

Task scheduling Algorithms in Multi cloud Environment



Aliva Priyadarshini

Abstract: One of the most commonly used technology with massive demands in the field of distributed computing is cloud computing. Cloud computing has evolved in various forms like single cloud, hybrid cloud and multi-cloud. The evolution of cloud to handle hundred and thousands of user demands, at a time, thereby facilitating resource sharing, reduction in loss of information, elimination of data storage on server side and many many more the topic of task scheduling will be prominent in all forms of cloud computing and in distributed architecture. Here, we discuss the multiple cloud architecture and the scheduling techniques applied to evenly distribute the workload across multiple clouds. Algorithms like Cloud list Scheduling (CLS), Cloud min min scheduling (CMMS), Minimum completion cloud (MCC), Median max algorithm (MEMAX), Multi-objective scheduling (MOS) are some methods suggested in the past for finding a near to optimal solution for task allocation.

Keywords: Cloud Computing, Task scheduling, Multi-Cloud, Make span.

I. INTRODUCTION

It has been more than a decade that , the cloud computing technology has been one of the hot topics of discussion .Several research has been done and many of them are in a on-going process to ameliorate the cloud computing paradigm.[9-11] [1] Cloud terminology enables web-enabled consumers to acquire unlimited resource or services, any moment they want from any location by just paying for those services. With the help of fast Internet, companies are able to provide services that are both dynamic and scalable. Means that services are made available on demand by the service providers and also the providers are capable of providing services because of the ability to handle growing and diminishing resources[16][17]. To improve cloud reliability and performance by decreasing makespan, and lowering the total cost is the major goal of any cloud enterprise. Users on-demand services can be fulfilled within the specified deadline. There are various categories of cloud like private cloud , public cloud, and multi cloud.

Multi-cloud strategy refers to the implementation of two or more cloud system. A private cloud infrastructure is dedicated to a IT organization or some individual and provides the same features as that of a public cloud. It is built on a company's intra-net with strong control over its own resources. Where as a Public cloud infrastructure is build with a third party serviceprovider, who provides services to customers or tenants via high speed Internet. Today multi-tenancy cloud architecture is implemented in public cloud, where multiple tenants or costumers share resources located on different servers. A

Hybrid cloud system refers to the alliance of both private cloud and public cloud like Amazon Web Services, Microsoft Azure or Google Cloud Platform. It is a form of a multi-cloud. A hybrid cloud delivers the private cloud's high-security features combined with the fast connection and easy-to-access features of the public cloud.This paper is particularly focused on the concept of scheduling in multi-cloud environment, its architecture, the benefits, and limitations.

II. MOTIVATION

Cloud computing with its emerging tools, techniques ,models, is becoming a critical infrastructure, thus gratifying spontaneous , and real time user demands for resources like storage space, network, applications, platform and more. As the limit is higher , and that must be contented within the specific deadline, it must be considered that due to huge number of user requests, the load on a single data center increases, which leads to vendor lock-in, increased makespan, lowered throughput and reliability .The scenario demands the distribution of the consumer request across multiple data centers[15]. .

III. MULTI-CLOUD MODELS

A multi-cloud model constitutes of two or more cloud systems connected with each other via Internet. Researchers in the past have assumed few multi-cloud models as per their scheduling requirements. In [3],[12] the cloud model is assumed that at every data center there will be a manager server as in Fig 1 which will have all information about its own Virtual machines including their current status. The other responsibility of a manager server is to keep track of resource characteristics available at other cloud by communicating with the manager servers of other clouds. So when a application request arrives at a cloud, its manager server divides the application into tasks, and decides which cloud will execute which task depending upon the resource availability of the other cloud. It distributes the tasks among those cloud with ready and available resources.

Manuscript received on February 10, 2020.

Revised Manuscript received on February 20, 2020.

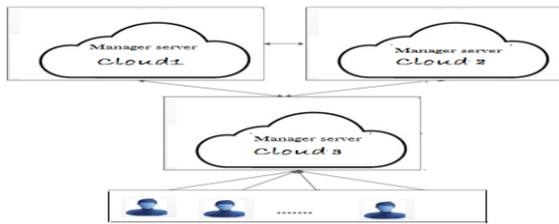
Manuscript published on March 30, 2020.

