IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Analysis of Two Phase Scheduling within Distributed System for Enhancement of Make span and Flow time

Jasjit Singh [1*], Anil Kumar [2]

[1*] CSE,G.N.D.U, Amritsar, Punjab, India
[2] CSE,G.N.D.U, Amritsar, Punjab, India

**Corresponmding Author:* jasjjit@gmail.com, *Tel.: +91-9803005912*

Available online at: www.ijcseonline.org

Accepted: 20/May/2018, Published: 31/May/2018

Abstract- Job scheduling is state of the art problem in advanced computing system. To tackle the issue of larger Make span and Flow time, several techniques are being researched over. This paper works toward secretion of job scheduling policy where Burst time is considered for arranging the jobs in clusters. Proposed system is categorised into two phases: first phase arranges the jobs by following shortest job first scheduling. The queue thus formed is presented to round robin scheduler with time quantum that varies depending upon the burst time of job. Jobs arranged are arranged in batches of 10% of total jobs in queue. SJF scheduler considered is non primitive where RRS scheduler is primitive. Second phase executes the jobs by looking at the resource clusters. Multi-source shortest path dynamic algorithm is used for selection of job that can be assigned to the resource cluster. Once job execution is complete credits are assigned which will be from 0-10. Higher the credit more proficient is the result. Optimal result is obtained by the application of proposed system in terms of Make span and flow time. Simulation is conducted in MATLAB showing improvement of 6% in overall result.

Keywords: Job Scheduling, SJF, RRS, PBS, Multi source shortest path.

I. INTRODUCTION

Advanced computing provides resources on pay per use to the users. As the user of cloud increases, resources availability degrades. Utilization of resources thus becomes critical within advanced computing like cloud. Resource allocation and management within distinct fields of advanced computing has been researched over. The in depth study of resource selection and allocation policies are provided in this section.

Resource allocation within cloud is critically based on cost. Cloud computing is meant to deliver the resources as a service over the internet. Dedicated connection is rare in case of cloud. In other words resources are shared among the regular users of cloud. Consumers not required to purchase expensive resources rather they can acquire cloud computing services on pay per use basis(Mirashe & Kalyankar 2010). The hardware support is provided with the help of data centers(DCs). In other words physical resources are provided through DCs. The DCs are further partitioned into virtual machines. High level scalability, agility and availability is ensured by DCs. According to (Armbrust et al. 2010) the elasticity of resource pool is unprecedented mechanism provided to help out IT industry without paying a heavy amount to service providers. This allows reduction is cost during maintenance and deployment. Demand of resources from the cloud is enhanced over the past decades. The availability associated with resources becoming an issue which is tackled using the techniques suggested by (Singh et al. 2012; Ranganathan et al. 2002). The growing demands of resources through single cloud provider are lacking scalability and hence availability decreases. In order to resolve the issue, public and private cloud is merged to form federation of cloud.

Cloud computing infrastructure weather associated with single or multiple clouds, is a complex multitude resource cluster. (Rajvir Kaur 2014)The computational resources are required to be managed to resolve the issue of starvation or performance degradation. (Kaur et al. 2014)Cloud resource management significantly affect the cost , availability and performance issues efficiently. Resource management in cloud distinct layers differ from one another in many aspects. Main differing aspects include elasticity, workload complexity and availability. In case fluctuation in workload is present, advance reservation policies comes into picture. In case unplanned spikes in workload auto scaling of workload can be done to balance the load out. Auto scaling of work load is provided by PaaS providers. Spikes in workload are unpredictable and hence centralised control of resources by single cloud provider is inefficient.

