DOI 10.29042/2017-862-868

Helix Vol. 7(1): 862-868

Role of Virtual Machine in Bio-Applications and its Significance

[*1] P. R. S. Naidu, [2] G. Lavanya Devi, [3] P. Sateesh, [4] B. Srinivas

[1, 3, 4]MVR College of Engineering, Andhra Pradesh, India, [2] AU College of Engineering, Andhra Pradesh, India

**Email: prsn1988@gmail.com*

Received: 10th Dec 2016, Accepted: 22nd Dec2016, Published: 1st Jan 2017

Abstract

Virtualization is becoming more and more important in the science of living things, enabling group of devices made up of smaller parts and provisioning of complete computer setups, including operating system, data, software, and services packaged as virtual machine images. In virtualization grid point of view, we discussed about relationship among Application Layer, Grid Middleware and Network Layers and also the limitations of using grids. And for experimantal setup or for practical analysis, we deployed Virtual Machine in NUTSHELL in which the properties of VM and also calculated the perfomacnce metrics based on different constituents like Querying, Copying, Pausing, Unpausing etc… are being discussed. However, virtual machines are very much useful in bio appliactions to abstarct diferent physical resources to enable multiple instances of software to utilize the same resources.

Keywords

VMM, VMManager, VM Repository, OGSA, OGSI, GRAM, RSL

Some General Terms:

VMM (Virtual Machine Monitor) – It is a tool which provides the interface between a Virtual Machine and the host machine. Examples: VM Ware, XEN

VMManager – For interaction between client and VMM we use a Grid Interface i.e., VM Manager

VMRepository – Grid service which catalogues VM images of a VM and which stores them for retrieval and deployment

Authorization Service – To know the user authorization VM Manger and VMRepository services call to check whether the requested operation have been performed or not.

1. Introduction

About Virtual Machines:

For any virtual machine to work, it has to be emulated. As many people will know, for a machine to work there is need for hardware and software (OS which consists of SHELL and kernel). Hence for hardware we allot some memory and processing power (RAM) to virtual machine we want to run. Now coming to software it can be three types. i.e., Software emulation, Hardware emulation and NO emulation.

In case of software emulation, there is one more layer on the top of Host Kernel called Hypervisor. It is like a manager, which would handle all the virtual machines running on it. Each virtual machine would have its own kernel and OS. Example: Virtual Box
In Hardware emulation we don't have Hypervisor, the host kernel is shared by all the virtual machines running on it. As a result of this we have overhead and more number of VMs can be run as compared to software emulation. This emulation is very useful to make virtual network in terms of memory and processing. Example: UML.

About Bio-Informatics:

Bio-Informatics is concerned with cellular related things. Example: Molecular Biology, Computational Biology etc. It is used to describe the whole larger field of which includes clinical informatics like creating, using data from clinical systems etc. And another thing about bio-informatics dealt with normalization of data. Bio-Informatics people are constantly dealing with hodgepodge of different data structures. The pain one has to undergo when integrating one biological database with another becomes complicated. Some bio informatics services are only accessible with perl, others with java, then again one need R and recently also should be proficient in python too. In short there is lot of overhead in doing Bioinformatics the old way.

2. APPLICATIONS

Bioinformatics Applications in general are divided into 4 categories.

1. **Distribution of Bioinformatics Tools:**
   Integrated bioinformatics platforms like chipset [10] needs many tools and databases to support their respective functionality. Virtual Machines is the only solution for distributing these tools and databases likewise the original server environment.

2. **Distribution of Bioinformatics Data:**
   Data Sets increases rapidly in the field of bioinformatics, in which downloading and

performance analysis are done in local computing resources which results in infeasibility of using public resources directly. And also this results in lot of work burden to local administrators and bioinformatics researchers to set up and query the data from data base resources each time. In this case, VMs play very important role and offers best solution where data and respective database software together with associated middle wares are packaged and delivered to the users which results in simplifying the establishment of local mirror of data source.
Example: cheminformatics tools [11]

3. **Reproducible Analysis:**

Virtual Machine Images allow performing reproducible analysis where all data, tools, and scripts can be packaged in a VMI by taking a snapshot of the system where the actual study was performed which results in easy sharing of complete experimental workflows. [12]

