Optimizing IT Infrastructure by Virtualization Approach

To cite this article: Thomas Budiman and Jarot S. Surosso 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **190** 012012

View the [article online](https://iopscience.iop.org/article/10.1088/1757-899X/190/1/012012) for updates and enhancements.
OPTIMIZING IT INFRASTRUCTURE BY VIRTUALIZATION APPROACH

Thomas Budiman¹, Jarot S. Suroso²

¹,²Graduate Program, Binas University, Jl. Kebon Jeruk Raya No. 27 Jakarta Barat 11530, Indonesia

thomas@rehobot.net¹, jsembodo@binus.edu²

Abstract. The goal of this paper is to get the best potential configuration which can be applied to a physical server without compromising service performance for the clients. Data were compiled by direct observation in the data center observed. Data was then analyzed using the hermeneutics approach to understand the condition by textual data gathered understanding. The results would be the best configuration for a physical server which contains several virtual machines logically separated by its functions. It can be concluded that indeed one physical server machine can be optimized using virtualization so that it may deliver the peak performance of the machine itself and the impact are throughout the organization.

1. Introduction

Virtualization is a way of using computer resources in such a way that no waste of computing resources during the idle time. Of course, it is also an active support to the Green Computing movement. The Green computing is the study and practices of using the computer resources efficiently [1].

Align with the movement, an organization may apply virtualization to either it’s infrastructure or the software being used. Both type not only support the green computing movement, but also organization may gain some advantages by implementing virtualization. In the computing environment, there are two types of virtualization that commonly applied: hypervisor (kernel-based virtualization, ie. KVM) and paravirtualization/container based virtualization (ie. LXC or Linux Container). Both techniques has its own advantages and disadvantages [2].

The kernel based virtualization allows us to deploy multi operation system as a guest on top of the host machine, while in container based, the virtual machine will inherit host operating system, hence the guest operating system will be the same as the host machine.

This research has the goal instead of hosting solution to self-managed data center to fulfil the need of organization. The virtualization adoption to their data center, leads the organization to support the Green Computing movement. The purpose of research is to get a better understanding of how a physical machine potential is optimized by implementing the virtualization technique on top of it. It is true that relevant papers about virtualization and green computing has given the author a wide perspective of how this research are done.
2. Theory and Method
In today’s dynamic age, businesses change rapidly and act to competition, hence requirements tend to change within projects. In such scenarios, costs of change can be overbearing [3].

Based on the survey results application development projects undertaken by the Standish Group in 2004 up to 2012, the average successful project (the project has been completed on-time and on-budget, with all features and functions as initially specified) ranges between 29% up to 39%. As for the unsuccessful project (The project was cancelled at some point during the development cycle) ranges from 18% to 24%. For projects that challenged (The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified) had a greater percentage ranging from 43% to 53% [4].

Scrum project management is one of the agile methodology that can be used as solutions in application development projects [4]. Scrum is a method or process comes from the concept of agile which is currently very popular used by companies in the field of industry or application software development around the world.

Scrum project management is a project management methodology that has higher adaptability to the changes that occur in each of its elements than waterfall method. The advantages of agile project management and particularly the Scrum-based approach is its simplicity. Within an agile project, roles are clearly defined and do not cross boundaries. Features can be completely developed and tested in short iteration cycles. Because each team members bears major responsibility for their part of the project,
ownership of the project is more broadly based. The methods of agile project management enforce extensive communication, which helps teams organize more effectively. This, ultimately, can lead to greater productivity for everyone involved [6]. SCRUM is best if the requirements frequently change [7].

![Scrum Framework](image)

**Figure 3. Scrum Framework [9].**

The heart of Scrum is a Sprint, a time limit for a month or less, with an increment of "Done", function, potentially to be released and developed. Sprint typically have a duration that is consistent throughout the product development process. Sprint's new, begins immediately after the previous one ends Sprint. Sprint load and consist of Sprint Planning, Daily Scrum, development, Sprint Review and Sprint Retrospective.  Sprint's new, begins immediately after the previous one ends Sprint. Sprint load and consist of Sprint Planning, Daily Scrum, development, Sprint Review and Sprint Retrospective [8].

