HEURISTIC DATA PLACEMENT AND REPLICATION FOR SCIENTIFIC WORKFLOW IN CLOUD COMPUTING

Dissertation submitted in fulfilment of the requirements for the Degree of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

VISHALI

11507078

Supervisor

MR. PARMINDER SINGH

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

May, 2017

PAC FORM

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

Program : P172::M.Tech. (Computer Science and Engineering) [Full Time]

COURSE CODE : CSE546 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGD0008

Supervisor Name : Parminder Singh **UID :** 16479 **Designation :** Assistant Professor

Qualification : ________________________ **Research Experience :** ________________________

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Vishali	11507078	2015	K1518	9814399694

SPECIALIZATION AREA : Intelligent Systems **Supervisor Signature:** ____________________

PROPOSED TOPIC : Heuristic Data Placement and Replication for Scientific Workflow Applications in Cloud

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	7.50
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.00
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.00
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.25
5	Social Applicability: Project work intends to solve a practical problem.	6.75
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	7.75

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): NA
PAC Member 2 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 3 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 4 Name: Kewal Krishan	UID: 11179	Recommended (Y/N): Yes
PAC Member 5 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): NA
DAA Nominee Name: Kanwar Preet Singh	UID: 15367	Recommended (Y/N): Yes

Final Topic Approved by PAC: **Heuristic Data Placement and Replication for Scientific Workflow Applications in Cloud**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11024::Amandeep Nagpal **Approval Date:** 05 Mar 2017

4/27/2017 4:25:43 PM

ABSTRACT

Cloud computing has turned up as an emerging platform for individual or personal computing. Cloud computing providers provides different cloud services. In order to improve overall performance of cloud, data placement is an important task. Data placement is a prime issue which aims at minimizing the cost of inter node transfers of data in cloud especially for applications which are data intensive, the performance of the entire cloud system get improved by eradicating this issue. Data placement is NP-hard problem. Many authors has proposed different techniques for optimizing the data placement strategy in scientific workflow like k-means clustering, Hadoop based data grouping strategy, Ant colony optimization, Genetic algorithm. No existing solution is best. Strategy used for one application may not used for another application. Author Zhao devised a technique to reduced the inner node data transfer. Heuristic based method has been designed by considering the fixed and non fixed datasets. The clustering of task has been done to reduce the scheduling overhead. Data intensive workflow moves on high speed network bandwidth. This work is not considering replication of task which further improves the performance. Some datasets are used by many task or many datasets are used by one task. In this research work, replication of datasets has been introduced with existing data placement strategy. The appropriate data placement strategy reduces the scheduling overhead and cost of data processing. Replication process is used to speed up the access by providing copies of datasets to nearby data centres so that availability of data become high, bandwidth will be less consumed, scalability will be improved and fault tolerance will be increased. It is impossible to satisfy all the conditions to place the datasets at appropriate position where all task can access the data with minimum data transfer cost and fulfilment of SLA objectives. Experiment results show the improvement of makespan time over the existing technique.

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation entitled "HEURISTIC DATA PLACEMENT AND REPLICATION FOR SCIENTIFIC WORKFLOW IN CLOUD COMPUTING" in partial fulfilment of the requirement for the award of Degree for Master of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, India is an authentic work carried out under supervision of my research supervisor Mr. Parminder Singh. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Vishali

11507078

SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in the M.Tech Dissertation entitled "**HEURISTIC DATA PLACEMENT AND REPLICATION FOR SCIENTIFIC WORKFLOW IN CLOUD COMPUTING"**, submitted by **Vishali** at **Lovely Professional University, Phagwara, India** is a bonafide record of her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

(Parminder singh)

Date:

Counter Signed by:

1) **Concerned HOD:**
   HoD's Signature: ________________

   HoD Name: ____________________

   Date: ___________________

2) **Neutral Examiners:**

   External Examiner

   Signature: _______________

   Name: __________________

   Affiliation: ______________

   Date: ___________________

   Internal Examiner

   Signature: _______________

   Name: __________________

   Date: ___________________

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor Mr.Parminder singh for the continuous support for this study, for her patience, motivation, enthusiasm and immense knowledge. His guidance helped me all the time of research and writing this dissertation1 report. I could not have imagined having a better advisor and mentor for my M.Tech study.

His meticulous approach has improved the precision and clarity of my writing. I sincerely thank my family for unconditional support, trust and infinite love because of their encouragement, I would have been able to complete my work with zeal. It is also a pleasure to praise my friends for their valuable help and debugging some of the problems during the studies.

This research work is made possible through the help and support from everyone, including my parents, teachers, family, friends, and in essence, all sentient beings.

TABLE OF CONTENTS

CONTENTS **PAGE NO.**

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION TO CLOUD COMPUTING

The cloud computing is growing technology and helping other technologies to grow simultaneously. The cloud provides the infrastructure, platform and software as a utility which helps the scientists and industry to meet the requirements that were challenge due to infrastructure cost. As many companies and research institutes moves to public cloud, still many of them are still using their private or community cloud. Further combining the private and community cloud with public cloud comes with term hybrid cloud [8].

NIST definition of cloud computing, "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models " [9].

1.1.1 ESSENTIAL CHARACTERISTICS OF CLOUD COMPUTING

- **On demand self-service:** Without using human interaction the services such as network, computing, storage can provision from the service provider.
- **Broad network access:** The services provided through internet services and accessible on all types of mobile resources such as mobiles, laptops etc.
- **Resource Pooling:** Through multi-tenant system, services are provided to multiple tenant with sharing of resources. The virtual and physical resources are assigned through pay per use basis. Isolation between the task of multi-tenant will be provided.
- **Rapid Elasticity:** The scale in-out property of the cloud resources rapidly. As per the requirement of the user, the resources acquired and release on demand. The user view consider the unlimited resources.
- **Measured Services:** To provide transparency of services to the users as well as providers, the cloud resources must be monitored, controlled and timely reported [9].

1.1.2 CLOUD SERVICE MODELS

Cloud services are isolated in three classes. Figure 1.1 describes the layered structure of cloud services. Following are the classification of cloud services.

- **Software-as-a-service** The software as a service is very popular due to its remote accessibility through the internet with client browser. Small or medium size companies or entrepreneur are not able to afford the software cost can access the software through SaaS with very low cost and without purchasing the software license on hourly basis.
- **Platform-as-a-service** It provides the environment to the client for deploying the application. It provides the auto scaling, fault-tolerance and load balancing. The designer design the application by considering the platform and PaaS guide the programmer about the platform and information about the hardware and software requirement and availability.
- **Infrastructure-as-a-service** IaaS provides the virtual resources over one or more CPUs with various operating systems and software. Geographically distributed data centres can be accessed remotely and manage the infrastructure through the provider interface. The resources with different VM, operating system and billing cycle can be choosing through APIs [10].

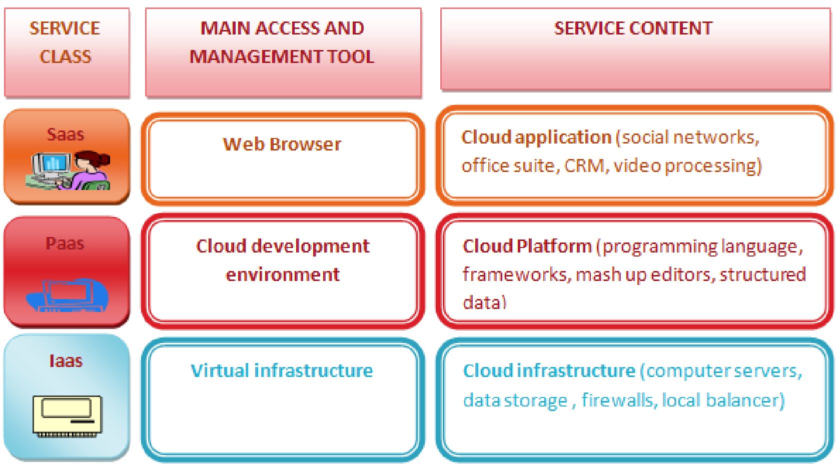

Figure 1.1: Cloud service model [1].

1.1.3 CLOUD DEPLOYMENT MODELS

The deployment of cloud is done in various ways. Figure 1.2 display the various type of cloud computing.

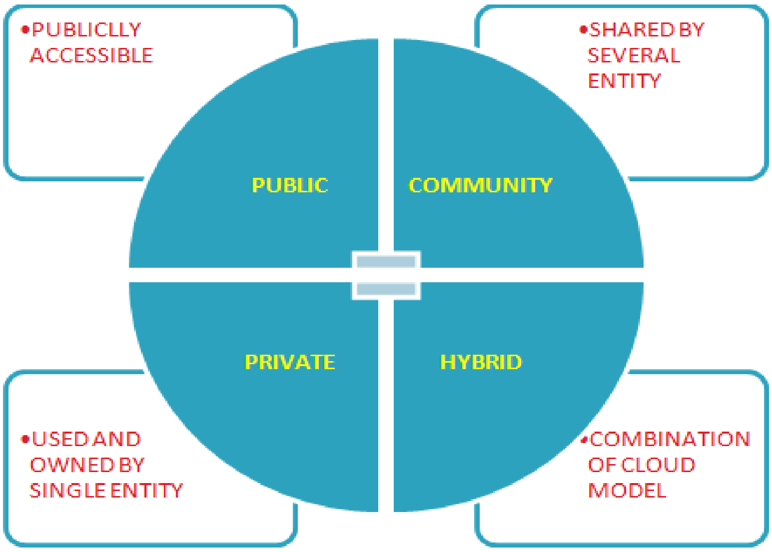

Figure 1.2: Cloud deployment model [2].

- **Private cloud** The private cloud is an infrastructure solely used for single organization for multiple users. The infrastructure can be managed by the same organization or the third party.
- **Public cloud** It is an infrastructure owned or managed by the private, government or any institute. It is for the use of general public. The cloud provider owned the infrastructure and other services and users access through the API.
- **Community cloud** Infrastructure is prepared to be used for the community of users sharing the same concern. This infrastructure can be managed by the any organization in the community or the group of organization from the same group or either by the third party.
- **Hybrid cloud** The combination of two or more cloud service models is hybrid cloud. They are working individually and combined through some standards. The load balancing between the data centres provided through cloud bursting [2].

1.1.4 SCIENTIFIC WORKFLOW

Scientific disciplines are knowledge driven with the help of data analysis and discovery pipeline. The series of data intensive and computational intensive tasks are designed composed and executed. The grid and cloud computing infrastructure attracts the scientist community the features such as sharing of computation, storage and software licenses. Multidisciplinary fields such as Bio-informatics, cheminformatics, geoinformatics etc. doing large investment on IT infrastructure [11]. The communities of scientists are interested in robust middleware which could afford the requirements of scientific tasks.

1.1.5 EXAMPLES OF SCIENTIFIC WORKKFLOWS

- **LIGO** Laser Interferometer Gravitational-Wave Observatory is observing and analysis astrophysical gravitational waves and further incorporates the data in the astronomy and physics research. The LIGO deals with black holes, gravity and nuclear matters. It is observing the birth of new black holes, stars and supernova in the universe [12].
- **MONTAGE** The montage scientific workflow is used to compute the mosaics of the sky images. The images collected first re-projected as per the coordinates. Next rectification of background has taken place. The co-added has done to create a big picture of sky. The project has been funded by NASA[13].
- **CYBERSHAKE** In the Southern California area a seismology application named CyberShake used to figures Probabilistic Seismic Hazard that are bends for geographic locations. This distinguishes all cracks inside 200km of the site of concern .It changes over burst definition into different varieties with contrasting hypocenter areas and slip circulations .Next step is to figures peak intensity measures and seismograms for each burst difference and joined with the first break probabilities to deliver probabilistic seismic risk bends for the site [3].
- **EPIGENOMICS** It is an information parallel work process. The Illumina-Solexa Genetic Analyzer gave us a Beginning information which is in the form of DNA grouping paths. Different paths of DNA successions can be produced by every Solexa machine. These information are changed over into a configuration that can be utilized by the software which is sequence mapping. One of two noteworthy undertakings can be done by the mapping software. It either takes all the short peruses, regards them as little pieces in an astound and afterward tries to amass a whole genome , or maps short DNA peruses from the arrangement information onto a reference genome. DNA arrangements then mapped to the right areas in a reference Genome by workflow. This produces a

guide that shows the grouping thickness demonstrating how often a specific arrangement communicates on a specific area on the reference genome [3].

