

Satiating A User-Delineated Time Constraints While Scheduling Workflow in Cloud Environments

Pratiksha¹, Afsarudfin², Mohd Haroon³, and Faiyaz Ahamad⁴

¹ PG Scholar, Department of Computer Science and Engineering, Integral University, Lucknow, India

^{2,3,4} Assistant Professor, Department of Computer Science and Engineering, Integral University, Lucknow, India

Correspondence should be addressed to Pratiksha, pratiksha.vermag8@gmail.com

Copyright © 2021 Made Pratiksha et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT- Cloud computing is used to achieve sustainability in terms of computing. It reduces energy and resource consumption. Most of the companies have been moving their applications to the cloud to reduce power, energy re-source, and carbon emission. Today's computing landscape is rapidly shifting toward creating applications to leverage Cloud platforms to have necessary features such as elasticity, virtualization, low cost, and pay-per-use. Cloud computing's rising demand and versatility are achieving acceptance in the research community as a means of implementing large-scale electronic systems in the format of workflows (set of tasks). One of the most important objectives of this effort is to trim down makespan which is the total period taken by the resources to complete all workflow activities. Another foremost objective of this work is to satiate all the user-delineated time constraints while scheduling workflow activities.

KEYWORDS- Decisive Path, Zadeh's Max-Min rule, Makespan, Workflow Scheduling.

I. INTRODUCTION

Computing is being reimagined as a distributed utility model, similar to that of customary utilities such as water, power, gas, and telephony. Numerous computing paradigms have been developed to ensure the provision of utility-based computing, including Cluster Computing, Grid Computing, and, most recently, Cloud Computing. Cloud Computing is a category of internet-based computing in which shared services, applications, and information are made available on-demand to computers and other devices, similar to a public utility. Numerous cloud computing vendors, including Amazon, Yahoo, Google, Microsoft, Salesforce, and Zoho, provide cloud computing services. Cloud computing technology stores and manages data and applications through the internet and central remote servers. This technology enables customers and companies to use applications without having to install them and to access their files from any device such as laptop, PDA, terminal, or tablet. Today's computing environment is rapidly moving toward developing software through the use of Cloud platforms that provide required features such as pliability, virtualization, stumpy cost, and pay peruse. [1] Our objective is completed by the concepts of virtual organization [2]. further, the group of virtual organizations can be formed by supervised and unsupervised learning

algorithms, for the computing purpose virtual organization can be chosen randomly [2]. each virtual organization on the cloud has a queue that is used to store the load information, based on that load information process can be schedule [4].

II. WORKFLOW

Cloud Computing is rapidly gaining popularity as a paradigm for deploying a variety of scientific applications by lowering the overhead and expense of securing infrastructure services. The workflows declaratively associate the individual computing components with their input and output data, and hence these applications are represented as Cloud work-flows [2]. For the information extraction, the DOM traverse every node in recursively fashion and its filter out all nonrelevant information. Each of the filters can be turned on or off and customized to a certain degree [3]. The information extraction algorithms will start working from the root node of the tree and it used to split the entire documents into a block, that is known as a node of the tree, and every node has all HTML tags like tags which denotes a block namely table, tr, hr, and ul[3]. Cloud technologies have progressed toward a service-oriented model, allowing science applications to use Cloud services through the internet. Scientific applications include workflow processing, in which tasks are performed following their mission and data dependencies. Because the structure of different science workflow implementations varies, many workflows can be handled simultaneously on various Cloud services. Workflow Scheduling is a methodology that can be used either with best-performance or quality of service restrictions to allocate activities to cloud servers. The best-effort scheduling approach minimizes execution time while disregarding other considerations such as the cost of Cloud computing, while the QoS constraint scheduling strategy reduces reliability by focusing on the most significant QoS constraints, such as time minimization for budget constraints and cost minimization for time limit constraints [3]. The growth of the internet and the growth of data generated in the virtual world creating a lot of challenges, like resources sharing trying to share the application and thereby increased the amount of load and traffic across the internet [4]. In the process of scheduling a lot of security risk are also associated with that we have distinguished process by traditional of password-based authorization; more

exceedingly, they can be identified by the waitperson as botched login efforts and fortified alongside [5]

III. WORKFLOW MODEL

A guided acyclic graph $G(T,E)$ is used to model an application.

