Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg


# VIRTUAL MACHINE LOGBOOK

DIPLOMA PROJECT – SUMMER-FALL 2008

---

## TASKS TO REALIZE

August 6, 2008


| | |
|---|---|
| STUDENTS: | ANDREA CAVALLI |
| | JULEN POFFET |
| SUPERVISORS: | FRÉDÉRIC BAPST |
| | PAOLO CALAFIURA |
| | OTTAR JOHNSEN |
| | YUSHU YAO |
| EXPERT: | PETER KROPF |

# SUMMARY

# INTRODUCTION

## VIRTUAL MACHINES

In computer science, a machine is composed by two main parts: the hardware and the software. The operating system (OS) directly communicates with the hardware, to offer to the applications an interface for access the physical resources.

A virtual machine (VM) is an abstraction of a whole computer: the software and also the hardware part. This means that a virtual machine contains applications, a guest operating system and a virtualized hardware. The guest OS and the applications execute like in a real machine. The difference is that instead to access to the physical hardware, the guest OS accesses to the virtualized one. It is the task of the virtual machine monitor (VMM, or hypervisor) to link the virtualized hardware to the real one[1].

There are several advantages of using virtual machines. The most important in this project are:

- a VM contains the entire environment of a machine: the operating system with the environment variables, the applications with their configuration and the user data;
- a VM which is in a given host machine can be serialized, moved and then executed in a different host machine.

A Just enough Operating System (JeOS) is a virtual machine which contains only the minimal components needed to run some specific applications in order to lighten as most as possible the size of the VMs.

## ATLAS EXPERIMENT

The Large Hadron Collider (LHC) at the CERN is a particle accelerator which will be used to run physics experiments. ATLAS is the name of one of the detectors in the LHC. During the experiments, it will generate hundreds GB of data to be treated and analyzed. ATLAS is also the name of the software used to run these analyses. It is composed by about 5M lines of codes organized in a thousand packages. This source code is used by the physicists to write ATLAS data treatment programs.

ATLAS software is in continuous development: at present hundreds of physicists run analyses on simulation data to tune up the algorithms, fix bugs and add missing features. Sharing the work among physicists and developers can be hard, because of the different working environments. To compile and run the same program on two machines with different configurations may not get the same behavior. Because of that, sometimes it is difficult to reproduce on different machines a crash or an analysis which produces an interesting result.

[1] The link may be directly or indirectly, i.e. by passing from the host operating system.

### CERNVM[2]

CernVM is a project which is still in development at the CERN. Its goal is to easily provide downloadable Just enough Operating Systems for the physicists working on the LHC experiments. The virtual machine available on the CernVM website is very light and can be personalized to the needs of the physicist. For example, it is possible to automatically install the components needed to run the ATLAS software.

The goal of using CernVM JeOS is to take the advantages offered by the VMs, allow the physicists to work with an official up-to-date release of the ATLAS software, and reduce the software maintenance cost.

## PROJECT DESCRIPTION

### CORE FEATURES

We have to develop the Virtual Machine Logbook, an application whose goal is to simplify the share of the work environments used in the ATLAS experiment. The key idea is to profit by the advantages offered by the virtual machines: if the physicists work into virtual machines instead of real machines, it is possible to easily share the work environments by sharing the virtual machines. In this project we suppose that the physicists work into virtual machines.

In this project we have to understand the needs of the physicists who work for the ATLAS experiment and develop a first working version of the Virtual Machine Logbook. This application will then be submitted to a team of testers that will provide a feedback on usability and real-time performance.

The first feature that Virtual Machine Logbook has to provide is the access to the logbook. The logbook is a repository which contains the environments used by the physicists. The environment is a virtual machine. There are three main features to access the Virtual Machine Logbook: add, commit and check out an entry (i.e., a virtual machine).

- The user who is working with a VM on a project can add it into the logbook by a simple command. The VM is then stored into the repository.
- During his work, the user can do other backups of his machine. All these backups are stored in the logbook. The logbook will then contain a history of the state of the virtual machine used by the physicist.
- If the user needs to restore an older state, he can retrieve the corresponding virtual machine from the logbook. After that, he will be able to work in the same environment which he was using when he did the commit of the VM.

