DOI 10.12694/scpe.v20i2.1530
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 2, pp. 377–398. http://www.scpe.org © 2019 SCPE

DATA PLACEMENT IN ERA OF CLOUD COMPUTING: A SURVEY, TAXONOMY AND OPEN RESEARCH ISSUES

AVINASH KAUR [*], POOJA GUPTA [†], MANPREET SINGH [‡] AND ANAND NAYYAR [§]

Abstract. In cloud computing, data placement is a critical operation performed as part of workflow management and aims to find the best physical machine to place the data. It has direct impact on performance, cost and execution time of workflows. Number of data placement algorithms is designed in cloud computing environment that aimed to improve various factors affecting the workflows and their execution including the movement of data among data centers. This paper provides a complete survey and analyses of existing data placement schemes proposed in literature for cloud computing. Further, it classifies data placement schemes based on their assess capabilities and objectives. Further objectives and properties of data placement schemes are compared. Finally future research directions are provided with concluding remarks.

Key words: Data placement, Cloud computing, workflow, replication

AMS subject classifications. 68M14, 68M20

1. Introduction.

With the increase in involvement of data in scientific workflows , the need for high end computing and large amount of storage increses. The processing of data is a very complex task.For a scientific workflow relatively tremendous data is to be stored in different data centers [1]. Data centers are facing unpredictable amount of visitors with the generation of data and development of Internet.This leads to the explosive increase in storage capacity [2]. The large amount of scientific data is accumulated by various fields of research such as meteorology, astronomy and bio-informatics. Processing of large scale data and obtaining valuable scientific discoveries is a tedious job.This lead to the rising need of high performance computing. [3]. This high performance computing,also known as cloud computing entails the need of mass storage resources.

The prime process in a cloud computing environment is during the implementation and execution of data intensive applications for meeting the demand of large volumes of data and storing the data at appropriate data center .This case arises the need of data placement strategy [4]. Data placement includes all activities related to movement such as replication,staging,transfer, space de-allocation and space allocation, unregistering and registering meta data, retrieving and locating the data.

There arises a question how the data placement is performed on cloud computing as in Figure 1.1.

The clients work on a data, evaluates it and passes it to the server.The server processes the request of the client.This is to and fro operation and at the end final output is to be stored on the storage device.This storage of output onto a storage device is decided by the data placement strategy.

1.1. Data Placement in cloud.

In the cloud computing era ,storage of data achieves terabytes magnitude scale ,diverse and complex data structures, high requirement of type and level of service have induced great pressure to data management.Cloud systems work on two kinds of applications ,data-intensive and compute intensive .These are handled concurrently in the systems and generates massive ammount of output/intermediate data.The cloud architecure that handles data and workflows is as in Figure 1.1.

The cloud architecture is classified into four layer architecture as shown in Figure 1.1.

- **Application layer:** In this layer the workflows are constructed and requirements are submitted by graphical and textual user interfaces to the system. The re-usability and flexibility of cloud service components and local resource component is provided by modular programming. The workflows are then changed to predefined mathematical models.These models are used in the next layer. This layer makes available many visualizations and presentation functions for calculating the results of applications.
- Service Layer: The scientific workflows are executed in this layer. For proper functioning of workflow management system , it deals with the fault tolerance and monitoring of workflows. The requirements

[*] Lovely Professional University, Phagwara, India (avinash.14557@lpu.co.in).
[†] Lovely Professional University, Phagwara, India (pooja.19580@lpu.co.in).
[‡] Guru Nanak Dev Engineering College, Ludhiana, India (pmpreet78@gmail.com).
[§] Graduate School, Duy Tan University, Da Nang, Viet Nam (anandnayyar@duytan.edu.vn).

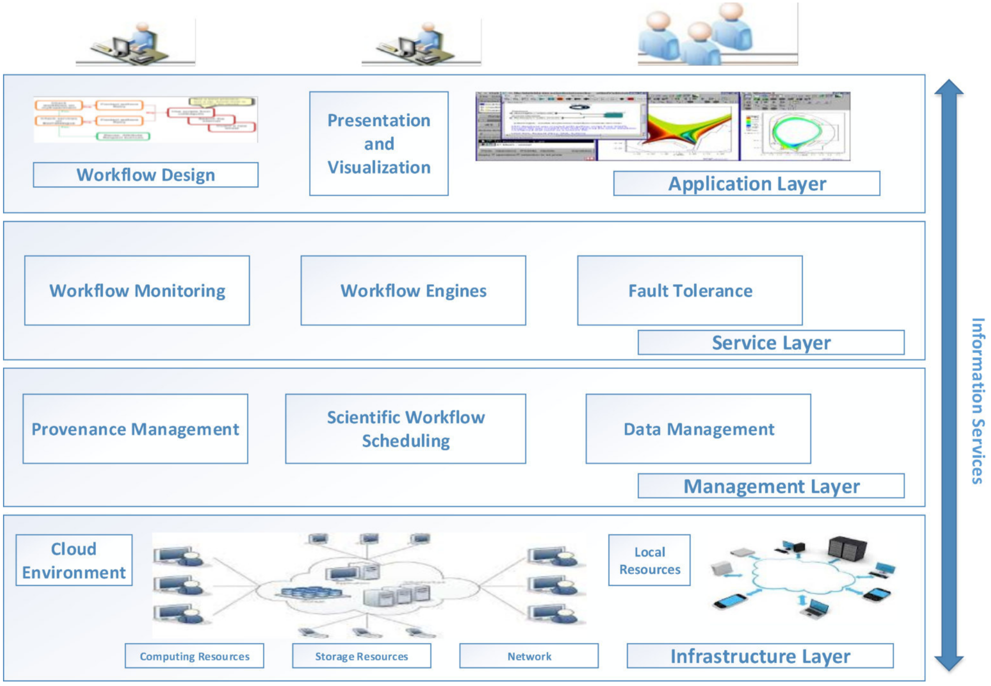

Fig. 1.1. *Workflow management system.*

are received from application layer.Then, the scheduling information is obtained from management layer. The separation of execution of the workflow from the task scheduling and task acquisition leads to the improvement in scalability and stability of the management system.

- Management Layer: It is the bridge of workflow execution and physical resources. The major part of this layer is module of scheduling the scientific workflows that determines which scheduling algorithm to use and it also provided different strategies for scheduling. It is responsible in optimizing the task procedures and parallelizing the processing of data. Before planning the scheduling it obtains the resources and application information about the available services of system. The movement of data is also involved during the scheduling process which is then needed to be handled by efficient data management module.
- Infrastructure Layer: It is responsible for expansion of forgoing services platform which is designed by adding local resources based on the cloud environment. If the local resources are used more ,then it saves the costs. Each computing environment includes network,storage, computing, etc. [5]

1.2. Need of Data Placement.

- Management of budget for storage: Some scientists perform their task without considering the budget whereas there are some that consider budget as one of the QoS parameter.So, it depends upon vendor to vendor.These kinds of vendors prefer the cloud service providers that adhere to the QoS parameters provided by them [6, 7].
- Effective distribution of data onto a storage device: There is a large variety of storge devices available with different parameters like storage capacity,transmission speed etc. So making the data evenly and considerably distributed among these storage devices is also an important task [4].
- Data placement of data intensive jobs is major challenge in cloud environment. Improper data placement policy increases data movement in the data centre which leads to increased cost and delayed services.
- Placement of runtime datasets is still a challenge in complex scientific workflow. Storage and computation capacity is bottleneck at some situations while transferring the intermediate datasets.

- At best of my knowledge and as per the literature review done, no placement policy considers the cached dataset (e.g. fixed position datasets) which affects the performance at significant level.

1.3. Related surveys. The two researchers [8] and [9] have done an inventive review of literature in the field of data placement. Still in the field of data placement research is persistently growing. For the integration and evaluation of existing research present in the data placement field there lies a need of a methodical survey of literature. This paper presents a survey in methodical form for discovering and evaluating research challenges on the basis of existing research in the field of data placement.

2. Background. With the drastic increase in data , requirements for both commercial and scientific applications are expected to reach several million terabytes scales. The matter of concern is about the increasing I/O needs and also about number of users accessing the same dataset.In various fields like genomics and biomedical more people will be accessing more datasets.The movement of large amount of data for replication and processing becomes the necessity.This brings a problem of reliable and efficient data placement.There is requirement of locating the data,moving the data to the place where it is required ,replicating and staging the data.Then allocating and de-allocating the storage and at the end cleaning up everything is required. As the scheduling and managing of computational and network resources is an important task ,similarly scheduling the data placement activities is also crucial.

2.1. Introduction to Cloud Computing. In 1960 , John McCarthy imagined that computation in certain time would be given as a utility . The acknowledgement of this is Cloud computing.The path of a cloud computing environment is being laid by technologies such as Web Services,Hardware Virtualization,Service Oriented Architecture and Mashups [31].

Cloud computing has emerged from various computing paradigms of distributed computing over the growth of both hardware and internet technology. Seamlessly using the power of virtualization and distribution, cloud today provides the use of applications and services over the internet as if they are being used on the local machine. A cloud computing stage is made out of broadly disseminated set of equipment stage, which is arranged and running assorted software services [32].

Most of us are unaware of the use of cloud, the use of e-mails as in gmail, social networking sites as when a picture is uploaded on facebook we use the services offered by or hosted on a cloud. Users do not need to have any information about the background services and can make a communication with many servers at the same time and also there is communication among servers themselves also. Cloud computing aims in providing services to the users that hosts documents on the internet, to outsource the IT infrastructure.

2.2. Workflow. The Workflow Management Coalition (WfMC) defined workflow as : The automation of a business process, in whole or part during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [33]. By the definition given by WfMC, work processes is a progression of organized exercises and calculations of a business procedure which is a unique depiction of the undertakings to be executed in that business process. They are utilized to join a few diverse computational procedures into a solitary lucid procedure. The business applications can now be seen as perplexing work processes, which comprise of different changes performed on the information fancied in accomplishing the goal.Workflows offer awesome points of interest in isolating capacities from applications and in this manner offering data framework to be segmented based by first arranging them and then incorporating them.