Resource allocation in grid generally depends of level of heterogeneous resources requirement along with advance reservation scheme. (Depoorter et al. 2014)Advance reservation in grid computing is critical since it provide concurrent access to resources hence increasing performance in terms of resource availability and speed. (Li et al. 2014)Advance Reservation (AR) for global grids turns into a critical research zone as it enables clients to increase simultaneous access for their applications to be executed in parallel, and ensures the accessibility of resources at indicated future circumstances. (Yousif et al. 2011; Rajvir Kaur 2014)Advance Reservation (AR) is a procedure of asking for Resources for use at a particular time later on. Regular Resources whose use can be held or asked for are CPUs, memory, circle space and system data transmission. AR for a grid Resource tackles the above issue by enabling clients to increase simultaneous access to sufficient Resources for applications to be executed. AR likewise ensures the accessibility of Resources to clients and applications at the required circumstances. Assessing different AR situations can't plausibly be completed on a genuine grid condition because of its dynamic nature. During advance reservation in Grid, there exists life cycle accompanied with states. These states are elaborated as under:

- Requested: it is an initial state when resources are first requested.
- Rejected: The advance reservation process is not successful or existing reservation is expired.
- Accepted: Advance reservation is successful. The reservation process yield resources for the process and it can be successfully completed.
- Committed: Resource reservation is committed before expiration of policy. Hence resource is assigned to the process and process can execute.
- Active: Process is currently executing since resources are successfully allotted to it.
- Cancelled: User does not require the resources anymore and request cancellation of allotted resources.
- Completed: Reservation end time is reached.
- Termination: The user ends the reservation before end time is reached.

Resource allocation in federated Cloud depends greatly on the distance of virtual machine from the jobs. Need for federated cloud originate as the resource deprecate from single cloud platform. As the distance increases so does the cost associated with the resources. In order to tackle the issue, nearest neighbour is identified and selected for allocation to virtual machines within federated cloud.

Environment considered for experiment in the proposed system is public cloud. The cloud environment is divided into data centers and virtual machines. Virtual machines are selected for allocation of resources. Rest of the paper is organised as under: section 2 gives the literature survey consisting of existing approaches used to optimise the resource allocation process, section 3 gives the flow of proposed system, section 4 gives the performance analysis and result, section 5 gives conclusion and future scope and last section gives the references

II. LITERATURE SURVEY

Optimal resource allocation to execute job within cloud environment is critical to conserve Makespan and Flowtime. (Horng & Lin 2015) proposed a collaboration of ant colony and ordinal optimization to solve the problem of job scheduling. Job schedule is located through the said mechanism and then executed. Result obtained is reduced execution time. Schedule to be followed is represented with ant in this case. Permutation representation is used to indicate operation within the schedule. Ordinal optimization mechanism is used in order to eliminate replication if any within the schedule to optimize the budget allocation process. (Barbosa & Monteiro 2008)proposed a mechanism to assign jobs initiated from multiple users to the cluster. Fixed set of resources are assigned to each user in this work. The execution of jobs depends upon the number of jobs within the cluster.

Makespan and Flowtime increases as number of jobs increases. (Xhafa et al. 2011) proposed hybridization of genetic and tabu search mechanism for job allocation and execution.(Rodger 2016; Elghirani et al. 2008) To execute the jobs genetic approach is followed and to locate the resource tabu search is used. Fitness function is defined in terms of cost. The fitness function thus has to be minimised and is achieved through said literature. (Switalski & Seredynski 2014)proposed a generalized extremal optimization (GEO) which is enhancement of genetic approach. The discussed approach consists of two phases. In the first phase, optimal virtual machine out of the available machines is selected. In the second phase, batches are scheduled to execute on selected virtual machine. (Kliazovich et al. 2013) proposed a energy aware job scheduling within the data centers. Energy efficiency and network awareness is being presented in this literature for achieving optimization in terms of Makespan and Flowtime.

Literature survey of multi heuristic approaches suggest that there is lack of basic scheduling policies along with premption. This could lead to starvation. To yield optimal result, premptive scheduling and be merged along with non premptive scheduling and credit based mechanism for selecting best possible solution for executing jobs out of available solutions. Next section presents flow of proposed system.

III. PROPOSED SYSTEM

The proposed system consists of two phases used to optimize job scheduling process. Phase 1 consists of hybridization of three algorithms: Non premptive SJF, Premptive Round robin and Process batch scheduling mechanism to form batches to be submitted to the next phase. Next phase consist of multi source shortest path to identify the jobs which are related to previous jobs for execution. The flow of proposed system is given in figure 1.

Initially job list is fetched from source. Source in proposed literature is initialised randomly. The burst time is obtained through random function in MATLAB environment. The obtained jobs are fed into shortest job first scheduler.

