International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

A Review on Efficient Resource Allocation & Scheduling Approaches in Cloud Computing

Shweta
M.Tech (Computer Science)
Kurukshetra University, Geeta Engineering College,
Panipat, Haryana, India

Ms. Mansi Singhal
Assistant Professor
Computer Science &Engineering,
Geeta University

Mr. Kapil Saini
Assistant Professor
Computer Science &Engineering,
Geeta University

Abstract— **Cloud computing offers flexible, versatile, asset sharing administrations by utilizing asset the executives. Asset observing and expectation are the keys to accomplish asset usage with superior execution the executives in distributed computing. Asset booking is one of the significant issue of distributed computing, the booking strategy and calculation influence the exhibition of cloud framework straightforwardly. Lately, Cloud Computing offers high-performance registering limit, which reminds cloud suppliers to use asset completely in light of the impediment of assets. This examination paper expects to screen the assets accessible in cloud utilizing Hidden Markov Model (HMM). The proposed model is utilized for asset checking and afterward the asset will be grouped in light of Less, Average, and Heavy stacked classes as the accessibility of the assets and the fitting planning calculation will be chosen on request, the proficiency of calculation has been adjusted utilizing different sort of responsibility situation. Specialist organization communication". [1]**

1. INTRODUCTION

Cloud computing systems are on the path of great success economically as they are capable of providing huge amount of different kinds of services and resources to their customers. Besides, intelligently developed recommendation systems have contributed vastly in successfully letting the customer decide if a particular service is required for him or not. In this epoch of cutting-edge technology, scheduling is one of the preferred strategies that assigns the user-defined requests to the resources allocated in a particular time frame. [2] Requests can exist in virtual computational form, where elements such as process or thread are executed on hardware resources such as expansion cards, network links and processors. A cloud has infinite number of resources where scheduling approaches play a crucial role of taking great benefits from resources by effectively utilizing them. Extensively, resources should be automated intelligently to execute the requests effectively. While considering the procurement of automation, an algorithm is the key element which is accountable for successfully arranging tasks' execution among several resources while preserving data security. [4]

Cloud Computing

First computer came into the existence in the form Abacus in 3000 B.C. Thereafter, Abacus was replaced with ENIAC (Electronic Numerical Integrator and Computer) drifting through numerous technology advancements. ENIAC was the first general purpose computer introduced in 1945 at Moore School of the University of Pennsylvania to solve large numerical problems. Latterly, to solve complex data and decision making, first mainframe computer (System/360) was introduced by IBM in April 1964. Personal computers were introduced to use computer personally in homes and offices. To exchange data and information among geographically distributed users, "internet" was introduced. To access internet based services easily, distributed computing, grid computing and cloud computing were proposed, respectively.

Cloud computing technique was introduced in 1960s [2]. John McCarthy once said, “Computation may someday be organized as a public utility” [3]. Afterwards, in the early 1990s, grid computing was introduced. The key notion behind grid computing was to access computing power as electricity [4]. Grid computing has a major contribution in originating “Cloud Computing”. The term “cloud computing” was used in its context by Ramnath Chellappa in one of his lectures in 1997 [5].

2. MOTIVATION

Scheduling approaches act as decision makers in taking leverage of assistance provided by cloud computing. The scheduling strategies have been broadly analyzed over the years on grid and cluster platforms. In recent years, researchers are aiming to develop the unequivocal resource model that offers the cloud services, and is essential to enable the automation of scheduling efficiently. Various approaches designed for other platforms can be applied on the cloud platform. However, these approaches generally fail in providing benefits of accessing unlimited resources on demand, are less economic due to not availability of cost model and are unsuccessful to provide cloud facilities such as dynamic performance requirement and resources' integrity. [8]

3. CLOUD COMPUTING ARCHITECTURE

Cloud Computing architecture is comprised of several elements where each element or component is independent and loosely coupled to each other. Cloud architecture can be broadly divided into two components:

- Front-end
- Back-end

The part of the system that represents the client infrastructure is called front-end, which is visible to the user. It comprises of various applications, browser and devices such as desktop, mobile, which assist in accessing various cloud services.

To illustrate, if a live ware wants to access Gmail, then it can be accessed by using browsers like Mozila Firefox, Google Chrome, Netscape and so forth.