* Correspondence Author

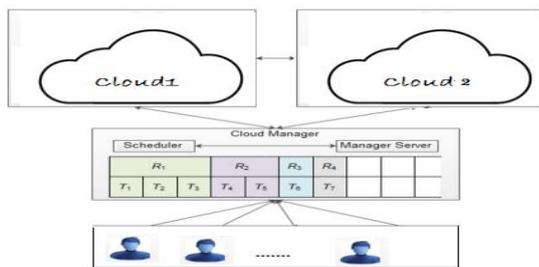
Aliva Priyadarshini, Lecturer, (Integrated MCA , UTAL UNIVARSITY)

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Task scheduling Algorithms in Multi cloud Environment



In [3] the cloud model almost remains the same as above except there will be centralized manager server known as the cloud manager that have the status information of all VMs of respective clouds. A user request for executing a application is received by the central cloud manager, it divides the application into several tasks, and puts them in a queue. Each task is then allocated to an available VM of a cloud. This centralized model shown in Fig 2 is somehow better then the above model, because it avoids the frequent communication between the cloud managers of different clouds.



IV. BENEFITS OF MULTI-CLOUD

This type of cloud implementation is very much handy to many businesses who want to benefit from cloud computing without being burdened by the risks associated with it. It can be a best option for large companies that constitutes several divisions/departments that perform different functions and hence require different cloud services [20]. The following are some benefits of multi-cloud

1- Lowering Vendor Lock-in :The vendor lock –in [18] problem refers to a situation during which customers becomes dependant on a single cloud provider , and is not able to transits to other service providers which may be due to incompatible technology or legal constraints. This situation is fully avoidable in heterogeneous environment like cloud

2- Flexibility: By deploying more than one cloud services an organization is not constrained to a single service provider for delivering user services[18]. Rather by choosing multiple cloud providers,a organization can smoothly distribute workloads, and application with different technical needs to multiple cloud providers.

3- Lower risk of DDOS attacks : [19] A higher degree of protection can be provided from the DDOS attacks. When multiple data centers are used ,then a possible DDOS attack will not necessarily lead to failure of the entire system , as because ,

4- As resources reside on multi cloud, it therefore creates a resilient architecture.

5 Limitations of Multi-cloud

1- As a cloud system grows wider and wider, more complex it becomes to manage such a large system.

2- Another issue to focus is security. As a no cloud services are used, so strong security algorithms needs to be implemented across multiple cloud platforms.

6 Application model and Problem Statement

In a scenario , where multiple number of clouds are collaborating , the following parameters are initially figured out:

- Variable C, is used to represent the set of clouds participating in the entire multi-cloud architecture, where $C = \{ C_1, C_2, C_3, \dots, C_m \}$
- Variable A, is used to represent the set of applications that are submitted to the multi-cloud, where $A = \{ A_1, A_2, A_3, \dots, A_n \}$
- Variable T_i is used to represent a set of tasks belonging to application A_i , where $T_i = \{ T_{i1}, T_{i2}, T_{i3}, \dots, T_{ik} \}$

Other then the above variables 2 data structures are used namely DAG(Directed acyclic Graph) and ETC(Expected time to compute) matrix.

- Every application has its corresponding DAG ,ie $G_i = (T_i, E_i)$

(T_i is declared above, and E_i represents a set of edges in G_i . An edge $E_{ijk} \in E_i$, where $E_{ijk} = (T_{ij} \rightarrow T_{ik})$)

For example let T_{13} and T_{12} are tasks belonging to application A_1 , and let task T_{13} is dependent on T_{12} . Then there will be an edge E_{123} in G_1 ,

where $E_{123} = (T_{12} \rightarrow T_{13})$, $i=1, j=2, k=3$.

- For every task belonging to the different applications submitted to the multi-cloud, their corresponding execution time on each cloud is represented using a 2×2 matrix known as ETC. Where each element in the matrix $ETC_{ij,c}$ refers to the execution time of j th task i th of application i on cloud c .

The problem statement, is to select a cloud from a given set of clouds to which the task of an user application is to be forwarded.

