International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. , No. (Mon-20..)

http://dx.doi.org/10.12785/ijcds/XXXXXX

Security Time Prediction of Big Data Workflows with AES Algorithm-aware Simulation

Faris Llwaah[1]

[1]*Department of Cyber Security,College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq*

Received Mon. 20, Revised Mon. 20, Accepted Mon. 20, Published Mon. 20

Abstract: Increasing the number of applications with data-intensive workflows, like healthcare workflow, has led to a tendency to embed highly popular cloud computing in the matter of delivering substantial computing resources, and ensuring the security and performance of these complex applications is of utmost importance while estimating sufficient execution time and the amount of resources required for deployment. While that cloud-based execution time includes CPU time, I/O operations, and security time, particularly for workflows involving big data. However, in our previous work, we introduced a methodology to model, simulate, and predict the runtime of big data workflow, including intricate Next Generation Sequencing (NGS) pipelines. This simulation approach provides a realistic estimate of the runtime for test data that is much larger than the training data used. In this paper, we tackle the problem of predicting big data workflow security time performance using a simulation model that takes the (Advanced Encryption Standard) AES algorithm into account. To precisely assess the runtime impact, our methodology entails modeling and simulating the encryption and decryption procedures within big data workflows. We demonstrate our method's effectiveness in generating precise runtime predictions and validate it using an NGS pipeline implemented in e-Science Central. For ensuring optimization performance without compromising data security, the results show the importance of considering security overhead in the NGS pipeline. However, this work makes contributions to the field by applying a practical simulation framework based on WorkflowSim to predict security-related performance impacts. The results confirm an exponential relationship between the stable execution time of implementing security algorithms and the volume of processed big data, indicating that the time doubles as the data volume doubles.

Keywords: Big data workflow, Data-Intensive simulation, cloud computing, WorkflowSim, Next Generation Sequencing NGS, Advanced Encryption Standard AES, security time performance, I/O operations

1. INTRODUCTION

In the domain of big data workflows, to keep sensitive information protected and confidential, it is crucial to ensure that data workflow becomes efficient and secure. As large organizations increasingly depend on cloud-based application and distributed systems to manage and process the huge quantities of data [1], predicting potential security threats and mitigating their impacts has become a significant challenge. This paper tackles this issue by introducing a novel approach for predicting time-related security metrics within the context of big data workflows using a simulation-based platform.

The rapid expansion of data has led to the widespread use of big data technologies and cloud computing services, which offer scalable and flexible ways for managing sensitive, huge data. Where, this development and flexibility come with complex security issues. As data is processed across diverse nodes and platforms, including cloud environments, the implementation of robust security measures becomes more challenging. Because attackers are always changing how they use weaknesses in these spread systems [2], proactive security management is very.

Despite the effectiveness of big data workflows in cloud computing, many previous studies have focused on predicting performance to reduce execution time, resource usage, and operational costs [3], [4], [5], [6]. However, the lack of data protection in this work makes it easy to compromise user data in the cloud, thereby increasing costs for cloud service providers. All of this poses a significant challenge to the field of cloud computing. Therefore, security issues must be considered in our agenda to enhance workflows performance and prevent potential economic loss risks for users.

Many big data applications not only require substantial computing resources and storage from cloud computing, but they also necessitate stringent security measures when processing their critical big data. Security considerations in big data environments significantly influence processing performance. Therefore, to reconcile the implementation of

E-mail address: f.llwaah@uomosul.edu.iq

security solutions for large-scale data processing, security analysis becomes more essential than ever. When a big data is considered as a collection of diverse and sensitive information, its components vary in terms of security categories, necessitating new analytical solutions for data security.

Time prediction in the security operations of large-scale data applications plays an essential role for the design and dissemination of data analysis systems. Cloud access facilitates the provision of almost unlimited resources but does not provide significant support in determining the optimal composition of the user workload. In the area of large-scale data analyses, this point is particularly important given the fact that these analyses consume a significant amount of determining time, which becomes a direct financial cost. Any inconsistency between available resources and the actual workload could therefore lead to an increase in cost. The execution of cloud-based applications requires time, much like the operations of the central processing unit and input/output devices. In addition to that, the new research that we conducted covers the security of big data and predicts the time required to perform security operations.

To perform this work, we used a recently proposed platform to predict the big data workflow execution time. We can model the behaviour of complicating workflow-based assignments with this platform. Using this method, we were able to predict the runtime performance of the NGS pipeline that was configured on the public cloud. The pipeline is a program used to find and annotate variations in the human exome, which is a small portion of the human genome. The pipeline requires thousands of CPU hours to evaluate four hundred GB of compacted data in order to identify variations in the group of 24 patients. Working with such applications is challenging due to the complicated dependencies between tasks, data, and the cloud, which can lead to failures, lower performance of the system, and increased time and cost. These factors are in addition to the volume of input data and the length of time needed to deal with it.

Users require security because cloud computing is a platform that shares resources for efficient big data analysis. Since security is a crucial component of cloud computing, cloud service providers bear the responsibility of ensuring security across all its features, including reducing implementation time, minimizing data transfer delays between assignments, and providing data security services. Traditional methods, which are already available, cannot effectively measure the security of cloud services [7]. Cloud computing's security framework streamlines management and access to computer resources, and after numerous attempts to predict the time required to calculate encryption and decryption times, we opted to use a simulation platform instead of a real cloud to save costs during experiment repetition. We should utilize this framework to reduce the time and delay involved in encryption and decryption, thereby improving the accuracy of predictions and enhancing data security in cloud computing through the implementation of the suggested algorithm.

The objective of this study is to develop a method for big data applications that are crucial for ensuring system security, dynamically controlling the implementation time in distributed systems, and predicting the security implementation time for large data flows involving complex calculations in distributed computer systems. Therefore, we introduced the AES algorithm, a widely used encryption method in software and hardware worldwide.

This paper describes our expansion of the WorkflowSim environment to model related-security time when the AES algorithm is applied for encoding and decoding of transmission data. We also demonstrate how the security performance estimation of our big data workflows is enhanced by the modification.

This work's primary contributions are:

- The primary contributions of this paper are the creation of a simulation-based model that accurately reflects the related-time security of big data workflows and their associated security challenges. The simulator offers the user a prediction time by incorporating a variety of security parameters, including the data volume of the task.
- The paper presents the potential of simulation-based approaches to advance the field of big data security, which is underscored by these findings. For the first time, the AES algorithm was applied to the NGS pipeline as a real-world illustration of our experiment.
- We will illustrate the improved precision of our runtime predictions for the total execution time and reflect on the benefits and constraints of the proposed component.

