클라우드 환경을 통한 확장 가능한 플랫폼 초기 상태의 자동 구성

고메즈 마우리시오*, 가레라 베르니**, 그로게르 기예르모*, 밤복흥*, 이슬람*, 허의남*
*경희대학교 컴퓨터공학과, **경희대학교 산업공학과
e-mail : mgomez@khu.ac.kr

Autoconfiguration of Initial State of a Scalable Platform Over Cloud Environment

Mauricio Alejandro Gomez Morales*, Berny Alfonso Carrera Gordillo**,
Guillermo Crocker García*, Pham Phuoc Hung*, Md. Motaharul Islam*, Eui-Nam Huh*
*Department of Computer Engineering, **Department of Industrial Engineering
Kyung Hee University

Abstract

Companies that have high need of some computational resources for an specific space of time, nowadays, the pay as they use manner for resources required, that, is a good solution provided these days by some Cloud computing providers. Also solutions that represent a distributed computation for processes with high demand of calculation have appeared lately, the only problem is that when they are created, they need also be configured to share the same working space, this is the scope that comprehends this work, where the aim is to propose a framework that can be used as a solution of automation of the configurations that sometimes can take undetermined time and sometimes the user that configures it has to have a lot of knowledge and also configurations can turn in tricky ones generating a delay in the time where real productivity should be exploited.

1. Introduction

Not long time ago, companies or institutions that needed to run big batch processes were used that them would last for a range of time between about hours or even half a day, that's the reason why those companies always have settled the time to run this kind of methods the last day of the month and usually also at night, but as we are in information era, then the need of updated information becomes more and more important day by day, almost to the limit of instant online information. In the other hand, as for the small companies the acquisition of huge platform in their farm server represents a big expenditure, and also for the big companies that represents unused resources and also a waste of money to maintain that big platform within their server room.

Cloud computing was designed for taking different resources of computing as a utility, and deliver them based on the users requirements resting importance to the fact of where the services are hosted. The key is the ubiquity that shares with various types of networking, storage devices and internet software. In addition, providers such as Microsoft, Google, IBM, Amazon and Salesforce have their established hosting for cloud computing applications such as social networking, business applications, gaming portals, etc. managed in different locations of the global world.

Through the use of virtualization, clouds help and hold a good promise for the performance of the companies sharing a set of physical resources that can be used for different needs, companies that has high need of some computational resources for some specific space of time, the pay as you use the resources required, is a good solution provided these days by the Cloud computing providers.

Clouds promise to be a cheap alternative to access to supercomputer and specialized cluster, this, represent distributed computations for processes with high demand of calculation have appeared day by day, making it an on-demand computing.

Hadoop is an open source project that was being used at petabyte scale and provided scalability using map-reduce programming model for a way to perform data-intensive computation on commodity of cluster computers [1] [2].

In this paper we propose a solution that had taken advantage of different tools to solve the problem when a parallel processing job is required to give big computation for a method that is commonly thought as is taking too much time to give an answer, or if the company owns the sufficient platform to give answer to that kind of request, then the company would have invested a huge amount of money to have that kind of platform in its own farm server. The real issue of this kind of solutions that already exist in the market is that still needs some personal configuration that will consume time and will also require specific knowledge from the user that will make the execution of the program. Remember that this problem becomes bigger according to the number of VMs that are required by the end user.

The main contribution of this paper is that we provide a model and a solution to the semi-automatic configuration of hadoop, the scalable platform, over cloud environment, that is XenServer taking advantage of small parts of solutions that already exist in the market.

The rest of this paper is organized as follows. First, in section number two, we take some consideration of the work that has already been done related to the scope of work of this paper. After that in section number three, we talk a little in detail about the problem definition to let the reader understand what aspects we are focusing to solve. Then, in chapter four, we place our work in context with the existing work in this area,

showing what our solution includes. Finally, we present our conclusions and future work.

2. Related Work

There are some works that have been done around the topic that covers this paper, given the necessity of users to run big processes that give results as soon as possible and trying always to give the enterprises the possibility to pay less for that. The works that had been done can be classified in two categories.

The first category for this kind of investigation is benchmarking, in other words, to make an evaluation of different environments of implementation of the platform of Hadoop over XenServer, in [3] they evaluate and provision a multi-domain Networked Clouds for Hadoop-based applications, where their evaluation includes issues for Hadoop aplications in the setting of built-to-order Hadoop instances on networked clouds, that includes presentations of new algorithms and mechanisms, using ORCA, they evaluate conditions, compare multi-cloud with single-cloud deployments of the Hadoop Distributed Filse System (HDFS). In the case of [4] where they evaluate different load charges in a new solution denominated by them as WasElephant that tunes various Hadoop parameters which affects the system performance, also they go to the granularity of simulation in job execution, also compares the job scheduling policies and simulates a large-scale Hadoop cluster to carry out different performance test at a low cost. So, the scope of these works differ in the way that our work is not making any comparison or evaluation, but may be a field of further study.

