

# Resource Isolation Analysis on Virtual Server Performance

Mardhani Riasetiawan, Ahmad Ashari, Sasongko Wahyu Kumoro

**Abstract**—Virtual Server on cloud technology environment has advantages on maintenance, scalability, security and control. Virtualization have capacities on operating system, storage and network resources allocation. Virtualization Machine Management is manage the several machine to deliver the services through application such as Proxmox, Openstack and Eucalyptus. Virtual Machines performance need to be analyse on resource isolation to manage the computation process performances. The paper works on analysis on virtual machine simulation, benchmarking on Proxmox VE 2.1, Ubuntu Eucalyptus Faststart 2.0, OpenStack Essex. The analysis goal is to determine the resource isolation performance in each virtual machine on handling computation process. The analysis has resulted Eucalyptus have the resource isolation performance better than others. Openstack has speed performance better than other on computation process.

**Index Terms**— Virtual Server, VMM, Proxmox, Eucalyptus, OpenStack, Resource Isolation, Computation Process.

## 1 INTRODUCTION

Virtual server is bring the server into several virtual resource to use together in infrastructures. The operation system, memory, processor, and storage can manage into several node and services. Virtual server work with virtualization technology on applications. Hypervisor technique use to handle the process and establishing paralel process on it. Hypervisor is managing the operating system and avoid to dis-trube each other.

Hypervisor [1] is also call Virtual Machine Management as like as manage several virtual machines. Virtual machine has advantage on delivering reliable performance infrastructure services. It can monitor by analyzing the resource isolation performances.

Proxmox [4], OpenStack [5] and Eucalyptus are the prominent software in virtual machine management. This three software is use by infrastructure to deliver the infrastructure as services. They can manage several servers on single resource and allocate the server resources to all virtual servers.

The virtual server has performance based on the resource baseline quantity and resource allocation management [3,6]. They need to manage processor, memory, storage and network to optimize the performance.

The research do comparative analysis on resources isolation in Proxmox, Eucalyptus and OpenStack. The research goal is analyze and discover the best performance on the virtual machines with the case on computation time ratio. The research use simulation model, implementing Proxmox VE 2.1, Ebuntu Eucalyptus Faststart 2.0, and Openstack Essex. The research

does not address the security issue. The research use scalabil-ity parameter, there are hypervisor characteristic on CPU re-source isolation. The research use Maxima application to ana-lyze the scalability and Htop for monitor the CPU resources.

The paper will consist several section, the first is introduc-tion to describe the research background and goals. The se-cond section describe the virtualization, scalability, and virtual machine. The Section III describe the methodology, and sec-tion IV describe the implementation and result. The last sec-tion describe the result and discussion.

## 2 VIRTUALIZATION

### 2.1 Virtualization

Virtualization is technology to build virtual version of physi-cal server resources, such as operating system, storage and network resources [1]. There are two method on virtualization, Full Virtualization and Para-Virtualization.

### 2.2 Full Virtualization

Full Virtualization [7] it is also call Virtual Machine Monitor, is same like hardware emulation. In this way, the full virtualiza-tion is emulate the all hardware for the operating system guest. The technology will use alomost all of the resources. The application that use the technology such as KVM and Xen. The explanation of the Full Virtualization architecture describe on Figure 1.

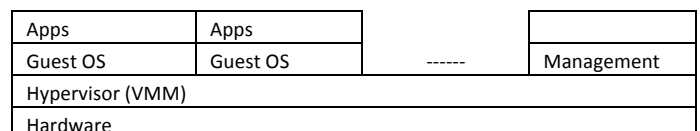


Figure 1. Full Virtualization Architecture

### 2.3 Para-Virtualization

Para-Virtualization [7] method use the kernel to customize virtualization. The goals if Guest and Host will use same lan-guage to communicate with the resources. It can resulted more efficient resource usage. The method need to use the

- Mardhani Riasetiawan is currently pursuing Doctoral degree program in Computer Science in Universitas Gadjah Mada, Indonesia, E-mail: mardhani@gmail.com
- Ahmad Ashari is lecturer and researcher in Computer Science and Elec-tronics Department , Faculty of Mathematic and Natural Science, Univer-sitas Gadjah Mada, Indonesia
- Sasongko Wahyu Kumoro is reently graduate from undergraduate program on Department of Computer Science and Electronic, Universitas Gadjah Mada

same operating system, and in the several case support only odified operating system that supported the hypervisor, such as Openvz. The explanation of the Para-Virtualization architecture describe on Figure 2.