There are several key roles in Scrum framework: 1) Stakeholders: the mission owners, possess the idea about why to build, what to build, and how processes should be; 2) Product owner: the product development owner, closely working with stakeholders, creates a prioritized wish list stories called a Product Backlog; 3) Scrum Master: the facilitator, closely working with product owner, makes sure the stories in the product backlog will successfully delivered as workable sub-products; 4) Scrum Team: the developers, closely working with Scrum Master, deliver workable sub-products according to the product backlog [9].

Scrum main characteristics may be summarized in [10]:
1. Software development is carried out through iterations, named Sprints, which last for 30 days. The result of each one of these is a feasible increase that is shown to the client.
2. Meetings all along the Project; it is noteworthy that the development team meets daily and briefly, only for 15 minutes, for coordination and integration.

The fundamental differences between traditional and agile concept is the traditional concepts, solutions and requirements (requirement) is very clearly defined early in the project so a large scope change was not expected. Traditional concept of work on the project on a regular basis and can be repeated by using a template that has been created. On the concept of agile solutions and the need (requirement) partially known and understood so that there is a possibility of additional features that were previously unknown to the project team. The concept of agile project teams preparing for big changes in the scope [11].
3. Method

When we choose to implement virtualization, we have only two options to choose, a full virtualization as known as Hypervisor, or a paravirtualization known as container based. Either one is very mature technology and widely used around the globe. In full virtualization, the guest machines might have a different operating system to use from the host machine, whereas no option in the paravirtualization. In paravirtualization the guest machines will inherits the host operating system. They will be configured as a block of user space (or container) to emulate the hardware limit. The following diagrams shows how these two are different from each other:

**Figure 4. The Abstraction of a Hypervisor Virtualization**

In hypervisor world, the virtual machine might have different operating system to the host machine, hence giving the flexibility to choose which OS to be deployed on the VM. It is possible because of the presence of the hypervisor abstraction layer, which mainly provides hardware emulation.

**Figure 5. The Abstraction of a Paravirtualization**

The paravirtualization a bit more simple than full virtualization, because they no longer employed the hypervisor abstraction layer. Because of this, the speed of hardware is nearly physical alike. The only drawback in the paravirtualization appeared that the guest OS will inherits what's available on the host, hence the guest OS will be the same as the host.

At some point the KVM might shows performance ahead of the paravirtualization, but in some cases the results shows differently where paravirtualization performance outperform KVM’s.

The USE method are used in this research to help in setting up the virtual machines. It allows the authors to get step by step diagnose work for performance monitoring while converting and creating the virtual machines on the physical server [12]. This technique was introduced by Brendan Gregg, the performance engineer specialist or system performance engineer, a consultant and a trainer. He introduced the step of his method by the diagram in Figure 6.

All steps in the USE flowchart employed two built in linux command to measure the system health. Those two commands are: `vmstat` and `top`. VMSTAT were used to monitor the health of the storage health, while the TOP command helps to monitor the processor health and utilizations. Figure 4 and 5 below displays the results of `vmstat` and `top` command.
Figure 6. The USE Flowchart

Figure 7. Output from top command

```
root@proxmox:~# vmstat
procs -------- memory -------- ---swap-- -----io--- system-- ----cpu----
 r b swpd free buff cache si so bi bo in cs us sy id wa
 1 0 0 4211732 78708 1304532 0 0 18 16 74 81 1 0 99 0
```

Figure 8. Output from vmstat command

For more information about the results from both top and vmstat command, the Linux man page site serves full linux commands documentations: www.linux.die.net. To get feedback from user who
engaged to one of the machine served by the virtual machine, a survey was set and deployed to the website. Further calculation of the questionnaire are using the LIKERT scale with the following scale:

| Statement       | Scale |
|-----------------|-------|
| Very Good       | 5     |
| Good            | 4     |
| Good Enough     | 3     |
| Fair            | 2     |
| Not Good        | 1     |