The second category is configuration where [5] talks about live configuration, that will measure the performance during execution time and will make the needed changes to improve it while the process is being carrying out, this is made with the help of history logs, then it analyzes the information that continue to extract relevant information for performance to finish using that information to construct a hypothetical job based that will be proposed based on Grid Hill Climbing algorithm to search for the optimal configuration given that Hadoop has about 20 parameters to be optimized. The difference of this paper and our own is that they work in the aspect of optimizing the live performance of the platform, on the other hand, our scope of work goes no further than the initial configuration of the environment.

Also, in they also work in the field of configuration and also a part of their job is in the same spectrum than ours, where they defined an auto-deployer that is called by Web Services and this is in charge of fetching the information of request, create XML configuration and pass it to the Job Manager also fetch the IP of each VM and the authentication information through VM Management adapter to finish making a remote running of some scripts to put on a working state the VMs. In [6] they were also focusing in the same working space, where they even mention that they used the same tools, and also worked in a second field where the whole platform can be destroyed when the end user wants to. There are two differences of within the jobs mentioned and ours, where the first variance is that they worked in customizable characteristics for the VMs that are going to be requested, but we create the VM just as the image was created. The other difference is that they didn't make the SSH connection configuration in an automatically way as we did have in our own program.

3. Problem Definition

In this section, we mention our assumptions in the problem addressed in this work that is to make the auto-configuration of the VMs required by the user, we also define the problem, and then detail our solution based on dynamic configuration of all the virtual machines master/slaves including the creation of VMs and the respective configuration.

A. Assumptions

For the problem definition we made the following assumptions for the roles that are associated in the complete process shown in Table 1 are as follows:

Table 1 : Roles in application

Name	Roles	
	Description	***Var***
End User	Is the one that places the request of hadoop platform in an already customized form, indicating only the number of VMs that are needed. Usually, this user is a programmer or a user that is in charge of running the procces that will consume a lot of computation, memory or network bandwidth.	EU
Hadoop Service Provider	Is the company that will provide the service of renting determined number of VMs, this kind of role is classified as SaaS.	HSP
Cloud Service Provider	Will give a service of a PaaS, that will provide the VMs that will be provisioned with the previously configured image.	CSP

In order to have a better understanding, the same variables are shown in the next Figure 1, where not continues lines shows the data flow that follows the process to provide the Hadoop environment with the number of VMs that EU will demand to HSP, and this will rent these resourses to the CSP. Also, the direction of each arrow, in the same Figure, shows the one that is the client and the service provider for each of the two different sub processes that are taking place for our problem diagram. The reason why is divided by two different sub processes is because of the term of value chain, that is not part of study of this paper, but is planned to be part of a further work.

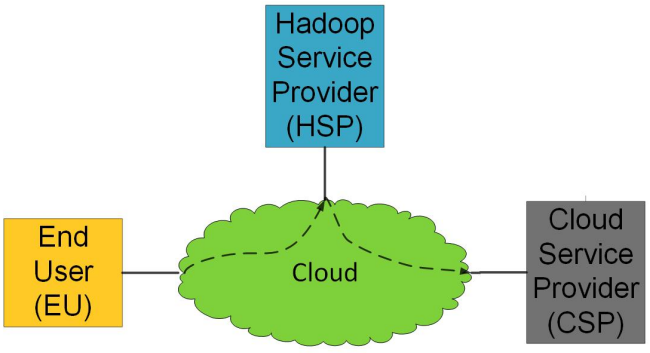

Figure 1 : Request / Problem Flowchart

So, if we see a little deep in this case, the problem is that when you get the VMs that already have installed Hadoop, have to be configured, and then, the programmers of the requesting company have to deal, determined time to make this configuration and also, the programmer should be well

informed to know where and what to change to make the specific VM work in the whole platform, and this is where the problem gets worse.

4. Proposed Solution

The proposed solution is an architecture model, the one that is shown in Figure 1 and a simple process that had taken advantage of some existing utilities, that will be explained in the next steps; nevertheless, it is the result of a hard investigation time invested to get the final outcome that we are going to show in the next four steps.