This paper's remaining sections follow this structure: Section 2 provides a general view of the background and Section 3, shows the related work. In Section 4, we demonstrate the major problem of NGS pipeline simulation. Section 5 describes the methodology for predicting the security time for big data when the AES algorithm is applied, which is followed by Section 6. This section will describe time prediction in big data workflows. In Section 7, we design and implement a prediction model for the security time component. We will evaluate our suggestions in Section 8. Section 9 includes the final discussion and conclusions.

2. Background

One of the most popular simulation platforms is the WorkflowSim [8] to represents a workflow environment for simulation. The tool allows for the modeling and simulating of cloud-based scientific processes (data flows) and is a modification to the CloudSim simulator[9].

The simulator comprises components such as the clustering engine, a workflow mapper, scheduler, and engine, all in accordance with the methodology suggested by the Pegasus WMS. [10]. These components make it possible for users of WorkflowSim to assess and optimize a range of algorithms and methods linked to resource allocation and workflow execution, which would be time-consuming and expensive to execute in a real cloud. This paper utilizes the same platform as our previous research [11], which presented an approach for predicting the execution time performance for intricate large scale workflows, including pipelines for Next-Generation Sequencing (NGS). In this paper, we relied on the advanced platform that was obtained from previous work and thus adopted WorkflowSim [3], which can predict the duration of execution and the volume of output data for dependent tasks. This makes it possible to predict increasingly complex scientific workflows, such as the NGS pipeline. Big data analysis is now an essential technique used in many fields of study. Big data analysis is commonly carried out in cloud computing settings due to its intensive nature.

A. Big Data Concept

The term Big Data can be defined formally as an information asset with three Vs characteristics high volume, velocity, and variety that may be used to acquire particular technologies and analytical techniques to turn them into valuable information [12]. (1) Volume is the term used to describe the enormous amount of data that has been collected from different sources; this amount of data can vary from terabytes to zettabytes [13]. (2) Variety pertains to the diversity and heterogeneity of material that has been taken from multiple sources, including journals, social media networks, websites, electronic medical records, and video. Data can really be presented in a variety of ways, such as structured, semi-structured, and unstructured information in a variety of media formats, such as text, image, and video. As such, different meanings and interpretations might be drawn from the same dataset [14]. (3) Velocity is the term used to describe the rate at which data is generated, which is now frequently and data-sensitive requires handling and processing in real time. Certainly, a multitude of data sources, including sensors, produce continuously updated data that must be monitored in real time [15].

B. Big Data Workflows

A big data workflow is a computational model to process and analyze data that is constantly growing in volume, complexity, and acquisition rate. It consists of a series of computing operations and their data dependencies [16], [17]. A big data workflow management system (BDWFMS) is a cloud-based platform that fully develops, adjusts, oversees, tracks, and performs scientific workflows in the sequence determined by the workflow logic [16], [17].

C. Sequencing of Next Generation (NGS)

A Workflow-controlled pipeline is our big data workflow case study. It can be defined as a sophisticated pipeline NGS for data processing in genomics that is hosted on the Microsoft Azure public cloud [18], [19]. To find variations in a patient's exome, the NGS pipeline is employed. It usually takes several days to process a cohort of 24 patient samples through the local deployment of this pipeline. Although there is potential for a significantly faster Azure deployment, due to financial constraints, an ideal or nearly optimal deployment that minimizes execution time and cost must be estimated.

D. Simulation Platform

Specifically, WorkflowSim has been extensively employed in the scientific field for a range of applications and issues, such as energy-conscious resource allocation, scheduling, and provisioning algorithms [20], [21], [22]. We have conducted far less research on runtime prediction for big data applications, where storage performance modeling is crucial.

E. Advanced Encryption Standard (AES)

It is the best global and widely used proportion block cipher algorithms. Hardware and software worldwide use this method, which has a structure that is unique for encrypting and decrypting sensitive data. When using the AES method for encryption, hackers have a a very difficult time deciphering the physical data. there is no clear evidence of a break to this algorithm. AES can handle three distinct key sizes, namely AES 128, 192, and 256 bits, with each cipher having a block size of 128 bits [23].

3. Related Work

Cloud computing is highly advantageous in contexts that frequently process massive datasets. It offers a way to ensure data security without sacrificing implementation speed, and that eventually impacts the overall cost. In order to better understand and improve the use of AES algorithm in big-data workflows, the research studies that follow focus on improving execution time and security through a variety of simulation and implementation strategies.

The paper [24] demonstrates how parallel processing can effectively improve the performance of the AES encryption and decryption algorithm, addressing security and efficiency concerns in contemporary data-intensive applications. This makes a significant contribution to the fields of big data and cryptography. In the paper [25], the authors addressed computing security concerns for some time and proposed a reliable storage solution for managing big data workflows

in multiple cloud setups. The paper's primary focus is on creating encryption methods that secure private information, such as those from healthcare providers, banks, and the military.

In their paper [26], the authors used deep learning to predict time-to-event in security logs using a joint predictor with a three-layered RNN structure, LSTM, and attention mechanism, enhancing prediction capability.

The paper [27], the authors presented a thorough strategy to improve cloud security and resource management. The research centers on dynamically evaluating workloads and predicting virtual machine (VM) hazards. The proposed solution integrates threat prediction methods with workload estimation to maximize resource utilization and proactively handle potential security vulnerabilities in industrial cloud settings. This article has also proposed a new paradigm to improve the reliability and performance of cloud-based industrial operations through the integration of predictive analytics and dynamic resource management.

In the paper [28], the authors designed a model for resource provisioning to enhance cloud service sustainability to prioritize high uptime and strong security measures. The article proposed a framework to optimize resource allocation to ensuring reliability of the service while implementing stringent security measures to protect against potential threats. The presented model addresses a critical need of ensuring the security and dependability of cloud services while both uptime and security have been focused. As outlined in the aforementioned related work, previous studies have explored various aspects of security and cloud computing, but none have considered the security implications of big data workflows. Therefore, by addressing these issues, we propose an expansion to create a simulated model framework with more precise and realistic scenarios. Furthermore, considering predictions on security time at the general implementation level of the big data workflow is beneficial for enhancing performance.