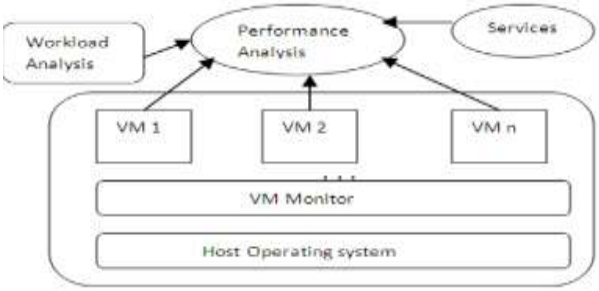
4. **Providing adequate facilities in Bioinformatics Education:**

If anyone who has given the subject that involves computers and software and also rapid change in versions results in crashing the program. This problem is mainly faced by beginners where they have to execute so many commands in Linux based OS which might finally reflects the subject being taught. So to avoid this problem, an environment must be set up which is identical to what the instructions are made for is to let every student start a VM containing everything needed for the exercise which results to increase performance like it reduces the startup time and makes it to run smoothly.

3. literature survey

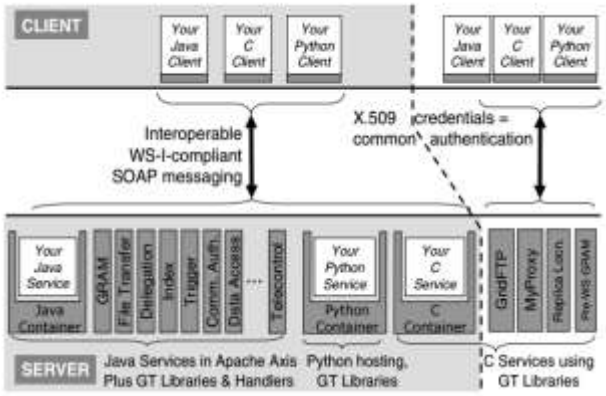
Performance metrics and gathering of resources of virtual machines completely depends upon the physical appearance like size and complexity of present day computing systems. [1] Virtualization plays a very important role in sever consolidation, availability of resources, difficulty of operating system deployment and disaster recovery. One of the main advantages of virtualization is throughput can be maximized with minimum loss of CPU and I/O efficiency. The diagram briefly depicts the analytic framework of virtualization.

Figure 1.1: Analytic Framework



Globus Toolkit-4 provides significant improvements over previous versions of performance, robustness, usability, documentation, compliance and functionality [2].The below figure gives the overview of GT-4.

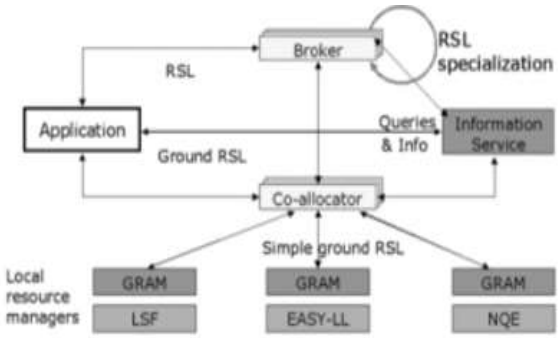
Figure 1.2: Schematic Architecture of GT4



White boxes: user code, Shared boxes: GT4 code

Research has been carried out in Argonne National Laboratories [3] where the core services and functionalities of Globus Toolkit are described in detail which address Global Security Infrastructure, resource management (it allows allocation and co-allocation of computational resources) and the Grid Service.

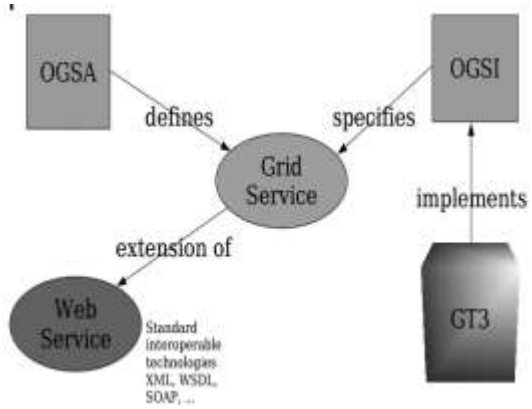
Figure 1.3: Overview of Resource Management Arhitecture



GRAM -Globus Resource Allocation Manager
RSL -Resource Specification Language

Micheal Brwon described the realtionship among OGSA,OGSI,GT3 shown below and also [4] the importance of Grid Computing, GT Security Services and diffferent interfaces.

