

# Exploiting Cloud Computing for enabling distributed testing of complex systems: the SELEX-SI roadmap

**Gabriella Carrozza, Ph.D.**

SESM s.c.a.r.l  
Via Circumvallazione esterna di  
Napoli, Giugliano in Campania, 80014,  
Naples, Italy  
[gcarrozza@sesm.it](mailto:gcarrozza@sesm.it)

**Massimo Loffreda**

SESM s.c.a.r.l  
Via Circumvallazione esterna di  
Napoli, Giugliano in Campania,  
80014, Naples, Italy  
[mloffreda@sesm.it](mailto:mloffreda@sesm.it)

**Vittorio Manetti, Ph.D.**

SESM s.c.a.r.l  
Via Circumvallazione esterna di  
Napoli, Giugliano in Campania,  
80014, Naples, Italy  
[vmanetti@sesm.it](mailto:vmanetti@sesm.it)

**Abstract** - *Complex systems are usually made up of several heterogeneous components glued together to get a System of System (SoS) demanding more and more effort in terms of integration, testing and maintenance due to the number of components, as well as to the several sources of failures that rise from heterogeneity. On the other hand, the strict reliability requirements of these systems ask for massive testing campaigns since they mostly fail due to software defects that can be either triggered systematically during system execution or manifest in a transient way during its operational phase. In the SELEX-SI scenario, where systems get developed and tested across different premises distributed all over Europe, performing traditional, manual, and on-site testing becomes dramatically expensive in terms of time and human resources. Cloud computing represents the most promising way for allowing the seamless access to distributed testbed from any site and for allowing remote testing activities, either at system and integration level. A cloud based infrastructure in charge of connecting all the company premises would allow to run testing experiments from anywhere and, more important, the possibility of reproducing distributed systems deployment scenarios to run integration testing in a pre-installation phase thus dramatically reducing company costs. This paper aims to describe the i) cloud research roadmap that SELEX-SI has been designing, ii) the architectural design of the cloud infrastructure and iii) the real ROI that the company expect from introducing such an innovation into the traditional software production process.*

**Keywords:** cloud computing, Open Source, testing.

## 1 Introduction

The need for massive testing campaigns in safety and mission critical systems comes from the strict dependability requirements and the catastrophic effects that any failures may have both on company business and human lives. However, performing exhaustive testing experiments, intended to test and assess the dependability of very complex SoSs, is not a trivial task due to the residual faults that usually manifest during system operational life and never before. These failures, indeed, are due to operational

conditions that are very hard to reproduce in pre-operational testing environment and sometimes manifest in a transient way despite the deterministic root cause, which is always a software defect, i.e., a bug into the source code. It can be definitely taken for granted that early testing is the best way to i) reduce the probability of operational failures, ii) improve system dependability and iii) reduce maintenance costs. Actually, discovering and fixing bugs before system installation represent a powerful mean for reducing costs, minimizing the number of people trips on site, and improving company's credibility towards customers that would be otherwise impacted by the greater number of system failures that can manifest during acceptance tests on the field. To actually get these benefits, companies are required to invest into the testing phases from a twofold perspective. First, testbed that mimic the real operational scenarios must be set up in order to reproduce system working condition as much as possible. Second, massive testing campaigns, aiming at checking both functional and nonfunctional requirements, must be performed despite of the great effort (in terms of infrastructures and human resources) that this may require.

This work aims at illustrating the SELEX-SI strategic roadmap that has been developing to get these challenges and that is mainly based on cloud computing investments. Actually, setting up an extended enterprise private cloud computing scenario, would allow the company to i) reproduce the real world scenarios, that usually encompass systems distributed over more operational premises, e.g., several ATC centers belonging to the same system and deployed over different cities in a given country, and ii) set up testbed platforms to perform distributed testing campaigns from different premises thus reducing people mobility costs. The paper discusses the technical and technological investigations that have been started to sustain and argument this intuition, providing preliminary results and illustrating the next future steps.

## 2 Rationale and motivations

Getting the great challenge of reducing costs and improving the dependability and quality of delivered

software systems, requires both methodological and technological investigations with the ultimate aim of developing a service oriented infrastructure to provide people working over different premises with remote facilities to perform testing and integration activities from anywhere and at any time. Different teams would be allowed working on the *same* platform, reducing lots of inefficiencies that actually hold mainly in terms of:

- Different platforms management overhead;
- Effort replication for installation and configuration in different premises;
- Need to move people from one site to another to work on different *physical* testbed;
- Static allocation of plenty of resources against the same system/task;
- Need to purchase as many hardware and software infrastructures as the testbed platforms;
- Maintenance costs increase due to the number of platforms;
- Need to train employers, with very specific skills, on any site to assure productive work and prompt response in case of problems.

