Performance Tests of CMSSW on the CernVM

Marko Petek[1] and Stephen Gowdy[2]

[1] UERJ-Rio de Janeiro; [2] CERN-GE

E-mail: mpetek@cern.ch, gowdy@cern.ch

Abstract. The CERN Virtual Machine (CernVM) Software Appliance is a project developed in CERN with the goal of allowing the execution of the experiment's software on different operating systems in an easy way for the users. To achieve this it makes use of Virtual Machine images consisting of a JEOS (Just Enough Operating System) Linux image, bundled with CVMFS, a distributed file system for software. This image can then be run with a proper virtualizer on most of the platforms available. It also aggressively caches data on the local user's machine so that it can operate disconnected from the network.
CMS wanted to compare the performance of the CMS Software running in the virtualized environment with the same software running on a native Linux box.
To answer this wish, a series of tests were made on a controlled environment during 2010-2011. This work presents the results of those tests.

1. Introduction

This article describes tests that were performed in order to evaluate the feasibility of using Cern Virtual Machine (CernVM), a virtual appliance that allows execution of CMSSW in several platforms. CMS wanted to know if the overhead imposed by the virtualization technology could severely degrade execution time.

Several tests were conducted; their results and conclusions are exposed on this text.

2. Virtualization

Since the beginning of the micro-computers age there were different Operating Systems (OS) available for users. As a general rule, those OS were not compatible among them; to run software created in one OS on a different OS platform was an impossible task.

During the decade of 1980 some solutions were released in order to allow programs written for a given OS to be executed on machines running a different one. Some of those early examples include Simultask and Merge/386. However, as much as those systems were considered interesting, there was no momentum build towards their wider usage. A possible explanation was the dominance of MS-DOS as the “de facto” OS for micro-computers during those years.

Things changed with the arrival of Linux who took over an important percentage of the OS market share and lately with the revival of Apple, making their computers (and Mac OS) more and more known and reducing the dominance that Microsoft had before.

As technical important as Mac OS and Linux are, the fact is that they are also not interchangeable in many aspects. And both are also incompatible with Microsoft Windows standards.

Another important milestone was the creation of Grid Computing, driven by Ian Foster.

Grid Computing advocated the world-wide sharing of computing resources enabling users to run their software everywhere, at any time, accessing as much computational resources as they needed.

However an important problem wasn't solved by the initial Grid architectures and middleware. The heterogeneity of platforms remained an issue, as the machines and OS of the Grid infrastructure could be of any model and version. And those versions very often didn't match the needs of the software being executed.

On this scene, virtualization found a prosper environment to rise again as a feasible and affordable solution. Because by running a Virtual Machine over a host computer OS, a different OS may be executed (called a guest) inside this Virtual Machine. Therefore using a Virtual Machine makes possible, for example, to run a Windows guest environment inside the virtualizer, and having this virtualizer to run on a Linux or Mac OS host machine.

This led to a fast development of virtualization technologies that today have major players offering solutions to it. Among them may be cited Oracle Virtual Box, Cisco VMware, Apple Parallels and KVM for Linux.

It also led to research of new applications that could benefit from it.

In Cern, the PH-SFT group developed a project called CernVM, a "Portable Analysis Environment using Virtualization Technology" [1] [2]

According to them:

"CernVM is a baseline Virtual Software Appliance for the participants of CERN LHC experiments. The Appliance represents complete, portable and easy to configure user environment for developing and running LHC data analysis locally and on the Grid, independently of Operating System software and hardware platform (Linux, Windows, Mac OS).

The goal is to remove a need for the installation of the experiment software and to minimize the number of platforms (compiler-OS combinations) on which experiment software needs to be supported and tested." [1]

CernVM uses a JEOS (Just Enough Operational System) which is a binary compatible with Scientific Linux, however with only enough resources in order to make it work and run applications. A graphical desktop interface is available for users who demand it but is not essential.

CMS started to study the possible adoption of CernVM by its researchers. The main author of this paper received the task of supporting its use and to study the impact of its adoption.

In order to have a proper measure of the easiness of use and overhead caused by the Virtual Machine when compared to a native OS a series of tests were performed. The tests and their results are hereafter described.

3. Tests performed

In order to evaluate the overhead imposed by running CMSSW inside a Virtual Machine, it was decided to compare the running time of exactly the same script in a control computer both inside and outside the Virtual Machine.

To do this a dedicated control computer was configured. In order to avoid any kind of interference no other applications were running on this control computer, besides the ones needed to perform the tests.

The configuration of the environment was as follows:

- HP DC7900 with 4GBs of main memory and SLC5 as Operational System
- Oracle VirtualBox version 3.2.5
- CernVM version (JEOS) 2.1.0
- CMSSW 3_8_5

The maximum amount of memory allowed by VirtualBox was 2.5 GBs. In order to equate this with the LinuxBox environment, the size of memory used by SLC5 was limited to be also 2.5 GBs.