4. Result and Discussion

The first two tasks were converting two websites that were running in the hosting environment to the proxmox environment. The process smoothly done by copying all required files from hosting account to the local web server, and also the databases. The other virtual machines were created to suit the organization need. By the end of the research, 6 out of 7 virtual machines were employed in the physical server. The final results shown in the following table:

| VM Configuration Summary |
|--------------------------|
| VM # | HDD | RAM | Core |
|------|-----|-----|------|
| 1    | 16 GB | 2GB | 2    |
| 2    | 8GB  | 4GB | 1    |
| 3    | 150GB | 4GB | 2    |
| 4    | 8GB  | 4GB | 2    |
| 5    | 10GB | 1GB | 4    |
| 6    | 10GB | 1GB | 1    |
| 7    | 4GB  | 1GB | 1    |
| TOTAL | 17GB | 13 |

The total configured memory and CPU cores were beyond the physically available. The physical available RAM were 8GB while CPU core were 12. Running both `vmstat` and `top` command every time a virtual machine were created to monitor the server health, produced results on Table 3.

| Server Health summary with 7 VMs |
|---------------------------------|
| VM | CPU Utilization | Load Average | Boot | Disk IO |
|----|-----------------|--------------|------|---------|
| 1  | 99.9            | 0.08         | 0.04 | 0.06    | 27    | 23 |
| 2  | 99.9            | 0.02         | 0.02 | 0.00    | 22    | 18 |
| 3  | 99.7            | 0.36         | 0.17 | 0.06    | 20    | 17 |
| 4  | 99.4            | 0.04         | 0.13 | 0.12    | 18    | 16 |
| 5  | 99.4            | 0.04         | 0.10 | 0.13    | 18    | 16 |
| 6  | 97.6            | 0.26         | 0.41 | 1.08    | 16    | 24 |
| 7  | 98.6            | 0.51         | 0.70 | 0.53    | 16    | 25 |
Saturation occurred when the virtual machine had over the limit of totally available.cpu.core. To avoid further errors to the server, the author had decided to stop the last virtual machine. From the survey results, it is considered no performance issues arise even when we run several virtual machines in one physical server.

5. Conclusion
The linux container applied in this research is truly helpful to the optimization of the physical server. One point should be noted that, even though it is true that guest machines could be configured beyond the physical resources limit on the host, it is better to keep the total configured virtual machine below the limit especially for the processor core. The fact that were found when this limit is ignored, had lead the host machine to suffer some of the saturations and IO waits that may leads to further errors if forced to run for a long period. To avoid this, the author(s) had decided to stop at the limit of 6 virtual machines configured and deployed.

References
[1] S. Aggarwal, M. Garg, and P. Kumar, “Green Computing is SMART COMPUTING – A Survey,” Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 2, pp. 297–303, 2012.
[2] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, “An Updated Performance Comparison of Virtual Machines and Linux Containers,” Technology, vol. 25482, pp. 171–172, 2014.
[3] A. Jha, Agile in the IT World, KMPG, 2011.
[4] The Standish Group, "Chaos Manifesto," The Standish Group, 2012.
[5] Versionone.com, "The 10th Annual State of Agile Report," VersionOne.com, 2016.
[6] C. H. Frank, "Understanding agile project management methods using Scrum," International digital library, 2010.
[7] L. Rich C., "The Success Factors of Running Scrum: A Qualitative Perspective," Journal of Software Engineering and Applications, 2012.
[8] K. Schwaber and J. Sutherland, "Scrum Guide," 2014.
[9] L. Rich C., "The Success Factors of Running Scrum: A Qualitative Perspective," Journal of Software Engineering and Applications, 2012.
[10] W. G. Barrios and C. T. Zarrabeitia, "SCRUM: Application Experience in a Software Development PyME in the NEA," JCS&T, p. Vol. 12 No. 3, 2012.
[11] S. Aljaž, " Agile Project Management – a Future Approach to the Management of Projects?," Dynamic Relationships Management Journal, 2013.
[12] B. Gregg, Systems Performance - Enterprise and the Cloud. Prentice Hall, 2014.