A COST-EFFECTIVE HYBRID-SCHEDULING ALGORITHM FOR SCIENTIFIC WORKFLOW IN CLOUD COMPUTING

Ali mohammed modhafar Alhwayzee
Ferdowsi University of Mashhad, Mashhad, IRAN
ali.alhwayzee@mail.um.ac.ir
ali.alhwayzee@gmail.com

ABSTRACT: Work flow is accepted as a good construction in a sent out computing environment. Broad domain of application areas like multi-tier Web applications, upper proficiency computing application, scientific computing, great information progressing, and exegesis applications are showed utilizing workflows. These kinds of area software comprise of hybrid computational and information-focused jobs. Scheduling of those responsibilities is shaped on the ground of workflow. Forcible scheduling of Scientific Workflow Scheduling (SWFS) steps in a cloud environment leave a challenging activity when coping with great and also complicated Scientific Workflow Applications (SWFAs). Cost optimization of SWFS profits damage device consumers and producer by declining moral and financial costs in ending SWFAs. Several Cost optimization attitudes have been offered to rise the cost-effective nature of SWFS in grid and cloud computing. The basic aim of the paper is to provide a new Hybrid Cost-effective Hybrid-Scheduling (HCHS) approach, which uses a meta-heuristic algorithm such as Completion Time Driven Hyper-Heuristic (CTDHH) and with using heuristic algorithms such as the IC-PCP and IC-Loss algorithms, our algorithm achieves better results at lower time order.

Keywords: Workflow, Cloud computing, Cost optimization, Scheduling, Heuristic and Meta-Heuristic, Hyper-heuristic approach.

1. INTRODUCTION

Scheduling is one of previous challenges to stabilize workload in a hybrid cloud. Various scheduling methods can modify resource usefulness ratio, reaction period, being reliable, activity cost and remain cost [1]. Optimal scheduling issue is actually recognized to be NP-complete issue [2-3]. Generally, there is not any offered scheduling attitude which is able to perform an optimal solution within the hybrid period, especially in the term of scheduling plus amount tasks [4]. Consumers can apply various accessible computational resources to perform the tasks in an effective behavior. Nevertheless, present restricted computational resources shortage in achieving consumers' asks in order to the strange rise in complexness and scale today's applications. So, consumers require to appoint a suitable computational environment that obtain the demanded space for storage and also computational resource for going great scale complex applications.

Cloud computing resources provide a good solution that can face the consumer's needs by giving climbable and adaptable solutions for noticed applications [5]. The cloud computing on the basis of task scheduling vary from grid computing according to scheduling in the continuing of two various steps [6]-[7]:

• **Resource sharing**: cloud computing suggests jutting devices by dividing resources utilizing the virtualization think by making use of internet technology. Therefore, it supports present division to completely exert the accessible resources when rising elasticity of cloud devices. In this way, the scheduler in a cloud workflow device requires to pay attention to the virtualization association (e. g., virtual services and virtual machines) to proficiently assist in the computational progresses. Basic grid Computing purposes to allow resource dividing and harmonized dissolving issues in active, multi-institutional virtual associations.

• **Cost of resource use**: cloud computing gives an adaptable costing mechanism in order to the consumer's needs. However, basic grid computing pursues a subgroup method to set the collected cost of expected devices.

Task scheduling is to schedule tasks on the sharing resources obtaining highest benefit, effective reference usage also to face customer's QoS need. Cloud computer resource management model has gathered shared resources and dependent to each other, dependent tasks which get into workflow application design [8]. In the research, investigators have labeled task scheduling plans into two basic groups: (i) workflow-related, and (ii) job-related [9]. Job scheduling is a task of setting the system resources to the different tasks that are expected for the CPU as well as appeared in a queue. The device must resolution what especial job took first give it the CPU coming back processing, so that whole the jobs can perform in justly and good behavior [10]. In comparison, workflow scheduling is one of important issues in cloud computing which is directed at perfect execution of workflow by paying attention to their manner of system needs like deadline and budget constraints. [11]. the workflow can be explained as hybrid stages or acts that are important to bring to perfection a suggested task. The parts of these acts are able to be none of exe samples with various set ups. The workflow scheduling obtained more consideration of expert workers when compared with the job scheduling, whereas workflow-based scheduling can proficiently set best solution for great and complex applications in order to previous constraints among the hidden tasks. Instigated with this, we concentrated on baying workflow-based scheduling in cloud. Workflow-based scheduling is typically showed by utilizing a Described Acyclic Graph (DAG) design [3].

