IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 8, 2021, accepted March 24, 2021, date of publication April 2, 2021, date of current version April 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070785

Scientific Workflows Management and Scheduling in Cloud Computing: Taxonomy, Prospects, and Challenges

ZULFIQAR AHMAD[1], ALI IMRAN JEHANGIRI[1], MOHAMMED ALAA ALA'ANZY[2], MOHAMED OTHMAN[2,3], (Senior Member, IEEE), ROHAYA LATIP[2], SARDAR KHALIQ UZ ZAMAN[1,4], AND ARIF IQBAL UMAR[1]

[1]Department of Information Technology, Hazara University, Mansehra 21300, Pakistan
[2]Department of Communication Technology and Networks, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia
[3]Laboratory of Computational Science and Mathematical Physics, Institute of Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia
[4]Department of Computer Science, COMSATS University Islamabad, Abbottabad Campus, Abbottabad 22060, Pakistan

Corresponding authors: Ali Imran Jehangiri (ali_imran@hu.edu.pk), Mohammed Alaa Ala'anzy (m.alanzy.cs@gmail.com), and Mohamed Othman (mothman@upm.edu.my)

This work was supported in part by the Universiti Putra Malaysia, and in part by the Ministry of Education Malaysia.

ABSTRACT Cloud computing provides solutions to a large number of organizations in terms of hosting systems and services. The services provided by cloud computing are broadly used for business and scientific applications. Business applications are task oriented applications and structured into business workflows. Whereas, scientific applications are data oriented and compute intensive applications and structured into scientific workflows. Scientific workflows are managed through scientific workflows management and scheduling systems. Recently, a significant amount of research is carried out on management and scheduling of scientific workflow applications. This study presents a comprehensive review on scientific workflows management and scheduling in cloud computing. It provides an overview of existing surveys on scientific workflows management systems. It presents a taxonomy of scientific workflow applications and characteristics. It shows the working of existing scientific workflows management and scheduling techniques including resource scheduling, fault-tolerant scheduling and energy efficient scheduling. It provides discussion on various performance evaluation parameters along with definition and equation. It also provides discussion on various performance evaluation platforms used for evaluation of scientific workflows management and scheduling strategies. It finds evaluation platforms used for the evaluation of scientific workflows techniques based on various performance evaluation parameters. It also finds various design goals for presenting new scientific workflow management techniques. Finally, it explores the open research issues that require attention and high importance.

INDEX TERMS Scientific workflows, scientific applications, resource management, scheduling, montage, cybershake.

I. INTRODUCTION

Cloud computing is an emerging and distributed computing platform that has now attained the goal of ''computer as utility'' [1]. Cloud computing is the sequential aroma of renowned computing paradigms i.e., cluster computing and grid computing [2]. Particularly, cloud computing allows provision of reliable resources, on demand and computing environments are customized in a way of pay-as-you-go [3]–[5]. Cloud computing offers a pool of dynamically available computing resources and services which are virtualized, abstracted and configurable/reconfigurable [6]. Cloud resources are provided on a subscription based environment and which are shaped in the form of: (a) networks, (b) storage, (c) servers, and (d) applications [1]. Cloud Services are delivered to external customers' on-demand and over a high-speed Internet with three segments computing architecture i.e., (a) Infrastructure as a Service (IaaS), (b) Platform as a Service (PaaS), and (c) Software as a Service (SaaS) [7]–[9].

Services and resources provided by cloud computing are broadly used for business and scientific applications [10], [11]. Business applications are task oriented applications

The associate editor coordinating the review of this manuscript and approving it for publication was Yanjiao Chen.

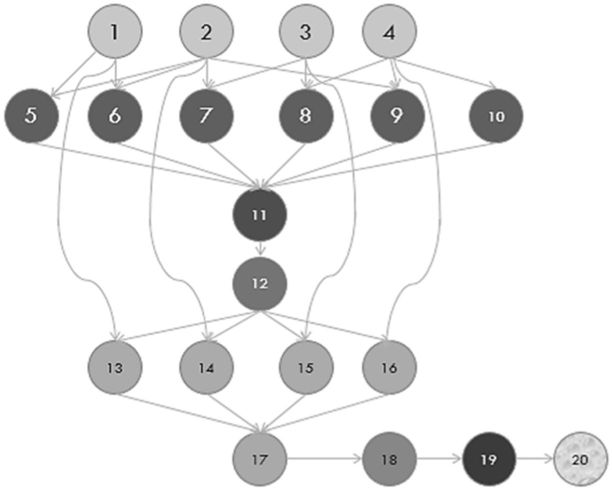


FIGURE 1. An executional architecture of Montage Scientific Workflow [37].

and structured into business workflows. For management of business workflows, business models are used e.g., Amazon EC2 [12], [13]. On the other hand scientific applications are data oriented applications and structured into scientific workflows. Scientific workflows are managed through scientific workflow management systems e.g., Pegasus workflow management system [14], [15].

Scientific workflows are data-intensive workflows that require high computation and storage power [16]–[18]. For example, CyberShake [19] is a real time scientific workflow related to seismology (earthquake science). In CyberShake the seismic hazards for a particular location are quantified by seismologists using probabilistic seismic hazard analysis (PSHA). PSHA provides a technique to estimate the probability of earthquake ground motions level at a particular location with intensity measure (IM), such as peak ground acceleration or peak ground velocity, over a given time period. Such probabilistic measures are useful for building engineers, insurance agencies and civic planners as such influence billions of dollars each year. CyberShake also requires the significant computational and storage resources as per site of interest there is 755 GB of data is processed within 14100 CPU hours [20]. Likewise, Montage [21] is a real time scientific workflow related to astronomy wherein input images are computed to form desired mosaics. It is a data-intensive application as it processes the high definition input images. These input images are taken by the astronomer from the region of sky for which the mosaics are desired. Figure 1 shows an executional architecture of Montage scientific workflow.

The size of the desired mosaic is represented in terms of square degree. Like, there are 203, 732 and 3,027 application tasks in Montage 1, 2, and 4 degree square workflows respectively. By considering a montage 4 degree square workflow consisting of 3,027 application tasks, the runtime is 85 CPU hours with a cost of $9 when running on 1 processor which is a quite considerable large running time. The executional steps as reflected from Figure 1 follows the following steps.

1. The input is four images of FITS (Flexible Image Transport System) format.
2. Images with common characteristics are separated and differences of each pair of overlapping images are calculated.
3. The differences of images are integrated.
4. Correction is applied to obtain a good global image.
5. Background correction is applied to each individual image.
6. Aggregate metadata from all the images and tables is created.
7. Co-adds all the re-projected images and FITS format images are created.
8. Size of the image is reduced by averaging blocks of pixels.
9. Images are converted into JPEG (Joint Photographic Experts Group) format and thus, final mosaic is created.

Scientific workflow management systems (SWfMSs) in terms of complex engines are used to model and execute the scientific workflows [22], [23]. The SWfMSs have the capabilities of fault-tolerance, monitoring and parallelization methods that provide data, storage and compute intensive experiments with high processing power [24]. High performance computing (HPC) and high throughput computing (HTC) such as cloud and fog computing [25] provide the required processing power for execution of distributed scientific workflows designed by the SWfMSs. On the other hand, many workflow applications come from big data processing and IoT (Internet of Things) applications [26]. For modelling and execution of scientific workflows, the big data frameworks attract more and more attention [27], [28]. Several big data frameworks including Hadoop [29] (MapReduce processing framework), Flink [30], Samza [31], and Storm [23] (stream processing frameworks), and Spark [32] (batch processing framework with stream processing capabilities) are available for modelling and execution of scientific workflows. Each framework is responsible for provision of distributed computation over data. The idea behind all these big data frameworks is to define core computations without spending time on parallelizing the applications. In the field of health care, the development of big data brought huge economic and social benefits to the society. For instance, it was indicated in [33] that after a hack of DataBreaches.net about half million patient records are compromised, the DarkOvrlord stole 180,000 patient records through trespass, phishing attack on Washington medical staff outcomes in the release of more than 80, 000 medical records. However, in the context of big data, the way of privacy disclosure is more secure [34]–[36].

The scientific workflows include the fields astronomy, earthquake science, biology, and gravitational physics [37]. Figure 2 shows an overview of cloud computing resources

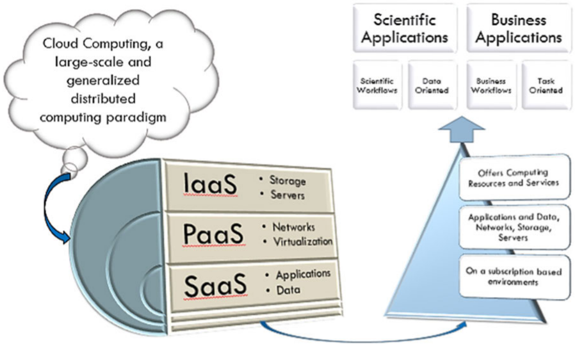

Cloud Computing, a large-scale and generalized distributed computing paradigm
IaaS
• Storage
• Servers
PaaS
• Networks
• Virtualization
SaaS
• Applications
• Data
Scientific Applications
Business Applications
Scientific Workflows
Data Oriented
Business Workflows
Task Oriented
Offers Computing Resources and Services
Applications and Data, Networks, Storage, Servers
On a subscription based environments

FIGURE 2. An overview of cloud computing resources and services.

and services to be provided for two broad categories i.e., scientific application and business applications.

One of the most important issues to handle the scientific workflows is to manage the resources and services of cloud computing [11]. Scientific workflows management is a process in which cloud resources and services are procured to evaluate scientific application and then released accordingly [38]. Scientific workflows can also consist of I/O (Input/Output) intensive tasks that take a non-negligible and one of the major parts of the execution. This part spends more time for doing I/O operation instead of computation due to some dependency structure [17].

The major contributions of this study are as follows:

- It provides an overview of existing surveys on scientific workflows management systems.
- It presents a taxonomy of scientific workflow applications and characteristics.
- It shows the working of existing scientific workflows management and scheduling techniques including resource scheduling, fault-tolerant scheduling and energy efficient scheduling.
- It provides discussion on various performance evaluation parameters along with definition and equation.
- It provides discussion on various performance evaluation platforms used for evaluation of scientific workflows management and scheduling strategies.
- It finds evaluation platforms used for the evaluation of scientific workflows techniques based on various performance evaluation parameters.
- It finds various design goals for presenting new scientific workflow management techniques.
- Finally, it explores out the open research issues that require attention and high importance.

