Intelligent Agents in Knowledge Acquisition and Structuring for the Fault Diagnosis of Virtualized Systems

Florin Postolache
Danubius University of Galati
florinpostolache@univ-danubius.ro

Viorel Ariton
Danubius University of Galati
variton@univ-danubius.ro

Florentina Loredana Tache
Dunarea de Jos" University of Galati,
florentina.tache@profinfo.edu.ro

Cătălin Năchilă
„Petrol – Gaze" University of Ploieşti,
katonec@yahoo.com

Alin Constantin Filip
Danubius University of Galati
flp_alyn33@yahoo.com

Abstract: The knowledge acquisition concerning the behavior at the fault of the complex systems is a systematic process, first, by the presentation of the processes, procedures and stages that occur throughout of a acquisition project of knowledge. Also, a good knowledge of the system, with all its features, is a good decisive factor concerning the successful realization of the knowledge acquisition. In addition, for the fault diagnosis it is required the knowledge and the knowledge acquisition for the fault behavior (anomaly/symptoms and manifestations, granularity of the defects, relations between them in various operating environments).

Keywords: intelligent agents; knowledge acquisitions; virtual machine; platform virtualization; application virtualization

Jel Classification: D80; D83

1. Introduction

Lately, the development of the information technology has opened new ways of structuring the hardware infrastructure, of the applications and thus by making operations. Thus, recently, the infrastructure of some organizations has become partially or fully virtualized, and because of this, the processes have become a dynamic and distributed character, and the static and hierarchical structures have become even more adaptable and flexible.

Thus, the existing tools and utilities provide a specific functionality to the management systems that are in the system, but they cannot know and detail in timely manner the way in which the various IT infrastructure components behave and interact. Although they are often difficult to use or not reliable, sometimes it is necessary to implement some more sophisticated management tools involving the placement of staff throughout their IT infrastructure having as purpose to collect the necessary information of the management systems to offer a efficient, effective and of control of the IT services. Some are skeptical about the implementation of the agents because they may affect the performance of the devices or could overload their network and starting from here, they conclude that that all of their IT infrastructure could destabilize.

2. Intelligent Agents

For better monitoring and managing of the virtualized system as and for a better distribution of tasks and competences of the agents, we consider necessary a division on layers of the system subject to the study case. Consequently we have the following layers: physical (FL), virtualization (VL), virtual machine (VM), operating system (OS) and applications (Apps), layers exemplified in Figure 1.

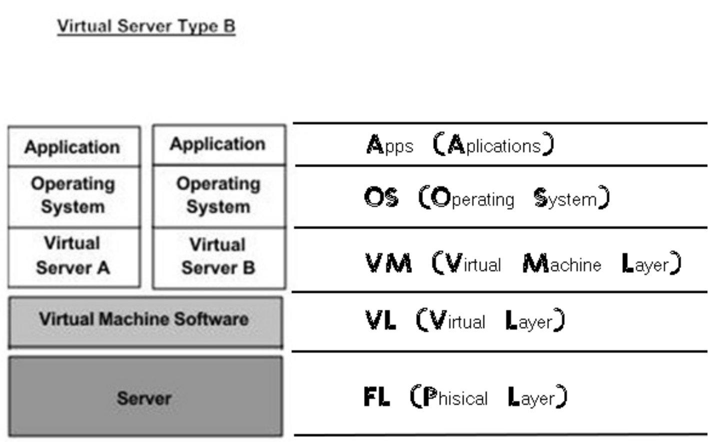


Figure 1. Computer system layers

Thus, the monitoring, administration and acquisition of knowledge for the fault diagnosis from within the system will be made with the help of the intelligent agents based on their location on the layer, with the mention that the layers VL, VM, but and OS and Apps will be monitored together, as being considered only one layer which we will name in the following VL&VM and OS & Apps. In addition, the storage space is considered as a separate layer which also requires a separate monitoring and managing. In other words, the distribution of the agents will be made on each layer, thus being specialized and having the level of competence only on the layer were they are placed.

Initially, within the system, the intelligent agents are placed only on the appropriate layer having implicit the level of competence corresponding to their placement within the system. Also, they have the monitoring and surveillance role of the followed characteristics and of insertion in the data base of the found events (cases). After the insertion into the data base of the events (cases) that lead to a functioning which is not within normal parameters (defective) or even lead to a possible failure of the components (is not referred only to the physical ones), the human agent plays the most important role, where it decides what operation or action should be taken.

Following the operations taken by the human agent for each case found in the database, is recorded and the action taken or if there were more identical cases, the action that produced the best results. Here, if is considered necessary that for the same fault to be performed the action (operation) which produced the best results, the responsibility can be transferred to the intelligent agent. In this case, at the occurrence of the same event, the intelligent agent inserts a new record in the database, and, following the granting of confidence by the human govern, searched in the database the action that must be taken, according to an event after it executes it, inserting of course and the action taken by him. After this, the human agent notified of the operations performed by the intelligent agent and thus he can determine whether it was acted correctly and after the consultation of the identical or similar cases can decide whether the further action that the agent will take to remain the last decision considered the best or must be changed with a new one.