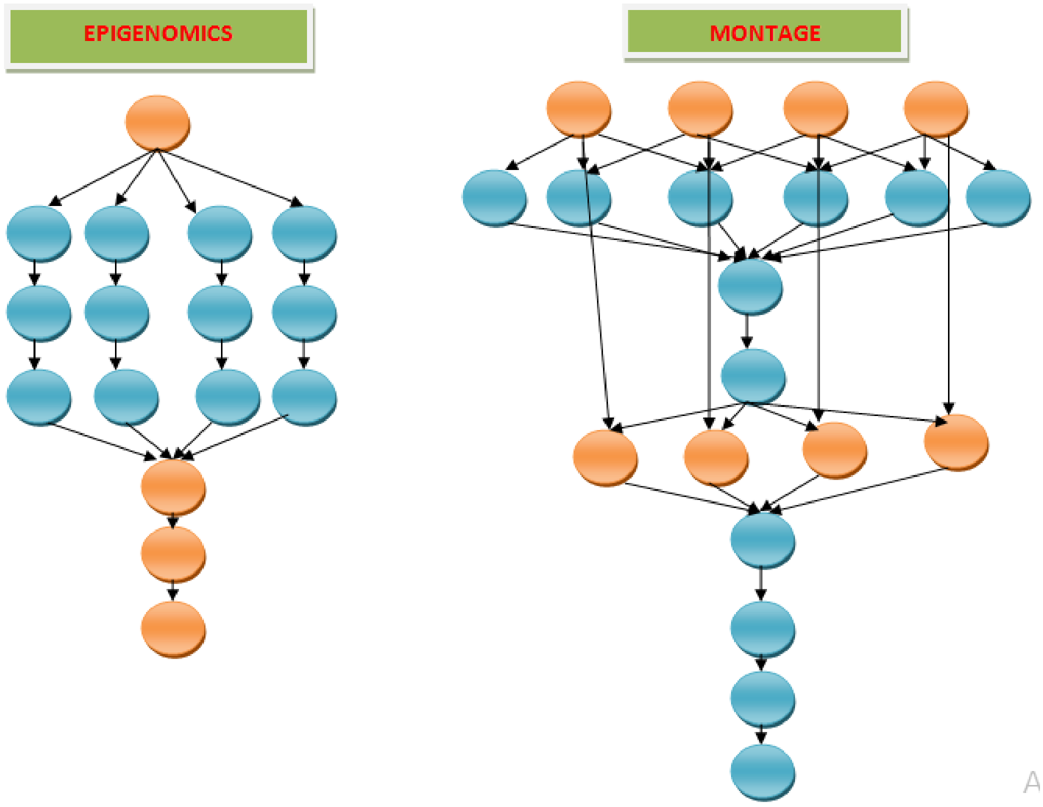

Figure 1.3: EPIGENOMICS and MONTAGE workflow [3].

- **SIPHT** The extensive expectation and comment of RNAs encoding qualities includes an assortment of individual projects that are executed in the correct request utilizing Pegasus. These include BLAST (Basic Local Alignment Search Tools), forecast of autonomous transcription eliminators, the comments of any RNAs that are found and correlations of the bury hereditary districts of various replicons. A wide look for little untranslated RNAs (sRNAs) conducts by SIPHT that manages a few procedures, for example, emission or destructiveness in microscopic organisms [3].

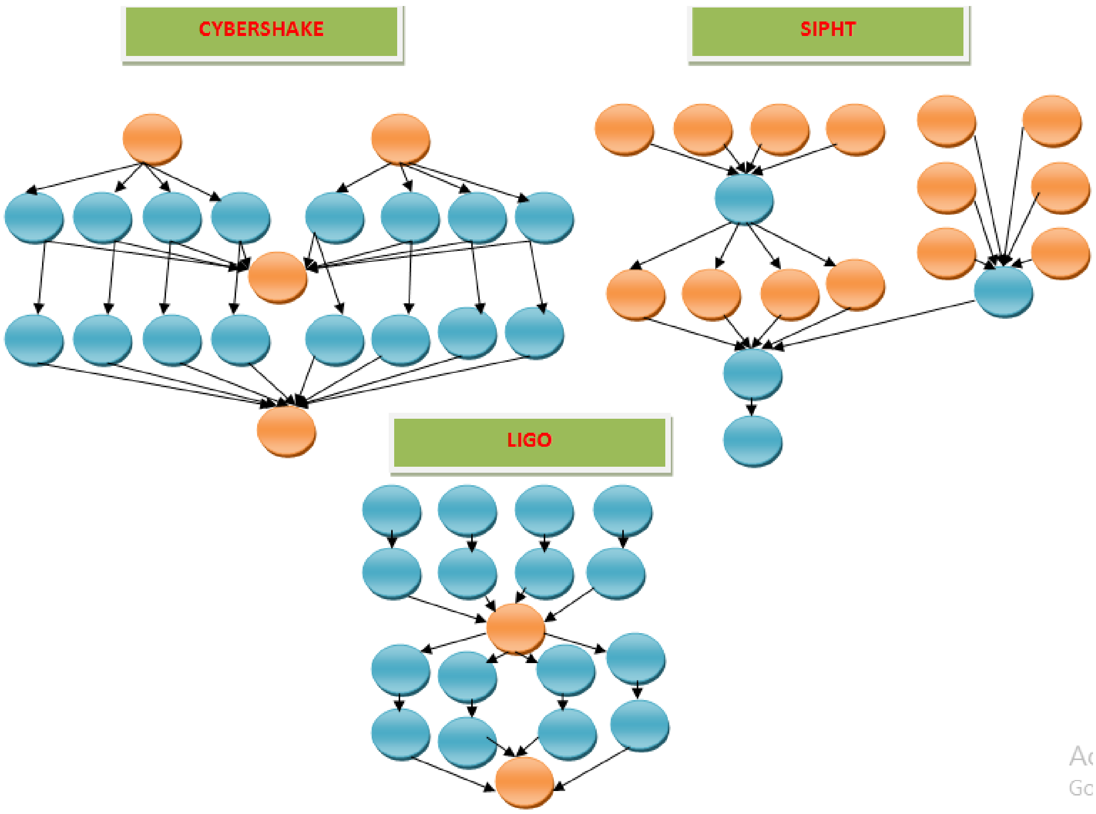

Figure 1.4: CYBERSHAKE, SIPHT and LIGO workflow [3].

1.2 SCIENTIFIC WORKFLOW MANAGEMENT SYSTEM

The four layer architecture for cloud computing has been implemented by researchers and scientist for the scientific workflow. The operation layer consists of the hardware resources which contains computing units, storage and network bandwidth. It also contains the virtual resources. Figure 1.5 display the scientific workflow architecture in cloud computing. WFMS permit clients to display and characterize work processes. They also fix spending restrictions, due date and the WFMS conditions in which we want to execute. At that point the WFMS assesses these information sources and executes them inside the characterized imperatives. The noticeable segments of a run of the mill cloud WFMS is given.

To characterize and display the unique work processes like assignments and their conditions a workflow portal is utilized. A dialect parser is used to parse in workflow enactment engine and takes the theoretical workflow. Non critical failure will be handled by workflow enactment engine. At that point, the assignment dispatcher investigations the conditions then dispatches the prepared undertakings to the scheduler. The scheduler, in light of the characterized planning calculations plans the work process undertaking onto an asset. It additionally contains an asset portion segment which designates assets to the errands through the asset agent. An Asset agent gives a brought together view to the enactment engine it directly interface with framework layer. The asset merchant speaks with process administrations to give the coveted asset. The index and inventory administrations house data about the process assets, the application and object of information .Workflow engine utilized this data, and the asset merchant to settle on basic choices. When all is said in done, workflow management service give critical administrations that are fundamental for the working of a WFMS. A guarantee validation and secure access to the WFMS will be given by workflow services. Observing devices continually screen fundamental parts of the WFMS it then raise alerts at fitting circumstances. A dependable stockpiling to intermediate and last information consequences of the work processes is given by Database administration segment. Provenance administration catches vital data, for example, elements of control streams and information, their movements, execution data, record areas, information and yield data, work process structure, framework data, work process development and frame [14]. Provenance is basic for deciphering information, investigating proprietorship, streamlining proficiency, giving again producible outcomes and deciding its quality furthermore to give adaptation to internal failure [15].

Work process scheduler, as specified before, a work process is an accumulation of assignments associated by information conditions. A fleeting connection between tasks demonstrates by the Work process structure. In both Directed Acyclic Graph (DAG) or non-DAG formats workflow can be spoken. In this proposition, work processes are spoken to in DAG positions (as shown in montage, ligo structure where the vertices speak to undertaking hubs and the coordinated edges speak to control as well as information dependencies. Scheduling maps work process errands on to circulated assets with the end goal that the conditions are not damaged. Work process Scheduling is an outstanding NP-Complete issue [16]. The placement of scheduler in WFMS is done by work process planning engineering

The detail working of SFMS layers are:

- **Operation Layer** This layer provides the hardware and software resources. The infrastructure layer provides the scale up and down features. This feature is provided dynamically by the cloud services. The Amazon EC2 [17], Google

[50] provides the cloud services. The interface is provided by the cloud providers.

- **Task Management Layer** This layer is providing the various components to manage the task and related data required for the application management. Data product management arranges and manage the data required for the scientific workflow management system. Provenance management look after the resources required for the specific task in cloud environment.
- **Workflow Management Layer** This layer is plays a major role in the scientific workflow management system. It contains the workflow engine that is input the scientific workflow in form of xml file and arranges the task in the workflow system. The monitoring of the workflow engine task is done by the workflow monitoring system. The failure of task is reschedule and this is monitored by the workflow monitoring. The workflow engine is used in the task management depend upon the type of workflow.
- **Presentation Layer** This layer provides the GUI interface through which user can input the scientific workflow. The interface can be browser specific or it can be console or window applications [18].

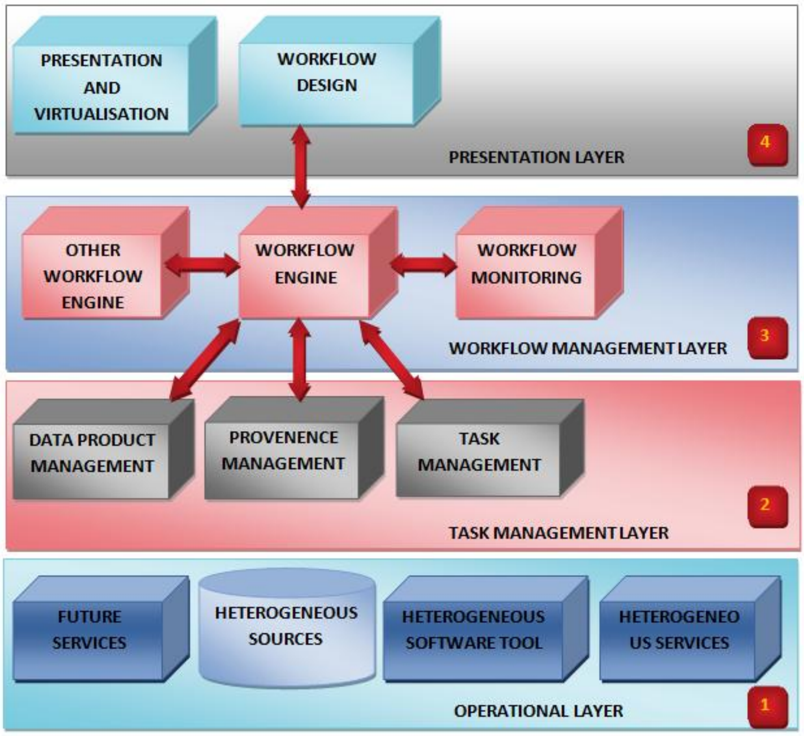

Figure 1.5: Workflow management system [4].

1.3 REPLICATION

Replication Technique is a vital strategy for expanding PC framework profit capacity. Using replication technique we can store same data on different servers. On the basis of Mater/slaves schema an algorithm is used for replicating the servers. At the starting of each administration period a selection for master is made. Every copy's status is condensed by an arrangement of epoch factor which is monotonically expandation. By Inspecting the epoch factor of replica of a dominant part uncovers updated information of different replicas. The arrangement of imitations can be changed progressively. Offline copies can be updated in backdrop. epoch variables are stored in deponent replicas but data is not stored in it and deponent replicas are the part of mass polling. The calculation does not require conveyed nuclear trans-activities. To customer machines algorithm additionally allows a storage of information replicas. It has strict reserve consistency which guaranteed for having the duplicated servers monitors which customers have reserved which information.

