FAIRNESS AS JUSTICE EVALUATOR IN SCHEDULING CLOUD RESOURCES: A SURVEY

B. Anuradha[1], S. Rajasulochana[2]

Department of Computer Science and Engineering, SriGuru Institute of Technology, India

anu.cse@sriguru.ac.in [1]

rjsulochana26@gmail.com[2]

ABSTRACT

Cloud computing, an internet based computing, is a new business paradigm that provides a dynamic and internet optimized infrastructure for deploying the applications. Cloud computing itself is a grid computing with additional property of elasticity and scalability. Since cloud has a heterogeneous pool of resources, scheduling plays a vital role in cloud computing. It is all about executing the current jobs in a given constraint. Scheduling, an NP-complete problem, is still under a serious research. Fairness in scheduling is an important criterion that improves the efficiency and provides optimal resource allocation. This paper provides a landscape and discusses various scheduling techniques that evaluate fairness.

KEYWORDS: *Fairness, Scheduling, Heuristics, Resource allocation.*

1. INTRODUCTION

Cloud computing is yet another buzz in the area of computing. In other words it is a revolution in the area of computing in this era. Cloud can be defined in various ways from different perspectives. The **N**ational **I**nstitute of **S**tandards and **T**echnology (NIST) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction". It is all about storing and accessing the data and other services in a centralized server called the data centre, accessed via internet. The seven key elements of cloud computing are utility pricing, managed operations, third-party ownership, self-service provisioning, management automation, virtualized resources and elastic resource capacity. Cloud, contains a pool of resources and provides an illusion of all the resources available in the internet and thus is the difference. Cloud computing is more advantageous than other computing technologies due to its property of elasticity and scalability and has an added advantage that it is dynamic. Yet cloud meets challenges in the areas of security, ensuring cost, ensuring adequate performance, data migration, scheduling, load balancing, resource allocation and the like.

Scheduling, load balancing, resource provisioning and traffic control are some of the ways of achieving QoS in cloud environment where, scheduling and load balancing goes hand in hand. Scheduling tasks to cloud resources is one main step in cloud resource management. Thus the main aim of scheduling algorithm is to assign tasks to available resources (virtual) in a cost-effective manner, in a way it meets the QoS and efficiency and hence scheduling is performed in the virtual machine layer and application layer. Resource allocation strategy using service level agreement (SLA) mainly depends on Response time, throughput and QoS parameters. Service level agreement in general is an agreement between the service provider and the recipient of the service. Task SLA is an instance associated with every task to ensure that the given task reaches the end condition in the given period of time.

Figure 1 Service Level Agreement (SLA)

Efficient scheduling mechanism should meet the QoS parameters and should enhance resource utilization. With the cloud-scale expansion fairness in scheduling and resource allocation is still an issue. Fairness is a challenging factor when tasks waiting for resources are to be mapped. There are several ways in which fairness can be determined. Fairness constraint proposed by Berger model plays an important role in determining the fair allocation of resources by means of justice evaluation function [8]. Yet it has its disadvantage when task-resource mapping is not met. This paper presents a bird's eye view on the various ways in achieving fairness in scheduling the cloud resources.
The paper is organized as follows: Section "Scheduling landscape" gives a brief introduction to scheduling. Section "Fairness in scheduling" depicts some of the ways in which fairness can be achieved in scheduling. Section "Fairness via resource allocation" describes how resource allocation strategies help in achieving fairness in cloud scheduling. Section "Heuristic approaches" provides a list of heuristic algorithms to achieve scheduling efficiency. Finally the concluding remarks are presented in the "Conclusion and Future work" section.

