Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 16(4): 264-270, Dec. 2018 Regular paper

Workflow Scheduling Using Heuristic Scheduling in Hadoop

Chintureena Thingom[1*], Ganesh Kumar R[2], and Guydeuk Yeon[3], *Member, KIICE*

[1]Centre for Advanced Research & Training, CHRIST (Deemed to be University), Bengaluru 560029, India
[2]Faculty of Engineering, CHRIST (Deemed to be University), Bengaluru 560029, India
[3]Innovation Centre, CHRIST (Deemed to be University), Bengaluru 560029, India

Abstract

In our research study, we aim at optimizing multiple load in cloud, effective resource allocation and lesser response time for the job assigned. Using Hadoop on datacenter is the best and most efficient analytical service for any corporates. To provide effective and reliable performance analytical computing interface to the client, various cloud service providers host Hadoop clusters. The previous works done by many scholars were aimed at execution of workflows on Hadoop platform which also minimizes the cost of virtual machines and other computing resources. Earlier stochastic hill climbing technique was applied for single parameter and now we are working to optimize multiple parameters in the cloud data centers with proposed heuristic hill climbing. As many users try to priorities their job simultaneously in the cluster, resource optimized workflow scheduling technique should be very reliable to complete the task assigned before the deadlines and also to optimize the usage of the resources in cloud.

Index Terms: Cloud, Data Centers, Hadoop, Heuristic hill climbing, Workflow scheduling

I. INTRODUCTION

A business's economies of scale can be achieved if there is possibility of processing large information in a very short response and completion time. As such, a large scale analytics in cloud datacenter is an essential in our dynamic business environment. There are many cloud vendors who provide Hadoop based MapReduce infrastructure which deals with the analytical services to many client including corporates. Whenever there is a job or task assigned, a new and unique virtual machines (VMs) will be created and the task will be executed in that distinct VM. It is important to understand that the workflows are the set of task or jobs with a specific deadline for the completion of the task. As per the existing scheduling technique, there is not much productivity for the task or the job assigned as some of the jobs from the workflow will be still kept in the VM though it is executed. It will be combined when the other jobs are complete and then integrate for the result. The scheduling process of the workflow in the VMs should be allocated effectively in the data center so as to use the resources in the most efficient manner. The factors of the resources mentioned are the number of the VMs used, data center bandwidth and the consumption of the electrical units. We need to ensure that the deadline of the job assigned in the workflows should never be compromised while we optimize the factors in data center.

With this research work, a new approach "stochastic hill climbing based" will be introduced for the resource optimization for task or jobs assigned in the workflows to reach our desired outcome. Previous works on the method did not yield much desired outcome as most of the resource optimization method work only when they consider flat network or

Received 12 August 2018, **Revised** 18 September 2018, **Accepted** 20 September 2018
***Corresponding Author** Chintureena Thingom (E-mail: chintureena.thingom@christuniversity.in, Tel: +91-9008929794)
Centre For Advanced Research & Training, CHRIST (Deemed to be University), Bengaluru 560029, India.

Open Access https://doi.org/10.6109/jicce.2018.16.4.264 print ISSN: 2234-8255 online ISSN: 2234-8883

don't consider network topology at all. We have factored-in the network inter-relationship to have the best optimization bandwidth and with it, it makes our research unique. The workflow scheduling helps the system to run effectively in cloud as the resource allocation and faster execution time for the job assigned in the VM can be completed and is reliable.

The remainder of the paper is structured as follows. We begin by the study of literature in Section II and then, look at the problem statement in Section III. Section IV. System model & methods has been described and to complete with a mathematical model, it is shown at Section V. The analysis and its result is shown in Section VI, and Section VII conclusion has been added.

A. Related Work

We looked at the previous literature and the current scheduling methodology for their weakness in terms of the workflows.

M. Islam had considered a new scheduling process for the workflows with four different streams, i.e., scalability, security, multi-tenancy, and operability. With this methodology, they had proposed all scalability solutions to meet the objectives. And it has made the client or the user to access only the authorized resource after processing. But for Hadoop based jobs, the cost structure has not been prioritized while scheduling the workflows and the user expects cost effectiveness in the VM [1].