1.3.1 DATA REPLICATION

Data can be stored either locally or shared way in workflow for replication. Processing machines are used to put away data locally. For storing data in distributed way it will be stored in DFS(distributed file system) like HDFS(hadoop distributed file system) through which data can be replicated consequently. In spite of the fact that the previous approach is proficient, especially in information serious work processes, it is not efficient for tolerating fault. We have to execute the task again which will be affected while disappointment of server having some data. Then again, the last approach offers more adaptation to non-critical failure however is not proficient because of noteworthy system overhead and expanding the execution time of the workflow. Workflow of data intensive applications are executed on hadoop. HDFS utilizes replication technique for adaptation the non critical fault, which is static in nature. clients need to quantify that how much copies they want to form from data for replication. A huge stockpiling overhead occurs in static and blind based approach of replication due to this a MapReduce execution will be moderate down. One way to deal with adapt to this issue is to modify the replication rate progressively in light of the use rate of the information. It will decrease preparing expense of the assets and the capacity. For reliable dependability execution of work process CIR(cost effective incremental Replication) technique is used for replication. Different Replication techniques are used as input, output and at intermediate. There are four types of data intensive Replication strategies:- Asynchronous Replication, Synchronous Replication, Selective Replication and Rack-level Replication.

- **Synchronous Replication** Workflow task producers are obstructed in HDFS until replication wraps up in case of Synchronous information replication. If task producer of a block *A* profits, every one of the reproductions of piece An are ensured to be indistinguishable and any pursuer of block *A* can read any copy which means synchronous replication technique prompts high flexibility. Regardless, the downside of this approach is that the execution of task producer may get influenced as they must be blocked.
- **Asynchronous Replication** Similarly, replication for asynchronous information [19] permits task providers to continue without sitting tight for a replication to finish. The asynchronous information replication consistency is not as precise as the synchronous strategy in light of the fact that regardless of the possibility that an task producer of A block profits, a reproduction of piece A may at present be in the replication procedure. In any case,execution of the task providers enhances due to the non-blocking nature. For example, in Map and Reduce and Hadoop of asynchronous replication data can continue without being blocked.

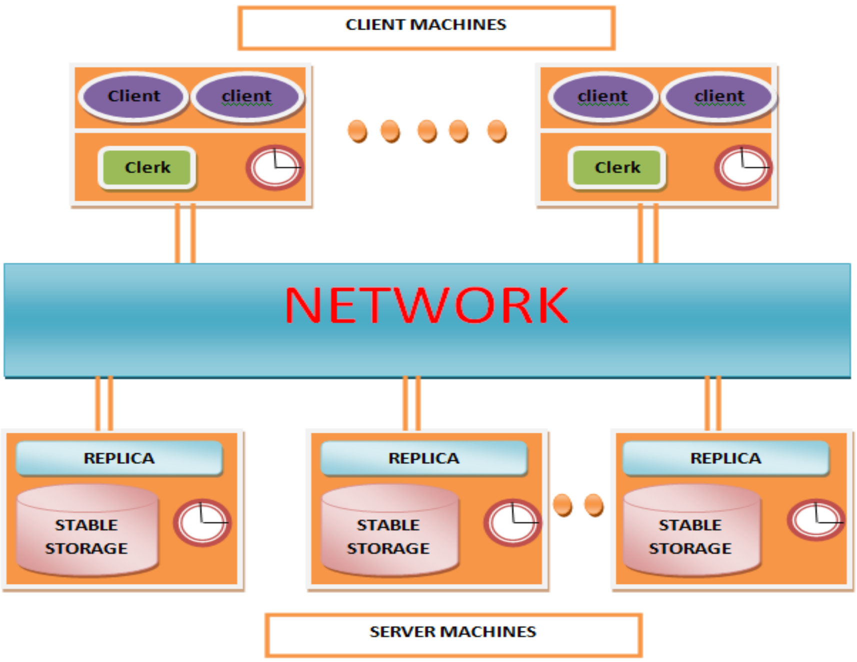

Figure 1.6: System Components [5].

- **Rack- Level** Information replication technique for Rack-level authorizes replication of the information obstructs on a similar rack in a server farm., machines are sorted out in racks with a various levelled arrange topology In cloud server farms. In this system topology the center switch can progress toward becoming bottleneck as it is shared by many racks and machines. Rack-level replication lessens the movement exchanged through the data transmission rare center switch. Be that as it may, the downside of the rack-level replication approach is that it can't endure rack-level disappointments and if a rack flops, everyone of the copies end up plainly inaccessible.
- **Selective Replication** of selective data is a technique where the information produced by the past stride of the work process are duplicated on the machine, where the fizzled assignment will be executed again.

1.3.2 TASK DUPLICATION

Undertaking duplication makes different copies or replicas of one task. Replication should be possible simultaneously for tasks, where every one of the reproductions of a specific assignment begin executing concurrently. When undertakings are imitated simultaneously, according to the schedule type the task for child begin its execution. For task replication there is scientific categorization .There are two types of schedule. one is strict and another one is lenient .when the execution of all the replicated task have completed then only task for child will started. while in the lenient type, the task for child undertakings begin execution when one of the reproductions completes execution. Replication of errand can likewise be performed in a reinforcement mode, where the recreated assignment is turned on when the essential errands fall flat [20]. This procedure is like retry or excess in time. Nonetheless, here, they utilize a reinforcement over-burdening system, which plans the reinforcements for numerous errands in a similar day and age to viably use the processor time [21].

Duplication is utilized to accomplish different destinations, the most widely recognized being tolerance from fault [22]. The excess undertaking helps in fulfilment of the execution. When one assignment fizzles. Furthermore, calculations utilize information duplication where information is imitated and pre-arranged, in this manner moving information close calculation particularly in information concentrated work processes to enhance execution and unwavering quality. Besides, evaluating undertaking execution time from the earlier in an appropriated situation is challenging.

Imitations are utilized to go around this issue utilizing the consequence of the most punctual finished reproduction. This limits the timetable length to accomplish hard due dates [23], as it is powerful in dealing with execution varieties. Increment asset in replication use with no additional cost, repeated assignments out of gear availabilities to diminish the calendar length.

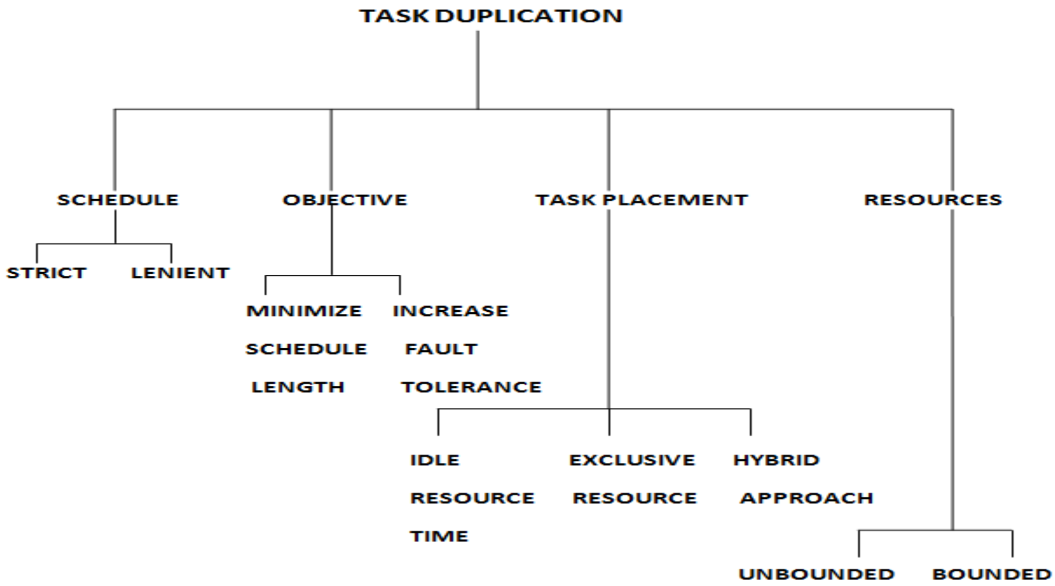

Figure 1.7: Task Duplication [6].

Researchers can replicate task in two ways on new resources or on idle cycle of resources and hybrid approach is also used for replication. Resources which remain idle without moving cycles are those schedule vacancies in the asset utilization period where the assets are unused by the application. Plans that duplicate in these sit without moving cycles profile assets to discover unused schedule vacancy, and repeat errands in those openings. This approach accomplishes advantages of errand duplication and at the same time considers money related expenses. By and large, be that as it may, these sit out of gear spaces won't not be adequate to accomplish the required goal. Thus, assignment duplication calculations generally put their undertaking imitations on new assets. These calculations exchange off asset expenses to their targets. There is a huge assemblage of work around there enveloping stages like bunches, cloud and matrices [24]. Assets considered can either be limited or unbounded relying upon the stage and the method. Calculations with limited assets consider a constrained arrangement of assets. So also, a boundless number of assets are accepted in an unbounded framework condition. Asset sorts utilized can either be homogeneous or heterogeneous in nature.

Homogeneous assets have comparable qualities, and heterogeneous assets in actuality differ in their attributes, for example, preparing speed, CPU centres, memory and so on. An improved pursuit and duplication based planning calculation (SDBS) that considers the variable undertaking execution time is introduced. They consider a dispersed framework with homogeneous assets and accept an unbounded number of processors in their framework.

1.4 DATA PLACEMENT

A data placement is a mechanism in which abundant datasets used to enhance the throughput or performance and it also includes the study of data movement cost. In scientific cloud workflows, a giant number of applications demands data to be deposited in distributed data centres. A data manager must brilliantly choose data centers to efficiently store these data where the deposited data will reside. Even though, it is not the scenario for data with a fixed location/position. The ideal data placement strategy upgrade the execution of the data intensive scientific workflows in cloud. This up gradation is done by allocating the tasks to the execution site in such a manner that the transfer of files and the cost associated with these are lowered. There are different type of placement services for data which are used in scientific workflow. Three general classes of calculations are: those that look to stage information proficiently into calculations; those worried with arranging information out of computational assets; and calculations intended to give information unwavering quality and sturdiness. The top of the line of information arrangement calculations is worried with organizing information into calculation investigations productively. substantial work process made out of thousands of related assignments, every errand that is apportioned for execution on a computational hub requires that before calculation can start its information records be accessible to that hub. the information position benefit should be considered a Particular qualities of information access amid work process execution additionally [25]. For instance, information things have a tendency to be gotten to as related accumulations as opposed to exclusively, and a position administration would preferably put things in a gathering together on a capacity framework to encourage execution. Also, information get to designs have a tendency to be busty, with numerous information arrangement operations occurring amid the phase in (or organize out) period of execution. It gaining position benefits that move informational indexes non concurrently onto capacity frameworks open to computational hubs, in a perfect world before work process execution starts with the goal that work process undertakings don't have to sit tight for information exchange operations to finish.

Other position calculations attempt to calendar occupations on or close hubs where informational collections as of now exist. By and by, arranging information out of computational assets effectively may likewise be a noteworthy test for logical applications. At the point when these applications run vast investigations on conveyed assets (e.g. on the Open Science Grid, which gives various assorted disseminated assets to logical joint efforts), the individual hubs that run computational occupations may have restricted capacity limit. At the point when an occupation finishes, the yield of the employment may should be arranged off the computational hub onto another capacity framework before another occupation can keep running at that hub [26]. In this manner, an information arrangement benefit that is in charge of moving information proficiently off computational hubs can largy affect the execution of logical work processes.

A third arrangement of information position calculations is worried with the upkeep of information to give high accessibility or toughness, i.e. insurance from information disappointments. These arrangement calculations repeat information to keep up extra duplicates to secure against impermanent or lasting disappointments of capacity frameworks. For instance, a situation administration of this sort may make another imitation of an information thing at whatever point the quantity of open reproductions falls underneath a specific edge. These replication calculations may be responsive to disappointments or might proactively make reproductions.

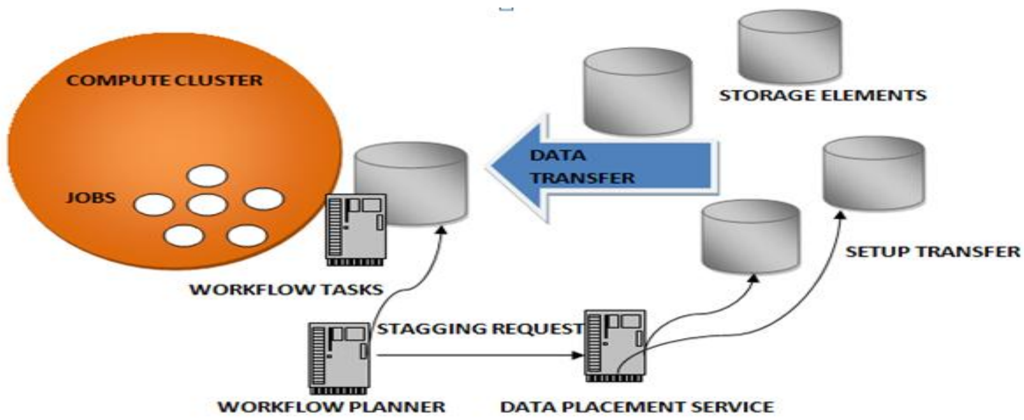

Figure 1.8: Data Placement Services [7].