Considering performing workflow tasks in cloud, it needs tasks demonstrating to group of heterogeneous references, that are frequently utilized in cloud as a group of Virtual Machines, VM={vm_0, ..., vm_k}. In addition, it is essential to pay attention to the computational cost (in conditions of period) of performing workflow tasks on accessible heterogeneous VMs follow by connection cost among these VMs.

It is essential to automate and optimize workflow scheduling progress in considering obtaining the best Workflow System (WfMS) [3]. A perfect administration performs is required for the effective corporation of workflow applications. Workflow Management system (WMS) can be utilized for right management of workflow performance on computing resources. Pegasus is such an open-source workflow management system which contains various connected technologies that gives you for performance of workflow-based applications in heterogeneous conditions and administers the workflow going on secretly compute resources and distributed data [12]. Architecture of Pegasus WMS is displayed in Fig. 1.

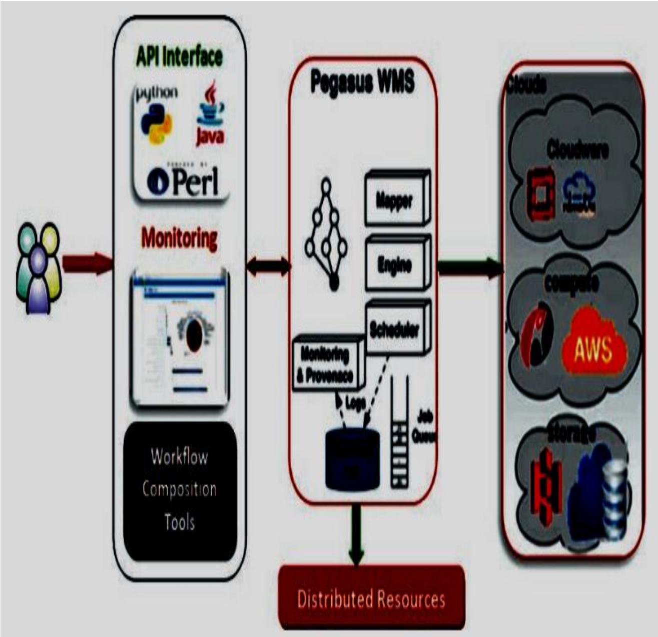

Fig. 1 Pegasus workflow Management System architecture [12]

On the other hand, Scientific Workflow Application (SWFA) often implies information flows with each other with the tasks execution [3], consisting of input scripts, which may be used to product, assay and incarnate output results. This provides impact on each other equipment to aid scientists' better do their own workflows and see conclusions in real period. Furthermore, the SWFA abridges the progress for scientists to reutilize exactly the same workflows and bring about them by a simple-to-use environment to follow and division output conclusions nearly. One of most challenging issues with SWFS in cloud is to optimize cost of workflow execution [13].

Since in cloud computing there are two basic actors participated, there are two sides of cost optimization: cost optimization executed by producers and cost optimization executed by consumers. Cost optimization executed by cloud producers basically concentrates on reducing the cost to hold a physical information center. The center reduction is usually obtained by lowering electricity use [14]. The center optimization obstacle of SWFS in cloud computing is a multi-purpose center–letting know issue that needs a thought of three basic natures [3]: (i) various consumers that commonly remain competitive for resources within the cloud or grid computing to fulfill QoS difficulties, (ii) the relationship between workflow tasks, and (iii) high connection cost as a result of relationship among tasks. Nevertheless, paying attention to all cost optimization issues in order to nature the real SWFS progress more elaborated as well as needs an upper number of computational resources in conditions of computational period. This paper presents a new method using three algorithms simultaneously (a meta-heuristic CTDHH and two heuristic IC-PCP and IC-Loss).

This paper is organized as pursued: In Section two, provides the related works. Section three explains the Completion Time Driven Hyper-Heuristic (CTDHH) attitude. Next, Section four explains the Improvement Completion Time Driven Hyper-Heuristic (ICTDHH) attitude Section five debates the evaluation section with utilizing a real-world cloud related experimentation environment. Then, in Section six, the results and arguments of this study paper are explained and lastly, Section seven gives a conclusion.

