Rate aware Meta task Scheduling Algorithm for multi cloud computing (RAMTSA)

BJ Hubert Shanthan¹, Dr. L. Arockiam²,³

¹ Research Scholar, hshanthan@gmail.com
² Associate Professor
³ Dept. of CS, St. Joseph’s College, Trichy

Abstract. Multicloud attains a larger momentum by gratifying the needs of the more number of heterogeneous cloud users. The inherent parts that are used in the multicloud systems are analogous, scheduling and rescheduling. In the analogous aspect, requests from the first user are given more priority and tasks are delegated to the virtual machines. In the scheduling phase, round robin based mechanism is used to accomplish the tasks in all the clouds. Rescheduling is a mechanism of shifting the executed tasks of one cloud to the another cloud. The expenditure of execution of the tasks around 15% are merely reduced in the rescheduling phase. This article is merely focused to dwindle the execution costs of the VMs in the clouds. The denouement of the propounded algorithm manifests better when it is compared with conventional min-min algorithm.

1. Introduction

In the neoteric evolution, cloud plays a predominant portrayal in the ubiquitous computing environment. The NIST [1] defines “Cloud is an exemplary model which provides the services to the consumers of the cloud rapidly with minimum interaction of the service provider”. The main intention of the IaaS service providers is to hike the profit and to appease the cloud clients [2][3]. In the IaaS service model, the cloud consumers will merely focused their requests only for storage and computation. Memory and processing speed are key constraints in the IaaS Service model. These constraints can be achieved only through the VM. The conflict arises, when bulk requests from the users for contradictinctive types of servers for storage and computation[4][5]. In the multicloud system there is no proper scheduling strategy is devised to execute the tasks in the VMcs. The scheduling conflicts arises, whenever the user demands increase and the data centers are not able to grasp the user requests [6][7]. Therefore, there is a need for migration of the user requests to other service providers or other cloud data centres. This process may lead to data loss or more expensive to transfer the client appeals from one cloud to another cloud. So to solve this problem, there is a need for a cardinal cloud manager to control the other producers of the cloud. Multicloud is a cloud service model consists of one or more service producers of the cloud. Multicloud environment consists of centralized cloud manager and it dominates all the non identical cloud service providers[8]. The individual producers of the cloud has their own scheduling policy and the job of the cardinal manager must follow appropriate scheduling policy to trace the tasks with the virtual machines in other clouds.

This article focused on a contemporary Rate Aware Meta Task Multicloud Scheduling Algorithm(RAMTSA) to decrease the makespan and the execution cost of the resources. This algorithm addresses the task based scheduling problem in a multicloud environment. The research article is formulated based on the sections as follows. The related works of the existing literature are reviewed in the Section 2. The proffered Multicloud model and algorithm are
elaborated in the section 3. The insights of the RAMTSA algorithm are represented in the section 4. Eventually, section 5 includes the closure with forthcoming research directions.

2. Related Works

Panda et al. [9] proposed algorithms for batch processing multicloud systems. Matching, allocation and scheduling are the three phases used in this algorithm. Matching phase is more habituated to match the tasks to the high speed virtual machines. Allocation phase, is more accustomed to assign the matched tasks to the virtual machines and it is an intermediary layer between allocation and scheduling phase. Scheduling phase, is more familiar in executing the tasks to the virtual machines, were all clouds are given due importance in this approach.

Kokilvani et al. [10] devised an algorithm to balance the virtual machine and to impoverished the maximum completion time of the tasks. The first juncture the tasks with minimum execution time were selected and executed in the virtual machines with minimum completion time. Total completion time of tasks are computed as the makespan value. The computed value was used as a outset value to reschedule the tasks.

George et al. [11] proposed a client aware price based technique for a solitary cloud environment. This algorithm provides cloud resources to its consumers with economic costs. Cost, makespan, load balancing are three major metrics are concentrated in this algorithm.

Marian et al. [12] designed a flexible rate policy for a multi cloud environment. The policy increases the adequate usage of the resources by the cloud customers. The pricing strategy lean on the on-demand client requests to the server. The price values had been increased, whenever the user demands were high (overloaded) and prices would drop when there very less requests from the clients. The proposed policy was designed for the tradeoff bounded by the load balancing and cost.

Sanjay et al. [13] recommended a profit based mechanism for multicloud environment. The profit was gained for a successive completion of the user requests. The penalty was levied from the Cloud users and producers of the cloud for the aborted tasks and also for the SLA violated constraints. The results of the suggested algorithm balance maximum completion time of the resources and the execution cost.