4. NGS PIPELINE SIMULATION

The standard NGS pipeline is separated into three phases: the first and last phase operate in *split of sample* mode, while the intermediate phase operates in a manner known as *split of chromosome* mode. The *split of sample* phases follow a primarily sequential process and consist of eight and two tasks, respectively. We reproduce the data samples based on their quantity, ensuring that every sample performs an individual sequence of tasks. The split of chromosome phase consists of an initial join task, followed by a specific number of tasks that run simultaneously on separate chromosomal regions, and concludes with two final tasks. In all, the pipeline comprises $9 \times N + 53$ tasks, where N represents how many input samples there are (see Fig. 4). As an example, our WorkflowSim simulation includes 269 tasks in the largest configuration. Each simulated task in the actual pipeline consists of multiple workflow blocks modeled in e-SC, as previously stated. As a result, the 24 sample run requires thousands of CPU hours for completing numerous tasks.

The primary purpose of WorkflowSim was to replicate Pegasus workflows [29]. Therefore, in order for modelling the pipeline created in eSC, some level of customization was necessary. Both systems share the common feature of supporting the scientific workflows execution, often known as data streams. However, there are notable distinctions between the two systems. More precisely, eSC workflows are defined by their granularity and can function with two distinct levels: *fundamental* and *mixed*. A fundamental workflow may consist of numerous tasks, but all of them are executed on an identical virtual machine, known as the engine. This facilitates the optimization and execution of small, or short term tasks by ensuring that data transport between them occurs within a localized area. A mixed workflow includes tasks that may launch basic and/or mixed sub-workflows. Each of these sub- workflows can be executed on a different engine, facilitating the management of task and data parallelism commonly encountered in scientific analysis.

The differentiation between *fundamental* and *mixed* workflows is particularly crucial when designing workflows for Big Data since it significantly impacts the manner in which data is transmitted between tasks. In a simple workflow, data is transferred through a local file system. However, in *mixed* workflows, data must be shared between virtual machines VMs, necessitates the building of a more sophisticated security platform.

One other distinction between the systems is that eSC workflows provide a significant amount of flexibility in terms of how and at what point workflows share their data. Like Pegasus [30], [31], [32], they may adhere to the read, process, and write pattern, in which all the data from the input is stored before the core tasks are executed and after then retrieved later. However, it is also possible for them to exchange data and trigger sub-workflows during the execution process. Currently, the use of AES encryption methods is necessary for securely transferring large volumes of sensitive data between several virtual machines. Consequently, it is essential to allocate sufficient time for ensuring the security of the data.

5. METHODOLOGY

The prediction of security time of big data workflows on the cloud includes a time required to implement security measures for data processing operations. This can be difficult because of the volume and complexity of big data, as well as the dynamic nature of cloud systems. The methodical strategy for making this prediction as follows:

- **Define Security Requirements:** Specifically, we can take into account security measures to ensure the confidentiality, integrity, and accessibility of our sensitive data during their processing and progression through various stages within the workflows execution on the

cloud.

- **Characterize Big Data Workflows:** We should break down the big data workflow into its components, which include data input, processing, storage, and analysis. Next, we classify the data based on its size to determine the time needed for security at each stage..
- **Analyze Security Mechanisms** When we implement an AES algorithm that uses the same key for encryption and decryption, the encryption transforms data into a secure format. Therefore, we employ a prediction security time model to gauge the effectiveness of runtime overheads.
- **Estimate Baseline Performance** Furthermore, the total time is increased by incorporating the time needed to secure the used data, along with estimating the CPU processing time and input/output operation time.
- **Simulation and Testing** We use the workflowsim platform to predict performance at scale by employing diverse scenarios to test the timing of security operations. Therefore, we measure time under various patient samples, such as 6, 12, and 24. We use real-time data to apply the smallest 6 sample as a set of training for prediction models, as this was the smallest the size of input data that the pipeline could successfully handle.

6. Time Prediction in Big Data workflows

As previously indicated, we presented an approach for estimating the execution efficiency of complex Big Data applications in our previous work. The approach can consider three primary elements that could impact the applications' run time: the computational workloads' size in relation to the simulated environment's CPU speed and the input and output data's size in relation to network bandwidth, in addition to predicting the time of large- scale data input and output. Even though the constraints of the environment of simulation, the nature of the *dataintensive* challenges, and the scarcity of the data sets of training. The designed simulator platform has managed to provide predictions with a respectable degree of accuracy, it can reach to a relative error of around 2- 10%. Yet, the suggested technique was able to estimate the duration of 10-, 12-, and 24-sample workloads using a tiny 6-sample input dataset spread across 12,24, and 48 VMs. The results are displayed in Fig. 1. for a training set of 6-samples with 12 VMs.

According to the situation, our predictor consistently approximated the larger difference between the training and testing sets when estimating in real time, while a public cloud platform like Microsoft Azure is being considered. The fact that each sample required processing and transmission several hundred gigabytes of data indicates that *I*nput/*O*utput operations were a significant factor in the expenditure of time for data security. Through the use of WorkflowSim and the source code analysis, we were able to verify that it does not take into account the usage of encryption algorithms for data protection, and that the calculation of data transfer time between virtual machines (VMs) is not taken into consideration either.

7. Prediction Models

As illustrated above, one source of incomplete prediction in time comes from a lack of the simulation environment we used. To point out this problem, we made the decision of designing and implementing the component of a prediction model of the security time to simulate encryption the data is being transferred between VMs in the WorkflowSim. Additionally, the use of this model improves a precision and accuracy of predictions about security time and its impact on the efficiency of the NGS pipeline workflow. With this model included, WorkflowSim can offer a thorough analysis of potential security-related delays and how they affect processing times overall.

Our work primarily concentrates on finding the necessary security time for data transmission to the task, which is currently in the waiting queue for execution. This task assumes the staging of all its input data into the machine. However, the current I/O model of WorkflowSim for scientific workflows closely matches Pegasus's data transmission mechanisms [32]. In summary, the workflow management system ensures the proper staging of all input data prior to the execution of a workflow task. Following the completion of processing, the system shares the output data with those tasks that follow. Therefore, in order to perform an evaluation of the time necessary for security, the duty of suggested prediction model starts at the stage of preparing all input data prior to the execution of the next task. Fig. 2, represents this simple model which has two distinct stages in the task execution to predict overall time.