## 2.1 EXPECTED BENEFITS

In order to minimize the process lacks discussed above, and the impact they have on productivity and costs, SELEX-SI has been investigating on both technological and architectural solutions, able to provide the following tangible benefits:

- Get the most of resources usage, to avoid they to get under loaded, aka “Too many servers for too little works!”;
- Maximize resource sharing in an heterogeneous environment, even enabling remote facilities (always keeping security and performance requirements in mind);
- Increase hardware and software support tools usability;
- Minimize inconsistencies among different testbeds;
- Minimize configuration and installation effort;

Through the activities and the investigation described in this work, the company does expect to identify the best way for getting all these benefits at a reasonable cost and in a medium term time horizon. Cloud computing paradigm looks the most promising solution to keep this goal. Furthermore, the availability of a plethora of Open Source platforms, would contribute to the overall costs reduction by keeping infrastructure setup, development and implementation costs much lower than any commercial solution.

The following section is devoted to provide basics on cloud computing, as well as to illustrate the technological roadmap followed by SELEX-SI in this direction.

## 3 Towards the cloud

SELEX-SI has been working and investigating in this field for years, passing through several technological options and methodological approaches that never revealed to be winning. As for example, dedicated automated testing environments have been setup over years for specific systems, to reduce costs or agile development and testing attempts have been made sometimes. We approached at cloud computing to define a one-for-all solution that can be used in every domain and applied to all the systems and solutions provided by SELEX-SI worldwide.

### 3.1 CLOUD COMPUTING BASICS

Cloud Computing (CC) is a paradigm having recent and growing popularity. Aim of this paradigm is to provide IT resources (such as computational resources, software components and storage resources) as services delivered through the network, hiding in such a way the sophistication of the underlying infrastructures.

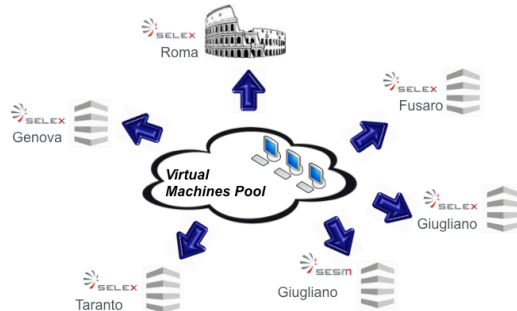
CC allows to manage resources in a *pay per use* way, and it guarantees the dynamic allocation of such resources against the current load requirements of the overall infrastructure (both in terms of users and operative load). As support for CC, the system-level virtualization techniques realize an abstraction of physical resources by multiplexing them in several virtual resources. The virtualization Hypervisor is a software layer having the goal to manage a number of Operating Systems on a single physical node, allowing in this way to optimize the use of the available resources. This is the basic principle to implement the so called *cluster consolidation* technique, which has its natural evolution in the CC concept.

### 3.2 SOLUTIONS FOR SYSTEM-LEVEL VIRTUALIZATION

First step in our technological roadmap has been the selection of a proper Open Source solution for system-level virtualization, being this mechanism the main brick to realize a private CC infrastructure. The results we obtained from the analysis and the comparison among solutions coming from a scouting campaign, led us to choose KVM (Kernel-based Virtual Machine), a full virtualization solution for Linux on x86 hardware containing virtualization extensions.

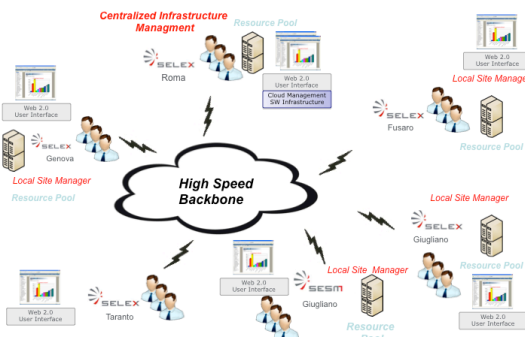
### 3.3 BUILDING THE CLOUD

Exploiting the aforementioned technologies, our final goal is to configure the corporate sites in Rome, Fusaro, Genova and Giugliano (both SELEX-SI and SESM lab) in order to build the cloud shown in Figure 1.



**Figure 1** - SELEX-SI Cloud, a logical view

The remote availability of hardware and software resources physically installed into the sites, is guaranteed by a dedicated network infrastructure configured as single backbone among the sites, and by Web interfaces providing entry points for the available services. The management of the overall CC infrastructure is in charge of the technical staff working at the Rome site, while the management of each single site platform is in charge of the technical staff working at such site (see Figure 2).