The system part which is not visible to the user is called back-end. In a cloud computing environment, cloud itself works as a back-end in the whole process. Back-end

of the cloud consists of network, data storage, virtual machine (VM), servers, deployment models, various services including SaaS, PaaS and IaaS, security mechanism and many more. Architecture of the cloud has been illustrated in figure 1 [17].

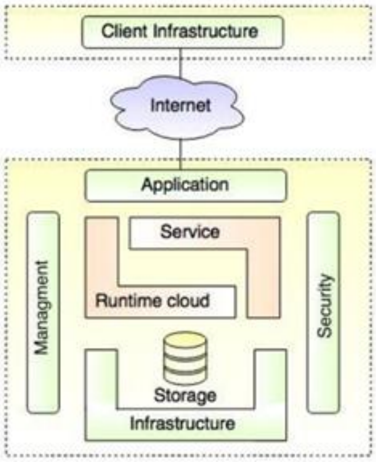

Figure 1 Cloud Computing Architecture.

Converged Technologies in Cloud Computing

Cloud Computing is not a single technology, rather it is a convergence of various technologies like web 2.0, SOA (Service Oriented Architecture), virtualization, utility, grid computing and distributed system. These technologies lead to deliverance of internet-based services. Different technologies that converge into Cloud Computing have been shown in figure 1.2 [18].

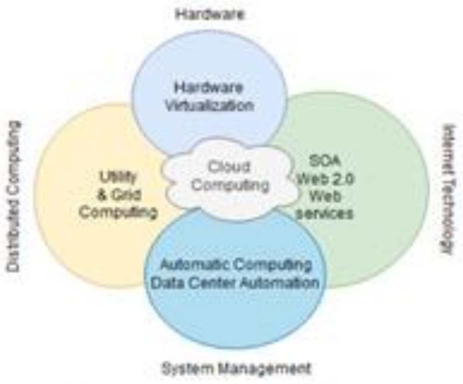

Figure 1. 2 Convergence of Technologies in Cloud.

Cloud Computing Deployment Models

In cloud, the way services are offered to the users is defined by cloud deployment models. There are four deployment models in cloud [24]:

- **Public cloud.** As the name implies, this model is freely available to the users, who want to make the use of cloud computing services. This service model is supported by all users' services either software based like database and application server or hardware services like CPU (Central Processing Unit), O.S. (Operating System), storage and memory, based on the selected subscription type. E-mail services, file sharing, software development and testing are the most commonly used services of public cloud. Some public cloud providers are Google Cloud Platform, Amazon Web Services, IBM Cloud and Microsoft Azure.
- **Private cloud.** This type of cloud is owned by the organization itself. Private cloud infrastructure is setup and managed by organizations for their personal use. Administration can be provided by some administrator group or by some party either online or offline. This cloud model is extremely expensive in comparison to the public cloud due to acquisition and maintenance of the cloud. Nevertheless, private clouds are considered more secure as they are privately owned by organizations.
- **Hybrid cloud**. Hybrid cloud is an interconnection of public and private clouds. This model assists the organizations to take benefits of public cloud services while maintaining security and privacy issues. To exemplify, during a particular season, if an online retailer desires for more computing resources, then he can obtain these services via public cloud.
- **Community cloud.** This type of model is used when numerous organizations share the computing resources, including banking branches situated at various locations, share their customers' data, researchers of different universities share their ideas and notions, and even police department also shares its data to various departments and branches via community cloud.

4. CLOUD SERVICE MODEL

The main objective of a cloud service or a cloud platform is to meet all needs by facilitating various requirements, where these needs are serviced by providing a customized or a packaged approach towards a specific problem. This packaged approach is nothing but a cloud service, where the service models and the reference models on which Cloud Computing is based. Also, computing is completely based on these service models with the assistance of service models. These service models can be categorized into three basic service models as follow [25]

- **Software-as-a-Service (SaaS).** In Software-as-a-Service model, software is provided to the end user as a service from vendor which is mandatory irrespective of the operating system. It is connected to the end users, where the end user applications are delivered as a service rather than the on-premises software. It comprises operating systems, applications stack, server storage and networks, where these resources are managed by the vendors. Furthermore, it provides the multi tenancy to the cloud users [24] as same resources are shared using only one instance of project code and underlying database to different customers simultaneously.
- **Platform-as-a-Service (PaaS).** PaaS delivers various types of services in a form of computing platform including programming language execution environment, operating system, and some other tools like building and designing, and also assist in the deployment of users' application onto the cloud.
- **Infrastructure-as-a-Service (IaaS).** In IaaS model, cloud services are accessed from computing resources in a virtualized environment. Computing unit, network, storage and other rudimentary computational resources are included in the services facilitated by IaaS model.