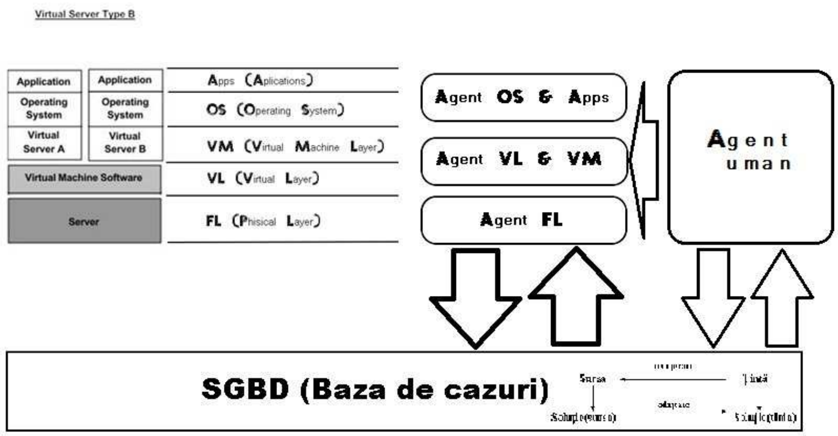


Figure 2.Distribution and role of agents in the acquisition of knowledge

Worthy to note is that if at the first the intelligent agents cannot take actions, along the way they earn a degree of freedom. Of course that to an intelligent agent can be granted the right to take action by themselves in the case of repeating of the same fault by a certain number of times, without the human intervention, but first we decided that it is more prudent that the human agent to decide when the intelligent agent can take automatically actions.

It must be specified that the intelligent agents have initially, as it was noted above, only the supervisory role, and of insertion in the database of the information (error message) in case of an operation that does not fall within the normal parameters or in the case of a fault. Thus, they are involved in observing the functionality of the physical resources and of the abstracted ones, checking the status of IP connectivity of the physical and virtual machine, reading and loading the resources but and consulting the loading logs.

Further we describe the role and functions of the intelligent agents according to the layer were they are placed.

Physical Level (FL)

The physical level consists of all the hardware equipments that enter in the component of the computer system. Here we have the equipments from the network, equipments from the composition of the center of data, the physical servers and the storage environments (SAN).

FL Coupling at Network

Within this level the human agent has the largest share in observing and intervening on the physical components from the system component. On this level most of the operations are performed by humans and therefore in the knowledge acquisition and structuring, the human expert occupies the principal place.

The intelligent agents are involved on this layer in the observation of the physical resources functionality, checking the state of the IP connectivity of the physical machine, reading and displaying the resources and but consulting the loading logs to insert the errors in the database.

The IP connectivity we will do with the help of the ping network tool that is used to verify if a certain computer can be accessed through some network of IP type. In this case we used free software called Server Ping, its role being pretty small, resuming just to a two color display of the response status ICMP "echo response" from the destination host after some request "echo request".

Resources Monitoring FL

Also, the agents besides the above mentioned role have and the task to read and display the situation of loading the physical resources as is seen from the figures below.

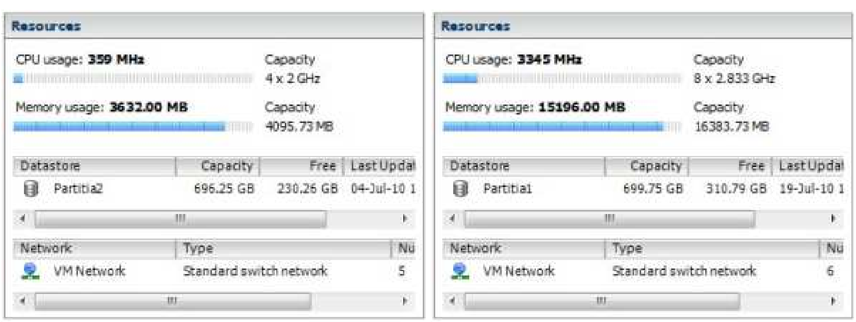


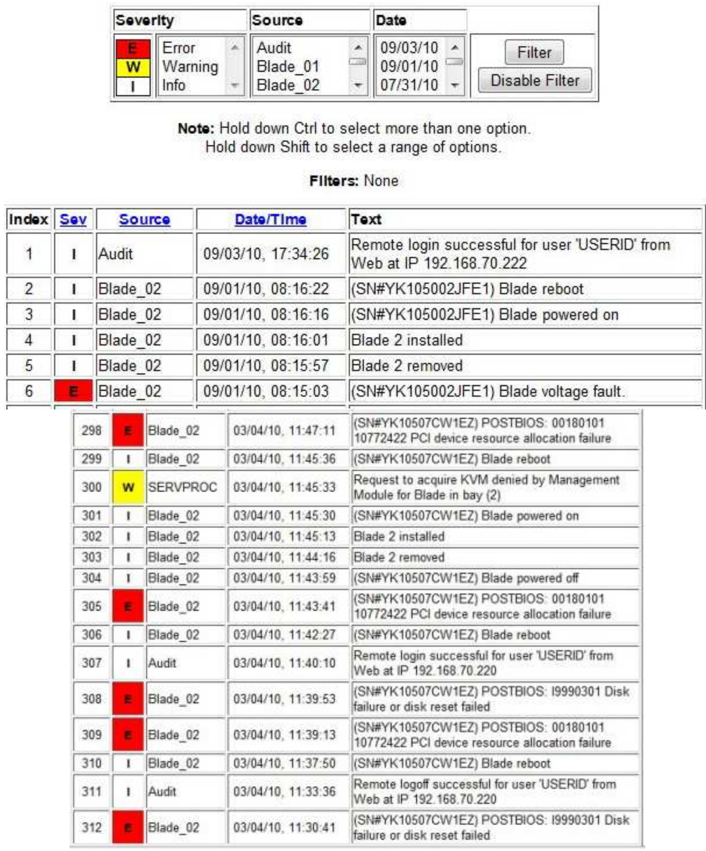
Figure 3. Resources monitoring Blade 1 and Blade 2

Figure 4 Monitoring resources load (CPU) Blade servers (1 and 2)

Signaling of Error Apparitions and Insertion in the Cases Base

Also, in the case in which appears a new entry that signals an abnormal functioning or a deviation from the nominal operating parameters of some physical resources, the agent has the role to undertake the record and to insert it into the database along with the degree of discretion or impairment of the functional status.

In the figure below are some examples of errors that have occurred throughout the case study, which were inserted into the cases base.