3.1 Task of Shortest job first scheduler

The shortest job first scheduler(Suri & Rani 2017) get the burst time of each job and arrange the jobs in ascending order to burst time. Pseudo code for the SJF is given as under

---

For i=1:N
For j=1:N-i
If($Job\_bursttime_j$>$Job\_bursttime_{j+1}$)
Temp= $Job\_bursttime_j$
$Job\_bursttime_{j=}$ $Job\_bursttime_{j+1}$
$Job\_bursttime_{j+1}$=Temp
End of for
End of for

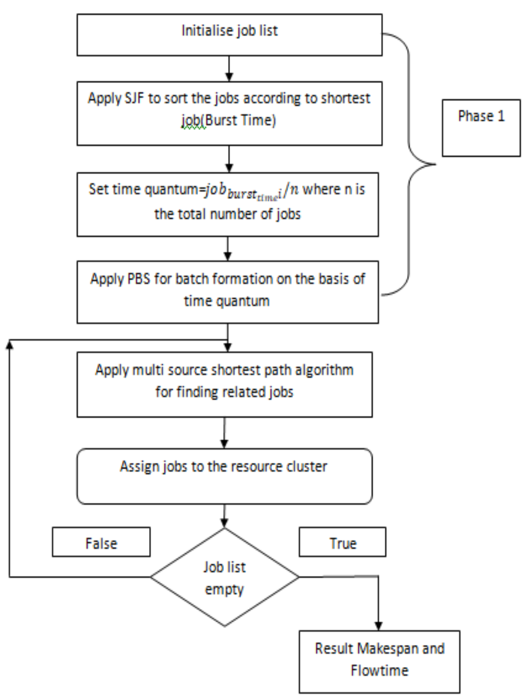

Figure 1: Flow of proposed system

3.2 Task of round robin scheduler

SJF scheduler once finished operation, round robin scheduler works to achieve premption. Time quantum is evaluated using the equation

$$Time_{Quantum} = \frac{JobBurst_i}{N}$$

Equation 1: Time quantum evaluation

3.3 Task of PBS

The main task of PBS scheduler is to form batches of jobs that can be allocated to the virtual machine clusters. Tasks that are present within PBS are analysed for relatedness. Since jobs are partitioned according to time quantum, hence tasks from same jobs may lie in distinct clusters. Relatedness thus becomes critical to join the processes for evaluation of parameters like Makespan and flow time.

3.4 Task of Multi source shortest path algorithm

This algorithm is used to identify the relatedness from PBS system. Multi source indicates that there are multiple clusters and tasks from multiple jobs may lie in distinct clusters. Shortest path indicates the closeness of tasks which are initiated from same job. The pseudo code for the multi source shortest path algorithm is given as under

---

Receive batches of jobs
$Buffer_i$=batches
For(i=1:N) where N is the total number of tasks within each batch
If($batch\_job_i\_Pid$== $batch\_job_{i+1}\_Pid$
 $Distance_i$=0
Else
$Distance_i$=1
End of if
End of for

The distance factor if 1 then job is from similar source. After finding the similarity jobs are merged and executed.

Result is obtained in terms of Makespan and Flowtime. Makespan is total time taken to execute each and every job. Flowtime on the other hand is obtained by determining execution of individual job. The steps are repeated and credits are assigned at each step of job execution. The simulation finishes when specified number of iterations are completed. Iteration with the highest credit value gives desired Makespan and Flowtime. Pseudo code for entire proposed system is given as under

---

Phase 1

- Index all the jobs using SJF-RRS-PBS with time quantum 4 and arrange them into temporary batch.

- Repeat the following until all jobs in cluster are processed
- Check whether resources are available in cluster or not, If available than
    - ➢ Allocate the resources and obtain the make span, flow time.
- Otherwise
    - ➢ Choose the new job from batch using Dynamic programming algorithm and update the jobs using SJF-RRS-PBS

  End if

  End Loop
- Go to Phase 2

Phase 2

- Shuffle the jobs in the clusters.
- Repeat the following steps until all jobs are processed
    - ➢ Apply dynamic programming to obtain optimal solution for jobs in cluster
    - ➢ Calculate the number of jobs for which optimal solution is obtained
    - ➢ Obtain the normalized function
- End Loop
- Apply credit depending upon the values of make span and flow time
- Go to phase 1 until required number of generations met.