1.4.1 CHALLANGES OF DATA PLACEMENT STRATEGY

There are number of issues occurred while moving data from one place to another place which violates SLA. After allocating data next step is to placing the data. Some of the challenges are faced by the data placement strategy of cloud computing which are as below:-

- **Data management:** For data intensive scientific application we need to collaborate the data from different data centres. huge amount of data sets need to be managed properly. so, manage data for data placement is big challenge.
- **Data movement:** Movement of data from one data centre to another data centre is also an another challenge .there are different conditions where moving of data becomes a challenge which are, when data sets are very large and not feasible to move, when datasets have fixed locations and it also includes a case where data can easily be moved but cannot moved.
- **Increasing Cost:** Data movement of different data sets accompanies various service providers of cloud which results in the cost increment.
- **Infrastructure:** Hidden infrastructure of cloud computing hides the physical location of datasets of user where their data is stored. They just provides the storage resources and computation to their users. So hiding information becomes a challenge.
- **Fault tolerance:** While moving multiple datasets from one place to another there will be a higher risk of occurring a failure in between the process.
- **Performance:** Speed of a whole process will be reduced when data will be placed at different data centres and also data are too large therefore performance will also be reduced.
- **Bandwidth:** More bandwidth will be used in case of placement of data from one place to another.

CHAPTER 2

REVIEW OF LITERATURE

Service providers are facing various data placement issues while deploying the scientific workflow in cloud environment. Taking into consideration the issues, following review of literature has been conducted.

Table 2.1: Taxonomy on Data Placement

AUTHOR	YEAR	TITLE	APPROACH
Jiong et al.	2010	”Improving map reduce performance through data placement in heterogeneous hadoop clusters”	Proposed a data placement strategy for hadoop heterogeneous system. It is considering only one type of system and cannot be generalize
Dong et al. .	2010	“A data placement strategy in scientific cloud workflows”	K-means clustering has been used with matrix. It included all conditions Heterogeneous environment except network bandwidth. It reduce the data movement significant level
Liu et al.	2011	”A novel general framework for automatic and cost-effective handling of recoverable temporal violations in scientific workflow systems”	To model the data dependencies DAG graph has been used. Storage resource selection is used to select the appropriate resource
Xiao Liu et al.	2011	”Preventing Temporal Violations in Scientific Workflows: Where and How”	To prevent temporal violations in scientific workflow an author proposed a novel cost effective probabilistic temporal setting technique inspite of expensive standard exception handling technique
Dharma Teja Nukarapu et al.	2011	”Data Replication in Data Intensive Scientific Applications with Performance Guarantee”	Replication algorithm and caching algorithm is used in this paper for decreasing the delay for aggregate information record the main reason for delay reduction is replication. Construction of a replication

			algorithm which is centralized is there of plynomial time. caching algorithm is constructed then in light of replication algorithm
Weiwei Chen	2012	”WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments”	workflowsim is the extended simulator of cloudsim that are used for scientific workflow . heterogeneous system failure and overhead was occurred in Previously used simulator of workflow because of its different framework. clustering of task was also not taken into consideration in existing workflow simulator.
Zhao et al.	2012	“A data placement strategy based on genetic algorithm for scientific workflows”	Genetic algorithm has been used for data placement strategy. Load balancing of data centers considered while placement. The method is not describing data placement in heterogeneous Environment
Cong Wan et al.	2012	”A QoS–Awared Scientific Workflow Scheduling Schema in Cloud Computing”	Experiment through simulation demonstrated the scheduling algorithm which decreases the processing time and cost.
Peter et al.	2012	”A constraints-based resource discovery model for multiprovider cloud environment”	Proposed a cloud infrastructure provider that provides cost effectual and extremely flexible resources for use.
Kassian Plankensteiner et al.	2012	”Meeting Soft Deadlines in Scientific Workflows Using Resubmission Impact”	Proposed a method for heterogeneous circulated and parallel figuring condition which is used for fault tolerant in scientific workflow. This is an another heuristic execution. Our strategy can be completely utilized in new conditions as opposed to the another methodology.
Jianwei et al.	2013	”A classification of file placement and replication methods on grids”	Considering the grid environment and demonstrate the file replication and data transfer where more number of job failures occurred.
Zhangjun et al.	2013	”A market-oriented hierarchical	Ant colony optimization (ACO) is used for data placement and demonstrates the

		scheduling strategy in cloud workflow systems"	performance is significantly better than other algorithms.To calculate the dependencies DAG graph has been used.
Christina Hoffa	2013	"On the Use of Cloud Computing for Scientific Workflows"	Focuses on the montage application of astronomy which is being widely Used nowadays. Basically Montage is a workflow that provide short job runtime.
Wang et al.	2014	"DRAW: A new Data gRouping-Aware data placement scheme for data intensive applications with interest locality"	Hadoop based data grouping strategy has been design by analyzing the log file and find the correlation among data to achieve maximum parallelization
Chase et al.	2014	"End-to-end Delay Minimization for Scientific Workflows in Clouds under Budget Constraint"	author discussed about the development of prototype generic workflow system using influence of existing technologies for rapid calculation of scientific workflow optimization strategies
Poola et al.	2014	Robust scheduling of scientific workflows with deadlines and budget constraints in cloud"	With the proposed strategies introduced, the RTC arrangement demonstrates the maximum robustness and in the meantime reduces make span of the workflows.
Rodrigo N. Calheiros	2014	"Meeting Deadlines of Scientific Workflows in Public Clouds with Tasks Replication"	They introduced an algorithm to relieve impacts of execution variety of resources on delicate due dates of applications of work flow. To imitate the task algorithm utilizes the idle time of spending surplus and planned resources.
Ghafarian et al.	2015	"Cloud-aware data intensive workflow scheduling on volunteer computing systems"	Figure portioning of DAG has been used to balance the size partitioning. It is only working on reducing data frequency but not considering the data size. Not suitable for data intensive workflows.

Zhao et al.	2015	”A data placement strategy for data intensive scientific workflows in cloud”	Not fixed dataset has been considered. But this paper not considering the network bandwidth.
Yong et al.	2015	” A Service Framework for Scientific Workflow Management in the Cloud”	The cloud platform like Eucalyptus and Open Nebula are used according to their service framework, in which cloud workflow management service, cloud resource manager, cluster monitoring service, client side tool are developed.
Weiwei et al.	2015	”Dynamic and Fault-Tolerant Clustering for Scientific Workflows”	This framework maximum likelihood Estimation based parameter for modeling workflow performance.
Jianbing et al. .	2016	” Auction-based cloud service differentiation With service level objectives”	They develop a auction mechanism which is very effective. Abacus as of now just handles independent calculation resources in the framework
Zhao et al.	2016	”Heuristic Data Placement for Data-Intensive Applications in Heterogeneous Cloud”	Data intensive workflow moves on high speed network bandwidth. It improve the cost and data movement. This work is not considering replication of task which further improve the performance.

Xie, Jiong et al. (2010) proposed the technique to place the data on heterogeneous environment. It improves the performance of the map-reduce system. The load balancing of task has been performed in all the nodes in cloud environment. The amount of data placed at particular node is an challenging task. The research work is performed on data intensive applications [27]. The result shows that placement strategy improves the load balancing performance. It resolves the issue of performance degradation into the scientific workflow. Fragmentation problem exists in the previous techniques and proposed technique overcome the issue.

Yuan, Dong et al. (2010) devised a method to work on data intensive applications on distinct memory location. It consider the two types of data, one is of fixed location as well as non fixed location. The k-means clustering has been used for the clustering of data. Clustering of task reduces the scheduling and data placement overhead. Dynamic clustering has been proposed to handle the new generated datasets within the data center. The major objective of the work is reduce the data movement inside the data center [28]. The hadoop replication mechanism is used to reduce the data movement among the datacenter. Balanced distribution of data is provided with the appropriate placement strategy.

Liu, Xiao et al. (2011) develop a novel method for complex scientific workflow. Temporary violation of SLO in reduces the performance of scientific workflow. To deal with violation of temporary types several cost effective methods are introduced and widely used in the industry. The novel exception handling mechanism has been used to resolve the temporal violations. The fine grained level temporal violations exception handling mechanism has been used. Checkpoint techniques have been used for temporal violation handling. The technique has been developed to make the process automatic and cost-effective [29].

Xiao Liu,Yun Yang (2011) proposed a novel cost effective probabilistic temporal setting technique inspite of expensive standard exception handling technique to prevent temporal violations in scientific workflow. where and how these are two problems for temporal setting have been efficiently examined and tended to. Two parts of where, specifically, the likelihood consistency area where setting of temporal is measurably powerful and the choice of vital and adequate change focuses, have been tended to, individually, by a likelihood based runtime temporal consistency model and a base likelihood time repetition based alteration point determination system [30].

Dharma Teja Nukarapu et al. (2011) Replication algorithm and caching algorithm is used in this paper for decreasing the delay for aggregate information record the main reason for delay reduction is replication. caching algorithm is constructed then in light of replication algorithm. Data grid is used in distributed environment that can be used for caching algorithm. By comparing their unified algorithm with other algorithm which are heuristic they demonstrate that both performs equivalently under various system parameters. Gridsim is utilized. The conveyed caching method essentially outflanks a current well known document storing strategy in Data Grids which is versatile and scalable [31] .

Weiwei Chen (2012) workflowsim is the extended simulator of cloudsim that are used for scientific workflow. Heterogeneous system failure and overhead was occured in Previously used simulator of workflow because of its different framework. clustering of task was also not taken into consideration in existing workflow simulator. At runtime of simulator there can be inaccuracy or error in result also because of failure and overhead of system. Workflowsim gives a higher layer of workflow administration. On the basis of real traces they compare the results of simulation. By using workflow simulator researchers evaluate a special and compelling platform for workflow [32]. In advancement in another research work they present another promising area of research.

Er-Dun, Zhao et al. (2012) proposed a data placement strategy according to the storage capacity of the datacenter. Data placement is an NP hard problem but some effort made to reduce the data movement among the data center. Existing techniques used the k-means clustering. The k-means clustering focused only on few data centers only which effect the data placement strategy. Genetic algorithm has been used to select the optimal solution [33]. Heuristic algorithm has been proposed for data placement as well as maintains the load balancing of data centers .

Cong Wan et al. (2012) discuss about the investigation of scientific workflow scheduling schema which is done under cloud computing environment .the result for experiment through simulation demonstrated the scheduling algorithm which decreases the processing time and cost .cost is reducing with response time limitation and processing with budget limitation in scientific workflow [34]. The future work in this paper is solving some problem like algorithm for time complexity is rather high. It will be cost quite a while given more sorts of example. Such Issue will be resolved with the help of heuristic calculation.

Peter Wright, Yih Leong Sun (2012) proposed a cloud infrastructure provider that provides cost effectual and extremely flexible resources for use. This infrastructure commercial center is growing quickly with new suppliers, infrastructure items and quality included services going to the business sector. This quick improvement environment places huge strain on previously infrastructure application clients as a result of the complexity of selecting suitable resources from a dynamic commercial center. But the biggest challenge is mapping an application’s necessities onto an arrangement of resources. Two stage resource choice model is build up utilizing a limitation based approach which empowers clients to coordinate their applications' prerequisites to infrastructure resources [35].

Kassian Plankensteiner et al (2012) proposed a method for heterogeneous circulated and parallel figuring condition which is used for fault tolerant in scientific workflow. This is an another heuristic execution. Without trace model of failure or authentic executions our strategy is effective. Rescheduling heuristic ready and dynamic order is proposed with a high level of adaptation to non-critical failure for executing workflow, while considering delicate due dates. Our strategy fundamentally diminishes the resource misuse contrasted with moderate assignment replication and resubmission methods which is showed by simulation experiment of three certifiable work processes in the Austrian Grid [36] .