WfMC introduces its reference model in recognizing the interfaces inside of this structure which empower items to operate interactively at an assortment of levels. It characterizes a work management framework and the most critical interfaces of system (see Figure 2.1).

- **Workflow Engine:** A software service that gives the run-time environment with a specific end term goal to make, oversee and execute work process cases.
- **Process Definition:** Specifies the information about the process and the workflows related to it.
- **Workflow Interoperability:** This interface makes interoperability possible between different processes of workflow.
- **Invoked Applications:** Interfaces to bolster cooperation with an assortment of IT applications.
- **Workflow Client Applications:** Interfaces to bolster connection with the client interface.

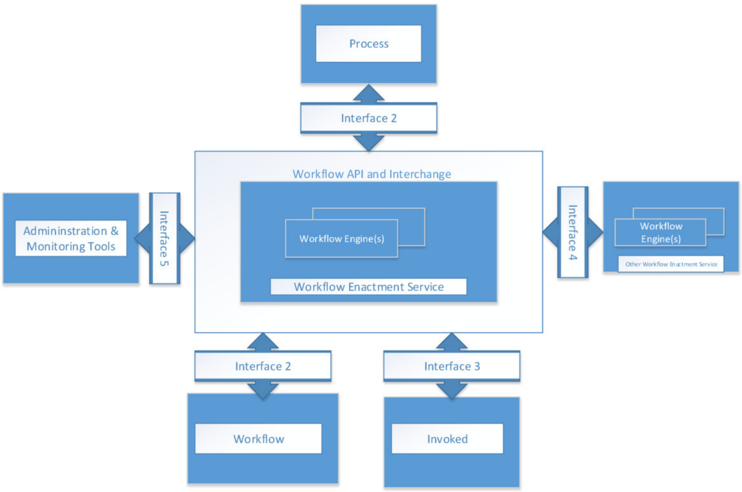
Process
Interface 2
Workflow API and Interchange
Admininstration & Monitoring Tools
Interface 5
Workflow Engine(s)
Workflow Enactment Service
Interface 4
Workflow Engine(s)
Other Workflow Enactment Service
Interface 2
Interface 3
Workflow
Invoked

Fig. 2.1. *WfMC reference model.*

- **Administration and Monitoring:** Interfaces to give a framework observing and metric capacities to encourage the administration of composite work process application situations.

Work process has developed from the idea appropriated programming and with a specific end goal to suite different programming styles and complex shell scripts. Numerous work process frameworks have been created for and productively sent on distributed computing infrastructures that offer high performances such as peer to peer computing,cluster computing and grid computing. Because of the a lot of calculation and information included, the force of the disseminated figuring is required to execute these work processes effectively.

Workflows have found their use in scientific applications,analysis of complex applications and optimization of the execution time for similar applications. In specific conditions researcher might do not have the earlier learning to execute and run the applications, or they do not have the required asset frameworks. The accessibility of the assets and their heterogeneity makes it perplexing and hard to be taken care of by an experimental application.Such application workflows require high performance computing infrastructures for executing data and compute intensive activities. To accommodate such requests large, expensive computing infrastructure are bought and installed. Their integration and maintenance involves huge costs, still the scalability of the resources is a bigger problem as the resources might not be available during the peak hours and they cannot be utilized by others during the off peak periods.

Such issues are not covered by cluster and grid computing paradigms, but is handled effectively by Clouds. Cloud offers a resource pool which can dynamically scaled and managed for delivering high computing power and performance. The services are provided on demand of the users over the internet depending on their deployed applications.

2.3. Workflow lifecycle. The phases in the life cycle of workflow are design, partitioning or clustering ,mapping and enactmentas shown in Figure 2.2.

In the design phase of workflow scientific or business workflows are designed by executable languages. In the initial stage of creation of workflow with the definition of goals, high level workflow model is designed. In a decentralized architecture model is deployed and process of clustering or partitioning is defined. The method of partitioning creates fragments of workflow by grouping elements such as control flows,data flows,activities etc. In clustering, the different tasks are clustered together at the same level, horizontal level or vertical level. In the mapping phase, these clustered or partitioned tasks are deployed in virtual machines or the engines of workflow. The enactment phase of workflow us responsible for implementation of clustered tasks or partitions. In this research, the focus is on mapping phase of workflow.

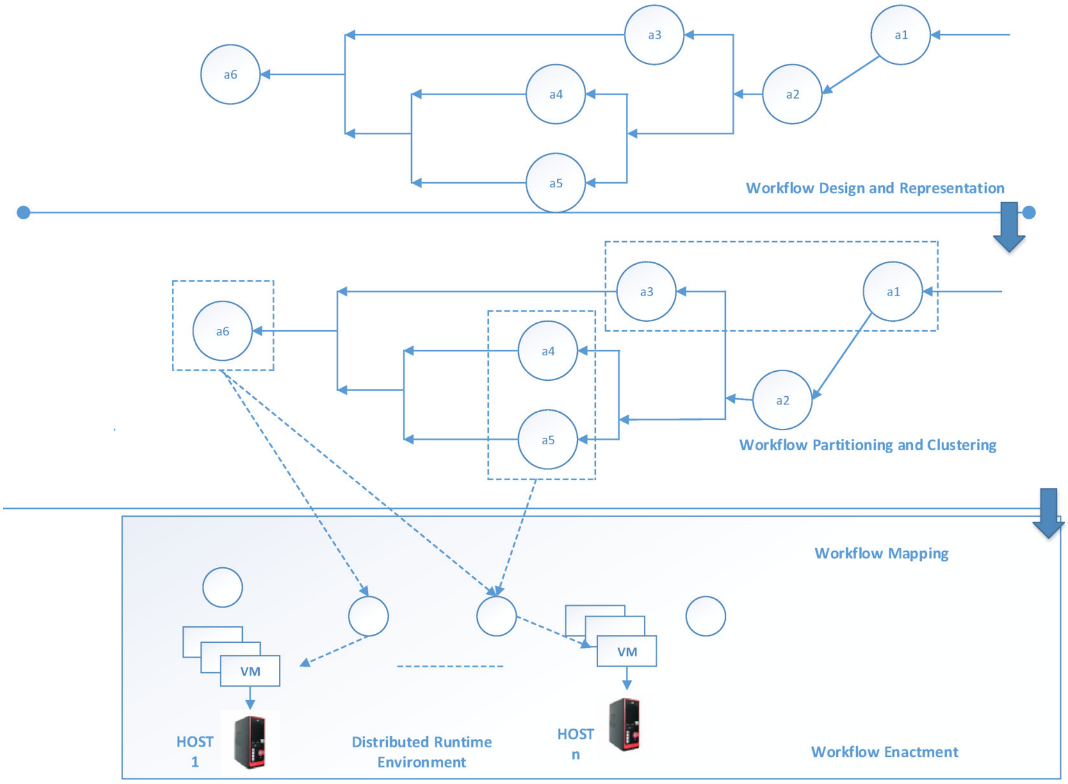
a1
a2
a3
a4
a5
a6
Workflow Design and Representation
Workflow Partitioning and Clustering
Workflow Mapping
VM
HOST 1
Distributed Runtime Environment
HOST n
Workflow Enactment

Fig. 2.2. *Workflow lifecycle.*

2.3.1. Workflow design phase. It is one of the important phase of the lifecycle of workflow. It is divided into Workflow model composition ,application type and interdependency model as in Figure 2.3.

Workflow Model composition. This is responsible to enable user of workflow provide the dependencies and steps in the workflow. The tasks or jobs in the workflow are defined as different steps and the order in which these steps can be executed is defined as dependency. The model composition of workflow can be accomplished using either graph based or language based modeling [55]. Extensible markup language(XML) is used to express the workflow in language based modeling. The control flow is defined by using block structures similar as in C language, such as while and if.So memorizing the syntax of language is required.It becomes tedious task for the users to express complex or large workflow manually using language based modeling.

The control flow of process in graph based modeling is defined using explicit control links between activities. In this workflow is defined graphically using some graph elements.As the graphical representation is easier ,so it is mostly used by the users.The graph based modeling can be further categorized in non-directed acyclic graph(NDAG) or directed acyclic graph(DAG) [56]. In a DAG, the workflow is specified as

$$W = (Ta, E) \; where \; Ta = ta_1, ta_2, ..., ta_n \tag{2.1}$$

is a finite set of tasks specified by the vertices's in DAG and E specifies its existing edges. The dependency of data and communication is represented by an edge [57].In a DAG based workflow ,control patterns are complemented by non_DAG iteration structure that initializes cycle or loop. The scientific application comprises the iteration structure and it requires cyclic execution of tasks. There are different workflow models [55].

Workflow Application Types.

1. Data/Computation intensive workflows: Scientific workflows are responsible for the visualization of complex scientific computations.These are commonly used in e-science technology and applications.