Index	Sev	Source	Date/Time	Text
1	I	Audit	09/03/10, 17:34:26	Remote login successful for user 'USERID' from Web at IP 192.168.70.222
2	I	Blade_02	09/01/10, 08:16:22	(SN#YK105002JFE1) Blade reboot
3	I	Blade_02	09/01/10, 08:16:16	(SN#YK105002JFE1) Blade powered on
4	I	Blade_02	09/01/10, 08:16:01	Blade 2 installed
5	I	Blade_02	09/01/10, 08:15:57	Blade 2 removed
6	E	Blade_02	09/01/10, 08:15:03	(SN#YK105002JFE1) Blade voltage fault.

298	E	Blade_02	03/04/10, 11:47:11	(SN#YK10507CW1EZ) POSTBIOS: 00180101 10772422 PCI device resource allocation failure
299	I	Blade_02	03/04/10, 11:45:36	(SN#YK10507CW1EZ) Blade reboot
300	W	SERVPROC	03/04/10, 11:45:33	Request to acquire KVM denied by Management Module for Blade in bay (2)
301	I	Blade_02	03/04/10, 11:45:30	(SN#YK10507CW1EZ) Blade powered on
302	I	Blade_02	03/04/10, 11:45:13	Blade 2 installed
303	I	Blade_02	03/04/10, 11:44:16	Blade 2 removed
304	I	Blade_02	03/04/10, 11:43:59	(SN#YK10507CW1EZ) Blade powered off
305	E	Blade_02	03/04/10, 11:43:41	(SN#YK10507CW1EZ) POSTBIOS: 00180101 10772422 PCI device resource allocation failure
306	I	Blade_02	03/04/10, 11:42:27	(SN#YK10507CW1EZ) Blade reboot
307	I	Audit	03/04/10, 11:40:10	Remote login successful for user 'USERID' from Web at IP 192.168.70.220
308	E	Blade_02	03/04/10, 11:39:53	(SN#YK10507CW1EZ) POSTBIOS: I9990301 Disk failure or disk reset failed
309	E	Blade_02	03/04/10, 11:39:13	(SN#YK10507CW1EZ) POSTBIOS: 00180101 10772422 PCI device resource allocation failure
310	I	Blade_02	03/04/10, 11:37:50	(SN#YK10507CW1EZ) Blade reboot
311	I	Audit	03/04/10, 11:33:36	Remote logoff successful for user 'USERID' from Web at IP 192.168.70.220
312	E	Blade_02	03/04/10, 11:30:41	(SN#YK10507CW1EZ) POSTBIOS: I9990301 Disk failure or disk reset failed

Figure 5. Signaling appearance event

The Virtual Level and Virtual Machine (VL & VM)

Is the state of abstraction of the hardware resources and is also the boundary of separation between the physical layer (hardware) and the software one. In our study case, we pushed the limits of virtualized resources by assigning an increased number of virtual machines by overbid. By overbid we understand tat the use of a fixed number „n" of resources as we have „n+k" resources, where {n.k}> 0 (Figure 6). From the allocation made is clearly that in the physical machine Blade 1 we have 2 physical processors with 4 cores for each (the equivalent of 8 processors), 16 GB Ram and 700 Gb of storage space, of which we allocated to the virtual machines the correspondent of 18 virtual processors and 54 GB RAM.

This allows us a degree of increased use of the physical available resources due to the allocation on more virtual machines simultaneously of the processing power of which we have it through overbid. In other words, the virtual machine can have a maximum virtual resource allocated equivalent with the physical one.

The fact that we can overbid the available resources of the abstracted system leads us to a resource allocation per virtual machine that we consider that we must have or to dispose it at some point, event that resources was theoretically allocated initially to some machine. Thus, the share of the abstracted resource allocated from the physical resources on each virtual machine it will be in the interval $[\min n = \frac{\mathrm{n}}{\mathrm{n+k}}, \max n = \frac{\mathrm{n}}{\mathrm{n+k}}]$, where n is the total available natural resources and n+k the total number of the abstracted allocated resources (k is the overbid number of resources). Within the report we will detail the approach modality.

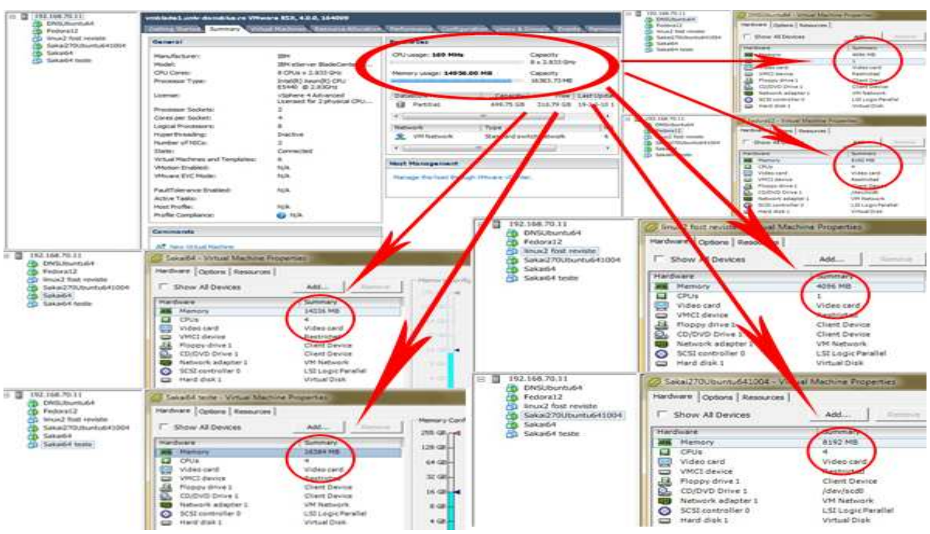


Figure 6 Virtual machine resource allocations

VL & VM Connectivity and Monitoring

On this level the intelligent agent, as in the case of physical layer, the IP connectivity but and the loading level of all the virtualized resources. We consider the IP connectivity the most important characteristic because this layer is the base of the virtual infrastructure.