The rest of the paper is organized as follows: Research methodology is provided in Section 2. Brief overview of existing surveys is presented in Section 3. Section 4 provides the detailed description, basic characteristics and terminologies used for workflow applications. Section 5 illustrates the taxonomy of scientific workflows management and scheduling techniques. Section 6 presents the major performance evaluation parameters used for scientific workflows. Section 7 presents the description on performance evaluation platforms. An overview of evaluation parameters and platforms are provided in Section 8. Section 9 provides discussion on design goals. Section 10 highlights various challenges, and Section 11 concludes the paper.

II. RESEARCH METHODOLOGY

To expand our comprehension of scientific workflows management and scheduling system, a systematic literature review (SLR) was carried out, with the benchmark proposed by [38], [39] with a precise concentration on research associated with scientific workflows scheduling and management mechanisms. Significant work has been done on scientific workflows management and scheduling in cloud computing after 2008 on advent of cloud infrastructures and real-world scientific workflow applications [37]. Thus, in the instant work, we are intended to provide perception, and discuss issues in most relevant techniques used for scientific workflows management and scheduling from January, 2008 to January, 2021. More specifically, the research articles were reviewed from IEEE, Elsevier, and Springer along with the other reputed platforms as these provided deep analysis. The reading of the papers was started from the title followed by the abstract. If the abstract did not provide enough details, then the whole article was read. Therefore, the articles included in this review based on careful exploration of the contents that deliver a clear and exhaustive understanding of scientific workflows management and scheduling techniques in cloud computing. Boolean functions (NOT, AND, OR) were used, with appropriate strings by synonyms and alternative spellings to dig deep into the hundreds of articles [39]. A combination of keywords were used, as in the following query:

*(“scientific workflows management” **AND** “cloud computing” **AND** (“workflowsim” **OR** “cloudsim” **OR** “real testbed” **OR** “simulation”))*
***OR** (“scientific workflows scheduling” **AND** “cloud computing”)*
***OR** (“management” **AND** “cloud computing”)*
***OR** (“scheduling” **AND** “cloud computing”)*
***OR** (“management” **AND** “scheduling” **AND** “cloud computing”)*
***OR** (“scientific data management” **AND** “scientific data scheduling” **AND** “cloud computing”)*
***OR** (“scientific applications management” **AND** “scientific applications scheduling” **AND** “cloud computing”)*
***OR** (“scientific tasks management” **AND** “scientific tasks scheduling” **AND** “cloud computing”)*

After the first filtration process, a re-filtering was conducted to obtain a set of articles more precisely related to the review scope, to ensure that there were no papers neglected in our review, as in the statements below:

The list of the articles included in this survey is based on the quality assessment checklist (QAC) as specified by [39].

((“scientific workflows management” ***OR*** *“scientific workflows scheduling”)*
AND *(“scientific data management”* ***OR*** *“scientific data scheduling”)*
AND *(“scientific applications management”* ***OR*** *“scientific application scheduling”)*
AND *(“scientific tasks management”* ***OR*** *“scientific tasks scheduling”))*

In this way, the list attain the scope of the review, as each of the article in the list met the following criteria:

- Does the research paper achieve scientific workflows management and scheduling?
- Does the research paper obviously identify the methodology?
- Does the research methodology use available tools to re-implement (simulation or real world environment)?
- Is the study analysis accomplished properly?

If “yes”, the articles will be selected after meeting the following criteria:

- Every article that met the criteria listed in the keywords box will be selected first.
- After filtering the article by reading the abstract, it will be listed in the final set.
- Articles related to scientific workflows management and scheduling will be included.

III. EXISTING SURVEYS

Although there are a number of surveys regarding resource management in cloud computing. But to the best of our knowledge, till now no survey is written more specifically on scientific workflows management in cloud.

One of the most recent survey works on taxonomy, prospects, and challenges in resource management in cloud computing was presented in [38]. The authors in this survey highlighted the resource management techniques with taxonomy, working mechanism, and problem in the working mechanism of resource management techniques. The authors also write out the most frequently used performance evaluation parameters for resource management techniques in cloud computing. In this survey, the various design goals to design the resource management techniques were suggested/recommended and then open research issues/challenges regarding resource management in cloud computing were pointed out.

Another survey in cloud on resource management for IaaS was presented in [40]. In this survey the authors briefly explain IaaS with its uses as: (a) need base provision of shared resources, (b) provisions of detail like on demand server images, storage, and information regarding other available resources, and (c) provision of server infrastructure’s full control. Similarly, issues in Iaas as: (a) multi-tenancy and virtualization, (b) management of resources, (c) management of network infrastructure, (d) security issues, and (e) data management, were also highlighted. Moreover, taxonomy on resource types, resource management problems in IaaS, possible solutions for resource management problems, and tools and technologies used for resource management in IaaS was discussed by the authors.

A survey on scientific workflows management as data intensive application management was presented in [41]. The authors discussed the techniques used for data intensive scientific workflows along with five layer functional architecture of scientific workflow management systems (SWfMSs). Parallelization technique’s taxonomy and comparative study on scheduling algorithms for scientific workflows was also presented. Finally, the authors concluded the discussion by pointing out the issues for improving the scientific workflows execution.

As scientific workflows are data intensive applications and thus, such types of applications have a number of computing challenges which are highlighted in [42]. The authors also discussed the surveys on SWfMSs written from 2013 to July, 2015. Finally, the design goals for future development of data intensive applications and techniques were proposed.

To be more specific, surveys on scheduling the scientific workflows in cloud computing were written in [43], [44] and [45]. In these works, the comprehensive analysis of scientific workflows scheduling was presented. The authors not only surveyed the most recent scheduling work for workflows in cloud computing but also introduced the various trends of analysis for workflow scheduling. Taxonomies on cloud workflow scheduling for existing studies were also presented and the authors identified the challenges regarding cloud workflows scheduling. The workflows scheduling techniques and problems were also discussed.

A review on cost optimization approaches, its classification and open issues for scientific workflows in cloud computing was presented in [46]. The authors in this paper focused the cost optimization problems for scientific workflows in cloud computing. To achieve the goal of cost optimization, the authors classified the overall work into three categories. Firstly, they classified and discussed the relevant cost optimization approaches. Secondly, the cost parameters were classified into temporal cost, monetary cost, and, the parameters of cost based on scheduling stages i.e., post-scheduling, during scheduling, and pre-scheduling. And finally, the authors find out the correlation between the profitability to service provider/consumer and the cost parameters.

Table 1 shows an overview of existing surveys regarding resource management in cloud computing.

IV. SCIENTIFIC WORKFLOWS

A scientific workflow is a collection of component functions with predefined order of execution and having several dependencies at various stages. In this section various scientific workflow applications and their characteristics are discussed. These applications involve the workflows of montage (astronomy), CyberShake (earthquake science), Epigenomics

TABLE 1. An overview of resource management surveys in cloud computing.

Reference	Survey Domain	Resource Management	Service Category		Orientation		Cloud Services and Resources			Environment	
			Scientific	Business	Data-oriented	Task	IaaS	PaaS	SaaS	Cloud Computing	Grid Computing
[38]	Resource management for Infrastructure as a Service in cloud computing	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
[40]	Data-Intensive Scientific Workflow Management	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗
[41]	Contemporary challenges for data-intensive scientific workflow management systems	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗
[42]	Towards workflow scheduling in cloud computing	✓	✓	✗	✓	✗	✓	✓	✗	✓	✗
[43]	Scheduling Workflows in Cloud Environment	✓	✓	✗	✓	✗	✓	✓	✗	✓	✗
[44]	Workflow scheduling problem and techniques in the cloud	✓	✓	✗	✓	✗	✓	✓	✗	✓	✗
[45]	Cost optimization approaches for scientific workflow scheduling in cloud and grid computing	✓	✓	✗	✓	✗	✓	✓	✗	✓	✓
[46]	Dynamic energy-aware scheduling for parallel task-based application in cloud computing	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗

(biology), Laser Interferometer Gravitational Wave Observatory (LIGO) related to gravitational physics and SIPHT (biology) [37]. Figure 3 shows a taxonomy of scientific workflows in respect of scientific workflow applications and characteristics.

A. SCIENTIFIC WORKFLOW APPLICATIONS

Scientific workflow applications involve the workflows of montage (astronomy), CyberShake (earthquake science), Epigenomics (biology), Laser Interferometer Gravitational Wave Observatory (LIGO) related to gravitational physics and SIPHT (biology) [37]. There are multiple instances of each workflow application. Each instance of workflow is represented by a circle. The instances of each workflow application are processed and executed at several levels. These levels involve multiple proceedings including Aggregation, Distribution, Re-distribution, Pipelined and Parallelism as labelled in Figure 4. Similar is the case for rest of figures i.e. from Figure 5 to 8. The well-known scientific workflows related to various fields of science are given below:

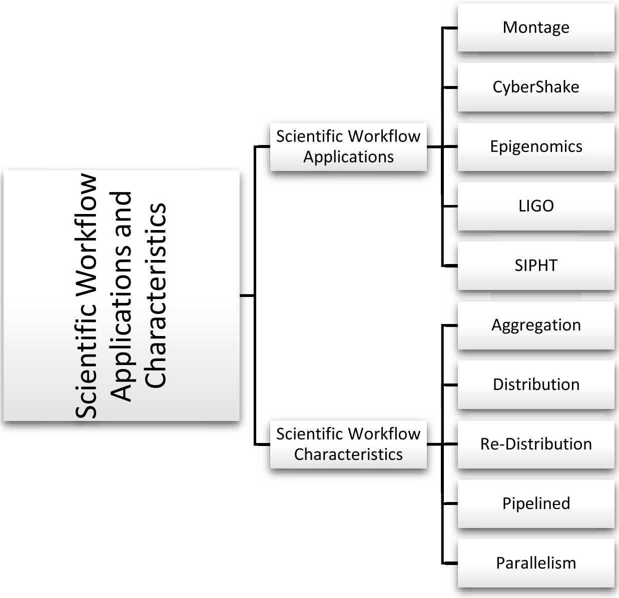

Scientific Workflow Applications and Characteristics
Scientific Workflow Applications
Montage
CyberShake
Epigenomics
LIGO
SIPHT
Scientific Workflow Characteristics
Aggregation
Distribution
Re-Distribution
Pipelined
Parallelism

FIGURE 3. A taxonomy of scientific workflow applications and characteristics.