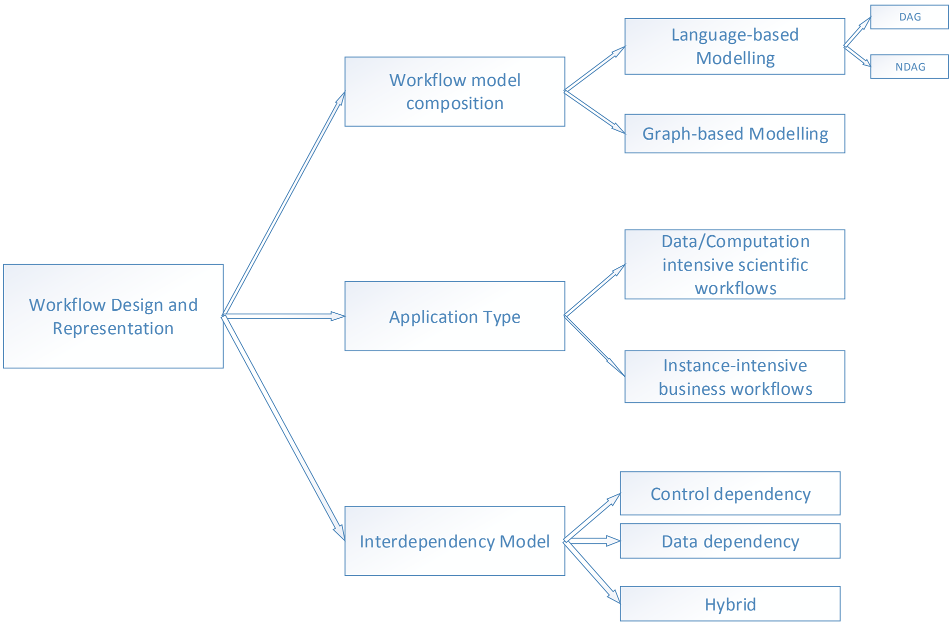
Workflow Design and Representation
Workflow model composition
Language-based Modelling
DAG
NDAG
Graph-based Modelling
Application Type
Data/Computation intensive scientific workflows
Instance-intensive business workflows
Interdependency Model
Control dependency
Data dependency
Hybrid

Fig. 2.3. *Workflow design phase.*

Various computing resources are involved in scientific workflows in de-centralized environments that is used in various areas of bioinformatics,astronomy etc. [58]

2. Instance-Intensive Business Workflow: The models of workflow are applied in business also. Business workflow composes of documented processes taking place in the routine business of an organization. A new workflow is introduced upon further detailing e-business and e-science application ,that is instance intensive workflow. For example, e-business is one of the example of instance intensive workflows [59].The instance of workflow requires parallel execution [61]. The various examples of e-business are insurance claim, bank check etc. [60]

   The basis for instance intensive workflow is data or computation intensive workflow.For example, scientific workflow for weather forecast can be used as commercial weather forecasting product .

Interdependency Model. The type of dependencies existing in the partitions of workflows is defined as interdependency. Data dependency and control dependency are both used for defining different types of dependencies.The control transfer between the tasks or partitions is defined by a link that is applicable to control structures such as conditionals, loops and sequences. Data flows or dependencies supports data-intensive applications. The data transfer between tasks is specified by data dependency. Hybrid models combines both the data and control dependencies for better management of models of workflow [62].

2.3.2. Workflow partitioning. After the workflow is designed in the life cycle of workflow ,then the workflow is partitioned or clustered into number of partitons. These clustered or partiotioned tasks are further sent to the mapping phase of the workflow.

2.3.3. Workflow Mapping. Workflow mapping is responsible for creation of executable instance of workflow that uses suitable resources as in Figure 2.4. Resource in a workflow is defined as agent/service/computer/ person that can execute task or fragment of workflow.

The mapping of workflow is performed either by the user manually or by a workflow system automatically.In the mapping done by the user each partition is assigned to one execution engine.Kepler and Taverna workflow systems are examples of these kinds of mapping The other kind of mapping is assigning the available resources to partitions/tasks using the internal schedulers. The workflow systems employing this type of mapping are

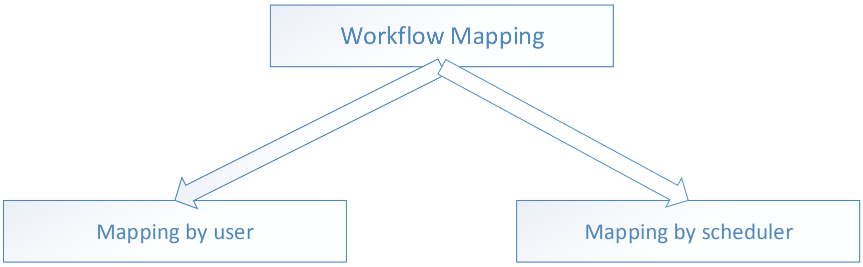

Fig. 2.4. *Workflow Mapping Phase.*

Triana ,Karajan and Pegasus [64]. The grid application toolkit (GAT) is used by Triana for choosing resources at runtime [65]. This research focuses on data placement so it described further in complete detail.

2.3.4. Workflow Enactment. In this phase the fragments or partitions are enacted in Grid or Cloud environment as on figure 6.It is described as a service responsible for executing and managing workflows ,also maintaining coordination between resources and management tools required for execution of workflow [33].The workflow engine is kind of software service that provide executable environment for instance of workflow.Workflow enactment is classified into further two categories Business workflow engines and scientific workflow engines as depicted in Figure 2.5.

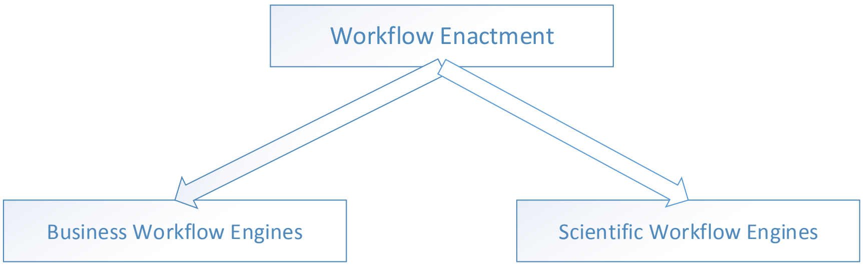

Fig. 2.5. *Workflow Enactment Phase.*

The emergence of Cloud computing introduced economy specific pay-per-use model . The Cloud is the biggest pool of virtual resources such as platform, services and hardware , making workflow management cheaper,easier and faster.There exists many kinds of workflow enactment engines. The enactment engines used in business workflow domain are NINOS [66] , SwinDeW-C(Swinburne decentralized workflow for cloud) [61], etc. and the workflow engines for scientific workflows are Kepler, Taverna etc.

2.4. Data Placement Process. In the computing systems,data placement process is defined as movement of input data of data intensive applications from remote site to the execution site, then a movement of output data from execution to another remote site or same site.To prevent the risk of disk full at the execution site, there is a requirement to allocate space and deallocating the space when job is done.

The data placement steps are presented as DAG (Directed Acyclic Graphs) and dependencies between them are represented by arc. The steps of data placement are as shown in Figure 2.6.

2.4.1. Data Placement Stages.

- Stage-in : This data placement stage is also known as build-time data placement.This involves getting the provenance information of the dataset and pre-clustering the similar data items before uploading the data to the appropriate data center. [13]

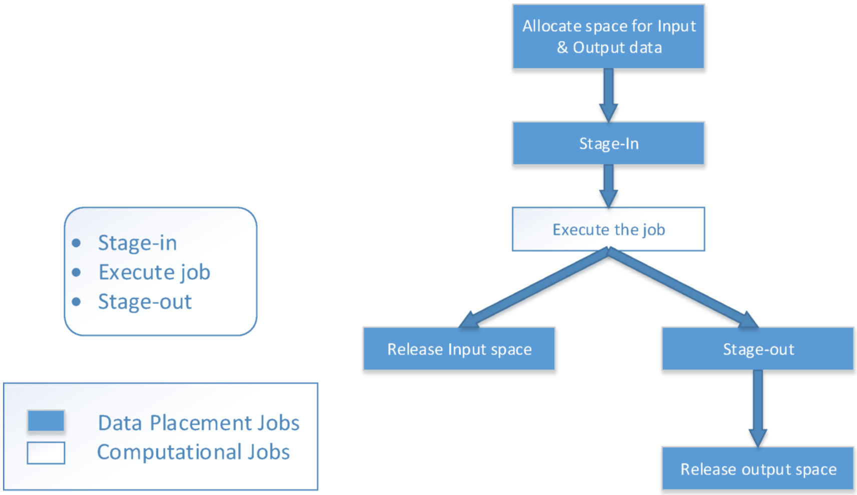

Fig. 2.6. *Data Placement Process.*

- Stage-out: This data placement stage is also known as runtime data placement.During the execution of workflows intermediate data is generated .This data may be the input for the subsequent tasks,so placing and managing this kind of data is again a complicated task. [13]

3. Data Placement Algorithms. The data placement algorithms are classified into different categories of algorithms as shown in Figure 3.1.

3.1. Coorelation Based data placement. A Data placement scheduler ,Stork provides an ability to queue,schedule,manage and monitor the data placement jobs.It also applies the technique of checkpointing the jobs.This approach provides a complete automation for processing the data. The proposed system automate the data transfer fully automatically between the hetrogenous systems.It possesses the ability to recover from network,software and storage system failures without human intervention.It performs dynamic adaption of data placement jobs at the execution time to system environment [34]

The author presents a filecule grouping technique for managing data in science Grids while saving the time for locating the data and grouping the file based on size.Also LRU-bundle an algorithm for file staging is proposed [35]. [36] introduced a BitDew programming interface for data management operations as replication,placement and fault tolerance.This architecture relies on independent services to transfer ,store and schedule the data.