B. Tools

In this section we want to mention all the tools that are related somehow with the solution presented in this work, as follows.

Table 2 : Tools Used

Name	Tool	
	Description	***Step***
XenServer	Hypervisor that controls all the virtual machines environment.	All
Ubuntu	Host operating system, already installed in the created image, and also is the operating system that uses XenCenter.	All
Hadoop	Parallel processing platform that works over Ubuntu, that is already installed in the created image.	All
XenCenter API	Development environment that allows to user some code to make automatically the creation of VMs.	1,2
C#.Net	Programming language that was used to access de XenCenter API and clone the VMs and start those machines.	1,2
Java Netbeans	Programming language that was used to create the .jar programs that are going to be ran when configuring the VMs.	3,4
Expect	Application that runs over Ubuntu and allows to simulate the user confirmation in some terminal commands.	3,4

C. Process Steps

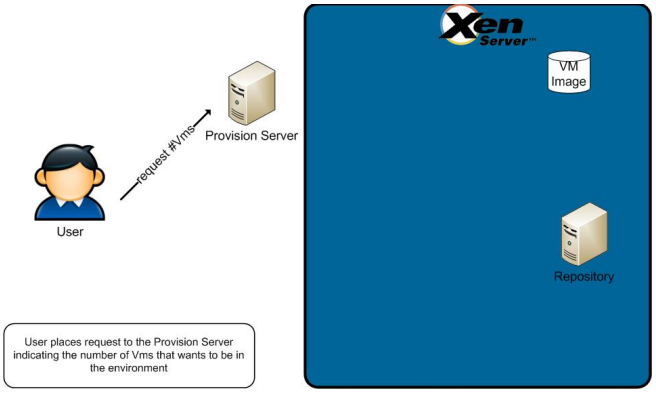

Figure 2: First Step

In Figure 2 is shown where the end user requests a determined number of VMs to the Provision Server that is the HSP described before. This program was developed in C#.Net and starts requiring the number of VMs to EU, this is how using the Xen Server API can be cloned all the machines according to the image that has to be already generated in the CSP space.

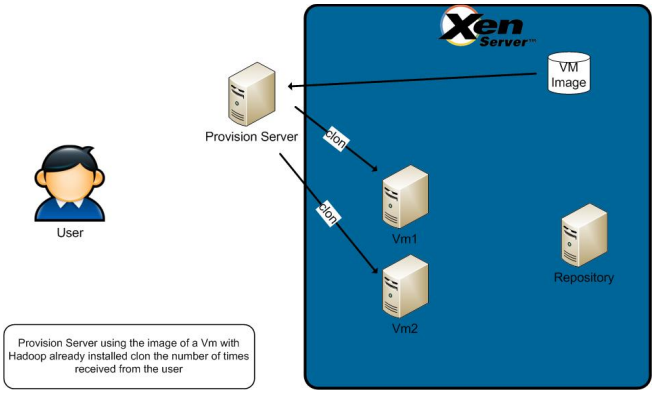

Figure 3 : Second Step

As mentioned before, in the second step, as is seen in Figure 3, this HSP use the VM image that already have the Hadoop installed to create automatically the VMs with a program created in C#.Net. After generating all the machines this program also makes a forced restart of all the machines, to be let the EU to run the specific programs that will make the respective configuration.

As was stated previously, the image stored and already configured should be prepared with Ubuntu as Operating System, Hadoop installed and configured as a stand-alone implementation, Expect application, and all the other programs that are needed for this programs, already mentioned in this paragraph, could work in a proper way.

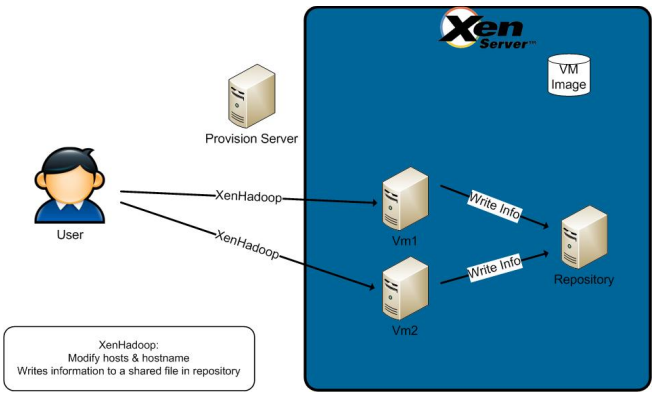

Figure 4 : Third Step

In Figure 4, the third step is represented as the program that was developed in Java Netbeans, to be run in each of the machines, this process has to be ran by EU, but the username only have to run “XenHadoop” without any parameter, and this program what makes is that it goes to write to a shared file the repository server, that holds the information in the meantime of configuration. The problems faced in this program were that to obtain some configurations of each VM was necessary to make a program that was going to take advantage of some Commands of the Operating System, for example to change the name of the machine and get the IP automatically assigned by Xen Server.