2. REVIEW OF PREVIOUS WORKS

A lot of investigators have been done in Grid environment for the condition of workflow applications scheduling [15]. The issue of workflow scheduling has been an open up domain of inquiry in order to the NP completeness. A lot of methods have been accepted to resolve the same, generic from heuristic to meta-heuristic [16]. Heuristic related consults with assist in locating an acceptable solution immediately. On another part meta-heuristic primarily based attitudes discover solution storage better and hold on filtrate their conclusions, but concurrently expend a rational number of period in progress. We considered some of the existing workflow scheduling algorithms in order to our suggested act as pursued.

Su et al. [17], suggested an economical task-scheduling algorithm by utilizing two heuristic principles. The first principle locomotive plans tasks to the most economical VMs in order to the principle of Pareto influence. The second principle, a match to the first principle, declines the monetary costs of non-critical tasks. They will perform comprehensive statistical temptations on great DAGs produced at random as well as on actual applications. The simulation conclusions represent that their algorithm can basically decline financial costs when generating make span as effective as the most popular task scheduling algorithm able to obtain.

Verma *et al.* [18], suggested Budget and Deadline Constrained Heuristic based over Heterogeneous Earliest Finish Time (HEFT) [19] to schedule workflow tasks along the accessible cloud references. The suggested heuristic shows a profitable trade-off among performance time and performance cost below given limitations.

Pandey *et al.* [20], suggested a Particle Swarm optimization (PSO) related heuristic to schedule applications to cloud resources that consider both two information transfer cost and calculation cost. Their conclusion represents that PSO able to obtain: a) just as much as 3 times cost storing in contrast to the Best Resource Selection (BRS), and also b) nice division of workload onto resources.

Coutinho et al. [21], suggested a mathematical design (CC-IP-fed) and a heuristic-related attitude (GraspCC-fed) for scheduling of equal scientific workflows in united clouds. All their mathematical design reduces both make span and cost aims by using a gathering procedure in which a measure is referred to each purpose by consumer and completes client-explained as well as budget restrictions. CCIP-fed design would not pay attention to the cost of information transfer.

Arabnejad et al. [22], suggested a heuristic scheduling algorithm with quadratic time contortion that earmarked two serious difficulties for QoS-based workflow scheduling, time and cost, known as Deadline-Budget Constrained Scheduling (DBCS). From the deadline and budget explained by the consumer, the DBCS algorithm figure out a possible answer that perform both two limitations with a success ratio the same as the other state-of-the-art search-related algorithms in conditions of the successful rate of possible answers, using at worst only nearly 4% of times.

Arabnejad *et al*. [23], suggested new algorithms, Proportional Deadline Constrained (PDC) and Deadline Constrained Critical Path (DCCP). PDC acts with increasing the parallelism in a workflow by demarcating it into reasonable classes and then comparatively sub allocating the whole workflow deadline along them. The DCCP formula is alike PDC, the key difference is that it as well appoints limited criticism way during workflow considering scheduling jobs which connect to same instance. Found in conditions of cost execution, all of PDC and DCCP algorithms regressed lowest calculate cost, whole workflows and illustration formations.

Verma *et al*. [24], offered the all-purpose Hybrid Particle Swarm Optimization (HPSO) algorithm related after no mastery scheduling way to unravel the cloud workflow scheduling issue. It is a blend of all-purpose particle swarm optimization algorithm and index related heuristic. Its execution is assayed by utilizing 3 unsociability aims of make span, energy usage and universal cost below budget and deadline limitations.

Abrishami *et al*. [4] expanded an attitude named IC-PCP. Within their act, the scheduling is founded on locating critical way according to every egress issue. The tasks like a way are greatly scheduled on same VM or on cheapest accessible illustration. Then, a critical way for each and every unassigned job is computed and same progress is followed till every task has been transferred to some of the VM.

Kaur *et al.* [25], offered an Augmented Shuffled Frog Leaping Algorithm (ASFLA) structured method for resource scheduling and workflow scheduling in Infrastructure as a Service (IaaS) cloud environment. Results assay of ASFLA presents that previous outperforms other methods in declining whole performance cost of regarded workflows.

