ISSN: 0976-3104
SPECIAL ISSUE: Emerging trends in Computer Engineering and Research (ECER)

ARTICLE

ENHANCED PERFORMANCE IN WORK FLOW SCHEDULING USING GTTD

V. Balaji[*], **P. Swarnalatha**

Department of SCOPE, VIT University, Vellore, Tamilnadu, INDIA

ABSTRACT

Researchers in each research team share their information and procedure distributed resources for directing their experimentations. These tests are actuality complemented in association with groups that are internationally distributed. Information area difficult and transmitted information overhead are main challenges for development such Information –demanding technical workflow application in cloud computing. These requests are important to the period of big data and task execution involves incontrollable and manufacturing huge amount of input/output information with data dependencies amongst responsibilities. Grouping Technique based Task Dependency | (GTTD) to decrease performance overhead and to increase the computational granularity of technical workflow tasks is obtainable in this paper. And this paper suggests the Information-intensive workflow development system to reduce make distance of the Information-intensive workflow applications, which can be exhibited as a focussed acyclic graph. Grouping technique is authenticated by using imitation based exploration though Work flow Sim.

KEY WORDS
application software,scheduling, grouping, cloud

Received: 29 May 2017
Accepted: 27 July 2017
Published: 22 Sept 2017

***Corresponding Author**
Email:
vuppala.balaji@gmail.com
Tel.: +91-8688778087

INTRODUCTION

A workflow could be a high-level specification of a collection of tasks that represent procedure science or business flows and therefore the dependencies between the tasks that has got to be glad so as to accomplish a particular goal. The business progress is sometimes management flow-driven and includes constructs to specify methods, conditions, and should embrace human interaction. Typically, a business workflow implements a company's product or service. Scientific workflows usually touch upon massive an outsized quantity of knowledge and advanced calculations and so utilize large storage capacities and computing resources. During a scientific progress, a task is associate degree workable program with a collection of input parameters and files. Scientific workflows are generally information flow-driven and don't have made management flow structures, whereas notable exceptions exist, like Askalon [1]

In several analysis fields, particularly in lay to rest disciplinary field like bioinformatics and climate simulation, scientific workflows are typically each computation and data-intensive. Workflow running typically wants large-scale computing resources however additionally large storage. Today, scientific workflows applications carries with it many thousands of tasks, consume gigabytes or terabytes input information sets and generate similar amounts of intermediate data. These applications are noted as data-intensive progress applications. A knowledge intensive application consists of applications that manufacture, manipulate, or analyze information within the vary of many megabytes (MB) to petabytes (PB) [2]. Having the power [3] to makeover workflows between execution resources needs a precise degree of flexibility once it involves information placement and information movement. Most workflows need a knowledge storage resource near the execution web site so as to execute in associate degree economical manner. The benefits of cost-effectiveness, on-demand resource provision and straight forward for sharing.

A workflow could be a high-level specification of a collection of tasks that represent procedure science or business flows and therefore the dependencies between the tasks that has got to be glad so as to accomplish a particular goal. The business progress is sometimes management flow-driven and includes constructs to specify methods, conditions, and should embrace human interaction. Typically, a business workflow implements a company's product or service. Scientific workflows usually touch upon massive an outsized quantity of knowledge and advanced calculations and so utilize large storage capacities and computing resources. During a scientific progress, a task is associate degree workable program with a collection of input parameters and files. Scientific workflows are generally information flow-driven and don't have made management flow structures, whereas notable exceptions exist, like Askalon [1].

In several analysis fields, particularly in lay to rest disciplinary field like bioinformatics and climate simulation, scientific workflows are typically each computation and data-intensive. Workflow running typically wants large-scale computing resources however additionally large storage. Today, scientific workflows applications carries with it many thousands of tasks, consume gigabytes or terabytes input information sets and generate similar amounts of intermediate data. These applications are noted as data-intensive progress applications. a knowledge intensive application consists of applications that manufacture, manipulate, or analyze information within the vary of many megabytes (MB) to petabytes (PB) [2]. Having the power [3] to makeover workflows between execution resources needs a precise degree of flexibility once it involves information placement and information movement. Most workflows need a knowledge storage resource near the execution web site so as to execute in associate degree economical manner. The benefits of cost-effectiveness, on-demand resource provision and straightforward for sharing Distributed computing has developed in prominence with research group for conveying work flows. At the point when work flows are broadly performed in cloud situations that comprise of various data centres, there is a critical requirement for creating systems which can put the application information

COMPUTER SCIENCE: Guest editor-Prof. S. Prabu & Prof. Swarnalatha P

crosswise over all-inclusive circulated data centres and plan assignments as per the information design to lessen both the dormancy and make span for work flow execution. Make span is a term alluding to the execution metric of work flow and characterized as the interim between the begin time of first errand and the end time of definite assignment [4].

