Publisher-Subscriber Based Architecture for Configuration of Heterogeneous Cloud Systems

Tasawer Nawaz[1], M. Junaid Arshad[2] and Ahsan Nazir Raja[3]

[1,2,3]Department of computer science engineering, University of Engineering and Technology, Lahore, Pakistan

***Abstract*— Cloud provides quality of services; these services may be hardware, software or any other product. Clouds systems have advantages, like backup and security, easy access and cost efficient. Cloud users face problem of managing large number of cloud machines. Single project may be span over single instance to more than hundred or even thousands of cloud instances. May be as single project have separate type of machine for different tasks like web hosting, request handling, business logic, cache and database storage. The expansion of projects leads to problem of expansion of instances with same set of configurations. There are some existing methods for configuration with limitation. We proposed new architecture based on publisher-subscriber modal, which is known as "pub-sub". Publisher will be the server where we will upload latest configuration scripts. Subscriber machines will get notification from server, download script and will run it for maintaining same state with sibling machines.**

***Keywords*— Publisher, Subscriber, Script, Workstation and Pub-Sub**

I. INTRODUCTION

Clouds systems are often used for web hosting and data storage nowadays. A single project may span on multiple number of instances. Each of these tasks like web hosting, request handling, business logic, cache storage and database storage might be done on different cloud machines. This expansion of projects leads to problem for project owner, as the need to expand their number of instance in less time. They need to configure and make installation similar as their existing ones.

Cloud systems are based on combining physically resources and dividing them logically. Main reasons behind the idea of cloud are providing users a facility of "pay as you go". Any user who requires some resources he can barrow resource from cloud for limited time as he need, with cloud systems no one need to buy his own resources. Cloud systems help users to get services of all kind of resources like follows [1]:

- PaaS: Product as a service
- Saas: Software as a service
- HaaS: Hardware as a service
- IaaS: Infrastructure as a service
- DaaS: Data as a service

Main goal of cloud systems is to provide secured, dynamic, customized environment to its end users. With the help of cloud systems users will pay attention only on writing business logics, and no need to reinvent the wheel.

Although it's true that cloud computing makes easy to work of software projects and cloud users are not supposed to manage or administrate their servers purchased from cloud providers. Along with these facilities provided recently there are many issues that need to be addressed. These issues are related to security, reliability and effectively usage. But having large number of instance is also an issue for maintenance and managing in the sense of installing or updating required software packages, configurations and security measures. People often don't want to save their critical data on external resources because of security and reliability. Because they cannot monitor systems be themselves [3].

Fig. 1: An overview of cloud computing architecture

Fig. 1 shows the basic architecture of cloud computing. It shows the different layers of cloud systems and shows the components at these layers. It shows that cloud is interacting with different kind of devices and contains different type of servers to provide services to their customers.

Configuring any machine takes time and cost depends on type machine and type of configurations. For all machines we require a person who can manage them and install new software packages and update them. If we have a web app hosted on one cloud machine, our database and web app is one the same machine. After some time due to increment of

users we decided to add one machine and separate both database and web app. After some time we realize that we need to add more machine. In this case we require repeating all process what we had done for our 1st machine. We will install all software packages required before hosting our app.

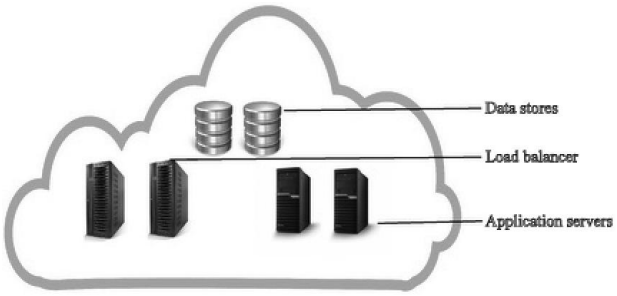

Fig. 2: Web application with different layers

Fig. 2 shows that an application hosted on cloud system contain different layers of systems including servers for database, web hosting, load balancing, cache etc.

In this research we are going to explain that how we can overcome our efforts for installing and updating software packages or configuration and settings. Previously we have two approaches "push" and "pull". In "Push architecture" we can save our time and effort by creating a script file or may be other set of rules of one machine called workstation and push it to all our related machines [9]. In "Pull architecture" we create script on workstation from where we manually download script file and can run on all machines [9].

In next section we have studied existing procedure of configuring cloud systems in detail, other than that we have studied previous papers related to this area.

In section 3 we will propose solution that how we can actually fast the process of configuring cloud instances. It will be used combination of both "pull" and "push" and will work as "publisher-subscriber" (pub-sub) architecture.