**Figure 2** – SELEX-SI Cloud, physical view and access channels to the resources

In order to observe cost and performance requirements, combinations of enterprise and Open Source solutions have to be taken into account, both concerning the site level and the infrastructure level. More in detail, the following alternatives may be considered:

- Site Level (SL)

- SL1. Proprietary SW: proprietary solutions for both the virtualization system and the CC platform
- SL2. Hybrid: Open Source virtualization system and proprietary CC platform, or vice versa
- SL3. Open Source: Open Source solutions for both the virtualization system and the CC platform

- Corporate Level (CL)

- CL1. Proprietary SW: the overall infrastructure is based on proprietary solutions
- CL2. Hybrid: the SELEX-SI sites are managed by proprietary solutions, while the SESM Lab site is managed by Open Source solutions
- CL3. Open Source: the whole corporate infrastructure is based on Open Source solutions

The final decision, will be driven by the system application requirements as well as from company management indications in this direction. However, (SL3, CL3) is the final point we wish to reach, even if intermediate “mixed” alternatives will be applied for a while to allow the complete transition from legacy environments to the cloud.

## 4 Experimental campaigns

The experimental session we performed has the goal to verify the robustness of the virtualization hypervisor we selected. In order to reach this goal we chose a reference class of applications, namely applications for automatic testing of software components in the Air Traffic Control and the Surveillance of Battlefields domains. Later in this section we refer such kind of applications as V&V (Verification and Validation) applications.

### 4.1 AIMS AND ORGANIZATION

The approach we devised and implemented consists in using virtual machines as execution environments for V&V applications, and in generating virtual machines to increase the computational load for physical machines on which the KVM Hypervisor is running. In the following we refer the virtual machines belonging to the first class as V&V VMs, and those belonging to the second class as Load VMs.

The virtual cluster we realized is then composed by both kind of VMs, while the whole testbed is accordingly made-up of a number of physical hosts and a number of virtual hosts.

We aim at determining and evaluating the behaviour of each element of such testbed. More in detail, we need to assess the following conditions:

1. No problems reported in the behaviour of physical machines running the KVM Hypervisor while increasing number and

- computational load of the hosted virtual machines;
- 2. No differences found in the behaviour of Virtual Machines running different Operating Systems;
- 3. No hangs and no crashes manifested on both virtual and real machines;
- 4. No problems reported during the execution of V&V applications on Virtual Machines.

The following section describes the used tools and the overall organization of the testbed.

## 4.2 THE TESTBED

The testbed is made-up of physical machines hosting the KVM Hypervisor, and a virtual cluster composed by VMs having different purposes. These VMs present also different configurations in terms of Operating System (we used Windows XP, Windows Server 2003, and several Linux distributions), RAM, CPU slot, Disk.

The physical hardware we exploited consists in 8 Dell Power Edge M600 blade with the following configuration:

- CPU: 2 \* QUAD-CORE Intel Xeon E5420, 2.5 GHz
- RAM: 32GB
- Network: 4\* Gb Ethernet NIC
- Operating System : CentOS 6 64-bit

In the following we refer to such nodes with the label *CentOS-KVM*.

As stated above, the reference class of software we selected, consists of applications for automatic testing of software components belonging to the Air Traffic Control field and the Surveillance of Battlefields field. In particular, we chose as V&V applications for our experiments two frameworks for the automatic testing of the two systems respectively<sup>1</sup>:

- YYY
- XXX

Both frameworks present several components; while the former requires a single VM (Win2003\_YYY), each component of the latter runs on a single VM; in the following the structure of the XXX automatic testing framework:

- XXX-ClientTST02 (Windows XP)
- XXX-GIS-TST02 (Windows Server 2003)
- XXX-CS-TST02 (Windows Server 2003)

Regarding the Load VMs, instead, in order to stress the virtual devices they are equipped with, we selected and exploited ad hoc applications; these last are

classified per Operating System and listed in the following :

Windows platform:

- MemAlloc to stress the Memory
- Core Damage to stress the CPU
- DITG to stress the Network

Linux platform:

- The *stress* Linux command to stress both the memory and the CPU
- The *iperf* traffic generator to stress the network

Since we aim to determine and evaluate the behaviour of both the physical nodes on which the KVM Hypervisor is running, and the VMs composing our virtual cluster, one important task is the monitoring of the overall testbed.

In order to reach this goal we chose the Nagios infrastructure monitoring system. We installed and properly configured the server side on a VM, and we created a number of plugins by modifying some scripts available on the Nagios Exchange database, to monitor the remote Linux hosts. Concerning the Windows VMs, we used the NSClient application and the Windows performance counters.