7 About The task scheduling algorithms in Multi cloud

A single cloud provider with the help of internet and a remote server stores data, applications. It handle user request for various services like software, storage space, network bandwidth etc. The application which is received at a data center is a collection of tasks, which may be dependent or independent on each other. These tasks are further assigned to different virtual machines in a scheduled manner. For this a scheduling policy is used and an optimal scheduling policy provides reduced makespan, increased cloud resource utilization, and reduced energy consumption. Therefore scheduling has been the most prominent topic in the field of distributed computing. Various studies and research have been conducted to obtain a optimal or near to optimal solution in a single unified cloud environment, but the same algorithms may not work for a multi cloud architecture. As workloads are distributed across multiple data centers, in order to fulfill the benefits achieved in a multi cloud environment, some efficient scheduling procedures needs to be put forward. As an application for a cloud environment is divided into a number of tasks, in cloud there are two different modes which are allocated to

tasks based on the Service level agreement between the customer and the cloud provider. The two modes are AR (advanced reservation mode), BE (best effort mode). The AR mode allows resource reservation in advance, where as in BE mode resources are allocated to tasks only when they are available. For many scheduling algorithms a ETC (Expected time to compute) matrix is constructed, and the DAG is drawn to determine the task dependency and execution time of tasks on all the clouds.

The first 2 algorithms to be implemented in a multi-cloud environment are Cloud List Scheduling (CLS), and Cloud Min Min Scheduling (CMMS), followed by Minimum completion time (MCC), Median Max (MEMAX), Cloud Max Min Normalization (CMMN), Multi Objective Scheduling, Allocation aware scheduling .

V. METHODOLOGIES

MCC (Minimum completion cloud Scheduling)

It is a preemptive task scheduling [12] that finds the minimum completion of the tasks which are ready. The completion time of a task is calculated which the sum of the expected computation time plus the shortest is waiting time, which a process waits for a cloud to be ready before execution. As multiple clouds exist, the cloud that gives the minimum completion time is selected for a task. When 2 task have the same completion time, then they follow the alphabetic order among the task. Another feature which the algorithm implements are the preemptive nature. When from 2 tasks i and j, if task i is executing on cloud 2 and task j has arrived and is a AR task. If task j is ready for the same cloud 2 then i is preempted from cloud 2 and j is assigned to cloud 2. MCC outperforms the traditional MCT algorithm used for unified cloud environment and grid computing [13-15] and CLS algorithm [2].MCC provides better make span then CLS but the average cloud utilization of MCC is closed to CLS.

Median MAX Scheduling (MEMAX)

This is a preemptive and 2 phase scheduling algorithm [12]. Two phase because the calculations are done in 2 steps. First the median of all ready tasks are calculated, and in the second step, the task with the maximum median value is chosen and is assigned to the cloud which takes minimum execution time for that task. Like MCC, MEMAX also preempts tasks for task arriving in AR mode. As per the authors MEMAX performs better than the traditional Max-Min algorithm in terms of make span but the average cloud utilization of MEMAX doesn't exceeds Max-Min.

CMMN(Cloud Min Min Normalization)

The main focus of this algorithm [12] is to improve make span. The algorithm has 2 phases. In the 1st phase the ETC matrix is normalized by taking the ratio between 2 elements. 1st is the difference between ETC_{ij,k} and minimum execution time of task T_{ij}. The 2nd element is the difference between minimum and maximum time of T_{ij}. Here T_{ij} is the jth task of ith application and ETC_{ij,c} is the execution time of T_{ij} on cloud c. The normalized ETC data is grouped into 2 groups/batches, a large and a small batch. In the 2nd phase Min-min algorithm is applied to the large batch followed by

the small batch. Tasks preemption is supported in CMMN, for AR tasks. The authors tested CMMN against CMMS[2] using bench mark data sets. CMMN provides better cloud utilization and reduced makespan then CMMS.

MOTS (Multi-Objective Task Scheduling)

It is Multi-objective [3] because it reduces both makespan and total cost combiningly .The algorithm is also a 2-phase scheduling that implements the ETC matrix along with a CO matrix. Each element of the CO matrix represents the execution time of a particular task on a particular cloud. In 1st phase both ETC and CO matrix are normalized by dividing the ETC matrix and CO matrix element with their corresponding maximum value. A threshold value is applied the 2 resultant matrices to get a new resultant matrix . In the 2nd phase the Min-Min algorithm [5-7] is applied to the new matrix.