Ma, Jianwei et al. (2013) devised the file placement classification methods with replication. The virtual image technique has been used in the paper. The replication process has been optimized in middleware of cloud. Cost is considered for file placement and replication strategy. Classes has been formed for the replication [37]. The grid infrastructure has been used for the implementation of the technique. This method can be implemented in cloud environment. New placement strategy can be designed by considering the cloud infrastructure. The replication strategy is easy to design with the cloud environment despite the cost of replication is higher.

Wu, Zhangjun et al. (2013) presents hierarchical scheduling as market-oriented strategy for cloud computing workflow system. The global market is mapped with the task assignment of the service. The local cloud data center optimization is main objective of this research. QoS constraints need to be satisfied for each task of cloud workflow. Metaheuristic based scheduling strategy has used such as GA, ACO and PSO. CPU time, makespan and cost considered as metrics [38]. Heuristic and meta-heuristic is need to be devised for task and service level scheduling. The optimization of local data center can be done without considering the global data centers.

Christina Hoffa1 et al. (2013) exploits the basic use of the scientific workflows in cloud. Focuses on the astronomy application montage which is being widely used nowadays. Basically Montage is a workflow that provide short job runtime. This approach is able to provide good compute time performance but the challenge or main problem in that is it can be suffer from the resource scheduling delay and wide areas communication. Previously the cloud was under developed that won't be able to provide solution to flexible, on demand computing infrastructure to various applications [39]. In cloud computing there are a number of virtual servers which work collaboratively by internet and they can be dynamically managed, maintained and monitor.

Wang, Jun et al. (2014) proposed the compute and storage intensive framework for hadoop based cloud. The load balancing and random placement strategy has been developed. The data grouping strategy has been designed for the data placement named as DRAW. The data grouped into nodes of small sizes. The parallelism has been achieved by divide the data into small size and distribute to the parallel nodes. The technique is learning the grouping data logs and clustering the data and organizing and reorganizing the data [40]. The result shows the significant improvement in the technique.

Chase Qishi Wu, Xiangyu Lin (2014) discussed about the development of prototype generic workflow system using influence of existing technologies for rapid calculation of scientific workflow optimization strategies. They used cloud based computing resources. what's more, define an undertaking booking issue to minimize the work process end-to-end delay under a client indicated money related limitation. They plan a heuristic answer for this issue, and show its execution predominance over existing strategies through vast simulation also, genuine work flow tests in view of evidence of-idea execution and arrangement in a local cloud tested [41].

Deepak Poola, Saurabh Kumar Garg (2014) presents three resource allotment policies with cost, robustness and makespan as its objective. For making the schedule robust by considering the budget and deadline constraint, the resource allotment policies adds slack time to it. Author test these policies with two failure models for five scientific workflows with two metrics for robustness. Results shows that these policies are being robust against doubts like performance variations and task failures of virtual machine. With the proposed strategies introduced, the RTC arrangement demonstrates the maximum robustness and in the meantime reduces make span of the workflows. The RTC strategy contributes a robust schedule with expenses hardly higher than the reference algorithm investigated [42].

Rodrigo N. Calheiros (2014) For correcting the delay a finite contingency methodologies is implemented. Delay has occurred due to some factors which are conveyed resources of public cloud performance fluctuate or execution time of task underestimation. They introduced an algorithm to relieve impacts of execution variety of resources on delicate due dates of applications of work flow. In this scientific workflow simulation is used which do experiment that shows within the budget their algorithm maintain the replication and total execution time so that work done within the deadline. When replication of task will increased we lessens the aggregate execution time [43].

Ghafarian et al. (2015) devised the technique to place and schedule the data intensive workflow on cloud resources as well as volunteer computing. The workflow has been divided into sub workflows to decrease the data dependency between the tasks. As per the requirement of resources first the sub workflows schedule to the volunteer computing system. If the task is taking more time as compare to expect than the task has be schedule to the cloud [44]. Provisioning algorithm VOLNT has been proposed. Two strategies have been proposed, first technique reduce the cost by considering the deadline and second strategy improve the cost by considering the deadline.

Zhao, Qing et al. (2015) proposed the 2-stage placement strategy for data placement. First stage refers to the clustering of the task by considering the correlation of task. The correlation of task has been compared with the co-tasks. The correlation of intermediate datasets and task also verified and considered in the technique. The novel correlation technique has been introduced as "first order conduction correlation". Second stage is runtime, the distribution and redistribution algorithm look after the layout of the data [45]. The technique can further improve by considering some other important factors such as network bandwidth. Replication technique can further used to improvise the method.

Yong Zhao and Wenhong Tian (2015) reference service framework is proposed for integrating scientific workflow management systems into different cloud platforms. In integrating the SWfMSs an implementation effort is also presented in this paper. The cloud platform like Eucalyptus and Open Nebula are used according to their service framework, in which cloud resource manager, cloud workflow management service, cluster monitoring service, client side tool are developed. The usage can rapidly be utilized for OpenStack as it is getting more demanding in scientific research area as well as in business applications. They are additionally researching the joining of different SWFMSs into these different clouds [46].

Weiwei Chen, Rafael Ferreira da Silva(2015) Here, Theoretical analysis is conducted by the author in which the analysis is of impact of transient failure on runtime performance of scientific workflow execution. they suggest general task failure modeling framework This framework maximum likelihood estimation-based parameter for modeling workflow performance. To improve runtime performance of workflow execution they suggest three fault tolerant clustering strategies which works in faulty execution environments. Results for experiment demonstrated that the proposed strategies essentially enhance the work flow make span when contrasted with a current task clustering technique utilized as a part of work flow management frameworks [47]

Jianbing Dinga, Zhenjie Zhangc (2016) talks about Abacus. It is auction based resource assignment framework which is for cloud computing. It gives effectual service difference with several budgets and priorities for jobs. They develop a auction mechanism which is very effective. Abacus as of now just handles independent calculation resources in the framework. So there can be the future work in which the dependent resources can be handled. Hence, another theme for future study is to analyze the convergence speed theoretically [48]. It will be difficult to sketch mechanism of resource allocation for complex cloud applications.

Zhao, Qing et al. (2016) devised a technique to reduce the inter node data transfer. Heuristic based method has been designed by considering the fixed and non-fixed datasets. The clustering of task has been done to reduce the scheduling overhead. Two methods has been proposed. First method deal with the non-fixed dataset and data allocation has been done at optimize location. Fixed has been placed to the appropriate location. Build time and run-time data placement heuristic has been proposed and implemented [49]. Further this technique can be improved by implementing the replication strategy.

CHAPTER 3
PRESENT WORK

3.1 PROBLEM FORMULATION

Data placement is an important task to reduce the data movement in the cloud datacenters. The appropriate data placement strategy reduces the scheduling overhead and cost of data processing. Existing strategy are worked on reducing the data movement among the nodes of data centers by clustering of jobs and placement at appropriate data centers. Data size is an important factor in data placement and placement of big data jobs is still a challenge. Coarse grained jobs take much time to transfer and increase cost in case of replication. Author Qing Zhao has worked on heuristic data placement by considering many factors and design heuristic on the behalf of those factors. Transfer the jobs through high speed links increase the performance of data placement. Some datasets are used by many task or many datasets are used by one task. So to increase the performance of data placement strategy replication of dataset can be done, so that data can be available at more than one place and further reduce the data movement among the data centers for scientific workflow applications.

3.2 OBJECTIVES OF STUDY

Therefore I devised the following objectives.

1. To develop a model for data allocation for scientific workflow applications.
2. To design a heuristic based data placement algorithm using replication to reduce the data movement among the datacenter nodes in cloud environment.
3. To implement the proposed and existing algorithms using workflowsim.
4. To analyze the algorithms on performance metrics e.g. makespan.

CHAPTER 4
RESEARCH METHODOLOGY

4.1 TREE STRUCTURE MODELING OF DATA SET

In the scientific workflow application provides two types of data granularity: fine grained and coarse grained. This research work focuses on the coarse grained workflow tasks. Placing the dependent data on nearby places or on same node decrease the data movement cost. The existing algorithm has considered all data set can be placed at any data centers, but in actual situation the scientific workflow task consist of fixed and non-fixed dataset due to the ownership of datasets.

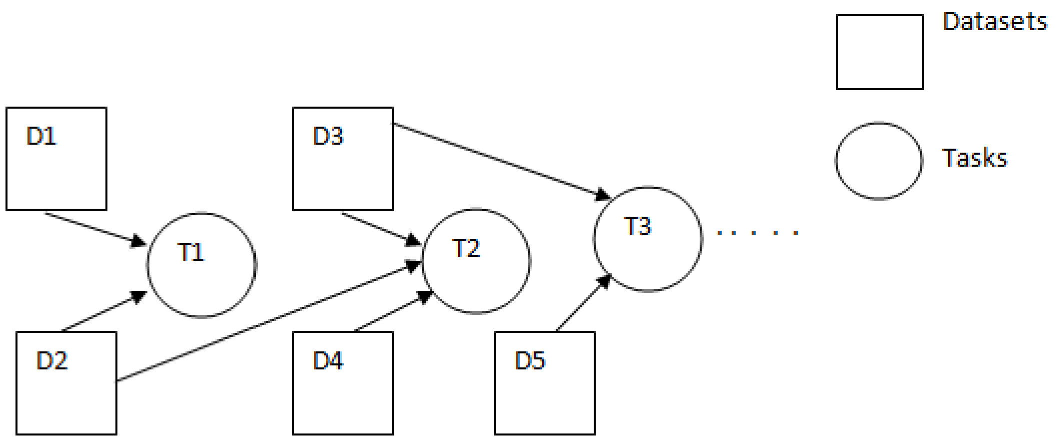

Figure 4.1: Tasks and Datasets

Case 1: (Storage requirement for non-fixed datasets) In this case prefer to move the smaller dataset to the position of bigger dataset, which reduce the data transmission.

$$c_{i,j} = |T_i \bigcap T_j| \times \min\left\{\text{Size}_{d_i}, \text{Size}_{d_j}\right\}, \quad d_i, d_j \in D_{\text{in}}(t). \qquad \text{......(i)}$$

In the above equation **Ti∩Tj** represent the tasks which required the data set **di** and **dj**. **min** {**Sizedi** , **Sizedj**} calculates the smaller data set.

$$c_{i,j} = \max\left\{\{\text{Size}_k \mid d_k \in D_{\text{in}}(t),\ k \neq i \neq j\} \cup \{\text{Size}_i + \text{Size}_j\}\right\} - \max\{\text{Size}_m \mid d_m \in D_{\text{in}}(t)\}. \quad \text{.......(ii)}$$

The increased amount due to data transfer is calculated as dependency gain. The dependency gain :-

$$c_{i,j} = \left(\sum_{d_m \in D_{\text{in}}(t)} \text{Size}_m - \max\{\text{Size}_m \mid d_m \in D_{\text{in}}(t)\}\right) - \left(\sum_{d_m \in D_{\text{in}}(t)} \text{Size}_m - \max\left\{\{\text{Size}_k \mid d_k \in D_{\text{in}}(t),\ k \neq i \neq j\} \cup \{\text{Size}_i + \text{Size}_j\}\right\}\right). \quad \text{......(iii)}$$

Case 2: (Replication of data set) the proposed technique has been introduced with existing placement strategy. It is impossible to satisfy all the conditions to place the dataset a appropriate where all task can access the data with minimum data transfer cost and fulfilment of SLA objectives. In order to do the same following methods will followed for data replication strategies.

Method 1: If the replication cost is less than data transfer cost for multiple transfer for the same dataset. Then replicate the dataset to nearby nodes with optimal number of replications.

Method 2: If the replication cost higher than data transfer, but the SLA has been violated for scientific workflow application than replication has done with appropriate node with minimum data transfer.

Case 3: (datasets with fixed locations) The datasets with fixed position datasets has been collected. If the size of fixed dataset has small and non-fixed data set is larger, and on same node required data again also, replication will be used in this case. If the non-fixed dataset is small than move the data to fixed data set nodes. Dependency derivation is used to generate CM matrix. Collected the similar values BEA transformation has been applied upon the matrix.

$$PM = \frac{\left(\sum_{i=1}^{p}\sum_{j=1}^{p} c'_{ij} + \sum_{i=p+1}^{n}\sum_{j=p+1}^{n} c'_{ij}\right)}{\left(\sum_{i=1}^{p}\sum_{j=p+1}^{n} c'_{ij}\right)}. \quad \text{.........(iv)}$$

The formula reached to its peak value than division point will be selected. The denominator is representing total dependency.

4.2 DATA DISTRIBUTION

The tree structured cloud server has been modelled. The next step is to allocate the data items to the cloud servers. Highest level of server sub-tree will be allocated to lowest level of data sub tree. It should take care of the storage space.

