International Journal of Engineering & Technology, 7 (3.28) (2018) 271-274

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Scientific Workflow Scheduling in Clouds: A Review

Tawfiq Alrawashdeh[1], Aznida Hayati Zakaria[2]*, Zarina Mohamad[2]

[1]Al Husein Bin Talal University, P.O. Box 20 Ma'an, Jordan
[2]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut Campus, Terengganu, Malaysia
**Corresponding author E-mail: aznida@unisza.edu.my*

Abstract

Due to their abundant resources that can be elastically provisioned with pay-as-you-go pricing, clouds have emerged as a promising cost-efficient platform to execute large scale scientific applications. Such applications consist of number of processes/tasks forming workflow. These tasks are connected by direct edges that show the data dependency between the tasks. Tasks perform their computation on the original data submitted by the user, or on data passed by its predecessor task. This work, classify and discuss proposals that investigate the problem of scheduling scientific workflows in clouds.

Keywords: *scheduling algorithms; scientific workflow; cloud computing.*

1. Introduction

With the rapid increment on the complexity of the workflow, and the resultant demand on the scalability of the environment, executing workflows on traditional environment become a very challenging task. To address this issue, cloud computing (especially Infrastructure as a Service) has emerge as an efficient environment to execute scientific workflows.

Infrastructure as a Service (IaaS) cloud offer an effectively available, adaptable, and versatile foundation for the arrangement of scientific workflows. Using IaaS, user can rent Virtual Machines (VMs) to execute their computational tasks. This allows user access to a practically unbounded pool of VMs that can be flexibly gained and discharged during the execution of the computational task(s). In this service, users are charged based on the number of resources they rent using pay-per-use cost model.

In this direction, to increase the utilization of the resources, must determine the right number of resources to rent. Over-renting is expected to increase the execution cost, and under-renting is expected to reduce the performance (increase execution time). This problem is underlined due to the bi-objective scheduling problem presented by scheduling workflows in cloud. Generally, want to establish an execution schedule for the tasks of the workflows such that the execution cost and time is minimized. This conflict in the objective results in highlighting the process of determine the number of resources to rent as a major challenge.

The problem of scheduling workflows in clouds is NP-Complete in nature. Many variations of this problem have been proposed in the literature. For instance, many proposals have investigated the problem of finding the cheapest schedules that satisfy pre-determined deadline on the execution of the workflow [1-5, 7]. On the other hand, in [6, 9-11], the authors addressed the problem of determining the schedule of executing the tasks such that the execution time is minimized and pre-determined execution cost constraint is satisfied. Introducing budget or/and deadline constraints results in reducing the optimization space to one of the objectives (cost, time).

This paper, discuss and classify proposals that investigate the problem of scheduling scientific workflows in clouds. Section 2 presents the most related approaches to this problem, and in section 3, conclude this paper.

2. Scheduling Approaches

Many proposals [1-24] have investigated the problem of scheduling scientific workflows in cloud. In this section, this paper discusses the related proposals in the literature. This paper categorizes these proposals in term of their objective function into: (1) single-objective, and (2) multi-objective.

2.1. Single-objective

The problem of scheduling scientific workflows on with the objective of minimizing the makespan or cost minimization has been studied extensively in the literature. For instance, given a pre-determined budget-constraint, in [1] propose a heuristic-based solution that aim to reduce the overall execution delay. In each iteration, the main idea of this approach is to improve the current schedule by considering the left budget. In [3] addressed the problem of minimizing the makespan under the presence of budget constraints. They termed this problem as the Minimum End-to-end Delay under Cost Constraint (MED-CC) problem. The authors addressed the hardness of this problem by proving that it is NP-Complete, and it is non-approximable. To address this problem, the authors proposed heuristic-based solution. This heuristic follows an efficient searching strategy, where in each iteration it tries to find a new schedule such that the makespan is minimized.

In [4] proposed priority based genetic algorithm termed as BCHGA, which address the problem of scheduling the tasks of the workflow under the presence of the budget constraint. In this algorithm, based on the locality of the tasks, each task is assigned either bottom level priority (b-level), or top-level priority (t-level). Then, in each round, the algorithm by trying to find better sched-

ule in term of makespan, while minimizing the execution cost. In [2, 5], the authors adopt similar strategy to minimize the execution cost under the presence of the budget and execution time constraints.

In [6] proposed a Deadline guarantee enhanced scheduling algorithm (DGESA). This algorithm target scenario where scientific workflows can be scheduled on hybrid. This algorithm starts by calculating the sub-deadline for each task. For each task, once the sub-deadline is calculated, the algorithm proceeds by determining the probability of violating thi s deadline. For each task, if the probability of violating the deadline is higher than pre-determined threshold, this task will be scheduled to be executed on public cloud. Otherwise, the task will be executed on private cloud.