1) MONTAGE

Montage is a type of scientific workflow that can be used to produce custom mosaics of the sky. It uses input images in the Flexible Image Transport System (FITS) format. The geometry of the input images is used to calculate the geometry of output and then final mosaic is produced. The input images

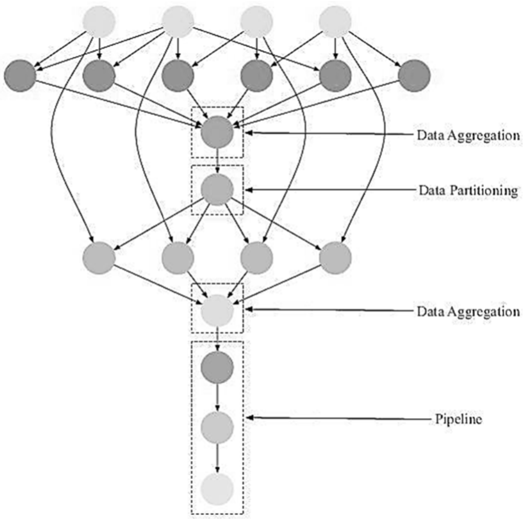


FIGURE 4. A montage workflow [37].

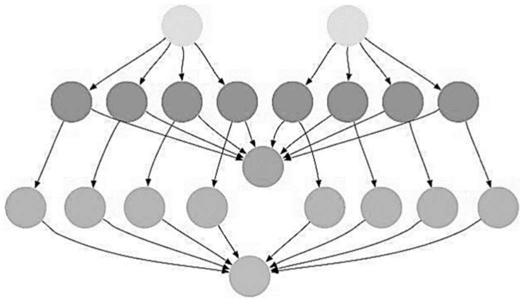


FIGURE 5. CyberShake workflow [37].

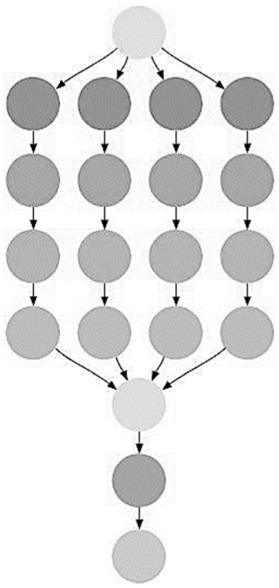


FIGURE 6. Epigenomics workflow [37].

are re-projected with the same spatial scale and rotation. The background emissions of all the images are corrected at the same level. Finally, the corrected and re-projected images are added to form the final mosaic. Figure 4 shows the architecture of montage workflow.

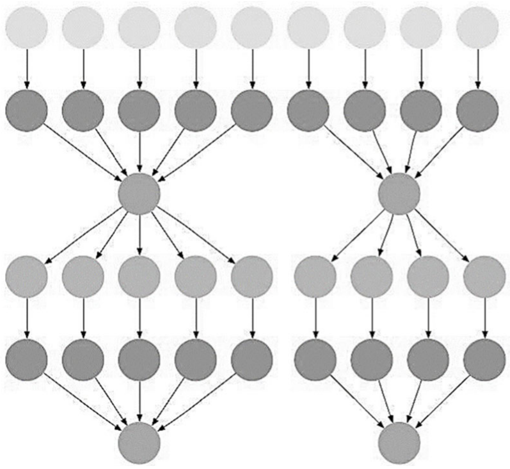


FIGURE 7. LIGO workflow [37].

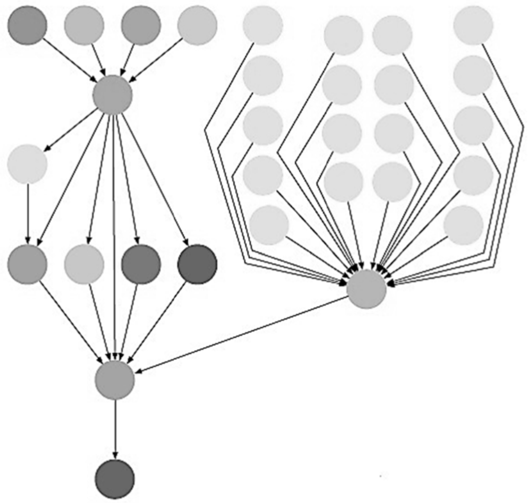


FIGURE 8. SIPHT workflow [37].

2) CYBERSHAKE

CyberShake is a type of workflow that characterizes the hazards in a specified region. It uses Probabilistic Seismic Hazard Analysis (PSHA) technique to characterize the same. In this technique, a region is specified then a finite difference simulation is performed that generates Strain Green Tensors (SGTs). Synthetic seismograms are calculated from SGT data for each of the ruptures so predicted previously. Thereafter, probabilistic hazard curves and spectral acceleration are generated. More than 800,000 jobs have been executed totally in CyberShake workflow to obtain the results. Figure 5 shows the architecture of CyberShake workflow.

3) EPIGENOMICS

Epigenomics workflow is a data processing pipeline that is used for execution of the genome sequencing operations automatically. After the generation of DNA sequence, it is split into multiple chunks which are to be operated parallel. The data of each chunk is then converted into file format. Afterwards, noise and contaminate sequence is filtered out and then mapped the sequences into the correct location in a given genome. It also generates a global map and identifies the density of sequence in the genome at each position. This type of workflow is used at Epigenomic Center for the production of histone modification data and DNA methylation. Figure 6 shows Epigenomics workflow.

4) LIGO

The Laser Interferometer Gravitational Wave Observatory (LIGO) is a type of workflow that is used to detect gravitational waves produced during various events as per Einstein's general relative theory. LIGO is used to analyse coalescing of compact binary systems data like black holes and binary neutron stars. Figure 7 shows LIGO workflow.

5) SIPHT

SIPHT is a program that is used to predict and annotate the genes and bacterial replicons. It involves multiple programs that are required to be executed in proper order. Figure 8 shows SIPHT workflow.

B. CONCEPTS AND TERMINOLOGIES USED FOR SCIENTIFIC WORKFLOWS

The scientific workflow is described as Directed Acyclic Graph (DAG) where a set of vertices represent individual tasks and a set of edges represent data dependencies between the tasks. We stress upon the multiple instances of a workflow, while each instance of workflow is represented by a circle. These instances are processed and executed at several levels, while these levels involve multiple proceedings i.e. Aggregation, Distribution, Re-distribution, Pipelined and Parallelism. We used basic concepts and terminologies of workflow that are used in [37] and which are given below.

1) AGGREGATION

When multiple instances of a workflow processed and outcome is a single instance then it is called aggregation. Figure 9 shows the process of aggregation i.e. four instances of a workflow combine to form a single instance of workflow.

2) DISTRIBUTION

When a single instance of a workflow processed and outcome is a combination of multiple instances then it is called distribution. Figure 10 shows the process of distribution i.e. single instance of a workflow distributed to a combination of four instances of a workflow.

3) RE-DISTRIBUTION

When multiple instances of a workflow processed to form a single instance and then it is further processed, while the final outcome is also a combination of multiple instances then it is called redistribution. Figure 11 shows the process of redistribution i.e. four instances of a workflow combine to form a single outcome and then distribute to form a combination of four instances of a workflow.

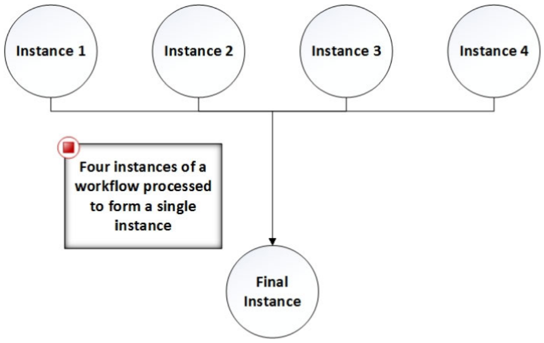

Instance 1
Instance 2
Instance 3
Instance 4
Four instances of a workflow processed to form a single instance
Final Instance

FIGURE 9. Aggregation (Four instances combine to form single instance).

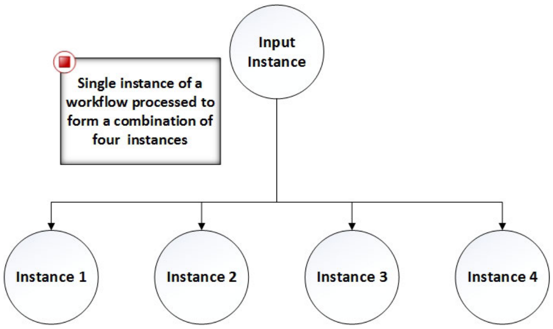

Input Instance
Single instance of a workflow processed to form a combination of four instances
Instance 1
Instance 2
Instance 3
Instance 4

FIGURE 10. Distribution (Single instance distributed to form a combination of four instances).

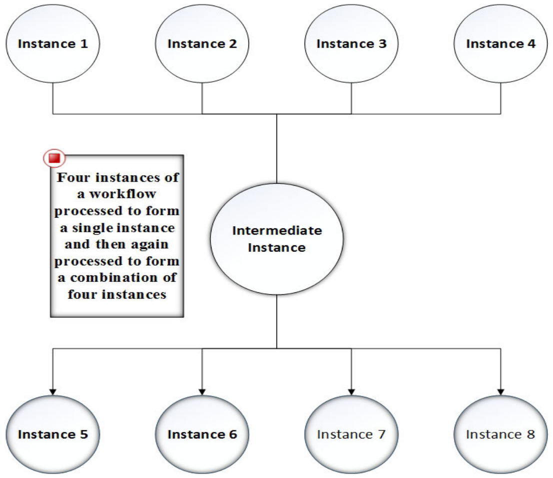

Instance 1
Instance 2
Instance 3
Instance 4
Four instances of a workflow processed to form a single instance and then again processed to form a combination of four instances
Intermediate Instance
Instance 5
Instance 6
Instance 7
Instance 8

FIGURE 11. Shows Re-Distribution (four instances to form single instance and then distributed to form a combination of four instances of workflow).

4) PIPELINED

When a single instance of a workflow is processed to form another single instance and then it further processed, while the final outcome is also a single instance then it is called pipelined. Figure 12 shows the process of pipelined i.e. single instance processed to form another single instance, while final outcome is also a single instance of a workflow.

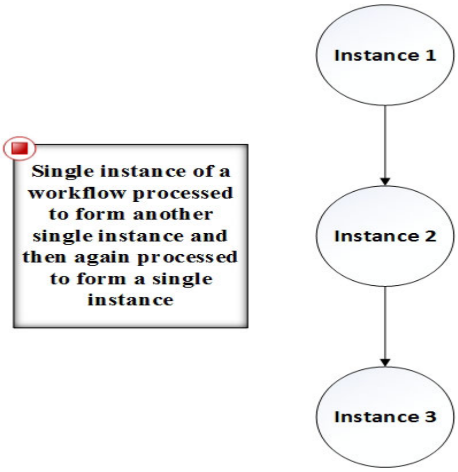

Instance 1
Instance 2
Instance 3
Single instance of a workflow processed to form another single instance and then again processed to form a single instance

FIGURE 12. Shows pipelined (single instance processed to form another single instance, while final outcome is also a single instance of a workflow).