Regularly, an information serious application work flow is booked to limit add up to information exchange time as well as cost, storage room utilized, add up to execution time or potentially a blend of these. In such conditions, information area and exchanged information overhead are essential difficulties for information escalated applications to decrease execution time and size of exchanged information. Overlooking area of information supports high data transmission utilization cost [5]. For a few applications [6] up to 90 % of the execution time is spent on document exchange. Numerous scientists [2] have proposed a few components for exchanging information with the goal that information exchange time is limited. These strategies are: information parallelism, information gushing, and information throttling. Information throttling is a procedure of depicting and controlling when and at what information rate is to be moved rather than moving information starting with one area then onto the next as right on time as could be allowed. Planning and execution overhead are high when low execution of fine-grained assignments is a typical issue in broadly disseminated stages. This paper proposes errand grouping strategy to limit these overhead in light of assignment reliance.

Information development and exchange overhead are not enormous issues in a little group condition. Be that as it may, logical work flow framework is intended for researchers to participate over a few server farms. In runtime of execution, information development and exchanged time can effect on aggregate execution of work flow application. "Moving information to a server farm will cost more than planning assignments to that inside [10]". The aggregate execution time of work flows is frequently influenced by different latencies, for example, the asset disclosure, booking and information get to latencies for the individual work flow forms. To build information region and to lessen above latencies, Meta Data Service (MDS) is executed to store sets of datasets and their area on the proposed framework with the assistance of throttling system. The decision of capacity design [7] likewise significantly affects work flow execution time and the cost of nearly takes after execution, and so forth. This framework utilizes nearby reserve for information stockpiling.

Whatever remains of this paper is sorted out as takes after. Area 2 gives a review of the related work. Area 3 displays the proposed framework and its segments. Segment 4 reports the exploratory outcomes. Area 5 closes with a conclusion.

RELATED WORK

A workflow could be a high-level specification of a collection of jobs that represent procedure science or business flows and therefore the dependencies between the jobs that has got to be glad so as to accomplish a particular goal. The business progress is sometimes management flow-driven and includes constructs to specify methods, conditions, and should embrace human interaction. Typically, a business workflow implements a company's product or service. Scientific workflows usually touch upon massive an outsized quantity of knowledge and advanced calculations and so utilize large storage capacities and computing resources. During a scientific progress, a task is associate degree workable program with a collection of input parameters and files. Scientific workflows are generally information flow-driven and don't have made management flow structures, whereas notable exceptions exist, like Askalon [1].

In several analysis fields, particularly in lay to rest disciplinary field like bioinformatics and climate simulation, scientific workflows are typically each computation and data-intensive. Workflow running typically wants large-scale computing resources however additionally large storage. Today, scientific workflows applications carries with it many thousands of jobs, consume gigabytes or terabytes input information sets and generate similar amounts of intermediate data. These applications are noted as data-intensive progress applications. A knowledge intensive application consists of applications that manufacture, manipulate, or analyze information within the vary of many megabytes (MB) to petabytes (PB) [2]. Having the power [3] to makeover workflows between execution resources needs a precise degree of flexibility once it involves information placement and information movement. Most workflows need a knowledge storage resource near the execution web site so as to execute in associate degree economical manner. The benefits of cost-effectiveness, on-demand resource provision and straightforward for sharing.

Many models have been proposed and actualized to successfully execute the work process job. Different heuristic and meta heuristic techniques have been explored streamlining single or composite parameters for work process booking. An auto scaling technique to enhance the cost while meeting the due dates is proposed by Mao et.al.[6] proposed a Deadline Early Tree heuristic calculation with due date as limitation which limits cost. Rizos et al. [7] proposed a best moderate task approach LOOS and GAIN for improving the cost and execution time. The underlying assignments are made utilizing heuristic calculation which are reassigned to meet the financial plan and time requirements. Yu et.al [8] proposed a Markov Choice Process based way to deal with limit the cost of executing application on matrix supporting the client characterized due date. Zheng et al.[9] stretched out the HEFT calculation to Financial plan obliged Heterogeneous Earliest Finish Time (HEFT). The calculation considers the spending imperative and streamlines the execution time. Durillo et.al [10] examined multi objective HEFTI (MOHEFT) based rundown booking heuristic which streamlines makespan and cost of executing work processes in Amazon EC2.