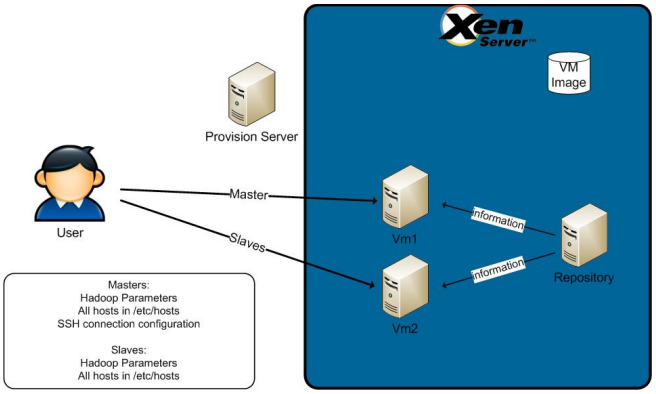

Figure 5 : Fourth Step

Figure 5 shows the last step that has to be executed in each of the machines, the only difference is that only in the master has to be the one "/runmaster" and in the all the slaves has to be "/runslave". And the order is first all the slaves and at the end the master. The reason for this is that the program of master has already the steps to turn on the hadoop, so, the programmer will only have to upload the parallel program that was the objective to demand that Hadoop platform. In this program, that was made also in Java Netbeans some problems more difficult were faced, given that to change some configurations was required to make the simulate that the end user was also confirming by one "Yes" or so, this is how in the image was also needed to have installed some applications that allows or cover this issue in the form that you assign some time to wait for the response and an specific character sequence, then it writes the next command structure as is determined in the program that was developed for this matter.

5. Conclusions and Future Work

Some tools already exists nowadays in the market called Cloud, and they are really good to let end users use and rent the resources that they want to solve any of the problems of computation, memory or network bandwidth, still some of those tools can be improved with small ideas that lead to a better performance or tuning or configuration, this is how, after running various times our project that includes an architecture diagram and a group of programs to give solution to the problem of making the creation and configuration of Hadoop in an environment a specified number of VMs determined by the end user, we can conclude that the proposed solution works well, and it can be applied to some other applications that need some live configuration or while is still starting each of the VMs.

As future work we are planning to convert it completely in an automatic solution, given that now is still three steps that are needed to be ran by the end user in each of the machines. To be able to accomplish this future work we are planning to use webservices to communicate and with this the solution could be completely transparent to the end user.

Also, as a future work we would like to focus to propose a model that would separate each level of the platform to determine in a chain value how each part of the solution are profitable or not, or also to determine the price that would be best to apply to a model like this.

6. Acknowledgment

This work was supported a grant from the NIPA(National IT Industry Promotion Agency) in 2013. (Global IT Talents Program). The corresponding author is Eui-Nam Huh.

References

[1] J. Dean. and S. Ghemawat., "Mapreduce: simplified data processing on large clusters.," in *OSDI'04: Proceedings of the 6th Symposium on Opearting Systems Design & Implementation, pages 10-10,*, 2004..

[2] J. Dean and S. Ghemawat, "Mapreduce: a flexible data processing tool.," *Commun. ACM,* vol. 1, no. 53, pp. 72-77, 2010.

[3] A. Mandal, Y. Xin, I. Baldine, J. Chase and V. Orlikowski, "Provisioning and Evaluating Multi-domain Networked Clouds for Hadoop-based Applications," in *IEEE International Conference on Coud Computing Technology and Science*, Athens, Greece, 2011.

[4] Z. Ren, Z. Liu, X. Xu, J. Wan, W. Shi and M. Zhou, "WaxElephant: A Realistic Hadoop Simulator for Parameters Tuning and Scalability Analysis," in *ChinaGrid Annual Conference*, Beijing, China, 2012.

[5] K. X. L. a. W. T. Wang, "Predator—An experience guided configuration optimizer for Hadoop MapReduce.," in *Cloud Computing Technology and Science (CloudCom)*, 2012 IEEE 4th International Conference on. IEEE, 2012.

[6] M. Hong, Z. Zhenzhong, Z. Bin, X. Limin and R. Li, "Towards Deploying Elastic Hadoop in the Cloud," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Beijing, China, 2011.