5. SECURITY ISSUES IN CLOUD

There are some security issues, which have been considered by Gartner [20]:

- **Compliance**: User must refuse the services of the cloud that do not provide the external audits and traditional security certificates.
- **Data segregation:** As cloud provides data sharing, therefore the organizations can consult the vendor about their segregation policy and data security. Although encryption assists in data protection, but cloud user must ensure that security mechanisms are properly tested.
- **Support:** There must be full support provided by vendor while investigating any illegal activity.
- **User access:** A cloud user must be cognizant of the administrators, who have the authority to look into their data and information.
- **Data location:** While using cloud computing services, organizations have the right to know about its data location. Cloud vendors are committed to follow the privacy rules locally.
- **Recovery:** In the phase of disasters, what recovery mechanism should be adopted, must be predefined by the cloud vendors. Also, the recovery of entire data by the vendor or replication of data in different sites, must be ensured by the cloud vendors.
- **Viability:** If a cloud vendor is overtaken by some third company, then vendor must ensure about the continuous serving of the services with same or high level of security mechanism.

6. RELATED WORK

Deadline-based approach has to confront a few major challenges in cloud based services. Since the tasks are executed on IaaS platform, therefore energy consumption is the primary concern. In this context, Lu Guan et al. proposed a methodology named Dynamic Resource Allocation Method Based on Deadline Time (DRAMDT) [13]. The methodology was based on grouping of VMs of similar deadline and waking up only those groups of VMs that are required in order to save energy. Javier Celaya et al. found the available computational resources (by means of network) that accomplish the jobs. The methodology was performed by using decentralized scheduler [14] that includes tree-based network overlay, where each level of tree denotes the sum to available nodes. This methodology supports decentralized scheduler. Global scheduler performed availability of nodes and local scheduler used EDF (Earliest Deadline First) scheduling policy to execute the requests. However, on cloud, flow deadline scheduling was implemented by Maciej Malawski et al. using cost and tasks [12]. This approach worked on different workflows of a task. Besides workflow, tasks priority of deadline based tasks were set to schedule the tasks efficiently. To reduce cost, Nitish Chopra et al. [15] enhanced the HEFT scheduling algorithm. Their approach worked on private as well as public clouds. Primarily, proposed methodology checked the availability of resources that could finish the tasks in time. If resources were not found, then private clouds were availed to timely fulfill the requests with in the cost constraint.

Suhradam Patel et al. presented another approach to handle the particular bottleneck [16]. The authors used two different approaches to schedule the jobs. The first one is not a complex one since in this approach, no real time or time-bound tasks were considered as prerequisites. Second one was adopted for deadline based tasks, where heterogeneous servers were created and scaled up and down to execute several tasks. Not only deadline, but cost which is another parameter of scheduling, was also taken by Zong-Gan et al. in their scheduling approach [17].

Dinesh Komarasamy et al. suggested an approach to minimize the make-span while handling deadline-based tasks [18]. The methodology was introduced on three major components: job manager to job dependency resolver & increased job priority, datacenter to execute the jobs and VM creation. Their methodology first removed the tasks' dependency, filtering according to their priority and then executed the tasks. Apart from these factors, Longkun Guo et al. [19] handled deadline based tasks along with escalation in the workload of the computational resources. This increased the CPU utilization and also minimized the number of used resources.

Chien-Hung et al. suggested to use minimum Weighted Bipartite graph to handle deadline based tasks with full resource utilization in their proposed approach [18] while ILP (Integer Linear Programming) was used by Zhao-Rong Lai et al. to handle deadline based tasks [12].

