Enabling Platform-As-A-Service through a Consolidated Resource Manager

Ganesh N, Prof. G S Mamatha

Information Science and Engineering, RVCE, Bangalore, India

Abstract*: Platform as a Service (PAAS) is a new initiative that is changing the structure of the organizations from traditional infrastructure services to the cloud based services. With the growing demand for these cloud services, organizations that offer these services need effective management of the resources provided. The paper provides an efficient solution in this regard by providing a common resource manager across all the platforms, coupled with administrative and reporting capabilities, further empowered with additional features like security and middleware messaging provided as optional add-ons. The resource manager automates the entire lifecycle of application workflow starting from its development until production. This multifaceted resource manager consolidated with a plethora of features, supplemented by the power of cloud, transforms into a tool that can modify and scale itself according to the ever-changing demands of the customers providing the right competitive business edge to the organization.*

Index Terms*: Cloud based services, Platform as a Service (PaaS), Resource Management, Workflow Automation.*

I. Introduction

The advent of Cloud Computing has revolutionized the service structure of organizations. Various products offered at the hardware and the middleware layers are offered as services provided as and when required. Platform as a service (PaaS) is one such category of cloud based services that provides a computational platform as a service [1]. In this category, various services such as servers, databases, ERP solutions, business intelligence tools and so on are provided to the application developers to develop, test and deploy their applications in a plug and play fashion.

PaaS operates at the middleware layer bridging the hardware to the applications. As part of any middleware service, the most important tool is a resource manager. A Resource Manager consolidates operations and management of similar Application Service Platforms in order to provide a coherent and consolidated toolset to the clients and support teams [2]. The paper introduces a resource manager that will provide operations and management of Application Web Server services which include Apache [3], WebLogic [4], Web Services Gateway [5], OBIEE (Oracle Business Intelligence Enterprise Edition) [6], BPM (Business Process Management) [7] etc.

A consolidated resource management tool in contrast to individual Web Administration, Certificate Management and Reporting enables uniformity within the platform. This uniformity helps in enabling the platform services to scale itself to the changing user requirements and upgrades from the infrastructure perspective. It also provides ease of use to the users and the platform developers while changing or expanding the platform.

There has been significant study on how the Cloud Service Architecture has impacted organizations in accelerating the delivery of their products and helping them tap the market in the right direction strategically. Wouter Belmans et al., [8]-[11] demonstrates the PaaS advantages in accelerating time-to market and delivering better quality software more quickly. Rake Revelant et al., [12] describes the opportunities of Platform as a service in the telecommunication domain. The project tries to implement the technique for Application Web Server services to develop web applications.

The Project introduces a novel concept in terms of Software delivery as a Service automating the entire workflow of an application's lifecycle starting from its development until its production. Although many organizations are trying to enter the field with their cloud strategies, there has not been a stable deliverable available in the market till now. The project tries to utilize this research gap to expand its cloud market and deliver the right product.

This rest of the paper is organized as follows. Section 2 describes methodology of the project describing the different modules being used in the project. Section 3 present the experimental work carried out to demonstrate the implementation of the project. Section 4 describes the results and inferences obtained. Finally, the conclusion and future work of this paper are presented in the section 5.

II. Methodology

A. Overview

Fig 1 describes the overall methodology of the system. Once an order is submitted by the application developer, the backend resource management is triggered which sets up the entire environment and triggers the web administration and reporting modules which dynamically update and incorporate these details.
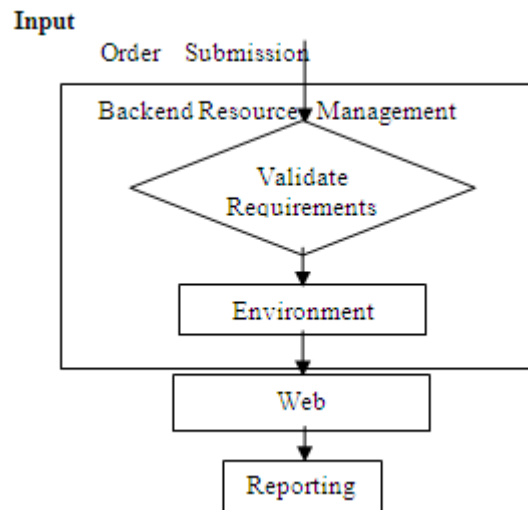