There are some existing methods for doing this, which are not fully automatically, and they are only for homogenous systems not for different kinds of systems like window server, Linux and Mac etc.

In this paper we have proposed a new architecture, which is based on pub-sub model. We have proved effectiveness of this architecture through case study and compare results of pervious method with our method.

Section 2 contains literature review that we have studied before purposing new solution. In section 3 we have proposed solution and components of pub-sub architecture. In next section 4 contains results and discussions.

II. LITERATURE REVIEW

We have studied in [1] that cloud computing was start in late of 2007, it provides IT infrastructures, computing environments and configurable software services. Cloud computing can be described as SaaS, HaaS, DaaS.

To guarantee the vision of Cloud Computing quality of service goals between both Cloud provider and their customers has to be met. This so-called Service Level Agreement (SLA) enactment requires little human resource that can interact in order to make system efficient and effective [2].

It is stated in [3] that people don't want to save their critical data to external resources. Main reason is that people often hesitate to locate data on external resources because security and monitoring is not in their control.

Trust on provider is starting point for any kind of customers. For cloud provider it is basic step that they can keep their client trusted and provide them a better way for effective and efficient use of cloud systems. Clouds systems needs the same management commands and mechanism available as users have locally so that they can manage their purchased instances and can use them efficiently [4].

Infrastructure as service (IasS) is one the main reasons behind the popularity of cloud systems. In [5] they have focused on the betterment of architecture so that users can easily use and monitor it from remote locations through console and GUI provided by cloud providers.

In [6] it is stated that with using cloud computing we are facing these problems as well; Portability issues and No standards for clouds.

It is stated in [7] that with the help of cloud we can virtualize our whole architecture like hosting server, operating system and storage. One of the major issue in future is computing with less consumption of power. With virtualization, not only flexibility, we can also get many advantages like security, scalability and efficient use of resources.

In the work [8], they propose a QoS-Aware Resource Elasticity (QRE) framework that allows cloud providers to take a review of behavior of the application and mechanisms of development that ensure scalability of resources those hosts the applications.

A green power management scheme is proposed [9] to determine how many physical machines should be run or turned off at any particular time, because in cloud we add machine physically and divide them logically its very challenging task to distribute in effective and efficient manners.

Since cloud technologies are rapidly increasing in last few years. The increment also increases the complexities of the cloud infrastructure, which leads to security and monitoring problems [10].

After studying different previous search we conclude that there must be a way to use cloud systems efficiently and economically. Users need to an easy to way to configure and monitor their purchased cloud systems.

III. PURPOSED PUB-SUB ARCHITECTURE

Rapid increment in usage of cloud systems for application deployment and data storage increases many problems like security, reliability, managing multiple instances etc. Along other problems one problem persists which is how we can use

efficiently and effectively cloud machines that we have borrowed from cloud providers. There are different solutions are provided in past to resolve this problem but all of them have some downside effects. Main problem is that they are not for same kind of machines and do not provide support for heterogeneous systems. Previously mainly we have two different kinds of structures [7] one is "push" and other is "pull".

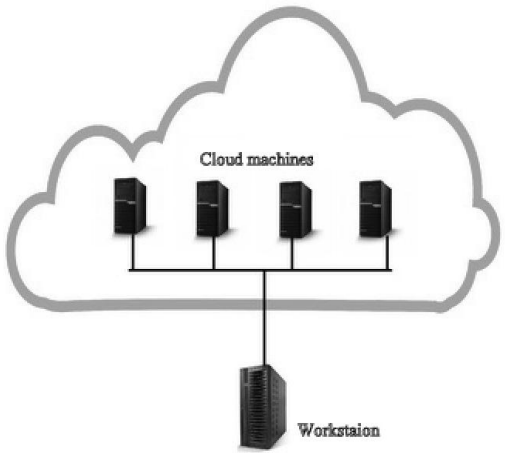

Fig. 3: Representation of push architecture

Fig. 3 is showing push structure. It describes how we can save our time by creating a script file or may be other set of rules of one machine called workstation and push it to all our related machines.

In Fig. 4 we have showed that how pull model works, we create script on workstation from where we manually download script file and can run on all machines.