The scripts used to evaluate the performance of both environments were chosen from the list used to evaluate new releases of CMSSW (RelVal). Those scripts are well known and controlled.

Some of the scripts chosen had very fast times of execution, even when the number of events was increased. Others were very slow, demanding sometimes days of running in order to collect proper data.

After experimenting, some adequate scripts were selected to perform the tests.

Three runs were made for each script with 10, 30 and 50 events respectively.

One set of results was chosen to be presented in this work. It shows the same behavior observed throughout all tests performed.

New tests were conducted on the last months, now with scripts taking a very long time of execution to stress the system as much as possible.

Their results are not displayed in this paper because are still under evaluation, but it is possible to say that even for runs over 20 hours of execution, the overhead had values near to the ones presented in this work.

4. Tests results

The graphics below show the performance of one test script submitted to runs of 10, 30 and 50 events.

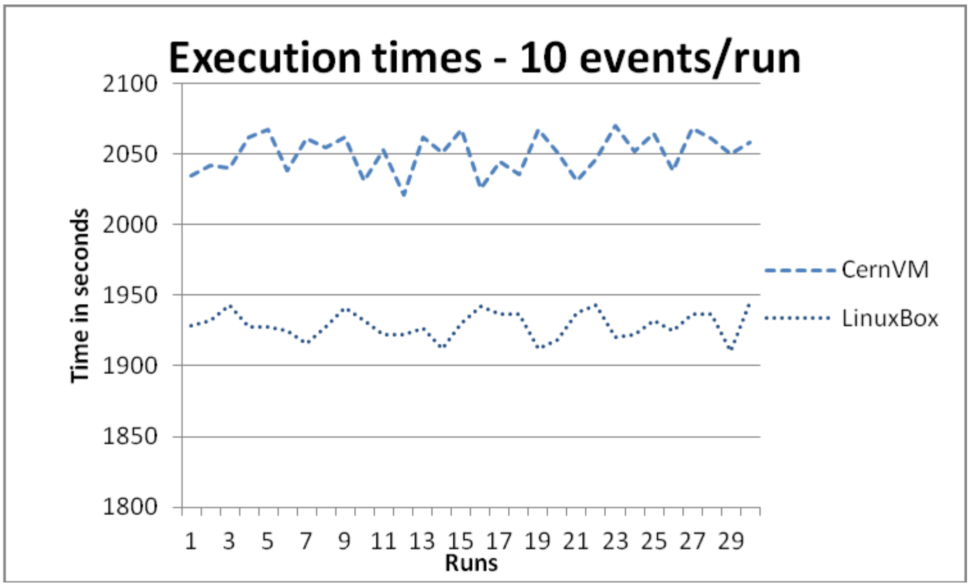

Figure 1 – Time of execution during 30 runs with 10 events processed in each one

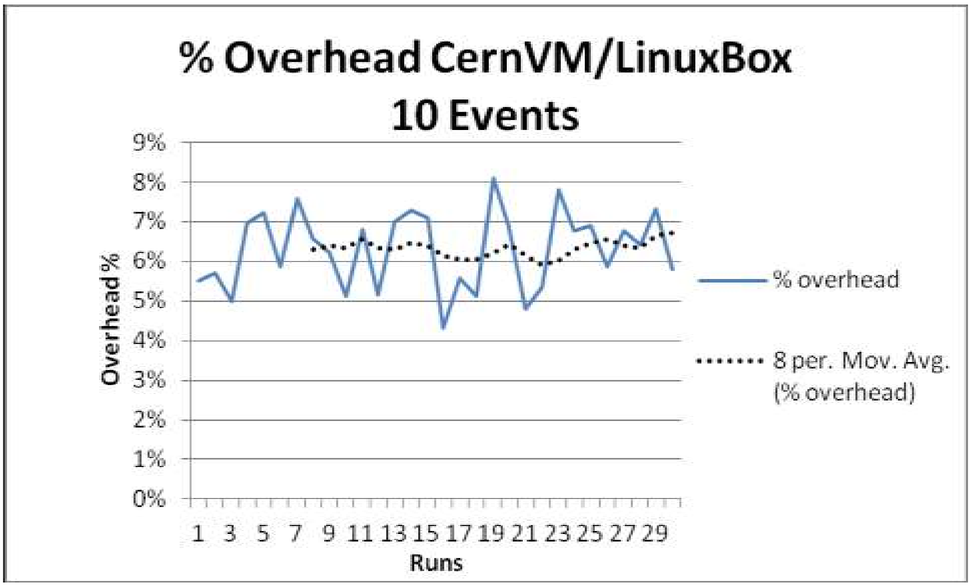

Figure 2 – Percentage of processing time overhead of CernVM compared with LinuxBox in runs with 10 Events