Vinay et al. proposed a methodology, in which resources were auto-scaled to execute sub-tasks [14]. In their work, child tasks were checked, if they can be executed in time or not. If not, then resources were auto-scaled to finish the tasks in time.

Mohit Kumar et al. proposed the deadline based approach which was based on autoscaling and fixed deadline time. In this approach, first of all the tasks that could not be fulfilled in time are rejected by applying some conditions. If tasks' rejection rate was equal or more than 30% then 20% of VMs were added to the system. If rejection rate was 10% or more, VMs were scaled, else tasks were rejected. Along with deadline, SLA was also considered in the respective approach [15].

Traditional Round Robin (RR) was improved by Stuti Dave et al for a cloud computing environment by implementing dynamic time quantum in round robin approach [11]. TQ (Time Quantum) was calculated based on the time taken by resource round. If round is odd, then TQ would be equal to the minimum request size. In this approach, if round was even, then TQ would be the average execution time of remaining tasks. Round robin scheduling policy for balancing the load was used by Priyanka Gautam et al. in their proposed algorithm "Extended Round Robin Load Balancing in Cloud Computing" [82]. Different cloudlets' MIPS and memory size in MB were dynamically allocated by them. The tasks or cloudlets were randomly selected using the approach. Seema Verma et al. proposed an efficient algorithm EARP-RR (Earlier Account Expire Prioritized with Round Robin) for scientific community [13]. The communities that are involved in some research work and use the same type of data should use the same resource in order to make better utilization of resources at minimum cost.

In max-min scheduling algorithm, task with maximum execution time was executed first with minimum execution time resource [19] [14–16]. O.M. Elzeki et al. propounded an improved max-min algorithm [17]. Execution time may be same for some tasks, but their main focus was on completion time of tasks. Hence, in their algorithm, task that had maximum completion time, was selected instead of maximum execution time. On contrary, Upendra Bhoi et al. proposed enhanced max-min algorithm [14]. Their proposed algorithm did not select task with the maximum completion time; instead it selected the tasks which had a completion time of nearly or equivalently to average. This approach distributed the workload on servers. But 53S. Devi Priya et al. focused on time taken by resources to accomplish a task rather than on task execution time [15]. For this purpose, in their proposed work, the first resource with minimum completion time was selected and the biggest task was assigned to that resource. Santhosh B et al. proposed an improved max-min algorithm [16]. The proposed algorithm used two approaches. In the first approach, average execution time was calculated using the arithmetic mean and in the second approach, geometric mean was used. If the values were independent, then arithmetic mean gives the best average execution time, whereas if the values were dependent on the other values, then the geometric mean gives the best average execution time. The user can select anyone of these approaches based on the characteristics of his data set.

7. COMPARISION AMONG SHEDULING ALGORITHM

Table 1.2: Comparison Among Various Scheduling Algorithms.

	Completion Time (in seconds)				
No. of cloudlets/tasks	FCFS	SJF	RR	Max-Min	Min-Min
50	79	70	65	59	55
100	102	89	81	79	76
200	165	154	150	135	133
300	209	201	192	188	186
400	237	229	217	214	210

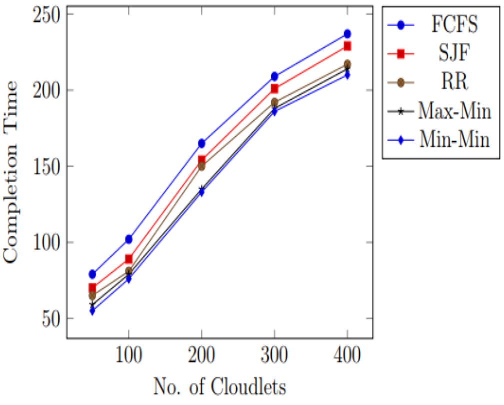

Figure 1.3 Graphical Representation of Results.