2. SCHEDULING LANDSCAPE

Scheduling in cloud computing is the series of following steps:

- Resource identification – involves identifying the relevant resources
- Resource selection – choosing the best optimized resource based on Qos parameters
- Task-Resource mapping – Mapping tasks to the relevant resource

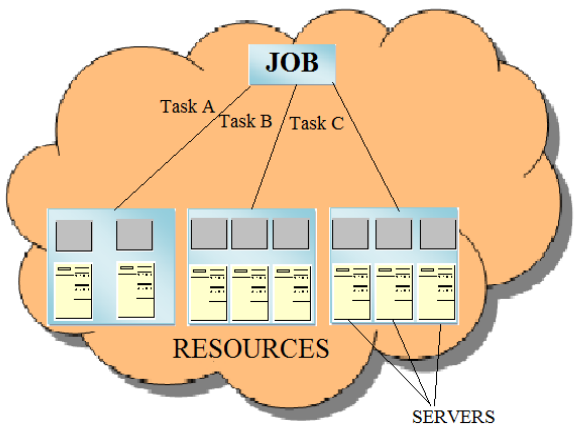

Figure 2 Scheduling tasks to available resources

Cloud being a heterogeneous environment, the resource pool is generated from various heterogeneous severs. Scheduling aims at reducing the makespan and increasing the resource utilization at an optimized cost.

2.1 Scheduling types

Scheduling can be done based on any one of the following:

- Pre-emptive and non-pre-emptive scheduling
- Gang scheduling
- Fair share scheduling
- Capacity scheduling
- Delay scheduling
- Appointment scheduling
- Priority scheduling

2.1.1 Pre-emptive and non pre-emptive scheduling:

Pre-emptive scheduling schedules the task based on priority. When a highest priority task is being executed and if another task with a priority higher than the current one is in the ready state, pre-emptive scheduling schedules the new task even before the current task completes its execution. Concourse, non pre-emptive scheduling doesn't execute the new task before the current task in the ready state completes its execution.
Pre-emptive scheduling helps overcome the problem of resource-reservation and thus achieves high throughput overcoming the problems of low machine utilization, high job waiting times.

2.1.2 Priority scheduling

In case of priority scheduling, priority is assigned with each task and tasks are executed based on the priority. Priority scheduling can be either pre-emptive or non pre-emptive. In case of pre-emptive priority scheduling when a task arrives with the highest priority than the currently executing job and if it is ready for execution, pre-emptive priority scheduler will pre-empt the currently running task with the newly arrived task. Concourse the non pre-emptive scheduler will simply put the newly arrived highest priority task to the top of the ready queue.
The major problem associated with priority scheduling is the problem of indefinite blocking and starvation. One feasible solution to the problem of indefinite blocking is a technique called *Aging*. Aging is the process of increasing the priority of a task that is waiting for a long time.

2.1.3 Delay scheduling

Delay scheduling overcomes the deficiency of FIFO scheduler that don't provide fairness on the job side and reduces data locality in the on the node side. Delay algorithm on the other side reduces the strict job order that is the case of FIFO scheduler. It provides maximum delay time for jobs to avoid starvation. Delay scheduling policy is also adopted in fair – share scheduler to achieve maximum fairness.

2.1.4 Gang scheduling

Gang scheduler schedules the tasks together in a synchronized fashion. It is the scheduling process in which two are more tasks are assigned to the same resource at the same time. The scheduler provides a dedicated access to a particular task for a pre-defined period of time after which it suspends the current task and executes the other. It makes use of priority in case of over subscription. It provides an illusion that a task is dedicated to a single resource at any time ths reducing the overhead of context switching.

2.1.5 Fair share scheduling

Fair share scheduling is a technique that ensures fairness, faster response time and minimum delay. It also ensures resource allocation in a fair manner and is an added advantage. Fair share scheduler is essentially used where effective resource management is an end criterion.
One of the major issues concerned with scheduling is that it possesses higher overhead while scheduling more number of tasks. Request partitioning/ request scheduling is one way of reducing the overhead while scheduling. It is quite difficult for a single algorithm to provide efficient scheduling, meeting all the user QoS parameters.

There are several scheduling algorithms used in cloud environment. Hadoop by default uses FIFO scheduling. Even though FIFO is the simplest form of scheduling, it has poor fairness. It is the reason that most of the organizations have moved on to Fair scheduling that it attempts to avoid starvation of resources for longer tasks. For example Linux kernel is found to use a Completely Fair Scheduling algorithm.

