International Journal of Advances in Computer Science & Its Applications – IJCSIA 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 8 : Issue 2- [ISSN : 2250-3765] - Publication Date: 28 December, 2018

A comparative study of Service Brokers scheduling policies in Cloud Computing

Prof. Najib A. kofahi

Tariq Alsmadi

***Abstract*— The growing importance of the cloud has attracted many researchers to look for possible improvements and contributions in the field. Their initiations were driven either by personal interests or funded by enterprises. Moreover, Cloud Computing is continuously evolving to meet users' needs and expectations which make it even more attractive. And one of the most interesting research topics is the service brokerage scheduling policy, because of its importance as a routing policy, which could form a bottleneck in the process. This article is a comparative study and analysis on some of the previously done work in the field; it fully explains the role of the service broker scheduling policy in the cloud and summarizes all the work in a single resource for researchers. The advantages and drawbacks are highlighted in terms of factors involved in the decision-making process. By the end of this article, a researcher will be able to formulate own theories about possible areas of improvements, where we have found that having a fully optimized service broker policy is very important in the cloud. Possible research approaches are also suggested in the summary and findings.**

***Keywords*— Cloud Computing, Service Broker, Job scheduling, Load Balance, Data Centre Selection, Static scheduling, Dynamic scheduling**

I. Introduction

Cloud computing (CC) is the new era of computer technology. It is rapidly dominating the Information Technology (IT) field, which made many organizations move toward CC, whether it is employed as a complete solution or a partial special-purpose solution. Scalability, availability, and portability are features that come seamlessly with the nature of CC and are big motives for technology adaption. In CC everything is provided as a service that is metered and charged to the customer in a 'pay as you go' manner, which makes this technology a preferable solution for some organizations. But it can be a source of disturbance for others depending on their needs and usage patterns. Also, there are some challenges imposed by the highly demanding communication requirements of the cloud, especially when there is relatively large data size involved (i.e. Backup operations). All of the previously mentioned reasons bring the need to have an efficient and effective usage of the available resources to maximize utilization and minimize costs [1, 2].

Prof. Dr. Najib A. kofahi
Faculty of Information Technology and Computer Sciences/
Yarmouk University
Jordan

Tariq Alsmadi
Computer Science Department / Yarmouk University
Jordan

CC works mainly by allocating the user job to a server in the cloud, this job could be computationally/data intensive, or even both which means that a smart choice must be made with every new job to allocate the job to the proper server. The jobs and the servers in a CC environment both have continuously variable states. For example, a server can be congested with highly demanding computational jobs in one minute but might be free at another. The same thing goes for jobs submitted by the users because users have different needs and demands, this problem is already known as an NP-Hard problem; which imposed a challenge for researchers motivated by the importance of CC emergence [3, 4].

Many research works were conducted to improve CC efficiency by building an efficient Service Broker (SB) policy, which is the topic of this paper. This paper studies and analyzes the most famous and recent work in the field, to sum up all of the previously done work in one paper that provides single stop and a useful resource for future researchers.

To be able to fully understand the scheduling dilemma in CC we must first have a closer look at the CC itself and get to know how this environment works. There are different CC technologies based on the services they provide and the users' scope. The next subsection explains in detail the various types of CC.

A. *Types of CC*

CC was mainly divided based on the provided services and the users' scope. For example, private clouds provided special and dedicated service that suite the needs of the owner organization. While public clouds provide diverse services to support diverse users and their needs. The latter kind of clouds is owned by organizations that provide IT services to the public such as Amazon. Figure (1) portrays these types.

