American Journal of Applied Mathematics
2015; 3(1-2): 14-17
Published online December 30, 2014 (http://www.sciencepublishinggroup.com/j/ajam)
doi: 10.11648/j.ajam.s.2015030102.13
ISSN: 2330-0043 (Print); ISSN: 2330-006X (Online)

A survey on various task scheduling algorithms toward load balancing in public cloud

Pradeep Naik, Surbhi Agrawal, Srikanta Murthy

CSE Department, PESIT-BSC, Bangalore, India

Email address:
pradeep.apr20@gmail.com (P. Naik), surbhiagrawal@pes.edu (S. Agrawal), srikantamurthy@pes.edu (S. Murthy)

To cite this article:
Pradeep Naik, Surbhi Agrawal, Srikanta Murthy. A Survey on Various Task Scheduling Algorithms Toward Load Balancing in Public Cloud. *American Journal of Applied Mathematics.* Special Issue: Frontiers in Mathematics and Computing. Vol. 3, No. 1-2, 2015, pp. 14-17.
doi: 10.11648/j.ajam.s.2015030102.13

Abstract: Cloud computing is one of the real innovation in giving a shared pool of resources globally. In cloud computing there exist diverse sorts of architectures in particular private, public, hybrid and so on. The cloud computing is also meant for providing service via platform (PaaS), Infrastructure (IaaS), Software (SaaS). The term capacity planning is used to dynamically upscale or downscale the computing resources accordingly. There will be many job requests which require to get executed among the available resources in a particular cloud in order to achieve the maximum throughput, minimum waiting time and best overall performance. In this paper a systematic review is made on various job scheduling algorithms along with their working procedure and a comparison among these algorithms is presented. The paper concludes with a proposed system with improvised QoS parameters to provide a load balanced task scheduling algorithm for public clouds.

Keywords: Cloud Computing, Bipartite Graph, Augmenting Path, Job Scheduling, Scheduling, Load Balancing

1. Introduction

Cloud computing is one of the major and efficient technique, which is used as an operating system called cloud OS. This cloud OS is supposed to manage the execution of tasks, operations, processes of virtual machines, virtual servers, virtual infrastructure as well as the back end hardware & software resources of cloud environment.

Virtualization plays an important role in any of the cloud computing application; mainly server virtualization which is a technique to divide one physical server into multiple virtual servers, each of these servers will act as an individual server which is capable of maintaining its own operating system & applications.

Basically there are three types of cloud namelyprivate, public & hybrid clouds. The public cloud is the one in which there will be maximum number of clients available, which intern create a flood of requests arriving to the cloud repositories in such a scenario a load balancing has to be maintained. Load balancing is a technique to distribute workload among the existing servers, network interfaces, hard drives and other computing resources. Today load balancing has become a major issue in cloud because of the vast ocean of requests and limited amount of resources. The problem can be actually stated as-

Consider we have m resources & n number of requests coming into cloud such that n>>m. Then how to satisfy the requests? Also, some of the resources may be overloaded in cloud environment, whereas others may be ideal. Hence a proper load balancing technique is must to be exploited. In this paper five existing task scheduling algorithms have been surveyed, in the next section. Then, a comparative study is done and at last we have proposed one more system which can be considered for achieving task scheduling along with balanced load in a public cloud.

2. Related Work

There have been different types of job scheduling algorithm applied in the cloud environment with suitable modification. The main aim of any job scheduling algorithm is to maintain fairness among the jobs for their execution and reduce waiting time and to improve performance, quality of service like throughput end to end delay. Following are the various works which are surveyed-

2.1. Priority Based Job Scheduling Algorithm

In,[3] proposed a priority based job scheduling algorithm

and named it as "PJSC" which can applied in cloud environment. This algorithm is consisted of three levels of priorities including scheduling level(objective level), resource level(attribute level), and job level(alternative level).

This algorithm considers two data sets mainly request and resource. Set of jobs(requests) $J=\{j_1,j_2,j_3,\ldots j_m\}$ and set of resources $C=\{c_1,c_2,c_3,\ldots,c_d\}$ which is available in cloud environment as input where (d<m). Every job with a determined priority requests a resource. The priority of each job is compared separately using

$$Pgij = \begin{cases} \left(\frac{1}{Pgij}\right) & i \neq j \\ 1 & i=j \end{cases} \quad (1)$$

In the above equation Pg denotes a matrix with m rows and m columns known as comparison matrix. Now assume M is the comparison matrix for resource level and β will be defined as priority vector of M. now PVS is calculated which is denoted as priority vector of scheduling jobs.

$$PVS = \Delta.\beta$$

Experimental results of PJSC has a reasonable complexity but its lacks in determining value of finish time(makespan) cannot be calculated, it depends on the priority of jobs and may change from worst value to the best value.

