Saving energy with XEN

Lotte-Sara Laan
University of Amsterdam
System and Network Engineering

February 5, 2007

February 5, 2007

Lotte-Sara Laan
University of Amsterdam, System and Network Engineering

Contents

February 5, 2007

Lotte-Sara Laan
University of Amsterdam, System and Network Engineering

1 Introduction

This project is part of the Master course System and Network Engineering of the University of Amsterdam. During this one year course, two research projects will be done. These projects will have a duration of four weeks and should result in a consultancy report.

The University of Amsterdam has its own research center with high performance computer clusters. These clusters are running CPU intensive programs and are operating in a grid. Since the computers are becoming more and more powerful, the amount of power they consume is growing as well. The load on these clusters varies and it is sometimes possible to put a part of the cluster into a sleep mode. Unfortunately, computers are still consuming power, even when they are turned off.

This project is born from the question if it is possible to save power on these clusters. Xen is a reasonably new project which makes it possible to run multiple operating systems on one machine. There are more projects which make this possible[1], but Xen runs as an OS itself using paravirtualisation to run the guest operating systems. This makes Xen perform better than for instance VMware, which does not have this feature. Another feature of Xen is live migration of virtual machines. This can be used to run multiple virtual machines on one system, but when more resources are required, a virtual machine can be migrated to its own system. Therefore this project will focus on using Xen for saving power.

1.1 Research Questions

The following questions are part of this research project:

- What are the different types of systems and (how) will their migration strategy differ?
- What is the amount of energy saved for a certain configuration, is it useful to implement this system in existing clusters?

This report will not go into details about Xen. More information about Xen can be found in reference [1], [2], [3].

[1] VMware, KVM, QEMU

2 Application types

There are several types of applications which may have an influence on the migration strategy to use. Therefore it is useful to sum up these different kind of applications and determine their requirements. This document focuses on the following application types:

- Memory Intensive
- CPU Intensive
- HDD Intensive
- Network Intensive
- Hardware Specific

Chapter 3 describes how these application types will reflect on the different migration strategies.

2.1 Memory Intensive

These are applications which require a large amount[2] of physical memory.

Examples of these kind of applications are:

- large in-memory databases
- graphical applications
- applications with a large dataset

There are two kinds of memory usage: allocate memory once to store data, and constantly altering the memory for processing. The second one could have a negative influence on the migration time. This will only be the case when using a dynamic migration type which is described in paragraph 3.3.

[2]For this project, a large amount of memory is defined as a constant usage of minimal 50% of the system's total physical memory

2.2 CPU Intensive

These applications require a large amount of CPU time. Running more than one of these on a machine will have a negative effect on performance.

Examples of these kind of applications are:

- applications for heavy calculations
- busy mail server with spam filter

2.3 HDD Intensive

These applications constantly read and or write to the harddisk.

Examples of these kind of applications are:

- file server
- newsgroup server
- mail server
- large database server

Some of these applications require high speed reading and/or writing to the harddisk. There are several factors which can influence this speed. A remote harddisk mounted over iSCSI could have a negative performance impact. A second factor when using remote harddisks, are the network resources you have. If you are using a 100Mbit network connection, you can only write 10MB a second. In this case you will loose about 40/50MB a second compared to a modern harddisk. If you choose local harddisks to avoid these performance issues, it is only possible to use the static migration method which is described in paragraph 3.2.

2.4 Network Intensive

These applications constantly use a large amount of network resources. Running more than one of these on a machine can have a negative effect on performance. Of course it is always possible to have multiple network devices on a machine and make each application run on its own device.

- file server
- newsgroup server
- mail server

These type of applications are also mostly harddisk intensive applications and, depending on the network device, can also be CPU intensive. When using the dynamic migration method, it should be considered that the node[3] will generate a lot of network traffic and will have an influence on performance of other applications which are using the same network device. These nodes can also have a negative influence on the migration time of nodes using the same network device.

2.5 Hardware Specific

These applications require special hardware to run on.

Examples of these kind of hardware are:

- Specialised Networking Hardware
- TV Card
- Special Graphics card
- Special clock source
- High speed network card

[3]Node: A Xen domU needed in the grid for processing jobs.