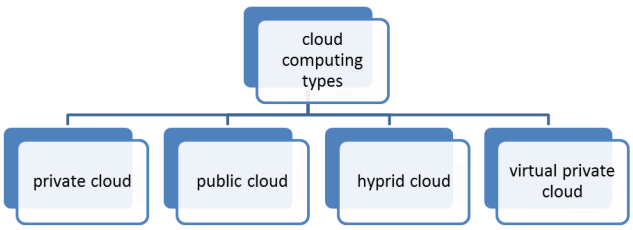

Figure 1. CC types

According to Figure (1) the CC is mainly divided into:

1) **Private cloud:** also called internal cloud, this type of clouds is constructed by building a cloud from own organization infrastructure, which is usually used to support the enterprise mission. In this case, usually, the main mission is not IT services but it is used to support and maintain the mission. The organization Data Center (DC) is used to form a cloud to build on and provide the needed services, these services could be only for on-premises usage or for an external customer. Furthermore, to elaborate on

this let's consider a bank with its own private cloud that provides services to customers for online banking. Private clouds, in general, are expensive and more useful for large enterprises than to medium or small sized organizations, they are most useful when privacy and confidentiality are most important by keeping data store on premises within the firewall [5].

2) **Public cloud*:*** Unlike private cloud, this type of cloud is mainly dedicated to providing cloud services as cloud service providers, where such enterprises are mainly focused in mission in providing IT services. An internet connection is used to access the cloud services. This type of cloud usually provides full services such as infrastructure, application, and platform to the public as pay-per-use services, where users have to pay only for their usage time or/and size. This type of cloud is economical and inexpensive for the users, because they don't have to own special expensive hardware (HW) (Servers, Firewalls ...etc.), which is not just an initial cost because having such HW requires having a special dedicated place with special requirements (Air conditioning, Un-Interrupted Power supply …. etc.) and the needed professional manpower to maintain and support the services. This type of clouds is mainly interesting for enterprises with no sensitive data that has no problem storing its data on the cloud [5].

3) **Hybrid cloud:** This type of cloud is a combination of public and private clouds, it was introduced to avoid the limitations in both (public and private clouds). In this type, an organization depends on private cloud for sensitive services, while it depends on the public cloud for the non-sensitive. The two main benefit of this type is the flexibility of applicability and still able to maintain data confidentiality [5].

4) **Virtual private cloud:** This type was mainly introduced to address the limitation of private cloud requirements, where a private cloud can be built upon a public cloud by using the Infrastructure service from the cloud to formulate a private cloud with a customized architectural design. Such as Firewalls and network specifications [5].

5) **Community Cloud:** In this type of cloud several organizations with common interest and requirements cooperate to share the same cloud infrastructure [5].

B. *The scheduling Dilemma*

The task scheduling (aka allocation) problem has been around since the early beginnings of the computer era. It aroused when computers started to evolve and the tasks need to be run became more complex, also users' expectations and demands started to rapidly inflate. It was shown when one computer was supposed to execute multiple tasks at a time, which required effective and efficient scheduling mechanism for these tasks to be able to run concurrently on a single processor machine. Later on, multiple processors were needed for more efficient scheduling but more problems were also present because more processors mean more complex task scheduling mechanisms that need to be developed. This problem is already known as an NP-Hard problem [3, 4]; which imposed a challenge for researchers motivated by the importance of computer technology emergence. Research works were conducted to design and improve scheduling algorithms to solve the problem. Most of these solutions were conducted based on the internal computer architecture, on one hand, they considered factors like available resources (processors, memory … etc.) and current requirement on the other hand. The scheduling problem is very similar to the service brokerage problem under study, where the scheduling principle is still the same but with different considerations to be counted for. The cloud nature imposes new challenges for the scheduling problem, one of these challenges is the network specifications which greatly impacts the performance. Especially that the cloud connectivity goes through variable and different communication channels with different specifications. For example, a user could be using a high-speed fiber optics connection that goes through different Internet Service Providers (ISPs), each one of them could be providing different connectivity specifications. As we shall see later on in this paper the different network specifications have a great impact on users' tasks processing time, which could positively or negatively be reflected on the cloud performance, efficiency, and effectiveness. This called for the need to start searching the problem to find efficient and effective solutions.