2.2. SLA-Tree

[12] proposed a novel data structure, called SLA-Tree to efficiently support profit-oriented decision making.

SLA-tree is built on two pieces of information:

- A set of buffered queries waiting to be executed, which represents the scheduled events that will happen in near future.
- A SLA for each query, which indicates the different profits for the query for varying query response time.

By constructing the SLA-Tree, certain profit oriented "what if" questions can be efficiently answered. In "SLA-tree" a metric is used to fetch the response time, which is the time between a query, is presented to the system and the time when the query execution is completed. In this algorithm two key questions are used for scheduling:

Postpone (m,n,t): how much profit will be lost if we postpone queries $q_m,q_{m+1},..q_n$ by time t?

Expedite (m,n,t): how much profit will be gained if we expedite queries $q_m,q_{m+1},..q_n$ by time t?

The algorithm reaches out its applications in different domains like scheduling, dispatching and capacity planning, but the algorithm needs an existing query execution order and the algorithm is a greedy algorithm.

2.3. Enhanced Max-Min Task Scheduling Algorithm

In[5],the authors proposed an Enhanced max-min scheduling algorithm that offers an improved task scheduling algorithm based on max-min. This algorithm selects a job with the maximum completion time and assigns it to the resource on which minimum execution time while enhanced min-max assign task with average execution time to resource produces minimum completion time.

The algorithms takes a set of input i.e. requests which need to be executed. For all the jobs submitted execution time and completion time is calculated. After that task T_k having cost average or nearest greater than average execution time is selected and assigned to resource R_j is updated.

2.4. Short Job Scheduling Algorithm

[6] Proposes an efficient job allocation algorithm SJS in multiple clouds. It can handle the over load condition and for that the load balancing scheme is presented.

The SJS algorithm focuses mainly on the following objectives-

- Create an intermediate architecture that will accept the user request and monitor the cloud servers for their capabilities.
- Scheduling of the user requests is performed to identify the order of allocation of the processes.
- Performing the effective resource allocation under defined parameters and the cloud server capabilities.
- Define a dynamic approach to perform the process migration from one cloud to another.
- Analysis of work using graph under different parameters.

The algorithm neglected a situation whenever the system reaches an under load and over load conditions the processes will not migrate from one cloud to another resulting in an inefficient load balancing.

2.5. Job Scheduling Model Based on Multi-Objective Genetic Algorithm

[7] Presents a scheduling model for cloud computing based on MO-GA algorithm to minimize energy consumption and maximize the profit of service provides under the constraints of deadlines.

Request cognition component should have information about needs for different businesses, which include the computing, storage and communication requirements for computing, security and privacy requirements, Quality of service etc. Service decomposition component decomposes the service request into different level of granularities with different processor preferences. Task manager analyze the resource requirementsof each granularity, and mapping it on to optimal processors to reach an effective solution. It is responsible for task status management (start, stop, cancel…), determining the scheduling sequence and resource assignment for the requests and allocating suitable resources to each jobunder the help of the scheduling algorithm. Resource cognition component manage the available resources, monitoring of the performances of resources, dynamic optimization of scheduling strategy and error notification.

Following operations of genetic algorithms are used:

1 Encoding Rule.
2 Population initialization
3 Individual evaluation

Following operations are performed over the chromosomes:

1 Selection operation
2 Crossover operation
3 Mutation operation

3. Comparison of Algorithms

Table 1 presents a comparison among these algorithms, which are being surveyed in this paper-

4. Proposed Solution

The bipartite matching algorithm is aimed at a public cloud which has numerous nodes with distributed computing resources in many different geographic locations. Thus this model divides the public cloud into several clusters depending upon their workload capacity.

Each cluster will have a clusterhead which monitors the entire node available in that cluster. When the environment is very large and complex, these divisions simplify the load balancing. The model has a main controller that chooses the right cluster for arriving jobs, while the balancer/scheduler for each cloud partition chooses the best matching using bipartite algorithm.

An update process which keeps on monitoring every node of every clusters continously, the details forwarded to the cluster head and is responsible for assigning task to every node by matching the request and resource information, which best matches the resource will beprovided to that job using maximum matching bipartite algorithm.

Maximum bipartite algorithm has two main partitions – reources & requests. It will select a an unsatisfied request from the set of requests and then will try to perform a match with the nearest available in the cloud. If the status of the node which has resource is not overloaded , request can be assigned to it , otherwise it will search for another resource. Algorithm will also trace augmented path in case if requests are still usatisfied & there is a resource available. Algorithm will keep iterating till there is no unsaturated or usatisfied requests.

4.1. Advantages of the Proposed System

- This model divides the cloud into several clusters, which helps in overcoming the disadvantage of CPU under utilization.
- When the environment is very large and complex, these divisions simplify the load balancing.
- Load balancing plays a major role in improving the performance and maintaining stability.