Figure 1.4: OGSA/OGSI/GT3 Relationship Mode



OGSA -Open Grid Services Architecture
OGSI -Open Grid Services infrastructure

Edvin described about various cloud virtual machine architetures, [5] security aspects, advantages and disadvantages. Examples of virtual machine architecture are HIMA, LARE, KVMSEC,XENACCESSVMSCOPE.
Microsoft provided and in a way of [6] discovering new security mechnaisms in the area of bio-informatics. Ian Foster [7] along with Carl, Jeffery, Steven described the importance of phisiology of grid in distributed systems. They also defined OSGA: the globus tool kit for scientific and technical computing and also web services. The below figure depicts the Globus Toolkit mechanism. And the figure depicts example/workflow of demo application.

Figure 1.5: Globus Toolkit Mechanism

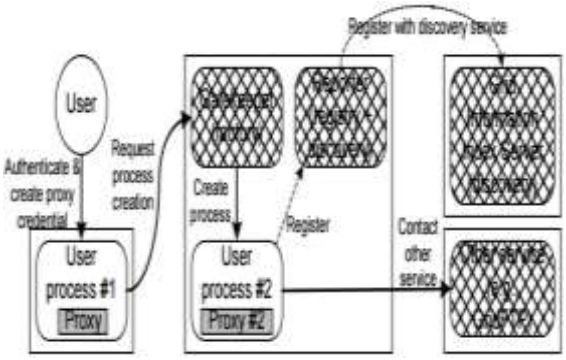
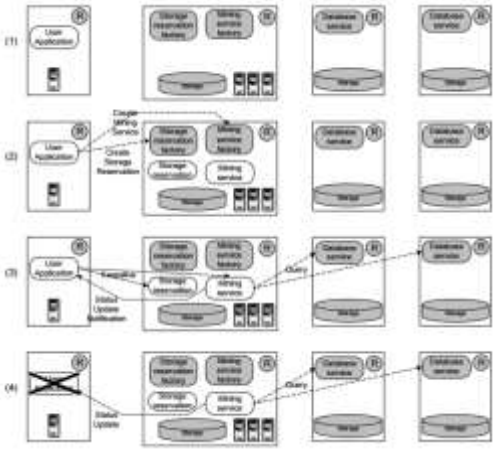


Figure 1.6: Example of Grid Services during work



4. SECURITY THREAT ASSESSMENT IN BIOINFORMATICS

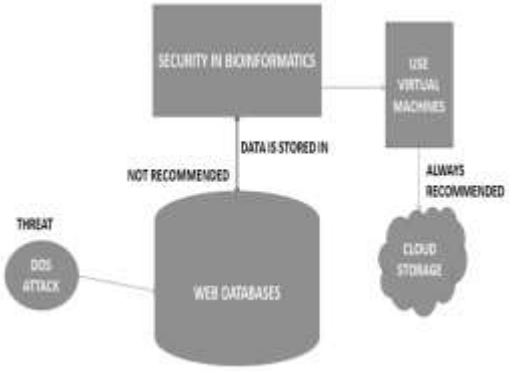
The strict combination of different things together that work as one unit of security within bioinformatics application development complicates the process, adding time to development and possibly its availability. More than that, using/getting to certain data sets or interacting with certain utilities may be complicated to make sure that security measures are properly met. On the user end, time and effort must be spent to figure out the worth, amount, or quality of the safety of using a utility rather than just completely/in a hinting way trusting the utility. These security things to carefully think about can interfere with or fight against scientific research for available useful things/valuable supplies.

DoS (Denial of Service) attacks are one of the major threats which make networked resources unavailable to the users which could seriously impact research efforts, particularly as many bioinformatics utilities such as Web services or API to access content and also causes failure to retrieve such content which could interfere with an entire application pipeline. Databases are vulnerable to insidious attacks such as change of record to faulty information in web database and returning back faulty results in the front end of web application. This results in wastage of both time and money, which in turn leads to web application security breach. While dealing with medical applications, the stakes could be even higher.

Pharming scams could be applied to hosted bioinformatics databases and applications which lead to industrial espionage. Monitoring user submission of data and requests with IP addresses make it possible to gain insights into the research activities of other research laboratories.
Web-based cloud hosting and distribution of source code and executables results in enhancement of security things to carefully think about, which includes the possibility that an application may be

used as a security breach like Trojan horse, an evil and cruel purpose hid behind some appearing to be useful ability to do things This evil and cruel ability to do things could damage files or online or paper forms that ask for a job, money, admission, etc. are used for spying by sending data location on the computer into the computer program to a remote site.