A. *Improving Security Time Model*

To simulate the security time model to estimate the required delay time for performing security tasks from the NGS pipeline based on the AES (Advanced Encryption Standard) algorithm for enhancing WorkflowSim, we propose a predictive model that estimates the necessary delay for security operations. However, this model considers factors such as data sizes and simulator processing capabilities. According to the existing task model, we make the assumption of staging all of the input files required from the shared storage before task execution. At this step, the security time model is tasked with calculating the time needed for the security process (the encryption process) before the task moves on to the processing stage. Similarly, upon task completion, all output files are staged out to shared storage. Afterwards, the purpose of this security time model is to decrypt the data before sending the output files to the next task.

B. *Prediction*

When the pipeline works successfully, all tasks inside the pipeline have the same function of predicting delay

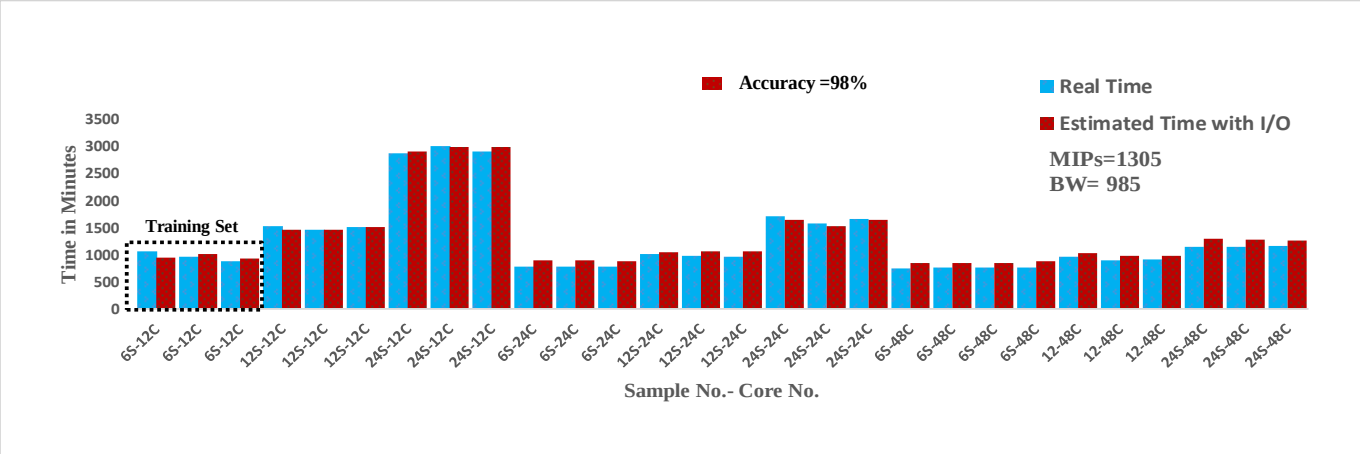

Figure 1. The Runtime estimation (CPU Time & I/O Time)

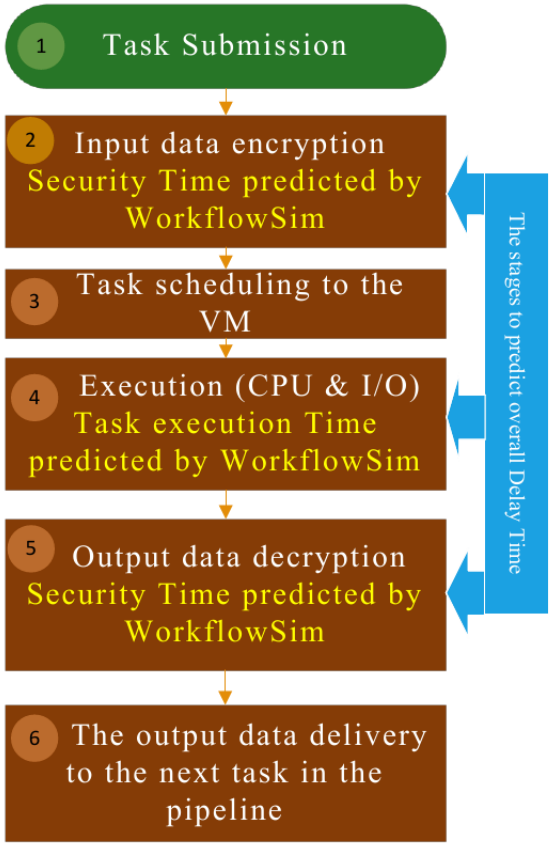

Figure 2. Distinction stages in the task execution to predict time

times. There are two phases: i) The data encryption process is depicted in step (2) in Fig. 2. During this phase, the data is encrypted using the AES encryption algorithm before being sent to the cloud VM. The delay time is determined by invoking Alg. 1 (*CalculateEncryptionRunTime*), which calculates the runtime for encryption security. ii) The data decryption stage is illustrated in step (5) in Fig. 2. At this point, the data may be decrypted using the AES decryption algorithm after the task is finished. However, all output files are stored suspended to be shared with the next task, the delay time of decryption is determined by invoking the Alg. 2 (*CalculateDecryptionrunTime*), which calculates the runtime for decryption security.

Algorithm 1: Calculate Encryption Run Time

Require: Block size in kilobytes ($blocksize_{KB}$)
Ensure: Total time for encryption in milliseconds (totalCipherTime)

```
Initialize totalCipherTime ← 0
Initialize output ← 0
Initialize blocksize ← 0
keyGenerator ← KeyGenerator.getInstance("AES")
secretKey ← keyGenerator.generateKey()
Function encryptData(blocksize_KB, secretKey)
    blocksize_Bytes ← blocksize_KB × 1024
    cipher ← Cipher.getInstance("AES")
    cipher.init ← (Encrypt_Mode, SecretKey)
Return cipher.doFinal(EncryptedData)
    startTime ← System.currentTimeMillis()
EncryptData = encryptData( Data, secretKey)
    endTime ← System.currentTimeMillis()
    totalCipherTime ← endTime – startTime
    return totalCipherTime
```

We additionally utilize two parameters to calculate the security response time in order to preserve the simplicity of the improved security model. On the other hand, this time, the data from the encryption and decryption processes shows the total latency of the security algorithms used in the data transfer. The view of cloud user, who finds it difficult to distinguish between these two forms of delays in the actual cloud environment, is reflected in this simplification. We apply the following formula to calculate