The algorithm provided decreased make span and reduced total cost then PBTS(Profit based task scheduling) and CMMS[2] algorithms , but on the other hand MOTS produced reduced average cloud utilization then CMMS.

CLS(Cloud list algorithm)

One of the earliest algorithms [2] for task scheduling in multi-cloud. The algorithm produces a list based on the earliest start time (EST) and the latest start time (LST) of a task.

$$EST(T_i) = \max \{EST(T_m) + AT(T_m)\}$$

$$T_m \in \text{pred}(T_i)$$

$$LST(T_i) = \min \{LST(T_m) - AT(T_i)\}$$

$$T_m \in \text{succ}(T_i)$$

AT(T_i) is the average execution time of task T_i. The task on the top of the list will be assigned to the cloud that can finish it in earliest time.

(CMMS)Cloud min min Scheduling.

The CMMS scheduling [2] is based on the popular Min-Min strategy [5], which doesn't include task dependencies. CMMS scheduling find the cloud that gives the earliest finish time of a task T_i, such that there will be no dependent task on T_i. Then the task-cloud pair with the earliest finish time is selected and that task is executed.

VI. RESULT ANALYSIS

There are 3 proposed algorithms [8] under one category , allocation-aware Min-Min batch (AminB), allocation-aware Max-Min batch (AMaxB) and allocation aware Min-Min Max-Min batch (AminMaxB). All these algorithm commonly map tasks to the clouds and then maps tasks to VMs in the selected cloud. There are 3 phases in the algorithms which are matching, allocation and scheduling. The authors implemented a multi cloud model (Fig-2), with a centralized cloud manager that communicates with the cloud manager of all cloud. The matching phase is the 1st phase and is common for all the 3 algorithms. During matching the centralized cloud manager finds the completion time of a task on all VMs of all the clouds and then it allocates the task to that VM which gives the earliest completion time.

In allocation phase, a particular scheduling policy is used by the cloud manager of the selected cloud by collaborating with the other cloud managers, to find the VM on some cloud that might gives the earliest completion time for that task. This is done because, the scheduling policy used by the centralized cloud manager is different from the scheduling policy of the individual cloud manager. Like in AMinB, Min-Min algorithm is used, in AMaxB, Max-Min policy is used and in AMinMaxB either Min-Min or Max-Min policy is used. In the last scheduling phase all the task are executed in the respective virtual machines concurrently. All the 3 allocation aware algorithms were tested against the existing Min-Min and Max-Min algorithm, and were found to produce better make span and cloud utilization then the existing algorithms.

VII. CONCLUSIONS

Most. Algorithms like MOTS, MEMAX, CMMN were not tested against popular algorithms like CLS and Round Robin. There are several other parameters like the overall throughput, energy consumption, reliability etc, which were not focused on these algorithms. of the algorithms proposed for multi-cloud environment don't include the deadline criteria of a task.