Over the prior time, the planning of information concentrated work flows is emerged as a critical research theme in conveyed figuring. Work flow planning for cloud is not the same as that in multiprocessors or lattice framework. This area concentrates on works managing information territory and information exchanges or information development in cloud condition. PiotrBryk et al. [6] proposed a novel element planning calculation that knows about the fundamental stockpiling framework that can be utilized on IaaS mists by considering information exchanges. This calculation upgrades plans by exploiting information reserving and document territory to lessen the quantity of record exchanges amid execution. Ke Wang et al. [8] proposed calculation for work taking (load adjusting) to decrease information area and information exchanging overhead to run information serious logical applications in Many-undertaking figuring

(MTC). Claudia Szabo et al. [9] proposed structure which incorporate portion and requesting chromosome for distribution of undertakings to hubs and execution arrange as per the logical work flow portrayal by considering information exchange and execution time utilizing hybrid and change administrators with the standard NSGA-II calculation. D.Yuan et al. [10] proposed k-implies bunching procedure for information position of logical Cloud work flows to diminish information development. Mingjun Wang et al. [11] additionally utilized k-implies bunching calculation to disperse input informational indexes into various server farms with the most related datasets set together. In this paper creators likewise proposed a multilevel assignment replication planning system to diminish enormous informational collections move in the runtime of the logical work flows.

Optimization of performance metrics such as quantity and potential in mixed computing situation develop sex trastimulating outstanding to the variance in the computing capability of accomplishment nodes and differences in the information transmission competency of communication associations among these nodes.

SaimaGulzar Ahmad et al. [12] obtainable a dual impartial Dividing based Data-intensive Workflow optimization Algorithm (DDWA) for heterogeneous calculating classifications. In this algorithm, the application job graph is divided so that the inter-partition information movement is minimal. PDWA provides significantly reduced latency with increase in the quantity.

THE PROPOSED SYSTEM

This section discusses the planned data-intensive workflow programming system. This method has 2 phases. The first section, tasks in workflow application square measure clustered so as to cut back execution overhead. the second section, these clustered workflows square measure allotted onto cloud resources like virtual machine (VM) supported the info neighbourhood as this planned system concentrate on data-aware programming. Within the data-intensive scientific workflows, tasks would like quite one dataset to execute. However, once these tasks square measure dead in several data centre, information transfer would become inevitable. To deal with these issues, this paper proposes clump methodology supported Task Dependency to optimize progress programming. The planned system conjointly uses Meta information Service (MDS) and information suffocation technique. By victimisation these techniques, the quantity of entomb cloud transfers and consequently the info traffic prices square measure heavily reduced and conjointly makes pan. The planned system for data-intensive progress programming is shown in [Fig. 2] and therefore the followings square measure parts of this planned system.

Application model

Technical workflows are showed as Directed Acyclic Graph (DAG). A DAG, G (V, E), contains of a set of vertices V, and edges, E. The edges characterise preference constrictions: each edge $e_{a,b}$ = (t_a,$t_{b)}$∈Erepresents a preference constriction that directs that jobt$_a$ should complete its accomplishment before task t_b $_b$starts. $e_{a,b}$also characterizes the volume of inter-task communication elaborate, e.g., the quantity of information(in bytes)that jobt$_a$must direct to jobt$_b$in order for jobt$_b$to start its execution as shown in.

SPECIAL ISSUE

ISSN: 0976-3104

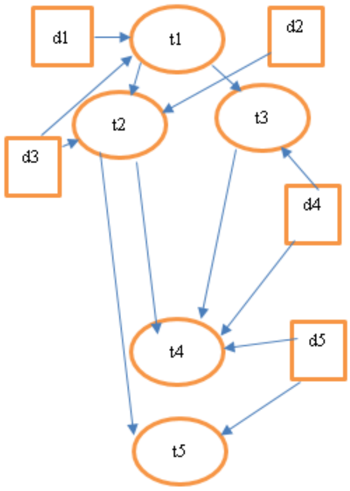


Fig. 1: Workflow diagram

Task grouping based on task dependency

Task grouping is a method that combines fined-grained jobs into coarse-grained jobs. Task grouping has verified to be an effective technique to decrease execution overhead and to increase the computational granularity of technical workflow jobs accomplishing on distributed resources. With job grouping, execution overhead can be removed by grouping small jobs are grouped together as one executable unit. This decrease benefits the cloud environment as an entire by decreasing traffic among the sites. The suggested scheme uses job grouping for fined grained tasks in user submitting workflow application to decrease information transfer time. First, we compute the dependences of tasks in workflow rendering to conditional probability.