The virtual machines to store in the logbook will have a big size. For this reason, the Virtual Machine Logbook will probably not always add and store whole virtual machines. We have to find a solution to lighten the size of the entries in the logbook. The Virtual Machine Logbook

[2] http://cernvm.cern.ch/cernvm

must also provide an easy way to download project-specific software from the CernVM repositories directly to a virtual machine.

The repository of the Virtual Machine Logbook can be placed in the local machine or in another remote machine. The physicist who needs to do backups of his work will probably store the VMs on his own machine. Using a remote logbook accessible by multiple users may be useful to share the virtual machines. A centralized Virtual Machine Logbook could be used as a central repository which contains the official ATLAS software releases.

The second feature that Virtual Machine Logbook has to provide is the deployment and management of the virtual machines on a remote server/cluster/cloud. The virtual machines stored in the logbook have to be easily deployed on one or more machines and then started. Then the physicist has to be able to run his analysis on these virtual machines. Once the job is terminated, the physicist has to use a Virtual Machine Logbook command to stop the virtual machines and release the resources which were used by them.

When a physicist is working inside a guest virtual machine, he probably needs to access some files stored in his host machine. We have to provide a simple way to share file system between physical hosts and guest virtual machines. This file sharing has to be provided for all the virtualization technologies used by CernVM. It must also be easily enabled or disabled.

## FURTHER WORK

If time allows, we will add some extensions to the Virtual Machine Logbook. There are some general ideas which could be studied and developed after the core featured described in the last section.

At this moment, the physicists work in their physical machines and not into virtual machines. Before starting work into a virtual machine, they need to move their environment data from the physical host machine to the guest VM. A useful feature would be the creation of a virtual machine from a physical machine. This VM should contain the same software environment of the physical machine, to allow the physicist to continue his work into the virtual machine, without any further configuration.

A virtual machine could have different network configurations. NAT (Network Address Translation) is a possible configuration which may complicate the access to mass storage systems. In particular, it is not possible to open a connection from the outside network. The virtual machines used by the physicists in the ATLAS experiment should be able to access all mass storage systems. It may be necessary to develop tools to allow that.

Data on the GRID should be accessible inside a virtual machine. We should investigate the technique to allow that.

At present, to retrieve data from the CernVM software repository, the user has to explicitly execute a command. It would be interesting to automatically synchronize the local disk cache with the content of the CernVM repository. This may be realized using a non-static file system.

## PREFERRED TOOLS AND BOUNDARY CONDITIONS

The Virtual Machine Logbook has to be compatible with the CernVM just enough operating systems. The virtual machines downloadable from the CernVM project web site are built using

rBuilder. This tool allows you to prepare virtual machines of all major VM platforms (QEMU/KVM, Parallels, Xen and VMware). The virtualization solutions should work on Linux hosts, but also on Windows and Mac OS X.

The applications for the ATLAS experiment have to be developed using open-source or at least freely available software products, because they are used by a large number of people of different universities and organizations.

## WORK TO DO

### STUDY

We have to study the different ATLAS software and understand how the physicists use them. We don't need to understand all the details of all the functionalities, but instead, we have to identify all the requirements of this software in order to make it able to run into a virtual machine (VM).

The physicist develops, tests and executes his analysis by working in an environment. The environment is composed by the operating system, the applications used by the physicist, the environment variables and the rest of the work area. We have to study how to make a "smart" snapshot of this working environment. Making a full snapshot each time will be a bad solution in term of disk space and speed.

ATLAS started an investigation on Virtual machine technologies, and it is currently collaborating with the CernVM project. Therefore we will use the tools supported by the CernVM project (QEMU/KVM, Parallels, XEN and VMware). We will study the drawback and advantages of these different virtualization tools.

We have also to study the tools that offer an infrastructure for the deployment and management of virtual machines and see in which measure we could use them in the Virtual Machine Logbook.