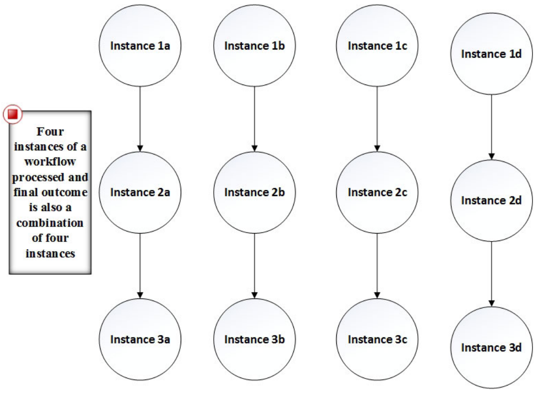


FIGURE 13. Shows parallelism (four instances processed and final outcome is also a combination of four instances of a workflow).

5) PARALLELISM

When multiple instances of a workflow simultaneously processed to form multiple instances then it is called parallelism. Figure 13 shows the process of parallelism i.e. four instances of a workflow processed to form four instances.

V. SCIENTIFIC WORKFLOWS SCHEDULING AND MANAGEMENT TECHNIQUES

In this section we briefly mentioned the taxonomy of scientific workflows management techniques in terms of workflow scheduling, fault tolerance and energy efficiency. Figure 14 shows a taxonomy of scientific workflows scheduling and management techniques. The techniques are classified into multiple categories based on the research description for scientific workflow applications.

A. SCIENTIFIC WORKFLOWS RESOURCE SCHEDULING TECHNIQUES

Scheduling is one of the major components not only for scientific workflows but also for execution of rest of the applications on clouds. Scientific workflows scheduling techniques, schedule the tasks of scientific workflows on best available resources, provide quality of service's parameters like time and cost as well as ensure the completion of tasks on a particular resource. Following are the most modern, commonly and currently implemented scheduling techniques used for execution of scientific workflows:

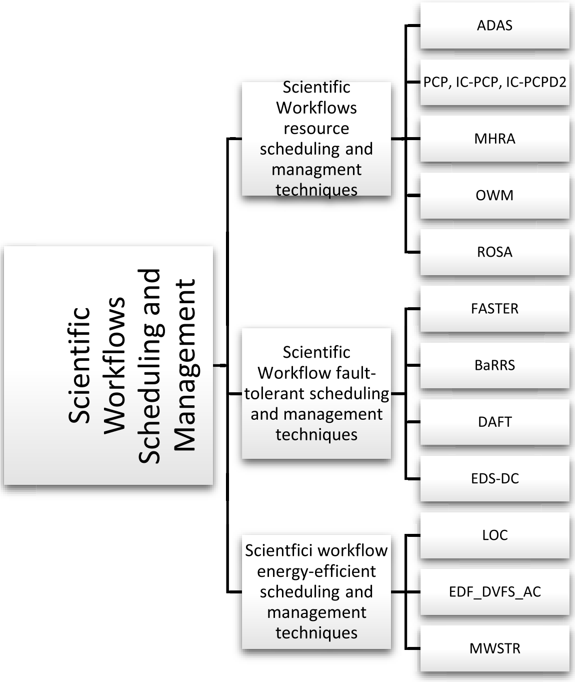

Scientific Workflows Scheduling and Management
Scientific Workflows resource scheduling and managment techniques
Scientific Workflow fault-tolerant scheduling and management techniques
Scientfici workflow energy-efficient scheduling and management techniques
ADAS
PCP, IC-PCP, IC-PCPD2
MHRA
OWM
ROSA
FASTER
BaRRS
DAFT
EDS-DC
LOC
EDF_DVFS_AC
MWSTR

FIGURE 14. A taxonomy of scientific workflows scheduling and management techniques.

Adaptive Data-aware scheduling (ADAS) is a scheduling technique for scientific workflows in cloud environment, which consist of two stages i.e., (a) set up stage, and (b) run-time stage [11]. The set-up stage is used to build the clusters for dataset and workflow tasks. The run-time stage is used to execute the workflows in overlapped form. The set-up stage is further divided into two phases. In the first phase, the initial clusters for the workflow tasks are built up through a matrix based approach. While in the second phase, the cluster of datasets/tasks is to be formed through the quality of profitable scheduling. ADAS is the best suitable scheduling strategy for execution of scientific workflows in a cloud based environment as it reduces the make-span for communication-intensive workflow applications and for those types of workflow applications that have a wider degree of parallelism. However, the scheduling technique is limited only to the extent of managing data-intensiveness of scientific workflows. Whereas, the special features of scientific workflows including integration, dis-integration, pipelined, and parallelism have not been considered. The consideration of special features for scientific workflows not only helpful for efficient resource management but also useful for implementation of effective fault tolerant techniques.

Two deadline constrained based scheduling algorithms for scientific workflows were proposed in [12] which were based on Partial Critical Paths (PCP). The algorithms were specifically designed for Infrastructure as a Service (IaaS). These algorithms are (a) IaaS Cloud-Partial Critical Paths (IC-PCP), and (b) IaaS Cloud-Partial Critical Paths with Deadline Distribution (IC-PCPD2). Both the algorithms were aimed to be designed for minimizing the total execution cost, while the user defined deadline is also satisfied. IC-PCP is the one phase scheduling algorithm in which it schedules each workflow task and submits to the resources for execution. The one phase of the IC-PCP is also called the planning phase. Whereas, the IC-PCPD2 is a scheduling algorithm which is based on two phases i.e., (a) the planning phase, and (b) the deadline distribution phase. In the planning phase, the workflow tasks are to be scheduled, while in the deadline distribution phase, it assigns the deadlines to all the workflows tasks which are to be scheduled. Although both the proposed algorithms were efficient in terms of completing the workflow tasks within the cost and deadline constraints. However, these algorithms will be more effective, if a budget driven mechanism is added along with a deadline distribution mechanism.

In [47], a multi-heuristic resource allocation (MHRA) algorithm was proposed, which is a faster search algorithm that works locally in respect of partial solutions. There are two phases of the proposed algorithm. In the first phase, a set of heuristic rules are combined in order to rank an eligible group of parallel tasks for a provided Directed Acyclic Graph (DAG). The rankings were done on the basis of execution time, the amount of data transfer and the number of tasks predecessors or successors. In the second phase, a set of important factors used for resource algorithms were combined. These factors are then used to find the best position of a specific task in the cloud for minimizing the energy consumption and make-span. The proposed algorithm provides an effective real-time scheduling solution with a significant time scale. However, the mechanism will be more effective, if it is implemented based on the given budget.

In [48], an online scheduling approach OWM (Online Workflow Management) was proposed for multiple mixed-parallel workflows in grid environments. The proposed approach was evaluated with a series of simulation experiments. The simulation results reveal that the proposed approach delivers good performance and outperforms other methods under various workloads. The work so presented is outdated in a sense that it was analysed in a grid environment with limited form of real-time scientific workflow applications.

In [49], an unceRtainty-aware Online Scheduling Algorithm (ROSA) was presented in order to schedule dynamic and multiple workflows under the constraint of deadlines. The ROSA strategy efficiently integrates both the proactive and reactive mechanisms. The results show that the ROSA performs better than the existing five algorithms with respect to costs (up to 56%), deviation (up to 70%), resource utilization (up to 37%) and fairness (up to 37%). The ROSA strategy can further be improved by considering the data and compute awareness of scientific workflows.

B. SCIENTIFIC WORKFLOWS FAULT-TOLERANT SCHEDULING TECHNIQUES

There is a large amount of data involved in scientific workflows and for which, the execution is completed at different aspects including pipelined, parallelism, integration and disintegration. In most of the cases when execution of scientific workflows is completed at a bottleneck, the importance of fault tolerance could not be denied. Following are the most modern, commonly and currently implemented scheduling techniques used for execution of scientific workflows:

In [50], a fault tolerant based scheduling algorithm known as FASTER (**F**ault-toler**A**nt **S**cheduling algorithm for real-**T**ime sci**E**ntific wo**R**kflows) was presented. It has three key features. Firstly, it incorporates the overlapping of tasks and fully utilizes the idle resources by employing a backward shifting method. Secondly, it provides resources for a burst of workflows by horizontal and vertical scaling-up methods. Thirdly, it avoids unnecessary use of resources (due to fluctuated workflow requests) by scaling-down mechanism. The FASTER strategy was evaluated with synthetic workflows which are collected from the real business and scientific applications. The FASTER strategy provides real-time based fault tolerance mechanism in the virtualized cloud for scientific workflows. It will be more effective, if the failures of tasks are managed by considering the special features of scientific workflows including integration, dis-integration, pipelined, and parallelism.

In [51], a Balanced and file Reuse-Replication Scheduling (BaRRS) algorithm was presented in order to efficiently schedule scientific workflows in cloud computing. The BaRRS strategy in order to balance the utilization of systems through parallelization makes multiple sub-workflows from a single scientific workflow application. It provides the mechanism of replication and date reuse technique to optimize the data which is required to be transferred at runtime. It also performs trade-off analysis between monetary cost and execution time of running scientific workflows for the purpose of finding the best solution. The BaRRS strategy is limited to the extent of replication and data reuse technique and as such the failures of tasks in scientific workflows are kept intact.

In [1], the basic concepts and definitions of error, fault and failure in cloud computing are provided. The principle of high fault tolerance objectives were also analysed systematically for large scale computing environments. Subsequently, a dynamic adaptive fault tolerance (DAFT) strategy was presented in [1]. It provides the analysis of different failure rates and builds dynamic fault tolerance models i.e., adaptive check-pointing and adaptive replication from two existing fault tolerant models. The proposed strategy was evaluated in a large scale cloud computing environment using CloudSim [52] under various conditions such as; fault tolerance degree, response time and fault tolerance overhead. However, the algorithm is generic in nature and is not specifically designed for scientific workflows by considering special features including integration, dis-integration, pipelined, and parallelism.

In [53], an Enhanced Data-oriented Scheduling strategy with Dynamic clustering fault-tolerant technique (EDS-DC) was presented. The proposed technique was specifically designed for scientific workflows that provides the data-oriented scheduling mechanism of scientific workflows. The proposed technique also provides the dynamic-clustering fault-tolerant mechanism for scheduling scientific workflows. The proposed strategy considers make-span, computational cost, deadline, budget and SLA violation as performance evaluation parameters. The simulation tool WorkflowSim [54] was used and simulation results show that the proposed technique outperformed as compared with the existing ones. The EDS-DC is designed by considering the characteristic of data-intensiveness of scientific workflows, however, characteristic of compute-intensiveness of scientific workflows still kept intact there.