Alkhanak *et al*. [26], offered a Completion Time Driven Hyper-Heuristic (CTDHH) attitude for cost optimization of SWFS in a cloud environment. The CTDHH attitude uses four famous population-related meta-heuristic algorithms that work as Low-level Heuristic (LLH) methods. Furthermore, the CTDHH procedure improves the vernacular randomly chosen path of existing hyper-heuristic attitudes by including the optimal calculated work flow completion time to become an upper-class chooser to actively select an appropriate algorithm from the set of LLH algorithms after every row. An actual cloud related temptation environment has been regarded to assess the execution of CTDHH attitude with evaluating it by five primary attitudes, i. e. four population-related attitudes and an existing hyper-heuristic attitude known as Hyper-Heuristic Scheduling Algorithm (HHSA). Some various scenarios have also been reflected to assess information condense and calculation-condensed execution. CTDHH strategy has obtained the best conclusions for almost all of the SWFA datasets as well as for almost all of the regarded situations compared with the primary and HHSA attitudes.

3.THE PROPOSED ALGORITHM

The idea presented in [26] is to use some of the meta-heuristic algorithms and has chosen one of these algorithms for execution in each phase in each phase. One of the problems in [26] is the high order times of the algorithm due to the lack of attention to the initial population of the algorithms. To solve this problem, the proposed method is to use the heuristic algorithms to improve the initial population of these algorithms in order to achieve better results at a shorter order times. For this purpose, the ic-pcp and ic-loss algorithms have been used to create initial population. The general algorithm of proposed method is as follows:

Algorithm 1: Cost-effective Hybrid-Scheduling (HCHS) Algorithm

Input: W=(T,E), T=$\bigcup_{i=1}^{n} T$i , E={(Ti ,Tj , Dataij)|(Ti,Tj)∈ $T \times T$}, Eij=(Ti,Tj,dataij), H(the set of the Heuristic algorithms: PSO and GA, d is the user deadline and c is user budget for w.

Output: The most optimal solution for cost optimization of w

```
α ← TimeOfFS (w)   // α is the time of Fastest Schedule of w
A ← ∅ , B ← ∅      // A is initial population with IC-PCP and B is initial population with IC-Loss
for i←1 to 5 do    // compute the algorithms for different deadline factors
    add compute IC-PCP(w with deadline d) to A
    add compute IC-Loss(w with budget c) to B
    α← α*i
end for
for 1 to 5 do
        Run h with initial population A,B , ∀h ∈ H
end for
compute DHHA (w)
```

The proposed algorithm needs two basic inputs for its proper acting. The inputs are: (i) W= (T, E), where T (vertex) is a collection of tasks and E (edges) is a collection of guided edges among the tasks and (ii) H (the collection of the Heuristic algorithms: PSO and GA). As presented in Algorithm 1, the first stage of first process of suggested HCHS is operating both of the two applied meta-heuristic algorithms (i. e. particle swarm optimization, genetic algorithm). In the other hand, the employed Heuristic algorithms goes each (h) of applied Heuristic methods (H) to schedule presented workflow tasks structured on the accessible VMs for five times for every single especial scenario, where (h)

(PSO), Genetic Algorithm (GA), and Completion Time Driven Hyper-Heuristic (CTDHH)). To compare and evaluate the proposed algorithm have been tested shows LLH algorithm that is a section of collection of (H). At first initial population is created using two algorithms the IC-PCP (with user deadline constraint) and the IC-LOSS algorithm (with user budget constraint). With creating the initial population by these algorithms execution time is reduced in the proposed algorithm.

3-1- An initial population with the IC-PCP Algorithm

IC-PCP is called as IaaS Cloud Partial Critical Paths and it was presents during the correction ICPCPD2 by Abrishami *et. al*. in [4]. The basic optimization purpose of algorithm is whole of the computing cost when creating workflow to complete within explained deadline. IC-PCP begins making schedule from the dummy node in end of workflow. The progress contains both basic sections - parents allocating and way allocating. On the parents allocating stage for every single task observe a critical path that includes critical parent's unsuitable mother or father tasks having the most recent information achieving time. After critical path putting at the first, algorithm allocates all of its tasks to cheapest resources pleasing tasks' recent complete periods. Later it attempts to make next critical path for task Provided that the IC-PCP formula is an algorithm that presents the best scheduling based on the deadline given by the consumer, this algorithm has recently been used to make the initial population.

