International Journal of Innovations in Engineering Technology and Applied Sciences
Volume 1, Issue 1 January 2025

A Modular Broking Architecture for Optimized Cloud Service Deployment and Scheduling

Harshit Kesharwani
School of Engineering andTechnology,
Sharda University,
Greater Noida, India
harshitkesharwani18@gmail.com

Aakash Nakarmi
School of Engineering andTechnology,
Sharda University,
Greater Noida, India
aakasnakarmee@gmail.com

Tamoshree Mallick
School of Engineering andTechnology,
Sharda University,
Greater Noida, India
tamoshree.mallick.24@gmail.com

Abstract. In today's world, most companies are shifting their servers from on-premises to cloud infrastructure to deliver the most efficient service to their customers. To serve this, many cloud service providers provide different services of virtual machines, storage, networking, load balancing, and all the facilities a company will require to run its business. For the best implementation of cloud services, these companies utilize different scheduling strategies for a particular service. The competition is not just providing the service but it is providing the quickest and most efficient service. Many companies fail to automatically manage all components involved during the deployment of architecture in the cloud. This paper proposes a modular architecture that can be used for the optimal deployment of services provided in the cloud. Along with that, this architecture will be compatible with scheduling strategies as well so that different optimization criteria like cost, performance, storage, load balancing criteria, and so on. There was also an analysis of the architecture by performing the deployment of a website under various conditions like optimization criteria and maximum user requests.

Keywords: Cloud, optimization algorithm, load balancing, scheduling strategies, service workload

1 Introduction

Since the past decade, the market of cloud computing has grown by a huge margin. Many companies are moving their business from on-premises to private or public cloud. All the service management and workload are handled in the cloud, removing the overload on on-premises infrastructure management.

There are many public cloud service providers in the market like AWS, Google Cloud, Rackspace, and Azure. Some companies also set up their private cloud to maintain the best security and data protection. Also, hybrid cloud service is followed by companies according to their system requirements and meets their needs.

The main objective of these cloud service providers is to provide virtual resources to the client but the functionalities and services provided by them are different. In

addition to that, there will change in the payment plan as well. The cost of the use of cloud resources will be different according to the use and customization of the service like the number of virtual machines/instances used, and the subscription of the cost model. To properly schedule all these services, a broker can help in performing the optimization of the service deployment which is important. A cloud broker can act as an intermediate node between the end users and the cloud service providers. The end users can just need to provide the required results and according to the requirements of the client, the broker will automate the process and choose the best platform for deploying any service. There will be also provision of switching between the different platform according to requirements.

This architecture will provide a bridge between the end users and public cloud service providers so that the services can be managed automatically and there can be optimality in their service deployment. The client will not have to worry about the downtime of the service and the cost control as well. The broker architecture will automatically manage it. There are several broking architectures that are used by the broker for the management of the services under different circumstances but they are mostly static conditions and some are not optimal in terms of their distribution of services. The client end user also gets to choose the scheduling strategy as well according to their requirement. Many research studies have been conducted on this topic.

So, this paper proposes a modular broking framework for the optimal deployment of services so that the client/ end user can get the optimal result according to requirements like performance optimization, cost optimization, and utilization of minimum resources to save energy. This will provide automatic management of service workloads in the cloud.

The rest of the paper is discussed as follows: Literature Review is discussed in Section 2. Proposed Architecture in Section 3. Section 4 describes different scheduling strategies. Section 5 explains the experiment and results performed according to this modular architecture. The conclusion of the entire research paper is summarized in section no 6**.**

2 Literature Work

Many research works have been done previously on this similar topic. Some of the major relevant research works have been discussed below. Srikumar Venugopal, Rajkumar Buyya and Chee Shin Yeo et al. [1] provide a design for the market-based distribution of resources inside Clouds. The most recent platforms for cloud computing have been covered in our discussion of a few sample platforms. We have also outlined a plan for building a worldwide cloud exchange for exchanging services.