C. SCIENTIFIC WORKFLOWS ENERGY EFFICIENT SCHEDULING TECHNIQUES

Scientific workflow applications are data and compute intensive applications that consume high amounts of energy during execution. As such, for the purpose of execution and management of scientific workflows, energy-efficient scheduling techniques are also the need of the modern cloud computing environment. The most frequently applied energy-efficient scheduling techniques used for execution of scientific workflows are given below:

In [15], an approach LOC (local storage based hot metadata management) was proposed in which hot metadata i.e., frequently accessed metadata was identified and exploited for scientific workflows scheduling in multisite cloud. The proposed approach was energy efficient and it was implemented within a scientific workflow management system. The proposed approach also reduces the execution time of parallel jobs highly up to 64% and as per whole scientific workflows, it reduces up to 55%. However, in case of integration, dis-integration, and pipelined of tasks of scientific workflows, the approach was not analysed. In [55], an analysis of power and energy consumption measurements was presented. The analysis reveals that I/O operations significantly affected power consumption, whereas, the CPU utilization does not have much impact on power consumption. In [17], two production scientific workflows were profiled on a distributed platform instrumented with power consumption parameters. After analysis of measurements of power and energy consumption, a power consumption model was proposed. It was analysed and seconded [55] through power consumption model that I/O operations significantly affected the power consumption instead of CPU utilization.

In [56], a Quality of Service (QoS) aware scheduling strategy for real-time scientific workflows applications in cloud computing was presented which was not only energy-efficient but also cost-effective. The proposed approach is referred to as, "Earliest Deadline First with Dynamic Voltage and Frequency Scaling and Approximate Computations" (EDF_DVFS_AC). The proposed strategy applies "Dynamic Voltage and Frequency Scaling" (DVFS) upon heterogeneous multicore processing units and approximate computations for the purpose of filling the schedule gaps. The proposed strategy also considers the input error during processing time. The authors in the proposed strategy tried to cover the trade-off between timeliness and energy efficiency through the result precision. The authors also maintained the jobs completion rate at acceptable standard and the mandatory cost at reasonable level applied for the execution of jobs. The authors signify the proposed strategy through simulation by comparing the result with some existing approaches. However, the approach was generic in nature and not considered diverse in the nature of tasks in scientific workflows with integration, dis-integration, pipelined and parallelism.

In [57], a Multiple-Workflows-Slack-Time-Reclaiming (MWSTR) algorithm was proposed in order to reclaim slack time using DVFS technology. The MWSTR algorithm preserved the precedence constraints of multiple workflows. From the experimental results, the authors draw the conclusion that interleaving workflows lead to a better average tradeoff when scheduling multiple workflows. The MWSTR algorithm is limited only to the extent of reclaiming the slack time and reducing energy consumption and as such there is no provision of workflows management and fault tolerance.

VI. PERFORMANCE EVALUATION PARAMETERS

In order to evaluate the credibility of the designed technique, various performance evaluation parameters are used. With the help of performance evaluation parameters, the original performance of the system is being compared with the expected one and the results of the proposed technique determine its success. In the current section, some currently used and important parameters are presented to evaluate scientific workflow management techniques.

A. MAKE-SPAN

The total time taken to complete a batch of tasks is referred to as a make-span. In case of execution of scientific workflows, it is the total time required for execution of complete scientific workflow [58]. It is denoted by M_{span} and can be evaluated with the help of Equation (1).

$$M_{span} = F_{time} - S_{time} \tag{1}$$

where, F_{time} is the finish time and S_{time} is the start time, when a scientific workflow is executed.

B. DEADLINE

The predefined completion time for execution of a batch of tasks is referred to as deadline. In case of execution of scientific workflows, it is predefined completion time for execution of complete scientific workflow [59]. It is denoted by D_L and can be evaluated with the help of Equation (2).

$$D_L = T_{computation} + T_{communication} + T_{overhead} \tag{2}$$

where, $T_{computation}$ is time required for computation, $T_{communication}$ is time required for communication, and $T_{overhead}$ is the extra time consumed, when failed tasks of scientific workflow are executed.

C. EXECUTION COST

The budget taken to finish a batch of tasks is referred to as Execution cost. In case of execution of scientific workflows, it is the total budget required, when a complete scientific workflow is executed [58]. It is denoted by $Execution_{cost}$ and can be evaluated with the help of Equation (3).

$$Execution_{cost} = F.T_{cost} - S.T_{cost} \tag{3}$$

where, $F.T_{cost}$ is the cost on finish time and $S.T_{cost}$ is the cost on start time of execution of scientific workflow.

Similarly, cost for a single task of scientific workflow execution is denoted by $Task_{cost}$ and can be evaluated with the help of Equation (4).

$$Task_{cost} = Processing_{cost} + Memory_{cost} + Storage_{cost} + Bandwidth_{cost} \tag{4}$$

where, $Processing_{cost}$ is processing cost, $Memory_{cost}$ is memory cost, $Storage_{cost}$ is storage cost, and $Bandwidth_{cost}$ is bandwidth cost, when a scientific workflow is executed.

D. BUDGET

The total available financial resources in order to execute a batch of tasks is referred as budget [59]. In case of execution of scientific workflows, it is predefined cost required, when a complete scientific workflow is executed. It is denoted by B and can be evaluated with the help of equation (5).

$$B = C_{computation} + C_{communication} + C_{overhead} \tag{5}$$

where, $C_{computation}$ is computation cost, $C_{communication}$ is communication cost and $C_{overhead}$ is the overhead, when extra cost consumed on re-execution of failed tasks.

E. SLA VIOLATION

The terminology used when utilization of resources of a system exceeds a given amount of resources is called Service Level Agreement (SLA) Violation. SLA is also violated when required resources as agreed upon between the parties were not fully given to the customer by the provider. In case of scientific workflows execution, it is a term used when the cost of the system is exceeded from a given budget or makespan is exceeded from the deadline [59]. It is denoted by $SLA_{violation}$. The equations (6) and (7) shows the condition for SLA violation [38].

$$SLA_{violation} = SLAV_{time} \tag{6}$$

$$SLA_{violation} = SLAV_{cost} \tag{7}$$

where, $SLAV_{time}$ is SLA violation due to increase of time per active hours and $SLAV_{cost}$ is SLA violation due to increase of cost per active budget.

F. ENERGY CONSUMPTION

The power consumed on processing a batch of tasks is called energy consumption [60]. In case of execution of scientific workflows, energy consumption is the power consumed, when a complete scientific workflow is executed. It is denoted by $Energy_{consumption}$ and can be evaluated with the help of Equation (8).

$$Energy_{consumption} = \sum_{i=1}^{n} (Energy_{trans(i)} + Energy_{exe(i)}) \tag{8}$$

where, $Energy_{trans}$ is energy consumed during transmission of tasks and $Energy_{exe}$ is the energy consumption on execution of tasks.

G. NETWORK USAGE

The utilization of a network in terms of load is referred to as network usage [61]. In case of scientific workflows, the network usage is the utilization of network resources, when a scientific workflow is executed. The network usage is denoted by $Network_{usage}$ and can be evaluated with the help of Equation (9).

$$Network_{usage} = \sum_{i=1}^{n} (Load_{node(i)} \times Transmission_{node(i)}) \tag{9}$$

where, $Load_{node}$ is the network load on each node and $Transmission_{node(i)}$ is the transmission of each node with other nodes.

H. NETWORK DELAY

The sum of total delay during execution of a batch of tasks is called network delay. It includes processing delay, transmission delay, and computation delay [60]. In the case of scientific workflow, the network delay is the sum of processing delay, transmission delay and computation delay, when a scientific workflow is executed. It is denoted by $Network_{Delay}$ and can be evaluated with the help of Equation (10).

$$Network_{Delay} = D_p + D_t + D_c \tag{10}$$

where, D_p represents the processing delay, D_t represents the transmission delay, and D_c represents the computation delay.

I. THROUGHPUT

The batch of tasks completed in a certain period of time is called throughput. It is often used as a measure of efficiency in cloud computing since, in cloud computing the tasks are executed on remote resources [38]. In case of scientific workflows, it is a certain time period when total tasks of a scientific workflow are executed. It is denoted by $Througput$ and can be evaluated with the help of Equation (11).

$$Througput = T_{total} - T_{Remaining} \tag{11}$$

where, T_{total} represents the total tasks of a scientific worklfow and $T_{Remaining}$ represents the remaining tasks of a scientific workflow.

VII. PERFORMANCE EVALUATION PLATFORMS

This section provides the description of simulation tools commonly used for evaluation of workflow management and scheduling strategies designed in cloud computing for scientific workflow applications.

A. CLOUDSIM

Nowadays, cloud computing requires complex development and composition. It is difficult to analyze the performance of cloud applications and resource models in case of different systems and varying configurations. The CloudSim [52] solves this problem by modelling components like virtual machines, data centers, network topologies, federated cloud environments, computational resources, scheduling and provisioning. It simulates the cloud infrastructure and application services in an extensible way as well as takes less amount of effort and time. CloudSim enables its users to test the performance of new developed applications. It provides an easy and controlled environment. The CloudSim layer supports the simulation and modeling of memory, storage, virtual machines and network bandwidth of cloud environment. It provides facility of system state motoring, managing the execution of applications and allocation of hosts to VMs. The user code layer provides basic entities i.e. number of VMs and their specifications, number of users, types of applications and policies of scheduling. CloudSim provides numerous facilities according to the needs of the user. The main features of CloudSim are as follows:

Regions: The user can model the geographic regions in which providers of cloud services can make allocation of resources to their consumers. There are six regions as we have six continents in the world.

Data centers: The modeling of infrastructure services can be done easily. It has numerous hosts and servers that may be homogeneous or heterogeneous, depending upon hardware configurations. The configuration of resources of datacenters can be modeled. All characteristics of datacenters can be modeled and viewed easily.

The user base: The users can also be modeled to analyze the traffic for the simulation. The modeling of users can be done by taking them as a single unit or group.

Hosts: The modeling of physical resources such as computation or storage can be done easily.

Cloudlet: The set of user requests can be specified easily. The requests have application ID, user bases and their names, input files, size of execution command (request command) and output files.

Service broker: The service broker can be selected whose task is deciding the data center. The selected data center provides the requested services by user bases.

VM (Virtual Machine) scheduler: It models the time or space shared, scheduling a policy to allocate processor cores to VMs. The VM scheduler is responsible for modeling of shared space and time, allocation policy of processor cores to VMs.