2.2 Scheduling in Hadoop

Hadoop, a multi-tasking file system has the capability to process multiple tasks for multiple users at the same time. Hadoop works in batch mode scheduling where the incoming tasks are processed without being stored. Hadoop cluster has been now designed to support varying workload and priority and thus has been moved on from FIFO scheduler to fair scheduler. Facebook, a Hadoop cluster uses fair scheduler whereas yahoo uses capacity scheduler for scheduling its tasks [14].

2.2.1 Fair scheduler

The fair scheduler upon receiving jobs organizes them into a pool of equal shares. When a new task slot is received it is assigned to the pool with minimum share, thus ensuring fairness [14] [15].

2.2.2 Capacity scheduler

Capacity scheduler differs from fair scheduler in a way that it is defined for larger clusters [14]. It can further be classified into finite capacity scheduling and infinite capacity scheduling.

3. FAIRNESS IN SCHEDULING

The task resource mapping can be compared to the to the commodity economy model similar to pay-per-use basis. This economy model can be implemented using a market-oriented strategy, auction mechanism and several other approaches in cloud environment. Cloud computing is a commercial environment that has its foundation in high Quality of Service (QoS) and fairness, where user satisfaction is the end goal.

Fairness can be dealt with user as well as resources. From the user perspective, fairness is to provide user satisfaction through minimal completion time etc and for the resources it is effective resource utilization and the like. An efficient scheduler should provide fair allocation of resources in a way it ensures no task is starving for resources. In other words, it maintains fairness by fairly distributing the available resources to the tasks at the same time satisfying user constraints.

Priority plays a major role in determining fairness. Scheduling based on priority maximizes the profit. A threshold limit can be assigned with each job and if the jobs actual start time doesn't exceed the threshold value it is considered that the job is fairly treated else the job is considered to have been treated unfairly. Fairness in turn should result in higher utilization. Fairness can be estimated using a special metric called as QoS parameters like makespan, latency, bandwidth etc.

3.1 The Backfilling Strategy

Backfilling is one such strategy that is an extension of FCFS, does provide fair allocation of resources. It achieves scheduling optimization through better resource utilization by scheduling the jobs out of order. Backfilling strategy may/may not schedule the jobs based on priority which is both its advantage as well as disadvantage. There are two commonly used backfilling approaches: conservative backfilling and EASY (Extensible Argonne Scheduling System) backfilling. These backfilling algorithms are used to overcome the problem of starvation and waiting time.

3.1.1 Conservative Backfilling

The conservative backfilling approach allows each job to reserve the resources it needs, when it is inserted into the job queue [10].

3.1.2 Easy Backfilling

Easy backfilling strategy also called as aggressive backfilling provides better utilization in higher load. It provides less fairness when compared to conservative backfilling technique. Both these

backfilling algorithms don't consider priority of jobs. Backfilling is similar to cutting in line termed as "queue jumping", considered as violation of social justice [13].
[3] proposes a Flexible Backfilling strategy that uses priority to schedule the jobs. Priority is calculated both at the beginning and ending of scheduling based on some heuristics.
Fairness can be defined based on two metrics [13]: fair start time and fair share of resources. By fair start time we mean the early start time and resource equality is what fair share is all about according to Raz et al [7]. Fairness can also be estimated using justice evaluation function. Several people contributed towards framing a justice evaluation function.

3.2 Previous works

3.2.1 The work of John Rawls

John Rawls in 1985 presented "A Theory of Justice", where fairness is stated with the help of two principles namely The Liberty Principle and The Equality Principle. According to the liberty principle "certain rights and freedoms are more important or 'basic' than others" and the equality principle states that "effective equal chance should be given to all the individuals".

3.2.2 The work of Homans

Homans assumed justice as the result of comparison between reward and investment. He formulated justice as

$$\text{Justice} = \frac{\text{A's reward}}{\text{B's reward}} - \frac{\text{A's investment}}{\text{B's investment}} = 0$$

If this ratio equates to zero it indicates perfect justice. Else if the ratio doesn't equate to zero it indicates injustice.
The definition of justice made by Homans is not clear. In case of injustice the model of Homans failed to specify whether the individual is under-rewarded or over-rewarded. Thus Homans specification of justice is not clear.