In [7] also addressed the problem of minimizing the execution time under the presence of the budget constraint. To address this problem, the authors' proposed budget-driven approach, which start by partitioning the workflow into bags of tasks, where each bag contains a group of parallel tasks. Then each bag is scheduled based on the characteristics and locality of its tasks. This is established by modelling the resource provisioning plan for each bag of tasks as a mixed integer linear programming (MILP) model.

In [8] proposed a new pulling-based workflow execution system with a profiling-based resource provisioning strategy (DEWE v2). This system consists of master and worker daemons, which can be run by the same node or different nodes. The master daemon is responsible for managing the progress of executing the workflows. The worker daemon informs the master daemon whenever a task has been successfully executed. In addition, this system has worker submission application, which can be used by the scientist to submit workflows

In [9], the authors proposed the IC-PCP algorithm that to schedule the workflow's tasks such the execution cost is minimized and a pre-determine execution deadline is satisfied. This algorithm starts by distributing the workflow deadline across the entire workflows tasks. Starting from the critical path tasks, the sub-deadline for these path tasks will be determined. Then, the algorithm proceeds by determining the deadlines for all sub-paths. Once the deadline for each task is determined, each task will be schedule on the resource that result in satisfying the task deadline and reducing the execution cost.

In [10], the authors propose scheduling strategy designed mainly for WaaS, which aim to minimize the execution cost while satisfying a pre-determined time-deadline. The main idea of this algorithm is to determine each task sub-deadline. Then, once a task become ready for execution, if no available VM can be used to schedule this task without violating its deadline, a new VM will be leased.

In [11], the authors proposed an approximation algorithm to address the problem of scheduling scientific workflows such that the execution cost is minimized, while satisfying a pre-determined time deadline. This algorithm works in iterative fashion, where in each iteration the scheduler tries to find the maximum number of tasks that can be executed at a specific node. This is established by considering the latest acceptable finishing time for each task.

2.2. Multi-objective

Now, this paper discusses in details proposals that deal with bi-objective problems. This paper focus on the problem of minimizing the execution cost and time. This problem has conflicted objective, since reducing the makespan results in increasing the cost, and reducing the cost result in increasing the makespan.

In [12], proposed solution that address the problem of minimizing both the execution cost and execution time. Their approach can be considered as an extended version of the well-known Heterogeneous Earliest Finish Time (HEFT) heuristic, and it attempt to establish a trade-off between execution time and execution cost. This main idea of this heuristic is to assign a priority value for each task. This priority value is determine based on the locality of each task, and it aim to prioritize executing tasks on VMs such that executing cost and time is minimized.

In [13], the Deadline-Budget Workflow Scheduling (DBWS) algorithm is presented to address the problem of scheduling scientific workflows in clouds, while the execution cost and time are minimized. This algorithm assume that the user specifies pre-determined time and cost deadlines. Makespan is handled as a primary objective and thus the resultant schedule must always satisfy the deadline constraint. In this algorithm, cost is handled as a secondary objective, where the cost constraint can be violated. At its core, this algorithm tries to find the cheapest schedule that can satisfy the deadline constraint.

In [14], the authors proposed dynamic auction-based workflow scheduling algorithm that dynamically allocate the workflow's tasks across multiple cloud domains. The objective of this algorithm is to reduce the execution cost, and satisfy the execution time constraint. In this algorithm, each task is ranked based on its level load, and its successor rank. In this algorithm, task with submit their input and output requirement, and resources will bid their computational capacity. In this paradigm, tasks will be assigned to resources with higher computational capacity.

In [15] consider a multi-cloud domain, where every supplier contributes a fixed number of heterogeneous VMs. In addition, they provide global storage service to store intermediate data files. The authors formulate the scheduling problem as a Mixed Integer Program (MIP). Then they proposed two algorithmic solutions to address two different scenarios. In the first scenario, they assume that the running time for each is in hour unit, where in the second scenario they assume that each task running time is less than an hour. They start by partitioning the tasks based on their level. In addition, they assumed that any VMs cannot be allocated different partitions. This simplify the execution of the proposed MIP. However, by limiting the solution space, this approach reduces the optimization space. In this direction, in [16] presented a clustering approach to schedule workflows in clouds, where the tasks levels are the key factor behind the schedule.

In [17] presented a dynamic resource provisioning and scheduling algorithm DPDS. This algorithm aims to schedule multiple workflows simultaneously on the cloud. Such schedule is established with the objective of satisfying the users cost and time constraints. It attempts to maximize the number of completed workflows, and does not take into account minimizing the execution cost. Furthermore, it assumes that the available resources have the same computational and communication capabilities.