Rubén S. Montero, Borja Sotomayor, Ignacio M. Llorente and Ian Foster et al. [2] proposed a modular broker design for the best service deployments in multi-cloud settings with dynamic pricing and concentrate on the scheduling module, which is tasked

with delivering an ideal deployment to optimise a certain service parameter among the many broker components. Daniel Henriksson, Lars Larsson and Erik Elmroth et al. [3] outline current efforts on core service administration duties that are crucial to federated Cloud setups. For use in Cloud federations, they provide a hierarchical graph structure for representing services and any placement limitations imposed on the service components, such as site-level affinity.

Fermn Galan Marquez, David Perales Ferrera, Daniel Henriksson and Erik Elmroth et al. [4] provide a solution for an accounting and billing architecture for usage in RESERVOIR and maybe other federated Cloud settings. Bu-Sung Lee, Sivadon Chaisiri and Dusit Niyato et al. [5] suggest an optimal virtual machine placement (OVMP) method that supplies the resources provided by various cloud providers. An IaaS model that makes use of virtualization technologies serves as the foundation of the algorithm.

Ruben S. Montero, Ignacio M. Llorente and Rafael Moreno-Vozmediano et al. [6] examine the difficulties and practicality of establishing a computing cluster on top of a multi-cloud architecture covering four separate sites in order to address weakly linked MTC applications. Ruben S. Montero, Ignacio M. Llorente and Rafael Moreno-Vozmediano et al. [7] analyze the efficiency and expandability of the cloud-bursting model by implementing two different distributed web server architectures. The first architecture is a simple web server cluster designed to serve static files, while the second architecture is a multi-tier server cluster built specifically for running the Cloud-Stone benchmark.

Ruben S. Montero, Johan Tordsson, Rafael Moreno-Vozmediano and Ignacio M. Llorente et al. [8] suggest a unique method for cloud brokering in terms of a generic architecture for supplying multi-cloud resources. A cloud broker's primary responsibilities are to manage the resultant architecture and optimise the deployment of virtual machines among various cloud providers in accordance with user-specified criteria.

3 System Architecture

For every framework, there is a need for the proper systematic flow of a process and architecture. So, in this segment, there is a proposal of proper middleware broker architecture for multi-cloud environments as shown in Fig. 1. This architecture is useful when a website needs to be deployed with the minimum cost and maximum performance. According to the requirements, the companies can utilize this architecture for building the best business.

There are different components in this architecture but there are three major building blocks: Scheduler, Cloud Manager, and VM Manager. Every single component has a major role. The Scheduler is responsible for choosing the bests scheduling strategy for the deployment of service. The necessary information about the cloud infrastructure and different service providers is provided by the cloud provider. After getting the

necessary information about the service provider and best scheduling strategy, the deployment is handled by the VM Manager.