Cascading is being used in the process of scheduling the jobs or task assigned Hadoop platforms by Wensel [2]. It is a cluster which is meant for developing the complex workflow after considering the application of the data processing service. It is more of capacity building which makes the process smoother but scheduling them effectively is not considered.

Olston et al. [3] had introduced NOVA in his paper for more efficient data processing. It ensures that the workflows are distributed in much more reliable manner using an independent NOVA scheduling. Workflows are divided into specific groups and are executed at different times/rates with no restriction on deadline or on cost effectiveness in the NOVA platform.

Dong and Akl [4] proposed a new scheduling algorithm for workflows in his research work for the desired outcome of better performance and execution of the workflows within the deadline. It is called PFAS (resource-performance-fluctuation-aware) and is proposed for grids. The scheduling does not consider the cost factor when choosing the nodes, it only considers the deadline of the task and executes it.

Research work by Wang et al. [5] had merged two of the best scientific workflow systems, i.e., Kepler & Apache Hadoop. It is merged to get the benefit of these two setups. In Kepler, MapReduce actor was introduced to execute the tasks in workflow in MapReduce platform. But scheduling the tasks to meet deadline and in optimum cost is not considered.

For big data, a workflow scheduler for data processing directed cyclic graph (DAG) MapReduce jobs is proposed in the research work done by Tang et al. [6]. The technique is comprised of two unique events, i.e., job prioritizing phase and task allotment phase. Job priority for execution is completely based on whether the task requires computation intensive or I/O intensive. The jobs with higher I/O intensive need to be executed first, hence they are allocated nearest to the data locality which ensures that the deadlines are met. The scheduler is not generic and does not consider cost effectiveness.

Krish et al. [7] introduced a methodology specific for Map-reduce jobs wherein the hardware scheduling is proposed. The scheduling approach considers task allocation to suitable node, data caching to improve locality. By these two approaches, efficiency in execution time can be increased and brought near to deadline.

Xu et al. [8] had proposed adaptive task scheduling strategy based on dynamic workload adjustment (ATSDWA) algorithm with which the workflows can be scheduled on its own. In this work, based on the capabilities, the jobs are assigned to the nodes. This research does not consider task deadline and cost effectiveness but load balancing and scalability are dealt extensively.

An algorithm was proposed by Chen et al. [9] in which the scheduling is based on speculative execution. In this slow jobs are found frequently and backup for them is started in a more capable node to speed up and meet the deadline. But the resource consumption in this algorithm is uncontrolled.

Crawl et al. [10] induces Kepler-Hadoop workflows with the scheduling process in the Hadoop cluster. But the scheduling procedure is only for placement of jobs and does not consider deadline time and resource effectiveness.

De Oliveira et al. [11] has introduced a new scheduler for the workflows in cloud VM. The paper had the following parameters: total execution time, reliability and financial cost. While VM selection will lead to differential costing, but in their work we consider costing from the perceptive of VM reuse. So when we consider VM selection cost, our solution will perform ahead of this solution. When compared to other solutions, this result is closer to our expected outcomes; hence we will use it for comparison of results.

The authors [12] has got an effective strategy for the workflow scheduling. In this solution, the workflow batch size for execution is decided based on the amount of resources available. But in our case resources are available and challenge is to meet the deadline with optimum cost.

Deng et al. [13] brings about a new methodology wherein the placement of the job and the input data for the scheduling is proposed in the paper where the data and task are collocated in same host to speed up the execution. But in cloud this cannot be always ensured. Also this work does not consider deadline.

From the survey, we realize that there is a significant gap in the research work on the cloud with cost minimization and limited time execution for the workflows.

B. Problem Statement

Workflow is given as $W = \{S, T, D, d\}$, here S is the jobs set $\{J_1, J_2, J_3 \dots J_n\}$, T is the frequency of the expected transaction of each job $T = \{T_1, T_2, T_3 \dots T_n\}$, D is the dependency between the jobs given as a set of dependency relation $\{J_1 \rightarrow J_2, J_2 \rightarrow J_3 \dots\}$, and d is the deadline time for the workflow execution.