Still here we dynamically manage the totality of the abstracted physical resources but and the allocation of the resources on each virtual machine (VM) as well as and their loading level.

Similar with the physical level, by a particular importance is the IP connectivity verification, with the help of the ping network tool (in our case, we utilized a demo called Server Ping), used to verify if the virtual layer as well as and the virtual machines are functional and presents connectivity, exemplified work in the figures below.

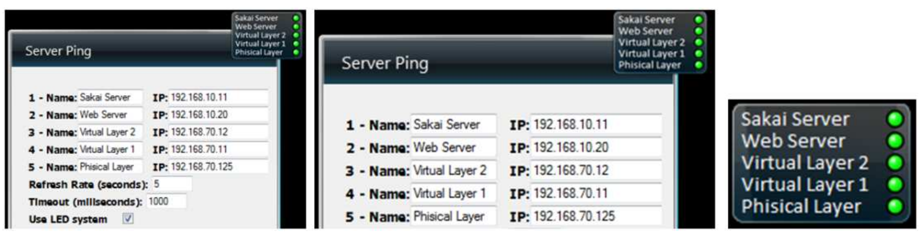


Figure 7. IP connectivity for VL & VM

The same modality of IP connectivity verification we have represented within the diagnostics site, where we find the state of the virtual machines, according to the figures below.

Figure 8. IP connectivity status of the abstracted servers

Figure 9. IP connectivity status of the virtualized machines

Thus, on each virtual machine we can allocate resources in a dynamic way, resources that we can manage quite easily with the help of the application VMware vSphere Client. The application allows us the visualization of all the physical abstracted resources and their allocation in dynamic way on the created virtual machines.

Here we are monitoring especially the loading of interest resources (processor, memory and disk) as and the evolution of their loading in time. Here, we talk of development we do not refer at the level of loading the resources but and to a balancing of their load.

We have also to mention that if the allocation of the available virtual resources is made for every virtual machine in part, the physical resources used in commune. Thus, if we have an "n" number of physical processors, the allocation of a "k" number of virtual processes (where$k = \overline{1, n}$), on each virtual machine is made according to the needs of processing that the system administrator considers them necessary at a given time, thereafter having the chance to increase or decrease the power allocated to the machine.

The paradox is represented by the fact that in the end we find that the amount of the virtual processors allocated to each virtual machine in part is higher than the total number of physical processors, this enabling the balancing of the processing power function by the loading that re giving the active processors. This is available in the case of all physical available abstracted resources.

Initially, the physical resources are divided proportionally to the amount of resources allocated on each virtual machine, and in the case in which a process needs more resources; these are dynamically distributed from the physical ones.

In other words, if a virtual machine, at a given time needs a high processing power, through balancing is allocated a higher percentage of the physical resource. In the figures below we illustrate the load level which it can have the virtual layer and the level of loading of the resources allocated to each virtual machine. Particularly, we will analyze the VL layer, but and a virtual machine on which is installed a application that is highly resources consumer and which is accessed by a large number of users. Here we refer to the virtual machine generically called Sakai64 after the application that supports it. On this virtual machine is installed a platform LMS called Sakai, application addressed to over 6000 clients and which requires considerable resources.

In the figure below is noted that on the physical machine Blade 1, correspondent to the IP 192.168.70.11 we have six virtual machines that use in commune the physical abstracted resources.

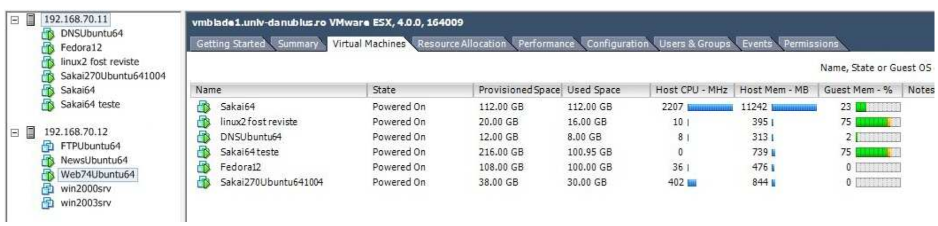


Figure 10. Blade 1 (192.168.70.11)

The loading degree of the abstracted resources, related to the physical machine Blade 1, is exemplified in the figure below.