Fig 1. Block diagram of the consolidated resource manger

B. Modules

The project is mainly divided into three modules:
1. Resource Management
2. Web Administration
3. Reporting

C. Resource Management

The first module within the project is a Resource Manager that gets triggered whenever an order is placed into the system. The first stage of the resource manager is to validate the order to check if there is enough capacity to accommodate the application. Once the application is validated, the necessary environment is set up to develop the application. The Resource Manager implements workflow automation that automatically manages the environment as per the lifecycle of the application. The relation between the three modules is described in Fig 2.

D. Web Administration

The Web Administration module provides various administrative capabilities though web interface. The administrative capabilities are provided in order to maintain the running applications by the service teams. This module supplements the working of the resource manager by providing updates on its working. It provides various capabilities like starting an instance of application, stopping an instance, restarting an instance, host administration download certificates and so on.

E. Reporting

The Reporting module provides reporting facility that gets dynamically updated from the. These databases are constantly monitored and updated by the resource management module on any changes. The reports are made into intelligent dashboards, charts and graphs that enable effective analysis and provide better clarity on the usage and give an overall summary of the various services offered within the project.
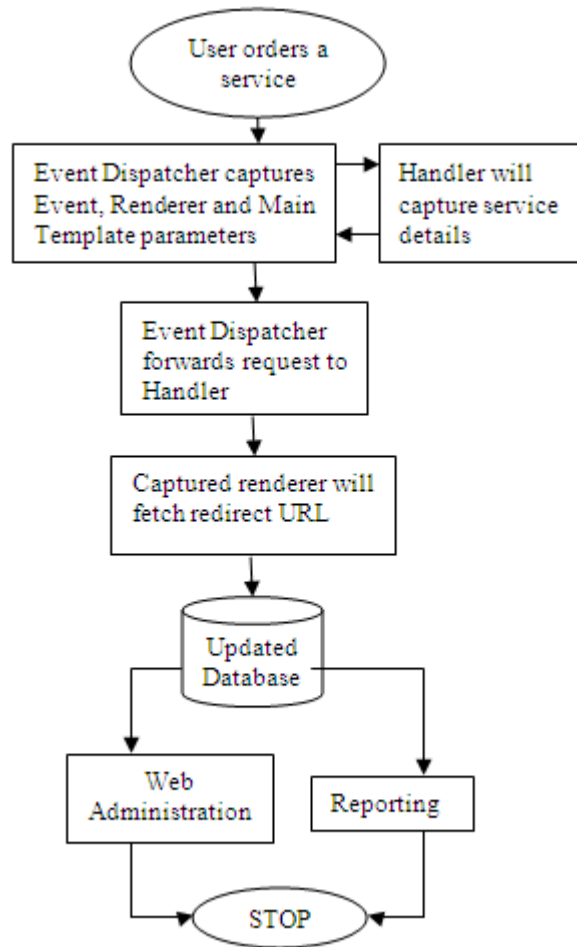
Fig 2. Relationship between the different modules

III. Implementation

The project has been carried out to cater the various requirements of Application Web Server (AWS) services. A set of 14 AWS services are implemented in this project over a set of more than 5000 customers from a varied background. The uniformity available across the various platforms enables the project to further provide additional support services in the form of Certificate Management for security features and Middleware Messaging Exchange (MMX) for middleware messaging [13]-[14]. The services are provided across various stages of an application lifecycle starting from development to production. Fig. 3 describes the typical set of lifecycles provided for the applications.

Fig 3. Stages of the Lifecycle of a Typical Application

The major task carried out in the project is to automate the entire workflow of lifecycle of the various applications. The entire environment needed as and when the application shifts form one stage to the other as described in Fig 3 is automatically carried out in the backend by the resource management module.