To summarize, these are the principal tasks we have to do for the first part:

- Familiarize with the ATLAS system.
    - Read documentation about ATLAS software. Do some tutorial to globally understand how the physicists use it.
    - Familiarize with the ATLAS/CERN grid, the CERN network structure and the ATLAS software repositories system.
    - Identify the different directories, environment variables and workspace area of ATLAS software.
    - Identify the other software needed by the physicists which is part of the environment.
- Study the virtualization tools.
    - Read documentation about the virtualization tools. Especially how to make a full, partial or incremental snapshot of the environment with these tools. Study the other possibility offered by their API in terms of reading or writing files

from/to the virtual machine. Also study the techniques to share files between the guest and the host machine.
    - Compare the different virtualization tools by focusing on the central functionalities needed by the Virtual Machine Logbook.
- Study the CernVM project.
    - Read documentation about CernVM and understand how CernVM already supports ATLAS software.
    - Understand how the CernVM virtual machine management interface works and how is done the file transfer from the CernVM repository to the VMs.
- Study the tools which could be useful to get and restore (some parts of) the environment, like file synchronization tools.
- Study the tools for developing the web user interface of the Virtual Machine Logbook.
    - Identify our needs for the web user interface and make a comparison between them.
- Read documentation about the tools which offer an infrastructure for the deployment and management of virtual machines (like Enomalism[3] and Globus Virtual Workspace[4]).
    - Identify our needs about the management tools and make a comparison between them.

After this part we will know the requirement of the different ATLAS software and we will have a general idea on the features offered by the different tools which will be used by the Virtual Machine Logbook.

## DESIGN

After studying the ATLAS software, the virtualizations tools and the VM deployment/ management tools, we will have enough information to do the design of the Virtual Machine Logbook application.

Firstly, we will specify all the modification that we will perform on the CernVM just enough operating system (JeOS) in order to build a virtual machine that contains all the ATLAS software. This virtual machine will be the base version that the physicists will use to do their analysis.

Secondly, we will clearly identify what a Virtual Machine Logbook entry will contains. Basically an entry is a full description of a working environment. The first entry added will be used as a reference by the next ones. We have also to find a way to describe only the difference between the given environment and the reference entry. For instance, a physicist who works on a specific analysis may add every day an entry on the logbook to keep a trace of his work. The first entry is the reference and the next ones will just contain the modifications done from this reference. This is the idea that we refer when we talk about "smart" snapshots. Instead of taking a picture of the whole environment at each time, we take one reference picture and then we pick only the difference between the reference one and the next ones.

[3] http://www.enomalism.com/
[4] http://workspace.globus.org/vm/

Then, we will specify all the basic functionality that will be offered by the Virtual Machine Logbook. In other word, we will think about how the physicist will add, commit and checkout his virtual machine to/from the Virtual Machine Logbook. The program must also offer the capability to delete a whole entry or a specific version of an entry. Note that the user has different places to launch a Virtual Machine Logbook command. He can do it directly into his virtual machine or he can also do it from his host machine. So we have to specify for each command if it can be executed only from the host machine, the virtual machine or from both.

Finally, we will describe the deployment features of the Virtual Machine Logbook, especially how the machine will be deployed on a remote server/cluster/cloud. For this point we will probably use an infrastructure and virtual machine management tool.

The detailed tasks we have to do for the design part are the following:

- Specify the modification we will apply to the CernVM JeOS.
  - …
- Specify all the basic functionalities of the Virtual Machine Logbook.
  - Describe the use cases of the application.
  - Draw the physical model of the application, i.e. the architecture in which the application will run.
  - Define which features will be available inside and outside the virtual machine.
  - Define which features will be available from the command line and from the web user interface.
  - Separate the features into components/modules and specify the interfaces between them. This will become the logical model.
  - Map the logical model to the physical model (i.e., where the modules will execute).
- Clearly specify the entries in the Virtual Machine Logbook.
  - Define what part of the environment has to be stored in an entry and what part must not.
  - Define how to create the reference entry (i.e., the first snapshot) of a virtual machine.
  - Identify the parts of the environment which have to be stored when the user does a commit (i.e., not the first snapshot).
  - Specify how the Virtual Machine Logbook can determine which parts of the environment are pertinent and have to be stored[5].
  - Define how the "smart" snapshot of the working environment is created.
  - Specify how the Virtual Machine Logbook will restore/check out the reference entry and the following snapshots.

[5] For instance, if from the last snapshot the user modified just the data in his working area, then the next "smart" snapshot should contain only this data. But if the user also installed a new application, the next "smart" snapshot should also contain the new application and maybe the new environment variables, the program configuration…
Automatically determine what parts of the environment should be included in the snapshot may be a solution, but maybe not trivial.