Ibrahim et al. [14] proposed a topical pricing model in the cloud. The algorithm was designed to lower down the makespan value and minimizes the expenditure of the execution of the tasks. Amazon EC2 and Google Cloud were used as a pricing model to examine the performance of the proffered algorithm. Virtual machines were allocated to the user based on the processing power of the VM. Makespan and cost are the two metrics focused on this algorithm. The experimental results proved that the algorithm reduced the makespan value around 20% and 67% of the charged of the utilizing of resources. This algorithm can be extended for multicloud systems to reduce cost and makespan.

Thiru et al. [15] proposed three scheduling algorithms and they were (Equal Load Sharing (ELS), high priority scheduling and rate based scheduling algorithm. The ELS algorithm work only when the user requested task is equal to the virtual machines available in the other collaborative cloud. High priority scheduling algorithm are used when the user requests are larger than virtual machines, the priority was given to user tasks to execute in the desired virtual machine. Rate based Scheduling (RBS) algorithm were used, whenever user requests are smaller than virtual machines available in the multicloud environment.
Nao et al. [16] devised algorithm to assign priority for the tasks in the multicloud environment. The priority was given for the tasks based on deadline, task length and task age. Clustered Approach was used to group the user requested tasks, cloud controller, maps the cluster of user requests to the virtual machines. Cloud Controller acted as a ecumenical of all cloud components and balanced the workload of all the virtual machines on the collaborative cloud environments. Cost was precise major constraint for the clients to select the micro services from the service providers.

Julinan et al. [17] proposed a mechanism to select best service provider from the group of service providers. The choosing best suitable service was tedious in a multicloud environment. So, to overcome this disadvantage a simple additive weighted method was used as choice for choosing a best service in the multi cloud. The producer of the cloud host the multiple micro services to the users. Users were requested to select appropriate services to cater their needs. Cost was a solitary constraint for the client to select the micro services from the service providers.

3. Multi Cloud Model for Rate Aware Meta Task Scheduling Algorithm (RAMTSA)

3.1 Cloud Model
Multi Cloud is a cloud model entailed of more number of cloud service providers that are available to complete the user tasks and to captivate of the contrary clients. The Multicloud model engrossed of three key components and they are Cloud Customer (CCU), Cloud Controller Manager (CM) and Cloud Service Provider (CSP’s).

3.2 Customer (CCU):
CCU is the client user, who requests for the service provisioned by the server. The requests from CCU are treated as tasks with the assistance of the cloud controller manager.

3.3 Cloud Controller Manager (CM):
The centralized unit receives the appeals the requisition of the cloud consumers and receives the status of the virtual machines that can be accessible to the other producers of the cloud.

3.4 Cloud Service Provider (CSP):
CSP is the builder of the cloud. It acts as a producer of the resources. The physical machines that are available in the datacenter acts as abstract representation to serve distinctive orders of the users. They serve on-demand services accoutered through the virtual machines. Every cloud has administrator server and scheduler that communicates other server when the customer demands are high and they are distributed over multiple clouds. Admin server juridicts the virtual machines that are available in the cloud. Scheduler executes the task in the virtual machine of the cloud.

In a multicloud environment cost is a solitary parameter in the scheduling algorithm. Amazon EC2 [19] and Google Cloud [19] are the two billing paradigms pre-owned in this proposed algorithm. The high speed virtual machines always cost high, whereas low speed virtual machine always costs low [18]. The individual cloud has its inherent strategy to execute the tasks on the virtual machines and this process is called as scheduling phase. The Figure 3.0 clearly depicts the architecture for the multi cloud.
Scheduling phase is done in the makers of the cloud component after the scheduling process, the Cloud Controller manager executes the tasks to the virtual machines that are available in the other clouds. The tasks are executed for each cloud in the round robin basis. The computed makespan value acted as a threshold value and the virtual machines of one cloud are transferred to another cloud virtual and this mechanism is called as rescheduling. The cost of the virtual machine is computed as follows [11]

\[
Pr(V_i) = \sum_{i=1}^{n} V_i \times \text{Cost per minute} \ldots (1)
\]

Where \( Pr(V_i) \) denotes the execution cost of the virtual machine. The notation \( V_i \) denotes the virtual machines where \( 1 \leq i \leq n \).