VM allocation policy: The modeling of allocation mechanism of VMs (Virtual Machines) to hosts is done by defining the policies first.

B. CLOUDANALYST

CloudAnalyst [62] was introduced in 2009 at University of Melbourne, aimed for supporting the evaluation of social networks tools based on geographic distribution of data centers and clients. In this tool, we can obtain the load on the data centers by characterizing the users.

1. It is a requirement of simulation to model the required infrastructure as well as software application in a reliable language that best defines the operations of the simulator. Most of the simulation tools require programming exercises instead of experimentation exercises.
2. The CloudAnalyst was especially designed to separate the experiments of simulation from programming exercises. It is not necessary for a modeler to be a very competent programmer. CloudAnalyst allows modellers to work on complexities of simulations rather than spending their attention on programming mechanics by using simulation toolkits.
3. This specialty of CloudAnalyst allows recurrent simulation practices as well as series of simulation exercises. This can be done by making slight changes in parameters according to needs.
4. CloudAnalyst is a GUI based simulator, an extension of CloudSim with numerous capabilities available in an easy way.
5. It can be used for behavioural examinations of large-scale internet applications with cloud environments.
6. Configurations regarding simulations can be saved in the form of xml files and results can be exported in PDF format.
7. Hence the focus of this tool is more on performing simulations and modelling instead of programming exercises.

C. GROUNDSIM

GroundSim (Java Based simulation toolkit) [63] is especially designed for scientific applications for *event-based* simulations on either Grid as well as Cloud environments. Unlike other tools, GroundSim does not acquire multi-threads, it works on a single thread. It uses a discrete-event simulation toolkit that is why it provides best performance as compared to process-based approaches. GroundSim is able to simulate Grid and Cloud resources and application execution for Workflow execution, Provisioning, Resource Management and scheduling. To tackle the complex simulation setups, it provides an exclusive set of features whether it would be an easy job execution on leased computing resources or high-level calculation like collective costs and total load on resources.

The desired simulations can be configured using parameters. These parameters are extendable and reusable as well. There are various probability distribution packages available to handle the failures occurring while simulating complex environments. There is a main class namely SimEngine implementing the time advance algorithm, future event list (FEL), the clock, saves the recently registered entities that is why called *registered entities* used tracking during simulation practices. Modeller has three choices to start a new simulation: Firstly, to run the simulation till the last event in future event list (FEL). Secondly, to define a time slice for simulation and thirdly, to run the simulation until the randomly defined time stop and exit the SimEngine.

There are some basic statistical and analytical views available in GroundSim to allow the modeller to write further complex analysis. This tool mainly supports modelling of Grid and Cloud computations and network resources, costs modelling, background loads, failure integrations, file transfer and job submission.

D. WORKFLOWSIM

WorkflowSim [54] is an extension of an existing well known simulator known as CloudSim [52]. WorkflowSim provides a higher layer of workflow management. In WorkflowSim, it is indicated that if we ignore system overheads and failure in simulating scientific workflows, it could cause significant inaccuracies in the predicted workflow execution time.

Scientific workflows can be composed upon a large number of tasks and their execution requires many complex modules and software. Existing simulators such as CloudSim fail to provide fine granularity simulations of workflows. For example, they lack the support of task clustering, which is a popular technique that merges small tasks into a large job to reduce task execution overheads. The simulation of task clustering requires two layers of execution model, on both task and job levels. Multiple layers were added on top of the existing workflow scheduling layer of CloudSim, which include the Workflow Mapper, the Workflow Engine, the Clustering Engine, the Failure Generator, and the Failure Monitor.

E. IFOGSIM

The simulation tool iFogSim [64] is one of the latest simulators related to distributed systems. The simulator iFogSim is used to evaluate network edge algorithms. It is an open source java based simulation tool. Nowadays, IoT is an emerging technology in the field of computer science in which scientists are trying to connect everything with the internet. Home appliances and equipment are controlled by application programs through the internet. When we connect all the equipment with the internet then we need space and processing power. If we provide the requirements for IoT from centralized networks then the network can be congested as huge traffic will be on network. So, a new mechanism of Fog computing is introduced [9] in which cloud services are placed on the edge of the network to make latency low and avoid congestion. In Fog computing, the iFogSim is used to simulate different Fog models and solve them. It also provides a GUI facility in which researchers can make visual datacenters and edge network nodes. The iFogSim is suitable for simulating edge node cloud services, however, it is complex because edge node cloud services is a sub-part of cloud.

F. IOTSIM

IOTSim [65] is illuminated by the works of CloudSim. It is designed through the layered architecture. It also provides support for big data processing frameworks. It consists of:

- CloudSim Core Simulation Engine Layer which is the bottommost layer and that supports core functionalities.
- Cloudsim Simulation Layer that provides support for modelling and simulation of virtualized Cloud-based datacentre environments.
- Storage Layer that supports modelling different kind of storage such as Amazon S3, Azure, Blob Storage, and HDFS.
- Big Data Processing Layer that includes two sub-layers. MapReduce sub-layer is to support applications where a batch-oriented data processing paradigm is required while Streaming Computing sub-layer that aims to support applications that need a real-time processing environment.
- User Code Layer which is the top-most layer that discovers basic entities for hosts (number of machines and their specification), VMs, number of users and their application types, IoT-based applications' configurations (Job Length and their requirements), and broker scheduling policies.

The IOTSim is one of the important simulation tools that enables and supports simulation of IoT big data processing through MapReduce model in cloud environment.

G. IOTSIM-EDGE

The IOTSim-Edge [66] captures the working of edge computing infrastructure with heterogeneous IoT. It permits the users to test their frameworks in an easy, efficient and configurable way. The IOTSim-Edge consist of mainly following two components:

- Sensing nodes: Sensing nodes collect information through sensors from surroundings and send for processing and storage.
- Actuators: Actuators will be activated based on the analysis of data.

There are two main layers performing various tasks in IOTSim-Edge.

- The communication layer is liable for data transfer to/from IoT devices, cloud, and edge devices.
- The services layers consist of various services which are directly accessible to the users. The example of services are a smart city, smart home, smart transportation, and smart healthcare.

TABLE 2. Evaluation Parameters and platforms used in various resource scheduling techniques.

References	Scheduling Policy	Evaluation Parameters									Evaluation Platform
		Make-span	Deadline	Execution Cost	Budget	SLA Violation	Network Usage	Network Delay	Throughput	Energy Consumption	
[1]	DAFT	✓	✗	✗	✗	✗	✗	✗	✓	✗	CloudSim
[11]	ADAS	✓	✗	✗	✓	✗	✗	✗	✗	✗	Real world cloud environment
[12]	PCP, IC-PCP, IC-PCPD2	✗	✓	✓	✗	✗	✗	✗	✗	✗	Utility Grids
[47]	MHRA	✓	✗	✗	✗	✗	✗	✗	✗	✓	COMPSs
[48]	OWM	✓	✓	✗	✗	✗	✗	✗	✗	✗	Real world grid environment
[49]	ROSA	✓	✓	✓	✓	✗	✗	✗	✗	✗	Cloud service environment
[50]	FASTER	✓	✓	✗	✗	✗	✗	✗	✗	✗	CloudSim
[51]	BaRRS	✓	✗	✓	✗	✗	✓	✗	✗	✗	VMware-ESXi-based
[53]	EDS-DC	✓	✓	✓	✓	✓	✗	✗	✗	✗	WorkflowSim
[56]	EDF_DVFS_AC	✗	✗	✗	✗	✓	✗	✗	✗	✓	C++
[57]	MWSTR	✓	✓	✗	✗	✗	✗	✗	✗	✓	Real world cloud environment

The IoTSim-Edge extends the characteristics of CloudSim in order to integrate the various features of IoT devices and edge computing.

VIII. OVERVIEW OF EVALUATION PARAMETERS AND PLATFORMS

This section provides an overview of parameters used by the researchers in various resource scheduling techniques. Table 2 shows the list of evaluation parameters and platforms used by the researchers for evaluation of various resource scheduling techniques. This section will be helpful for the researchers in future in order to select suitable parameters and platforms for evaluation of their proposed strategies.

IX. DESIGN GOALS

Following are the major goals that may be helpful to design and implement an effective workflows scheduling and management system for scientific workflows in cloud computing:

A. CODE EXPORTABILITY

For evaluation of scientific workflows management and scheduling strategies, the simulation environment should be multiple code exportable. Most of the currently implemented simulators support C, C++, Java and Python. The research problems are sometimes language dependent or easily implemented in one language as compared with the other. The researchers also have different expertise in respect of code exportability, thus, a simulator with code exportability of C, C++, Java and Python may provide an efficient implementation.

B. GRAPHICAL USER INTERFACE

Keeping in view the vast and rapidly growing field of scientific applications, the importance of Graphical User Interface (GUI) cannot be denied at any stage. Most of the researchers relating to various scientific fields are not directly related to the field of Information Technology. Instead their problems relate to some other fields including Medical Sciences and Material Science. Thus, an efficient simulator with interactive graphical interface is now the need of concerned researchers/organizations.

C. PROTOCOLS IMPLEMENTATION

As reflected from the literature, till now, not a single and universal protocol/infrastructure has been implemented for scheduling and management of scientific applications. It is due to the varied environments supported by various scientific workflows. Thus, a good scheduling and management system has to implement all the major protocols/infrastructures supported by scientific applications in cloud computing.

D. SCALABILITY

Scientific applications are highly data-intensive applications as these processes the huge amount of compute intensive and data intensive tasks. Therefore, the requirement of the resources for scientific workflows grows exponentially due to the development in scientific fields. The scalability is one of the major features to reflect the growing size of scientific applications in cloud computing infrastructure.

X. OPEN RESEARCH CHALLENGES AND ISSUES

Beside the features and characteristics provided by different resource scheduling and management strategies, there are also some open research challenges and issues. These challenges and issues are provided below:

A. COMPUTATIONAL RISK MANAGEMENT

Computational risk management is one of the open research challenges for scientific workflows in cloud computing. Computational risk management is the process of handling the computational resources in respect of efficiency, accuracy, reliability and scalability. Since, for management and scheduling of scientific workflows in cloud computing, the computations and processing involve at a geographical based environment. Thus, for efficient, reliable and accurate computations, there should be the techniques/algorithms to handle the computational risk management.

B. INDEPENDENT RESOURCE MANAGEMENT

Independent resource management is the process in which the computational, memory and storage resources are managed autonomously. For scientific workflows management and scheduling strategies, as reflected from the literature, there is no such technique/algorithm that is used to handle the resources independently and efficiently. Since, for management and scheduling of scientific data, the resources are dispersed locally and globally, thus, there should be the techniques/algorithms used to manage resources independently.