To make better and reliable system, when a job or a task in the workflow reaches the datacenter; it should be allocated to the VMs for its execution within the deadline. Also there should be optimum usage of VM, reduced bandwidth and cost of energy. Let's say solution "X" is derived through the scheduler at a cost of CX to schedule the jobs to VM, it will be beneficial if in case there is no other solution "Y" to schedule and complete the job or task within the deadline at cost of CY and also that CY is less than CX. The problem is that we do not get our expected system wherein the cost can be reduced and the performance does not get compromised. We need to get a system where there is less number of VMs for usage, minimized cost of energy & bandwidth all at the same time and also be able to complete the workflow on time.

II. SYSTEM MODEL AND METHODS

The workflow will be queued and then processed in the system in first-in-first-out manner. The resource optimized scheduling consists of three phases: job preprocessing, scheduling, and optimization.

A. Job Pre-processing

The scheduling will start only when the resources are available, the job pre-processing needs to ensure that the workflow, the VM capability, the network inter-relations and network traffic matrix are available. All these VMs have different capability, so by taking the average configuration as reference and deadline as d, the deadline and the earliest start time for all the individual tasks is computed in the workflows. The composite details of the workflows in terms of the jobs, its start time, and the deadline time are added to the queue. If there are tasks with the same earliest time, then it is saved in a link list with a time queue in the same spot.

B. Scheduling

After the job pre-processing, we identify the task or the job with the earliest start time in the setup and they are picked up from the available queue. All these individual tasks or the jobs are scheduled and sent to the VM setup according to our heuristic procedure wherein while scheduling the task, the VM which can complete the job before the deadline is to be chosen. Each of the task or job in the workflows except for the start task, accepts data from the previous task and this data transport will cause bandwidth consumption. With the network topology, a matrix with the traffic from the precedent VM is sent to the all the selected VM. Then the setup of VM is split into the group of 1 hop, 2 hop, etc., based on network topology matrix and categories the VM in each of all cluster, based on the higher amount of the data transmitted. The groups are chosen sequentially till all of the resource workflow scheduling is done fully. The increment of the energy cost on the VM is derived and those VM is dropped from selection if the cost of energy is above an upper energy threshold or below a lower energy threshold. And those VM which is within the range, it is selected for further processing in the cloud setup. With the cloud setup, efficient VM needs to be selected. And for this selection, we device a cost optimization based on heuristic. For further computation, we would design a target cost (TC) of all the activities in the time period concerned (year or month). Scheduling the workflow will incur a cost and which can be computed as

tc = VM usage time cost + energy cost of VM + bandwidth cost of data transfer.

The total cost at time t as

tct = cost of the previous task + cost of the current task at time t.

$$Y = (tc - tct) \,/\, (T - t1) \tag{1}$$

From the VM setup, "Y" is determined and with those VM with least "Y" value is processed for all the task or job in the workflow after scheduling. To make the overall operational cost less than the TC, the value of the lowest slope needs to be chosen at time period t. With our heuristic approach, we always target to choose the lowest slope, so as to ensure that ATC is less than TC.

C. Optimization

Optimization stage comes after the task or the job is executed in the VM setup. In this phase, we identify the CPU usage of the VM. If we find that there is a VM with low threshold CPU usage, we check to find out if the task is possible to move to other VM where we save cost of energy. All the tasks or the jobs are then screened from the VM setup and then migrated with heuristic scheduling algorithm inserted to analyze whether the VM can be allocated before the deadline. If the task of the workflow can be scheduled to other VM, then the concerned VM will be shutdown. With this phase, the chances of energy wastage due to keeping VM active can be minimized. The heuristic scheduling algorithm is shared below:

```
While true
  Task ← getTaskFromTimeQueue();
  VMlist ← getVMForTaskDeadline(Task);
  VMCluster1 ← get hop1 VM from VMlist;
  VMCluster2 ← get hop2 VM from VMlist;
  VMCluster3 ← get hop3 VM from VMlist;
  VMClusterFilter1←filterVMbetweenThreshold(VMCluster1,LThres,HThres);
  VMClusterFilter2←filterVMbetweenThreshold(VMCluster2,LThres,HThres);
  VMClusterFilter3←filterVMbetweenThreshold(VMCluster3,LThres,HThres);
  PVM ← [ VMClusterFilter1 VMClusterFilter2 VMCluster Filter3];
  bestVM ← PVM(1);
  Ybest = 999999999;
  For i = 1:length(PVM)
  TCt = PredictCost(PVM(i));
  Yt = (TC-TCt)/(T-t1);
  If Yt<Ybest
  Ybest = Yt;
  bestVM←PVM(i);
  End
  assisgnTask(Task,bestVM);
  End
```