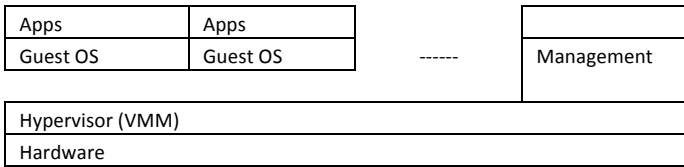


Figure 2. Para-Virtualization Architecture

Scalabilities is the system capabilities on handle the work load. The scalable system is manage the load with address the service level and without decreasing the performances in same time. There are several parameter can show the performance, one of them is resource isolation performance.

In the virtualization, resource isolation give limitation on resources for the components. It is manage the component not use the resource massively and distrube the performance. resources isolation ensure the equal situation between the virtual machines, and can access the resource in balance.

### 2.4 Openstack, Proxmox and Eucalyptus

Openstack [4] is one of the virtual machines technology that deliver the cloud operating system on public and provate under the Apache Licences. There are three project under Openstack, they are Openstak Compute (codename: Nova), Openstack Object Storage (codename: Swift), and Openstack Image Service (codename: Glance).

Proxmox VE [5] is linux distro based on Debian build special for hypervisor. Virtual Machine Manager (VMM) type 1 by default deliver OpenVZ and KVM in console mode. Proxmox use web based management to proceed all the activities. Proxmox have several capabilities on live migration, open source software support, backup and restore and role based administrator.

Proxmox is merge full virtualization and container visualization. Full virtualization is use KVM and on the container use OpenVZ. The technology give capabilities for Proxmox to create more VM host.

Eucalyptus is platform on delivering Infrastructure as a service supported by Ubuntu Operating system. Eucalyptus delvier the cluster virtual that can use as private and hybrid cloud. Maxima is a open source software on Computer Algebra System (CSA) that combine graphic, symbol and numeric. Maxima is use for computation on differentiation, integram, linear, polynomial, Laurent function, Taylor, 2d and 3D graphic. Htop is small utility that use for internal VPS/dedicated server resources on terminal.

## 3 METHODOLOGY

The research consist of several step on analyzing the isolation resource performance in three virtual machines application, Proxmox, Openstack and Eucalyptus. The research is use testing method with computation of square root of two. The test

uis use Maxima and Htop for isolation test. The research build simple application to capture the result on square root of two and time. The test is implement in Proxmox, Openstack and Eucalyptus.

### 3.1 Testing Method

The test is implement to capture the consume time on test with Maxima application. The test is act in 30 times. The test object is square root of two with maximum digit abot 1.500.000. The test is choosen because the processor resource is use continuously with low memory usage. The application is execute on script for documentation.

The resource isolation performance is implement in two virtual machines (VM1 and VM2) and act in paralel. The test scenario is prepare in tow ways. First, VM2 is execute two same command in the same times, VM1 is execute one command. This two process will execute in the same times. Ta is execution time on one vritual machine, Ta1 and Ta2 is execution application time on VM1 and VM2 (Quetier, Neri, et. all. 2007) test scenario can explain on Figure 3.

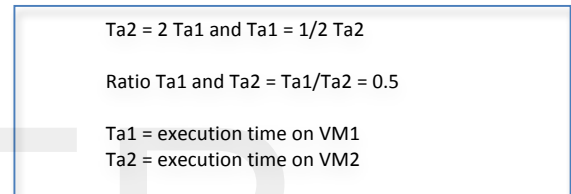


Figure 3. Time Execution

The test parameter in this research is result time on square root of two computation (maximaize 1.500.000 digit) on each machines.