The working procedure of the rescheduling process enables the cloud controller manager to transfer task from high speed VM to the low speed VM. The average utilization of the cloud resource (ACT) is computed by the formula stated as follows

\[
ACT = \frac{Mksp(VMj)}{Mksp(ETCK(i,j))} \ldots (2)
\]

ACT denotes the average cloud resource utilization. The \( Mksp(VMj) \) denotes the makespan value for individual virtual machine in each cloud, The \( Mksp(ETCK(i,j)) \) denotes the makespan for all clouds.

3.5 Rate Aware Meta Task Scheduling Algorithm (RAMTSA)

The primary goal of this proposed algorithm is to decrease makespan, cost improve the average resource utilization of the virtual servers that are available in the multi cloud environment.

3.5 Rate Aware Meta Task Scheduling Algorithm (RAMTSA)

Algorithm RAMTSA

**Input:** ETC matrix

**Output:** Makespan, Cost, Average Resource Utilization

Step 1: Start

Step 2: Read the values of ETC matrix

Step 3: Goto PROCEDURE RBANALOGUE

Step 4: Goto PROCEDURE RBSCHEDULER

Step 5: Goto PROCEDURE RBRESCHEDULER

Step 6: Stop

**PROCEDURE RBANALOGUE**

Step 1: Check whether queue is not empty

Step 2: Check total number of tasks are not empty

Step 3: Select the tasks with minimum completion time for each cloud

Step 4: Match the selected tasks with the virtual machines

![Figure 3.0 Architecture for Multi Cloud](image-url)
Step 5: Assign the set of tasks to the virtual machines and form a Task-VM pair
Step 7: Stop

**PROCEDURE RBSCHEDULER**
Step 1: Input the set of Task-VM pair
Step 2: Select the best Task-VM pair with minimum completion time
Step 3: Estimate the maximum completion time and execution cost of the virtual machines
Step 4: Update the ready time for all virtual machines in all clouds
Step 5: Stop

**PROCEDURE RBRESCHEDULER**
Step 1: Input Makespan and cost
Step 2: Check for all tasks available in the virtual machines in all clouds
Step 3: Sort all assigned tasks in descending order
Step 4: Select the tasks with maximum execution time in the high speed cloud server
Step 5: Check the selected task must be less or equal to than makespan else goto step 3
Step 6: Reassign the tasks in the clouds of high speed virtual machine in the other low speed virtual machine of other clouds.
Step 7: Update ready time for all virtual machines in all clouds
Step 8: Compute Makespan and cost
Step 9: Compute Average cloud utilization
Step 10: Stop

4. Results and discussion

The proposed RAMTSA algorithm is simulated by using a dataset generated by EMGEN tool [21]. The denouement the propounded algorithm are simulated by using tool called Cloudsim 3.0[20]. The Table 4.0 delineates the simulation parameters of RAMTSA.

| Parameters               | Description                      |
|--------------------------|----------------------------------|
| Simulation Tool          | NetBeans IDE (Cloud Sim 3.0)     |
| Operating System         | Windows 7                        |
| RAM                      | 2 GB                             |
| Total No of data centers | 2                                |
| Virtual machines         | 4                                |
| Processing Speed (MIPS)  | 100-1000 Ms                      |
| Total Number of tasks    | 10, 20, 50                       |
| Cost of Cloud 1(Amazon)  | $0.02 - $0.04 (per minute)       |
| Cost of Cloud 2(Google)  | $0.01 - $0.03 (per minute)       |

The Table 4.1 clearly illustrates the ETC(Expected time to compute) matrix and it is generated by the tool EMGEN [21] . The matrix values are consistent type and Low level of heterogeneity of the clouds. The Cloud C1 in the table 1 denotes the VMcs in the Amazon EC2 [19] cloud. Cloud C2 denotes the VMcs in the Google cloud [19].
The Table 4.2 neatly interprets the prices for the VMcs that are available in the clouds.

| Cloud | Virtual Machines | T1  | T2  | T3  | T4  | T5  | T6  | T7  | T8  | T9  | T10 |
|-------|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Cloud C1 | VM1             | 7   | 6   | 4   | 4   | 4   | 3   | 5   | 3   | 6   |     |
|        | VM2             | 11  | 14  | 9   | 8   | 6   | 8   | 14  | 7   | 10  | 8   |
| Cloud C2 | VM3             | 9   | 9   | 6   | 5   | 5   | 6   | 6   | 5   | 7   |     |
|        | VM4             | 13  | 15  | 10  | 8   | 14  | 15  | 10  | 11  | 9   |     |