Algorithm 2: Calculate Decryption Run Time

Require: Block size in kilobytes ($blocksize_{KB}$), SecretKey
Ensure: Total time for Decryption in milliseconds (totalDecipherTime)

```
Initialize totalDecipherTime ← 0
Initialize output ← 0
Initialize blocksize ← 0
Function decryptData(blocksize_KB, secretKey)
    blocksize_Bytes ← blocksize_KB × 1024
    cipher ← Cipher.getInstance("AES")
    cipher.init ← (Decrypt_Mode, SecretKey)
Return cipher.doFinal(DecryptedData)
    startTime ← System.currentTimeMillis()
DecrData = decryptData(encryptedData,secretKey)
    endTime ← System.currentTimeMillis()
    totalDecipherTime ← endTime – startTime
    return totalDecipherTime
```

the delay security time $T_{security}$ of processing task i:

$$T_{security(i)} = t_{Encrpt(i)} + t_{Decrpt(i)} \quad (1)$$

where $T_{security}$ is the delay of the security time of the $i-th$ task in the pipeline workflow. The overall time required to provide security for a single pipeline task includes, first, the delay time needed to encrypt the data before the task is executed denoted by $t_{Encrpt(i)}$, and second, the delay time needed to decrypt the data when the task is finished denoted by $t_{Decrpt(i)}$.

$$T_{security(Total)} = \sum_{i=1}^{n} T_{security(i)} \quad (2)$$

In a concise manner, the variable $T_{security(Total)}$ shows the amount of time needed to complete security tasks in the big data workflow. Where n, corresponds to the total number of tasks in the pipeline. To provide the proposed prediction model, we will conduct a very simple experiment by executing the pipeline three times. Each time, the input data will vary, such as the input data will be in different sizes as follows: 6, 12, 24, and on different operational engines as well: 12 and 24. So We will explain the prediction results and their accuracy in the subsequent section 8-C.

C. Simulation framework expansion

The proposed security model technique was implemented by extending the WorkflowSim environment and making modification to some elements of WorkflowSim, such as Scheduler. Fig. 3 provides the structure of our recently implemented component, the security model, which interfaces with other workflowSim layer components.

The WorkflowSim architecture, seen in Fig. 3, includes

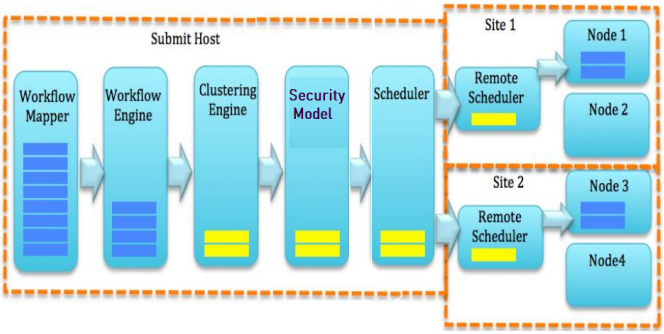

Figure 3. Interaction between new and existing components [8]

our suggested model security model, a workflow scheduler, a clustering engine, a workflow engine, and a workflow mapper. All of these components contribute functions to processing the task as follows:

Workflow mapper has its own function to create workflow task lists and map them to available executable resources. Managing the data relationships between the workflow tasks is the responsibility of a workflow engine. Where A task can begin its execution when all of its parent tasks have been completion successfully.

The main purpose of the clustering engine is to combine smaller tasks into larger tasks using clustering techniques in order to minimize scheduling overhead. A task is a small program that the user intends to execute. The user can execute these tasks either sequentially or in parallel i.e. pipeline. A workflow scheduler is employed to arrange the tasks for accessible resources based on the scheduling algorithm with a specific criterion that is specified by the user or by the provider. As a result, WorkflowSim uses dynamic scheduling, allocating tasks to the remote scheduler only when the appropriate resource is not in use. WorkflowSim also operates via an event-based methodology.

The newly introduced security model encrypts and decrypts data during the preparation stage for any task in the execution phase that requires data transfer over virtual machine, in addition to decrypting upon task completion. This model also focuses on predicting the time needed for encryption and decryption processes, which is a key element in improving performance efficiency.

8. Evaluation

In this section, we use our NGS pipeline for monitoring the impact of the added element which predicts time, in addition to previous work on predicting total time by operating this complex workflow to ensure high accuracy in measuring efficiency.

A. Evaluation of security time Performance

The general aim of this work is to improve the accuracy of predictions of the runtime for large and complex workflow data, such as our NGS pipeline. In considering

our previous work results, we attempted to take into consideration all the requirements of this type of workflow regarding their execution, from protection against latency in accessing shared storage due to the large data volume to other considerations. In our current work, we focused on safeguarding that data and the time taken to apply the protection algorithm. Therefore, we extended a new module to predict the time required to secure the data, incorporated it into our estimating framework, and subsequently conducted experiments using a new set of data. The recently data was necessary of increasing the total number of samples and expansion the range of sets. This enable users of testing predictions when changing both the size of sample and the amount of virtual devices.

B. Setting the experiment

We performed an assessment utilizing the Next-Generation Sequencing (NGS) pipeline in the e-Scale Computing (eSC) environment, which was deployed across 3, 6, and 12 virtual machines on the Azure cloud. The workflow execution threads were supported by using Class D13 virtual machines, each equipped with an 8 core CPU, and RAM with 56 GB, and local SSD storage with (400) GB. The four threads would be executed one after another. The experiment involved sequencing 6, 12, and 24 number of the patient samples, each has data volumes ranging from 98 to 390 GB. We used the shortest running of six samples as training data. Each execution ran on three virtual computers, resulting in a total of 12 workflow execution threads. We collected runtime data from three cloud runs for each evaluation point, the specific number of virtual machines, and input samples. The volume of input sets was calculated by considering the number of patient samples (30–40) used in clinical practice and the cost of operating the pipeline on the cloud. The minimum input size required for the pipeline to properly complete was a set of training consisting of only six samples.