T is a collection of n tasks t_1, t_2, \dots, t_n .

E is a collection of arcs connecting two tasks.

each arc $e_{i,j} = (t_i, t_j)$ represents a precedence constraint

Dumb tasks: t_{entry} and t_{exit}

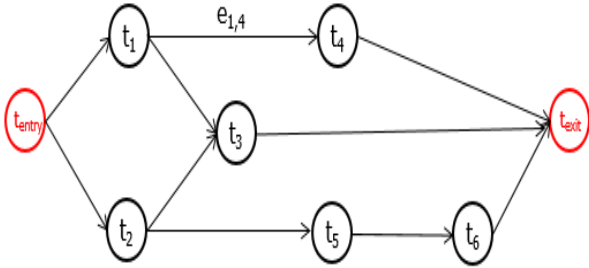


Fig. 1: Workflow Example

IV. LITERATURE REVIEW

Ritu Garg et al. [9] suggested the reliability and energy effectual workflow scheduling procedure, which optimises both application lifetime dependability and energy usage and promises the user-specified QoS restriction. This algorithm consists of four phases: task prioritisation, task clustering, goal time distribution, and cluster assignment to processing elements with the required voltage/frequency ratios. The simulation findings attained using arbitrarily produced task graphs and Gaussian Elimination task graphs demonstrate that the proposed solution is more efficient than current algorithms at optimising the system's lifetime reliability and energy consumption simultaneously. Pandey et al. [10] presented a guidance focused on PSO for applications that rely on computation outlay as well as data transmission. This algorithm is applicable to workflow implementations with varying computational and communication overheads. Experiments demonstrated that PSO can reduce costs and efficiently spread workloads to cloud services. Additionally, the authors extended PSO to include resource scheduling and provisioning based on deadlines [6]. However, these scholars made little attempt to justify resource shortages or an excessive reliance on essential activities. Attiqa Rehman et al.[7] developed a gap search algorithm for optimising cloud resource use. They contrasted their findings to those of genetic algorithms, taking budget, deadline, and energy usage into account separately. Additionally, they contrasted MOGA's efficiency to that of Multiobjective Particle Swarm Optimization (MOPSO) when the goals were the same as those of MOGA. The findings indicated that their algorithm MOGA dramatically improved not only in terms of schedule, deadline, and energy consumption, but also in terms of cloud resource usage.

V. PROPOSED WORKFLOW SCHEDULING APPROACH

Proposed algorithm has two stages. The first stage is

deadline allocation and second is workflow scheduling. Deadline allocation stage tackles the time constraints for overall workflow activities and scheduling stage is responsible for order of execution.

Followings are the prerequisites of proposed approach:

Each task can be processed by a number of services on different GSPs

$ET(t_i,s)$ and $EC(t_i,s)$: estimated execution time and execution cost for processing task t_i on service s

$TT(e_{i,j},r,s)$ and $TC(e_{i,j},r,s)$: Estimated time and cost of data transmission along $e_{i,j}$ from service s (processing task t_i) to service r (processing task t_j)

Minimum Execution Time:

$$MET(t_i) = \min_{s \in S_i} ET(t_i, s)$$

Minimum Transfer Time:

$$MTT(e_{i,j}) = \min_{s \in S_i, r \in S_j} TT(e_{i,j}, s, r)$$

Earliest Start Time:

$$EST(t_{entry}) = 0$$

$$EST(t_i) = \max_{t_p \in \text{parents}(t_i)} EST(t_p) + MET(t_p) + MTT(e_{p,i})$$

VI. ZADEH'S MAX-MIN RULE

If X is A then Y is B means $A \rightarrow B$, we can written the given expression by zadeh max min rule is given below

$$R_{mn} = (A * B) \cup (A' * Y)$$

Here Y is discourse with membership value for all $Yy \in Y$ and is 1 taht is $\mu y(Y) = 1 \forall y \in Y$