8. CONCLUSION

The past work concentrate on effective asset assignment to improve goals of cloud clients, IaaS supplier and SaaS supplier in distributed computing. The work proposes the arrangement of various layers in the cloud like IaaS and SaaS and its joint improvement for proficient asset portion. The effective asset portion enhancement issue is led by sub issues. The proposed cloud asset assignment improvement calculation is accomplished through an iterative calculation. Another assignment booking calculation for running huge projects in the cloud. Most ordinary errand booking calculations don't think about money related costs, and so they can't be straightforwardly applied in a cloud setting. In this work, our calculation processes booking plans that produce spread the word about length as great as the best calculation of while altogether diminishing money related costs. For the future Virtualization-based full-framework estimation and observing apparatuses are likewise included to help with utilizing the proposed framework for co-plan of elite execution registering framework programming and compositional highlights.

9. REFERENCES

[1] Wikipedia, "Scheduling in Production," 2013 (accessed in 2019). Available at https://en.wikipedia.org/wiki/Scheduling_(production_processes).

[2] C. Donnelly, "A History of Cloud Computing," April 2018 (accessed in 2019). Available at https://www.computerweekly.com/feature/A-history-of-cloud-computing.

[3] S. Garfinkel, "The Cloud Imperative," 2011 (accessed in 2019). Available at https://www.technologyreview.com/s/425623/the-cloud-imperative/.

[4] A. Incorporated, "History of Grid Computing," (accessed in 2019). Available at http://www.avarsys.com/grid_computing_history.html.

[5] R. K. Chellappa, "Goizueta Business School," (accessed in 2019). Available at https://goizueta.emory.edu/faculty/profiles/ramnath-k-chellappa.

[6] A. W. Services, "AWS Documentation," (accessed in 2019). Available at https://docs.aws.amazon.com/index.html?nc2=h_ql_doc.

[7] Wikipedia, "Google Code-in," (accessed in 2019). Available at https://en. wikipedia.org/wiki/Google_Code-in.

[8] Microsoft, "What is Azure," (accessed in 2019). Available at https://azure. microsoft.com/en-in/overview/what-is-azure.

[9] L. Peng, "Cloud computing," Publishing House of Electronic Industry, 2010.

[10] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop, vol. 5, Jan. 2009.

[11] M. Naghshineh, R. Ratnaparkhi, D. Dillenberger, J. R. Doran, C. Dorai, L. Anderson, G. Pacifici, J. L. Snowdon, A. Azagury, M. VanderWiele, and Y. Wolfsthal, "IBM Research Division Cloud Computing Initiative," IBM Journal of Research and Development, vol. 53, pp. 1–10, July 2009.

[12] R. L. Grossman, "The Case for Cloud Computing," IT Professional, vol. 11, pp. 23–27, March 2009.

[13] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," tech. rep.,National Institute of Standards and Technology, Sept. 2011.

[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," tech. rep., Feb. 2009.

[15] Gartner, "IT Glossary," (accessed in 2019). Availavle at https://www.gartner.com/it-glossary/cloud-computing.

[16] G. Perry, "The Open Cloud Manifesto: A Call to Action for the Worldwide Cloud Community," March 2009 (accessed in 2019).

[17] T. Ride, "Cloud Computing Architecture," (accessed in 2019). Available at https://www.tutorialride.com/cloud-computing/cloud-computing-architecture.htm.

[18] R. Buyya, J. Broberg, and A. Goscinski, Cloud Computing Principles and Paradigms. John Wiley & Sons, 2011.

[19] Wikipedia, "Utility computing," (accessed in 2019). Available at https://en.wikipedia.org/wiki/Utility_computing.

[20] M. Rouse, "Cloud Automation," Aug. 2017. Available at https://searchcloudcomputing.techtarget.com/definition/cloud-automation.

[21] W3C, "Simple Object Access Protocol (SOAP)," 2004 (accessed in 2019). Available at http://www.w3.org/tr/soap/.

[22] "OASIS UDDI Specification TC," 2017 (accessed in 2019). Available https://www.oasis-open.org/committees/uddi-spec/.

[23] W3C, "Web Service Description Language (WSDL)," June 2007 (accessed in 2019). Available at http://www.w3.org/tr/wsdl/.

[24] T. Laszewski and P. Nauduri, "Migrating to the Cloud," pp. 1 – 19, Boston:Syngress, 2012.

[25] M. F. Kacamarga, B. Pardamean, and H. Wijaya, "Lightweight Virtualization in Cloud Computing for Research," in Intelligence in the Era of Big Data, pp. 439–445, 2015.