ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Skill Based Workflow Scheduling for Efficient Utilization of Resources

[1]K.S. Sathya, [2]P.Esakkiappan, [3]S.Renugadevi, [4]V.Samruthaa
[1]Assistant Professor, Computer Science and Engineering, Nandha Engineering College, Erode, India.
sathya.ks03@gmail.com
[2-4]UG Students, Computer Science and Engineering, Nandha Engineering College, Erode, India.
ssvrenuga@gmail.com

Abstract:-

Workflow applications are collections of several structured activities and computational tasks. Efficient workflow scheduling has become critical resulting in reduced scheduling overhead minimizes the cost and maximized resource utilization while meeting user specified overall deadline. The proposed system aims at deriving a system which maximizes the resource utilization, minimizing the financial cost and less scheduling overhead. The proposed approach splits the work flow into tasks and groups the tasks based on dependency of data and priority constraints. Thereafter optimizing the allocation and scheduling of resources in heterogeneous projects dynamically.

Index Terms— Efficient workflow, Scheduling, Minimal cost, priority, Resource allocation.

I. INTRODUCTION

Fields such as Astronomy, Medicine, Bioinformatics, Science and technology involves the use of development of large scale complex projects for validating behavior of different real world activities. Most of such projects are constructed as workflows with a set of dependent and constraint based computational tasks. Workflow scheduling is the process of mapping those tasks to resources and managing the development while satisfying all dependencies constrains and objective functionalities. It is well known that the workflow problems are NP –complete, so finding perfect solution is not feasible in all cases. However, development environments may incur significant computing overheads that can adversely affect the overall performance and costs of the workflow execution.

A dynamic and scalable cost-efficient deadline constrained algorithm for scheduling workflows using dynamic provisioning of resources is proposed.

II. RELATED WORKS

Rodriguez and Buyya [1] considered an application model similar to the one addressed in this paper and proposed a scheduling algorithm for optimizing a project execution time (that is, makespan) under a budget constraint. They proposed resource provisioning plan for a subset of the project tasks as a Mixed Integer Linear Programming model. However, their solution is tailored for fine-grained billing periods such as per-minute billing.

Moreover, the algorithm's objective is to minimize the makespan instead of meeting a deadline constraint. Despite these differences, we consider their work relevant as the authors not only consider dynamic resource provisioning and delays but also consider several VM instance types with different characteristics and prices.

Dziok et al. [2] presented an adaptive algorithm and used the MIP approach to schedule tasks in IaaS clouds for optimizing cost under a deadline constraint. However, they do not consider reusing of already assigned VMs and also do not consider data transfer time.

Copyrights © International Journal of Intellectual Advancements and Research in Engineering Computations, www.ijiarec.com

Malawski et al. [3] presented a dynamic resource provisioning and scheduling algorithm called DPDS. They explicate a deadline and budget constraint scheduling algorithm which tries to maximize the amount of work completed.

The scheduling algorithm proposed by Byun et al. [4] estimated the minimum number of resources needed to execute the workflow in a cost- effective way. But, they considered a single type of VM and fails to consider the heterogeneous nature of clouds. It schedules workflows under given deadline and budget constraints along with the information about resource utilization for VM provisioning and scheduling. However, it attempts to maximize the number of completed tasks rather than minimizing the rental cost of a single BoT workflow.

Moreover, it supports single instance type and do not consider resource heterogeneity which is in contrast to the current IaaS cloud model that offer a wide variety of instance types and dynamic provisioning and scheduling of resources. Task granularity has been addressed in several research studies for reducing the impact of overheads that may arise when executing scientific workflows in distributed environments, such as the cloud. Task grouping methods reduce computational granularity by reducing the number of computational activities by grouping fine-grained tasks into course-grained tasks. These techniques attempt to minimize queuing times and scheduling overheads when resources are limited. However, task granularity may limit the degree of inherent parallelism, therefore it must be done optimally.

Mao and Humphrey [5] proposed an auto-scaling mechanism that schedules jobs in order to minimize the monetary cost while meeting the application deadlines on clouds. A static resource provisioning plan is made based on a global optimization heuristic and then adapted to dynamic changes by running the global optimization algorithm every time a task is scheduled. However, they do not consider the different priorities of each job and considered dynamic and unpredictable workloads of workflow. Furthermore, the high computational overhead in Scaling-Consolidation-Scheduling (SCS) hinders its scalability in terms of the number of tasks in the workflow and they do not provide a near-optimal solution.

The proposed algorithm covers all of these deficits and presents a dynamic and scalable cost-efficient deadline constrained algorithm for scheduling workflows using dynamic provisioning of resources.

III. PROPOSED SYSTEM

The proposed system provides a model which represents defects and presents a cost efficient constrain based algorithm for scheduling workflow. The work flow is described as a directed Acyclic Graph where the notes denote separate task or job and the edges denote the dependency, control and constrains between the tasks.