3 Migration Strategies

Doing Xen live migrations for power saving can be done in several ways. This chapter will be focused on two migration strategies: static migration, and dynamic migration. Each strategy will have its pros and cons. Sometimes it will not even be possible to use one of them in combination with an application type. It should be considered that there are also combinations possible between the different kinds of applications. As told in paragraph 2.4 for example, a network intensive application can be CPU intensive and hard-disk intensive as well. In that case, all characteristics of these kinds of applications will apply while choosing the right migration strategy. One of the requirements for using Xen live migration is having shared storage. More information about different kind of implementations for shared storage can be found in Appendix A. References about how to set this up can be found in reference number [4] and [5].

3.1 No Migration

First of all it is sensible to realise when it is not possible to use Xen live migration for power saving at all. This can be under the following circumstances:

- In a grid with applications which have a very short idle[4] time and differ a lot in load generation.
- When using machines with different architectures like 32bits vs 32bits PAE vs 64bits.
- When using machines with different CPU flags like SSE2 vs 3DNow.
- When for any reason whatsoever it is not possible to use a network storage for running the DomU root file systems.
- In a network setup other than a grid, where it is not needed to keep nodes running while they are idle. In this case it is easier to just shut down these machines without migrating the nodes first.

[4]Idle node: A Xen domU which is not currently needed for running jobs and has no active processes.

3.2 Static Migration

In this case migration will only be done when a node is idle in its processes. The migration will take place between a parking machine[5] and the other machines where the nodes should be running. The strategy is as follows:
The scheduler receives a new incoming job. The scheduler notifies the Migration Administration Tool (MAT) about this new job. When there are not enough resources on the current running nodes, the MAT will boot a sleeping machine[6]. The designated node is then migrated to this machine and the scheduler will be notified when the node is up and running. The job can now be started. When the job is done, the scheduler will notify the migration administration tool. If the node is not needed anymore it will be migrated back to its parking place. The machine can be turned off again. A grid setup for this type of migration is illustrated at figure 1

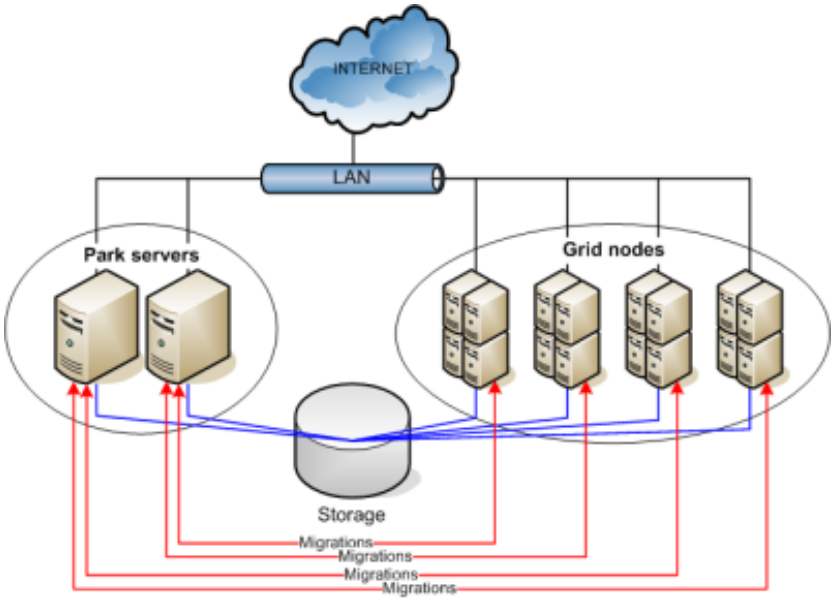

Figure 1: Grid setup for static migration

[5]Parking machine: A dedicated machine for parking nodes which are idle. This machine will always be running and is a perfect candidate for iSCSI server as well.
[6]Sleeping machine: A machine shutdown for energy saving, waiting until it is needed in the grid again

3.2.1 Tools

For this migration strategy, there are two tools needed to cooperate and manage the migrations. As described above, these are the scheduler and the MAT.

Scheduler:
A grid will already have a scheduler to distribute the incoming jobs over the available nodes. There are a lot of different type of schedulers [6]. But there are some extra requirements to use the scheduler in this setup. First of all, the scheduler should have notice of the MAT and should be able to communicate with it. The scheduler should also tell the MAT what resources are needed to run the job. The MAT needs to know this in order to decide which machine it should boot.