REFERENCE

1. Duraio F, Carvalho JFS, Fonseca A, Garcia VC (2014) A systematic review on cloud computing. J Supercomput 68(3):1321–1346.
2. J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin and Z. Gu, "Online Optimization for Scheduling Preemptable Tasks on IaaS Cloud System", Journal of Parallel Distributed Computing, Elsevier, Vol. 72, pp. 666-677.
3. Sanjaya K. Panda, Prasanta K. Jana "A Multi-Objective Task Scheduling Algorithm for Heterogeneous Multi-cloud Environment", International Conference on Electronic Design, Computer Networks and Automated Verification (EDCAV), IEEE, At Meghalaya., pp. 82-87.
4. M Muhil, UH Krishna, RK Kumar, EAM Anita, "Securing Multi-Cloud using Secret Sharing Algorithm", 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
5. O. H. Ibarra and C. E. Kim, "Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors", Journal of the Association for Computing Machinery, Vol. 24, No. 2, pp. 280-289, 1977.
6. R. F. Freund, M. Gherrity, S. Ambrosius, M. Campbell, M. Halderman, D. Hensgen, E. Keith, T. Kidd, M. Kussow, J. D. Lima, F. Mirabile, L. Moore, B. Rust, H. J. Siegel, "Scheduling Resources in Multi-User, Heterogeneous, Computing Environments with SmartNet", 7th Heterogeneous Computing Workshop, pp. 184-199, 1998.
7. R. Armstrong, D. Hensgen and T. Kidd, "The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-time Predictions", 7th IEEE Heterogeneous Computing Workshop, pp. 79-87, 1998.
8. Sanjaya K. Panda, Indrajeet Gupta, Prasanta K. Jana, "Task scheduling algorithms for multi-cloud systems: allocation-aware approach", Springer Science+Business Media New York 2017, Volume 21, Issue 2, pp 241
9. Lacheheb, M.N., & Maamri, R. (2016). Towards a Construction of an Intelligent Business Process based on Cloud Services and Driven by Degree of Similarity and QoS. Information Systems Frontiers, Springer, 18(6), 1085–1102
10. Li, G., & Wei, M. (2014). Everything-as-a-Service Platform for On-Demand Virtual Enterprises. Information Systems Frontiers, Springer, 16(3), 435–452
11. Chunlin, L., & LaYuan, L. (2015). Optimal Scheduling Across Public and Private Clouds in Complex Hybrid Cloud Environment, Information Systems Frontiers, Springer, pp. 1–12.
12. Panda, S.K., & Jana, P.K. (2015). Efficient Task Scheduling Algorithms for Heterogeneous Multi-cloud Environment. The Journal of Supercomputing, Springer, 71(4), 1505–1533

13. Freund RF, Gherrity M, Ambrosius S, Campbell M, Halderman M, Hensgen D, Keith E, Kidd T, Kussow M, Lima JD, Mirabile F, Moore L, Rust B, Siegel HJ (1998) Scheduling resources in multiuser, heterogeneous, computing environments with smartNet. 7th IEEE heterogeneous computing workshop. pp 184–199
14. Kwok Y, Ahmad I (1996) Dynamic critical-path scheduling: an effective technique for allocating task graphs to multiprocessors. IEEE Trans Parallel Distrib Syst 7(5):506–521
15. Topcuoglu H, Hariri S, Wu M (2002) Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans Parallel Distrib Syst 13(3):260–274
16. <https://cloudacademy.com/blog/when-to-use-a-multicloud-strategy/>
17. https://www.webopedia.com/TERM/C/cloud_services.html
18. <https://www.stratoscale.com/blog/cloud/scalability-cloud-organizations-win-cloud/>
19. <https://www.networkworld.com/article/3237184/the-benefits-of-multi-cloud-computing.html>
20. <https://www.cloudhealthtech.com/blog/multicloud-advantages-disadvantages>
21. <https://www.apriorit.com/dev-blog/552-multi-cloud-computing>

Name of the algorithm	No of phases	Time Complexity	Parameters effected	Type of Multi cloud model
MCC (Minimum completion cloud Scheduling)	one	$O(m)$	Reduced Makespan	Individual cloud manager model
Median MAX Scheduling (MEMAX)	Two	$O(kl^2m)$	Reduced Makespan	Individual cloud manager model
CMMN(Cloud Min Min Normalization)	Two	$O(kl^2m)$	Reduced Makespan and average cloud utilization	Individual cloud manager model
MOTS (Multi-Objective Task Scheduling)	Two	$O(kl^2m)$	Reduced Makespan, better cloud utilization, Reduced total cost	Individual cloud manager model
CLS(Cloud list algorithm)		NA	Reduced energy consumption, and reduced workload balance	Centralized cloud manager model
Allocation Aware approach	Three	$O(kl^2m)$	Reduced Makespan and average cloud utilization	Centralized cloud manager model
(CMMS)Cloud min min Scheduling.		NA	Reduced energy consumption, and reduced workload balance	Centralized cloud manager model

AUTHORS PROFILE



Aliva priyadarshini, qualification:; MTECH(IT), UTAL UNIVERSITY
 Profession: Lecturer (Integrated MCA, UTAL UNIVERSITY).