3-2- An initial population with the IC-LOSS Algorithm

IC-Loss is a corrected version of Loss algorithm [27] for cloud environment. Loss algorithm has recently been suggested for scheduling workflow in grid environment and in re-scheduling class it holds exchanging tasks among the resources with selecting tasks with the most compact loss weight. Whereas in IaaS clouds re-scheduling single task on a cheaper machine may improve whole execution cost, so re-scheduling process of the Loss algorithm ought to be accepted. IC-Loss attempts to reschedule all tasks of an instance to a cheaper existing or new instance, regarding to minimize whole execution cost. Considering that the IC-LOSS algorithm is an algorithm that presents the best scheduling with regard to the budget given by the consumer, this algorithm has recently been used to make the first population.

3-3- Completion time driven hyper-heuristic approach

CTDHH [26] has recently been shown for cost optimization challenge of SWFS in a cloud environment. The algorithm is regarded as a new jutting method which can increase the runtime of a meta-heuristic algorithm. CTDHH uses the High Level Heuristic (HLH) method by making use of four famous population-based Meta heuristic algorithms that become the Low Level Heuristic (LLH). The basic aim of HLH method is to brilliantly silently move the search progress structured on the execution of the applied Meta heuristic LLH algorithms. The execution of CTDHH attitude has recently been widely assessed by evaluating it with four population-based attitudes (i. e. GA, PSO, IWO, HIWO) and an existing hyper-heuristic attitude known as Hyper-Heuristic Scheduling Algorithm (HHSA). Depending on the lowest obtained completion time, the attitude meaningfully leads the beating procedures to locate an ideal answer with continually selecting the computed time results (i. e. completion times of former runs) of all of the LLH algorithms for every single regarded scenario along with each run. Therefore, the mechanism of the completion time driven hyper heuristic becomes more useful in a path letting applying and reusing most ideal durability of applied LLH methods in finding for best answer of purposed cost optimization issue. Of whole computational cost conclusions, method has obtained cheapest whole computational cost discussion with baseline methods. These conclusions are still impacted by SWFA's kind and size. This type of is basically due to sophisticated and great size of the presented SWFA work flow tasks, that finally create SWFS attitudes to take longer time to perform these tasks. This type of is regard to the truth of tasks of Montage SWFA have less previous limitations. So, the CTDHH attitude has obtained best conclusions for almost all of SWFA datasets as well as for almost all of regarded scenarios assimilated with baseline and HHSA approach. In our proposed method, we used this algorithm by creating an appropriate initial population.

4- Evaluation of results

For the real-world improvement environment, suggested attitude Hybrid Cost-effective Hybrid-Scheduling (HCHS) strategy has been assimilated with three baseline meta-heuristic attitudes (i.e. Particle Swarm Optimization five workflows Inspiral, Montage, Epigenimics, Cybershake and SIPHT with the available classes in Table 1.

Table 1 - Workflow classes

Workflow	large	small
Montage	1000	25-50
Inspiral	1000	30-50
Epigenimics	997	24-46
Cybershake	1000	30-100
SIPHT	1000	30-50

The first two pso and genetic algorithms, that their initial population has been initialized using ic-loss and ic-pcp, is executed five times (according to the algorithm presented in [26]) and the runtime value of each of them are stored in the TS variable, and then the average of these five run times for each algorithm is calculated separately. Then the algorithm with the lowest average value is selected and executed. Then the new execution time replaces the previous execution time, and so on. As long as we reach the end (Thirty times the execution of this method). At the end, the final execution time is considered as execution time of the workflow.

Table 2 illustrates the statistical results of completion time for Inspiral, Epigenomics Montage, CyberShake, and Sipht workflows in real-world environment.

Table 2 Statistical results of completion time for workflows

#	Workflows	time (GA)	time (PSO)	time (CTDHH)	time (HCHS)
1	inspiral30	1.8	2.21	1.45	1.32
2	Inspiral 50	2.2	2.5	1.9	1.65
3	Inspiral 1000	2.43	2.78	2.12	1.8
4	Epigenomics 24	2.85	2.59	2.44	2.31
5	Epigenomics 46	2.69	2.55	2.8	2.25
6	Epigenomics 997	2.41	2.68	2.74	2.18
7	Montage 25	2.77	4.5	3.1	2.54
8	Montage 50	3.24	3.94	3.37	2.69
9	Montage 1000	3.11	3.48	2.91	2.51
10	CyberShake30	4.8	5.4	5.6	3.41
11	CyberShake100	5.2	4.3	4.87	2.89
12	CyberShake1000	3.8	3.2	3.7	2.21
13	Sipht30	1.01	1.95	1.21	0.89
14	Sipht100	1.18	1.54	1.14	1.08
15	Sipht1000	1.32	1.27	1.92	1.22