III. RESULTS

Here, we derive the model for VM cost and cost of Energy & Bandwidth. Cost of VM cost can be computed as the total combination of all cost of the VM setup, migration, and running cost, as follows:

$$Vmc = VMs + VMm + VMr \tag{2}$$

where

Vmc: total VM cost,
VMs: VM setup cost,
VMm: VM migration cost, and
VMr: VM running cost.

The VM cost can be modelled in more efficient way and be formulated if the duration of scheduling "N" numbers of VM's are created and "Nm" of the VM's are migrated to each of the VM for the period of "Vmt", then the each unit cost for the VM setup will be "Ps" and each unit cost for maintenance will be "Pm", each unit cost for running will be "Pr".

$$Vmc = N{*}Ps + Nm{*}Pm + \sum_{i=1}^{N} Vmt * Pr \tag{3}$$

The cost can be minimized if there can be control on the number of VM, setup cost and other cost including maintenance. Thus, the total cost will be more or less based on the number of VM exist in the setup. The cost of energy can be computed in terms of all the energy used for the cloud setup, i.e., energy consumed for communication, storage and computation.

At each node or the switch, the sum of all the bandwidth is computed as bandwidth cost, e.g., assume that there are "N" switches and "Di" is the outgoing bytes per second in each switches, the cost of bandwidth is formulated as

$$BC = \sum_{i=1}^{N} Di * T. \tag{4}$$

The solution we propose tries to minimize the cost as much as possible.

$$Ct = Vmc + EC + BC \tag{5}$$

For the energy cost, the increment in heuristic scheduling can be achieved for the minimum value of

$$C_T = Ct + \mathrm{Min}(d(TC - Ct)/dt). \tag{6}$$

To prove the solution analytically let us consider a following scenario.

Let there be 6 tasks in a job with deadline and dependency as shown in Table 1.

Say 1 Mb data passes from one task to another. VM setup cost is 10 and VM movement cost is say 10. Run time cost is say 1 per unit time.

In our proposed scheduling, A is first allocated a VM1. After VM1 completes at deadline time 10, B, C can be

Table 1. Job with deadline and dependency

Task	Deadline	Dependency
A	10	Start
B	20	A→B
C	30	A→C
D	40	B→D
E	40	C→E
F	50	D,E→F

started. If any other scheduler were there then, it will allocate one more VM. But in the proposed, we measure the cost of allocating one more VM or use the existing VM at the same time and also meet the deadline.

Solution 1: Cost addition with existing VM = 30×1 + 20×1 = 50
Solution 2: Cost addition with 1 additional VM = 10 + 20×1 + 30×1 + 1 = 61
Solution 3: Cost addition with 2 additional VM = 10 + 10 + 20×1 + 30×1 + 1 + 1 = 72

Among all the solutions, Solution 1 is the least according to hill climbing, but it will not meet the deadline, Solution 3 will meet the deadline, but cost is high. So the stochastic hill climbing chooses Solution 2.

IV. DISCUSSION

Our proposed technique of heuristic hill climbing is executed through CloudSim environment and derived our results with the solution of different workflow of various deadline. We used the scheduling algorithm of [12] and [13] for comparison of our scheme with the solution proposed. We measured following parameters: average resource utilization (ARU), cost of VM, energy cost, bandwidth cost, and service level agreement (SLA) violation.

Once the jobs and the tasks in the workflows with different deadline reach the system, we put the above parameters to test. All jobs or the tasks are designed and automated to transfer only 1 Mb of data in the system. The ARU is calculated through the average of utilization of CPU of the all the VM in the cloud setup.

The result of ARU with different combination in the number of jobs is shown in Fig. 1.

From the above, we can conclude that the heuristic scheduling has higher average CPU utilization then the existing solution of [12] and [13].

VM cost is the total cost of having the VMs in the system. It is a total of running cost, startup cost, shifting cost etc for the VMs and the result is shown in Fig. 2.