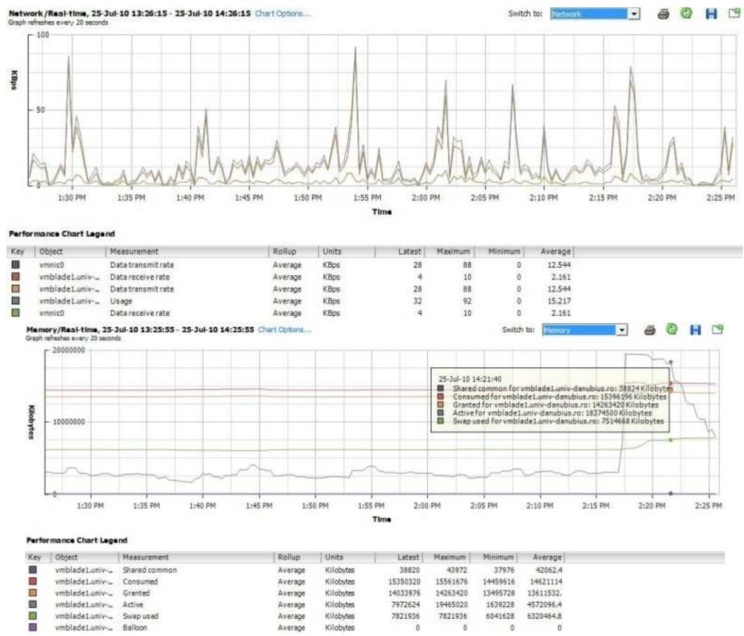


Figure 11 Network and memory loading Blade 1

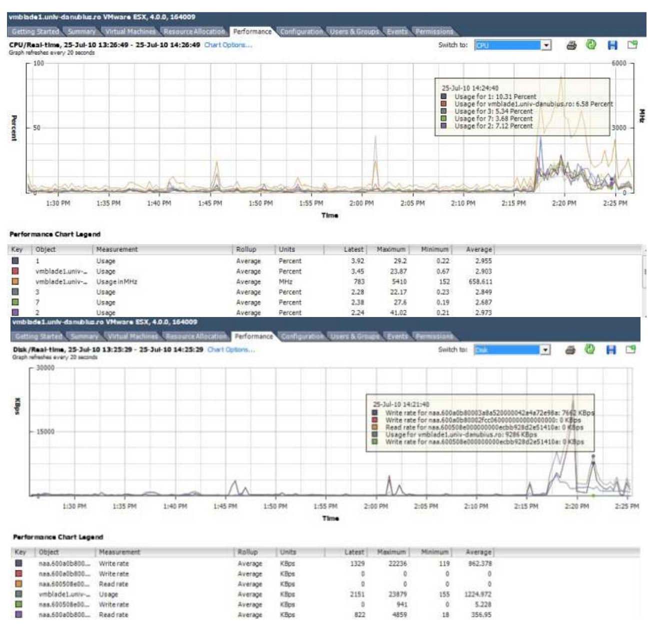


Figure 12. Processor and disk loading Blade 1

The virtual machine Sakai64 uses from the physical abstracted resources according to the figure below, but the interesting part is that from the total memory RAM allocated, the usage degree is around 90%.

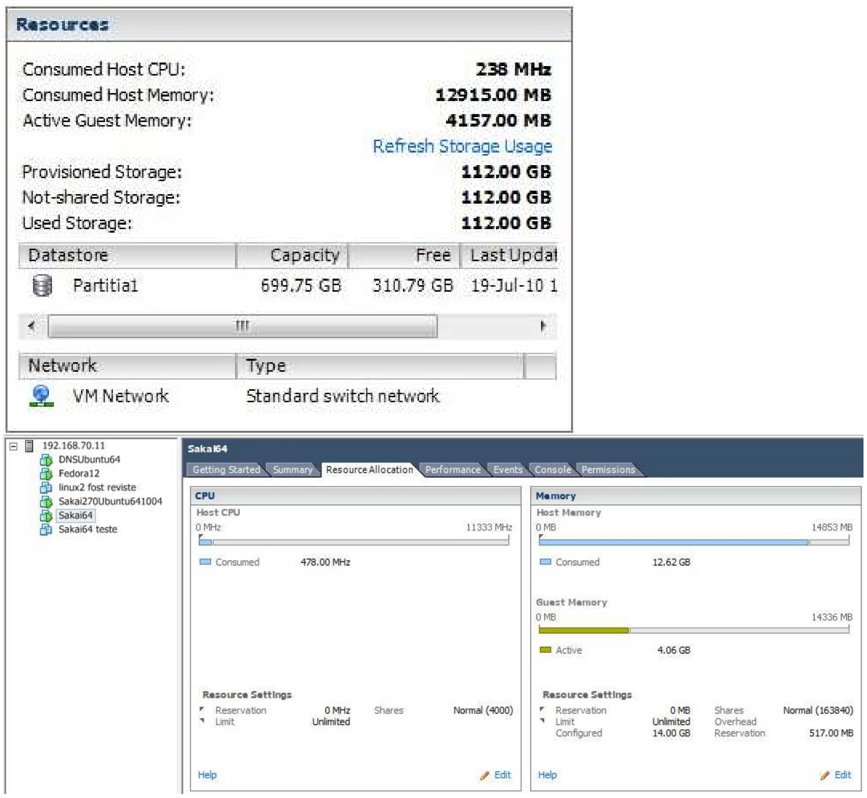


Figure 13 Sakai64 virtual machine allocations

Also the loading degree on each abstracted resource allocated to the virtual machine Sakai64 is exemplified in the figures below. We observe that from the allocated resources, the loading degree is rather low with the exception of RAM memory, which present a high degree (about 90%). This leads to the conclusion that in the case in which the applications will need resources, especially the RAM, we can expect a blocking or delay of the processors. Also, in the case of some increased number of requests from the clients we can expect at blockings or interruptions of the applications functionality due to the prioritization processes.