C. *The Service Brokerage Anatomy*

Going more in-depth into the service brokerage problem, we can notice the resemblance with the scheduling problem; the logic is still the same. We need to have maximum utilization. But let's consider some of the differences, in the cloud there is always multiple users with multiple tasks that needs to be performed and scheduled on the cloud, the cloud also has globally spread resources over multiple geographical locations each of which could have different physical specifications due to the fact that different geo-locations properly has different communication specifications and also physical servers providing the cloud services could have different physical specifications. All of this called for the need to create a fair point of distribution to route user jobs to appropriate cloud resource, this became known as one of the main tasks of the SB. The SB is the first point of contact for all cloud users when any user requests a task from the cloud the first point that receives the user task is the SB. Then the SB makes the decision to route the user's task to the appropriate cloud resource. The cloud resource here is referred to as Data Center (DC), the DC is a logical grouping of multiple physical servers or processing units that usually share the same physical location.

D. *More on the Service Brokerage importance*

The cloud services are provided in a pay-as-you-go manner through three main models IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service). Figure 2 from [6] depicts these services.

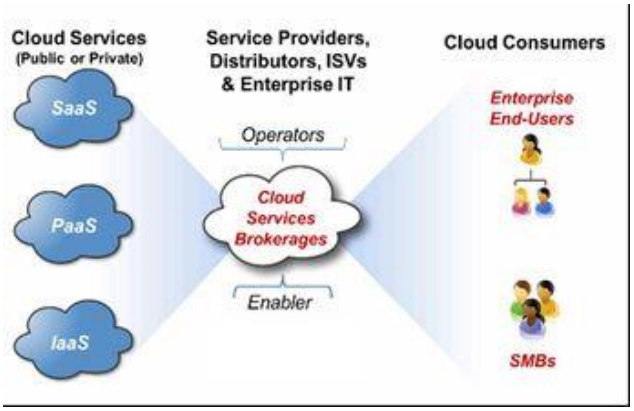

Figure 2. Cloud Services

Having these services provided to the user in a pay-as-you-go manner means that these services should be perfectly utilized, in order to minimize the overall cost incurred by the user and maximizes user satisfaction, through fast and reliable task processing. It all starts and ends at the SB since it is the first point receiving the user's task and the last point responsible of delivering the result to the user after the task has been processed in the selected DC.

E. *The Virtualization Technology*

The cloud provides services by mainly sharing physical resources through Virtualization Technology. It employs an efficient use of resources, where one physical hardware unit (single server), that can be utilized to run multiple Operating Systems (OSs) concurrently. Each OS acts as a standalone independent server, while it is sharing the same physical resources with other OS's. Also, each OS is called a Virtual Machine (VM) that has its own reserved resources (CPU, RAM, Storage, and Networking). Another scheduling level takes place in the VM level, whereas mentioned earlier the service broker routes users' tasks to the appropriate DC, which is a group of physical servers that is used to run multiple VMs. These VMs needs also to be utilized correctly. Here the scheduling level is called Load Balancing (LB), even though this level functionality is very similar to the SB functionality but still has different factors to be considered. However, the LB functionality is outside the scope of this paper, but we have gone through some of the work to benefit from a certain aspect, which will be shown later on in section (II).

So far, we have introduced in brief words the high-level abstracted model of the Cloud Computing environment as matter as what concerns this paper, the rest of this paper is organized as follows. Section (II) explains and discusses the SB problem in the previously done research work in the field. Section (III) summarizes the whole paper and shows possible research areas that need further improvements.

II. LITERATURE REVIEW

The CC is becoming the choice of many organizations in employing IT services, because of the many profitable factors that come with the nature of CC, like high availability and portability. These two factors come along by default with the CC since the CC high availability is guaranteed by the service provider through high hardware redundancy and professional manpower. While the portability is a valuable factor comes naturally from the fact that CC is available and accessible from anywhere in the world as long as an adequate internet connection is available. Portability could be very crucial depending on the service being provided (i.e. Email services) or the organization work requirements. For example, an organization that has many sites spanning globally geographical locations.

The importance of CC attracts many researchers, whether it is sponsored solely by individuals or by profitable organizations to improve their services. In this paper, we will discuss some of the most important and recent researches in the field. We will be analyzing the designed algorithms by illustrating the implemented methodology and showing their weaknesses and their strengths.