The Table 4.3 clearly depicts the analogous phase mechanism of clouds with the VMcs.

| Cloud | Virtual Machines | T1  | T2  | T3  | T4  | T5  | T6  | T7  | T8  | T9  | T10 |
|-------|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Cloud C1 (Amazon) | VM1 (0.04 $) | 0.28 | 0.24 | 0.16 | 0.16 | 0.16 | 0.12 | 0.2 | 0.12 | 0.24 |
|        | VM2 (0.02 $)    | 0.22 | 0.28 | 0.18 | 0.16 | 0.12 | 0.16 | 0.28 | 0.14 | 0.2  | 0.16 |
| Cloud C2 (Google Cloud) | VM3 (0.03 $) | 0.27 | 0.27 | 0.18 | 0.15 | 0.15 | 0.18 | 0.24 | 0.18 | 0.15 | 0.21 |
|        | VM4 (0.01 $)    | 0.13 | 0.15 | 0.1  | 0.13 | 0.08 | 0.14 | 0.15 | 0.11 | 0.09 |

The Table 4.4 clearly presents the scheduling phase of the RAMTSA algorithm. The results show the executed tasks with the VMcs that are available in the cloud.

| Cloud | Virtual Machines | T7  | T6  | T9  | T5  | T4  | T2  | T1  | T8  | T10 | T3  |
|-------|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Cloud C1 | VM1             | 3   | 3   | 4   | 6   | 4   |     |     |     |     |     |
|        | VM2             |     |     |     |     |     |     |     |     |     |     |
| Cloud C2 | VM3             | 6   |     |     |     |     |     |     |     |     | 13  |
|        | VM4             |     |     |     |     |     |     |     |     |     |     |
The Table 4.5 clearly presents the results of the rescheduling phase.

| Cloud | Virtual Machines | T7 | T6 | T9 | T5 | T4 | T2 | T1 | T8 | T10 | T3 |
|-------|------------------|----|----|----|----|----|----|----|----|-----|----|
| C1 VM1 | 3 | 3 | | | | | | | | |
| C2 VM2 | 6 | 7 | | | | | | | | |
| C3 VM3 | 6 | 5 | 9 | | | | | | | |
| C4 VM4 | 13 | | | | | | | | | |

The table 4.6 represents the performance analysis of the RAMTSA with the existing min-min algorithm. The results in the table 4.6 show that the RAMTSA algorithm performs far better than the conventional min-min algorithm in terms of metrics such as cost, makespan and average cloud utilization.

| Algorithm | VM1 | VM2 | VM3 | VM4 | Average Cloud Utilization | Cost |
|-----------|-----|-----|-----|-----|---------------------------|------|
| Min-Min   | 22  | 19  | 11  | 17  | 0.8065                    | 2.00 $ |
| RAMTSA    | 16  | 13  | 20  | 13  | 0.8289                    | 1.65 $ |

The figure 4.0 clearly depicts the performance analysis of the RAMTSA algorithm concerning the makespan value. In the above Figure 4.0, the x axis denotes the time required to complete the tasks. The y axis in the figure clearly denotes the total number of tasks in the algorithm.
The figure 4.1 clearly illustrates that the proposed RAMTSA reduces the execution cost of the tasks for the resources. The graph plotted in x axis clearly illustrated the execution cost of the virtual machines. The y axis in the graph neatly sketches total number of tasks.

5. Conclusion

In this article, the broached mechanism comprised of three phases, namely analogous phase, scheduling phase and rescheduling phase. These phase enable this algorithm to fit into the multi cloud environment. The observatory appraised matrix for this mechanism are tested on fabricated datasets generated from the EMGEN tool. The observatory outcomes justify that his propounded algorithm outplays conventional min-min algorithm with reference to the makespan, cost and average cloud resource utilization. The RAMTSA algorithm must be tested on the original multicloud environment. The future works focus on other measuring parameters such as fault tolerance, uncertainty tasks, security, energy. Meta heuristic approaches such as Particle Swarm optimizations and evolutionary algorithms are needed to ensure scalable and reliable multi cloud service to the clients.