3.2.3 The work of Berger, Zelditch, Anderson and Cohen

The model proposed by Berger et al defines justice with respect to a "referential structure" that addresses the issues of Homans model. Thus this model uses the referential structure to distinguish between under-reward and over-reward. Berger model maps the user needs to social wealth.

Justice evaluation function (JEF) = Actual reward – Just reward

Berger model fails when user need to social wealth match is not achieved.

3.2.4 The work of Jasso and Ross

Evaluating justice involves the following steps:

The difference between the actual reward and the just reward (JEF) gives the justice value.

If JEF = 0

Perfect justice

Else if JEF > 0

Over-reward

Else if JEF < 0

Under-reward

Jasso's law of justice evaluation says that the relationship between justice evaluation and the ratio between actual and just reward is logarithmic

$$J = \theta \ln(A/C)$$

Where,

J = justice evaluation function

A = actual reward

C = just reward

Jasso expressed just reward as follows

$$C = A \exp(-J/\theta)$$

Where,

Θ = proportionality constant

Baomin Xu proposed an algorithm based on Berger model for job scheduling to achieve fairness [4]. The scheduling algorithm makes use of dual fairness constraint.

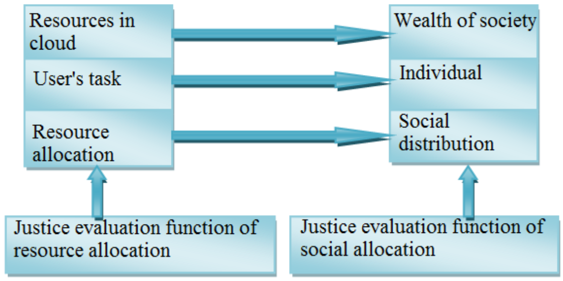

Figure 3 mapping between cloud and Berger model

Berger model is based on the expectations state theory that deals with the nature or behaviour of actor and others. The model establishes dual fairness constraints and two justice evaluation functions namely task justice and system justice respectively. In the dual fairness constraint, the first constraint restrains fairness in the resource selection whereas the second constraint judges the fairness in resource allocation. The two justice evaluation functions, task justice is about fair execution of each task and system justice deals with the fairness of the system as a whole.

3.3 Fairness in Resource Allocation

Resources can be allocated to both homogeneous as well as heterogeneous resources in the cloud.

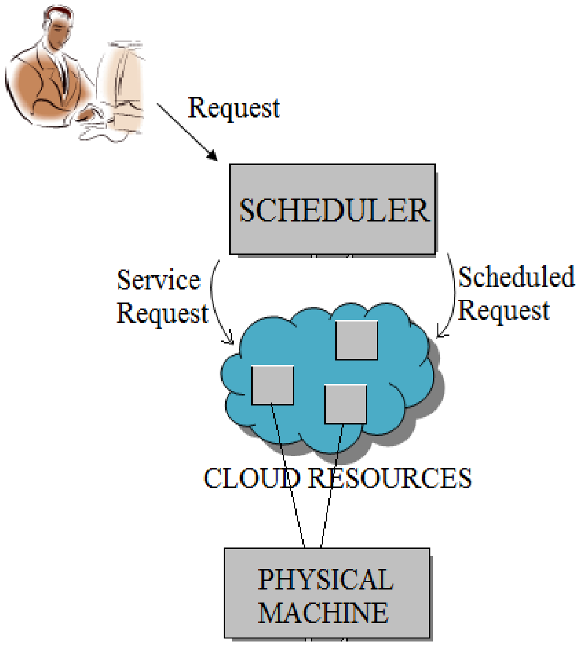

Figure 4 Resource allocation

Fairness in resource allocation is another criterion that deals with justice. In cloud environment resource allocation is mainly based on heuristics for better optimization. Thus based on QoS heuristics, task can be assigned to an optimal resource. Leivadeas et al proposed a hierarchical framework for resource mapping where resource mapping is done in two phases: Request partitioning phase followed by embedding phase. In the request partitioning phase requests are grouped based on functional attributes and cost-effective request is been selected for further processing. Embedding phase, upon receiving the cost-effective request to the virtual resource through an appropriate algorithm [5].