Migration Administration Tool:
The MAT will be a new tool created specially for the power saving purpose. As the name says, it is a tool to administrate the migrations. Therefore it should should be aware of all machines in the grid and their hardware specifications. It should be able to communicate with the scheduler to receive messages and report back to it when a machine is running and the node has been migrated. The best place to run this tool will be on one of the parking machines.

3.2.2 In Combination With Application Types

The real question is, how this migration strategy will work in combination with the several application types as described in chapter 2.

Memory Intensive Applications:
Migrating a node with applications which are memory intensive should be no problem at all since the migration will only take place when the node is idle. It is possible to dynamically add or remove the amount of assigned memory from a domU in Xen, using the balloon driver[7]. Therefore it is not necessary to have large amounts of memory in the parking machine for handling this kind of applications.

CPU Intensive Applications:
Static migration in combination with CPU intensive applications is not a problem either. If a node is migrated to the parking machine, it will be idle and not requiring much CPU time anymore.

HDD Intensive Applications:
There are several ways for using static migration in combination with hard-disk intensive application. It mostly depends on the type of applications. When a local hard-disk is needed, will the application still need this disk while it is idle? If this is not the case, then there will be no problem with static migration since the node will always return on its own machine when it is needed to perform again. If the disks can not be unmounted, even when the node is idle, then static migration will not be possible with local hard-disks. In that case, a remote storage solution will be necessary.

Network Intensive Applications:
Static migration in combination with network intensive applications will cause no problems at all. If a node is migrated to the parking machine, it will be idle and will no longer generate a lot of network traffic.

Hardware Specific Applications:
Migrating a node with application which depend on specific hardware could give issues if the application needs a constant interaction with this hardware, even when it is idle. When migrating a node using Xen, the handle for this hardware can break and should be initialised again after migration. If this is the case and the application does not recover from this, it is not possible to use Xen migration for power saving. Another thing that should be considered, when the application does recover, is if it needs the hardware on the machine where it spends its idle time. This could only be a problem in a grid where all nodes need the same special hardware, when the applications need the hardware dedicated to themselves. The parking machine will have multiple nodes running on it and, in that case, will need several of these pieces of hardware as well. In conclusion, static migration in combination with hardware specific applications will work when:

- The application does not need the hardware when the node is idle and the application will recover the handle with the hardware when running again.
- The application does need the hardware when idle, but it does not need it dedicated to itself and the handle will not break when migrating the node to a different machine.
- The application does need the hardware when idle, but it does not need it dedicated to itself and it will recover the handle with the hardware after migration.

3.3 Dynamic Migration

Dynamic migration can be useful in a grid with applications which sometimes require few resources and sometimes generate a high load. Migration will take place between all machines. When using dynamic migration, the strategy is as follows:
While a node uses many resources, it remains on its own machine. A monitoring tool will monitor resource usage and report this to the MAT. When a node starts to use less resources but still needs to be running, the node will be migrated to another low resource using node's machine. The machine can be shut down while the node can still do its job on its new host. When a node, running on a 'shared' machine starts using many resources again, the MAT will boot a sleeping machine. When the machine is up and running again, the node will be migrated to this one. A grid setup for this type of migration is illustrated at figure 2.