IV. Results and Discussions

The consolidated resource management tool has been successful in catering to the needs of 14 different services. These services mainly include Application Web server services like apache, Web Logic, WebSphere etc. along with Database services like Schema, Oracle Database etc. and application Messaging services like BPM. The consolidation of the resource manager for all the 14 services provides uniformity among the various services. This uniformity has resulted in increasing the user experience providing ease of use to the varied set of users. Along with the ease of use, this uniformity helps the developers and maintenance teams to easily handle the upgrades and

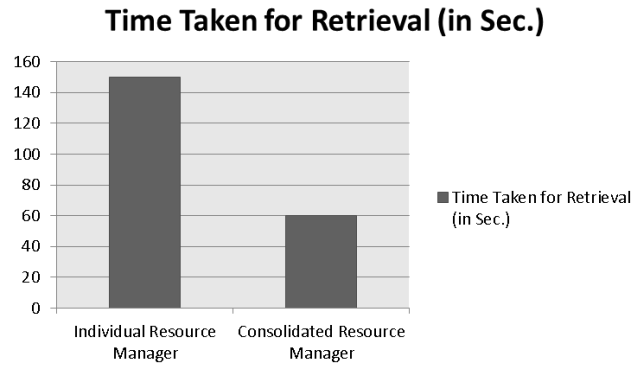changes that occur to the environment setup.

Time Taken for Retrieval (in Sec.)

Fig 4. Time Taken for Retrieval of Resource Mangers

Fig 4 describes the time taken for retrieval of data from the database to update reports and dashboards for individual resource managers and a consolidated resource manager. It can be observed from the figure that the time taken for retrieval for a consolidated resource manager is almost half that of an individual resource manager. This fact can be substantiated from the fact that a common database with a clear distinction of various layers of interacting code is provided in a consolidated resource manager.
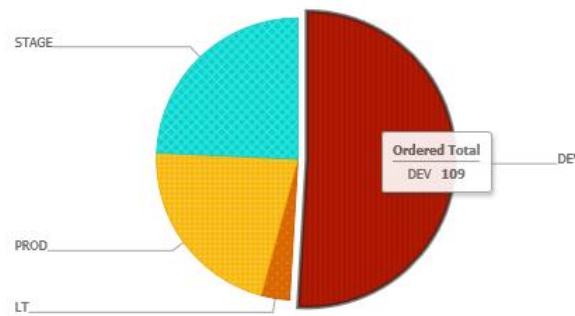

Fig 5. Dashboard with various lifecycle stages

The reporting module of the project is converted to form intelligent dashboards and graphs though a web interface to form the front end of the project along with the web administration capabilities. Fig 5 portrays the dashboard provided as part of the reporting module to describe the various stages of lifecycle as of the different applications within a particular service.

host_name	host_grp_name	max_sga	avail_sga
dbs-proj1-vm-001	STANDALONE_E	48	36
dbs-proj1-vm-002	STANDALONE_E	48	38
dbs-proj1-vm-003	STANDALONE_E	48	32
dbs-proj1-vm-004	STANDALONE_E	48	34
dbs-nprd2-vm-001	STANDALONE_N	48	8
dbs-nprd2-vm-002	STANDALONE_N	48	6
dbs-nprd2-vm-003	STANDALONE_N	48	18
dbs-nprd2-vm-004	STANDALONE_N	48	10
dbs-prd1-vm-001	STANDALONE_P	48	34
dbs-prd1-vm-002	STANDALONE_P	48	34
dbs-prd2-vm-001	STANDALONE_P	48	34
dbs-prd2-vm-002	STANDALONE_P	48	38
dbc-proj1-vm-001	GRID_E1C001	48	26.5
dbc-proj1-vm-002	GRID_E1C001	48	26.5
dbc-proj1-vm-003	GRID_E1C001	48	39.5
dbc-proj1-vm-004	GRID_E1C001	48	39.5
dbc-proj1-vm-005	GRID_E1C002	48	32.5
dbc-proj1-vm-006	GRID_E1C002	48	34.5

Fig 6 Report displaying capacities of various applications

Fig 6 shows a snapshot of reports of the capacities of various applications of a particular service. It includes the maximum capacity of the particular application as well as the available capacity giving an option of scalability.
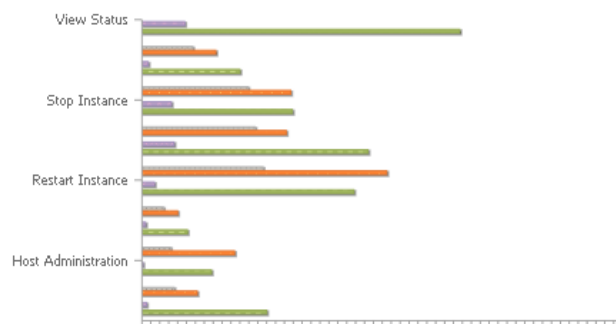
Fig 6. Dashboard of few Administrative actions