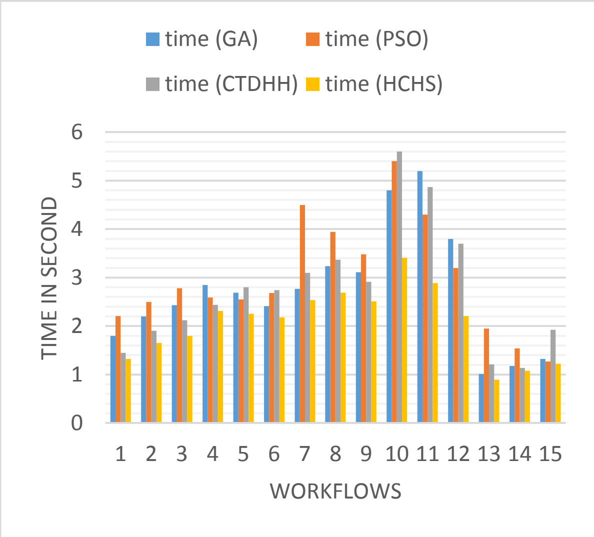

Figure 2- Comparison of proposed algorithm with other algorithms

The average completion time worth of suggested HCHS approach is better than average completion time of CTDHH attitude for all of the workflows. This is in order to utilized heuristic algorithms (IC-LOSS, IC-PCP) in creating initial population of algorithms (PSO, GA) considering to find most optimal solution for cost optimization of workflow. As you can see in Table 2, the results of offered algorithm are better than other algorithms, and time order is reduced for workflow.

5- CONCLUSION

Cloud computing technology, a new design of device providing in allocated devices, has been increased on the ground of a procedure to perform workflow applications. To take advantages of this technology for performing workflow applications, it is essential to expand workflow scheduling algorithms which regard various QoS details such as execution time and. as well as cost. We showed a Hybrid Cost-effective Hybrid-Scheduling (HCHS) approach for companies to rent VM illustrations from cloud producers to progress multiple workloads, when interacting with the needed response period for services which have impact on each other and the deadlines of batch tasks. By comprehensive statistical simulations we presented that our proportioned scheduling considerably outperforms existing attitudes that it is to increase the economical nature of SWFS in grid and cloud scheduling. The main goal of the paper is to present a new attitude, which uses a meta-heuristic algorithm such as Completion Time Driven Hyper-Heuristic (CTDHH) and with using two heuristic algorithms such as the IC-PCP and IC-Loss algorithms, our algorithm achieves better results at lower time order.

REFERENCE

[1]. Wang, Wei-Jen, Yue-Shan Chang, Win-Tsung Lo, and Yi-Kang Lee. "Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments." *The Journal of Supercomputing* 66, no. 2 (2013): 783-811.

[2]. Tyagi, Rinki, and Santosh Kumar Gupta. "A Survey on Scheduling Algorithms for Parallel and Distributed Systems." In *Silicon Photonics & High Performance Computing*, pp. 51-64. Springer, Singapore, 2018.

[3]. Alkhanak, Ehab Nabiel, Sai Peck Lee, Reza Rezaei, and Reza Meimandi Parizi. "Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues." *Journal of Systems and Software* 113 (2016): 1-26.

[4]. Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema. "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds." *Future Generation Computer Systems* 29, no. 1 (2013): 158-169.

[5]. Samah Alnajdi, Maram Dogan, and Ebtesam Al-Qahtani. “A survey on resource allocatio9n in cloud computing.” International Journal on Cloud Computing: Services and Architecture (IJCCSA) Vol. 6, No. 5, October 2016.

[6]. Hluchý, L., V. Tran, and M. Dobrucký. "Grid computing and EGEE project." All content following this page was uploaded by Miroslav Dobrucký on 13 July 2016.

[7]. Foster, Ian, Yong Zhao, Ioan Raicu, and Shiyong Lu. "Cloud computing and grid computing 360-degree compared." In *Grid Computing Environments Workshop, 2008. GCE'08*, pp. 1-10. Ieee, 2008.