For this purposes, the research is use Maxima and Htop on terminal. Maxima will implement script on square root of two and Htop is show the CPU usage. The virtual machines will implement to be 2 VMs, the VM1 is implement the square two of root script one time. VM2 is execute the script twice in the same time. Each of the test will literate for 30 times. The result will analyze in mean, and comparable for each of the machines.

## 4 IMPLEMENTATION

The research is implement on Ubuntu Server 12.04 (64bit), CentOS (64 bit) and Debian 6 (Squeeze 64 bit). The research is use Proxmox VE 2.1, Ubuntu Eucalyptus Faststart 2.0, and Openstack Essex. The script is implement on Maxima 5.28.0, and monitorize on Htop 1.0.2. The server specification are Intel® Xeon™ Processor E3-1220 3.1Ghz, 2GB RAM DDR3 ECC and HD 1TB for test case server. The virtual machine has built on specification VCPU 1 core, 512Mb RAM, and 5GB Hard disk.

### 4.1 Script

The result has built the script *hitungpangkat* to call the maxima script. The script is capture the time, date, into the log file. The

Figure 4 is describe the script.

```
echo "direkam pada =" >> rekampangkat.log;
echo "$(date) =" >> rekampangkat.log;
maxima -b coba.mac >> rekampangkat.log;
echo "selesai pada = $(date)" >> rekampangkat.log;
echo "-----" >> rekampangkat.log
```

Figure 4. Script hitungpangkat

### 4.2 Maxima

The implementation in Maxima, the research is use Maxima version 5. The research built coba.mac. In this script, it is use *fpprec* (floating point precision) that will make limitation until 1.500.000 digit. It is use *bfloat* to make the result is use big float format. *sqrt* have function to calculate square root. Time command (%) is show the execution time that consume on calculation. All the script is implement on maxima script, Figure 5.

```
fpprec:1500000$
bfloat(sqrt(2))$
time(%);
```

Figure 5. Script coba.mac

## 5 RESULT AND DISCUSSION

The section is discuss the implementation result and performance testig result on each machines.

### 5.1 Test Result on OpenStack

Table 1. Test Result on OpenStack

| Test | OpenStack |     |    |                    |
|------|-----------|-----|----|--------------------|
|      | VM1       | VM2 |    |                    |
|      |           | P1  | P2 | Average on P1 & P2 |
| 1    | 14        | 22  | 21 | 21.5               |
| 2    | 11        | 21  | 21 | 21                 |
| 3    | 11        | 19  | 19 | 19                 |
| 4    | 11        | 19  | 19 | 19                 |
| 5    | 11        | 19  | 19 | 19                 |
| 6    | 11        | 19  | 19 | 19                 |
| 7    | 11        | 19  | 19 | 19                 |
| 8    | 11        | 19  | 20 | 19.5               |
| 9    | 11        | 20  | 20 | 20                 |
| 10   | 11        | 18  | 18 | 18                 |
| 11   | 11        | 19  | 19 | 19                 |
| 12   | 11        | 20  | 20 | 20                 |
| 13   | 11        | 19  | 19 | 19                 |
| 14   | 11        | 20  | 20 | 20                 |
| 15   | 11        | 20  | 20 | 20                 |

|         |             |             |             |             |
|---------|-------------|-------------|-------------|-------------|
| 16      | 11          | 20          | 20          | 20          |
| 17      | 11          | 20          | 20          | 20          |
| 18      | 11          | 19          | 19          | 19          |
| 19      | 10          | 19          | 19          | 19          |
| 20      | 10          | 20          | 20          | 20          |
| 21      | 11          | 19          | 20          | 19.5        |
| 22      | 11          | 20          | 19          | 19.5        |
| 23      | 11          | 19          | 20          | 19.5        |
| 24      | 11          | 20          | 19          | 19.5        |
| 25      | 11          | 19          | 19          | 19          |
| 26      | 11          | 19          | 19          | 19          |
| 27      | 11          | 19          | 20          | 19.5        |
| 28      | 11          | 20          | 20          | 20          |
| 29      | 11          | 19          | 20          | 19.5        |
| 30      | 11          | 19          | 20          | 19.5        |
| average | 11.03333333 | 19.46666667 | 19.56666667 | 19.51666667 |