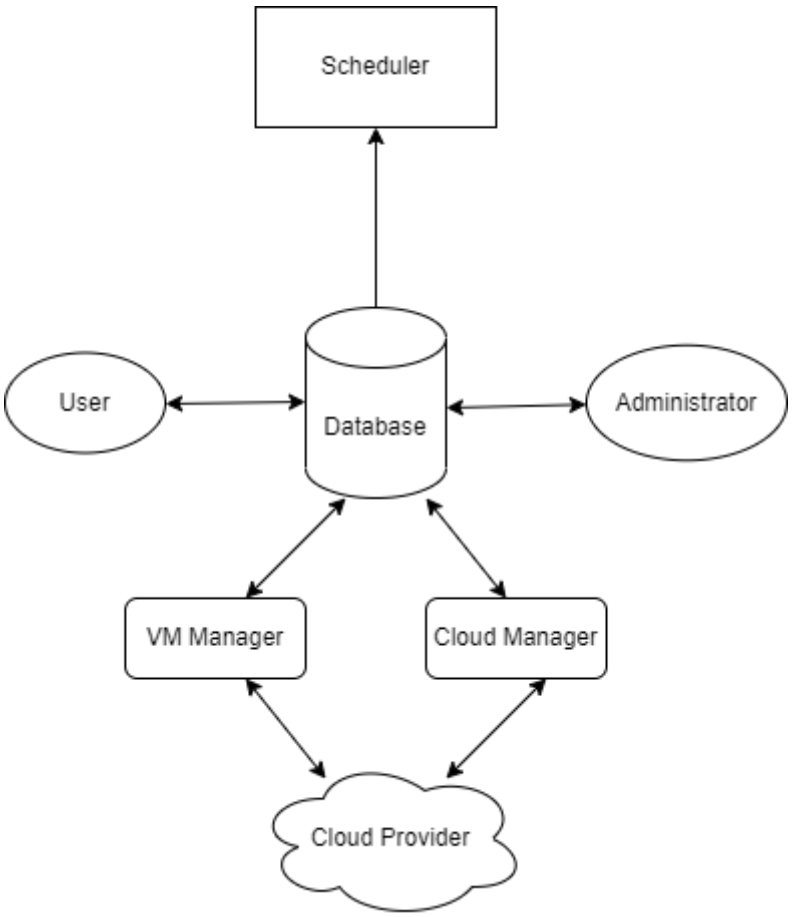

Fig. 1. Cloud Brokering Architecture

In this architecture, there are main primary actors: the user and administrator. The user is the client of the broker architecture. The client will provide the necessary information about the service to deploy and the requirements as well. According to the criteria provided by User, the administrator will use the scheduler, Cloud Manager and VM Manager to deploy the service. During deployment, various cloud services are utilized like virtual machines, containerization tools, databases and storage services.

There are various layers and sections that the cloud manager needs to look out like costing of instances, availability of instances, and comparison of the services. Basically, The Cloud manager will be playing the role of an interface that will be providing all the information in different cloud environments so the services can be scheduled, deployed, and backed up according to the requirements of the client.

The scheduler in this architecture plays one of the major roles in this architecture. With the help of a scheduler only, there is a systematic workflow of process and the optimization of cost and performance can also be achieved. The scheduler works on the following way:

- At first, the scheduler will receive the description document about the service and then points out the requirements of the client like cost-optimized service or performance-based service.
- After that, the scheduler will check out the availability of the instances, services offered in the cloud, and their price. After doing the analysis, the scheduler will apply a particular scheduling strategy with respect to the description file of service.
- Then, a database is prepared in which there is mentioned which set of VM instances will be used for the deployment of that service.

After the scheduling strategy is prepared, the VM manager extracts the set of VM instances that needs to be utilized for the deployment of the service. There are two major tasks of the VM Manager. The first task is the deployment of services by utilizing the VM instances according to the requirement of the client. After the deployment is done, the next task is to monitor them. It is very important to keep track of every instance so that if any error occurs, that can be resolved immediately.

4 Strategies of Scheduling

The architecture described in previous sections displays the advantages of using different scheduling strategies in the scheduler module according to the client's needs. In previous broker architectures, which have been discussed in the past they did not have these scheduling strategies which provided the most optimal framework. This section will propose different scheduling strategies which will show the importance of those strategies during the deployment of services in cloud.

4.1 Problem Definition

The objective of this project is to introduce scheduling algorithms that facilitate optimal deployments of web applications in a multi-cloud environment. The goal is to deploy a enterprise based web service consisting of virtual machine components executed across various cloud providers. To accomplish this, a specific number of virtual machines will

be deployed across the available clouds to optimize user criteria such as infrastructure cost or service performance.

It should be noted that the number of virtual machines (VMs) in this project is not constant, but rather dynamic to accommodate varying scenarios. For instance, if user demand shifts or if one VM of a particular instance type performs better than several VMs of another instance type, it may be necessary to adjust the number of VMs accordingly.

In the multi-cloud environment, each provider offers various hardware configurations, commonly known as instance types (ITs), with specific quantities of cores, memory, and disk storage, which have a minimum and maximum limit for each component. Some providers even allow users to customize their instance types by adjusting the quantity of cores, memory, or disk to suit their requirements. However, this project uses a standard set of instance types available in most clouds to ensure uniformity and ease of understanding of the use cases.