    - Specify how to manage the entries storage in a centralized/shared repository.
- Specify the accesses and the transfer from/to the Virtual Machine Logbook.
    - Describe how the entries are transferred from/to the Virtual Machine Logbook in both local and remote repository.
    - Describe how to manage the user right access to a centralized/shared repository.
    - Specify how to connect to the CernVM repository and download project-specific software from it.
    - Design a web user interface to manage the Virtual Machine Logbook.
- Specify how we will use the VMs deployment and management tool to allow the user to deploy virtual machines on a remote server/cluster/cloud.
    - *To develop…*
- During this phase of design, we have to choose:
    - a virtualization platform,
    - a web development tool and
    - a virtual machines deployment and management tool.
- Before to start with the realization phase, we have also to know what programming language(s) we will use. It may be necessarily to familiarize with the programming language(s) before to start the development of the Virtual Machine Manager.

**REALIZATION**

After the design, the Virtual Machine Logbook application will be separated into different components/modules. Each module will need and/or provides a service to other modules. So for this part, we have to implement these modules.
Here is a first sketch of all the components/modules we will implement during this phase:

- Realize the Virtual Machine Logbook repository management. The modules to develop have to implement the following tasks:
    - Prepare an entry.
    - Store the entry in the repository (locally and remotely).
    - Identify the parts of the environment to store in the entry.
    - Allow the user to add/exclude files or directory from a given list in order to know which files will be included into the snapshot (for instance, through a configuration file).
    - Package all the files before transferring them to the Virtual Machine Logbook in order to decrease the network overload.
    - Transfer the files from/to the Virtual Machine Logbook.
    - Remove a whole entry or just a version of the Virtual Machine Logbook.
    - Build a virtual machine from a Virtual Machine Logbook entry.
    - Check out an entry from the Virtual Machine Logbook to a local or remote machine.
- Realize the additional requirements of the application.
    - Management of the Virtual Machine Logbook from a web interface.

  - Management of the Virtual Machine Logbook user access.
  - Selection and download of the project-specific software from the CernVM repository.
  - File system sharing between the host machine and the guest VM.
- Realize the virtual machines deployment and management.
  - Deploy/ship a virtual machine to a remote server/cluster/cloud.
  - Manage the state of the virtual machine.
  - *To develop…*
- Prepare a test virtual machine.
- Realize and perform the test scenarios.
- Finalize the application.
  - Prepare a deployment package and an installation manual of the application.
  - Write a user manual for the application.

### TESTS

We have to prepare some test scenarios to verify if our application correctly runs. These tests will help us during the development to localize bugs and fix them. Each feature offered by the Virtual Machine Logbook has to be tested.

Some test scenarios will be conceived to verify if a single module/component of the application works as we expected, for instance if the entries are correctly structured and filled during the preparation of the entries. Other test scenarios will be realized to check the global behavior of the application, for example the integrity of the environment has to be guarantee after a commit and check out.

The test scenarios will be documented in order to reproducing them. We expect that we will not write a lot of automatic tests. We will write documents containing the instruction to manually execute the tests and the expected behavior of the application.

## ORGANIZATION OF THE PROJECT

The project is done at the Lawrence Berkeley National Laboratory (LBNL) for a period of 15 weeks. We will work together on the same project, but we have to separate it into two personal projects and we will be individually evaluated.

### KEY DATES

- July 27 – Beginning of the project
- November 7 – End of the project
- November 12 – Back to Switzerland
- November 14-15 – Public presentation at the EIA-FR

### WEB SITE

All the documents concerning the project will be posted on our web site:

http://phyweb.lbl.gov/atlaswiki/index.php/VMLogbook

login: guest-vml
password: vmlogbook

The web site contains two separate sections, one for each personal project.

## CONTACTS

### STUDENTS


- Andrea Cavalli cavalli86@gmail.com
- Julien Poffet julienpoffet@gmail.com


### SUPERVISORS


- Frédéric Bapst EIA-FR – frederic.bapst@hefr.ch
- Ottar Johnsen EIA-FR – ottar.johnsen@hefr.ch
- Paolo Calafiura LBNL – pcalafiura@lbl.gov
- Yushu Yao LBNL – yyao@lbl.gov


### EXPERT


- Peter Kropf UniNE – peter.kropf@unine.ch