The proposed technique for scheduling scientific workflows is achieving high success rates with lower cost while satisfying the deadline constraint. Thereafter optimizing the allocation and scheduling of resources in heterogeneous projects dynamically. The main components of the project workflow scheduling environment are given below:

1. Project submission
2. Project mapper
3. Cluster engine
4. Project scheduler
5. Execution environment

1. Project submission

The user submits the projects to the Workflow Management System for scheduling of tasks.

2. Project mapper

The Project mapper produces an executable workflow on the basis of the project submitted by the user. It identifies the appropriate resource required for the implementation of a task. Moreover, the mapper can restructure the workflow for performance optimization.

3. Cluster engine:

To reduce resource overheads, one or more small tasks are clustered into single execution unit called job. The engine submits the jobs the mapper in order of their dependency constraints. Thereafter, the jobs are submitted to the project scheduler

4. Project scheduler

The project workflow scheduler manages and supervises individual jobs using the resources. The elapsed time between the submission of a job to the job scheduler and its execution in a remote compute node (potentially on cloud) is denoted as the queue delay.

5. Execution environment

On the basis of the available resources, the Workflow Management System maps the given abstract workflow into an executable workflow and executes them. Moreover, it monitors the execution and manages the input data, intermediate files and output files of the workflow tasks.

The diagrammatic representation of the scheduling environment is represented in the figure [1] below.

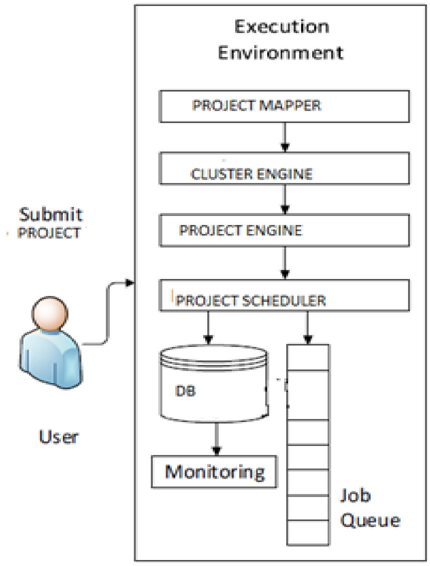

Figure 1: Project workflow scheduling environment architecture

IV. CONCLUSION

The efficient workflow scheduling algorithm has been proposed for the dynamic allocation of resources considering constrained based cost minimization to meet user defined deadlines. The proposed approach considers the environment of heterogeneity and delays. The mathematical model of director acyclic graphs is used effectively for the dynamic utilization of resources.

REFERENCES

1. Rodriguez, M.A. Buyya, R. **“A taxonomy and survey on scheduling algorithms for scientific workflows in iaas cloud computing environments**”. Concurr. Comput. Pract. Exp. 2017, 29, 1–32.

2. Dziok, T. Figiela, K. Malawski, M. “**Adaptive multi-level workflow scheduling with uncertain task estimates**”,in Parallel Processing and Applied Mathematics; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9574, pp. 90–100.

3. Malawski, M. Juve, G.Deelman, E.Nabrzyski, “**Cost- and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds**.”, in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Salt Lake City, UT, USA, 10–16 November 2012; pp. 1–11.

4. Byun, E.-K.Kee, Y.-S. Kim, J.-S. Maeng, S,“**Cost optimized provisioning of elastic resources for application workflows**”. Future Gener. Comput. Syst. **2011**, 27, 1011–1026.

5. Mao, M. Humphrey, M. “**Auto-scaling to minimize cost and meet application deadlines in cloud workflows”.** In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Seatle, WA, USA, 12–18 November 2011.

6. Ullman, J.D. “**Np-complete scheduling problems**”. J. Comput. Syst. Sci. 1975, 10, 384–393.

7. Deelman, E. Vahi, K. Juve, G.Rynge, M. Callaghan, S. Maechling, P.J. Wenger, K. Pegasus,**“A workflow management system for science automation**”. Future Gener. Comput. Syst. **2015**, 46, 17–35.

8. Armbrust, M.Fox, A. Griffith, R. Joseph, A.D. Katz, R. Konwinski, A. Zaharia, M. “**A view of cloud computing**”. Commun. ACM **2010**, 53, 50–58.

9. Ostermann, S. Iosup, A. Yigibasi, N.; Prodan, R. Fahringer, T. Epema, D. “**A performance analysis of EC2 cloud computing services for scientific computing. In Cloud Computing**”; Lecture Notes of the Institute for Computer Sciences, SocialInformatics and Telecommunications Engineering; Springer: Berlin/Heidelberg, Germany, 2010; Volume 34.

Copyrights © International Journal of Intellectual Advancements and Research in Engineering Compu tations,
www.ijiarec.com