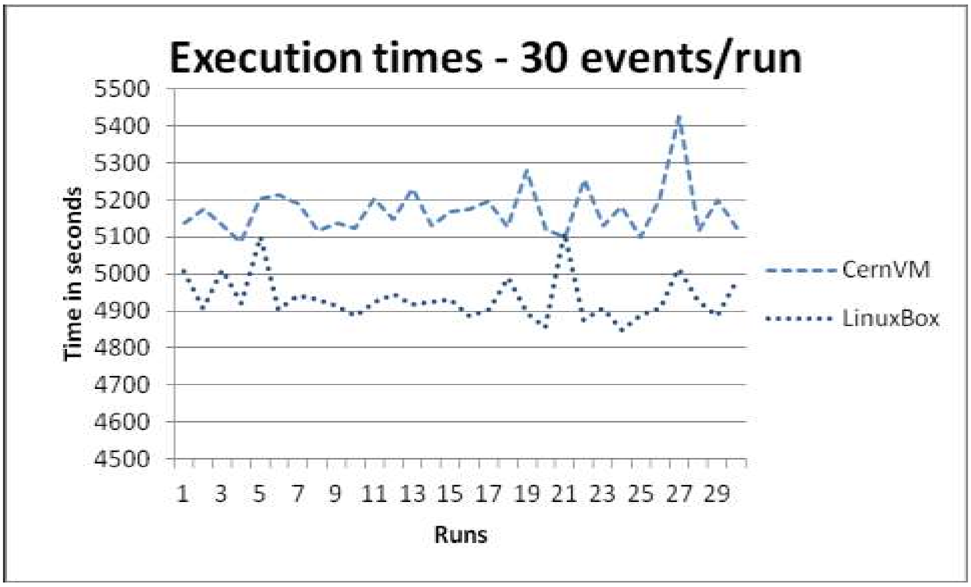

Figure 3 – Time of execution during 30 runs with 30 events processed in each one

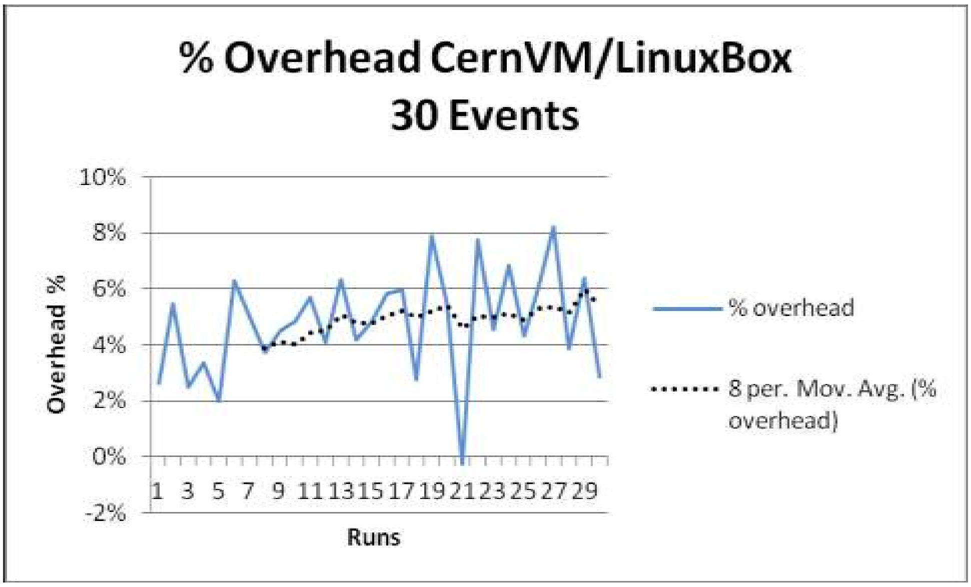

Figure 4 – Percentage of processing time overhead of CernVM compared with LinuxBox in runs with 30 Events