In [29] the investigator proposed an algorithm for run time as well as for build time stage.In the initial stage dependency between all the data sets is calculated and dependency matrix is built.For the partitioning and clustering of data sets BEA(Bond energy algorithm) is used.These partitions are shared among different data centers.These data centers are partitioned using k means algorithm.After the generation of intermediate data ,the newly proposed clustering algorithm deals with new datasets.The dependencies for each data center are judged and then accordingly data is moved.The factors of data movement and gathering of data at one point is covered up.Two algorithms proposed are as follows:

1. Build-Time Stage Algorithm:In this stage dependencies between the tasks is calculated and dependency matrix is built.For the transformation of dependency matrix to clustered dependency matrix ,BEA is used.Global measure(GM) for clustered dependency matrix is defined

$$GM = \sum_{i=1}^{n} \sum_{j=1}^{n} DM_{ij}(DM_{i,j-1} + DM_{i,j+1}) \tag{3.1}$$

Then further partiotined datasets are mapped to data center using binary partiotining algorithm as per

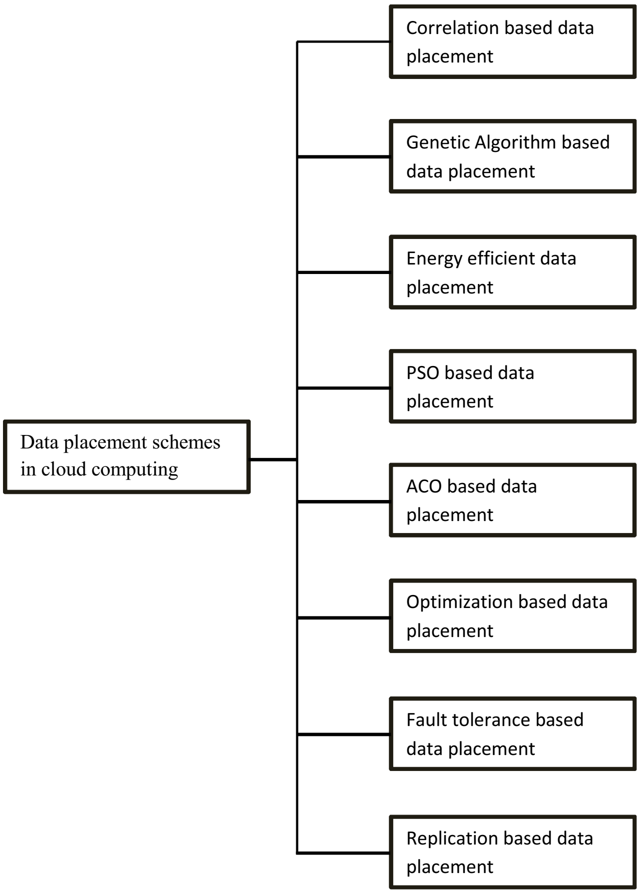
Data placement schemes in cloud computing
Correlation based data placement
Genetic Algorithm based data placement
Energy efficient data placement
PSO based data placement
ACO based data placement
Optimization based data placement
Fault tolerance based data placement
Replication based data placement

Fig. 3.1. *Data Placement Algorithms.*

equation

$$PM = \sum_{i=1}^{p}\sum_{j=1}^{p} CM_{ij} * \sum_{i=p+1}^{n}\sum_{j=p+1}^{n} CM_{ij} - \left(\sum_{i=1}^{p}\sum_{j=p+1}^{n} CM_{ij}\right)^2 \tag{3.2}$$

This process continues recursively.

2. Run-Time Stage Algorithm: In this stage , newly generated datasets are clustered to the data center on the basis of dependency between them using K means algorithm.The dependency for all datasets with all data centers is calculated.At last the dataset placement is performed depending on the maximum dependency on a data center.

In [9] author proposed DCCP(Dynamic Computation correlation placement) as a data placement technique based on dynamic computation correlation.The data sets with highly dynamic correlation are placed at same data center.The factors considered are capacity load and Input/Output load of data centers.During the execution of computations ,data sets processed are stored on local data centers thus leading to the reduction of execution time.This leads to Input/Output and statistical load balancing.

3.2. Genetic Algorithm based data placement. In [30] author proposed algorithm for reduction of movement of data among data centers leading to load balancing in data centers.The heuristic and genetic are combined together for improving the ability to local search and reducing the search time.The heuristic idea is implemented in gene operations and initial population selection.The integer encoding rules are applied and

Table 3.1
Objectives of correlation based data placement schemes

Schemes	**Energy aware**	**Cost aware**	**Resource aware**	**Application aware**	**Factors**
[35]	-	✓	-	-	Data dependency,File Size
[36]	-	✓	✓	✓	Data Transfer
[34]	-	✓	-	✓	Data dependency
[29]	-	✓	-	-	Data dependency ,Data Movemement, Execution Time
[9]	-	✓	-	-	Load balance ,Execution Time

Table 3.2
Properties of coorelation based data placement schemes

Schemes	**SLA support**	**Security**	**Cloud/ Grid**	**Algorithm**	**Clustering Algorithm**	**Tool/Programming Language**
[35]	✓	-	Grid	Greedy Request Value	-	Java
[36]	✓	-	Grid	Information Dispersal	-	Java
[34]	✓	-	Grid	-	-	Stock Server
[29]	✓	-	Cloud	Bond Energy	K means	SwinDeW-C
[9]	-	-	Cloud	DCCP	-	cloud computing application platform (CApp)

placement process is represented by a gene.The ineffective fragments of genes that cannot be coded at data centers is determined by the encoding rule.Fitness function depicts the advantages and disadvantages of genetic individuals.It depicts degree of dependency between data sets and load balance of centers.D_dc is the total degree of data dependency in a data center d_c, which is defined as:

$$D_{d_c} = \sum_{ds_i, ds_j in d_c} Dij \tag{3.3}$$

The fitness function is defined as

$$fitness = \frac{1}{|DC_u|} \left(\sum_{dc \epsilon DC_u} D_{dc} \right) \tag{3.4}$$

where DC_u is data centers set which consist of at least one data set.Larger the value depicts better gene. The overloading of data centers are adjusted by non normal chromosome.If the gene is found to be invalid ,it refers to overloading of data centers.So this gene is to be neglected.If the invalidity is depicted at initial stage then the new gene should be generated.If this occurs at cross stage then genes to be re-crossed and if it occurs at mutation stage then genes to be re-mutated.Optimization of genes is performed for reducing movement of data.If the data set is fixed location data set then calculate the dependency of each data set with data center as per formula .If the size of data set is more then the storage capacity of data center then data set is moved to next data center.

In [28] investigator proposed a mathematical data placement technique on the basis of genetic algorithm.The crossover rate(P_c), size of population (G) and mutation rate(P_m) is determined.After the generation of initial population BG , fitness value for of each individual is obtained. Fitness value of is denoted by $F = 1/\Gamma(B_t)$. At last individuals are selected using roulette wheel selection [25].Crossover and mutation are performed on the selected matrix. [26, 27]. The individuals not adhering to the requirements of storage capacity are abandoned.

[37] also introduced technique of data placement based on a genetic algorithm for reducing the time as well as the cost of data scheduling between centers.

Authors considered the factors of dependency among slices of data and distributed cost of transaction for the placement of data using genetic algorithm.It minimizes the transaction cost while balancing the load.Author

Table 3.3
Objectives of Genetic Algorithm energy based data placement schemes

Schemes	**Energy aware**	**Cost aware**	**Resource aware**	**Application aware**	**Factors**
[30]	-	✓	✓	✓	Data movement,Load balance
[28]	-	✓	✓	-	Storage Capacity,Data Transfer
[37]	-	✓	✓	-	Data dependency,Data movement,Cost

Table 3.4
Properties of Genetic algorithm energy based data placement schemes

Schemes	**SLA support**	**Security**	**Cloud/ Grid**	**Algorithm**	**Clustering Algorithm**	**Tool/Programming Language**
[30]	✓	-	Cloud	Heuristic genetic	-	Cloudsim
[28]	-	-	Cloud	Optimal genetic	-	-
[37]	-	-	Cloud	Hashing	K-Means	-

used roulette wheel for realization model.The fitness function is represented as

$$Fitness = \frac{\sum_{DCT_n} C_{ij}}{|DCT|} \tag{3.5}$$

Where $|DCT|$ are number of data center, DCT_n the data center numbered n and C_{ji} is the total cooperation cost between data sets j and i

3.3. Energy Efficient data placement. In [38] [39] [40] [41] [42] many authors proposed energy saving methods but at hardware level such as processor speed adjustment,voltage settings,enlarging memory, etc.But these methods are unable to reach maximum energy optimization as saving of energy by these methods is comparitively less then turning the computer off.These methods are only limited to a single PC or computer.

In [21]author proposes heuristic algorithm for data placement and two node scheduling techniques in order to save consumption of energy during execution of tasks.In the proposed algorithm data blocks are kept rational for finding minimum set of nodes containing the collection of blocks of data.The goal of energy saving is achieved by turning on minimum nodes required covering the maximum data blocks.Greedy algorithm is employed for covering data block with computing node.The author addresses two goals in it .First is the power consumption upper bound is known,accordingly execution time of task requests by node scheduling is minimized.Second ,the execution time of tasks is known, accordingly execution time of task requests by node scheduling is minimized.This optimization of batch scheduling achieves energy saving effect by providing a solution for time constrained and power restrained issues in the platform of cloud.Each node $s \epsilon S$,in node s, q is the number of data blocks , p is the number of data blocks which meet the job requests , so node cover rate

$$\gamma(s) = \frac{p}{q} \tag{3.6}$$

If node s and node s_t have the same data block replica, then node s and node s_t are data-exchangeable. The formula used to compute resource utilization of nodes is

$$U = e.U_{cpu} + (1 - e).U_{disk} \tag{3.7}$$

where U_{cpu} stands for utilization of cpu ,u_{disk} stands for utilization of disk and e is scale factor. [21]

In [43] investigator proposes Popular Data Concentration (PDC) algorithm .It dynamically migrated mostly used data on the disk to different subset of disks in an array.The purpose is to transfer the load to few disks so that other disks can be sent to mode of energy saving.In [44] author addresses conservation fo energy for cluster of nodes that execute Map Reduce jobs.Reconfiguration of clusters is performed on the basis of current workload.When the average utilization rises, clusters are turned on or off.

