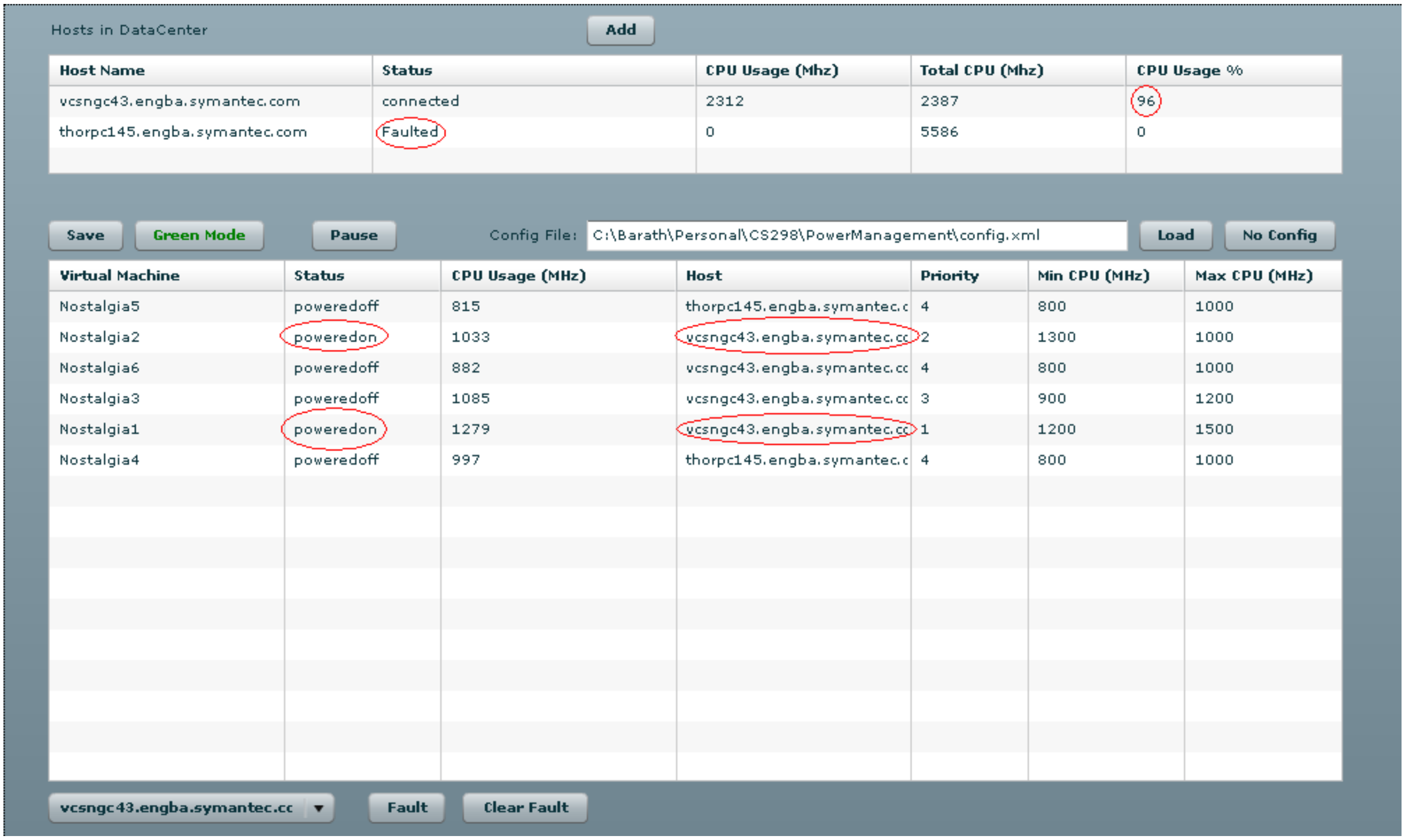
Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredoff	871	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredOn	1227	thorpc145.engba.symantec.c	2	1300	1000
Nostalgia6	poweredoff	922	thorpc145.engba.symantec.c	4	800	1000
Nostalgia3	poweredOn	992	vcsngc43.engba.symantec.cc	3	900	1200
Nostalgia1	poweredOn	1408	thorpc145.engba.symantec.c	1	1200	1500
Nostalgia4	poweredOn	849	thorpc145.engba.symantec.c	4	800	1000

vcsngc43.engba.symantec.cc ▼

Figure 20: Critical Virtual Machines and No Faulted Hosts

Unfortunately, we’re not living in an ideal world where every thing works perfectly forever. In a real world, one of the machines can crash and could be rendered useless. If the crashed machine was hosting a mission critical application, then there will be detrimental consequences.