To begin problem formulation, the scenario is first introduced, and the following definitions are established:

- The term "t" refers to a time slot of 1 hour.
- The scheduling time period is defined as the next one-hour time slot for scheduling. As a result, the scheduler runs before each time period/slot to generate a total or partial reconfiguration of the VM infrastructure.
- During a specific time interval t, Xi,j,k(t) is a binary value that is assigned a value of 1 if virtual machine vi,j, where i refers to the specific machine and j indicates the instance type (with j ranging from 1 to l), is hosted on cloud service Ck. Alternatively, if the machine is not hosted on Ck during that time period, the value is set to 0.

4.2 Policy for Cost Optimization

The objective of this method is to reduce the cost of each virtual machine by selecting the cloud provider that offers the lowest prices. Typically, the cost function to be minimized is the Total Infrastructure Cost (TIC(t)), which is calculated as the sum of the costs of all virtual machines during a specified time period. Two distinct scenarios are identified: static and dynamic scenarios.

$P_{j,k}(t)$ represents the actual cost incurred for utilizing a virtual machine of a specific instance type j on a particular cloud service k during a given time interval t. The values of j, k, and t range from 1 to l, 1 to m, and any relevant time period, respectively.

Instance Variety	S	L	XL	High CPU
CPU	1	3	7	6
Performance	3.55	10.57	16.32	12.99
Ratio	3.55	3.52	2.33	2.165

Table 1. Table for Instance Type Features

During dynamic pricing environments, prices of comparable instances can fluctuate over time due to changes in demand. This approach assumes that the cost of every instance type for the next time slot of scheduling is known beforehand. If prices were unknown or subject to fluctuations during the scheduling period, prediction techniques based on historical data would be employed. However, these prediction techniques are not considered in this study. To integrate these prices into the scheduler, the TIC function, as presented in equation (1), is defined.

$$\mathrm{TIC}(t) = \sum_{i}^{n} \sum_{j}^{l} \sum_{k}^{m} X_{i,j,k}(t) * P_{j,k}(t). \quad (1)$$

4.3 Policy for Performance Optimization

The objective of this method is to optimize the overall performance of the infrastructure by placing each virtual machine in the cloud provider that offers the highest performance. The cost function to be maximized is the Total Infrastructure Performance (TIP(t)), which is calculated as the sum of the performances of all virtual machines during a specified time period. Similar to the cost minimization approach, static and dynamic scenarios are also distinguished in this study.

The performance of a virtual machine depends on various factors such as application requirements, type of virtual instance used, and physical infrastructure. These factors can affect the application's hard disk, memory, cache use profile, CPU utilization, and other performance parameters. Hence, the application owner needs to test the application in every instance type across all cloud providers to provide the performance information. It's important to note that the performance of an application may vary across different cloud providers due to differences in technologies, virtualization software, or

physical infrastructure placement. Equation (2) displays the equation for Total infrastructure Performance:

$$\mathrm{TIP}(t) = \sum_{i}^{n} \sum_{j}^{l} \sum_{k}^{m} X_{i,j,k}(t) * \mathrm{Perf}_{j,k}(t).$$

(2)

The results obtained by applying the benchmark to several Google Cloud EC2 instance types are presented in Table 2.

Instance Type	S	L	XL	High CPU Med
CPU	1	5	9	4
RAM(in Gb)	1.8	8.5	16	1.8
Storage(in Gb)	170	860	1700	360

Table 2. Instance types performance table

5 Modeling and Testing a Web Application

This section will provide details about the modeling and testing of a web application using the architecture manually. To deploy the application, there was the use of Google Cloud as the Cloud platform. At first, a normal web application was developed using HTML, CSS, and JavaScript. The web application was a gaming platform containing strategic games like chess, sudoku, etc. The web application was a gaming platform so that there can be great user interaction. The number of user requests is important for the analysis. The application was deployed with a different number of instances to test the performance. With every number of instances, the web application was tested to monitor and analyze the performance, the logs of the user request, and the time taken by the web application to respond was measured. For every measurement, a diagrammatical representation was done to have a visual clearance of the performance of the website according to the number of instances.