A list of the activities we monitor with Nagios is listed below :

- CPU Load
- Disk Status
- Memory Usage
- Network Transfers

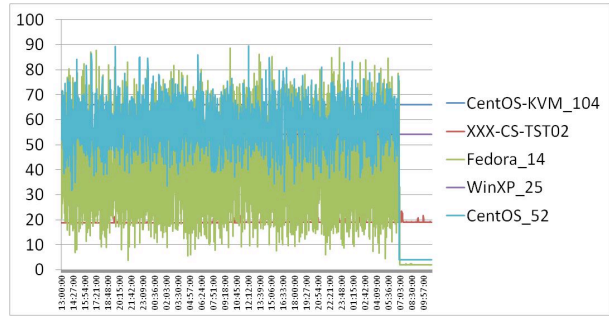
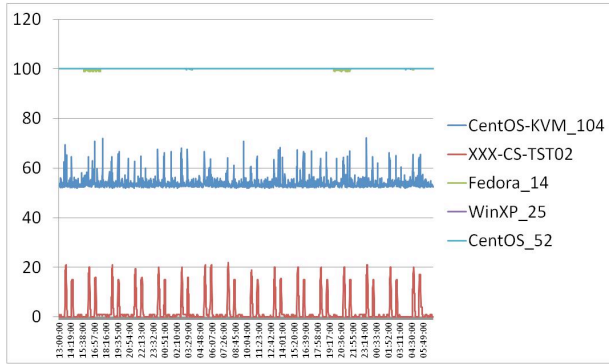
The schema we implemented is then enriched by the Convirture Enterprise-class management system, and an OpenFiler iSCSI SAN. The Convirture management system is equipped with a very friendly web interface accomplishing the organization and the management of VMs composing the virtual cluster. Convirture is itself running on a dedicated VM. Concerning the SAN, OpenFiler is an Open Source storage management appliance that we use to manage a 12 Terabyte storage space RAID-10 configured.

## 5 Preliminary results

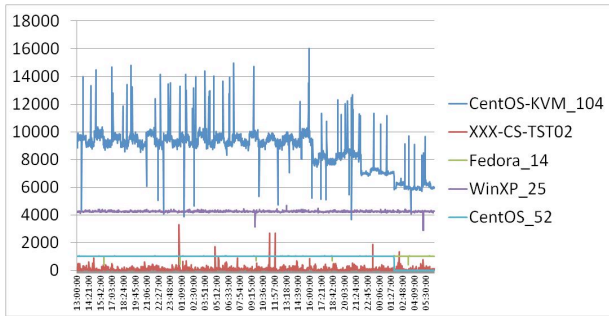
The experimental campaign has been splitted in two phases. The former aims to verify the first three points listed in section 5.1; the latter to demonstrate the last one. In the first phase, the resources composing the overall testbed have been used, both virtual and real. More in details:

- 8 physical nodes (one of them hosts management and monitoring applications, the other host VMs);
- 3 V&V VMs composing the XXX testbed;
- 1 V&V VM composing the YYY testbed;
- 18 Load VMs; each of them is characterized by a particular configuration.

<sup>1</sup> System names are omitted for the sake of information confidentiality



b) Memory Usage



c) Network Transfer

**Figure 3 – First phase**

The duration of such experiment has been fixed in 48 hours.

The data we collected during the experiment is related to the main metrics we took into account, namely CPU Load, Memory Usage, Network Transfer. In Figure 3 we present for each metric the data related to one physical node and the relative hosted VMs. The first two metrics express a percentage, while the last metric is in Kbps. Time on the x axe.

The graphs in the figure show that the behaviour of the physical nodes meets our needs in terms of scalability, robustness, and support of different Operating Systems. Values related to such nodes never overcome properly selected thresholds, and this demonstrates that the nodes work fine also when the computational load on the hosted VMs (and then on the physical node itself) increases.

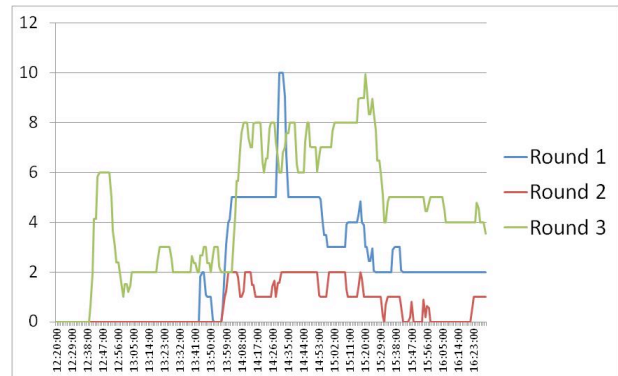
Concerning the VMs behaviour, no hangs and no crashes occurred on both Linux and Windows platforms, even though a heavy computational load stressed the core components of such machines.