The service broker name came from the fact that it acts on behalf of the user as an intermediate between the user and the cloud by providing many functionalities, such as data marshaling in heterogeneous networks. But in this research, we mainly focus on the job scheduling problem, which is an important factor in the cloud, because it is concerned with resource utilization and reliability. These two factors are very important in the cloud because an optimized resource usage means less cost to the user and higher dependability at the same time. Early CC just like any other technology started in a less complex environment and fewer demands. The more the technology got evolved the more requirements and demand were needed to meet users' expectations, which also are increasing with the rapid application of CC. For example, jobs submitted by users were more defined and relatively small, but now they are becoming more complex and increased in size. A good example for this would be using cloud storage (Google Drive, DropBox …etc.) for personal data backup. These storage solutions provided storage ranges from 1GB to 2GB, which was more than enough to store and backup user's files. But now it is very small for partial backup. The previously mentioned reasons elaborated the need to have a scheduling policy that's aware of the cloud resources on one hand and the user's needs on the other hand.

Scheduling policies in the cloud can be categorized into two categories based on their approach to dealing with the available resources and load distribution [7]. However, we must elaborate that this categorization does not actually fit in case of the service broker policies, because most service broker policies fall under a static and dynamic classification. So, we have made our own classification mainly based on our understanding of the methodologies of the previous work.

The next two subsections explain and show the previous work pros and cons.

A. *Static scheduling*

This approach depends on values previously defined prior algorithm initialization. These values could specify task execution sequence in priory or may be pre-set values of available resources such as DCs specifications (Processing power, Ram ……. etc.). This type of scheduling could be useful for special cases but in case of CC, the environment is always changing where a whole DC can go down, get overloaded. Also, a communication channel can get congested and became unstable. An effective scheduler should be able to detect such situations and deal with them.

However, a closer look should be made on some of the successful and famous works in the field in order to highlight successes and drawbacks, which might shorten the way to future research.

The followings are some of the static schedulers and their pros and cons:

1) **Proximity Service Broker:**

One of the very first used policies was the Proximity Service Broker (aka Closes DC) [8, 9], this policy worked in a simple manner, where every job is allocated to the closest DC based on the network delay. The list of available DCs is sorted according to their proximity, and the closest DC is selected for the job allocation. In case of more than one DC are available with same network delay, one of them is selected randomly for each new job.

- Pros: This policy is suitable for small sized jobs with minimum data transfer requirements because small jobs don't require high BW and usually less computationally demanding. So it would be ironic to select a DC with a communication channel that has relatively high delay time in regard to the transmission or processing time.
- Cons: The main drawback of this policy is that it doesn't consider any factors other than the network delay. Also, repeatedly selecting the same DC (closest DC) might overload the DC, while other DCs are idle. Moreover, the cost is not considered at all.

2) **Mishra, Kumar [10]:**

They proposed an enhancement on the Proximity Service Broker, by avoiding the random selection of DCs with same network delay. Their contribution was to use a round-robin algorithm to select DCs based on their characteristics.

- Pros: The technique showed an improvement since it considers DCs specifications.
- Cons: This approach still suffers from the same drawbacks in the Proximity technique.

3) **Kapgate [11]:**

The author proposed a policy similar to (Mishra et al., 2014) policy, where they used a weighted round robin policy for selecting DCs with same network delay. They gave every DC a weight based on processing capability and cost.

- Pros: The authors claim to enhance processing time and cost.
- Cons: The authors failed to explain how they used the weighted round robin to give weights to DCs, they only settled with stating they sued it.

4) **Chudasama et al. [12]:**

They proposed an enhancement of the Proximity policy by adjusting the random selection of DCs with the same network delay to become based on cost, where the least cost DC is selected to improve the cost-effectiveness.

- Pros: The technique showed an improvement in the cost factor.
- Cons: This approach doesn't consider the DCs specifications, so it will select a DC with least cost even if it is less capable than other DCs with slightly higher cost, which might greatly impact the overall processing time.

5) **Ahmed [13]:**

The author proposed another enhancement over the Proximity policy. While the Proximity chooses the closer data center only, this policy chooses the neighboring DC in addition to the closes DC to distribute the load over multiple DCs.