$$P(T_a|T_b)=P(T_a \cap T_b)/P(T_a),\ P(T\square) > 0 \quad (1)$$

Which is the probability of task $t_a$occurring, given that task $t_b$occurs based on common usage of datasets. Clustering Method based on Task Dependency, noted as GTTD can be described as follows:

Algorithm 1.GroupingTechnique based Task Dependency (GTTD)

Input:task list task List

Output:all tasks are clustered

```
Begin
For t_a∈taskList Do
    Calculate P(t_a|t_b) for each pair of tasks
    DM [a] [b] = P(t_a/t_b)
End For
Index=0
For each element ele(a,b) ∈DM Do
    Selectmax (ele(a,b)) and a≠b
    Marka and note Task t_ais most dependent on t_b
    Addt_aand t_bto CL_index and remove row a and columna from DM
    Index=index+1
End For
End
```

In our illustration, we use the example scientific workflow as shown in [Fig. 1]. In this figure, there are five tasksand five datasets. As flow described in GTTD, conditional probability matrix should be built from joint and marginal probability of each task pair in workflow. Joint and marginal probability table can be created from contingency table as shown in [Table 1].

Table 1: contingency table

t_b t_a	T1	T2	T3	T4	T5	total
T1	1	1	0	0	0	2
T2	1	3	1	1	0	6
T3	0	1	2	2	0	5
T4	0	1	2	2	0	5
T5	0	0	0	0	1	1
total	2	6	5	5	1	19

Each value in contingency table is frequency of common usage of datasets for each task. According to Table 1, joint and marginal probability table can be built as displayed in [Table 2].

Table 2: Joint and marginal

t_b t_a	T1	T2	T3	T4	T5	total
T1	0.090	0.090	0	0	0	0.180
T2	0.090	0.136	0.045	0.045	0	0.316
T3	0	0.045	0.090	0.090	0	0.225
T4	0	0.045	0.090	0.090	0	0.225
T5	0	0	0	0	0.045	0.045
total	0.180	0.316	0.225	0.225	0.045	0.991

Table 3: Conditional Probability

t_b t_a	T1	T2	T3	T4	T5
T1	0.5	0.5	0	0	0
T2	0.284	0.430	0.142	0.142	0
T3	0	0.2	0.4	0.4	0
T4	0	0.2	0.4	0.4	0
T5	0	0	0	0	1

From [Table 2], conditional probability $P(t_a|t_b)$ for individually pair (t_a, t_b)of jobs can be designed as offered in [Table 3]. Constantly, job dependency for individually pair can be considered by means of conditional probability. In GTTD, jobs are grouped by the extreme value of to eachjob dependency from row by row if these pair of tasks is not same in number. In this case, if two pairs of jobs are not same in number, eg. $t_a \neq t_b$but concentrated value of tasks have same value; this techniqueprocedures the first pair of jobs as dependence. Task

Table 4: Task grouping by maximum conditional probability

t_b t_a	t1,t2	t3	t4	t5
t1,t2	0.5	0	0	0
t3	0	0.4	0.4	0
t4	0	0.4	0.4	0
t5	0	0	0	1

(b)

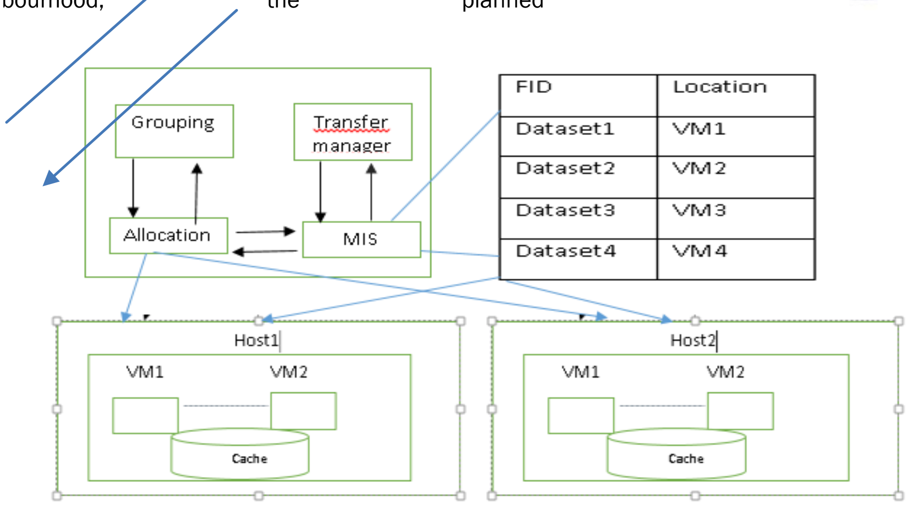
t_b t_a	t1,t2	t3,t4	t5
t1,t2	0.5	0	0
t3,t4	0	0.4	0
t5	0	0	1