Fortunately, my software handles unprecedented events such as computer crashes so that mission critical applications are automatically transferred from crashed machine to another machine which is running fine. In order to test this feature of my software, I simulate a fault on the machine where mission critical applications are running. The reason why I simulate a fault rather than actually faulting or shutting down the machine is that I cannot bring this system back online remotely after faulting or shutting down the machine remotely. The following screenshot indicates what will happen if the machine where high priority virtual machines Nostalgia1 and Nostalgia2 are running faults or dies.



Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	2312	2387	96
thorpc145.engba.symantec.com	Faulted	0	5586	0

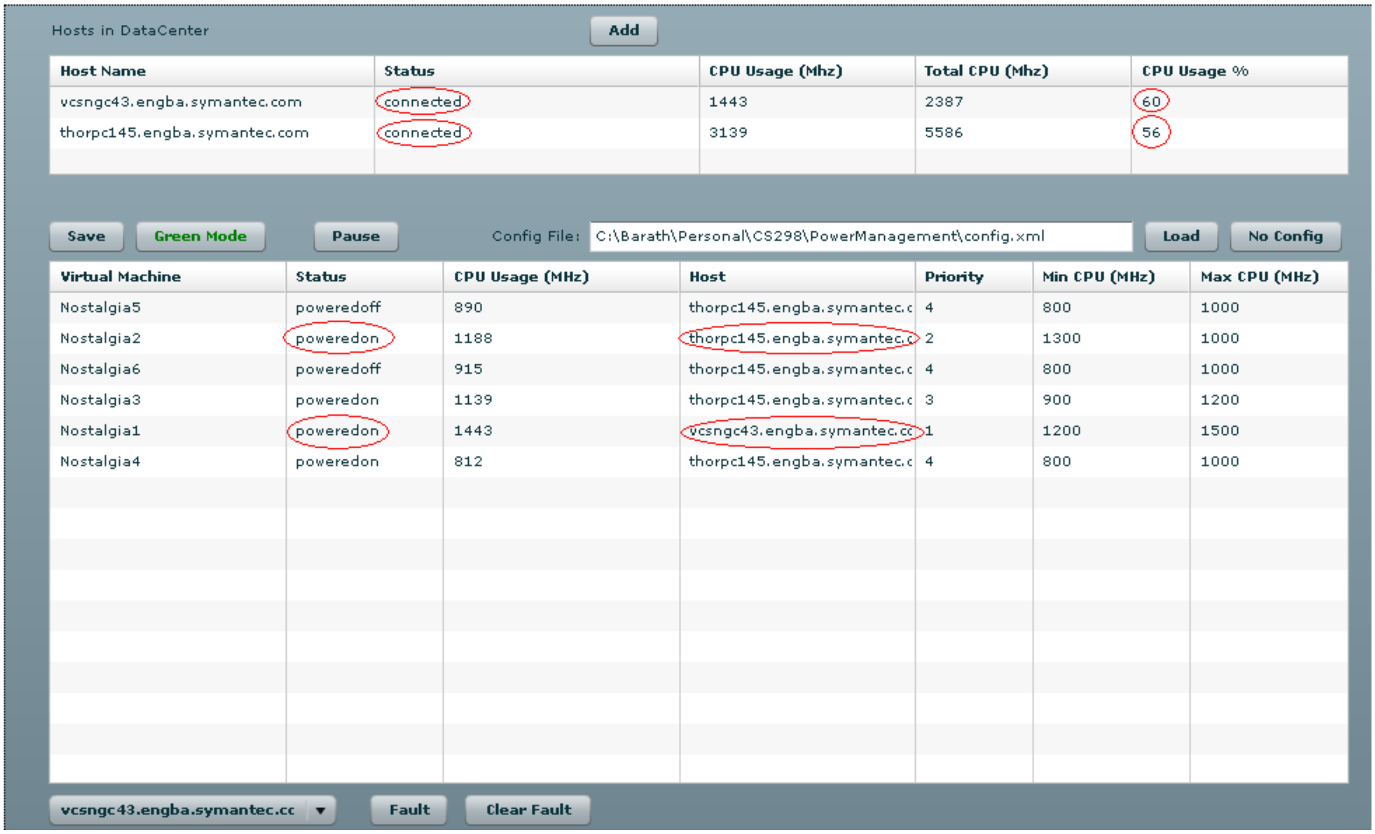
Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredoff	815	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredon	1033	vcsngc43.engba.symantec.cc	2	1300	1000
Nostalgia6	poweredoff	882	vcsngc43.engba.symantec.cc	4	800	1000
Nostalgia3	poweredoff	1085	vcsngc43.engba.symantec.cc	3	900	1200
Nostalgia1	poweredon	1279	vcsngc43.engba.symantec.cc	1	1200	1500
Nostalgia4	poweredoff	997	thorpc145.engba.symantec.c	4	800	1000