- Pros: The technique showed an improvement in the overall processing time since jobs are being more distributed.
- Cons: This approach doesn't consider the DCs specifications, cost or the network BW, so it will select the closest DCs with disregard to BW or cost.

6) **Flextic:**

Henzinger et al. [14] proposed Flextic as a static scheduling model with user intervention required. It works by first having the user writing the program for the jobs in "Flextic job description language" and specifying job characteristics, like maximum execution time. Then the program is passed to a directed-acyclic-graph to produce an execution plan, after that, a static scheduler is used to compute the possible schedules. Finally, the user has to choose from these schedules what best suits his needs in terms of execution time and cost. The schedule selected by the user is allocated and after execution, the results are returned to the user.

- Pros: The authors used a real environment on Amazon EC2 cloud to evaluate Flextic in comparison to Hadoop [15]. Due to the high communication overhead in Hadoop, Flextic outperformed Hadoop. And as a static scheduler, it's a benefit to having the user decide what scheduling scheme is best suited for him.
- Cons: We only introduced Flextic to show that static schedulers are not suitable for real-time clouds. And employing such techniques will have a negative effect since the cloud is always changing. Also, the users' needs are always changing. One more drawback in Flextic is that there is an unconsidered offline overhead in building the execution plan and the time needed for user intervention.

B. Dynamic scheduling

This type of schedulers is mainly the used approach in service brokerage paradigm, because of the ability to satisfy the problem needs, where a real-time evaluation of the resources and the jobs need is done in a dynamic manner. The following techniques are examples of the previous work with an explanation of their methodologies and their pros and cons:

1) **Performance Optimized Routing:**

This technique is an enhancement on the original basic Proximity Service Broker. It was an actual improvement because of the dynamicity in resource evaluation, where this technique works by accounting for current DCs load. It also monitors their performance by tracking last job execution time in each DC, where a relative increment in this time is an indication of overloading. So whenever an overload situation is detected a time period known as "Cool_Off_Time" is waited before allocating new jobs to the corresponding DC.

- Pros: This technique showed a noticeable improvement in regard to the original Proximity Service Broker. Especially in case of multiple DCs available with widespread regions, because it means distributing the load across multiple DCs rather than sending them to single DC like the Proximity Service Broker, which also might reduce the overall processing time.

- Cons: There is a shown wasted time during the "Cool_Off_Time" period. Because an increment in processing or response, not necessarily an overloading, it could be due to different job requirements. One more drawback is that this technique doesn't consider the cost.

2) **Dynamically reconfiguring policy:**

This policy is a multi-level scheduling where the number of VMs is increased or decreased to minimize the load [8].

- Pros: The technique showed an improvement in the overall processing time.

- Cons: The cost is not considered.

3) **Jayarani et al. [16]:**

They proposed a full scheduling system that uses a two-level of scheduling. One level is on the broker level while the other on the VM level. They used Backfilling scheduling technique where smaller jobs are given higher priorities to be executed as long as they don't affect other jobs [17]. The authors claim to enhance the Cloudsim broker policy which is a random selection policy [18] to become based on the number of processors per DC. They also introduced other improvements on the VM scheduling level by using Intra VM scheduler that implements "conservative backfilling strategy" that deals with different scheduling cases namely "regular dispatch, backfill, and backlog". No further explanation was provided by the authors about these cases implementation.

- Pros: The technique showed an improvement in the overall processing time.

- Cons: The authors stated that they were intending to enhance the cost but it wasn't mentioned in the evaluation and testing phase. They also used a fixed task size (100000 MIPS) to categorize jobs to small and big jobs, which is usually outside the scope of the service broker to analyze and break down jobs because otherwise, it might cause an overhead and more delay.

4) **Variable Service Broker Routing Policy (VSBRP):**

Manasrah et al. [19] proposed variable service broker routing policy that reduces the overall needed response time to execute the jobs. Their work was an enhancement over the Proximity and the Performance Optimized Routing policies. They enhanced Proximity Broker by using the availability ratio, which is the ratio of network delay to the network Bandwidth. It was used as an indication of network availability instead of just using the delay. They also enhanced the Optimized Routing by eliminating the "Cool_Off_Time" and using the next job processing time instead of the previous job, which gave a more realistic expectation of DCs allocation status. Their technique works by sorting the list of available DCs according to the availability ratio, and when a new job is received, if it is less than 10KB in size, it is routed to the closest DC as in the Proximity Broker. Otherwise, it is routed based on the expected response time (transmission + execution), where it will be routed to the DC with least response time to minimize the overall processing time.