In [18] presented the Workflow scheduling on Hybrid Cloud to maintain Data Privacy (WHPD) algorithm. The objective of this algorithm is to maintain the privacy of the scheduled tasks such that the makespan constraint is satisfied. This algorithm handled minimizing the execution cost as a secondary objective. This algorithm starts by determining the latest starting time for each task, in order to satisfy the pre-determined time-deadline. Then, each task will be allocated to the VM such that the gap between the task earliest execution time and the actual execution time is minimized. The last step of this algorithm is to attempt improving the obtained schedule by moving tasks between VMs in order to reduce the execution time. The performance of this algorithm depends on the initial schedule since, this schedule establishes constraints on the allowed movements in the schedule improvement step.

In [19], the authors proposed the Cost with Finish Time-Based (CwFT) Algorithm. This algorithm is an extended version of the HEFT algorithm, and it aim to reduce the execution time and cost. This algorithm consists of two phases: (1) task prioritizing and (2) node selection. In the task prioritizing phase, the priority of each task will be determined based on its locality. In the node selection phase, each task will be assigned to the VM with the objective of minimizing that ratio of executing cost and time. In [20], the authors investigated the same problem, and they also proposed an auto-scaling algorithm, which adopt the strategy of the well-known HEFT algorithm.

Several authors [21-24] have investigated the problem of scheduling scientific workflows in cloud using nature-inspired algorithm. Although techniques like Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), and Genetic Algorithms

(GA), can lead to find a near optimal solution, its main issue is practicality. Such techniques may require relatively large processing time, and this reduces the scalability of these approaches.

Table 1: Summary table of related work on workflow scheduling

References	Algorithm	Environment Tools	Optimization Strategy	Constraints
[1]	Critical-Greedy(CG)	CloudSim	Heuristic	single-objective
[2]	BPSO	Synthetic workflows	Heuristic	single-objective
[3]	MED-CC	CloudSim	Heuristic	single-objective
[4]	BCHGA	Java	Metaheuristic	single-objective
[5]	PDC and DCCP	Compared to existing algorithms	Heuristic	single-objective
[6]	DGESA	Simulation environment of hybrid computing	Heuristic	single-objective
[7]	BAGS	Synthetic workflows	Heuristic	single-objective
[8]	DEWE v2	Pegasus	Metaheuristic	single-objective
[9]	IC-PCP and IC-PCPD2	Synthetic workflows	Heuristic	single-objective
[10]	EPSM	Synthetic workflows	Heuristic	single-objective
[11]	EES	Synthetic workflows	Heuristic	single-objective
[12]	BDHEFT	Synthetic workflows	Heuristic	multi-objective
[13]	DBWS	Synthetic workflows	Heuristic	multi-objective
[14]	Novel replication aware dynamic workflow scheduling	CloudSim toolkit	Heuristic	multi-objective
[15]	Mathematical models	Amazon EC2	Heuristic	multi-objective
[16]	Efficient workflow scheduling algorithm	WorkflowSim and CloudSim	Heuristic	multi-objective
[17]	DPDS	CloudSim	Heuristic	multi-objective
[18]	WHPD	Synthetic workflows	Heuristic	multi-objective
[19]	CwFT	Gaussian Elimination program	Heuristic	multi-objective
[20]	HEFT	Compare with approaches	Heuristic	multi-objective
[21]	Algorithm based on the meta-heuristic optimization technique	Synthetic workflows	Metaheuristic	multi-objective
[22]	CEGA	Synthetic workflows	Metaheuristic	multi-objective
[23]	A combined resource provisioning and scheduling strategy	CloudSim framework	Metaheuristic	multi-objective
[24]	A hybrid genetic algorithm	WorkflowSim	Metaheuristic	multi-objective

3. Conclusion

This paper discussed the problem of scheduling scientific workflow in clouds under the presence of budget and/or deadline constraints. Based on the objective function of the problem, this paper is mainly interested in three variations of this problem. In the first problem, discussed proposals related to the problem of minimizing the execution cost. In this problem, the users are typically main concern with the cost of the execution and have no restriction on the execution time. In the second problem, focus on minimizing the execution time. In this problem, users target minimizing the makespan, and this may lead to increase the execution cost. In the last problem, focus on proposals related to minimizing the execution cost and time. In this problem users try to find a solution that balance the importance of time and cost.

References

[1] Wu, C., Lin, X., Yu, D., Xu, W., & Li, L. (2015). End-to-end delay minimization for scientific workflows in clouds under budget constraint. IEEE Transactions on Cloud Computing, 3(2), 169-181.

[2] Verma, A., & Kaushal, S. (2015). Cost minimized PSO based workflow scheduling plan for cloud computing. I.J. Information Technology and Computer Science, 8, 37-43.