Table 3.5
Objectives of Energy efficient based data placement schemes

Schemes	**Energy aware**	**Cost aware**	**Resource aware**	**Application aware**	**Factors**
[21]	✓	-	✓	-	power consumption,Execution time
[43]	✓	✓	-	-	File characterstics,workfload characterstics
[44]	✓	-	-	-	power consumption,cost
[45]	✓	✓	-	✓	Execution cost,power consumption

Table 3.6
Properties of Energy efficient based data placement schemes

Schemes	**SLA support**	**Security**	**Cloud/ Grid**	**Algorithm**	**Clustering Algorithm**	**Tool/Programming Language**
[21]	✓	-	Cloud	Dynamic Algorithm	-	Cloudsim
[43]	-	-	File server	Popular data concentration	-	Execution Driven simulator
[44]	-	-	Grid	Cluster reconfiguration	-	Gridsim
[45]	-	-	Cloud	EnCloud algorithm	-	iVIC with Phython

In [44] author proposed energy saving algorithms that switch off some number of nodes for saving energy.In [45] investigator proposed EnaClous approach that enables dynamic live application placement considering the energy efficieny.In this approach ,virtual machine is used for encapsuation of an application that perform scheduling of an application and live migration of an application for saving energy.Bin packing algorithm is used for apllication placement.The author proposes energy aware heuristic algorithm.For dealing with the varying resource demands ,investigator presented an over provision technique.

In this [46] author proposed an architectural framework based on resource allocation and principles for energy efficient Green cloud computing.It introducesa virtual node placement scheme and decides to turn on some physical nodes of the active virtual nodes, but data placement problem is not considered.

3.4. PSO based data placement. In [**?**] near optimal assignment of tasks in reasonable time is obtained using proposed hybrid particle swarm optimization approach.PSO based algorithm is presented for conquering TAP(Task Assignment Problem).PSO iteration is embedded with hill climbing heuristic for convergence.The factors considered are execution and communication costs.

In [7] author proposed an algorithm based on particle swarm optimization that leads to reduction of cost and execution time of transferring.The investigator works on on-demand method as standard for computing.This refers to paying per on per hour basis with no long term commitments.According to charging standard of Amazon.P_{out_k} is defined as pricing of data center from DC_k. p_{in_t} as pricing of data transfer to DC_l. P_k is the processing pricing of standard on-demand and T_p is the execution time of tasks.The i represents a task executing on data center k.Cost of data processing is given by

$$C_p = T_p \times P_k \tag{3.8}$$

Cost of data transfer is represented as

$$C_t = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l \neq k} x_{ik} \times x_{jl} \times (DT_{ij} \times P_{out_k} + DT_{ij} \times P_{in_l}) \tag{3.9}$$

Fitness function is sum of cost of data transfer and processing represented as

$$C = C_p + C_t \tag{3.10}$$

According to the best fitness value ,data is placed at the data center.

Table 3.7
Objectives of PSO based data placement schemes

Schemes	Energy aware	Cost aware	Resource aware	Application aware	Factors
[?]	-	✓	-	-	Execution and Communication cost
[7]	-	✓	-	-	processing time, transferring time, processing cost, transferring cost
[22]	-	✓	✓	-	Data dependency ,Data Transmission Cost
[17]	-	✓	✓	-	Execution time,Transferring time,Cost
[48]	-	✓	-	-	Execution cost,Communication cost
[8]	-	✓	-	-	Data dependency ,Data Movement

Table 3.8
Properties of PSO based data placement schemes

Schemes	SLA support	Security	Cloud/ Grid	Algorithm	Clustering Algorithm	Tool/Programming Language
[?]	✓	-	-	Hybrid-PSO	-	-
[7]	✓	-	Cloud	PSO	-	Matlab
[22]	✓	-	Cloud	PSO	-	-
[17]	✓	-	Cloud	PSO-SPV	-	Cloudsim
[48]	✓	-	Cloud	PSO	-	-
[8]	-	-	Cloud	PSO	Bond Energy(BEA)	Visual Studio 2010

In [22] author proposed cost effective data placement scheme for multi data centers.During the execution of workflows , the data sets get transferred from one data center to another.So mapping of these data sets to an appropriate data center is the issue addressed by the author. The focus is on reducing the cost of data transfer by considering multiple workflows.Let $DC = \cup_{i=1,2,...|DC|} dc_i$be data center set. G_i is a single workflow. Datasets of multiple workflow(MWS) are represented as $DS = \cup_{i=1,2,...n} ds_i$ Data placement of multiple workflow $MWS = \cup_{i=1,2,..n} G_i$ is represented by

$$M_{multi} = \cup_{i,j=1,2,....,|DS|} d_i \to dc_j \tag{3.11}$$

where $dc_j \epsilon DC \forall d_i \epsilon DS$,$d_i$ is stored in unique data center in M_multi.

In [17] investigator introduced Particle swarm optimization based algorithm embed in SPV(Small Position value) [47].The aim is to optimize task scheduling in cloud computing for minimization of processing cost.The algorithm is based on a small position value.The focus is on reducing execution and transferring time.The fitness function for an algorithm is taken as combination of cost and time $CT = T + C$ [7].

In [48] article investigator presents particle swarm optimization(PSO) based heuristic for scheduling applications to resources of cloud.The factors of data transmission cost and computation cost is considered.The mapping of tasks to resources is performed by varying the computation and communication costs.

In [8] investigator clustered datasets on the basis of dependency using proposed hierarchical clustering based technique.A new factor as size of dataset is introduced.This leads to reduction in data movement ,In the partitioning matrix improved dichotomy algorithm is used.For mapping between data groups and servers PSO based algorithm is used.

3.5. ACO based data placement. In [67], author presents data placement scheme for intermediate data placement by considering the factor of security.The security of data is based on aspects of data integrity,authentication access and data confidentiality. A model of security quantitatively measures services of security provided by data center.Ant Colony Optimization (ACO) based technique is used for dynamic selection of data center.In this the datasets are divided into flexible location and fixed location data sets according to location of stored data.For the evaluation of security services, DDSD (Degree of data security deficiency) is proposed.

Table 3.9
Objectives of ACO based data placement schemes

Schemes	**Energy aware**	**Cost aware**	**Resource aware**	**Application aware**	**Factors**
[67]	-	-	✓	-	data security,data transfer time
[22]	-	✓	✓	-	Data dependency ,Data Transmission Cost
[17]	-	✓	✓	-	Execution time,Transferring time,Cost
[48]	-	✓	-	-	Execution cost,Communication cost

Table 3.10
Properties of ACO based data placement schemes

Schemes	**SLA support**	**Security**	**Cloud/ Grid**	**Algorithm**	**Clustering Algorithm**	**Tool/Programming Language**
[67]	✓	✓	Cloud	ACO	-	Cloudsim
[22]	✓	-	Cloud	PSO	-	-
[17]	✓	-	Cloud	PSO-SPV	-	Cloudsim
[48]	✓	-	Cloud	PSO	-	-

3.6. Optimization based data placement. In the [49] proposed system considers the factors of WAN bandwidth ,data center capacity limit,data interdependency,user mobility and application changes.The proposed approach analyzes the logs submitted by the data centers.It uses iterative optimization algorithm based on access patters of data ,locations of clients.Further ,it outputs recommendations of migration back to cloud service.For scaling large volume of data of data logs ,this technique works in SCOPE [14]. This algorithm works in three phases.Mapping of each client to a set of geographical candidates is performed using weighted spherical mean calculation as per equation

$$wsm((w_i, \vec{x}_i)_{i=1}^{N}) = interp\left(\frac{w_n}{\sum w_i}, \vec{x}_N, wsm(w_i, \vec{x}_i)_{i=1}^{N-1}\right) \tag{3.12}$$

It handles the complexities of data inter-dependencies and shared data.In the second phase ,improvement in the placement is done by applying proposed approach .In the third phase, mapping of data to data center is performed by considering the factor of storage capacity of the data center.

In [68] author aims to place data files into and assigning tasks to the sites of execution for reducing the cost while considering weights.To accomplish this , workflow is modelled as a hypergraph.A heuristic based on hyper-graph partitioning is proposed for generating appropriate placement of data and assignment of task . According to the proposed technique, computational and storage loads are distributed evenly according to some pre-decided ratios.The hypergraph partitioner is implemented by modifying PaToH [15].

In multilevel framework,net costs are incorporated to PaToH in three phases. In the first phase of coarsening of net costs if performed.In the phase of initial partitioning the algorithm modification of GHGP [15] is performed to be used with target weights and net costs.In the last phase of refinement ,heuristic FM [16] is modified for accurate calculation of the vertex move gains and cutsize.

In [29] investigator proposed an algorithm for each build time and run time stage.In the initial stage, dependency between all the data sets is calculated and dependency matrix is built.For the partitioning and clustering of data sets BEA(Bond Energy Algorithm) is used.These partitions are distributed among different data centers.The data centers are partitioned using k means algorithm.After the generation of intermediate data ,the newly proposed clustering algorithm deals with new datasets.The dependencies for each data center are judged and then accordingly data is moved.The factors of data movement and gathering of data at one point is covered up.This strategy allocate application data among data centers automatically and reduces the movement of data.

In [50] the investigator propose non linear programming model(NLP) for minimization of retrieval of data and execution cost of workflows.The proposed technique aims at minimizing the computation cost and data transfer cost on the compute resource.For case study,intrusion detection application is considered.While using

Table 3.11
Objectives of optimization based data placement schemes

Schemes	Energy aware	Cost aware	Resource aware	Application aware	Factors
[49]	-	-	✓	-	storage capacity, data interdependency, WAN bandwidth, IP address
[68]	-	✓	✓	-	Communication cost
[29]	-	✓	-	-	Data dependency,Cost
[48]	-	✓	-	-	Execution cost,Communication cost
[50]	-	✓	-	-	execution cost
[?]	-	✓	✓	-	execution cost, execution time, data dependency

Table 3.12
Properties of Optimization based data placement schemes

Schemes	SLA support	Security	Cloud/ Grid	Algorithm	Clustering Algorithm	Tool/Programming Language
[49]	✓	-	Cloud	iterative optimization	Cloud	-
[68]	-	-	Cloud	combinatorial algorithm	-	C programming language
[29]	✓	-	Cloud	Bond Energy	K-Means	SwinDeW-C
[48]	✓	-	Cloud	PSO	-	-
[50]	✓	-	Cloud	NLP	-	Amazon CloudFront

the storage and compute resources ,NLP model is applied on intrusion detection application.