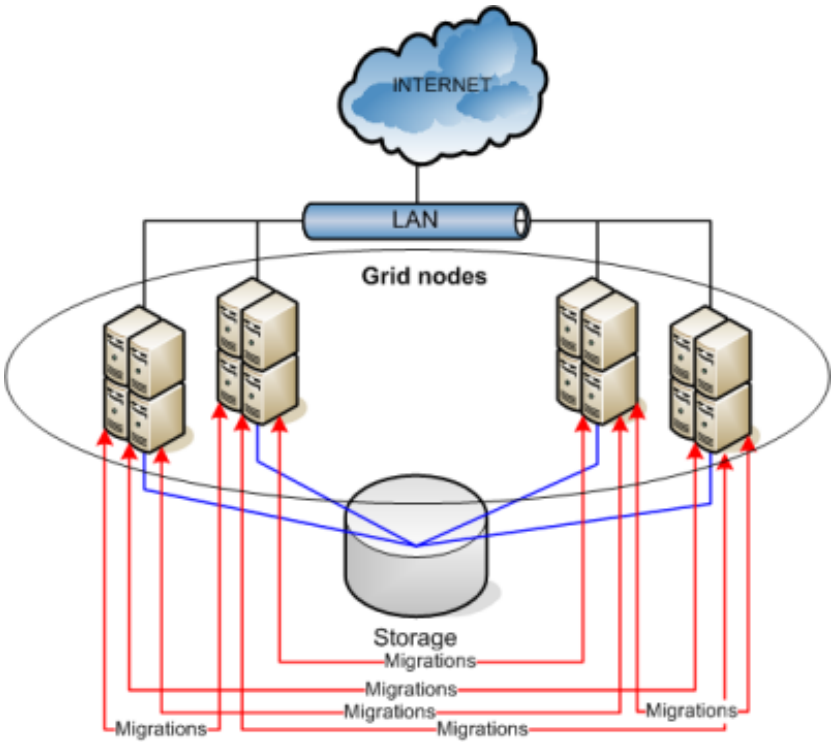

Figure 2: Grid setup for dynamic migration

3.3.1 Tools

For this migration strategy, there are three tools needed to cooperate and manage the migrations. As described above, these are the scheduler, a monitoring tool and the MAT.

Scheduler:
The scheduler has the same requirements as it has for static migration which is described in paragraph 3.2.1.

Monitoring Tool:
The monitoring tool should be installed on all nodes to monitor the usage of the system's resources. Since it will only be monitoring, it should have an interface for the MAT to get the required data for making migration decisions.

Migration Administration Tool:
The MAT will have the same requirements as in had for static migration which is described in paragraph 3.2.1. But next to these requirements, it should be a little smarter since in this case it should also know when a machine can be shared by running nodes. Therefore it should communicate with the monitoring tools of all nodes.

3.3.2 In Combination With Application Types

Memory Intensive Applications:
Memory intensive applications in combination with dynamic migration should have no problems at all. Though it should be considered that the migration time can take longer when the application is constantly allocating and freeing memory for processing. This is because Xen migration will be syncing the memory from the node to the new machine. When the memory data will change a lot during migration, it will take longer to synchronise it.

CPU Intensive Applications:
Just like static migration, this type of application should encounter no problems. But the MAT should know about the minimum amount of CPU time an application requires so it knows which machine can have multiple nodes running on it while they are not fully active.

HDD Intensive Applications:
hard-disk intensive application can be a bigger problem for this migration strategy. Since a node can be running on any machine in the grid, it will not be possible to use a local hard-disk. Therefore it will only work when the applications can work with remote storage.

Network Intensive Applications:
With a smart MAT, it should be no problem to use dynamic migration for this type of applications. In that case the MAT makes sure that the nodes will not interfere in their network usage by migrating them to the right nodes.

Hardware Specific Applications:
Migrating hardware specific applications with dynamic migration has the same problems as the static method which is described in paragraph 3.2.2. But there are some extra issues with this migration strategy. It still is possible to use dynamic migration with these kind of applications when the MAT is informed of the application's requirement of having specific dedicated hardware. In that case the MAT can dedicate machines that have this hardware for these applications to run on.

4 Measurements

Now we know when and how we want to set up our grid to use Xen live migration for energy saving, but how much energy will we save? And what effects will it have on our current grid? This chapter will show some formulas to calculate this.

4.1 Migration Times

The migration time of a node should be calculated from the moment the MAT decides to do a migration until the moment the node is fully running on the target machine. Factors which influence on this migration time are:

- The time it takes to boot/shutdown the target machine
- The amount of memory the node has (this indicates the amount of data which needs to be migrated over the network)
- The amount of memory changes which happen during the migration
- The type of network connection

The time it takes to boot or shutdown a machine depends on the machine's configuration. We have tested a machine with the following specifications:

- Manufacturer: IBM
- CPU: Dual PIII 1Ghz
- Memory: 1.5 GB
- 1 Network card 100Mbit
- 1 Network card 1Gbit
- 40GB SCSI disk
- OS: OpenSuse 10.2

This machine took 5,25 minutes to boot which is very slow and could probably be optimised a lot since it took 3 min to get through the PXE section. The operating system took 1,5 minutes. Note that disk checks can also slow down the booting process since these will be done after a certain number of disk mounts.