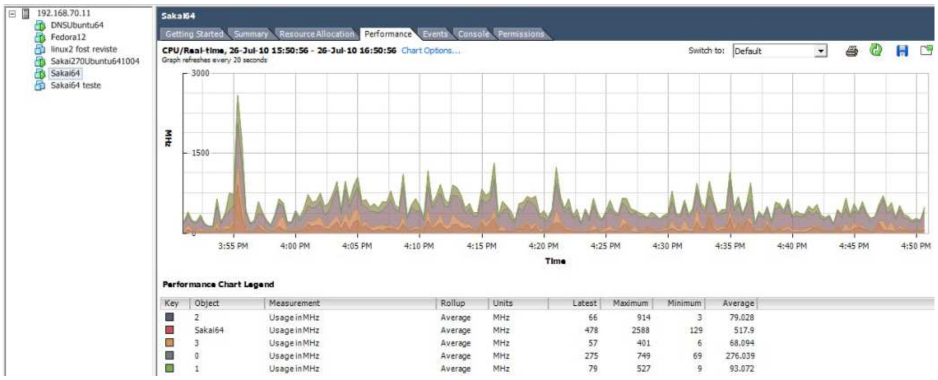


Figure 14 Processor loading virtual machine Sakai64

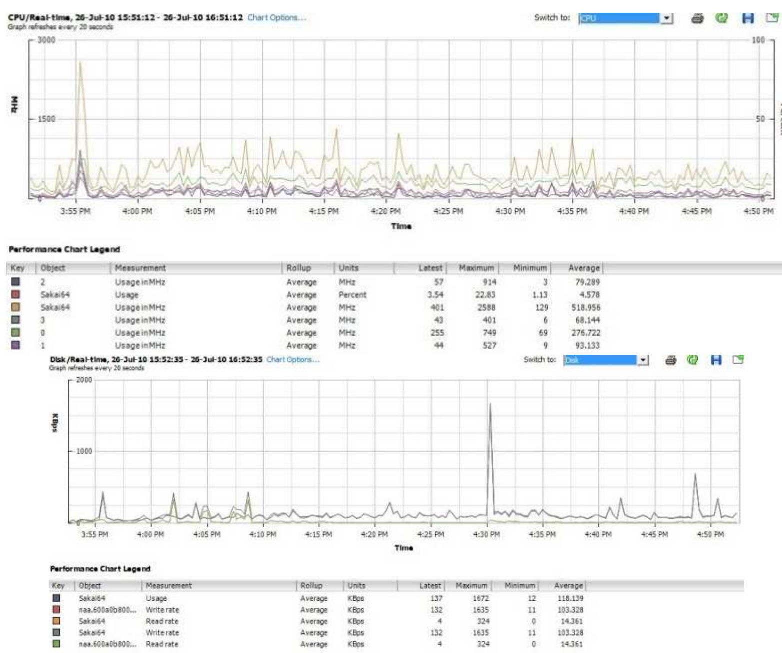


Figure 15 Processor and disk loading virtual machine Sakai64

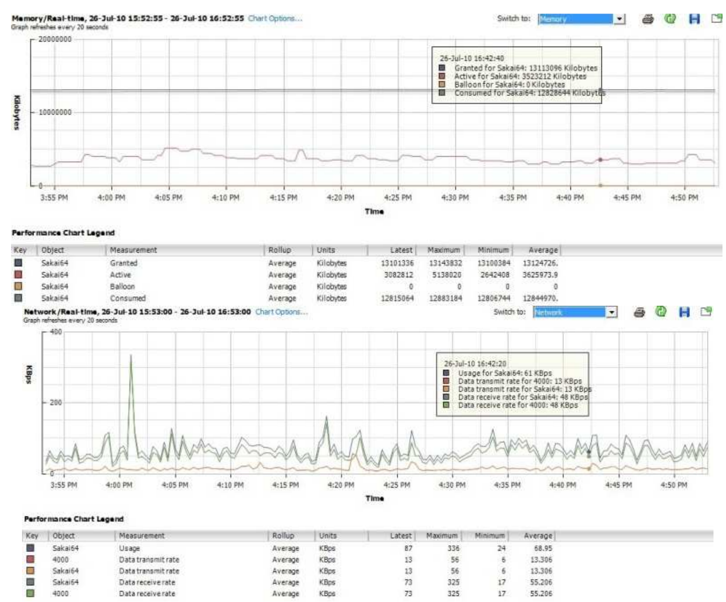


Figure 16 Memory and network loading virtual machine Sakai 64

OS & Apps Level

Installing on a virtual machine of the operating system (regardless of the manufacturer and version) and of the application does not poses particular problems compared to the classic installation when we have a physical machine. In the contrary, the applications and operating system once installed on a virtual machine, the latter can be duplicated as many times as we want, with the condition to rename them (not to have the same names) and to reassign them in network different IP's. In other words, we can have a growing number of servers unprecedented in a relatively short period of time. At the level of operating system the focus is directed towards the local resources management of application, transfer rate towards HDD, on the number of processes that are found in background and on the swap memory loading.

Also, on this level, the functionality of the applications is the most important objective that must be carefully followed. Here we put accent especially on the response time, at a request or login of the application, on the number of clients simultaneously connected and at the way in which their number affect the application's functionality. As we previously mentioned, we will focus our attention, within this level, on the Sakai application, which we will describe briefly below.

Sakai

The Sakai Project was initially launched by an academic collaboration between the University of Michigan, Indiana University, MIT and Stanford University, today bringing together over 200 universities that are using it, making integrant part of the Sakai community. Fulfilling the specific requirements of using the ICT in higher education, the collaborative environment Sakai was created by academics for academics, being oriented towards collaboration both in purposes of education, as well as for scientific research projects. Consequently, it can be used to create a Course Management Environment (VLE), which integrated with other collaborative applications leads to the completion of a portal dedicated for collaboration with all the university members (teachers, students and researchers) to achieve the objectives from the education and research.

Making a short trip through time, after a trial period of testing of 6 months with the version 2.5.0, on different hardware platforms having different operating systems – iMac (MacOS X Leopard), x86 (Ubuntu 8.04 LTS) and x86 (Windows Server 2008) – to observe the various problems that may occur at installation and utilization of the platform but and the incompatibilities of Sakai CLE with the operating systems and implicit with the hardware platforms, there were created several test of course sites where the students could experience and use the tools provided by the collaborative environment Sakai. The conclusions at which were reached were in the favor for starting the pilot phase, so it was decided the formation of the team T&L (T&L WORKING GROUP).

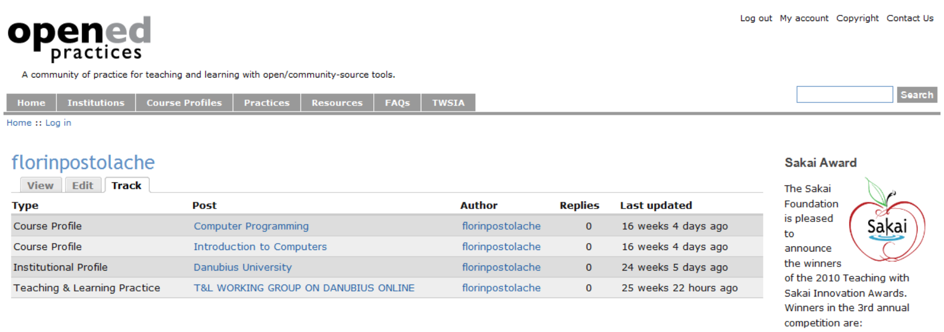


Figure 17. Opened practices community

A particularly important role in the positive evaluating of the collaborative environment Sakai, was that we are active members of the Sakai community and of Opened Practices community – A community of practice for teaching and learning with open/community source tools (Figure 17) (http://openedpractices.org/users/florinpostolache/track).