Suppose $X=\{a,b,c,d\}$ and $Y=\{1,2,3,4\}$ then we will consider

X is $ET(t_i,s)$ and Y is $EC(t_i,s)$, with the implication of zadeh max min rule can be written the relation as follow

$$R = (ET(t_i, s) * EC(t_i, s)) \cup AET(t_i, s)' * Y$$

Here Y is universe of discourse with membership value is one that is

$$\varphi y(Y) = 1 \forall y \in Y$$

Now suppose $ET(t_i,s)$ is a discourse fuzzy set of estimated execution time and $EC(t_i,s)$ fuzzy set of execution cost now $ET(t_i,s)$ represented in fuzzy form $\{(a,.0), (b,.8), (c,.6), (d,1.0)\}$ and $EC(t_i,s)$ is represented into another fuzzy set $\{(1,.2), (2,1.0), (3,.8), (4,.0)\}$ now with the help of zadeh max min rule , the execution time is minimized along with cost, for that , relation have to developed , now first perform Cartesian product between $EC(t_i,s)$ and $ET(t_i,s)$, then following result is calculated

$$ET(t_i, s) * EC(t_i, s) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ .2 & .8 & .8 & 0 \\ .2 & .6 & .6 & 0 \\ .2 & 1 & .8 & 0 \end{bmatrix}$$

Now another relation is calculated between cost and discourse of element from execution time that is matrix Y , element from y matrix having membership value is

$$AET(t_i, s)' * Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ .2 & .2 & .2 & .2 \\ .4 & .4 & .4 & .4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Finally, after apply zadeh max min theory for optimization of execution time along with cost,

Now relation matrix

$$R = (ET(ti, s) * EC(ti, s)) \cup AET(ti, s) * Y$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & .8 & .8 & .2 \\ .4 & .6 & .6 & .4 \\ .2 & 1 & .8 & 0 \end{bmatrix}$$

The above matrix showing the relation between estimated execution time and execution cost for processing task ti on service s, lower membership value showing the better scheduling of the process.

VII. PROPOSED SCHEDULING ALGORITHM SCHEDULE WORKFLOW(G(T,V), DEADLINE

Substantiate prevailing reckoning services add tentry , textit and their corresponding edges to G calculate ELT (ti), ECT (ti) and LFT(ti)for each task in G ALT(tentry) =0, ALT(textit) = Deadline mark tentry and textit as scheduled call algorithm Assign Parents (textit) If this procedure was successful make advance reservations for all tasks in G according to the schedule, otherwise return failure For process scheduling if any process have the execution time is high, then process avoid to schedule., now various fuzzy rule are applicable in course of process scheduling if execution time high the cost of calculation of process is high, if execution time is low the cost of calculation is also low. for thar a various fuzzy rule can be infer, the relation between execution time and cost can be draw like Execution time → execution cost

$$\int_{MEct * Ecost} M(Ect) * M(Ecost)$$

VIII. PARENTS SCHEDULING

The Critical Parent of a node t is the unscheduled parent p of t for which EST(p)+MET(p)+MTT(ep,t) is maximal The Partial Critical Path of node t is: empty if t does not have unscheduled parents consists of the Critical Parent p of t and the Partial Critical Path of p if t has unscheduled parents Critical parent and partial critical path change over time

IX. SCHEDULE PARENTS(T)

If t has no unscheduled parents then return success Let Critical Path be the partial critical path of tCall Schedule Path(Critical Path) If this procedure is unsuccessful, return failure and a suggested start time for the failed node (try to repair) For all ti on Critical Path /* from start to end */ Call Schedule Parents(ti) Iterate over all non-scheduled parents of t

X. SCHEDULE PATH

Schedule Path attempts to find the cheapest schedule for a Path while respecting the real start times of the activities on the Path's planned children. Schedule Path is based on a technique known as backtracking. A task-specific service is admissible if the exact start times of the task's planned children can be reached. It traverses the Path from the first to the last mission. For each mission, it selects an untested possible service. If the selected service generates an admissible (partial) schedule, it progresses to the next task;

if not, it chooses another untested service for that role.