- Pros: The technique showed an improvement in reducing the overall response time and minimizing DCs load. One more enhancement was noticeable with big sized jobs where the BW factor has a higher impact.

- Cons: This approach used a fixed size (10KB) to be considered as small jobs, while an optimization technique should have been used to dynamically consider what is relatively small or big. Also, this approach had no consideration for the DCs cost.

5) **Load Balancing Ant Colony Optimization (LBACO):**

Li et al. [20] implemented the Ant colony algorithm [21] as a heuristic task scheduling in the VM level. Their work depended on a dynamic and real-time evaluation of the available resources (VMs), by using the Ants approach in searching for food sources in the real life. They also used objective functions to optimize the scheduling process. Their algorithm works by sending test jobs to available VMs at initialization to perform dynamic evaluation and give an initial Pheromone trail value to available VMs, where the pheromone trail is a chemical material left by ants on trails leading to food sources. The concentration of this trail is an indication of the food availability and proximity in regards to the ants' nest. And in the scheduling mechanism, it is used to indicate VMs preference. The VM with highest trail value is selected for next job scheduling and so on.

- Pros: The technique showed an improvement in reducing the overall response time and VM utilization.

- Cons: This approach had no consideration for the cost and network state.

6) **Kai Li et al. [22]:**

The authors approached the problem as an optimization problem by using an objective function that optimizes the factors involved in the decision-making process. Their approach was very interesting except for the fact they used theoretical equations to evaluate the available resources (DCs, Network) and optimize their usage.

- Pros: The technique showed an improvement in reducing the overall response time and minimizing DCs load.

- Cons: This approach had no consideration for the DCs cost.

The following table summarizes all the techniques discussed earlier and shows their accountability to the factors involved in the selection process.

TABLE 1. LITERATURE SUMMARY

Policy	**Factors involved in the election process**				
	Bandwidth	***Network Delay***	***New job load***	***DC allocation Status***	***Cost $***
Service proximity	No	Yes	No	No	No
Performance Optimized	Yes	Yes	No	Yes	No
Dynamically reconfigure	No	Yes	No	Yes	No
Weighted Round-Robin	Yes	Yes	No	Yes	No
Kapgate (2014)	Yes	Yes	Yes	No	No
Mishra et al. (2014)	No	Yes	No	No	No
Kai Li et al. (2014)	Yes	Yes	Yes	No	No
VSBRP	Yes	Yes	Yes	Yes	No
LBACO	Yes	No	Yes	Yes	No
Jayarani et al. (2009)	Yes	No	Yes	Yes	No
Flextic	No	No	Yes	Yes	Yes
Ahmed (2012)	No	Yes	No	No	No
Chudasama et al. (2012)	No	Yes	No	No	Yes

III. SUMMARY AND FINDINGS

The fast emergence of CC in various computational fields imposed a challenge on the technology, where users' needs and expectations are rapidly increasing. Users need to have highly reliable, efficient and cost-effective solutions.

The previous sections through the article showed the importance of having a fully optimized service broker policy that counts for all the factors involved in the DC selection process. Using improper policy might form a bottleneck in CC. This bottleneck could cause serious issues regarding the user and the service provider at the same time. This is since miss using the resources could be a waste of user time or/and money. Also, it can be a waste for the service provider resources, and a possible performance degradation factor. So, the service broker policy needs to be fully optimized by developing techniques capable of counting for all the factors and still computationally economic with no imposed overhead in the process.

Moving toward using dynamic optimization techniques seems a possible solution to the problem. Also employing Heuristic techniques can improve the dynamicity of the process.

References

[1] Subashini, S. and V. Kavitha, A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications, 2011. 34(1): p. 1-11.