In the 2009 – 2010 academic year, it was put in service in the pilot phase the portal Danubius Online supported by the Sakai version 2.6.0. The portal operated with a relatively small number of course sites both from the license and from the master, the main objective being the accumulation of experience of usage in real conditions of exploitation. In the 2010 – 2011 academic year, it will be passed to a newer version Sakai 2.7.0, installed also on a visible virtual machine on Blade 1, called Sakai 270Ubuntu641004.

The portal was installed on a virtual machine (Sakai64) hosted on a server VMware ESX 4.0, with internal memory of 16 GB and with 100 GB space initially allocated on the Hard Disk, operating on a physical machine IBM Blade Center. It was used a database server MySQL 5.0, and as web servers there were used Apache 2.2.8 and Tomcat 5.5.26.

Installed on the virtual machine Sakai64 having as operating system Ubuntu Linux (64-bit), Sakai 2.6.0 is very important application, so that is why we will turn to it four our case study, especially to the fact that in present on this platform are enrolled about 6000 students but also from the consideration that, over a year, we tested this platform, having sufficient data.

Thus, during one year of testing, we concluded that the hardware platform is successfully facing supporting 6 virtual machines, which at their turn supports the application that work within normal parameters. From the data analysis, is showed that a large part of the physical available resources are not used at their full capacity. Thus on the physical machine Blade 1, the processor presents a load varying between 0.1 and 30%, this being apparent from the data collected within an hour over the several days in a year of testing. In the figure below is observed the CPU loading that varies between 0.29 and 11.56% on 12.07.2010.

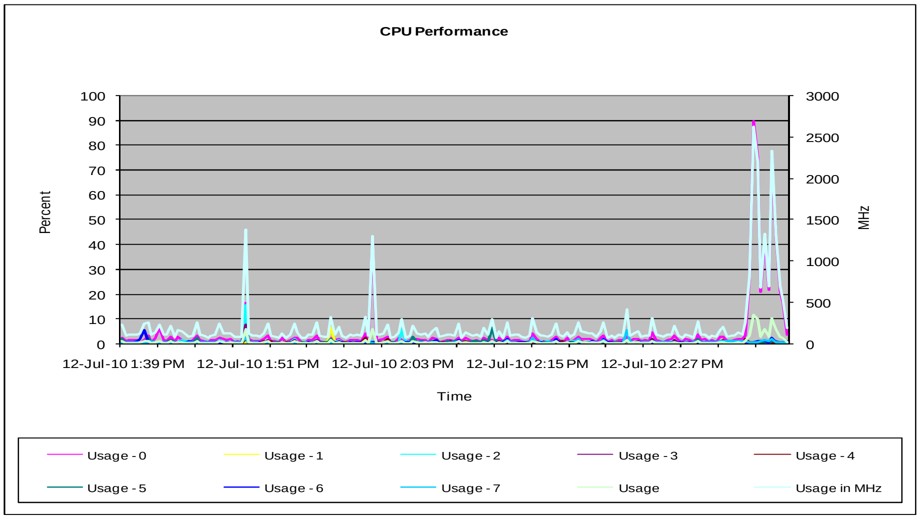


Figure 18 Processor loading Blade 1

From the data collected, it was found that identical to the processor's loading, the network and disk loading is placed at a low level. A special case presents the memory loading. From the data collected in the testing period we concluded that of the total memory available, over 90% is currently used, which is a sub-dimensioning of the memory capacity, visible fact and in the figure below.

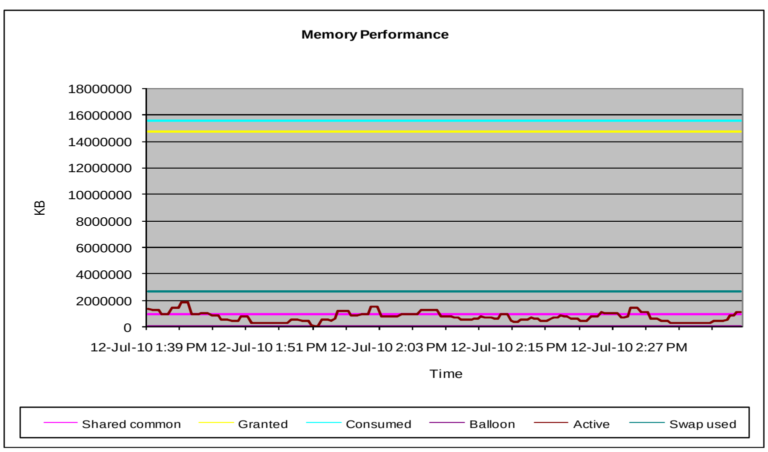


Figure 19 Memory loading Blade 1

Because the application is accessed simultaneously by a large number of clients, we followed the processes occurred at the level of operating system but and at the application level. Thus, in the connecting moment of a simultaneously large number of clients it was noted a delay, even a blocking on a short time of the application.

Also, it was followed the resources loading throughout the pilot phase and finally it was concluded that the processor, disk and network loading it is not major, but even insignificant. Instead, because of the insufficient physical memory as well and the allocation of a part to the virtual machine led to the conclusion that the usage Swap has radically increased, which explains the delay and short blockings in generating a answer to the incoming requests.

Also, we mention that a continuously running for a long period of time led to an abnormal functioning of the Tomcat server, presenting errors or blockings, which required restarting the application or of the virtual machine. Taking into account of all these factors, the knowledge acquisition seems a challenge as the process of abstraction.