4.2.1 DATA ALLOCATION

Data allocation will be started from the highest level of sub tree in server farm. That highest level sub tree will allocate to lowest level sub tree in server farm. so number of times, extremely correlated dataset could be assigned to similar node. If because of some limited storage issue, our data item would not stored at similar storage, researcher would placed that data to the nearest nodes .

Algorithm 1 Heuristic based data allocation algorithm
Input: Root of data binary tree (dtNode), Root of server binary tree (ctNode).
Output: Boolean decision for allocation of server tree to data sub-tree.

Step 1: Find the smallest sub-tree in ctNode, which hold the storage capacity greater than dtNode sub-tree.

Step 2: If not able to able find the ctNode, than repeat the step 3 and step 4 till finding of ctNode or till the end of server binary tree.

Step 3: If the ctNode has been find, than allocate the dtNode to root and return true.

Step 4: Otherwise search for the next smallest tree as per the requirement of dtNode.

Step 5: If no allocation has been done than return false.

Algorithm 1: Heuristic based data allocation algorithm

4.2.2 DATA PLACEMENT USING REPLICATION

After data allocation in tree structure researcher places the data by considering success or non success judgements in both top-down and bottom-up selection strategy. If the replication cost is less than data transfer cost for multiple transfer for the same dataset. Then replicate the dataset to nearby nodes with optimal number of replications. After replication data will be placed on different data centers.

Algorithm 2 Heuristic based data placement using replication algorithm (Recursive)
Input: dtNode: Root of data sub-tree ready to be allocated.
ctNode: Root of server sub-tree ready to have the data sub-tree.
Output: Boolean decision for placement sucess of server tree to data sub-tree.

Step 1: Check the ctNode's left and right node, if both are NULL it means current node is leaf node.

Step 2: Check the required space for dtNode, if it is sufficient, repeat step 3 for every node rooted by ctNode.

Step 3: Assign the ctNode number to the dtNode number

Step 4: Update the computation and storage capacity of ctNode and return TRUE;

Step 5: If condition in the step is not satisfied than go to step 6.

Step 6 :Check the dtNode size and left ctNode size, if dtNode size is less than again call the algorithm with new argument, dtNode and ctNode.LeftNode and return true;

Step 7: If the condition is not false for step 6, than check the dtNode size of right ctNode.

Step 8: If dtNode size is less than call data placement recursive function with arguments, dtNode and ctNode.RightNode.

Step 9: Take two bool variables as Lsucess and Rsuccess.

Step 10: Allocate the sub-tree with more storage requirement.

Step 11: Replicate the data required in two binary tree having no dependency.

Step 12: Allocate the remainder sub tree to server sub tree, if success return true.

Step 13: If left and right success is true than return true, otherwise return false.

Algorithm 2: Heuristic based data placement using replication algorithm (Recursive)

CHAPTER 5
RESULTS AND DISCUSSION

5.1 EXPERIMENTAL RESULTS

5.1.1 SIMULATION SETUP

Workflowsim has been used for implementation. This simulator is an open source workflow simulator. It is giving a workflow level support by amplifies cloudsim. Workflow will be modelled using DAG model. In which DAG model demonstrate a model of delay happening in the different levels of Workflow Management System stack, elaborating model of failure of node and task clustering algorithm and the executions of a few most mainstream dynamic and static workflow scheduler Parameters are straightforwardly gained from hints of genuine executions.

5.1.2 APPLICATION MODELING

Different tasks are used in our experiment having different sizes. These are varied from 30 tasks to 1000 tasks. MONTAGE, LIGO, EPIGENOMICS, CYBERSHAKE, INSPIRAL, SIPHT are the main workflows. Some of the workflows are having more makespan time and some are having less makespan time.

- **LIGO** Laser Interferometer Gravitational-Wave Observatory is observing and analysis astrophysical gravitational waves and further incorporates the data in the astronomy and physics research [12].
- **MONTAGE** The montage scientific workflow is used to compute the mosaics of the sky images. The images collected first re-projected as per the coordinates [13].
- **CYBERSHAKE** In the Southern California area a seismology application named CyberShake used to figures Probabilistic Seismic Hazard that are bends for geographic locations [3].
- **EPIGENOMICS** It is an information parallel work process. The Illumina-Solexa Genetic Analyzer gave us a beginning information which is in the form of DNA grouping paths [3].

- **SIPHT** The extensive expectation and comment of RNAs encoding qualities includes an assortment of individual projects that are executed in the correct request utilizing Pegasus [3].

5.1.3 RESOURCE MODELING

Different resources in workflow with different specification have been taken . There are various virtual machine parameters in our workflow.

Table 5.1: Virtual machine parameters

VMM Name	XEN
MIPS	1000
No. Of CPUs	1
VM Memory(RAM)	512 MB
VM Bandwidth	1000
Image Size	1000

All the above specifications influence the performance of the technique directly. In VM, to run a task, MIPS is necessary and the available host's MIPS is observed periodically. The Replication mechanism minimize the makespan time of workflow. Virtual machine is created virtually on the host. To effectively process the cloudlets without any overhead, four types of VM is used with homogeneous configuration. The actual machine that is available in data centres is host.VM memory which is RAM is of 512 MB is used in our setup. Bandwidth and MIPS is considered 1000 in it.

5.1.4 BASELINE ALGORITHM

Heuristic based data placement using Replication Algorithm has been developed. The appropriate data placement strategy reduces the scheduling overhead and cost and makespan time of data processing. Existing strategy are worked on reducing the data movement among the nodes of data centres by clustering of jobs and placement at appropriate data centres. Data size is an important factor in data placement and placement of big data jobs is still a challenge. Coarse grained jobs take much time to transfer and increase cost in case of replication. Transfer the jobs through high speed links increase the performance of data placement. Some datasets are used by many task or many datasets are used by one task. So to increase the performance of data placement strategy replication of dataset can be done, so that data can be available at more than one place and further reduce the data movement among the data centres for scientific workflow applications.

5.2 COMPARISON WITH EXISTING TECHNIQUE

In this section different scenarios have been discussed in which our proposed algorithm works. Replication algorithm improves the makespan time using heuristic based data placement. Three scenarios have been taken for different no. of VMs. In first scenario 2 VMs have been considered. In second scenario 10 VMs have been considered. In last scenario 20 VMs are used. Different scenario gives different results.

5.2.1 SCENARIO 1 (2 VMs)

In first scenario 2 VMs have been taken with different no. of tasks of different workflows. First montage workflow has been taken which is having varied tasks ranging from 50 to 1000. Second epigenomics workflow has been taken which is having tasks 24, 46 and 100.Different tasks shows different results for existing and proposed techniques.

- MONTAGE workflow tasks

Table 5.2: Makespan time comparison of MONTAGE

MONTAGE Tasks	Existing	Proposed
MONTAGE with 25 tasks	187.2	169.32
MONTAGE with 50 tasks	309.34	308.62
MONTAGE with 1000 tasks	5933.69	5867.81

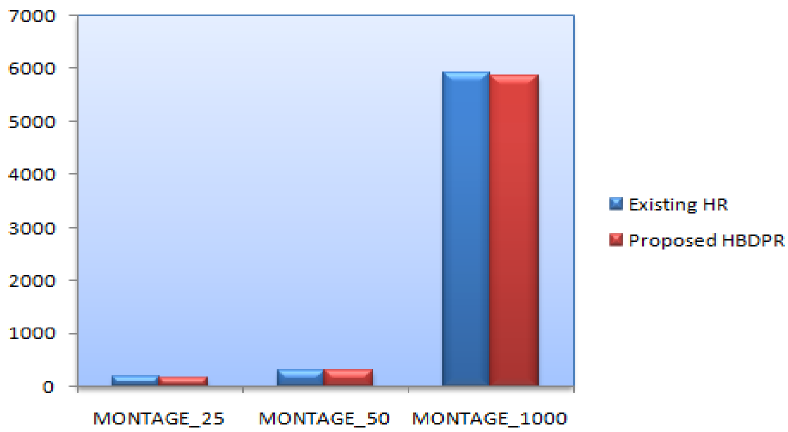

Figure 5.1: Makespan time comparison of MONTAGE

In the Figure 5.1, the MONTAGE tasks for 2 VMs shows existing and proposed results. The performance for MONTAGE tasks for 2 VMs gives better result than existing. Our proposed work have less makespan time than the existing work. The results obtained for proposed scenario is better than existing scenario. Efficiency of proposed algorithm is improved in case of MONTAGE.

- EPIGENOMICS workflow tasks

Table 5.3: Makespan time comparison of EPIGENOMICS

EPIGENOMICS Tasks	Existing	Proposed
EPIGENOMICS with 24 tasks	46386.13	15788.2
EPIGENOMICS with 46 tasks	36205.54	36777.76
EPIGENOMICS with 100 tasks	254792.74	237698.15

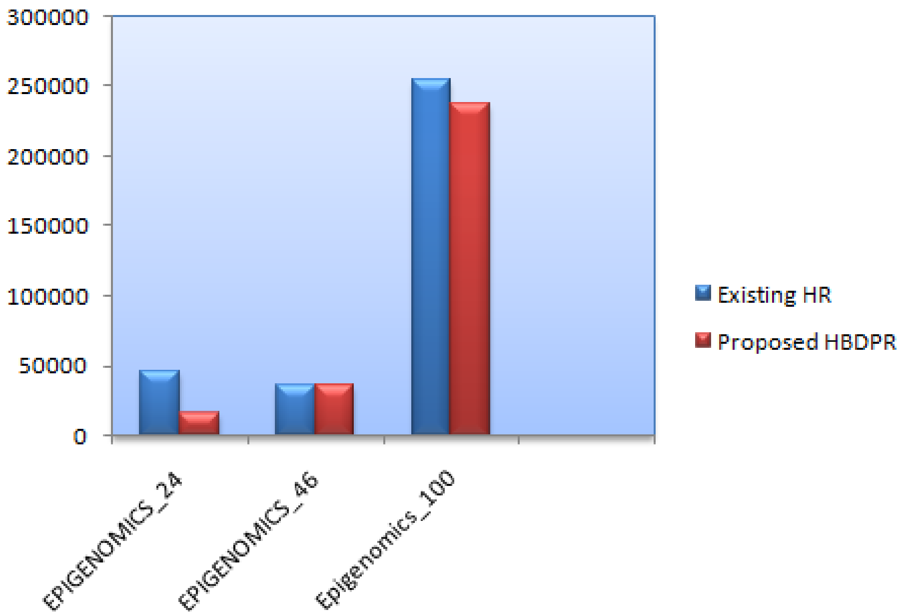

Figure 5.2: Makespan time comparison of EPIGENOMICS

In the above scenario, EPIGENOMICS tasks has been taken to show the difference in existing and proposed technique. In the graph, epigenomics has been shown with different tasks having 2 Virtual Machines. The time difference in epigenomics tasks decreases in proposed scenario as compare to the existing scenario. As the number of tasks increases or decreases our proposed algorithm gives better results. Makespan time improves in proposed algorithm.

5.2.2 SCENARIO 2 (10 VMs)

In second scenario 10 VMs have been taken with different no. of tasks of different workflows.execution will be done by taking the different workflows like montage, sipht, epigenomics and inspiral having different tasks.

- MONTAGE workflow tasks

Table 5.4: Makespan time comparison of MONTAGE

MONTAGE Tasks	Existing	Proposed
MONTAGE with 50 tasks	121.56	136.38
MONTAGE with 100 tasks	191.53	166.06
MONTAGE with 1000 tasks	1525.99	1446.32

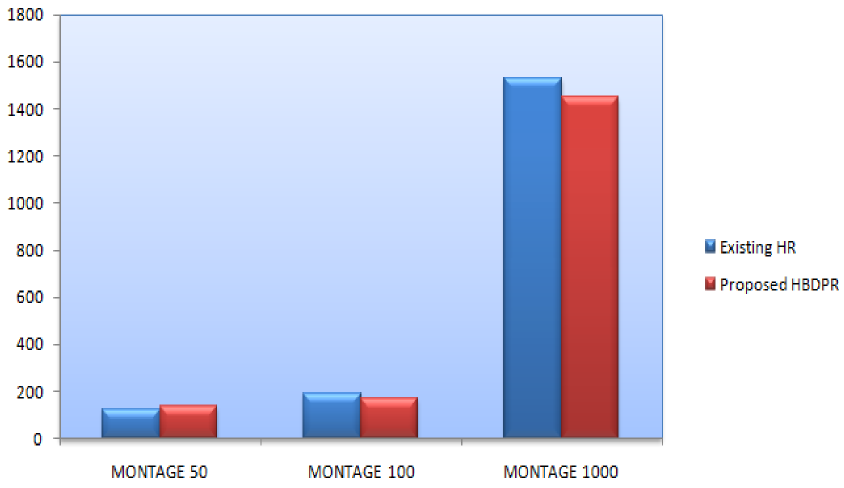

Figure 5.3: Makespan time comparison of MONTAGE

In the above scenario, the Montage task shows existing and proposed results. In this Montage tasks with 10 Virtual Machine has been taken. The time difference alone makes the difference in both existing and proposed scenario. As the number of tasks increases, the proposed algorithm gives better results than existing. In case of Montage 1000, time consumption in proposed algorithm is less as compare to existing algorithm. By decreasing the time, performance has been improved.