[8]. Suri, P. K., and Sunita Rani. "Design of Task Scheduling Model for Cloud Applications in Multi Cloud Environment." In *International Conference on Information, Communication and Computing Technology*, pp. 11-24. Springer, Singapore, 2017.

[9]. Czarnul, Pawel. "Modeling, run-time optimization and execution of distributed workflow applications in the JEE-based BeesyCluster environment." *The Journal of Supercomputing* 63, no. 1 (2013): 46-71.

[10]. Mandeep Kaur, Sugandha Sharma, Rajinder Kaur. “Optimization of Job Scheduling in Cloud Computing Environment.” International Journal of Advanced Research in Computer Science and Software Engineering, Vol 4, Issue 7, July 2014.

[11]. Masdari, Mohammad, Sima ValiKardan, Zahra Shahi, and Sonay Imani Azar. "Towards workflow scheduling in cloud computing: a comprehensive analysis." *Journal of Network and Computer Applications* 66 (2016): 64-82.

[12]. Kousalya, G., P. Balakrishnan, and C. Pethuru Raj. "Execution of Workflow Scheduling in Cloud Middleware." In *Automated Workflow Scheduling in Self-Adaptive Clouds*, pp. 103-118. Springer, Cham, 2017.

[13]. Li, Jian, Sen Su, Xiang Cheng, Meina Song, Liyu Ma, and Jie Wang. "Cost-efficient coordinated scheduling for leasing cloud resources on hybrid workloads." *Parallel Computing* 44 (2015): 1-17.

[14]. Nodari, Andrea. "Cost optimization in cloud computing." (2015).

[15]. Rodriguez, Maria Alejandra, and Rajkumar Buyya. "A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments." *Concurrency and Computation: Practice and Experience* 29, no. 8 (2017).

[16]. Singh, Vishakha, Indrajeet Gupta, and Prasanta K. Jana. "A novel cost-efficient approach for deadline-constrained workflow scheduling by dynamic provisioning of resources." *Future Generation Computer Systems* 79 (2018): 95-110.

[17]. Su, Sen, Jian Li, Qingjia Huang, Xiao Huang, Kai Shuang, and Jie Wang. "Cost-efficient task scheduling for executing large programs in the cloud." *Parallel Computing* 39, no. 4-5 (2013): 177-188.

[18]. Verma, Amandeep, and Sakshi Kaushal. "Cost-time efficient scheduling plan for executing workflows in the cloud." *Journal of grid computing* 13, no. 4 (2015): 495-506.

[19]. Topcuoglu, Haluk, Salim Hariri, and Min-you Wu. "Performance-effective and low-complexity task scheduling for heterogeneous computing." *IEEE transactions on parallel and distributed systems* 13, no. 3 (2002): 260-274.

[20]. Pandey, Suraj, Linlin Wu, Siddeswara Mayura Guru, and Rajkumar Buyya. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." In *Advanced information networking and applications (AINA), 2010 24th IEEE international conference on*, pp. 400-407. IEEE, 2010.

[21]. Coutinho, Rafaelli de C., Lúcia MA Drummond, Yuri Frota, and Daniel de Oliveira. "Optimizing virtual machine allocation for parallel scientific workflows in federated clouds." *Future Generation Computer Systems* 46 (2015): 51-68.

[22]. Arabnejad, Hamid, Jorge G. Barbosa, and Radu Prodan. "Low-time complexity budget–deadline constrained workflow scheduling on heterogeneous resources." *Future Generation Computer Systems* 55 (2016): 29-40.

[23]. Arabnejad, Vahid, Kris Bubendorfer, and Bryan Ng. "Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources." *Future Generation Computer Systems* 75 (2017): 348-364.

[24]. Verma, Amandeep, and Sakshi Kaushal. "A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling." *Parallel Computing* 62 (2017): 1-19.

[25]. Kaur, Parmeet, and Shikha Mehta. "Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm." *Journal of Parallel and Distributed Computing* 101 (2017): 41-50.

[26]. Alkhanak, Ehab Nabiel, and Sai Peck Lee. "A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing." Future Generation Computer Systems(2018).

[27]. Sakellariou, Rizos, Henan Zhao, Eleni Tsiakkouri, and Marios D. Dikaiakos. "Scheduling workflows with budget constraints." In *Integrated research in GRID computing*, pp. 189-202. Springer, Boston, MA, 2007.