After grouping, the projected scheme assigns jobs to resources in cloud environment with regard to data-aware development in [13]. Jobs are allocated to resources where essential input data are existed.

Meta information Service (MIS)

Data transfers among knowledge centres are unavoidable once running data-intensive advancement applications. The information transfer times don't seem to be negligible and should take a significant portion of the whole advancement execution time. Hence, each cloud system used for information intensive applications ought to showing intelligence assign the information to confirm a low level of worldwide data transmission quantity. This technique assumes that knowledge needed by the task should be out there at native cache of resources before each task will initiate execution and whereas running. It additionally assumes that knowledge are required for an advancement application is in a position to move from one execution host to different. To cut back information access latencies and increase knowledge

neighbourhood, the planned



FID	Location
Dataset1	VM1
Dataset2	VM2
Dataset3	VM3
Dataset4	VM4

Fig. 2: Proposed system architecture

…………………………………………………………………

System uses MIS holds key-value sets: file-ids (FID) and site (locations of file) in many VMs. The successor tasks in another VM will begin execution when needed input file from their predecessor tasks is accessible. These successor tasks got to request to Transfer Manager to induce the placement of intermediate knowledge that are generated in runtime of execution. During this case, Transfer Manager will facilitate to search out location wherever knowledge will exist via MIS, as pictured knowledge asphyxiation.

In scientific workflows with data-intensive work, individual jobs might have to be required to watch for massive amounts of knowledge to be delivered or made by different tasks. Advancement systems arrange to succeed high performance by showing intelligence planning tasks on resources, with making an attempt to move the most important knowledge files on the highest-capacity links. If the network link has unrestricted information measure, knowledge asphyxiation moves knowledge to the tasks that desires the information a lot of desperately than the opposite. Knowledge asphyxiation will scale back the unnecessarily wasted information measure that will be utilized by the opposite applications if the congestion limits the information measure for a few transfers.

In the planned system, Transfer Manager uses knowledge asphyxiation technique to move massive size of knowledge on highest capability links. This part will facilitate moving largest knowledge files by asphyxiation up and down between links once it receives requests from the tasks. It's hoped that this system can scale back transfer time and additionally improve performance.

RESULTS AND DISCUSSION

In this research, Montage workflow application is chosen from dissimilar technical regions as this scheme focuses on information-intensive workflow. According to [14], Montage workflow has low CPU consumption for several job types in the workflow (mBackground, mImgtbl, mAdd, mShrink). This is as Montage jobs spend much of their time on I/O operations

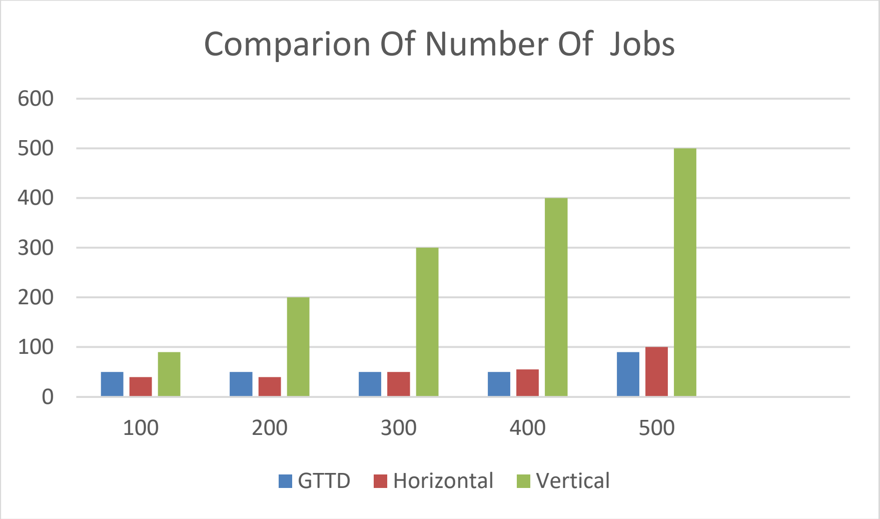


Fig. 3: Comparison of three Grouping methods.