The execution time that has shown in Figure 3, the average is 11.0333333 (Ta1), and Ta2 is 22.06666667 second. The test result it self is 19.51666667. Based on this case, we can seen that the ration on Ta1 and Ta2 is 0.565328779. It is mean that the performance is close to the Ta1/Ta2=0.5. The test result for 30 times shown in Table 1.

### 5.2 Test Result on Proxmox

Table 2. Test Result on Proxmox

| Test | Proxmox |     |    |         |
|------|---------|-----|----|---------|
|      | VM1     | VM2 |    |         |
|      |         | P1  | P2 | Average |
| 1    | 33      | 76  | 78 | 77      |
| 2    | 15      | 31  | 30 | 30.5    |
| 3    | 15      | 30  | 29 | 29.5    |
| 4    | 16      | 29  | 29 | 29      |
| 5    | 16      | 30  | 30 | 30      |
| 6    | 16      | 27  | 28 | 27.5    |
| 7    | 16      | 29  | 28 | 28.5    |
| 8    | 15      | 29  | 29 | 29      |
| 9    | 16      | 29  | 29 | 29      |
| 10   | 16      | 28  | 29 | 28.5    |
| 11   | 15      | 29  | 29 | 29      |
| 12   | 15      | 29  | 30 | 29.5    |
| 13   | 16      | 29  | 29 | 29      |
| 14   | 15      | 29  | 29 | 29      |
| 15   | 16      | 28  | 29 | 28.5    |
| 16   | 16      | 28  | 29 | 28.5    |
| 17   | 16      | 29  | 29 | 29      |
| 18   | 15      | 28  | 28 | 28      |
| 19   | 16      | 29  | 29 | 29      |

|         |             |             |             |             |
|---------|-------------|-------------|-------------|-------------|
| 20      | 15          | 29          | 29          | 29          |
| 21      | 16          | 29          | 29          | 29          |
| 22      | 15          | 28          | 29          | 28.5        |
| 23      | 15          | 29          | 29          | 29          |
| 24      | 16          | 28          | 29          | 28.5        |
| 25      | 16          | 30          | 29          | 29.5        |
| 26      | 16          | 30          | 29          | 29.5        |
| 27      | 15          | 29          | 29          | 29          |
| 28      | 15          | 29          | 28          | 28.5        |
| 29      | 15          | 29          | 29          | 29          |
| 30      | 16          | 30          | 30          | 30          |
| average | 16.13333333 | 30.53333333 | 30.63333333 | 30.58333333 |

|         |             |        |             |             |
|---------|-------------|--------|-------------|-------------|
| 26      | 586         | 1162   | 1160        | 1161        |
| 27      | 650         | 1259   | 1256        | 1257.5      |
| 28      | 593         | 1154   | 1148        | 1151        |
| 29      | 572         | 1146   | 1143        | 1144.5      |
| 30      | 605         | 1174   | 1171        | 1172.5      |
| average | 599.6666667 | 1170.3 | 1170.166667 | 1170.233333 |

The execution time for VM 1 on Eucalyptus is 599.666667, and for VM 2 with 2 process is 1170,233333 in averages.

The execution time in every virtual machine management shown that close to 0.5. It is describe the resource isolation on every virtual machine have same characteristics.

The execution time in Proxmox when run for 1 VM is 16.13333333 that shown in Table 2. The execution time in Proxmox for two process in VM 2 in average is 30.58333333. It is shon that ratio between Ta1 and Ta2 is 0.527520436.