The algorithm collaborate all the steps from 3.1 to 3.4 to determine optimal value of Makespan and Flowtime. Next section gives the results obtained and performance analysis.

IV. PERFORMANCE ANALYSIS AND RESULTS

Job scheduling within cloud environment is simulated through this research. The job relatedness is the main criteria along with the credit assigned to each iteration of batch execution. The process continues until specified number of times, simulation executed. Highest credit value assigned to iteration gives desired Makespan and Flowtime. Obtained results are given in this section.

First of all results are given in terms of Makespan

Number of Jobs	Without credit based System	With Credit based system
10	43	39
20	58	40
30	61	52
40	75	65
50	87	79

Table 1: Result in terms of Makespan

Plot for the Makespan is given as under

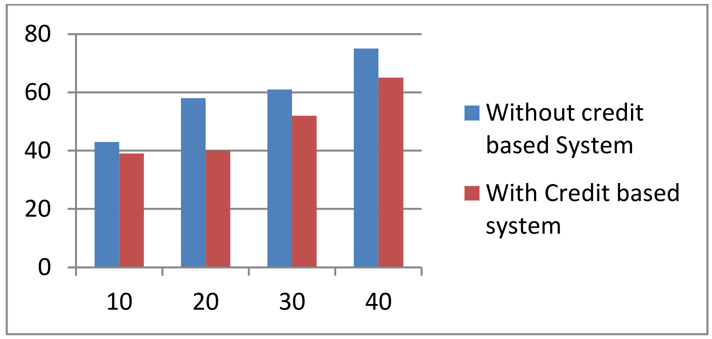

Figure 2: Makespan plot

Flowtime is generally smaller than the Makespan. The result obtained through proposed system and existing system is given as under

Number of jobs	Without credit based system	With credit based system
10	25	10
20	32	16
30	42	20
40	49	25
50	58	28

Table 2: Result in terms of Flowtime

In terms of plot Flowtime is given as under

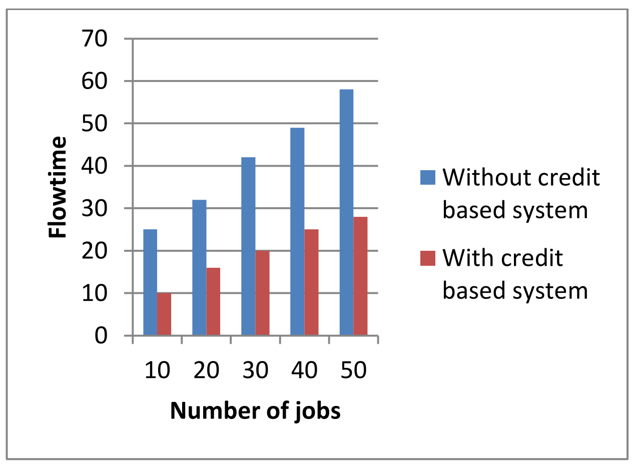

Figure 3: Flowtime plot

Results obtained from proposed system in terms of Makespan and Flowtime is better hence proving worth of the study.

V. CONCLUSION AND FUTURE SCOPE

Job scheduling becomes critical in order to utilize the resources efficiently. To this end, proposed system uses hybridization of multiple scalar schedulers along with PBS for efficiently forming a batch. Job execution is associated with the credit values. Schedule formed are entered into the execution mechanism and relatedness is identified by the application of multi source shortest path algorithm. Obtained results are in terms of Makespan and Flowtime. Result improvement of nearly 10% is observed.

In future, relatedness calculation mechanism can be further improvised by incorporating multiheuristic algorithm like genetic approach to improve Makespan and Flowtime.

VI. REFERNCES

[1]. Armbrust, M. et al., 2010. A view of cloud computing. *Communications of the ACM*, 53(4), p.50. Available at: http://portal.acm.org/citation.cfm?doid=1721654.1721672.

[2]. Barbosa, J. & Monteiro, A.P., 2008. A list scheduling algorithm for scheduling multi-user jobs on clusters. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5336 LNCS, pp.123–136.