For calculating the concert of the above grouping algorithm, WorkflowSim [13] framework is used in simulation. There are 20 single similar core virtual machines (worker nodes) are used to compose replicated computing platform. Each virtual VM has 2 cores as processing element (PE) and ability to process 1,000 million commands per second (MCPS). And network bandwidth is 15MB and 512 MB of memory. And Workflow Generator [15] is used to produce mock workflows through task runtimes and different number of tasks based on allocations gathered from running actual workflows.

CONCLUSION

Cloud computing has grown up in admiration in research community for organizing workflows. Tasks and datasets of workflow submission across worldwide circulated data centres need to be programmed conversing to information layout and makes pan for workflow implementation. This paper compares the quantity of jobs with additional grouping methods. The experimental consequence specifies GTTD beats others. Allowing to results, this suggested scheme will use GTTD to decrease execution overhead. It is predictable that this scheme will confidently aid the implementation of technical workflows well.

CONFLICT OF INTEREST

There is no conflict of interest.

ACKNOWLEDGEMENTS

None

FINANCIAL DISCLOSURE

None

REFERENCES

[1] Ostermann S, Plankensteiner K, Prodan R, Fahringer T, and Iosup A. [2009]Workflow monitoring and analysis tool for ASKALON. In Grid and Services Evolution.

[2] Suraj Pandey, [2010] Scheduling and Management of Data Intensive Application Workflows in Grid and Cloud Computing Environments, PhD Thesis, December 2010.

[3] Deelman E, Vahi K, Juve G, Rynge M, Callaghan S, Maechling PJ[2014]Pegasus: a Workflow Management System for Science Automation, Future Generation Computer Systems, pp. 17-35

[4] Park SM Humphrey. [2008] Data throttling for data-intensive workflows. In: Proceedings of IEEE international symposium on parallel and distributed processing, IEEE, pp 1–11

[5] JigneshLakhani, Hitesh Bheda.[2012] Scheduling Technique of Data Intensive Application Workflows in Cloud Computing, NIRMA UNIVERSITY INTERNATIONAL CONFERENCE ON ENGINEERING, NUiCONE,

[6] Piotr Bryk, Maciej Malawski, Gideon Juve, Ewa Deelman. [2015] Storage-aware Algorithms for Scheduling of Workflow Ensembles in Clouds, Journal of Grid Computing, to appear

[7] Fuhui Wu, Qingbo Wu, Yusong Tan. [2015] Workflow scheduling in cloud: a survey, The Journal of Supercomputing.

[8] Wang Ke, Qiao Kan, Iman Sadooghi, Xiaobing Zhou, Tonglin Li, Michael Lang, Ioan Raicu.[2015] Load-balanced and locality-aware scheduling for data-intensive workloads at extreme scales, Concurrency and Computation: Practice and Experience, 00:1-29

[9] Zhengxiong Hou, Jing Tie, Xingshe Zhou I. Foster M. Wilde. [2009] ADEM: Automating Deployment and Management of ApplicationSoftware on the Open Science Grid. GRID 2009.

[10] Dong Yuan, YY Xiao liu, Jinjun Chen, [2010]A data placement strategy in scientific cloud workflows, Future Generation Computer System,26(8): 1200-1214

[11] Mingjun Wang, Jinghui Zhang, Fang Dong, JunzhouLuo, [2014]Data Placement and Task Scheduling Optimization for Data Intensive Scientific Workflow in Multiple Data Centers Environment, Second International Conference on Advanced Cloud and Big Data,

[12] SaimaGulzar Ahmad, Chee Sun Liew, M. Mustafa Rafique, EhsanUllahMunir, Samee U. Khan. [2014]. Data-Intensive

Workflow Optimization based on Application Task Graph Partitioning in Heterogeneous Computing Systems. IEEE International Conference on Big Data and Cloud Computing (BdCloud2014)

[13] Chen W and Deelman E.[2012] WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments, in The 8th IEEE International Conference on eScience

[14] Juve G, Chervenak A, Deelman E, Bharathi S, Mehta G, Vahi K.[2013] Characterizing and Profiling Scientific Workflows, Future Generation Computer Systems, 29(3): 682-692

[15] Workflow Generator, https:// conflence.pegasus.isi.edu/ display/Pegasus/Workflow Generator.