C. SERVICE MANAGEMENT

For management and scheduling of scientific data, the services like platforms, software and infrastructures are provided from centralized places to end users. Thus, there should be the service management techniques which are applied not only at centralized level but also at the end user side. Such service management techniques should be able to implement the routing and network protocols, fault tolerance and Quality of Service (QoS) parameters. Similarly, there should be a mechanism for selection of an appropriate service provider.

D. SYSTEM MODELLING

In order to manage and schedule scientific applications, system modelling is an essential component. In System modelling, it is necessary to develop an appropriate model for implementation of research problems. Since, for execution of scientific workflows, the resources are heterogeneous and the nature of the problems also varies from each other, thus, there should be the efficient system modelling to manage resources and solve the research problems.

XI. CONCLUSION

This study presented a comprehensive review on scientific workflows management and scheduling in cloud computing. It presented: an overview of existing surveys on scientific workflows management systems; a taxonomy of scientific workflow applications and characteristics; the working of existing scientific workflows management and scheduling techniques including resource scheduling, fault-tolerant scheduling and energy efficient scheduling; and the discussion on various performance evaluation parameters along with definition and equation. It also provided discussion on various performance evaluation platforms used for evaluation of scientific workflows management and scheduling strategies. It finds evaluation platforms used for the evaluation of scientific workflows techniques based on various performance evaluation parameters. It also finds various design goals for presenting new scientific workflow management techniques. Finally, it explores the open research issues that require attention and high importance.

Since, most of the researchers relating to various scientific fields are not directly connected to the field of information technology. Instead their problems relate to some other fields including Medical Sciences and Material Science. Thus, an efficient simulator with interactive graphical interface is now the need of concerned researchers/organizations. This work concluded that the researchers have different expertise in respect of code exportability, thus, a simulator with code exportability of C, C++, Java and Python may provide an efficient implementation. This work also concluded that a good scheduling and management system has to implement all the major protocols/infrastructures supported by scientific applications in cloud computing. The study further concluded that scalability is one of the major features that reflect the growing size of scientific applications in cloud computing infrastructure.

The study will be extended for scientific workflows and big data management and scheduling in fog and edge computing as future work.

CONFLICTS OF INTEREST

The authors have no conflict of interest.

ACKNOWLEDGMENT

The authors would like to thank the financial support and facilities provided by Universiti Putra Malaysia and the Ministry of Education Malaysia for the execution, completion and publication of this paper.

REFERENCES

[1] D. Sun, G. Chang, C. Miao, and X. Wang, "Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments," *J. Supercomput.*, vol. 66, no. 1, pp. 193–228, Oct. 2013.

[2] A. U. Rehman, Z. Ahmad, A. I. Jehangiri, M. A. Ala'Anzy, M. Othman, A. I. Umar, and J. Ahmad, "Dynamic energy efficient resource allocation strategy for load balancing in fog environment," *IEEE Access*, vol. 8, pp. 199829–199839, 2020.

[3] M. Riedel, F. Wolf, D. Kranzlmüller, A. Streit, and T. Lippert, "Research advances by using interoperable e-science infrastructures The infrastructure interoperability reference model applied in e-science," *Cluster Comput.*, vol. 12, no. 4, pp. 357–372, 2009.

[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.

[5] Z. Ahmad, B. Nazir, and A. Umer, "A fault-tolerant workflow management system with quality-of-service-aware scheduling for scientific workflows in cloud computing," *Int. J. Commun. Syst.*, vol. 34, no. 1, p. e4649, 2021.

[6] W. Saeed, Z. Ahmad, A. I. Jehangiri, N. Mohamed, and A. I. Umar, "A fault tolerant data management scheme for healthcare Internet of Things in fog computing," *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 1, pp. 35–57, 2021.

[7] D. Lifka, I. Foster, S. Mehringer, M. Parashar, P. Redfern, C. Stewart, and S. Tuecke, "XSEDE cloud survey report," 2013.

[8] N. Dimitri, "Pricing cloud IaaS computing services," *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–11, 2020.

[9] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[10] A. Ullah, J. Li, Y. Shen, and A. Hussain, "A control theoretical view of cloud elasticity: Taxonomy, survey and challenges," *Cluster Comput.*, vol. 21, no. 4, pp. 1735–1764, Dec. 2018.

[11] L. Zeng, B. Veeravalli, and A. Y. Zomaya, "An integrated task computation and data management scheduling strategy for workflow applications in cloud environments," *J. Netw. Comput. Appl.*, vol. 50, pp. 39–48, Apr. 2015.

[12] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 158–169, Jan. 2013.

[13] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything*. Singapore: Springer, 2018, pp. 103–130.

[14] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, "Pegasus, a workflow management system for science automation," *Future Gener. Comput. Syst.*, vol. 46, pp. 17–35, May 2015.

[15] J. Liu, L. Pineda, E. Pacitti, A. Costan, P. Valduriez, G. Antoniu, and M. Mattoso, "Efficient scheduling of scientific workflows using hot metadata in a multisite cloud," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1940–1953, 2018.

[16] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Algorithms for cost- and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds," *Future Gener. Comput. Syst.*, vol. 48, pp. 1–18, Jul. 2015.

[17] R. F. da Silva, H. Casanova, A. C. Orgerie, R. Tanaka, E. Deelman, and F. Suter, "Characterizing, modeling, and accurately simulating power and energy consumption of I/O-intensive scientific workflows," *J. Comput. Sci.*, vol. 44, 2020, Art. no. 101157.

[18] M. A. Rodriguez and R. Buyya, "A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 8, 2017, Art. no. e4041.

[19] S. Callaghan, P. Maechling, P. Small, K. Milner, G. Juve, T. H. Jordan, E. Deelman, G. Mehta, K. Vahi, D. Gunter, and K. Beattie, "Metrics for heterogeneous scientific workflows?: A case study of an earthquake science application," *Int. J. High Perform. Comput. Appl.*, vol. 25, no. 3, pp. 274–285, 2011.

[20] S. Callaghan, P. Maechling, E. Deelman, K. Vahi, G. Mehta, G. Juve, K. Milner, R. Graves, E. Field, D. Okaya, D. Gunter, K. Beattie, and T. Jordan, "Reducing time-to-solution using distributed high-throughput mega-workflows–experiences from SCEC CyberShake," in *Proc. IEEE 4th Int. Conf. eScience*, Dec. 2008, pp. 151–158.

[21] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: The montage example," in *Proc. SC Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2008, pp. 1–12.

[22] D. de Oliveira, F. Porto, C. Boeres, and D. de Oliveira, "Towards optimizing the execution of spark scientific workflows using machine learning-based parameter tuning," *Concurrency Comput., Pract. Exper.*, p. e5972, 2020.

[23] T. Selker, "Touching the future," *Commun. ACM*, vol. 51, no. 12, pp. 14–16, 2008.

[24] Z. Lv, X. Li, H. Lv, and W. Xiu, "BIM big data storage in WebVRGIS," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2566–2573, Apr. 2020.

[25] V. Mauch, M. Kunze, and M. Hillenbrand, "High performance cloud computing," *Future Gener. Comput. Syst.*, vol. 29, no. 6, pp. 1408–1416, 2013.

[26] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on SDN and NFV for next generation IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5706–5712, Jul. 2020.

[27] W. D. Santos, L. F. M. Carvalho, G. D. P. Avelar, A. Silva, L. M. Ponce, D. Guedes, and W. Meira, "Lemonade: A scalable and efficient spark-based platform for data analytics," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2017, pp. 745–748.

[28] V. M. Gottin, E. Pacheco, J. Dias, A. E. Ciarlini, B. Costa, W. Vieira, Y. M. Souto, P. Pires, F. Porto, and J. G. Rittmeyer, "Automatic caching decision for scientific dataflow execution in apache spark automatic caching decision for scientific dataflow execution in apache spark," in *Proc. 5th ACM SIGMOD Workshop Algorithms Syst. MapReduce Beyon*, Jun. 2018, pp. 1–10.

[29] S. Tansley and K. M. Tolle, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, vol. 1, A. J. Hey, Ed. Redmond, WA, USA: Microsoft Research, 2009.

[30] W. Oliveira, D. D. E. Oliveira, and V. Braganholo, "Provenance analytics for workflow-based computational experiments?: A survey," *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–25, 2018.

[31] V. Silva, D. de Oliveira, P. Valduriez, and M. Mattoso, "DfAnalyzer: Runtime dataflow analysis of scientific applications using provenance," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 2082–2085, 2018.

[32] K. A. Ocaña, D. de Oliveira, E. Ogasawara, A. M. Dávila, A. A. Lima, and M. Mattoso, "SciPhy: A cloud-based workflow for phylogenetic analysis of drug targets in protozoan genomes," in *Proc. Brazilian Symp. Bioinf.* Berlin, Germany: Springer, Aug. 2011, pp. 66–70.

[33] Z. Lv and L. Qiao, "Analysis of healthcare big data," *Future Gener. Comput. Syst.*, vol. 109, pp. 103–110, Aug. 2020.

[34] C. Qi, "Big data management in the mining industry," *Int. J. Minerals, Metall. Mater.*, vol. 27, no. 2, pp. 131–139, 2020.

[35] X. Yao, G. Li, J. Xia, J. Ben, Q. Cao, L. Zhao, Y. Ma, L. Zhang, and D. Zhu, "Enabling the big Earth observation data via cloud computing and DGGS?: Opportunities and challenges," *Remote Sens.*, vol. 12, no. 1, p. 62, 2020.

[36] C. Lin, P. Chen, C. Lin, and W. Wu, "Big data management in healthcare?: Adoption challenges and implications international journal of information management big data management in healthcare?: Adoption challenges and implications," *Int. J. Inf. Manage.*, vol. 53, Aug. 2020, Art. no. 102078.

[37] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, and K. Vahi, "Characterization of scientific workflows," in *Proc. 3rd Workshop Workflows Support Large-Scale Sci.*, Nov. 2008, pp. 1–10.

[38] S. Mustafa, B. Nazir, A. Hayat, A. U. R. Khan, and S. A. Madani, "Resource management in cloud computing: Taxonomy, prospects, and challenges," *Comput. Electr. Eng.*, vol. 47, pp. 186–203, Oct. 2015.

[39] M. Ala'Anzy and M. Othman, "Load balancing and server consolidation in cloud computing environments: A meta-study," *IEEE Access*, vol. 7, pp. 141868–141887, 2019.

[40] S. S. Manvi and G. Krishna, "Journal of network and computer applications resource management for infrastructure as a service (IaaS) in cloud computing?: A survey," *J. Netw. Comput. Appl.*, vol. 41, pp. 424–440, May 2014.