Energy cost consumed in terms of watts by the data center is measured. The result is shown in Fig. 3.

From this graph, we conclude that the energy consumption is definitely less as compared to [12] and [13] in proposed Heuristic.

Bandwidth cost with respect to total network consumption (in kB), is measured and the result is shown in Fig. 4.

SLA violation is measured in terms of percentage of tasks missing the deadline. As such, we have put it on permutation and combination for the number of tasks. It is measured through the SLA violation and the result is shown in Fig. 5.

From the result, we understood that SLA violation of [12] and [13] is higher than our prosed scheduling technique.

The rationale of slope with the total cost is shown as Figs. 6 and 7.

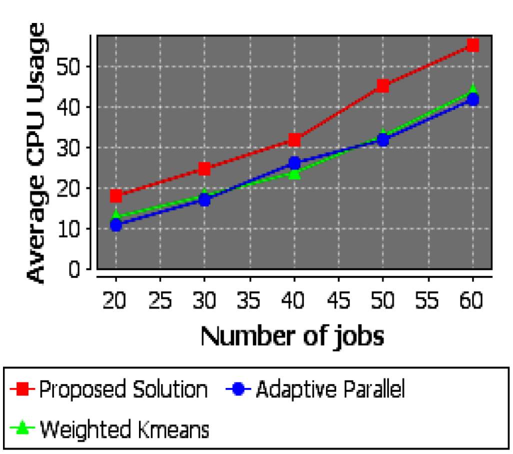

Fig. 1. Comparison of result for ARU (%).

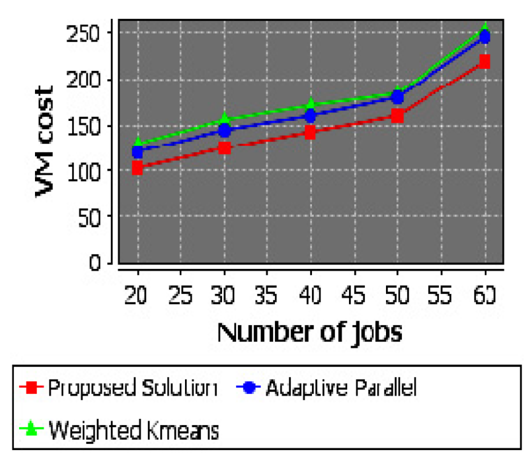

Fig. 2. Comparison of result for VM cost (the number of VM).

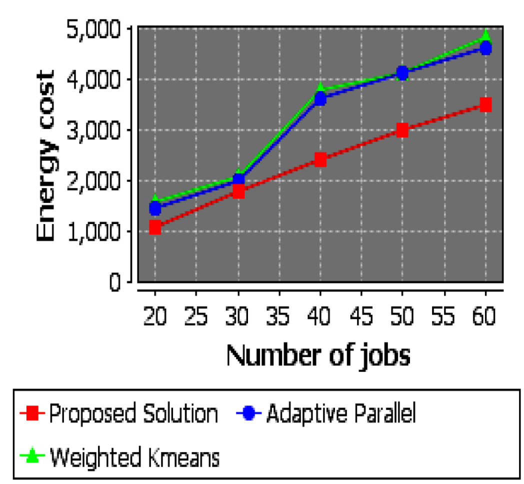

Fig. 3. Comparison of result for energy cost (Watts).

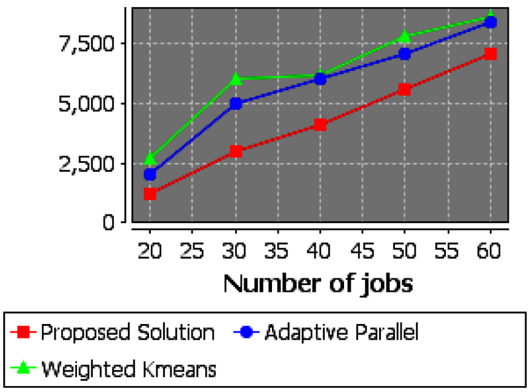

Fig. 4. Comparison of result for bandwidth cost (kB).