Figure 21: Critical Virtual Machines Relocated After Host Failure

From the above screen shot, we can conclude that my software has actually migrated the highest priority applications or virtual machines, which are Nostalgia1 and Nostalgia 2, from faulted system to the system which is running. These two virtual machines are in running state in new host. Hence, these two virtual machines show little or no down-time.

The host hosting the two mission critical virtual machines understandably is under heavy CPU utilization since it is hosting two mission critical virtual machines. Priority-Based Power Management is helpless at this time to bring down the CPU utilization of this host to 50 % since there is only one host in the Data Center and the virtual machines are marked critical. But, to show the strength of Priority-Based Power Management and Reduced Downtime system when the faulted host comes back online, I cleared the fault on the faulted system, and the screenshot below indicates what my new system will do when it sees faulted system coming back online.



Host Name	Status	CPU Usage (Mhz)	Total CPU (Mhz)	CPU Usage %
vcsngc43.engba.symantec.com	connected	1443	2387	60
thorpc145.engba.symantec.com	connected	3139	5586	56

Config File: C:\Barath\Personal\CS298\PowerManagement\config.xml

Virtual Machine	Status	CPU Usage (MHz)	Host	Priority	Min CPU (MHz)	Max CPU (MHz)
Nostalgia5	poweredoff	890	thorpc145.engba.symantec.c	4	800	1000
Nostalgia2	poweredon	1188	thorpc145.engba.symantec.c	2	1300	1000
Nostalgia6	poweredoff	915	thorpc145.engba.symantec.c	4	800	1000
Nostalgia3	poweredon	1139	thorpc145.engba.symantec.c	3	900	1200
Nostalgia1	poweredon	1443	vcsngc43.engba.symantec.cc	1	1200	1500
Nostalgia4	poweredon	812	thorpc145.engba.symantec.c	4	800	1000

Figure 22: Auto Load Balancing After Host Recovery

From the above screen shot, we can notice that Priority-Based Power Management and Reduced Downtime system has again re-distributed the load among the running hosts in such a manner that CPU utilization of both the hosts are now close to efficient operating CPU load of 50% recommended by Sharma and Lu in order to minimize energy consumption [7]. Thus, my software not only provides Priority-Based Power Management to minimize power consumption, but also reduced downtime to the virtual machines so that there is little or no down time for mission critical virtual machines present in data centers.

I began my experiments with the goal of determining whether my new Priority Based Power Management and Reduced Downtime software can help data centers to reduce their energy consumptions without compromising a guarantee that the mission critical applications have minimum downtime. The experiment results showed that the new software makes energy consumption in the data center to go down by as much as 35% by making all the machines in the data center operate close to 50 % CPU load range as recommended by Sharma and Lu to conserve energy usage [7] without compromising on the promise that the high priority virtual machines have minimum downtime. Furthermore, the results also show that the software also enables the administrators of data centers to set priorities for virtual machines and ensures that the critical virtual machines have minimum downtime. Hence, the new Priority Based Power Management and Reduced Downtime software not only conserves energy in the data centers, but also ensures minimum downtime for critical applications running in data centers. Hence, my software meets the overall goal of helping data centers reduce their energy consumption while ensuring that critical applications in data centers have maximum running time.