3.7. Fault tolerance based data placement. In [19] investigator proposed the strategy based on clustering algorithm of consistent hashing and minimum distance.The data is clustered efficiently by the clustering algorithm by placing item based and user based data .For addressing effect of outliers and noises ,cluster centers and threshold is updated.CBR strategy [18] and item based algorithm CF is used to fill sparse matrix of users.Consistent hashing algorithm is used for improving fault toleration and scalability.

A Data placement scheduler ,Stork provide an ability to queue,schedule,manage and monitor the data placement jobs.It also applies the technique of checkpointing the jobs.This approach provides a complete automation for processing the data. The proposed system automate the data transfer fully automatically between the heterogenous systems.It possessess the ability to recover from network,software and stoarge system failures without human intervention.It performs dynamic adaption of data placement jobs at the execution time to system environment [34]

3.8. Replication based data placement. In [51] investigator present a heuristic based on ordering jobs cleaned up in workflow and reduce time taken for execution.Also genetic algorithm based approach vary the amount of storage and number of processors for generating schedules with low cost.Tightly coupled data staging approach is used.As per the model introduced DPS and workflow manager work together for fulfilling the requirements of data management.

In [53] investigator proposed CDRM(Cost effective dynamic replication management) that captures the relations between replica number and availability.It aim to increase availability of data for storage system of cloud for balancing load and improving performance while considering failures.CDRM instincts proposed model for maintaining minimum number of replicas according to requirement availability.Placement of replica is on the basis blocking capacity and probbiility of data nodes.CDRM dynamically redistributes according to changing workload and capacity of node.CDRM is implemented in Hadoop Distributed File System(HDFS).

In [52] author proposed two-level DHT (TDHT) approach for applying Lazy update and minimize communication cost.At first level, trade off on security and availability between TDHT and conventional pure data partitioning approach is performed.This approach is integrated with DHT and is called GDHT(Global DHT) .

[6] The investigator used the replication strategy for achieving availability,reliability and appropriate utilization of network bandwidth.PC cluster system is used for implementation of cloud storage system.

Table 3.13
Objectives of Fault tolerant based data placement schemes

Schemes	**Energy aware**	**Cost aware**	**Resource aware**	**Application aware**	**Factors**
[19]	-	✓	✓	-	Fault tolerance,Execution Time
[34]	-	✓	-	✓	Data dependency,Fault tolerance

Table 3.14
Properties of Fault tolerant based data placement schemes

Schemes	**SLA support**	**Security**	**Cloud/ Grid**	**Algorithm**	**Clustering Algorithm**	**Tool/Programming Language**
[19]	✓	-	Cloud	Consistent Hashing	K means	VMware
[34]	✓	-	Grid	-	-	Stock Server

Author proposed new interconnection network MyHeawood for cost effective data placement.It is based on different hashing functions.It is a hierarchical network.It is composed of small switch and dual NIC server.Data placement strategy is based on the hashing function composed of different hash functions. Hash functions are composed of hash key for computing server address for master replica. After that, on the basis of address of master replica ,remaining replicas are allocated in different layer.Three replicas are used for solving overhead of storage caused by multiple replicas.This solves the issue of reliability in cloud. [24]

Further work on replication is improved by [54].

In [23] author introduced a Balanced and file Reuse-Replication Scheduling (BaRRS) algorithm for cloud computing.This algorithm divides the scientific workflow into multiple workflows for balance utilization through parallelization. It deals with reuse of data and different replication techniques for optimization of transferable data.It considers execution time of task,patterns of dependency among tasks and size of files for adapting to current techniques of replication and data reuse.At last it selects the optimal solution on basis of monetary cost and execution time.

4. Results. The main motivation of this research work is to find all the available research in data placement of scientific workflows . In all, 64 research articles are published in prominent journals, symposiums, workshops and in foremost conferences on cloud computing. Most of the research articles on data placement algorithms are published in comprehensive variety of journals and conference proceedings.We perceived that conferences like IEEE/ACM , Future Generation Computer systems, Proceedings of the 4th generation workshop on workflows in support of large-scalescience, Encyclopedia of parallel computing: Springer , Proceedings of fourth international workshop on data intensive distributed computing, Proceedings of VLDB Endowment, Parallel and Distributed Computing international symposium, Cluster Computing and Grid 8th IEEE international symposium, Proceedings of 17th international symposium on high performance distributed computing, Grid computing environments workshop, International Conference on cloud computing 2009, Future Information Technology : Springer, International Journal of High Performance Computing Applications, Procedia Environmental Sciences, The Computer Journal, Cloud Computing and Distributed System Laboratory, International Conference on Super Computing, Computing in science and Engineering, Journal of Grid Computing, Computer standardsand interfaces, Journal of Electrical and Computer Engineering, Chinese Journal of Computers contribute significantly to our review area. Figure 8 shows the percentage of research paper discussing different data placement algorithms (Coorelation, genetic, Energy Efficient, PSO based, ACO based, Optimization Based, Fault Tolerance based, Replication Based) from year 2008 to 2017. Summary table for comparison of data placement algorithms is shown in Table 4.1.

Figure 4.2 depicts the maximum research in Replication based(17%) and PSO based(17%) data placement while very few research in fault tolerance based data placement(6%) and genetic based data placement(9%). Coorelation based and Optimization based contributes 14% each while Energy Efficient, ACO based and Genetic based contributes 12%, 11% and 9% respectively.

Table 3.15
Objectives of Replication based data placement schemes

Schemes	Energy aware	Cost aware	Resource aware	Application aware	Factors
[51]	-	✓	-	-	Makespan,execution cost,storage limit
[34]	-	✓	-	✓	Data dependency, Fault tolerance
[53]	-	✓	-	-	CPU power, memory capacity, network bandwidth, access latency, load balance
[52]	-	✓	-	-	Response Latency,Cost
[24]	-	✓	-	✓	Cost
[23]	-	✓	-	✓	Execution time,Data dependency,Task Size

Table 3.16
Properties of Replication based data placement schemes

Schemes	SLA support	Security	Cloud/ Grid	Algorithm	Clustering Algorithm	Tool/Programming Language
[51]	-	-	Grid	Coarse and fine grained genetic algorithm	-	Cluster viz16
[34]	✓	-	Grid	-	-	Stock Server
[53]	✓	-	Cloud	CDRM	-	-
[52]	✓	✓	cloud	uTLA	-	PlanetLab (Plab) platforms
[24]	-	-	Cloud	Hashing	-	Cloudsim
[23]	-	-	Cloud	BaaRS	-	VMware-ESXi-based (version)

4.1. Data Placement implication platforms. For the implementation of the data placement algorithms as in figure 4.2 , 79% of the algorithms are implemented on cloud while 17% on Grid and 4% on file servers. For the simulation of cloud environment, the tools used are in Figure 4.2.

5. Conclusion and Future Directions. Cloud computing is an emerging computing platform that presents different schemes for the large amount of data arising from scientific workflow application.Data placement is one of the necessary schemes that make cloud computing possible.Recenetly data placement problem has become a hot topic in the area of cloud computing while considering the scientific workflow applications because it can greatly reduce the execution time of workflows and increase the efficiency of data centers.

This paper presents a comprehensive survey and analysis of the data placement schemes proposed for cloud computing.For this purpose, first the definition of data placement schemes are provided, and then the classification of data placement schemes is presented based on the type of placement.Then various proposed data placement schemes are described and factors of data placement in each scheme are investigated to illuminate the advantages, limitations and properties of each placement scheme in detail. Moreover, complete comparative comparisons of the data placement schemes are presented which highlight the items that should be considered in future studies and researches. As specified in these comparisons, most data placement schemes are aimed to improve the performance and execution related issues in data centers.There is less focus on security related issues in data placement.But with the ever increasing security attacks on services of cloud,providing security in the cloud environment has become a critical issue. As a result, security is one of the crucial issue which should be considered in the future data placement.

As per the literature survey there is scope of further improvement in data placement in cloud.Following section describes further improvements in this field.