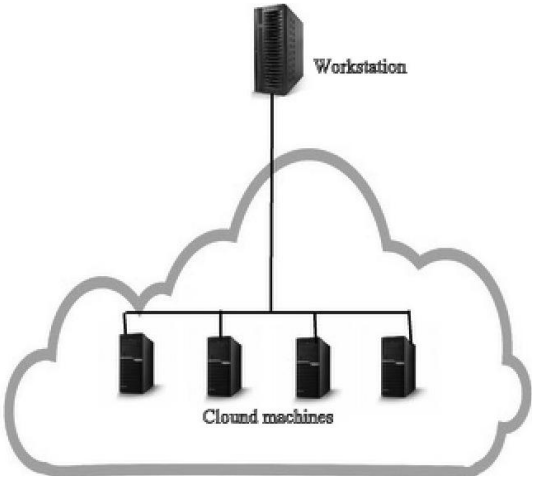

Fig. 4: Representation of pull architecture

Solution we have proposed is required one workstation, that machine used for creating or updating our script. Script will be published and machines can subscribe different scripts. Whenever there is something new or update on server, it will further send notification to all machines those have subscribed that there is something new or update available. Our machines will pull script, which will auto run and perform operations required.

In this research we have proposed a solution that will overcome these problems and will help out cloud users so that they can easily install and update required packages on their barrowed machines from cloud providers. We are going to propose solution, which will have following main components Fig. 5.

- Workstation
- Server
- Subscribed machines.
- Script

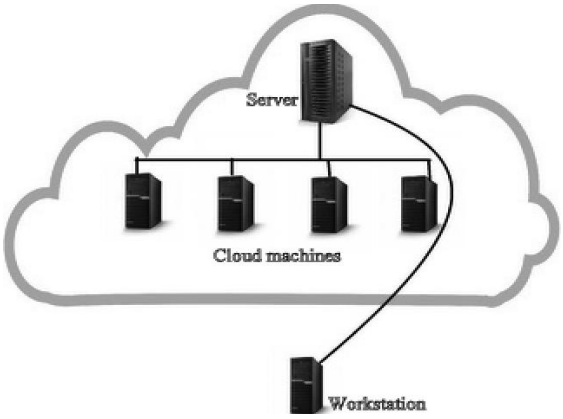

Fig. 5: Represents Pub-Sub Architecture

A. Workstation

There will be on machine which will be publisher and responsible for publisher script on the server. User for creating and updating scripts will use this machine or updating our script will use it. User will write logic for downloading packages and their dependencies that will be published to the server machines and will be used latter by the subscribed machines.

B. Server

Server machines will be main machine that will be responsible for communication with both workstation and subscribed machines. Workstations will communication with subscribed machines through this machine. Whenever user will publish new or updated script on server. Server will scan through subscribed machines list and determine which of them needs updates. Server machine will send notification on relevant subscribed machines that will download script and will run on themselves.

C. Script

Script will be list of commands that needs to be run on our machines purchased from cloud. There can be one to many numbers of scripts. These commands are combination of installation and updated of current packages and new packages. These scripts will also have commands for downloading and installation packages of different kind of systems including Windows, Linux and Macintosh.

D. Subscriber

Script will be published and machines can subscribe different scripts. Whenever there is something new or update on server, it will further send notification to all machines those have subscribed that there is something new or update available. Our machines will pull script, which will auto run and perform operations required.

By using this structure we work much faster than the previous structure, for calculating time for different structure we have derived following equation.

$$T = a1\ (x1 + x2 + x3 + \ldots..) + a2\ (x1 + x2 + x3 + \ldots..) + a3\ (x1 + x2 + x3 + \ldots.) + \ldots.. \quad (1)$$

Where,

- T → is total time
- a → type of machine (win, Linux etc.)
- x → software package (new or update)

Both 'a' and 'x' can be as many as we have different kind of machines and different type of software packages respectively.

IV. RESULTS AND DISCUSSIONS

We did experiment of installing different dependencies on a single machine and calculate time required for installation. For experiments we have used Linux server and install different packages of python and their dependencies. For manual installation time required is calculated in seconds as given in the Table I.

Table I: Time Require For Packages Installation on Linux

Packages	Time Required (in seconds)
Django	300s
Scrapy	200s
Pycrypto	120s
Lxml	60s
Requests	60s
Libsass	50s

We assume that we have total numbers of 20 machines same as we have calculated time. As we may have different machines with different operating systems we have divided 20 machines into different kinds. Out of them we have 10 Linux machines, 5 with mac and 5 with 5 Windows. As we have calculated time taken by a person for manual installation on Linux server we are assuming that for manual installation person will take same amount of time. Be putting these values in eq. (1).

T = 10 (300 + 200 + 120 + 60 + 60 + 50) + 5 (300 + 200 + 120 + 60 + 60 + 50) + 5 (300 + 200 + 120 + 60 + 60 + 50)

T = 7900 + 3950 + 3950

T = 15800s (263.33 in minutes)

In contrast when we have pub-sub architecture we need to do effort 1 time for all kind of machines. We will write our script one time. Script will be deployed to server that will take care of installation every subscriber machine and its installation. Script will have code through which we can identify the machine type for running specific commands for recommended installation.