Abirami et al [9] proposed a "Linear scheduling strategy for scheduling in cloud computing environment" where best fit strategy is used to satisfy the requests based on available VM.

Wang et al [4] designed simple heuristics based "Dominant resource fairness in cloud computing system", where a user can demand his allocation by reducing the allocation of other task thereby maintaining a demand in the resource demand substantially providing higher resource utilization and reduced completion time.

Lawrence et al [6] proposed a resource scheduling methodology using potentially all pair-wise rankings of all possible alternatives (PAPRIKA). PAPRIKA method evaluates fairness based on user satisfaction. This method makes use of both task matrix and resource matrix. Priority of the resource is calculated with respect to a threshold value. Tasks are then mapped to the resource that gives higher user satisfaction, thus improving resource utility and minimizing the allocation time.

Wu et al [2] proposed a task scheduling algorithm based on QoS-driven in Cloud Computing. The algorithm schedules the task by computing the priority of tasks based on the special attributes of the task. The model provides lesser Makespan and average latency when compared with berger model and Min-Min algorithm.

4. HEURISTIC APPROACHES

Scheduling tasks on heterogeneous environment is considered to be a NP-Complete problem and does scheduling is based on previous history, heuristics / meta-heuristics algorithm are used for better optimization. Several heuristic algorithms are available for both static mapping as well as dynamic mapping. Some of the QoS based heuristic algorithms for static mapping are min-min algorithm, max-min algorithm, opportunistic load balancing, and suffrage heuristics etc whereas algorithms for dynamic scheduling includes immediate mode heuristic algorithms and batch mode heuristic algorithms.

4.1 Static mapping algorithms

Static mapping is employed, when the situation under which scheduling is performed remains unchanged. The following are some of the static mapping techniques:

4.1.1 Opportunistic Load Balancing Algorithm

Opportunistic load balancing algorithm assigns task in any order to the resource with minimal load. It keeps the machine busy always. The algorithm doesn't provide better optimization always and may result in poor optimization.

4.1.2 Task Scheduling Algorithm based in Load Balancing

Y.Fang et al [1] proposed a two-level task scheduling algorithm based on load balancing. Apart from user satisfaction it also provides high resource utilization. In the first level it maps user task to the virtual machine and in the second level the algorithm maps virtual resources onto the resources. Thus the response time, resource utilization and performance have been improved.

4.1.3 Min-Min algorithm

The task with overall minimum expected completion time is assigned to the corresponding resource. The disadvantage of min-min algorithm is longer tasks have to wait for a long to get processed and the algorithm fails to balance the load as it schedules the smaller task always.

4.1.4 Max-Min algorithm

The task with overall maximum expected completion time is assigned to the corresponding resource.

4.2 Dynamic mapping algorithms

Dynamic mapping is employed when the situation under which scheduling is performed is changing. Dynamic mapping techniques can be further classified as Dynamic Immediate mode mapping and Dynamic batch mode mapping. Following are some of the dynamic immediate mode mapping techniques:

4.2.1 Switching algorithm

As is the name the algorithm toggles between minimum execution time (MET) and minimum completion time (MCT) to filter the tasks and to balance the load accordingly.

4.2.2 K-Percent best algorithm

The algorithm assigns the task to the machine with a minimum completion time in the chosen subset of size k.
Following is one of the dynamic batch mode mapping technique:

Suffrage based heuristics

In case of suffrage heuristics, task with minimum suffrage value is mapped to the machine with minimum completion time. The suffrage value is the result of difference between two minimum completion times.

5. CONCLUSION

Cloud computing is an emerging topic in distributed computing and is a door to the future of information technology. Scheduling is among the several issues that cloud computing is facing today. In future, resources can be effectively and efficiently scheduled by leveraging techniques like, scheduling and resource provisioning, request scheduling and task scheduling, scheduling and load balancing etc where efficiency and fairness will be the end goal.