The second phase is in turn splitted on three rounds having each one a duration of 4 hours. Only three physical nodes have been used to perform such second phase, except for the node hosting the management and monitoring applications.

What changes among the three rounds is the number of VMs hosted on each physical node. The main goal of this experiment is to verify that the V&V applications (hosted themselves on VMs) do not suffer the increasing computational load (consisting of VMs) on the physical nodes.

In Figure 4 we present for each metric the data related to one V&V VM (Win2003\_YYY); each graph presents a comparison among the lines related to each round of the experiment. The first metric (CPU Load) express a percentage, the second one (Memory usage) is in MB, while the last metric (Network Transfer) is in Kbps. Time on the x axe in this case too.

The graphs show that the behaviour of the V&V VM under test (Win2003\_YYY) is the same in each of the implemented rounds. This is a demonstration of what we need to verify: the increasing load on the physical nodes does not impact the performance of the applications running on the hosted VMs.



a) CPU Load





c) Network Transfer

Figure 4 – Second phase

## 6 Lessons learnt and future work

The need for improving the efficiency of software V&V process, in terms of costs and product quality, has been driving the investigation and the experimental campaigns described in this paper. Although the work is at a very early stage, some interesting considerations emerged from this first bundle of experiments. First, they confirmed the intuition that virtualization is the best way to get the most of your hardware infrastructure if you set it up properly. However, what made we happy with results is the good behavior we get from an Open Source hypervisor, namely KVM. Second, the platforms we are going to select are ready to support the kind of load that is typical of SELEX-SI application scenarios, without any impact on performances and improving the efficiency of real world testing campaigns. This was the *condicio sine qua non* we wanted to assess before going towards CC platform setup that, actually, represents the second part of the story. What we plan to do actually in the next future is to:

1. Enrich experimental campaigns on different applicative workload, even more demanding in

terms of number of test procedures and computational load.

2. Run the same experiments described here, and the ones we are going to do more, on top of a commercial hypervisor to compare the performance results;
3. Start a CC platform scouting, on top of KVM, to select the one in charge of meeting the requirements of both the hardware we have and the applications we must address. Actually this is a task already running and interesting feedback are coming from OS CC environment like OpenNebula, OpenStack and Cloud Stack;
4. Start performing distributed campaigns on at least to different SELEX-SI premises to demonstrate the real effectiveness of the approach.

To conclude, the results gained and illustrated in this work show that KVM is a valuable Open Source alternative to build up a CC environment against both the performance and application requirements exposed by SELEX-SI scenarios. This paves the way to the actual development of a CC platform aimed at supporting the company V&V processes and optimizing costs.

## References

- [1] P.M. Chen, B.D. Noble. When Virtual is Better Than Real. Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems. Schloss Elmau. Germany. May 2001.
- [2] M. Armbrust, A. Fox, et. al. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS-2009-28. University of California at Berkeley, USA. Feb. 10, 2009.
- [3] Q. Zhang, L. Cheng and R. Boutaba. Cloud Computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, Volume 1. Number 1, 7-18. Springer 2010.
- [4] R. Canonico, P. Di Gennaro, V. Manetti, and G. Ventre. Virtualization techniques in network emulation systems, volume 4854/2008, Euro-Par 2007 Workshops: Parallel Processing of Lecture Notes in Computer Science, pages 144-153. Springer Berlin / Heidelberg, 2008. ISSN:0302-9743, ISBN:978-3-540-78472-2
- [5] A. Shah. Kernel-based virtualization with kvm. Linux Magazine, vol. 86, pp. 37{39, 2008.
- [6] A. Botta, A. Dainotti, A. Pescapè. A tool for the generation of realistic network workload for emerging networking scenarios. Computer Networks (Elsevier), 2012.
- [7] W. Barth. Nagios, System and Network Monitoring. No Starch Press, u.s (2006)
- [8] B. Childers. Openfiler: an Open- Source Network Storage Appliance. Linux Journal 2009. ACM Digital Library. Retrieved on June 22, 2009 from <http://www.linuxjournal.com/article/10414>. appliance. Linux J., 2009(181):6, 2009.
- [9] <http://www.soft.tahionic.com/download-memalloc/index.html>
- [10] <http://damage.vigilantesoftware.com/http://www.nsclient.org/>