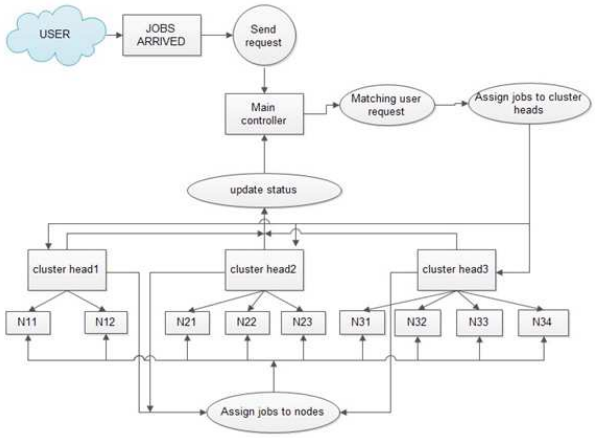

Figure 1. *Arcitecture for proposed model*

Table 1. *Comparison of different task scheduling algorithms*

Algorithm			Complexity		Scalability		Recommended in
Name	**Method**	**Factor**	**Cost**	**Time**	**Static**	**Dynamic**	
Priority based job scheduling algorithm.	Batch Mode	Meta-tasks	✓	✓	X	✓	Cloud
SLA Tree	Batch mode	Meta tasks	X	✓	✓	x	Grid
Short job scheduling algorithm	Batch mode	Meta tasks	X	✓	✓	x	Cloud
Enhanced Min-Max algorithm	Batch mode	Meta tasks	✓	x	✓	✓	Cloud
Multi- objective genetic algorithm	Batch mode	Meta tasks	✓	✓	✓	✓	Cloud

5. Future Work

Here, a method is proposed based on bipartite matching. Further work is to do the feasibility study of the proposed system and to implement it. After that we need to do a **compa**rison with other work. Efficiency of algorithm has to be checked against other existing algorithms.

6. Conclusion

A resource allocation scheme on different clouds or on different clusters of a single cloud in under-load and over-load conditions is being proposed. As the jobs are requested by the user to get executed, certain parameters are defined with each user request which includes the disk space required, CPU usage and memory allocation required for each job. Here each node of the cloud is defined with some virtual machines, to perform effective allocation, which is done by the maximum matching using bipartite graphs, which shows total resource available in terms of augmenting path; if the graph contains saturated vertex means the resource is available. Each nodeis fixed in terms of memory, load etc.

References

[1] Jhonclark and Derek Allan Holton, A first look at Graph Theory, Allied publications limited first edition 1995

[2] Swachil Patel, and Upendra Bhoi “A priority based job scheduling Algorithm in cloud computing: A Systematic review” International journal of scientific & Technology Research volume 2, issue 11, November 2014.

[3] Ghanbari, Shamsollah, and Mohamed Othman. "A Priority based Job Scheduling Algorithm in Cloud Computing." Procedia Engineering 50 (2012): 778-785.

[4] Rohit O Gupta and TusharChampaneria "A survey of proposed job scheduling algorithm in cloud computing environment", International journal of advanced research in computer science and software engineering, ISSN: 2277 128X.

[5] UpendraBhoi, "Enhanced Max-Min Task scheduling algorithm in cloud computing". International journal of application or innovation in engineering and Management.

[6] Poonam Devi, "Implementation of cloud computing by using short job scheduling", International journal of advanced research in computer science and software engineering, july-2013, ISSN :2277-128X

[7] Jing Liu, Xing-GuoLuo, Xing-Ming Zhang and Bai-Nan Li, "Job Scheduling model for cloud computing based on multi-objective Genetic Algorithm", IJCSI International journal of computer science issues, January 2013, ISSN (Print): 1694-0784 | ISSN(online)1694-0814.

[8] AshishPatro, MinJae Hwang, Thanumalayan S., ThawanKooburat, "Garuda a cloud-based job scheduler".

[9] P. Patel, A. Ranabahu and A. Sheth, "Service level agreement incloudcomputing",://knoesis.wright.edu/library/download/OOPSLA_cloudwsla_v3.pdf, (2009).

[10] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, vol. 25, issue 6, (2009), pp. 599-616.

[11] W. Sun, Y. Zhang and Y. Inoguchi, "Dynamic task flow scheduling for heterogeneous distributed computing: algorithm and strategy", IEICE Trans. on Inf. & Sys., vol. E90-D, no. 4, (2007), pp. 736-744.

[12] Yun Chi, Hyun Jin Moon, HakanHakigumus, Junichi Tatemura, "SLA-Tree: A framework for efficiently supporting SLA-based Decisions in cloud computing", EDBT/ICDT'2011 Proceedings of the 14th international conference on existing database technology, ACM 978-1-4503-0528-0/11/0003

[13] http://www.openstack.org/software/