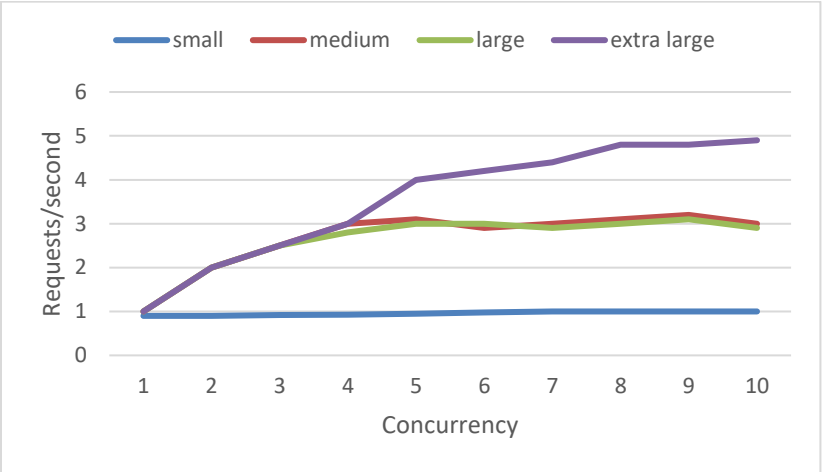

Fig. 2. Requests attended per second

Fig. 2 depicts the number of requests attended per second. According to a number of instances used, there are different lines that represent their respective number of instances and their request handling per second under concurrent load conditions. It is clearly observed that when the number of instances is increased the number of requests attended per second increases and vice versa. But the problem with this is the costing of instances.

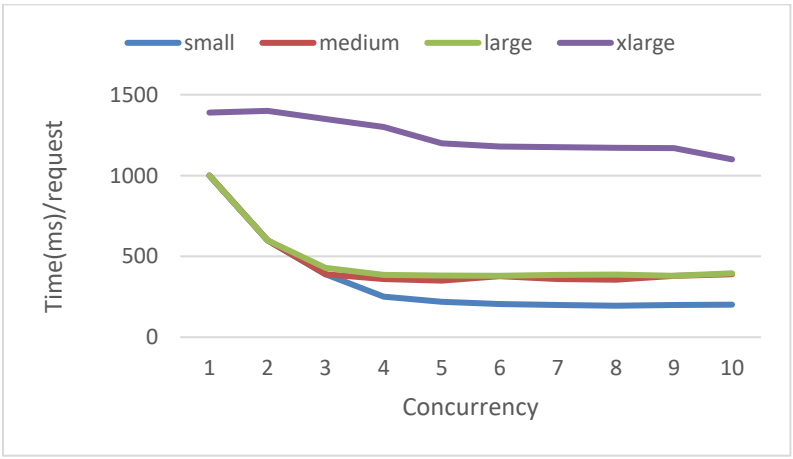

Fig. 3. Time duration to attend a single request

Fig. 3 illustrates the duration required to process a single request. As the number of instances rises, the time needed to handle a request substantially decreases. Conversely, when there are fewer instances, the time needed to attend to one request is relatively higher.

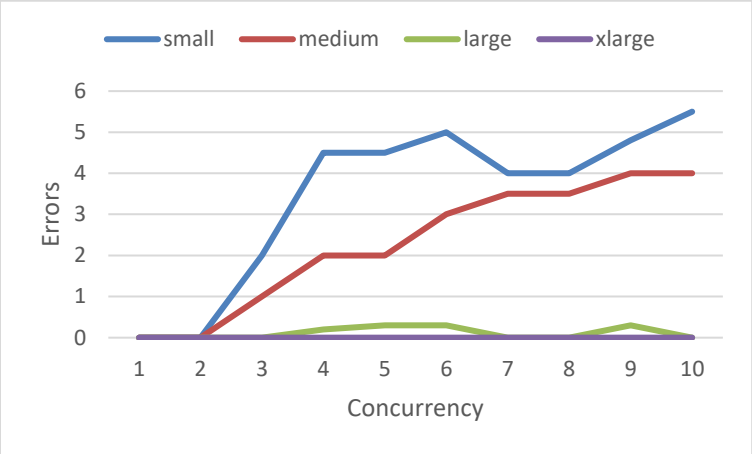

Fig. 4. Failed requests

Fig. 4 shows the number of failed requests when the different numbers of instances are operational. When the concurrent load is given to the web application, the number of errors is very less when the number of instances is less. And with an increase in the number of instances, there is less number of errors and vice-versa.