**5.3 Test Result on Eucalyptus**

Table 3. Tes Result on Eucalyptus

| Test | Eucalyptus |      |      |         |
|------|------------|------|------|---------|
|      | VM1        | VM2  |      |         |
|      |            | P1   | P2   | Average |
| 1    | 673        | 1286 | 1281 | 1283.5  |
| 2    | 651        | 1229 | 1225 | 1227    |
| 3    | 717        | 1144 | 1141 | 1142.5  |
| 4    | 592        | 1182 | 1177 | 1179.5  |
| 5    | 638        | 1219 | 1221 | 1220    |
| 6    | 581        | 1156 | 1163 | 1159.5  |
| 7    | 582        | 1148 | 1147 | 1147.5  |
| 8    | 570        | 1136 | 1136 | 1136    |
| 9    | 606        | 1191 | 1189 | 1190    |
| 10   | 576        | 1135 | 1131 | 1133    |
| 11   | 603        | 1174 | 1176 | 1175    |
| 12   | 581        | 1165 | 1166 | 1165.5  |
| 13   | 622        | 1183 | 1195 | 1189    |
| 14   | 593        | 1166 | 1168 | 1167    |
| 15   | 592        | 1193 | 1188 | 1190.5  |
| 16   | 578        | 1148 | 1150 | 1149    |
| 17   | 544        | 1149 | 1157 | 1153    |
| 18   | 594        | 1154 | 1159 | 1156.5  |
| 19   | 574        | 1145 | 1145 | 1145    |
| 20   | 621        | 1195 | 1192 | 1193.5  |
| 21   | 575        | 1145 | 1141 | 1143    |
| 22   | 593        | 1161 | 1163 | 1162    |
| 23   | 587        | 1146 | 1146 | 1146    |
| 24   | 578        | 1143 | 1145 | 1144    |
| 25   | 563        | 1121 | 1125 | 1123    |

**6 CONCLUSION AND FUTURE WORK**

The research have several conclusion, first is Eucalyptus have better performance on resource isolation than Proxmox and Openstack. Eucalyptus have ratio 0.512433418 (Ta1/Ta2), Proxmox have ratio 0.52751920436 and openstack have ratio 0.565328779. The performance is looking for close to 0.5.

In the computation test, Eucalyptus have time execution average is 599.666667 second, Proxmox have average is 30.58333333, and Openstack have average is 19.5167777777. It has shown that Openstack have execution time more faster than others.

Based on the result, we conclude that for performance of resource isolation Eucalyptus have better characteristics on it, but for execution time Openstack is more faster than others.

**ACKNOWLEDGMENT**

The authors wish to thank the Computer System and Network Laboratory, Department of Computer Sciences and Electronics, Faculty of Mathematic and Natural Sciences, Universitas Gadjah Mada, Indonesia. Gamabox supercomputer team for benchmark facilities.

**REFERENCES**

- [1] Locus Cloud Service., n.d, *Perbedaan Antara Virtualisasi an Cloud Computing* <http://www.locus.co.id/?pg=13>, access at 25 Januari 2013
- [2] Xen Project, <http://www.xenproject.org/developers/teams/hypervisor.html>, access at December 1, 2013
- [3] Foley, Samantha , et al., 2007, *benchmarking using virtual machine, paper*, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.9186>, access at 26 januari 2013.
- [4] Openstack Indonesia., <http://www.openstack-id.org/home>, access at 26 Januari 2013.
- [5] Proxmox , n.d, *Proxmox Virtual Environment*, <http://proxmox.com/products/proxmox-ve/features>, access at 26 Januari 2013
- [6] Quétier, Benjamin. Neri, Vincent. Cappello, Franck. 2007, *Scalability Comparison of Four Host Virtualization Tools*, Journal of Grid Computing, Vol. 5 pp. 83-98.

- [7] VMWare.,  
[http://www.vmware.com/files/pdf/VMware\\_paravirtualization.pdf](http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf). access at December 5, 2013

IJSER