- CYBERSHAKE workflow tasks

Table 5.5: Makespan time comparison of CYBERSHAKE

CYBERSHAKE Tasks	**Existing**	**Proposed**
CYBERSHAKE with 30 tasks	240.66	223.73
CYBERSHAKE with 50 tasks	482.3	245.8
CYBERSHAKE with 100 tasks	545.34	590.26

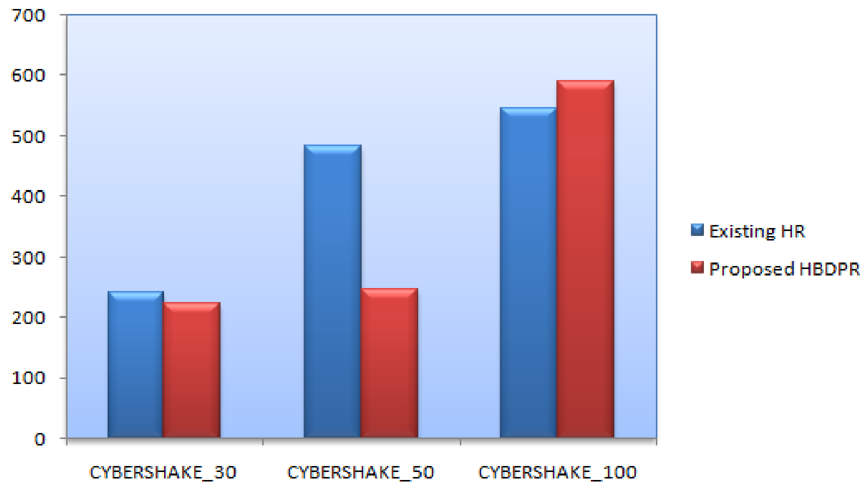

Figure 5.4: Makespan time comparison of CYBERSHAKE

In the above Figure 5.4, the Cybershake tasks shows existing and proposed results. In this Cybershake tasks with 10 Virtual Machine has been taken. The Proposed technique takes advantage over existing technique. This Scenario is beneficial for less number of tasks. The results obtained for proposed scenario is better in case of less number of tasks.

- SIPHT workflow tasks

Table 5.6: Makespan time comparison of SIPHT

SIPHT Tasks	**Existing**	**Proposed**
SIPHT with 30 tasks	5889.17	5884.98
SIPHT with 60 tasks	6360.99	5948.69
SIPHT with 1000 tasks	22710.76	22164.81

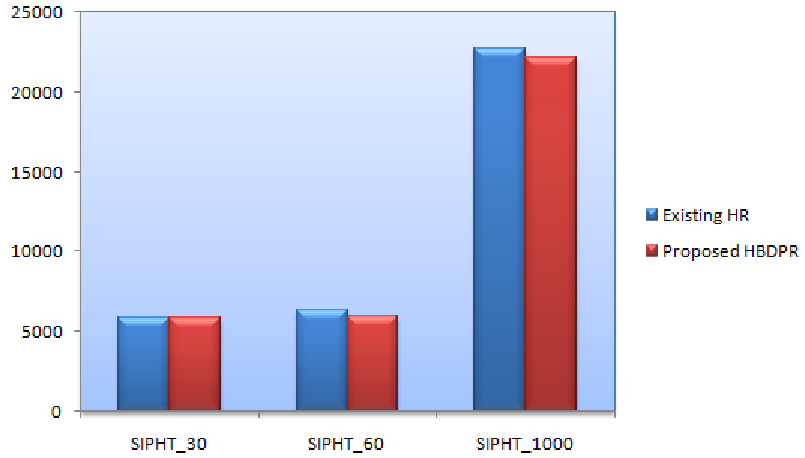

Figure 5.5: Makespan time comparison of SIPHT

In the above scenario, SIPHT has been taken tasks to show the difference in existing and proposed technique. In the graph, SIPHT has been taken with different tasks having 10 Virtual Machines. As the time difference in SIPHT tasks decreases in proposed scenario as compare to the existing scenario. As the number of tasks increases, the proposed algorithm gives better results than existing and improve the efficiency.

- EPIGENOMICS workflow tasks

Table 5.7: Makespan time comparison of EPIGENOMICS

EPIGENOMICS Tasks	**Existing**	**Proposed**
EPIGENOMICS with 24 tasks	9031.09	9969.31
EPIGENOMICS with 46 tasks	16238.66	13767.08
EPIGENOMICS with 100 tasks	88705.57	55592.49

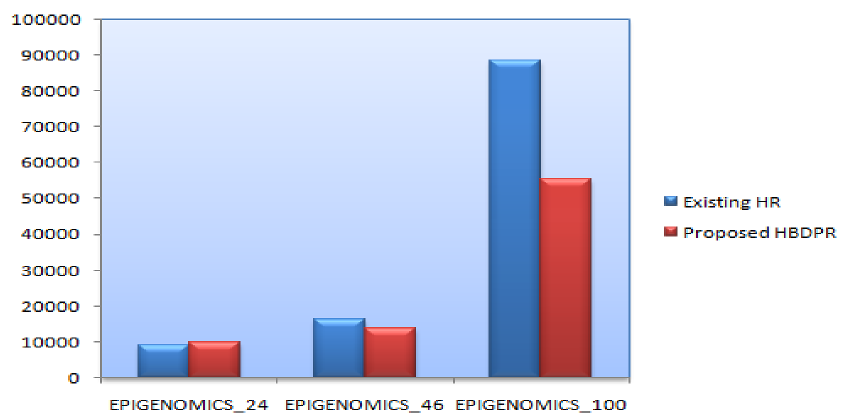

Figure 5.6: Makespan time comparison of EPIGENOMICS

In the above Figure 5.6, the EPIGENOMICS tasks shows existing and proposed results. In this Epigenomics tasks have been taken with 10 Virtual Machine. The Proposed technique takes advantage over existing technique. This Scenario is beneficial for more number of tasks. The results obtained for proposed scenario is better in case of less number of tasks. By decreasing the makespan time, performance has been improved in proposed work.

- INSPIRAL workflow tasks

Table 5.8: Makespan time comparison of INSPIRAL

INSPIRAL Tasks	Existing	Proposed
INSPIRAL with 30 tasks	3038.55	3038.55
INSPIRAL with 50 tasks	2146.1	1893.38
INSPIRAL with 1000 tasks	24836	23403.12

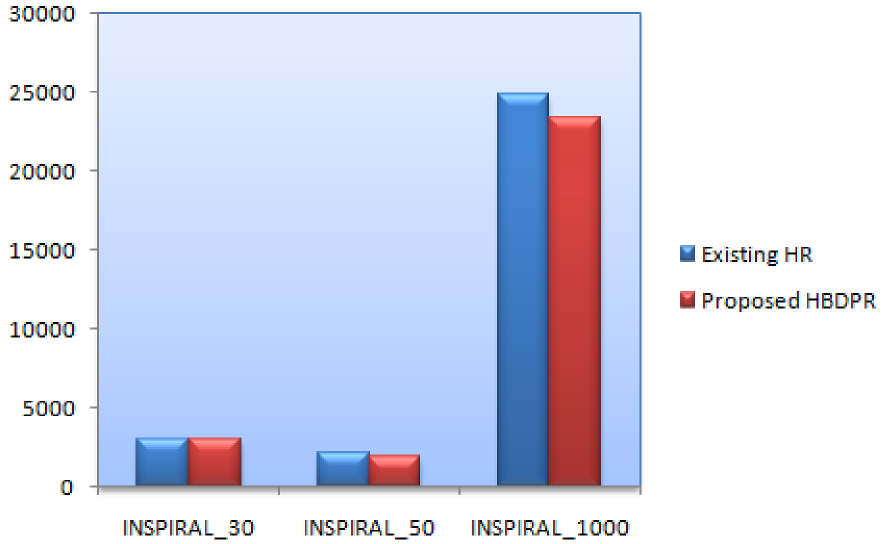

Figure 5.7: Makespan time comparison of INSPIRAL

In above figure 5.7, INSPIRAL tasks has been taken to show the difference in existing and proposed technique. In the graph, INSPIRAL performance has been shown with different tasks having 10 Virtual Machines. As the Makespan time in INSPIRAL tasks decreases in proposed scenario as compare to the existing scenario. As the number of tasks increases, the proposed algorithm gives better results than existing and improves the efficiency.

5.2.3 SCENARIO 3 (20 VMs)

In Third scenario 20 VMs have been taken with different no. of tasks of different workflows. As number of VMs increase performance of each workflow will also improved.

- MONTAGE workflow tasks

Table 5.9: Makespan time comparison of MONTAGE

MONTAGE Tasks	Existing	Proposed
MONTAGE with 50 tasks	138.66	94.34
MONTAGE with 100 tasks	166.45	127.82
MONTAGE with 1000 tasks	950.42	906.49

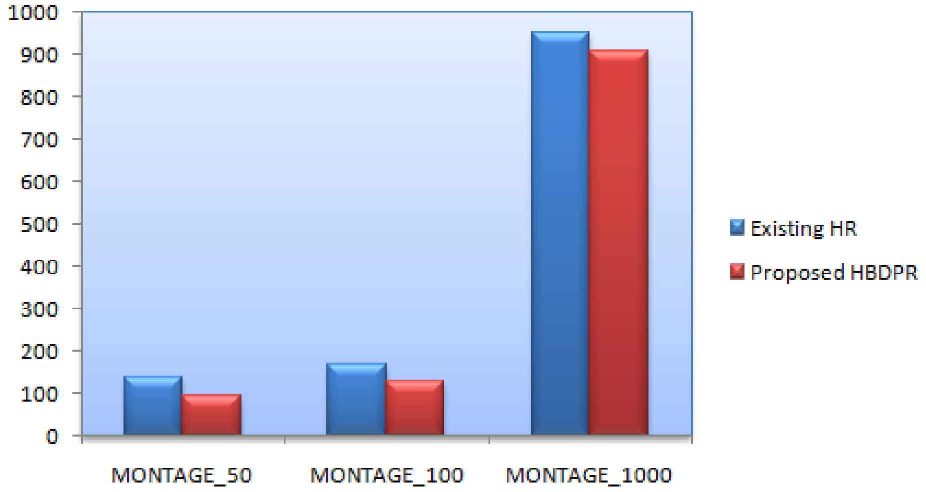

Figure 5.8: Makespan time comparison of MONTAGE

In above scenario, 20 VMs are used for MONTAGE tasks. There is drastic change in performance of existing and proposed work. Our proposed work gives better result in increasing VMs than existing work. Makespan time decreases in case of our proposed work. Better efficiency obtained in proposed work.

- SIPHT workflow tasks

Table 5.10: Makespan time comparison of SIPHT

SIPHT Tasks	Existing	Proposed
SIPHT with 30 tasks	5887.18	5884.98
SIPHT with 60 tasks	5953.69	5948.69
SIPHT with 1000 tasks	11967.64	11462.94

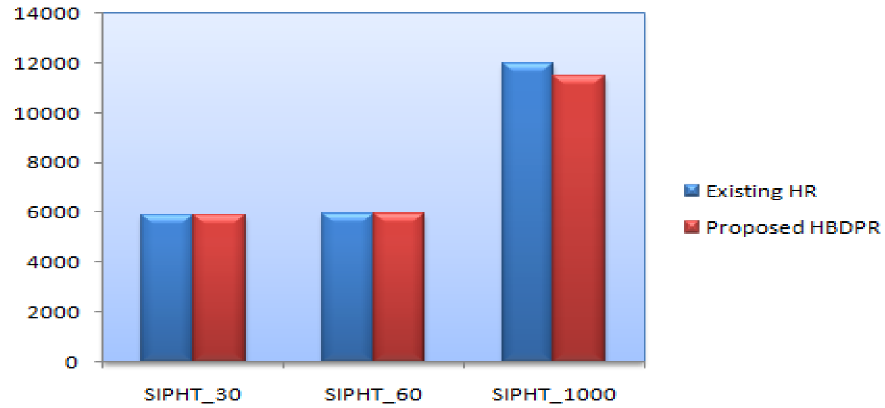

Figure 5.9: Makespan time comparison of SIPHT

In the above scenario, SIPHT tasks has been taken to show the difference in existing and proposed technique. In the graph, SIPHT has been shown with different tasks having 20 Virtual Machines. The Makespan time in SIPHT tasks decreases in proposed scenario as compare to the existing scenario. As the number of tasks increases, the proposed algorithm gives better results than existing and improves the efficiency.