ACKNOWLEDGMENT

We would like to thank the principal and staff members of SriGuru Institute of technology, friends and family members for their support and guidance in bringing this article. We also like to thank them for their valuable reviews.

REFERENCES

[1] Y. Fang, F. Wang, and J. Ge*, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing"*, Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[2] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou, "*A task scheduling algorithm based on QoS-driven in Cloud Computing*", Science Direct, Elsevier Publications, Procedia Computer Science 17 (2013) 1162 – 1169.

[3] A.D.Techiouba, G.Capannini, Ranieri Baraglia, D.Puppin, M.Pasquali, Laura Ricci, *"Backfilling strategies for scheduling streams of jobs on computational farms",* European CoreGRID NoE, IST-2002-004265.

[4] Wei Wang, Baochun Li, Ben Liang, *"Dominant Resource Fairness in Cloud Computing Systems with Heterogeneous Servers".*

[5] Aris Leivadeas, Chrysa Papagianni, *"Efficient Resource Mapping Framework over Networked Clouds via Iterated Local Search-Based Request Partitioning",* IEEE Transactions on parallel and distributed systems, Vol. 24, No. 6, June 2013.

[6] Hilda Lawrance, Dr. Salaja Silas, *"Efficient Qos Based Resource Scheduling Using PAPRIKA Method for Cloud Computing"*, International Journal of Engineering Science and Technology (IJEST), Vol. 5 No.03 March 2013.

[7] D. Raz, B. Avi-Itzhak, H. Levy, "Fairness considerations in multi-server and multi-queue systems", In Proc. 1st Intern. Conf. Perf. Eval. Methodolgies and Tools, 2006.

[8] Baomin Xu, Chunyan Zhao, Enzhao Hu, Bin Hu, *"Job scheduling algorithm based on Berger model in cloud environment"* in *"Advances in Engineering Software"*, Elsevier Ltd, 2011.

[9] Abirami S.P, Shalini Ramanathan, *"Linear Scheduling Strategy for Resource Allocation in Cloud Environment",* International Journal on Cloud Computing: Services and rchitecture(IJCCSA),Vol.2, No.1,February 2012.

[10] D. Feitelson, L. Rudolph, and U. Schwiegelshohn, *"Parallel job scheduling - a status report",* June 2004.

[11] Jayadivya S K, S. Mary Saira Bhanu, "*QoS Based Scheduling of Workflows in Cloud Computing*", International Journal of Computer Science and Electrical Engineering (IJCSEE) ISSN No. 2315-4209, Vol-1 Iss-1, 2012.

[12] Linlin Wu, Saurabh Kumar Garg and Raj kumarBuyya, "*SLA –based Resource Allocation for SaaS Provides in Cloud Computing Environments"* (IEEE, 2011), pp.195-204.

[13] Jayadivya S K, S. Mary Saira Bhanu, "*QoS Based Scheduling of Workflows in Cloud Computing*", International Journal of Computer Science and Electrical Engineering (IJCSEE) ISSN No. 2315-4209, Vol-1 Iss-1, 2012.

[14] Cited http://www.ibm.com/developerworks/library/os-hadoop-scheduling/

[15]Cited //hadoop.apache.org/docs/r1.2.1/fair_scheduler.html

BIOGRAPHY

B. Anuradha completed her PG in Computer Science and Engineering under Anna University and currently pursuing Ph.D in Information and Communication Engineering under Anna University. She is having 13 years of Academic experience in Engineering stream and currently she is with SriGuru Institute of Technology, Tamil Nadu, India. She is a professional member of ACM, CSI and ISTE. Her research interest includes cloud environment, green computing and compilers.

S. Rajasulochana was born in Coimbatore on 26th June 1990. She received her B.E. (CSE) from Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu in 2012. She is currently pursuing her M.E. (CSE) in SriGuru Institute of Technology, Tamil Nadu, India. She has published a paper in international journal and has attended two national level conferences. She also has published an application for Windows Phone 7 and got certified. Her areas of interest include Cloud Computing and Machine Learning.