The Web administration module provided on the front end is coupled with the reporting module to form dashboards describing the various actions being taken for the particular service. Fig 6 depicts the dashboard displaying summary of some of the administrative actions taken for the service.

V. Conclusion

As part of middleware services, the paper introduces a novel and efficient resource manager coupled with various capabilities like web administration and reporting providing an excellent alternative to the existing resource managers. The paper also describes the advantages of consolidation of various services that are catered by the resource managers in various dimensions including ease of use, easier maintenance, effective handling of upgrades and performance in terms of retrieval time. The resource manager is furthermore supported by provision of Software Delivery as a Service to automate the lifecycle workflow of an application starting from development till production. In brief, the paper provides a tool that   In short, the paper tries to provide a tool to enable Platform as a Service to deliver the right product to customers.

Acknowledgement

References

[1]. William Y. Chang, Hosame Abu-Amara, Jessica Feng Sanford, "Transforming Enterprise Cloud Services", Springer, pp. 55, Nov. 2010
[2]. Han Xingye, Li Xinming, Liu Yinpeng, "Research on Resource Management for Cloud Computing Based Information System", 2010 International Conference on Computational and Information Sciences (ICCIS), pp. 491-494, Dec. 2010.
[3]. Erwin Fischer, "APACHE Web Server in BS2000/OSD", Fujitsu White Paper, pp.1-4, April 2009.
[4]. David Baum, Ruma Sanyal Ayalla, Goldschmidt Mike Lehmann, "Introducing Web Logic Server 12C", ORACLE White Paper, pp. 10-16, January 2012.
[5]. Younkwan Lee, Kangchan Lee, Seungyun Lee, "Web Services Deployment Model Based on WSG (Web Services Gateway) in NGN", International Journal of Software Engineering and Its Applications, Vol. 7, No. 1, pp. 229-236, January 2013.
[6]. Robert Stackowiak, Tobin Gilman, George Lumpkin, Charlie Berger, "ORACLE Business Intelligence, ORACLE Database and Exadata Integration", ORACLE White Paper, pp. 1-4, October 2010.
[7]. Dan Tortorici, "ORACLE BPM Suite 11g: BPM without Barriers", ORACLE White Paper, pp. 6-10, November 2010.
[8]. Wouter Belmans, Uwe Lambrette, "Cloud Value Chain Exposed", Cisco white paper, pp. 4-8, March 2012.
[9]. Qianhui Liang, Bu-Sung Lee, "Delivering High Resilience in Designing Platform-as-a-Service Clouds", IEEE Cloud 2011, pp. 4-9, July 2011.
[10]. David M Grimes, "Technology Evolution driving cloud Solutions", Society of Table Communication Engineers, Atlanta, pp. 4-7, October 2013.
[11]. Boniface M, Nasser B, Papay J, Phillips S C, "Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds", 2010 Fifth International Conference on Internet and Web Applications and Services (ICIW), IEEE, Barcelona, pp. , 155-160, May 2010.
[12]. Rake Revelant J, Holschke O, Offermann P, Bub U, "Platform-as-a-Service for business customers ", 2010 14th International Conference on Intelligence in Next Generation Networks (ICIN) , IEEE, Berlin, pp. 1-6, Oct. 2010.
[13]. Stefan Tai, Thomas A Mikalsen, Isabelle Rouvellou, "Using Message-oriented Middleware for Reliable Web Services Messaging" , Springer-Verlag Berlin Heidelberg, Volume 3095, pp. 89-104, 2004.
[14]. Taton C, De Palma N, Philippe J, Bouchenak S, "Self Optimization of Clustered Message Oriented Middleware", Fourth International Conference on Autonomic Computing 2007 ICAC '07, Jacksonville, FL, IEEE, pp. 18, June 2007