To calculate the time it takes from the moment you start the xen migration command until the command is done, the following formula can be used: memory / total of Mbit of network speed.

Migrating an idle node with 1GB of memory using a 100Mb network connection resulted in transferring 1081MB of outgoing traffic and 13MB of incoming traffic. Calculating the time this would take results in 86.5 seconds. In reality it took 93 seconds which means an overhead of 6.5 seconds.

Together with the 5.25 minutes of time it takes to boot a machine, it would in this case take a total of 6,75 minutes to do a live migration of a node.

4.2 Energy Savings

The following factors influence the amount of energy you can save using Xen migration:

- The percentage of idle time in your grid
- The number of machines in your grid
- The amount of energy a machine in your grid consumes
- The costs of 1kWh (the unit commonly used for measuring electric energy)

To calculate the savings, the following formula can be used: (total machines * idle percentage * Watt * hours) / 1000 * costs of 1kWh.

Example:

- 35 machines
- 70% idle
- 300 Watt per machine
- 8760 hours (1 year)
- EUR 0,11 per 1kWh

(35 * 0.7 * 300 * 8760) / 1000 * 0.11 = EUR7082.46 per year

Note that this formula will not tell you how much you save on cooling.

5 Conclusion

There are a lot of different kinds of applications which need to be considered when you are thinking about setting up a migration strategy for energy savings. Although this kind of setup can be very effective for a grid with a lot of idle time, we have not determined whether it is profitable for a more utilised grid.

We came up with two migration methods, the static and dynamic one, but more methods could be possible. The static migration method can be used in most cases, but when you want to use the dynamic one, you could run into more issues with certain application types. Dynamic migration also requires more advanced tools to make sure the migrations will take place at the right time.

Our measurements showed that it can be profitable to implement the Xen setup, but more experiments should be done to achieve more accurate results. The energy saving measurements have only been done for the static method.

A SAN Implementations

There are several indications that it is only possible doing Xen live migration using a Storage Area Network. This is because you only migrate the virtual memory of a machine. The following options are the most viable in this project:

- iSCSI
- NFS
- AoE

A.1 iSCSI

- "The filesystem should be journaling because the iscsi shutdown path can be ugly. You may need to hit the reset button to get a reboot to work. Thus you want a journaled FS." [8]
- For each node, a separate initrd is needed to make the disk available
- Does not allow shared access on targets
- More overhead than AoE since it uses TCP/IP
- Also overhead translating ATA to SCSI commands when using ATA drives

A.2 NFS

- Slow [9]
- SWAP is not possible over NFS and should be located at the host machine itself. This requires disabling the SWAP before migration.
- There is no need for a seperate block device for all domUs since you can share paths like /usr. This simplifies OS administration.

A.3 AoE

- Not routable
- Does not allow shared access on targets
- Overhead translating SCSI to ATA commands when using SCSI drives
- Need kernel patch to use as root device or initrd

References

[1] Wikipedia. Xen. http://en.wikipedia.org/wiki/Xen.

[2] Rami Rosen. Introduction to the xen virtual machine. http://www.linuxjournal.com/article/8540.

[3] The Xen Team. Xen interface manual. http://www.xensource.com/files/xen_interface.pdf.

[4] Gregory Cockburn. Xen live migration with iscsi. http://www.performancemagic.com/iscsi-xen-howto/.

[5] Paul Virijevich. Live migration of xen domains. http://www.linux.com/article.pl?sid=06/07/17/1916214.

[6] Wikipedia. Scheduling (computing). http://en.wikipedia.org/wiki/Scheduling_(computing).

[7] Paul T. Barham, Boris Dragovic, Keir Fraser, Steven Hand, Timothy L. Harris, and Alex Ho Rolf Neugebauer. The art of virtualization - paragraph 3.3.3. http://cs.uni-salzburg.at/~ck/teaching/CS-Seminar-Summer-2004/marcus-survey.pdf.

[8] Britt Bolen. iscsi-root mini-howto. http://www.eludicate.com/~bolen/iscsi/.

[9] Inc. TechnoMages. Performance comparison of iscsi and nfs ip storage protocols. http://www.technomagesinc.com/papers/ip_paper.html.