[3] Lin, X., & Wu, C. Q. (2013, October). On scientific workflow scheduling in clouds under budget constraint. Proceedings of the IEEE 42nd International Conference on Parallel Processing, pp. 90-99.

[4] Verma, A., & Kaushal, S. (2013). Budget constrained priority based genetic algorithm for workflow scheduling in cloud. Proceedings of the IET Fifth International Conference on Recent Trends in Information, Telecommunication and Computing, pp. 8-14.

[5] Arabnejad, V., Bubendorfer, K., & Ng, B. (2017). Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources. Future Generation Computer Systems, 75, 348-364.

[6] Luo, H., Yan, C., & Hu, Z. (2015). An Enhanced Workflow Scheduling Strategy for Deadline Guarantee on Hybrid Grid/Cloud Infrastructure. Journal of Applied Science and Engineering, 18(1), 67-78.

[7] Rodriguez, M. A., & Buyya, R. (2017). Budget-driven scheduling of scientific workflows in IaaS clouds with fine-grained billing periods. ACM Transactions on Autonomous and Adaptive Systems, 12(2), 1-22.

[8] Jiang, Q., Lee, Y. C., & Zomaya, A. Y. (2015). Executing large scale scientific workflow ensembles in public clouds. Proceedings of the IEEE 44th International Conference on Parallel Processing, pp. 520-529.

[9] Abrishami, S., Naghibzadeh, M., & Epema, D. H. (2013). Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. Future Generation Computer Systems, 29(1), 158-169.

[10] Rodriguez, M. A., & Buyya, R. (2018). Scheduling dynamic workloads in multi-tenant scientific workflow as a service platforms. Future Generation Computer Systems, 79, 739-750.

[11] Ma, Y., Gong, B., Sugihara, R., & Gupta, R. (2012). Energy-efficient deadline scheduling for heterogeneous systems. Journal of Parallel and Distributed Computing, 72(12), 1725-1740.

[12] Verma, A., & Kaushal, S. (2015). Cost-time efficient scheduling plan for executing workflows in the cloud. Journal of Grid Computing, 13(4), 495-506.

[13] Ghasemzadeh, M., Arabnejad, H., & Barbosa, J. G. (2017). Deadline-budget constrained scheduling algorithm for scientific workflows in a cloud environment. Proceedings of the LIPIcs-Leibniz International Proceedings in Informatics, pp. 1-16.

[14] Gayathri, T., & Subashini, B. V. (2015). Task ranking based allocation of scientific workflows in multiple clouds with deadline constraint. International Journal of Engineering and Computer Science, 4(2), 10543-10546.

[15] Malawski, M., Figiela, K., Bubak, M., Deelman, E., & Nabrzyski, J. (2015). Scheduling multilevel deadline-constrained scientific workflows on clouds based on cost optimization. Scientific Programming, 2015, 1-13.

[16] Prathibha, D. A., Latha, B., Sumathi, G., Vani, R., Sangeetha, M., Davis, P., Nithyanandam C, Mohankumar G, Suratanee A, Lertsari N, & Kamphasee, S. (2014). Efficient scheduling of workflow in cloud environment using billing model aware task clustering. Journal of Theoretical and Applied Information Technology, 65(3), 595-605.

[17] Malawski, M., Juve, G., Deelman, E., & Nabrzyski, J. (2015). Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds. Future Generation Computer Systems, 48, 1-18.

[18] Abrishami, H., Rezaeian, A., & Naghibzadeh, M. (2015). Workflow scheduling on the hybrid cloud to maintain data privacy under deadline constraint. Journal of Intelligent Computing, 6(3), 92-103.
[19] Man, N. D., & Huh, E. N. (2013). Cost and efficiency-based scheduling on a general framework combining between cloud computing and local thick clients. Proceedings of the IEEE International Conference on Computing, Management and Telecommunications, pp. 258-263.
[20] Jiping, Z., Chunhua, G., & Feng, W. (2014). HEFT based cloud auto-scaling algorithm with budget constraints. International Journal of Advances in Computer Science and Technology, 3, 13-18.
[21] Goyal, M., & Aggarwal, M. (2017). Optimize workflow scheduling using hybrid ant colony optimization (ACO) and particle swarm optimization (PSO) algorithm in cloud environment. International Journal of Advance Research, Ideas and Innovations in Technology, 3(2), 1-9.
[22] Meena, J., Kumar, M., & Vardhan, M. (2016). Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint. IEEE Access, 4, 5065-5082.
[23] Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Transactions on Cloud Computing, 2(2), 222-235.
[24] Kaur, G., & Kalra, M. (2017). Deadline constrained scheduling of scientific workflows on cloud using hybrid genetic algorithm. Proceedings of the IEEE 7th International Conference on Cloud Computing, Data Science and Engineering-Confluence, pp. 276-280.