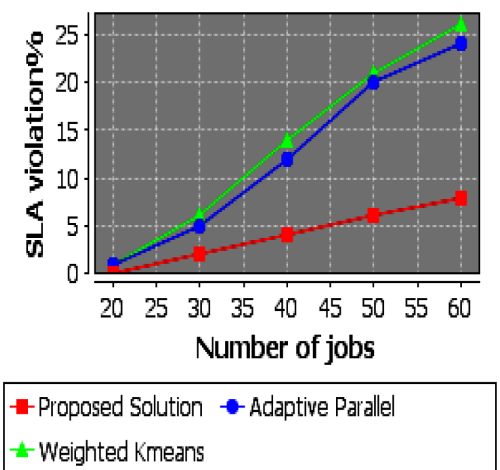

Fig. 5. Comparison of result for SLA violation.

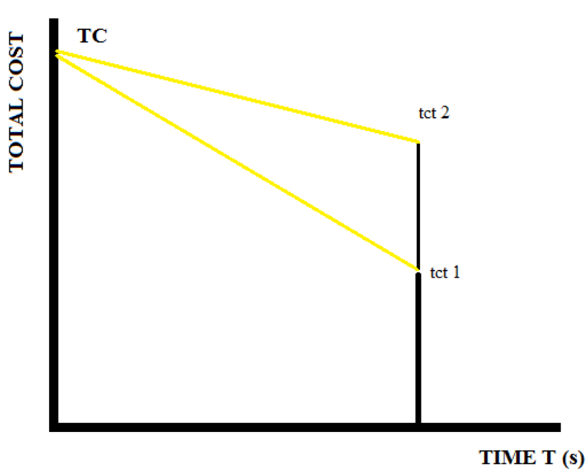

Fig. 6. Slope to total cost.

The diagrammatic representation in in MATLAB is shown in Fig. 8.

Fig. 7. Comparison between target total cost (tct) and actual total cost (ATC).

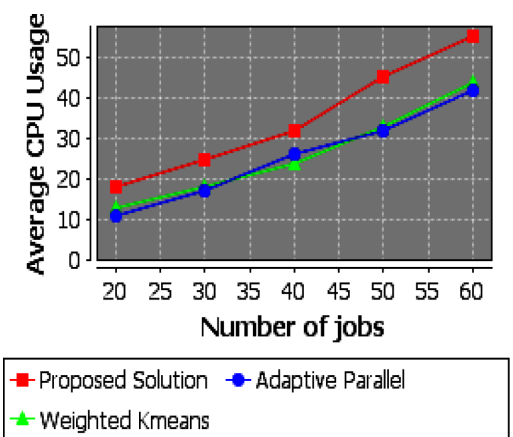

Fig. 8. MATLAB result for the proposed solution & other solutions. Task size is the number of task.

V. CONCLUSION

With our desired outcome wherein the proposed heuristic hill climbing based scheduler will be able to achieve the execution of the jobs before the given deadline, we have optimized the resources based on the factors of number of VMs used, cost of bandwidth, and energy. The jobs or the tasks in the workflows are the need of the hour. The workflow system comprises the workflow engine and a resource plug-in with various technological platforms. Workflow scheduling is a process that maps and manages the execution of interdependent tasks on the distributed resource. It allocates the suitable resources to the workflow tasks such that the execution can be achieved before deadline. By executing the workflows in time and at minimum cost, user can save cost which they would have otherwise paid for resource wastage in the datacenter.

Further, there can be a scope of future research wherein, a new workflow scheduling technique can be introduced which will be better than the proposed heuristic scheduling in terms of execution time.

REFERENCES

[1] M. Islam, A. K. Huang, M. Battisha, M. Chiang, S. Srinivasan, C. Peters, A. Neumann, and A. Abdelnur, "Oozie: towards a scalable workflow management system for Hadoop," in *Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies*, Scottsdale, AZ, 2012. DOI: 10.1145/2443416.2443420.

[2] C. Wensel, "Cascading: defining and executing complex and fault tolerant data processing workflows on a Hadoop cluster," 2008 [Internet], Available: https://github.com/cwensel/cascading.