The simulation environment was set up in such a manner that each virtual machine (VM) is simulated corresponding to a specific execution thread of workflow or task in the actual cloud. The model has been trained using 12 (VMs) from WorkflowSim and six samples of patients. We then evaluated it on simulated VMs, specifically 12, 24, and 48. Due to the limitation of one task per (VM), we implemented task scheduling based on the space shared mode. Both the training and testing stages used the shared storage component. When I/O contention was enabled, the model training provided the parameters for optimal simulation: MIPs ($millioninstructionpersecond$)= 1305 and bandwidth = 985 Mb/s. Notably, the value of 985 Mb/s X 12 is significantly closer to the maximum throughput of real cloud Storage. The storage account settings for ingress and egress access are 10 and 15 Gb/s, respectively.

There are three steps in the NGS process: In the first and last steps, data is split into samples. In the stages in between, data is split into groups based on chromosomes. The sample-split stages follow a primarily sequential order, consisting of eight tasks in the first phase and two tasks in the final phase. We copy the raw samples based on how many there are, making sure that each sample goes through a different set of events. The chromosome-split phase begins with a join task, then a set number of tasks are run at the same time on different chromosomal regions, and finally two tasks are run to finish. The total tasks in the pipeline are $9\times N+53$. N represent the total number of input samples (see Fig. 4). As we already said, our WorkflowSim model had 269 actions when it was set up to its fullest potential. Each simulated activity in the actual pipeline consists of multiple process blocks modeled in eSC. This means that running 24 samples requires completing numerous activities and several thousand hours of CPU processing time.

C. Results

After analyzing the provenance information from all the original data that was collected by eSC, we obtained real-time duration and actual size of data of each task available in the workflow. We then converted the duration of time and the size of data (Input / Output) for a single task through a simulation process, which allowed us to identify the training set and obtain predictions of duration time and data sizes from the platform. By comparing these predictions with the real times and good accuracy results was reached, as shown in Fig. 1.

In the initial experiment, we activated shared storage during data transfer without taking into account the security startup time. The graph mentioned above clearly demonstrates that the application of the proposed predicting platform, when relying on the shared storage element for big data workflow, can achieve high prediction accuracy. Wherein, we used a single small prediction point, consisting of six samples on 12 virtual machines, as the training set. Testing larger experiments with the same training point, such as those containing 12 or 24 samples on a 12 virtual machine, revealed an error rate of just 2%.

We used a prediction module to simulate the encryption and decryption processing time, which yielded time estimates directly proportional to the processed data size. We conducted tests on all test groups consisting of 6, 12, and 24 samples, which represent the highest evaluation points and are the furthest from the training set. Notably, we achieved higher accuracy and measured the time in seconds compared to the actual hours required to perform those tasks on Azure cloud computing. It is observed that the security time relative to the operation time is considered negligible, but it is essential to consider all the time consumed in achieving the processing of such workflows. Because it is time-consuming due to their association with big data and they need to account both the time of execution and the cost as illustrated in the Fig. 5.

In order to compare the implementation of the simulation of the security time prediction with the data size in the stage input for each task in the NGS pipeline, we observe

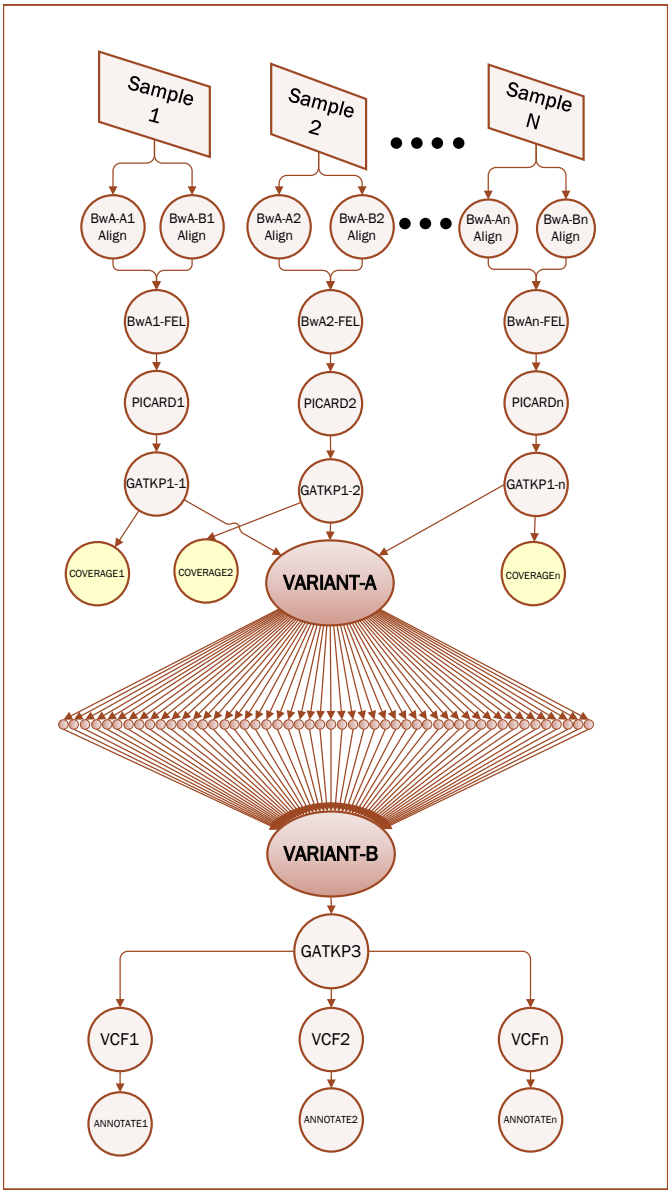

Figure 4. The structure of our NGS pipeline modelled in WorkflowSim.

a correlation where the prediction time nearly doubles the data size for each task, as shown in Fig. 6. We express this proportionality as a function of magnitude, indicating that a half-increase in the volume of data leads approximately a doubling of the prediction security time.