1. For improving the inter data center traffic ,identification of nearby sites can be pursued that leads to improved latency and costs.The location of data center can be provided as an input.In future more placement factors can be considered such as computation capacity of each server and load balancing.Replicating the used data can also be used for performance in terms of system response time and

Table 4.1
Comparison of different data placement algorithms on placement criteria

Schemes	**Security**	**Storage Capacity**	**Network Bandwidth**	**Task Characterstics**	**Data dependency**	**Fault Tolerance**	**Transfer Time**	**Transfer Cost**	**Execution Time**	**Data Movement**
Coorelation Based Data placement										
[35]	-	-	-	✓	✓	-	-	-	-	-
[36]	-	-	-	-	-	-	-	✓	-	-
[34]	-	-	-	-	✓	-	-	-	-	-
[29]	-	-	-	-	✓	-	✓	-	✓	✓
[9]	-	-	-	✓	✓	-	-	-	✓	-
Genetic Algorithm Based Data placement										
[30]	-	-	-	-	-	-	-	-	-	✓
[28]	-	✓	-	-	-	-	-	-	-	✓
[37]	-	-	-	-	✓	-	-	✓	-	✓
Energy Efficient based Data placement										
[21]	-	-	-	-	-	-	-	-	✓	-
[43]	-	-	-	✓	-	-	-	-	-	-
[44]	-	-	-	-	-	-	-	✓	-	-
[45]	-	-	-	-	-	-	✓	✓	-	-
PSO Based Data placement										
[?]	-	-	-	-	-	-	-	✓	✓	-
[7]	-	-	-	-	-	-	✓	✓	✓	✓
[22]	-	-	-	-	✓	-	-	✓	-	-
[17]	-	-	-	-	-	-	✓	✓	✓	-
[48]	-	-	-	-	✓	-	-	✓	✓	-
[8]	-	-	-	-	✓	-	-	-	-	✓
ACO Based Data placement										
[67]	✓	-	-	-	-	-	✓	-	-	-
[22]	-	-	-	-	✓	-	-	✓	-	-
[17]	-	-	-	-	-	-	-	✓	✓	✓
[48]	-	-	-	-	-	-	-	✓	✓	-
Optimization Based Data placement										
[49]	-	✓	✓	-	✓	-	-	-	-	-
[68]	-	-	-	-	-	-	-	✓	-	-
[29]	-	-	-	-	✓	-	-	✓	-	-
[48]	-	-	-	-	-	-	-	✓	✓	-
[50]	-	-	-	-	-	-	-	-	✓	✓
[?]	-	-	-	-	✓	-	-	-	✓	-
Fault Tolerance Based Data placement										
[19]	-	-	-	-	-	✓	-	-	✓	-
[34]	-	-	-	-	✓	✓	-	-	-	-
Replication Based Data placement										
[51]	-	✓	-	-	-	-	-	✓	✓	-
[34]	-	-	-	-	✓	✓	-	-	-	-
[53]	-	✓	✓	-	-	-	-	-	-	-
[52]	-	-	-	-	-	-	-	✓	✓	-
[24]	-	-	-	-	-	-	-	✓	-	-
[23]	-	-	-	✓	✓	-	-	-	✓	-

reliability.

2. The data sets are of two types fixed location and variable location data sets. The fixed location datasets are the one that are located in a predetermined locations in the cloud environment and are non transferable whereas the variable location datasets can be moved from one place to another. There are works listing the use of variable location datasets, but for enhancement fixed location datasets can

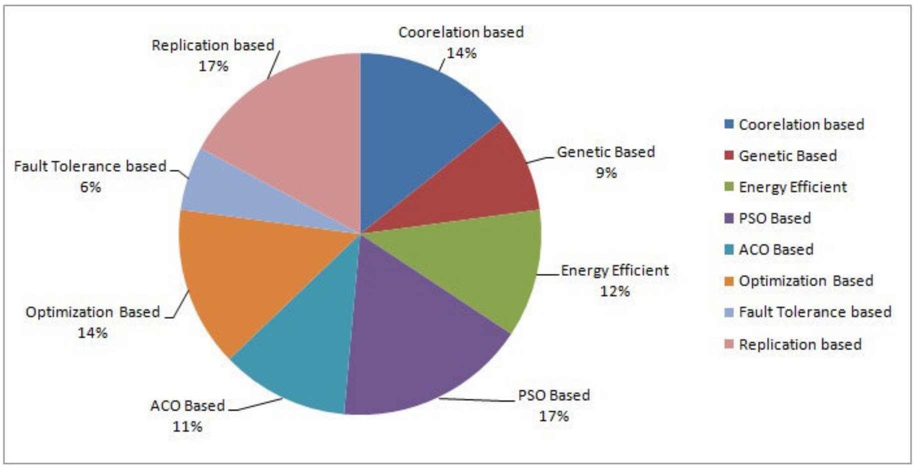

Fig. 4.1. *Data Placement Algorithms in Cloud.*

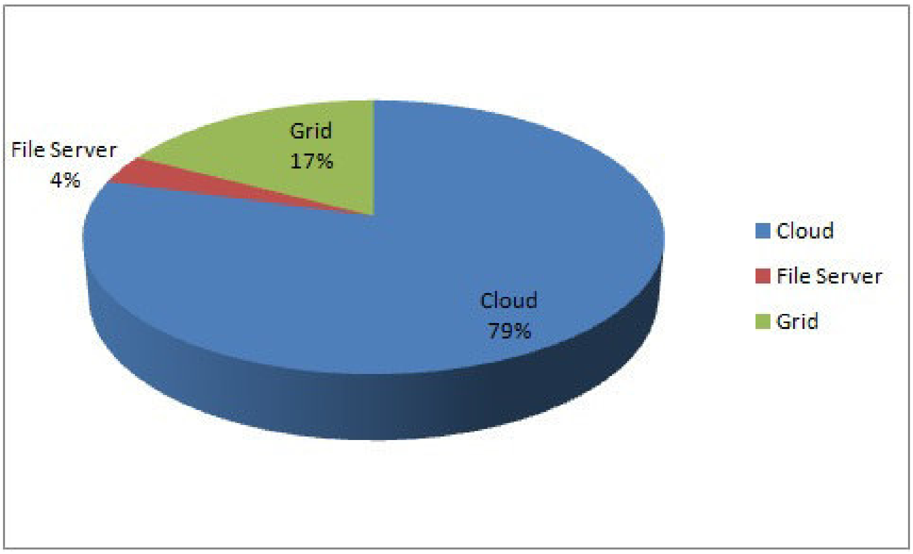

Fig. 4.2. *Platform for Algorithms.*

also be considered for further improvement of cost,price etc.

3. The system performance of cloud can also be improved further by detailing the characteristics of tasks that can further be used for improvement of scheduling mechanism in a heterogeneous environment.Further policies can be implemented on the basis of characteristics of tasks and system with different policy schemes can be implemented.
4. The cost of running the workflow and the energy consumption can be analyzed .The further investigation is required on the issue for the minimization of power consumption that will lead to improvement of performance in scientific workflows.The data placement can be explored by considering the energy/power efficiency and power/performance trade-offs.
5. In future more placement factors can be considered such as computation capacity of each server and load balancing.Replicating the used data can also be used for better performance in terms of system reliability and response time.
6. Different methods of genetic selection affect the performance of the algorithm which requires further study.Fuzzy strategies are not outlined.The granularization approach most suitable for the entire work-

load and optimal storage location of each of the resulting granular,sub workloads can be determined.

REFERENCES

[1] Ewa Deelman and Ann Chervenak. Data management challenges of data-intensive scientific workflows. In *Cluster Computing and the Grid, 2008. CCGRID'08. 8th IEEE International Symposium on*, pages 687–692. IEEE, 2008.

[2] Alexandros Labrinidis and Hosagrahar V Jagadish. Challenges and opportunities with big data. *Proceedings of the VLDB Endowment*, 5(12):2032–2033, 2012.

[3] John J Rehr, Fernando D Vila, Jeffrey P Gardner, Lucas Svec, and Micah Prange. Scientific computing in the cloud. *Computing in science & Engineering*, 12(3):34–43, 2010.

[4] Han Ning Wang, Wei Xiang Xu, and Chao Long Jia. A high-speed railway data placement strategy based on cloud computing. In *Applied Mechanics and Materials*, volume 135, pages 43–49. Trans Tech Publ, 2012.

[5] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.

[6] Julia Myint and Thinn Thu Naing. A data placement algorithm with binary weighted tree on pc cluster-based cloud storage system. In *Cloud and Service Computing (CSC), 2011 International Conference on*, pages 315–320. IEEE, 2011.

[7] Lizheng Guo, Zongyao He, Shuguang Zhao, Na Zhang, Junhao Wang, and Changyun Jiang. Multi-objective optimization for data placement strategy in cloud computing. In *International Conference on Information Computing and Applications*, pages 119–126. Springer, 2012.

[8] Qing Zhao, Congcong Xiong, and Peng Wang. Heuristic data placement for data-intensive applications in heterogeneous cloud. *Journal of Electrical and Computer Engineering*, 2016, 2016.

[9] Tao Wang, Shihong Yao, Zhengquan Xu, and Shan Jia. Dccp: an effective data placement strategy for data-intensive computations in distributed cloud computing systems. *The Journal of Supercomputing*, 72(7):2537–2564, 2016.

[10] Ewa Deelman, James Blythe, Yolanda Gil, and Carl Kesselman. Pegasus: Planning for execution in grids. *GriPhyN*, 20:2002, 2002.

[11] Ian Foster, Jens Vockler, Michael Wilde, and Yong Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation. In *Scientific and Statistical Database Management, 2002. Proceedings. 14th International Conference on*, pages 37–46. IEEE, 2002.

[12] Fran Berman, Geoffrey Fox, and Anthony JG Hey. *Grid computing: making the global infrastructure a reality*, volume 2. John Wiley and sons, 2003.

[13] Xin Liu and Anwitaman Datta. Towards intelligent data placement for scientific workflows in collaborative cloud environment. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 1052–1061. IEEE, 2011.

[14] Ronnie Chaiken, Bob Jenkins, Per-Åke Larson, Bill Ramsey, Darren Shakib, Simon Weaver, and Jingren Zhou. Scope: easy and efficient parallel processing of massive data sets. *Proceedings of the VLDB Endowment*, 1(2):1265–1276, 2008.

[15] Ümit Çatalyürek and Cevdet Aykanat. Patoh (partitioning tool for hypergraphs). In *Encyclopedia of Parallel Computing*, pages 1479–1487. Springer, 2011.

[16] Charles M Fiduccia and Robert M Mattheyses. A linear-time heuristic for improving network partitions. In *Papers on Twenty-five years of electronic design automation*, pages 241–247. ACM, 1988.

[17] Lizheng Guo, Shuguang Zhao, Shigen Shen, and Changyuan Jiang. Task scheduling optimization in cloud computing based on heuristic algorithm. *JNW*, 7(3):547–553, 2012.

[18] Zeina Chedrawy and Syed Sibte Raza Abidi. An intelligent knowledge sharing strategy featuring item-based collaborative filtering and case based reasoning. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, pages 67–72. IEEE, 2005.