[41] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, "A survey of data-intensive scientific workflow management," *J. Grid Comput.*, vol. 13, no. 4, pp. 457–493, Dec. 2015.

[42] R. Mork, P. Martin, and Z. Zhao, "Contemporary challenges for data-intensive scientific workflow management systems," in *Proc. 10th Workshop Workflows Support Large-Scale Sci.*, Nov. 2015, pp. 1–11.

[43] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *J. Netw. Comput. Appl.*, vol. 66, pp. 64–82, 2016.

[44] X. Ye, J. Liang, S. Liu, and J. Li, "A survey on scheduling workflows in cloud environment," in *Proc. Int. Conf. Netw. Inf. Syst. Comput.*, Jan. 2015, pp. 344–348.

[45] S. Smanchat and K. Viriyapant, "Taxonomies of workflow scheduling problem and techniques in the cloud," *Future Gener. Comput. Syst.*, vol. 52, pp. 1–12, Nov. 2015.

[46] E. Nabiel, S. Peck, R. Rezaei, and R. Meimandi, "Cost optimization approaches for scientific workflow scheduling in cloud and grid computing?: A review, classifications, and open issues," *J. Syst. Softw.*, vol. 113, pp. 1–26, Mar. 2016.

[47] F. Juarez, J. Ejarque, and R. M. Badia, "Dynamic energy-aware scheduling for parallel task-based application in cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 257–271, Jan. 2018.

[48] C.-C. Hsu, K.-C. Huang, and F.-J. Wang, "Online scheduling of workflow applications in grid environments," *Future Gener. Comput. Syst.*, vol. 27, no. 6, pp. 860–870, Jun. 2011.

[49] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Trans. Services Comput.*, early access, Aug. 21, 2019, doi: 10.1109/TSC.2018.2866421.

[50] X. Zhu, J. Wang, H. Guo, D. Zhu, L. T. Yang, and L. Liu, "Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 12, pp. 3501–3517, Dec. 2016.

[51] I. Casas, J. Taheri, R. Ranjan, L. Wang, and A. Y. Zomaya, "A balanced scheduler with data reuse and replication for scientific workflows in cloud computing systems," *Future Gener. Comput. Syst.*, vol. 74, pp. 168–178, Sep. 2017.

[52] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *Proc. Int. Conf. High Perform. Comput. Simul.*, Jun. 2009, pp. 1–11.

[53] Z. Ahmad, A. I. Jehangiri, M. Iftikhar, A. I. Umer, and I. Afzal, "Data-oriented scheduling with dynamic-clustering fault-tolerant technique for scientific workflows in clouds," *Program. Comput. Softw.*, vol. 45, no. 8, pp. 506–516, Dec. 2019.

[54] W. Chen and E. Deelman, "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments," in *Proc. IEEE 8th Int. Conf. E-Sci.*, Oct. 2012, pp. 1–8.

[55] R. F. da Silva, A. C. Orgerie, H. Casanova, R. Tanaka, E. Deelman, and F. Suter, "Accurately simulating energy consumption of I/O-intensive scientific workflows to cite this version?: HAL Id?: Hal-02112893 accurately simulating energy consumption of I/O-intensive scientific workflows," in *Proc. Int. Conf. Comput. Sci.*, 2019, pp. 138–152.

[56] G. L. Stavrinides and H. D. Karatza, "An energy-efficient, QoS-aware and cost-effective scheduling approach for real-time workflow applications in cloud computing systems utilizing DVFS and approximate computations," *Future Gener. Comput. Syst.*, vol. 96, pp. 216–226, Jul. 2019.

[57] J. Jiang, Y. Lin, G. Xie, and L. Fu, "Time and energy optimization algorithms for the static scheduling of multiple workflows in heterogeneous computing system," *J. Grid Comput.*, vol. 15, no. 4, pp. 435–456, 2017.

[58] D. Poola, K. Ramamohanarao, and R. Buyya, "Fault-tolerant workflow scheduling using spot instances on clouds," *Procedia Comput. Sci.*, vol. 29, pp. 523–533, Jan. 2014.

[59] T. Mathew, K. C. Sekaran, and J. Jose, "Study and analysis of various task scheduling algorithms in the cloud computing environment," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2014, pp. 658–664.

[60] N. Verba, K.-M. Chao, J. Lewandowski, N. Shah, A. James, and F. Tian, "Modeling industry 4.0 based fog computing environments for application analysis and deployment," *Future Gener. Comput. Syst.*, vol. 91, pp. 48–60, Feb. 2019.

[61] R. Kumar, K. K. Pattanaik, S. Bharti, and D. Saxena, "In-network context inference in IoT sensory environment for efficient network resource utilization," *J. Netw. Comput. Appl.*, vol. 130, no. Jun. 2018, pp. 89–103, 2019.

[62] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A CloudSim-based visual modeller for analysing cloud computing environments and applications," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 446–452.

[63] K. Wölfel, T. Werner, and D. Henrich, "GroundSim: Animating human agents for validated workspace monitoring," in *Tagungsband des 3. Kongresses Montage Handhabung Industrieroboter*. Berlin, Germany: Springer, 2018, pp. 205–213.

[64] H. Gupta and R. Buyya, "iFogSim?: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge," *Softw., Pract. Exper.*, vol. 47, no. 9, pp. 1275–1296, May 2017.

[65] X. Zeng, S. K. Garg, P. Strazdins, P. P. Jayaraman, D. Georgakopoulos, and R. Ranjan, "IOTSim: A simulator for analysing IoT applications," *J. Syst. Archit.*, vol. 72, pp. 93–107, Jan. 2017.

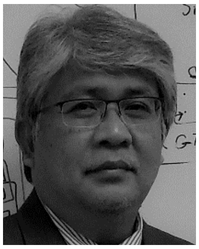
[66] D. N. Jha, S. Dustdar, and R. Ranjan, "IoTSim-Edge?: A simulation framework for modeling the behavior of Internet of Things and edge computing environments," *Softw., Pract. Exper.*, vol. 50, no. 6, pp. 844–867, May 2019.

ZULFIQAR AHMAD received the M.Sc. degree (Hons.) in computer science from Hazara University, Mansehra, Pakistan, in 2012, and the M.S. degree in computer science from COMSATS University, Islamabad, Pakistan, in 2016. He is currently pursuing the Ph.D. degree in computer science with the Department of Information Technology, Hazara University. He is currently a Lecturer with the Department of Information Technology, Hazara University. His research interests include fog computing, cloud computing, high-performance computing, and scientific workflows scheduling and management.

ALI IMRAN JEHANGIRI received the degree from Bergische University, Wuppertal, Germany, in 2010, and the Ph.D. degree in computer science from Georg-August-University, Goettingen, Germany, in 2015. He gained industrial experience in service computing by working as a Research Assistant with GWDG. He is currently a Lecturer with the Department of Information Technology, Hazara University, Mansehra, Pakistan. He is involved in research activities dealing with parallel, grid computing, cloud computing, and big data. He is the author of several publications in international journals and conferences.

MOHAMMED ALAA ALA'ANZY received the master's degree in computer science from University Putra Malaysia, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Information Technology, University Putra Malaysia.

His current research interests include cloud computing, green computing, load balancing, task scheduling, and fog computing. He has authored several high-reputed journal/conference papers and a Reviewer of Scientific.Net journal.



MOHAMED OTHMAN (Senior Member, IEEE) received the Ph.D. degree (Hons.) from the National University of Malaysia. He was a Visiting Professor with South Kazakhstan State University, Shymkent, Kazakhstan, and the L. N. Gumilyov Eurasian National University, Astana, Kazakhstan. He is currently a Professor of computer science with the Department of Communication Technology and Networks, Universiti Putra Malaysia (UPM). Prior to that, he was the Deputy Director of the Information Development and Communication Centre, where he was in charge of the UMPNet Network Campus, the uSport Wireless Communication Project, and the UPM Data Center. He is also an Associate Researcher and a Coordinator of high-speed machines with the Laboratory of Computational Science and Informatics, Institute of Mathematical Science, UPM. He has published more than 300 International journals and 330 proceeding articles. He has also filed six Malaysian, one Japanese, one South Korean, and three U.S. patents. His main research interests include computer networks, parallel and distributed computing, high-speed interconnection networks, network design and management (network security, wireless, and traffic monitoring), consensus in the IoT, and mathematical models in scientific computing. He is a Life Member of the Malaysian National Computer Confederation and the Malaysian Mathematical Society. He was awarded the Best Ph.D. Thesis in 2000 by Sime Darby Malaysia and the Malaysian Mathematical Science Society. In 2017, he received the Honorary Professorship from SILKWAY International University (formerly known as South Kazakhstan Pedagogical University), Shymkent.

ROHAYA LATIP received the bachelor's degree in computer science from University Technology Malaysia, Malaysia, in 1999, and the M.Sc. degree in distributed systems and the Ph.D. degree in distributed database from University Putra Malaysia. She was the Head of the HPC Section, University Putra Malaysia, from 2011 to 2012, where she consulted the Campus Grid Project and the Wireless for hostel in Campus UPM Project. She is currently the Head of the Department of Communication Technology and Network, where she is also an Associate Professor with the Faculty of Computer Science and Information Technology. She is also a Co-Researcher with the Institute for Mathematic Research (INSPEM). Her research interests include big data, cloud and grid computing, network management, and distributed database.

SARDAR KHALIQ UZ ZAMAN received the M.S. degree in computer science from the Department of Computer Science, COMSATS University Islamabad (CUI), Abbottabad Campus, Pakistan, in 2014. He is currently pursuing the Ph.D. degree with the Department of Information Technology, Hazara University, Mansehra, Pakistan. He joined CUI, Abbottabad Campus, as a Lecturer, in 2015, where he is currently an active member of the Scalable Processing and Analytics Research in Communications (SPARC) Laboratory. His research interests/publications include mobile edge networks, large-scale computing systems, social informatics, cloud computing, and data centers systems. With six years of research and teaching experience, he has published different articles in international journals and conferences.

ARIF IQBAL UMAR received the M.Sc. degree in computer science from the University of Peshawar, Pakistan, and the Ph.D. degree in computer science from Beihang University (BUAA), Beijing, China. He has been working as an Associate Professor of computer science with the Department of Information Technology, Hazara University, Mansehra. He has been leading the Department as the Chairman. He has supervised seven Ph.D. candidates and 34 M.S. candidates. He is the author of more than 70 research publications in the leading research journals and conferences. He has at his credit 27 years' experience of teaching, research, planning, and academic management. His research interests include data mining, machine learning, information retrieval, digital image processing, computer networks security, and sensor networks.