Additionally, Fig. 6 illustrates the extent to which the various configurations scale in comparison to the baseline with 12 cores and 24 cores, with a specific input sample size S representing the values 6, 12, 24, and 48. The $Ds(n)$ is a data size to be configured with n VMs. We define the RST an estimating security time relative to the baseline b-VM configuration as:

$$RSTs(b,n) = 2 \times Ds(b,n) \quad (3)$$

For instance, in the first scenario, we use the same number of resources ($n = 12$) as the number of VMs, but we increase the input data size ($b = 6, 12, and 24$) as a baseline. When equation 3 is achieved with b = 6, n = 12, the estimation of security time, that relative to the chosen baseline is $RSTs(6,12) = 2 * Ds(6,12) = 2 * 500 = 1000$ milliseconds. Additionally, when we increase the baseline to $b = 12$ on the same VMs, $n = 12$, we obtain $RSTs(12,12) = 2 * Ds(12,12) = 2 * 1000 = 2000$ milliseconds. Furthermore, when increasing the baseline to $b = 12$, we will obtain a security time = 4000 milliseconds. In the second scenario, the resources are used with the number of VMs ($n = 24$), while the input data size as a baseline ranges from 6 to 24. The results of the estimated security time for the chosen baseline are $RSTs(6,24)$, $RSTs(12,24)$, and $RSTs(24,24)$ of 1000, 2000, and 4000 milliseconds, respectively. So, using Fig. 6, we can investigate how the most similar behavior affects NGS execution in all scenarios. However, despite the doubled number of sources, there is no impact on the results of the security-time prediction despite the doubling of the number of sources, and the main factor affecting time remains the volume of data.

Thus, the developed framework can deliver enough results to ensure that the application and its security time, for the complex big data workflow, and with appropriate resource deployment and input data volume, can be predicted in an efficient way.

9. Conclusions and Future Work

In this paper, we have presented the novel security model and implementation of the AES algorithm as a new component embedded into the enviroument of WorkflowSim to model security time prediction. The security model component has worked on an AES algorithm that handles one task at a time. Each pipeline step executes numerous sub-workflows concurrently across multiple VMs, prompting the security model to calculate the security time for all sub-workflows at a single level before proceeding to the next step in the pipeline.

Despite the suggested additional model is simple, the evaluation results show that it is a potential method for improving the overall accuracy of runtime prediction. By applying a limited training set comprising only 3 measurements of the least demanding executions, we successfully estimated the runtime with security time of significantly larger configurations. Actually, through simulation, we have seen a significant correlation between the input sample size and the predicted security time. Specifically, when using a large sample size of 24 or 48 samples, the security time consistently rises. Therefore, we have seen a link in which the estimated time it takes to ensure the security of the processed data is almost double the volume of the data for each task.

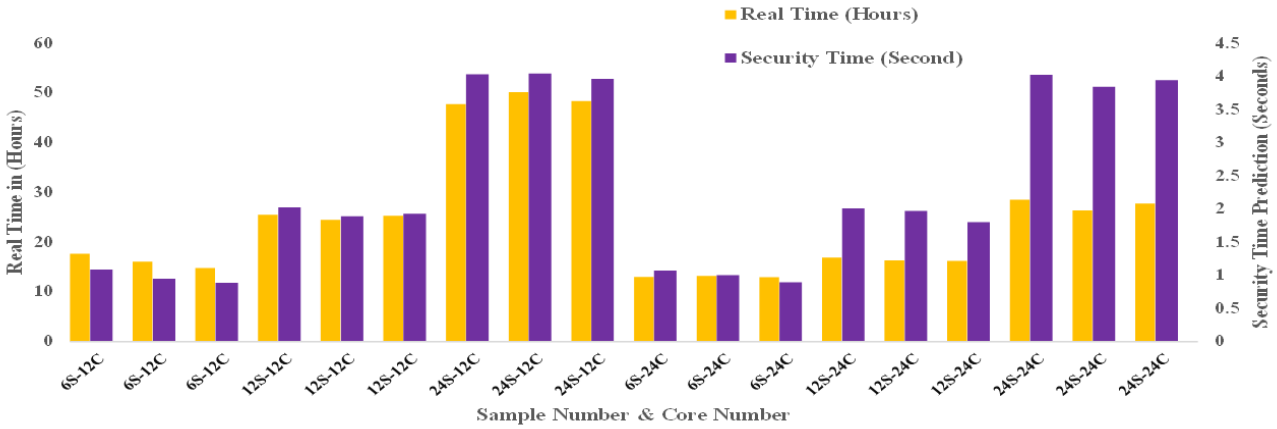

Figure 5. Prediction security Time Compared to Real Time

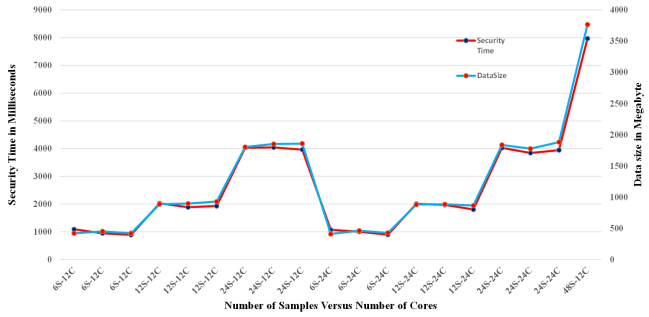

Figure 6. Ratio of Security Time to Data size

It is intriguing that the proposed simple security time model can accurately emulate the much more complex data access mechanism than the real cloud access mechanism. So, given the large amounts of data, we would upgrade the use of this type of platform available to calculate the time of the actual NGS pipeline on the real cloud.

Therefore, even though each step of the pipeline requires numerous sub-workflows to operate concurrently across multiple VMs, the pipeline must wait for all sub-workflows to complete before proceeding to the next step. These simultaneous points at each step make the calculation of sequential and parallel security time consume a similar amount of time; it depends on the size of the input data. Our obtained and demonstrated results suggest that the runtime prediction still has some space for improvement.

In our future work, we would determine whether the suggested security model can simulate more sophisticated workflow process models using other security algorithms (i.e., the RSA algorithm, the DES algorithm, the Blowfish algorithm, etc.). This is an area that requires to be looked into more in the future.

References

[1] M. N. Sadiku, S. M. Musa, and O. D. Momoh, "Cloud computing: opportunities and challenges," *IEEE potentials*, vol. 33, no. 1, pp. 34–36, 2014.

[2] G. Kapil, A. Agrawal, A. Attaallah, A. F. H. Algarni, R. Kumar, and R. A. Khan, "Attribute based honey encryption algorithm for securing big data: Hadoop distributed file system perspective," *PeerJ Computer Science*, vol. 6, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:212425560

[3] F. Llwaah, J. Cała, and N. Thomas, "Runtime performance prediction of big data workflows with i/o-aware simulation," in *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS 2017. New York, NY, USA: Association for Computing Machinery, 2017, p. 74–81.