Figure 1.7: Overview of Security in Bioinformatics wrt attacks and databases



Here the open source nature of much bioinformatics application development offers both advantages and disadvantages. The ability for anyone to read the code associated with the application can make such breaches easier to spot, but the community-based approach to development can also facilitate such breaches into the application source, since a security flaw could be added by the incorporation of a submitted patch or enhancement.

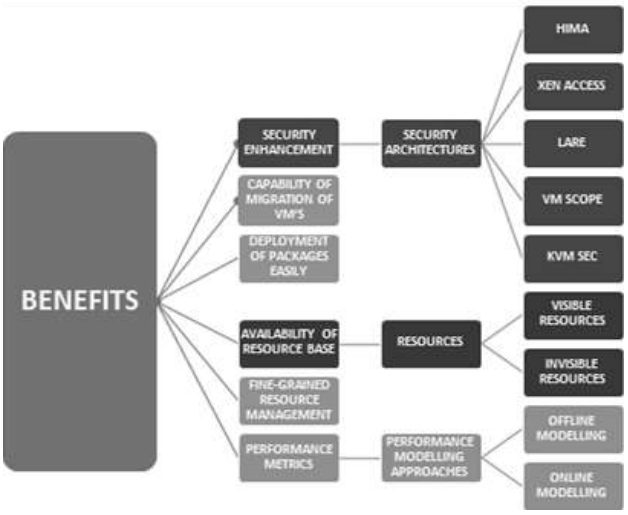
Potential/Physical security failures cause by application installation need not be evil and cruel; many uses have been released that contained carelessly accidental security bold or daring acts, such as buffer overflows. This way bioinformatics computer programs could introduce almost the same security weaknesses that could be used to hurt something or someone, especially when developers do not take the possibility into the process of carefully thinking about something.

Users of bioinformatics software should figure out the worth, amount, or quality of any downloaded computer program for security bold or daring acts, on purpose or not, before use. And it is always recommended to the users of bioinformatics to use the process of virtualization. (Implementing through virtual machines)

5. BENEFITS OF USING VIRTUAL MACHINES

Security Enhancement: Virtual Machines provide isolation mechanism between a user and a resource and also among users themselves which results in enhancing the features of security. This can be achieved by different security architectures. [1].

Figure 1.8: Benefits of Virtual Machines



Capability of Migration of VM's: VM Image which is executable at one resource can be easily transferred to another resource and also can be restarted in milliseconds.

Deployment of packages easily: VM's acts as a distribution packages, and can be duplicated by just copying the VM Image.

Effectiveness in Cost: Virtualization technology, which offers users the ability to pool their hardware resources, is playing a major role in making IT more flexible and cost-effective.

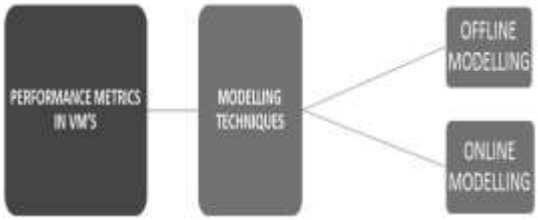
Availability of Resource Base: Virtual Machine can be configured before only with OS, library signature and applications based on the user requirement. Then the installation and deployment can be done on different nodes independently with respect to node's configuration.
Resources are categorized into two types:

1. **Visible Resources:** These resources of OS/VMM include core, memory capacity and potential I/O devices which are currently exposed or visible.
2. **Invisible Resources:** These resources of OS/VMM include shared cache and core pipeline resources.

Fine-Grained Resource Management: Resources usage may be confined within most VM implementations.

Performance metrics: In order to achieve VM performance, it is mandatory to characterize all shared resource contention effects in the platform. It is then achieved by two modelling techniques.

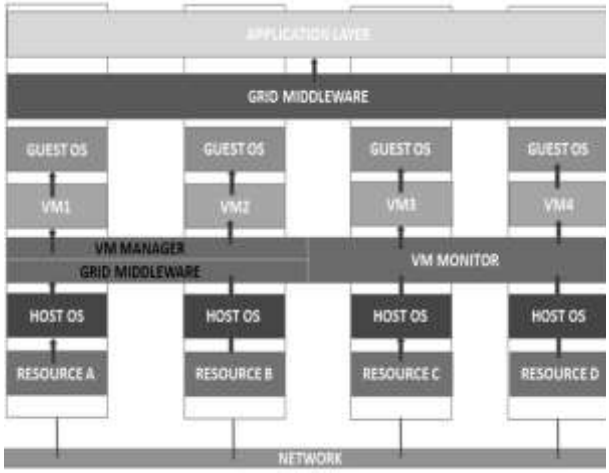
Figure 1.9: Types of Modeeling Techniques



1. **Offline Modelling:** This assumes that the workload performance can be measured on several platform configurations alone as well as pairwise with other virtual machines.
2. **Online Modelling:** No offline analysis is available and the characterization of the VM performance effects needs to be done online.