If no other untested service is available for that purpose, it reverts to the route's first task and chooses another service. This may lead to failure

XI. PERFORMANCE EVALUATION

To appraise the proposed algorithm, we need to measure its performance in some workflow models. One preliminary work in same field was presented by Bharathi et al. [8]. They studied realistic workflow structure for various scientific applications; one of the structures we used in this article is the LIGO. Figure 4 shows the approximate structure of this workflow: To assess the performance of proposed algorithm, deadline for each workflow is set. We first delineate the fastest schedule. As the makespan of the best program in a workflow, indicated by MF, is only a least duration for that workflow to execute, so to establish time limit for workflows, we delineate the time constraint factor α and delineate the workflow's time constraints to the time of its advent plus $\alpha * MF$. As there is no elucidation for $\alpha = 1$, we take the range of α from 2 to 8 in our experiment. Figure 3 shows a comparison of the proposed approach with the stochastic approach and the PSO [11]. To evaluate the proposed algorithm, we must quantify its output in a variety of workflow models. Bharathi et al. [12] proposed a preliminary study in the same area. They investigated functional workflow mechanisms for a variety of scientific applications; one of the structures they examined is the LIGO, which we included in this article. The estimated form of this workflow is depicted in Figure 4: To evaluate the proposed algorithm's output, a deadline is established for each workflow. We begin by establishing the quickest schedule. As the makespan of the best programme in a workflow, denoted by MF, is just the minimum time required for that workflow to execute, we define the time constraint factor α and constrain the workflow's time constraints to the time of its inception plus $\alpha * MF$. Due to the lack of an explanation for $\alpha = 1$, we use the spectrum from 2 to 8 in our experiment.