[4] S. B R, A. K C, and N. Ramaiah, "Resource utilization prediction in cloud computing using hybrid model," *International Journal of Advanced Computer Science and Applications*, vol. 12, p. 2021, 04 2021.

[5] T. Mehmood, S. Latif, and S. Malik, "Prediction of cloud computing resource utilization," 10 2018, pp. 38–42.

[6] T. Pham, J. J. Durillo, and T. Fahringer, "Predicting workflow task execution time in the cloud using a two-stage machine learning approach," *IEEE Transactions on Cloud Computing*, vol. 8, no. 01, pp. 256–268, jan 2020.

[7] A. M. Talib, R. Atan, R. Abdullah, and M. A. Azmi Murad, "Security framework of cloud data storage based on multi agent system architecture - a pilot study," in *2012 International Conference on Information Retrieval Knowledge Management*, 2012, pp. 54–59.

[8] W. Chen and E. Deelman, "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments," in *2012 IEEE 8th International Conference on E-Science*. IEEE, Oct 2012, pp. 1–8.

[9] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and*

Experience, vol. 41, no. 1, pp. 23–50, Jan 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1951445.1951450http://doi.wiley.com/10.1002/spe.995

[10] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, and M. Livny, *Pegasus: Mapping Scientific Workflows onto the Grid*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 11–20.

[11] F. Llwaah, J. Cała, and N. Thomas, "Simulation of runtime performance of big data workflows on the cloud," in *European Workshop on Performance Engineering*. Springer, 2016, pp. 141–155.

[12] A. D. Mauro, M. Greco, and M. Grimaldi, "A formal definition of big data based on its essential features," *Library Review*, vol. 65, pp. 122–135, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:61480668

[13] D. Laney, "3D data management: Controlling data volume, velocity, and variety," February 2001. [Online]. Available: https://tinyurl.com/25x7k4b8

[14] G. Manogaran, C. Thota, D. Lopez, V. Vijayakumar, K. M. Abbas, and R. Sundarsekar, "Big data knowledge system in healthcare," pp. 133–157, 2017. [Online]. Available: https://researchonline.lshtm.ac.uk/id/eprint/4661434/

[15] G. Manogaran, D. Lopez, C. Thota, K. M. Abbas, S. Pyne, and R. Sundarasekar, "Big data analytics in healthcare internet of things," pp. 263–284, 2017. [Online]. Available: https://researchonline.lshtm.ac.uk/id/eprint/4661433/

[16] A. Kashlev and S. Lu, "A system architecture for running big data workflows in the cloud," pp. 51–58, 2014.

[17] A. Kashlev, S. Lu, and A. Mohan, "Big data workflows: A reference architecture and the dataview system," *Services Transactions on Big Data*, vol. 4, pp. 1–19, 01 2017.

[18] J. Cała, E. Marei, Y. Xu, K. Takeda, and P. Missier, "Scalable and efficient whole-exome data processing using workflows on the cloud," *Future Generation Computer Systems*, vol. 65, no. Supplement C, pp. 153 – 168, 2016, special Issue on Big Data in the Cloud.

[19] J. Cała, Y. Xu, E. A. Wijaya, and P. Missier, "From scripted HPC-based NGS pipelines to workflows on the cloud," in *First Int. Work. Cloud Bio (C4Bio 2014)*, 2014.

[20] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755 – 768, 2012, special Section: Energy efficiency in large-scale distributed systems.

[21] G. Belalem, F. Z. Tayeb, and W. Zaoui, *Approaches to Improve the Resources Management in the Simulator CloudSim*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 189–196.

[22] W. Long, L. Yuqing, and X. Qingxin, "Using cloudsim to model and simulate cloud computing environment," pp. 323–328, 2013.

[23] A. Menezes and D. Stebila, "The advanced encryption standard: 20 years later," *IEEE Security Privacy*, vol. 19, no. 6, pp. 98–102, 2021.

[24] R. Gupta, P. Jai, D. J. Singh, and P. Tiwari, "Simulation of aes encryption and decryption algorithm with parallel data execution," *IJECCE*, vol. 3, pp. 2249–71, 06 2012.

[25] G. Viswanath and P. Krishna, "Hybrid encryption framework for securing big data storage in multi-cloud environment," *Evolutionary Intelligence*, vol. 14, 06 2021.

[26] S. Wu, B. Wang, Z. Wang, S. Fan, J. Yang, and J. Li, "Joint prediction on security event and time interval through deep learning," *Computers Security*, vol. 117, p. 102696, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404822000943

[27] D. Saxena, R. Gupta, A. K. Singh, and A. V. Vasilakos, "Emerging vm threat prediction and dynamic workload estimation," *IEEE Transactions on Automation Science and Engineering*, 2023.

[28] D. Saxena and A. K. Singh, "A high up-time and security centered resource provisioning model towards sustainable cloud service management," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2024.

[29] W. Chen and E. Deelman, "Workflowsim: A toolkit for simulating scientific workflows in distributed environments," *2012 IEEE 8th International Conference on E-Science*, pp. 1–8, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:5611210

[30] E. Deelman and Y. Gil, "Managing Large-Scale Scientific Workflows in Distributed Environments: Experiences and Challenges," in *2006 Second IEEE Int. Conf. e-Science Grid Comput.* IEEE, Dec. 2006, pp. 144–144.

[31] E. Deelman, G. Mehta, G. Singh, M. Su, and K. Vahi, "Pegasus: Mapping large-scale workflows to distributed resources," in *Workflows for e-Science, Scientific Workflows for Grids*, I. J. Taylor, E. Deelman, D. B. Gannon, and M. S. Shields, Eds. Springer, 2007, pp. 376–394. [Online]. Available: https://doi.org/10.1007/978-1-84628-757-2_23

[32] 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," *Futur. Gener. Comput. Syst.*, vol. 46, pp. 17–35, May 2015.

Dr. Faris Llwaah achieved a Ph.D. in computer science in 2018 from Newcastle University, UK, in the High Performance of Computing on the Cloud. He received a master's in computer science from Al-Nahrain University in Baghdad, Iraq in 1993. His degree is in computer science from Iraq's University of Mosul. His job is Cyber Security coordinator. He is a Cyber Security Department scientific committee member and rapporteur. He is on the department's examination committee. He has higher education teaching expertise. He is very interested in the performance of all cloud computing applications and computer operating systems.