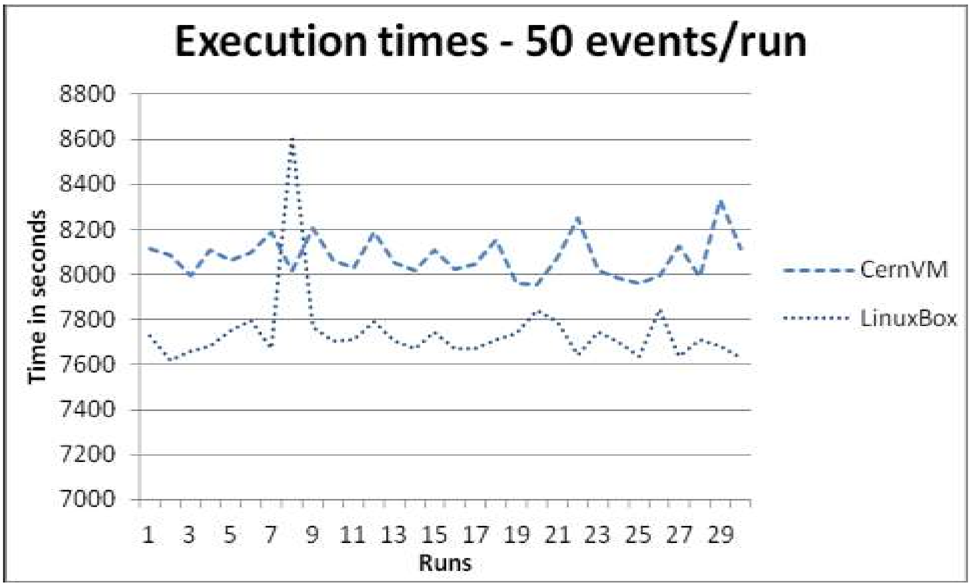

Figure 5 – Time of execution during 30 runs with 50 events processed in each one

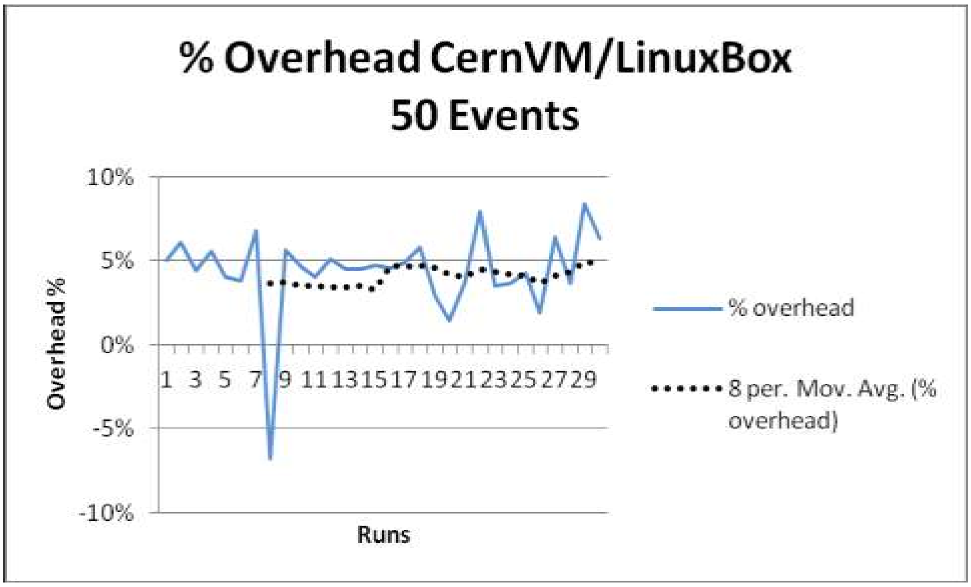

Figure 6 – Percentage of processing time overhead of CernVM compared with LinuxBox in runs with 50 Events

Based on the data represented by the above graphics, it is possible to see that the overhead due to the use of a Virtualizer is on the range of 4% to 6% when compared with a plain LinuxBox.

Tests performed in both Mac OS x and Microsoft Windows gave similar results.

Table 1 synthesizes the overhead according to the number of events analyzed. It is possible to see that the overhead goes down as the number of events increase. At this moment there is not a final explanation about the reasons for this, but preliminary tests point out that there is a start-up time in the virtualizer that remains basically constant every time a batch of runs is fired. So, for a larger number of events this amount of time represents a lower percentage than for a smaller number.

Number of events	10	30	50
Overhead	6%	5%	4%

Table 1 – Average percentage of overhead among the 3 tests

5. Conclusions

Based on the performed tests and their results, it is fair to say that the average overhead presented by the usage of a Virtualizer is small enough to be surpassed by the advantages offered by the technology.

With a CernVM running in its desktop or laptop, a researcher can have a private testing environment accessible from any place, even without an internet connection.

The Virtual Machine also allows an easier configuration of the CMSSW environment that can be up and running in just a few minutes through a fast setup procedure. It also allows the submission of jobs for the Computing Grid and has a built-in feature (CVMFS) that keeps the new CMSSW releases always available and updated.

References

[1]CernVM site: http://cernvm.cern.ch/portal/ Accessed in May 15th, 2012

[2] Predrag Buncic, Jakob Blomer, Pere Mato, Carlos Aguado Sanchez, Leandro Franco, and Steen Klemer. Cernvm - a virtual appliance for lhc applications. In Proceedings ACAT08, 2008