For writing we need to define script one time and we can take advantage of that for multiple machines, so we will again put these values in eq. (1).

$T = a_1\ (x_1 + x_2 + x_3 + \ldots..) + a_2\ (x_1 + x_2 + x_3 + \ldots..) + a_3\ (x_1 + x_2 + x_3 + \ldots..) + \ldots..$

T = 1 (300 + 200 + 120 + 60 + 60 + 50) + 1 (300 + 200 + 120 + 60 + 60 + 50) + 1 (300 + 200 + 120 + 60 + 60 + 50)

T = 790 + 790 + 790

T = 2370s (39.5 in Minutes)

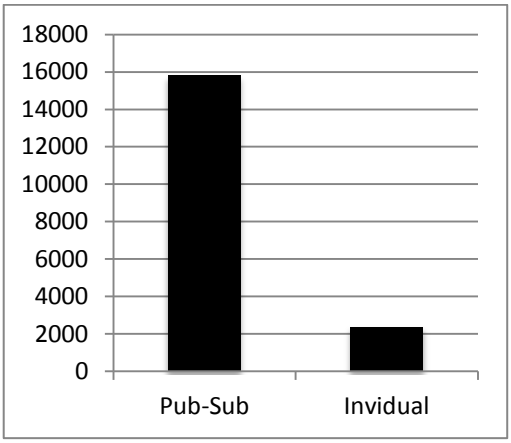

Fig. 6: Comparison between sub-sub architecture and individual installation

We can see in the Fig. 6 that 2370s are almost 6 times less than 15800. Through pub-sub architecture we can save a significant amount of time and adding new machines into our system will not have a major issue. It will provide way to cloud users so that they can enhance their applications and more machines with ease and within no time.

V. CONCLUSION

Since last decade in computer industries cloud computing has a major impacts. Cloud computing makes it easy to develop and deploy software projects by only working on domain logic. We have studied cloud computing's configurations current methods that how can we easily manage our purchased computer resources from cloud providers. In this research, we have proposed a new architecture, which is based on pub-sub modal. We have proved it with case study and concluded that how we can configure our cloud instances effectively and efficiently. In future we will implement this architecture and provide user interface to cloud purchasers where they can manage their cloud instances.

REFERENCES

[1] Lizhe Wang, Gregor, von Laszewski, Marcel Kunze and Jie Tao, "Cloud computing: A Perspective study", 2008.

[2] Michael Maurera, Ivona Brandic a and Rizos Sakellariou "Adaptive resource configuration for Cloud infrastructure management", Future Generation Computer Systems, Vol. 29, Issue 2, February 2013, pp. 472–487.

[3] Thomas Loru nser, Charles Bastos Rodriguez, Denise Demirel, "Towards a New Paradigm for Privacy and Security in Cloud Services", Technische Universit¨at Darmstadt, Germany

[4] Tuomas Kekkonen, Teemu Kanstrén, Kimmo Hätönen, "Towards Trusted Environment in Cloud Monitoring" ITNG '14 Proceedings of the 2014 11th International Conference on Information Technology: New Generations, pp. 180-185.

[5] Magdalena Kostoska, Marjan Gusev, Sasko Ristov, Ss. Cyril and Methodius, "A New Cloud Services Portability Platform", 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013, Vol. 69, 2014, pp. 1268–1275.

[6] Loganayagi, S.Sujathaa, "Enhanced Cloud Security by Combining Virtualization and Policy Monitoring Techniques", International Conference on Communication Technology and System Design 2011, Vol. 30, 2012, pp. 654–661.

[7] Pankaj Deep Kaur, Inderveer Chana, "A resource elasticity framework for QoS-aware execution of cloud applications", Future Generation Computer Systems, Vol. 37, July 2014, pp. 14–25.

[8] Chao-Tung Yang, Jung-Chun Liu, Kuan-Lung Huang, Fuu-Cheng Jiang, "A method for managing green power of a virtual machine cluster in cloud", Vol. 37, July 2014, pp. 26–36.

[9] Online available at http://www.infoworld.com/article/2614204/data-center/puppet-or-chef--the-configuration-management-dilemma.html [visited: January, 2016]

[10] Giuseppe Aceto, Alessio Botta, Walter de Donato and Antonio Pescape, "Cloud Monitoring: A survey", Computer Networks, Elsevier 2013 – Article.