Conclusion

Data center contains many servers that run web applications, and some of these applications are mission critical. However, only few of the servers in a data center today are running critical applications while others servers are just sitting idle, consuming power and waiting to be used as backups in case of failure of servers hosting mission critical applications. As a result, lot of power is getting wasted.

Since Sun Microsystems predicts that the price of energy used by data centers is going to increase rapidly in future [5], it is logical that data centers implement some power management technique to save money and environment. At the same time, the data center should not give up on the core value of providing reduced downtime to mission critical applications by having backup servers. Although there are products out there in the market that provide load balancing technique to data centers to save power, those software doesn't address the issue of providing reduced downtime to mission critical applications. At the same time, there are products that provide reduced downtime to mission critical applications, but don't provide a load balancing technique to conserve power. These issues lead me to think that we need a hybrid model that combines the technique of load balancing with the reduced downtime technique that the data centers can use to save power and to provide reduced downtime to mission critical applications.

My Priority-Based Power Management and Reduced Downtime system uses this hybrid technique to combine the features of load balancing and reduced downtime for data centers. Based on the experiments, we can conclude that the new Priority-Based Power Management and Reduced Downtime system successfully conserves power for data centers without compromising a promise of reduced downtime for mission critical applications. Although this new system helps data centers by conserving energy and ensuring maximum running time for critical applications, the system has shortcomings. It still does not give the data center administrators the flexibility to see the power usage trend of all or any virtual machines in the data centers and to schedule energy-consumption based policies using a calendar.

Future Work

But, the new Priority-Based Power Management and Reduced Downtime system can be improved by incorporating a scheduling feature where the administrator can classify the priorities of individual virtual machines based on time. This feature will give further control to the administrator in terms of virtual machines and their varying importance based on certain time of the day.

Furthermore, features such as Trends and Recommendations that give administrator an additional picture of how virtual machines have been consuming power in the past and how to configure their priorities to get additional energy savings respectively will add merits to the Priority-Based Power Management and Reduced Downtime system.

References

[1] Bianchini and Rajamony. “Power and Energy Management for Server Systems.” Technical Report DCS-TR-528, Department of Computer Science, Rutgers University, June 2003

[2] David Filani, Jackson He, Sam Gao, Murali Rajappa, Anil Kumar, Pinkesh Shah, Ram Nagappan. "Dynamic Data Center Power Management: Trends, Issues, and Solutions." Intel Technology Journal. 12.1 (2008)

[3] M. Elnozahy, M. Kistler, and R. Rajamony. Energy efficient server clusters. In 2nd Workshop on Power Aware Computing Systems, February 2002.

[4] “Hypervisor.” Wikipedia: The free encyclopedia. 3 May 2009. Wikipedia. 3 May 2009. <http://en.wikipedia.org/wiki/Hypervisor>

[5] Nauman, Matt. "Energy costs for data centers forecast to leap 13-fold by 2012". The MercuryNews.com. 27 Jun. 2008. The Mercury News. 25 Aug. 2008 <http://www.mercurynews.com/greenenergy/ci_9716379?source=email>

[6] Bohrer, Elnozahy, Keller, Kistler, Lefurgy, McDowell, Rajamony. The case for power management in web servers. Power aware computing, Kluwer Academic Publishers, Norwell, MA, 2002

[7] Sharma, Lu, Thomas, Abdelzaher, Skadron. Power-aware QoS Management in Web Servers. Proceedings of the 24th IEEE International Real-Time Systems Symposium, p.63, December 03-05, 2003

[8] Stoess, Lang, Bellosa. Energy management for hypervisor-based virtual machines. 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference, p. 1-14, June 17-22, 2007, Santa Clara, California, USA

[9] “Download VMware SDKs & APIs.” 2009. VMware, Inc. 3 May 2009. <http://www.vmware.com/download/sdk/>

[10] “Virtual Appliance Marketplace” 2009. VMware, Inc. 3 May 2009. <http://www.vmware.com/appliances/directory/cat/508/>