[19] Qiang Li, Kun Wang, Suwei Wei, Xuefeng Han, Lili Xu, and Min Gao. A data placement strategy based on clustering and consistent hashing algorithm in cloud computing. In *Communications and Networking in China (CHINACOM), 2014 9th International Conference on*, pages 478–483. IEEE, 2014.

[20] Mingjun Wang, Jinghui Zhang, Fang Dong, and Junzhou Luo. Data placement and task scheduling optimization for data intensive scientific workflow in multiple data centers environment. In *Advanced Cloud and Big Data (CBD), 2014 Second International Conference on*, pages 77–84. IEEE, 2014.

[21] Yanwen Xiao, Jinbao Wang, Yaping Li, and Hong Gao. An energy-efficient data placement algorithm and node scheduling strategies in cloud computing systems. In *Proc. of the 2nd Intl Conf. on Advances in Computer Science and Engineering (CSE 2013). Paris: Atlantis Press*, volume 63, 2013.

[22] Xuejun Li, Yang Wu, Fei Ma, Erzhou Zhu, Futian Wang, Lei Wu, and Yun Yang. A new particle swarm optimization-based strategy for cost-effective data placement in scientific cloud workflows. In *Future Information Technology*, pages 115–120. Springer, 2014.

[23] Israel Casas, Javid Taheri, Rajiv Ranjan, Lizhe Wang, and Albert Y Zomaya. A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems. *Future Generation Computer Systems*, 2016.

[24] Xin Huang, Yu Xing Peng, and Peng Fei You. Data placement and query for cloud computing based on myheawood network. In *Applied Mechanics and Materials*, volume 543, pages 3100–3104. Trans Tech Publ, 2014.

[25] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[26] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[27] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic

algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[28] Wei Guo and Xinjun Wang. A data placement strategy based on genetic algorithm in cloud computing platform. In *Web Information System and Application Conference (WISA), 2013 10th*, pages 369–372. IEEE, 2013.

[29] Dong Yuan, Yun Yang, Xiao Liu, and Jinjun Chen. A data placement strategy in scientific cloud workflows. *Future Generation Computer Systems*, 26(8):1200–1214, 2010.

[30] Zhao Er-Dun, Qi Yong-Qiang, Xiang Xing-Xing, and Chen Yi. A data placement strategy based on genetic algorithm for scientific workflows. In *Computational Intelligence and Security (CIS), 2012 Eighth International Conference on*, pages 146–149. IEEE, 2012.

[31] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. *INC, IMS and IDC*, pages 44–51, 2009.

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

[33] David Hollingsworth and UK Hampshire. Workflow management coalition: The workflow reference model. *Document Number TC00-1003*, 19, 1995.

[34] Tevfik Kosar and Miron Livny. A framework for reliable and efficient data placement in distributed computing systems. *Journal of Parallel and Distributed Computing*, 65(10):1146–1157, 2005.

[35] Shyamala Doraimani and Adriana Iamnitchi. File grouping for scientific data management: lessons from experimenting with real traces. In *Proceedings of the 17th international symposium on High performance distributed computing*, pages 153–164. ACM, 2008.

[36] Gilles Fedak, Haiwu He, and Franck Cappello. Bitdew: a programmable environment for large-scale data management and distribution. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*, pages 1–12. IEEE, 2008.

[37] Pai Zheng, Li-Zhen Cui, Hai-Yang Wang, and Meng Xu. A data placement strategy for data-intensive applications in cloud. *Jisuanji Xuebao(Chinese Journal of Computers)*, 33(8):1472–1480, 2010.

[38] Meikel Poess and Raghunath Othayoth Nambiar. Tuning servers, storage and database for energy efficient data warehouses. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 1006–1017. IEEE, 2010.

[39] Andreas Beckmann, Ulrich Meyer, Peter Sanders, and Johannes Singler. Energy-efficient sorting using solid state disks. *Sustainable Computing: Informatics and Systems*, 1(2):151–163, 2011.

[40] Lei Rao, Xue Liu, Le Xie, and Wenyu Liu. Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[41] Willis Lang and Jignesh Patel. Towards eco-friendly database management systems. *arXiv preprint arXiv:0909.1767*, 2009.

[42] EN Mootaz Elnozahy, Michael Kistler, and Ramakrishnan Rajamony. Energy-efficient server clusters. In *International Workshop on Power-Aware Computer Systems*, pages 179–197. Springer, 2002.

[43] Eduardo Pinheiro and Ricardo Bianchini. Energy conservation techniques for disk array-based servers. In *ACM International Conference on Supercomputing 25th Anniversary Volume*, pages 369–379. ACM, 2014.

[44] Nitesh Maheshwari, Radheshyam Nanduri, and Vasudeva Varma. Dynamic energy efficient data placement and cluster reconfiguration algorithm for mapreduce framework. *Future Generation Computer Systems*, 28(1):119–127, 2012.

[45] Bo Li, Jianxin Li, Jinpeng Huai, Tianyu Wo, Qin Li, and Liang Zhong. Enacloud: An energy-saving application live placement approach for cloud computing environments. In *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*, pages 17–24. IEEE, 2009.

[46] Anton Beloglazov, Jemal Abawajy, and Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, 28(5):755–768, 2012.

[47] Jason M Cope, Nick Trebon, Henry M Tufo, and Pete Beckman. Robust data placement in urgent computing environments. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–13. IEEE, 2009.

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

[49] Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, and Harbinder Bhogan. Volley: Automated data placement for geo-distributed cloud services. In *NSDI*, volume 10, pages 28–0, 2010.

[50] Suraj Pandey, Kapil Kumar Gupta, Adam Barker, and Rajkumar Buyya. Minimizing cost when using globally distributed cloud services: A case study in analysis of intrusion detection workflow application. *Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, Melbourne, Australia, Tech. Rep*, 2009.

[51] Shishir Bharathi and Ann Chervenak. Scheduling data-intensive workflows on storage constrained resources. In *Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science*, page 3. ACM, 2009.

[52] Yunqi Ye, Liangliang Xiao, I-Ling Yen, and Farokh Bastani. Cloud storage design based on hybrid of replication and data partitioning. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, pages 415–422. IEEE, 2010.

[53] Qingsong Wei, Bharadwaj Veeravalli, Bozhao Gong, Lingfang Zeng, and Dan Feng. Cdrm: A cost-effective dynamic replication management scheme for cloud storage cluster. In *Cluster Computing (CLUSTER), 2010 IEEE International Conference on*, pages 188–196. IEEE, 2010.

[54] Suraj Pandey and Rajkumar Buyya. Scheduling workflow applications based on multi-source parallel data retrieval in distributed computing networks. *The Computer Journal*, 55(11):1288–1308, 2012.

[55] Jia Yu, Rajkumar Buyya, and Chen Khong Tham. Cost-based scheduling of scientific workflow applications on utility

grids. In *e-Science and Grid Computing, 2005. First International Conference on*, pages 8–pp. Ieee, 2005.

[56] Hua Huang, Yi Lai Zhang, and Min Zhang. A survey of cloud workflow. In *Advanced Materials Research*, volume 765, pages 1343–1348. Trans Tech Publ, 2013.

[57] Sucha Smanchat and Kanchana Viriyapant. Taxonomies of workflow scheduling problem and techniques in the cloud. *Future Generation Computer Systems*, 52:1–12, 2015.

[58] Jan Hidders, Paolo Missier, and Jacek Sroka. Recent advances in scalable workflow enactment engines and technologies. *Future Generation Comp. Syst.*, 46:1–2, 2015.

[59] Bang Ouyang, Farong Zhong, and Huan Liu. An eca-based control-rule formalism for the bpel process modularization. *Procedia Environmental Sciences*, 11:511–517, 2011.

[60] Rongbin Xu, Xiao Liu, Ying Xie, Futian Wang, Cheng Zhang, and Yun Yang. Logistics scheduling based on cloud business workflows. In *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on*, pages 29–34. IEEE, 2014.

[61] Ke Liu, Hai Jin, Jinjun Chen, Xiao Liu, Dong Yuan, and Yun Yang. A compromised-time-cost scheduling algorithm in swindew-c for instance-intensive cost-constrained workflows on cloud computing platform. *International Journal of High Performance Computing Applications*, 2010.

[62] Ewa Deelman, Dennis Gannon, Matthew Shields, and Ian Taylor. Workflows and e-science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540, 2009.

[63] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R Pocock, Anil Wipat, et al. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[64] Gregor von Laszewski and Mike Hategan. Java cog kit karajan/gridant workflow guide. Technical report, Technical Report, Argonne National Laboratory, Argonne, IL, USA, 2005.

[65] Ian Taylor, Matthew Shields, Ian Wang, and Andrew Harrison. Visual grid workflow in triana. *Journal of Grid Computing*, 3(3-4):153–169, 2005.

[66] Guoli Li, Vinod Muthusamy, and Hans-Arno Jacobsen. A distributed service-oriented architecture for business process execution. *ACM Transactions on the Web (TWEB)*, 4(1):2, 2010.

[67] Wei Liu, Su Peng, Wei Du, Wei Wang, and Guo Sun Zeng. Security-aware intermediate data placement strategy in scientific cloud workflows. *Knowledge and information systems*, 41(2):423–447, 2014.

[68] Ümit V Çatalyürek, Kamer Kaya, and Bora Uçar. Integrated data placement and task assignment for scientific workflows in clouds. In *Proceedings of the fourth international workshop on Data-intensive distributed computing*, pages 45–54. ACM, 2011.

[69] Weiwei Chen, Rafael Ferreira, Ewa Deelman, and Rizos Sakellariou. Balanced Task Clustering in Scientific Workflows. pages 1–8.

Edited by: Pijush Kanti Dutta Pramanik
Received: Mar 16, 2019
Accepted: Apr 2, 2019