[2] Xu, X., From cloud computing to cloud manufacturing. Robotics and computer-integrated manufacturing, 2012. 28(1): p. 75-86.

[3] Dorigo, M., M. Birattari, and T. Stützle, Ant colony optimization. Computational Intelligence Magazine, IEEE, 2006. 1(4): p. 28-39.

[4] Reeves, C.R., Modern heuristic techniques for combinatorial problems. 1993: John Wiley & Sons, Inc.

[5] Zhang, Q., L. Cheng, and R. Boutaba, Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications, 2010. 1(1): p. 7-18.

[6] Wiki, M.M., What Is a Cloud Service Broker. 2016.

[7] Sidhu, A.K. and S. Kinger, Analysis of load balancing techniques in cloud computing. International Journal of Computers & Technology, 2013. 4(2): p. 737-41.

[8] Limbani, D. and B. Oza, A Proposed Service Broker Policy for Data Center Selection in Cloud Environment with Implementation. International Journal of Computer Technology & Applications, 2012. 3(3).

[9] Sharma, V., Efficient Data Center Selection Policy for Service Proximity Service Broker in CloudAnalyst. 2014.

[10] Mishra, R.K., S. Kumar, and B. Sreenu Naik. Priority based Round-Robin service broker algorithm for Cloud-Analyst. in Advance Computing Conference (IACC), 2014 IEEE International. 2014.

[11] Kapgate, D., Efficient Service Broker Algorithm for Data Center Selection in Cloud Computing. International Journal of Computer Science and Mobile Computing, 2014. 3(1).

[12] Chudasama, D., N. Trivedi, and R. Sinha, Cost effective selection of data center by proximity-based routing policy for service brokering in cloud environment. International Journal of Computer Technology & Applications, 2012. 3(6): p. 2057-2059.

[13] Ahmed, A.S., Enhanced proximity-based routing policy for service brokering in cloud computing. International Journal of Engineering Research and Applications, 2012. 2(2): p. 1453-1455.

[14] Henzinger, T.A., et al., Static scheduling in clouds. memory, 2011. 200(o1): p. i1.

[15] Apache. Apache Hadoop. 2014; Available from: http://hadoop.apache.org.

[16] Jayarani, R., S. Sadhasivam, and N. Nagaveni. Design and implementation of an efficient two-level scheduler for cloud computing environment. in Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on. 2009. IEEE.

[17] Jackson, D., Q. Snell, and M. Clement. Core algorithms of the Maui scheduler. in Workshop on Job Scheduling Strategies for Parallel Processing. 2001. Springer.

[18] Calheiros, R.N., et al., CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 2011. 41(1): p. 23-50.

[19] Manasrah, A.M., T. Smadi, and A. ALmomani, A Variable Service Broker Routing Policy for data center selection in cloud analyst. Journal of King Saud University-Computer and Information Sciences, 2016.

[20] Li, K., et al. Cloud task scheduling based on load balancing ant colony optimization. in 2011 Sixth Annual ChinaGrid Conference. 2011. IEEE.

[21] Dorigo, M. and C. Blum, Ant colony optimization theory: A survey. Theoretical computer science, 2005. 344(2): p. 243-278.

[22] Kai Li, Yong Wang, and M. Liu, A Task Allocation Schema Based on Response Time Optimization in Cloud Computing. arXiv, 2014. 2: p. 1-19.

About Author (s):

Najib A. Kofahi is a professor of computer science at Yarmouk University (YU) since 2006. He received his Ph.D. in computer science from the University of Missouri-Rolla (USA) in 1987. He was granted scholarship for MSc. and Ph.D. from YU in 1981. During his stay at YU he was the vice dean and dean of the Faculty of Information Technology and Computer Sciences for several years. He also worked as the chairman of the computer science department during the years 1990-1992. He has several international journal and conference research publications in a number of research areas. His research interest focuses on Cloud Computing, Operating Systems, Performance Evaluation and computer & data security.

Tariq Alsmadi has an Msc in computer science and an experienced IT tech with CC and networks and information security interests.
"Moving toward using dynamic optimization techniques seems a possible solution to the service broker problem".