[3] C. Olston, G. Chiou, L. Chitnis, F. Liu, Y. Han, M. Larsson, et al., "Nova: continuous Pig/Hadoop workflows," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, Athens, Greece, pp. 1081-1090, 2011. DOI: 10.1145/1989323.1989439.

[4] F. Dong and S. G. Akl, "PFAS: a resource-performance-fluctuation-aware workflow scheduling algorithm for grid computing," in *Proceedings of IEEE International Parallel and Distributed Processing Symposium*, Rome, Italy, pp. 1-9, 2007. DOI: 10.1109/IPDPS.2007.370328.

[5] J. Wang, D. Crawl, and I. Altintas, "Kepler+Hadoop: a general architecture facilitating data-intensive applications in scientific workflow systems," in *Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science*, Portland, OR, 2009. DOI: 10.1145/1645164.1645176.

[6] Z. Tang, M. Liu, A. Ammar, K. Li, and K. Li, "An optimized MapReduce workflow scheduling algorithm for heterogeneous computing," *The Journal of Supercomputing*, vol. 72, no. 6, pp. 2059-2079, 2016. DOI: 10.1007/s11227-014-1335-2.

[7] K. R. Krish, A. Anwar, and A. R. Butt, "[phi]Sched: a heterogeneity-aware Hadoop workflow scheduler," in *Proceedings of 2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)*, Paris, France, pp. 255-264, 2014. DOI: 10.1109/MASCOTS.2014.40.

[8] X. Xu, L. Cao, and X. Wang, "Adaptive task scheduling strategy based on dynamic workload adjustment for heterogeneous Hadoop clusters," *IEEE Systems Journal*, vol. 10, no. 2, pp. 471-482, 2016. DOI: 10.1109/JSYST.2014.2323112.

[9] Q. Chen, D. Zhang, M. Guo, Q. Deng, and S. Guo, "SAMR: a self-adaptive Mapreduce scheduling algorithm in heterogeneous environment," in *Proceedings of 2010 IEEE 10th International Conference on Computer and Information Technology (CIT),* Bradford, UK, pp. 2736-2743, 2010. DOI: 10.1109/CIT.2010.458.

[10] D. Crawl, J. Wang, and I. Altintas, "Provenance for Mapreduce-based data-intensive workflows," in *Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science*, Seattle, WA, pp. 21-30, 2011. DOI: 10.1145/2110497.2110501.

[11] D. de Oliveira, K. A. Ocana, F. Baiao, and M. Mattoso, "A provenance-based adaptive scheduling heuristic for parallel scientific workflows in clouds," *Journal of Grid Computing*, vol. 10, no. 3, pp. 521-552, 2012. DOI: 10.1007/s10723-012-9227-2.

[12] D. de Oliveira, E. Ogasawara, K. Ocana, F. Baiao, and M. Mattoso, "An adaptive parallel execution strategy for cloud - based scientific workflows," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1531-1550, 2012. DOI: 10.1002/cpe.1880.

[13] K. Deng, L. Kong, J. Song, K. Ren, and D. Yuan, "A weighted k-means clustering based co-scheduling strategy towards efficient execution of scientific workflows in collaborative cloud environments," in *Proceedings of 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)*, Sydney, Australia, pp. 547-554, 2011. DOI: 10.1109/DASC.2011.102.

Chintureena Thingom

received B.E. and M.Tech. degrees in Computer Science & Engineering from Dayananda Sagar Institute, Bangalore, India, in 2012 and 2014, respectively. She is currently working at CHRIST (Deemed to be University), India and also pursuing Ph.D. in Computer Science & Engineering. Her area of interest lies in the stream of Cloud Computing, Image Processing, Artificial Intelligence and Internet of Things. She have numerous research publication in various national and international arena including IEEE, Springer and other Scopus Indexed journals.

Ganesh Kumar R

received M.E. and Ph.D. degrees in Sathyabama University, India, in 2007 and 2015, respectively. He is currently working at CHRIST (Deemed to be University), India in the capacity of Associate Professor of Faculty of Engineering.

Guydeuk Yeon

Received M.Phil. and Ph.D. degrees in Jawaharlal Nehru University, India, in 1999 and 2007, respectively. He is currently working at CHRIST (Deemed to be University), India in the capacity of Director of Innovation Centre. He is also an Associate Professor of Economics.