[3]. Depoorter, W., Vanmechelen, K. & Broeckhove, J., 2014. Advance reservation , co-allocation and pricing of network and computational resources in grids. *Future Generation Computer Systems*, 41, pp.1–15. Available at: http://dx.doi.org/10.1016/j.future.2014.07.004.

[4]. Elghirani, A. et al., 2008. Performance enhancement through hybrid replication and genetic algorithm co-scheduling in data grids. *AICCSA 08 - 6th IEEE/ACS International Conference on Computer Systems and Applications*, pp.436–443.

[5]. Horng, S.C. & Lin, S.S., 2015. Integrating Ant Colony System and Ordinal Optimization for Solving Stochastic Job Shop Scheduling Problem. *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, 2015-Octob, pp.70–75.

[6]. Kaur, M., Sharma, S. & Kaur, R., 2014. Optimization of Job Scheduling in Cloud Computing Environment. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(7), pp.2277–128.

[7]. Kliazovich, D., Bouvry, P. & Khan, S.U., 2013. DENS: Data center energy-efficient network-aware scheduling. *Cluster Computing*, 16(1), pp.65–75.

[8]. Li, B. et al., 2014. Resource availability-aware advance reservation for parallel jobs with deadlines. , pp.798–819.

[9]. Mirashe, S.P. & Kalyankar, N. V, 2010. Cloud Computing N. Antonopoulos & L. Gillam, eds. *Communications of the ACM*, 51(7), p.9. Available at: http://arxiv.org/abs/1003.4074.

[10]. Rajvir Kaur, S.K., 2014. Analysis of Job Scheduling Algorithms in Cloud Computing. *International Journal of Computer Trends and Technology (IJCTT)*, 2(March), pp.12–22.

[11]. Ranganathan, K., Iamnitchi, A. & Foster, I., 2002. Improving data availability through dynamic model-driven replication in large peer-to-peer communities. *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGrid 2002*, pp.0–5.

[12]. Rodger, J.A., 2016. Informatics in Medicine Unlocked Discovery of medical Big Data analytics : Improving the prediction of traumatic brain injury survival rates by data mining Patient Informatics Processing Software Hybrid Hadoop Hive. *Informatics in Medicine Unlocked*, 1(2015), pp.17–26. Available at: http://dx.doi.org/10.1016/j.imu.2016.01.002.

[13]. Singh, D., Singh, J. & Chhabra, A., 2012. High availability of clouds: Failover strategies for cloud computing using integrated checkpointing algorithms. *Proceedings - International Conference on Communication Systems and Network Technologies, CSNT 2012*, pp.698–703.

[14]. Suri, P.K. & Rani, S., 2017. Design of Task Scheduling Model for Cloud Applications in Multi Cloud Environment. , 2, pp.11–24.

[15]. Switalski, P. & Seredynski, F., 2014. Scheduling parallel batch jobs in grids with evolutionary metaheuristics. *Journal of Scheduling*, 18(4), pp.345–357. Available at: http://dx.doi.org/10.1007/s10951-014-0382-0.

[16]. Xhafa, F. et al., 2011. A GA+TS hybrid algorithm for independent batch scheduling in computational grids. *Proceedings - 2011 International Conference on Network-Based Information Systems, NBiS 2011*, pp.229–235.

[17]. Yousif, A., Abdullah, A.H. & Nor, S.M., 2011. SCHEDULING JOBS ON GRID COMPUTING USING. *IEEE Access*, 33(2), pp.155–164.

Author Profles

Mr. Jasjit Singh pursued Bachelor of Computer Science from CTIT, India in 2016 and pursuing Master of Science from GND University in the year 2016. He has published more than 3 research papers in reputed international journals including. His main research work focuses on Cryptography Algorithms, Network Security, Cloud Security and Privacy, Big Data Analytics, Data Mining, IoT, and Computational Intelligence based education.

Mr. A Kumar pursued Bachelor of Computer Science from Punjabi University, India, and Master of Science from GND University in the year 2016. He is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computer Sciences, He has published more than 20 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences including IEEE and it's also available online. His main research work focuses on Cryptography Algorithms, Network Security, Cloud Security and Privacy, Big Data Analytics, Data Mining, IoT and Computational Intelligence based education. He has 5 years of teaching experience and 4 years of Research Experience.