6. QUALITY OF LIFE IN VIRTUAL MACHINES WITH RESPECT TO GRID MIDDLWARE

Figure 2.0: Relationship among Application Layer, Grid Middleware and Network Layers



The low level features of our architecture are detailed in the diagram to the right. The diagram describes for nodes, each running a (potentially different) host OS. Each node is running a VMM and a VMManager Grid Service. On top of that layer, run the actual VMs, which are installed with Grid software, allowing them to be run as Grid nodes. The VMs could also be used as independent execution environments, without Grid middleware installed on them. (Instead they would run applications directly).

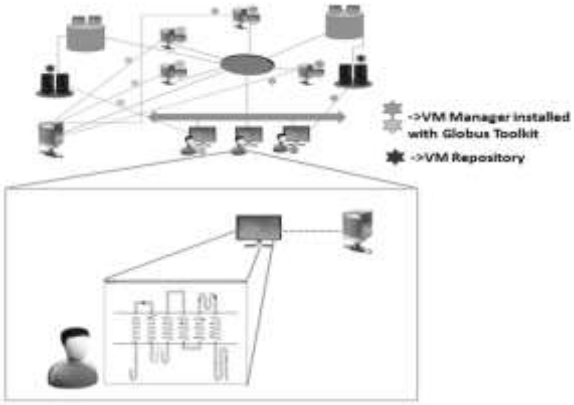
7. LIMITATIONS OF USING GRIDS

a. Customized software configurations must be made for complex applications. This type of environments may not be available in Grid nodes.
b. Manual installation of scientific applications can be arduous, lengthy and error prone; it would be helpful if this process is amortized over many installations.
c. The current mechanism of usage of UNIX accounts is not sufficient with respect to security. So in order to achieve this, isolation of Grid Computation is a key security requirement.
d. The current Grid Frameworks do not support the ability to migrate or restart the applications which would be a enormous value of Grid Environment.
e. Fine-grained resource usage enforcement is critical for more efficient use of Grid resources, yet such technology is not widely available.

8. VIRTUAL MACHINE DEPLOYMENT IN NUTSHELL

VM Deployment

Figure 2.1: VM Deployment in NUTSHELL



In NUTSHELL: Instead of running Grid software within VMs, we integrated VM deployment into the Grid infrastructure: mapping a client credential to a Unix account was replaced by deploying a VM and starting the client's environment within it.

The VM deployment process has 3 major steps:

1. The client queries the VM repository, sending a list of criteria describing a workspace. The repository returns a list of VM descriptors that match them.
2. The client contacts the VMManager, sending it the descriptor of the VM they want to deploy, along with an identifier, and a lifetime
3. for the VM. The VMManager authorizes the request using an access control list.
4. The VM instance is registered with the VMManager and the VM is copied from the VMRepository. The VMManager then interfaces with the VMM on the resource to power on the VM.

After a scientist has deployed a VM onto the resource, he may run an application in it. For this purpose, each VM was configured with the Globus Toolkit. The above picture represents a scientist running the program, creating an image of a transmembrane protein.

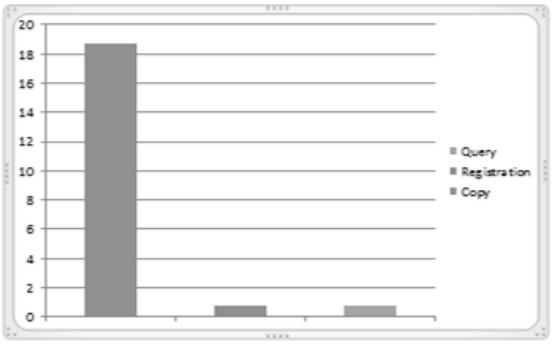
The below architecture is being implemented by using Globus Tool kit, which is an open source grid

middleware toolkit. It provides a framework for data, resource and security management.

The graph to the shows the proportion of time taken by the constituents of the deployment process, which is always measured in seconds.

The authorization time is not included, but it is compared to registration time. The dominant factor in overall deployment time depends on network latency and bandwidth.