On summarization, researcher can see that in every scenario the performance is improved in every case with different condition. In scenario first the performance of proposed technique is improved with makespan difference with less percentage. But as per increasing the VMs and increasing the tasks our performance gives better result than existing technique. After observing the different scenarios, it is observed that proposed technique is better in case of increasing tasks and increasing no. of VMs than existing technique. The makespan time improves in case of our proposed technique. Proposed technique's performance is better than existing technique's performance in case of different scenarios.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In this research work, replication of datasets has been introduced with existing data placement strategy. It is impossible to satisfy all the conditions to place the datasets at appropriate position where all task can access the data with minimum data transfer cost and fulfilment of SLA objectives. Some datasets are used by many task or many datasets are used by one task. considering replication of task performance has improved. The proposed performance improves effectively with large number of jobs and VMs.

6.2 FUTURE SCOPE

In proposed work, researchers replicate the datasets for decreasing the makespan time. The appropriate data placement strategy reduces the scheduling overhead and cost of data processing. Existing strategy has worked on reducing the data movement among the nodes of data centers by clustering of jobs and placement at appropriate data centers. But in proposed technique as per increasing the VMs and workflow tasks, our performance gives better result than existing technique. After observing the different scenarios, it is observed that proposed technique is better in case of increasing tasks and increasing no. of VMs than existing technique. The makespan time improves in case of our proposed technique. In future, this technique can be extended by using the cache memory for storing the replicated data which further can be reused for improving the performance upto the next level.

REFERENCES

[1] Krutz RL, Vines RD. Cloud security: “A comprehensive guide to secure cloud computing”. *Wiley Publishing;* 2010 Aug 9.

[2] Bulla CM, Bhojannavar SS, Danawade VM. “Cloud Computing: Research Activities and Challenges”. *International Journal of Emerging Trends & Technology in Computer Science.* 2013 Sep;2(5).

[3] Juve G, Chervenak A, Deelman E, Bharathi S, Mehta G, Vahi K. “Characterizing and profiling scientific workflows*”. Future Generation Computer Systems*. 2013 Mar 31;29(3):682-92.

[4] Altintas I, Berkley C, Jaeger E, Jones M, Ludascher B, Mock S. Kepler: “an extensible system for design and execution of scientific workflows”. *InScientific and Statistical Database Management, 2004. Proceedings. 16th International Conference* on 2004 Jun 21 (pp. 423-424). IEEE.

[5] Mann T, Hisgen A, Swart G. “An algorithm for data replication”. *In DIGITAL SYSTEMS RESEARCH CENTER TECH*. REP 1989.

[6] Benoit A, Hakem M, Robert Y. Fault tolerant scheduling of precedence task graphs on heterogeneous platforms. InParallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on 2008 Apr 14 (pp. 1-8). IEEE.

[7] Chervenak A, Deelman E, Livny M, Su MH, Schuler R, Bharathi S, Mehta G, Vahi K. “Data placement for scientific applications in distributed environments”. *InProceedings of the 8th IEEE/ACM International Conference on Grid Computing 2007 Sep 19 (pp. 267-274). IEEE Computer Society.*

[8] Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. “A view of cloud computing”. *Communications of the ACM*. 2010 Apr 1;53(4):50-8.

[9] Mell PM, Grance T. Sp 800-145. “the nist definition of cloud computing”.

[10] Buyya R, Broberg J, Goscinski AM, editors. “Cloud computing: Principles and paradigms”. John Wiley & Sons; 2010 Dec 17.

[11] B. Lud¨ascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee,

J. Tao, and Y. Zhao, "Scientific workflow management and the kepler system," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 10, pp. 1039–1065, 2006.

[12] Althouse WE, Zucker ME. LIGO: "The laser interferometer gravitational-wave observatory". *Science*. 1992 Apr 17;256(5055):325.

[13] Deelman E, Singh G, Livny M, Berriman B, Good J. "The cost of doing science on the cloud: the montage example". *InProceedings of the 2008 ACM/IEEE conference* on Supercomputing 2008 Nov 15 (p. 50). IEEE Press.

[14] A Vouk M. "Cloud computing–issues, research and implementations". *CIT. Journal of Computing and Information Technology*. 2008 Dec 30;16(4):235-46.

[15] Davidson SB, Boulakia SC, Eyal A, Ludäscher B, McPhillips TM, Bowers S, Anand MK, Freire J. "Provenance in scientific workflow systems". *IEEE Data Eng. Bull*. 2007 Dec;30(4):44-50.

[16] Casas I, Taheri J, Ranjan R, Wang L, Zomaya AY. "A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems". *Future Generation Computer Systems*. 2016 Jan 6.

[17] Juve G, Deelman E, Vahi K, Mehta G, Berriman B, Berman BP, Maechling P. "Scientific workflow applications on Amazon EC2". *In E-Science Workshops, 2009 5th IEEE International Conference* on 2009 Dec 9 (pp. 59-66). IEEE.

[18] Yu J, Buyya R. "A taxonomy of workflow management systems for grid computing*". Journal of Grid Computing*. 2005 Sep 1;3(3-4):171-200.

[19] Li W, Yang Y, Yuan D. "A novel cost-effective dynamic data replication strategy for reliability in cloud data centres". *In Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference* on 2011 Dec 12 (pp. 496-502). IEEE.

[20] Dean A. "Automating Hardware to Software Migration for Real-Time Embedded Systems".

[21] Cirne W, Brasileiro F, Paranhos D, Góes LF, Voorsluys W. "On the efficacy, efficiency and emergent behavior of task replication in large distributed systems. Parallel Computing". 2007 Apr 30;33(3):213-34

[22] Hashimoto K, Tsuchiya T, Kikuno T. "Effective scheduling of duplicated tasks for fault tolerance in multiprocessor systems". *IEICE Transactions on Information and Systems.* 2002 Mar 1;85(3):525-34.

[23] Dogan A, Ozguner R. LDBS: A duplication based scheduling algorithm for heterogeneous computing systems. InParallel Processing, 2002. Proceedings. International Conference on 2002 (pp. 352-359). IEEE.

[24] G. Kandaswamy, A. Mandal, and D. A. Reed, "Fault tolerance and recovery of scientific workflows on computational grids, in Cluster Computing and the Grid", 2008. *CCGRID'08. 8th IEEE International Symposium* on, pp. 777–782, IEEE, 2008

[25] Barrass TA, Wu Y, Semeniouk IN, Bonacorsi D, Newbold D, Tuura L, Wildish T, Charlot C, De Filippis N, Metson S, Fisk I. "Software agents in data and workflow management". *CERN*; 2004 Nov 1.

[26] Chun BG, Dabek F, Haeberlen A, Sit E, Weatherspoon H, Kaashoek MF, Kubiatowicz J, Morris R. "Efficient Replica Maintenance for Distributed Storage Systems". *In NSDI* 2006 May 8 (Vol. 6, pp. 4-4).

[27] Xie J, Yin S, Ruan X, Ding Z, Tian Y, Majors J, Manzanares A, Qin X. "Improving mapreduce performance through data placement in heterogeneous hadoop clusters". *In Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium* on 2010 Apr 19 (pp. 1-9). IEEE.

[28] Yuan D, Yang Y, Liu X, Chen J. "A data placement strategy in scientific cloud workflows". *Future Generation Computer Systems.* 2010 Oct 31;26(8):1200-14.

[29] Liu X, Ni Z, Wu Z, Yuan D, Chen J, Yang Y. "A novel general framework for automatic and cost-effective handling of recoverable temporal violations in scientific workflow systems". *Journal of Systems and Software.* 2011 Mar 31;84(3):492-509.

[30] Yang Y, Jiang Y, Liu X, Chen J. "Preventing Temporal Violations in Scientific Workflows": Where and How.

[31] Nukarapu D, Tang B, Wang L, Lu S. “Data replication in data intensive scientific applications with performance guarantee”. *IEEE Transactions on Parallel and Distributed Systems.* 2011 Aug;22(8):1299-306.

[32] Chen W, Deelman E. Workflowsim: “A toolkit for simulating scientific workflows in distributed environments”. *InE-science (e-science), 2012 IEEE 8th International Conference* on 2012 Oct 8 (pp. 1-8). IEEE.

[33] Abbes W, Kechaou Z, Alimi AM. “A New Placement Optimization Approach in Hybrid Cloud Based on Genetic Algorithm”. *In e-Business Engineering (ICEBE), 2016 IEEE 13th International Conference* on 2016 Nov 4 (pp. 226-231). IEEE.

[34] Wan C, Wang C, Pei J. “A QoS-awared scientific workflow scheduling schema in cloud computing”. *InInformation Science and Technology (ICIST), 2012 International Conference* on 2012 Mar 23 (pp. 634-639). IEEE.

[35] Wright P, Sun YL, Harmer T, Keenan A, Stewart A, Perrott R. “A constraints-based resource discovery model for multi-provider cloud environments”. *Journal of cloud computing: advances, systems and applications*. 2012 Dec 1;1(1):6.

[38] Wu Z, Liu X, Ni Z, Yuan D, Yang Y. “A market-oriented hierarchical scheduling strategy in cloud workflow systems”. *The Journal of Supercomputing*. 2013 Jan 1:1-38.

[39] Hoffa C, Mehta G, Freeman T, Deelman E, Keahey K, Berriman B, Good J. On the use of cloud computing for scientific workflows. IneScience, 2008. eScience'08. IEEE Fourth International Conference on 2008 Dec 7 (pp. 640-645). IEEE.

[40] Wang J, Shang P, Yin J. DRAW: “A new Data-gRouping-AWare data placement scheme for data intensive applications with interest locality”. *InCloud Computing for Data-Intensive Applications 2014 (pp. 149-174). Springer New York.*

[41] Wu CQ, Lin X, Yu D, Xu W, Li L. End-to-end delay minimization for scientific workflows in clouds under budget constraint. IEEE Transactions on Cloud Computing. 2015 Apr 1;3(2):169-81.

[42] Poola D, Garg SK, Buyya R, Yang Y, Ramamohanarao K. “Robust scheduling of scientific workflows with deadline and budget constraints in clouds”. *In Advanced*

Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on 2014 May 13 (pp. 858-865). IEEE.

[43] Calheiros RN, Buyya R. "Meeting deadlines of scientific workflows in public clouds with tasks replication*"*. *IEEE Transactions on Parallel and Distributed Systems*. 2014 Jul;25(7):1787-96.

[44] Ghafarian T, Javadi B. "Cloud-aware data intensive workflow scheduling on volunteer computing systems". *Future Generation Computer Systems*. 2015 Oct 31;51:87-97.

[45] Zhao Q, Xiong C, Zhao X, Yu C, Xiao J. "A data placement strategy for data-intensive scientific workflows in cloud. In Cluster, Cloud and Grid Computing (CCGrid)", *2015 15th IEEE/ACM International Symposium* on 2015 May 4 (pp. 928-934). IEEE.

[46] Zhao Y, Li Y, Raicu I, Lu S, Lin C, Zhang Y, Tian W, Xue R. "A service framework for scientific workflow management in the cloud". *IEEE Transactions on Services Computing*. 2015 Nov 1;8(6):930-44.

[47] Chen W, da Silva RF, Deelman E, Fahringer T. "Dynamic and fault-tolerant clustering for scientific workflows". *IEEE Transactions on Cloud Computing*. 2016 Jan 1;4(1):49-62.

[48] Ding J, Zhang Z, Ma RT, Yang Y. "Auction-based cloud service differentiation with service level objectives". *Computer Networks*. 2016 Jan 15;94:231-49

[49] Zhao Q, Xiong C, Wang P. "Heuristic Data Placement for Data-Intensive Applications in Heterogeneous Cloud". *Journal of Electrical and Computer Engineering*. 2016 May 12;2016.

[50] Zahariev "A. Google app engine". *Helsinki University of Technology*. 2009 Apr 27:1-5.