References

[1] Mell P and Grance T, 2009, The NIST Definition of Cloud Computing, National Institute of Standards and Technology, Information Technology Laboratory, Technical Report Ver. 15.
[2] Rajkumar Buyya, James Broberg and Andrzej Goscinski, 2014, Cloud Computing Principles and Paradigms, Wiley & Sons Pvt.
[3] Frederico Durao, Jose Fernando, Carvalho, Anderson Fonseka and Vinicius Cardoso Garcia, 2014, A systematic review on Cloud Computing, Journal of Super computing, Springer.
[4] Braun T D, Siegel H J, Beck N, Boloni L L, Maheswaran M, Reuther A J, Robertson J P, Theys M D, Yao B, Hensgen D and Freund R F, 2001, A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems, Journal of Parallel and Distributed computing, Vol 61, No 6, p 810–837.
[5] Victor Ion , Calin Sandru and Dana , 2014, Multi-cloud resource management: cloud service interfacing, Journal of Cloud Computing, Vol 3, No1, p 1-23.
[6] BJ.Hubert Shanthan and Dr. L. Arockiam, 2018, Resource Based Load Balanced Min Min Algorithm(RBLMM) for static Meta task Scheduling in Cloud, Int.,Conf.,on Advances in computer Science and Technology, International Journal of Engineering and Technology, Special Issue, p 1-8.
[7] Jane V A, Hubert Shanthan B J and L.Arockiam, 2018, Survey of Algorithms for Scheduling in the Cloud: In a metric Perspective, International Journal of Computer Sciences and Engineering, Vol 6, No 2, p 66-70.
[8] Fatma and Haithem , 2017, Multicloud service composition: A survey of current approaches and issues, Journal of Software Evolution, WILEY&SONS Pvt , p 1-24.
[9] Sanjaya K Panda, Indrajeet Gupta and Prasanta K Jana,2017, Task scheduling algorithms for multi-cloud systems: allocation-aware approach, Inf Syst Frontiers, Springer, p 1-17.
[10] Kokilavani T and Amalarethinam DD, 2011, Load balanced min-min algorithm for static meta-task scheduling in grid computing, International Journal of Computer Applications, Vol.20, No.2, p 43-49.

[11] George Amalarathinam D I and Lucia Agnes Beena T, 2015, Customer Facilitated Cost-based Scheduling (CFCSC) in Cloud, Elsevier, Procedia, p 660-667.

[12] Marian and Yong, 2011, Dynamic Resource Pricing on Federated Clouds, IEEE Computer Society, IEEE/ACM Int., Conf., on Cluster, Cloud and Grid Computing, p.513-517.

[13] Sanjaya K Panda and Prasanta K. Jana, 2017, SLA-based task scheduling algorithms for heterogeneous multi-cloud environment, Journal of Super Computing, Springer, p.1-33.

[14] Elhossiny, Nimeen A. El-Bahnasawy and Fatma A, 2016, Task Scheduling Algorithm in Cloud Computing Environment Based on Cloud Pricing Models, IEEE 2016 World Symp, on Computer Applications and Research, p 65-71.

[15] Thirumalaialevan C and Venkatachalam V, 2017, A strategic performance of virtual task scheduling in multi cloud environment, Cluster Computing, Springer, p 1-9.

[16] Naoufal Er-raji and Faouzia Benabbou, 2017, Priority Task Scheduling Strategy for Heterogeneous Multi-Datacenters in Cloud Computing, International Journal of Advanced Computer Science and Applications(IJACSA), Vol 8, No 2.

[17] Juliana, Dario and Fernando, 2018, Dynamic Selecting Approach for Multi-cloud Providers, Springer Nature Publishers, Springer, p 37-51.

[18] Panda Kumar Panda, Sohan Kumar Pande and Satyabrata Das, 2017, Allocation-Aware Task Scheduling for Heterogeneous Multi-Cloud Systems, ArabJ Sci Eng, p 176–184.

[19] Artan Mazrekaj, Isak Shabani and Besmir Sejdiu, 2016, Pricing Schemes in Cloud Computing: An Overview, International Journal of Advanced Computer Science and Applications, Vol 7, No 2, p 80-86.

[20] Rodrigo Calheiros N, Rajiv Ranjan Anton Beloglazov C´esar, De Rose A F and Rajkumar Buyya, 2010, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software Practice and Experience, Wiley&Sons Pvt, p 23-50.

[21] Kokilavani Thangavel, George Amalarathinam, 2013, EMGEN – A Tool To Create ETC matrix With Memory Characteristics For Meta Task Scheduling In Grid Environment, International Journal of Computer Science and Applications, Vol 2, No 3, p 1-5.