Graph 1.1: Proportion of time taken by the constituents (Query, Registration, and Copy) (in seconds)



VM Properties: A VM constitutes a virtual workspace configured to meet the requirements of Grid computations. We use an XML Schema to describe various aspects of such workspace including virtual hardware which includes RAM size, disk size, Virtual CD-ROM etc, installed software including the operating system (e.g. kernel version, distribution type) library signature, other properties such as image name and VM owner. Based on those descriptions VMs can be selected, duplicated, or further configured.

VM Migration: Migration of applications from one node to another is easier while **i**ntegrating Virtual Machines with Grid technology. The steps are as follows:

Using Grid software, the client freezes execution of the VM

The client then sends the "migrate" command to the VMManager, specifying the new host node as a parameter

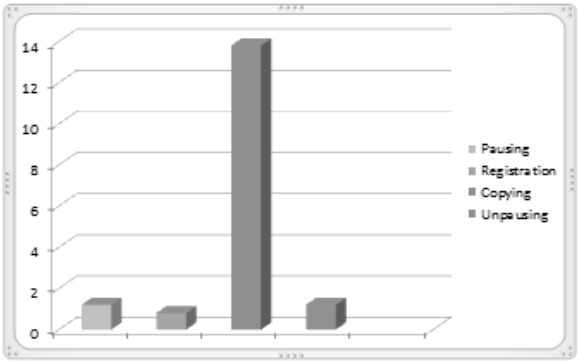
After checking for the proper authorization, the VM is registered with the new host and a Grid FTP call transfers the image.

In terms of performance this is on a par with deployment – it is mainly bound by the length of transfer. In our tests, we migrated a 2GB VM image from two identical nodes through a Fast Ethernet connection.

The graph below shows the proportion of time taken by the constituents of the deployment process. It is always measured in seconds. And the graph does not include time for authorization, but those times and registration time is compared. The migration time depends on the network latency and bandwidth. The pause and resume times always depends on 3rd party VMM.

Performance Implications: The performance of applications running on a VM depends on the third-party VMM's and the applications themselves. A purely CPU-bound program will have almost no performance which results in worsening all instructions which will be executed directly on hardware. Usually, virtual machines combine with privileged I/O instructions resulting in a performance hit for those instructions although new methods, such as those implemented by Xen, improve this factor. Putting all these things in use, we experimented with VMWare Workstation and Xen and in our experience slowdown was never more than 30% and is often less than 5%, where the Xen slowdown was *much* less than 30%.

Graph 1.2: Proportion of time taken by the constituents (Pausing, Registration, Copying, and Unpausing) (in seconds)



CONCLUSION

To conclude with, many researchers of life sciences lack knowledge in the area of informatics expertise in which they can be able to install all the necessary components to run complex workflow analysis, where Virtual Machines plays very important role in providing great assistance for complex systems. For example, if scientists want to run different analysis like proteomics, RNS Sequence analysis, they can simply download and switch/migrate among images without reconfiguring the computers once again. And by the way, Virtual Machines are getting popular in bioinformatics arena. Furthermore, one can discuss about security aspects of Virtual Machine with respect to Bioinformatics.

REFERENCES

[1] Praveen, G. "Analysis of performance in the virtual machines environment." (2011).

[2] Markov, Stefan. "Globus Toolkit Version 4: Software for Service-Oriented Systems." (2006).

[3] Aloisio, Giovanni, et al. "The grid-dbms: Towards dynamic data management in grid environments." International Conference on

Information Technology: Coding and Computing (ITCC'05)-Volume II. Vol. 2. IEEE, 2005."

[4] Micheal Brown "Introduction to Grid Programming with the Globus Toolkit Version 3" (2003)

[5] Bright Prabahar P "Survey on virtual machine security" (2012)

[6] Juha Saarinen "Microsoft releases encryption tech for bioinformatics " (2015)

[7] Foster, Ian, et al. "Grid services for distributed system integration." Computer 35.6 (2002): 37-46.

[8] Tickoo, Omesh, et al. "Modeling virtual machine performance: challenges and approaches." ACM SIGMETRICS Performance Evaluation Review 37.3 (2010): 55-60.

[9] Nocq, Julie, et al. "Harnessing virtual machines to simplify next-generation DNA sequencing analysis." Bioinformatics 29.17 (2013): 2075-2083.

[10] Chipster: An open source platform Analysis http://chipster.csc.fi/

[11] Ochoa R, Davies M, Papadatos G, Atkinson F, Overington JP. myChEMBL: a virtual machine implementation of open data and cheminformatics tools. Bioinformatics. (2014); 30:298–300.

[12] Amazon EC2 – Elastic Compute Cloud. (2015).