6 Conclusion

To sum up, this paper proposes a framework for the optimal deployment of web applications in a multi-cloud environment. The optimality may be in terms of price and performance according to the requirement of the client. There are different parameters which will be responsible for this optimal approach like choosing of best service with low-cost price according to the nature of the application, scheduling strategy, availability of resources, fault tolerance/backup of the instance if any instance fails during the workload distribution, and so on. The major challenge of this architecture will be the automation of this entire framework and the integration of all the components involved in the architecture. The trust and goodwill that the broker will have to maintain by providing the best service is also a challenge.

In the future scope, more research is possible on deploying huge enterprise micro-service-based applications using this architecture as handling microservice-based applications is a huge task today. Along with that, the architecture can also be tested when applications have some customization and the adaptability of architecture with those modifications.

References

1. Buyya, Rajkumar, Chee Shin Yeo, and Srikumar Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities." 2008 10th IEEE international conference on high performance computing and communications. Ieee, 2008.
2. Sotomayor, Borja, et al. "Virtual infrastructure management in private and hybrid clouds." IEEE Internet computing 13.5 (2009): 14-22.
3. Larsson, Lars, Daniel Henriksson, and Erik Elmroth. "Scheduling and monitoring of internally structured services in cloud federations." 2011 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2011.
4. Accounting and Billing for Federated Cloud InfrastructuresRavishankara K, Dhanush, Vaisakh, Srajan I S, "Whatsapp Chat Analyzer", International Journal of Engineering Research & Technology (IJERT), 2020.
5. Chaisiri, Sivadon, Bu-Sung Lee, and Dusit Niyato. "Optimal virtual machine placement across multiple cloud providers." 2009 IEEE Asia-Pacific Services Computing Conference (APSCC). IEEE, 2009.
6. Moreno-Vozmediano, Rafael, Ruben S. Montero, and Ignacio M. Llorente. "Multicloud deployment of computing clusters for loosely coupled mtc applications." IEEE transactions on parallel and distributed systems 22.6 (2010): 924-930.
7. MorenoVozmediano, Rafael, Ruben S. Montero, and Ignacio M. Llorente. "Elastic management of web server clusters on distributed virtual infrastructures." Concurrency and Computation: Practice and Experience 23.13 (2011): 1474-1490.
8. Tordsson, Johan, et al. "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers." Future generation computer systems 28.2 (2012): 358-367.
9. Ferrer, Ana Juan, et al. "OPTIMIS: A holistic approach to cloud service provisioning." Future Generation Computer Systems 28.1 (2012): 66-77.
10. Buyya, Rajkumar, et al. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation computer systems 25.6 (2009): 599-616.
11. Mateescu, Gabriel, Wolfgang Gentzsch, and Calvin J. Ribbens. "Hybrid computing—where HPC meets grid and cloud computing." Future Generation Computer Systems 27.5 (2011): 440-453.
12. Qawqzeh, Yousef, et al. "A review of swarm intelligence algorithms deployment for scheduling and optimization in cloud computing environments." PeerJ Computer Science 7 (2021): e696.
13. Frey, Sören, Florian Fittkau, and Wilhelm Hasselbring. "Search-based genetic optimization for deployment and reconfiguration of software in the cloud." 2013 35th international conference on software engineering (ICSE). IEEE, 2013.

14. Wada, Hiroshi, et al. "Evolutionary deployment optimization for service-oriented clouds." Software: Practice and Experience 41.5 (2011): 469-493.
15. Xu, Bo, et al. "Dynamic deployment of virtual machines in cloud computing using multi-objective optimization." Soft computing 19 (2015): 2265-2273.