Conclusions and Future Research Directions

1. Structuring the field concerning the facilitation of knowledge acquisition

By structuring the field we obtain the advantage that every specialized agent on a certain layer will identify and insert in the cases database only the cases where this is placed, allowing a better localization, allowing a deep level of discrimination of the components as precisely as possible, implicit a small granularity, the diagnostic in such cases being detailed facilitating a complete intervention in reworking.

2. Modelling interactions between the physical and virtual resources

The fact that we overbid the resources allows us a better balancing of the needs that an application may have at a time.

3. Weight of influence of a “situation” in the global functioning of the system

Depending on the degree obtained by the system at the moment t1 and the degree given to the system in the moment t2, results the share that is having the fault in the appreciation/depreciation of the functional state of the system.

4. Characterization of the global state of the system which intervenes in the punctual allocation of the resources by a critical factor (resources allocation for a machine which has critical need of resources) because:

1) Situation (state of fault) – network connection, client, background services
2) Components (processor, RAM, HDD)
3) Direction of recourses criticism
4) I/O constraints
5) Resource sharing

The future study will refer to the modality of approaching the virtual system if we see a fault functioning at the "n" virtual systems, to enter in functioning the system "n+1" Basically, when accessing some application by a large number of clients, is being tried a cloning of the virtual machine which support the respective application, thus allowing a clients redistribution, basically a relief of the traffic on a single machine and implicit a decrease of loading resources.

This it is possible to remove the blockage or delay in what concerns the response time to the customer requests

Bibliography

Muller, A., Wilson, S. & Happe, D. (2005). *Virtualization with VMware ESX Server*, Publisher: Syngress.

Russell, Steve; Olsen, Keith; Sallah, Gabriel; Seetharaman, Chandra & Watts, David (2003). *Server Consolidation with the IBM xSeries 440 and VMware ESX Server,* Publisher: Redbooks.

Rule, David & Dittner, Rogier (2007). *Server Virtualization*. Publisher: Syngress, Elsevier.

Kelbley, John & Sterling, Mike (2010). *Windows Server 2008 R2 Hyper-V: Insiders Guide to Microsoft's Hypervisor*. Publisher: Sybex.

Scott Lowe, Mastering VMware vSphere 4, Wiley Publishing.Inc, 2009, Indianapolis, Indiana.

Lowe, Scott; McCarty, Jason W. & Johnson, Matthew, K. (2010). *VMware vSphere 4 Administration*, Wiley Publishing.Inc, Indianapolis, Indiana.

Troy, Ryan & Helmke, Matthew (2009). *VMware Cookbook: A Real-World Guide to Effective VMware Use*, O'Reilly Media Inc, USA.

Laverick, Mike (2010). *VMware vSphere 4 Implementation*, The McGraw-Hill Companies, USA.

Vaughn, Stewart; Slisinger, Michael; Touchette, Larry & Learmonth, Peter & Layton, Trey (2010). *NetApp and VMware vSphere - Storage Best Practices*, NetApp, Inc, USA.

Bumbaru, Severin; Pusca, Andy & Postolache, Florin. *Teaching with Technology: Danubius University Case Study,* The 2nd International Conference on Institutional Strategic Quality Management in Higher Education ISQM 2010, 14-16 October 2010, Sinaia – Romania, http://proiecte.aracis.ro/academis/promovare-diseminare-transfer/conferinte-internationale/

Ariton, Viorel; Palade, Vasile & Postolache, Florinn (2007). Combined Deep and Shallow Knowledge in a Unified Model for Diagnosis by Abduction. *Euroeconomica* No. 1(20)/2008. http://EconPapers.repec.org/RePEc :dug:journl:y:2007:i:19:p:33-42

Ariton, Viorel & Postolache, Florin (2006). The Diagnosis by Abduction Using Human Expert Knowledge. *Acta Universitatis Danubius. Œconomica*, No.1/2006.

Postolache, Florin; Bumbaru, Severin & Filip Alin Constantin (2010). *Framework on Virtualisation Applications and Benefits*, The 4th International Workshop on Soft Computing Applications - SOFA2010, 15-17 July 2010 – Arad, Romania, IEEE Catalog Number: CFP1028D-CDR.

Constantin, Filip Alin; Postolache, Florin; Curteanu, Valentin; Tureac, Cornelia Elena; Marinescu, Liviu Mihail & Marinescu, Emanuel Stefan (2010) *Framework of Danubius Online Collaboration and Learning Environment*, The 6th International Seminar on Quality Management in Higher Education, QMHE 2010, 8-9 July 2010, Tulcea, The Center for Continuing Education and Training (CETEX) at "Gheorghe Asachi" Technical University of Iasi, Romania, http://www.cetex.tuiasi.ro/qmhe2010/index.php

Răileanu, Alina Beatrice; Vîrlan, Gabriela; Ariton, Doinita & Postolache, Florin (2010) *Collaborative Learning and Research Environment on Danubius University Portal Web 3.0 Generation*, The VIII-th International Conference 'Challenges in Higher Education and Research in the 21st Century,'CHER 2010, 2-5 June 2010, Sozopol – Bulgaria, http://elfe.tu-sofia.bg/cher21/

Postolache, Florin; Bumbaru, Severin & Ariton, Viorel *Knowledge Acquisition and Structuring for Diagnosis in Complex Virtual Systems* (2010). The 1st International Conference “Advances in Engineering & Management” ADEM 2010 – Drobeta Turnu-Severin, 19-21 Mai 2010, Faculty of Engineering and Management of Technological Systems, Drobeta Turnu-Severin, University of Craiova, http://www.imst.ro/adem/adem.htm.

http://www.vmware.com/.

http://www.brocade.com/index.page/.

http://www.php.net/.

http://www.mysql.com/.

http://openedpractices.org/institution/danubius-university/.

http://openedpractices.org/users/florinpostolache/track/.