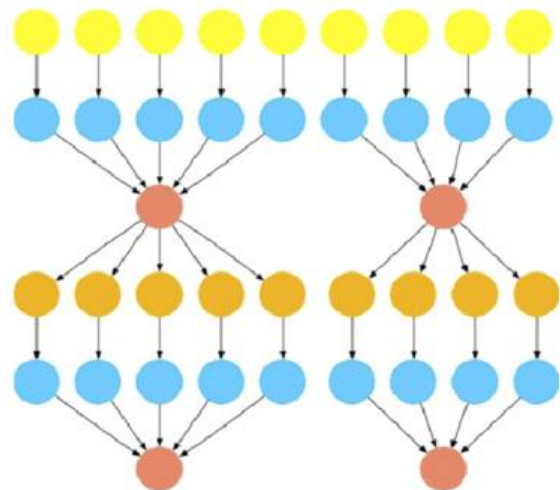


Fig. 2: Structure of LIGO workflow

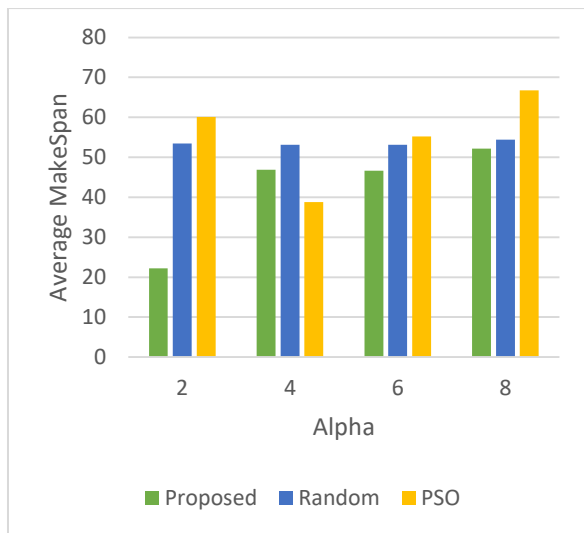


Fig. 3: Comparison

XII. CONCLUSION

cloud computing because it aims to ensure the complete execution of workflows while also taking into account their QoS requirements such as deadline and budget constraints. Scheduling a workflow's tasks to cloud resources is a well-known NP difficult problem. The stakeholders in a cloud environment have varying interests in resolving the scheduling problem. Along with the traditional objectives of timeline, budget, and deadline, optimising workflow scheduling and taking into account the green aspect of cloud computing (ie, energy consumption) adds to the problem's complexity. We considered the problem of workflow scheduling in a cloud environment and proposed a new algorithm for utility grid workflow scheduling that minimizes the total makespan while meeting a user-defined deadline. Scheduling, parent assignment, and path assignment are the three primary phases of the proposed algorithm. We conducted extensive simulation experiments on the structure of the LIGO workflow. Our experimental evaluation demonstrates that the system performs admirably in small and medium workflows.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

- [1] Zhang, G., Sun, J., Zhou, J., Wei, T. and Hu, S.. Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT. *Future Generation Computer Systems*, 93, pages.278-289,2019
- [2] M. Haroon and M. Husain, "Interest Attentive Dynamic Load Balancing in distributed systems," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 1116-1120.
- [3] S. Srivastava, M. Haroon, and A. Bajaj, "Web document information extraction using class attribute approach," Proc. - 4th IEEE Int. Conf. Comput. Commun. Technol. ICCCT 2013, pp. 17-22, 2013, doi: 10.1109/ICCCT.2013.6749596.
- [4] R. Khan, M. Haroon, and M. S. Husain, "Different technique of load balancing in distributed system: A review paper," *Glob. Conf. Commun. Technol. GCCT 2015*, no. Gcct, pp. 371-375, 2015, doi: 10.1109/GCCT.2015.7342686

- [5] M. S. Husain and D. M. Haroon, "an Enriched Information Security Framework From Various Attacks in the Iot," *Int. J. Innov. Res. Comput. Sci. Technol.*, vol. 8, no. 4, 2020, doi: 10.21276/ijrcst.2020.8.4.3.
- [6] Zhou, Junlong, Jin Sun, Mingyue Zhang, and Yue Ma. "Dependable scheduling for real-time workflows on cyber-physical cloud systems." *IEEE Transactions on Industrial Informatics*, 2020.
- [7] Shang, Q., "A Dynamic Resource Allocation Algorithm in Cloud Computing Based on Workflow and Resource Clustering", *Journal of Internet Technology*, 22(2), pp.403-411, 2021.
- [8] Garg, R. and Mittal, M., 2019. Reliability and energy efficient workflow scheduling in cloud environment. *Cluster Computing*, 22(4), pp.1283-1297.
- [9] Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, and Rajkumar Buyya. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. *Proceedings IEEE International Conference*, pages 400-407, May 2010.
- [10] Maria Alejandra Rodriguez and Rajkumar Buyya. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *Bibliography IEEE Transactions On Cloud Computing*, (2):222-235, May 2014.
- [11] Rehman, A., Hussain, S.S., ur Rehman, Z., Zia, S. and Shamshirband, S., 2019. Multi-objective approach of energy efficient workflow scheduling in cloud environments. *Concurrency and Computation: Practice and Experience*, 31(8), p.e4949
- [12] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. SuK.Vahi, Characterization of scientific workflows, in: *The 3rd Workshop on Workflows in Support of Large Scale Science*, 2008

ABOUT THE AUTHORS



Pratiksha, PG Scholar, Department of Computer Science and Engineering, Integral University, India



Afsaruddin, working as an Assistant Professor, Department of Computer Science and Engineering, Integral University Lucknow. Research interest: cloud computing, IOT, artificial intelligence.



Mohd. Haroon is working as Associate Professor in the Department of Computer Science & Engineering at Integral University, Lucknow (U.P), India. He has completed his PhD in "Dynamic Load Balancing in Distribution System" in 2016. research area: distributed machine learning, artificial intelligence.



Faiyaz Ahmad is working as Assistant Professor in the Department of Computer Science & Engineering at Integral University, Lucknow (U.P), India. He has completed his PhD in "Soft Computing". research area: Soft Computing machine learning, artificial intelligence, Cyber Security.