VECTrust: Trusted Resource Allocation in Volunteer Edge-Cloud Computing Workflows

Ashish Pandey, Prasad Calyam, Saptarshi Debroy*, Songjie Wang, Mauro Lemus Alarcon
University of Missouri-Columbia, USA; *City University of New York, USA
Email:{apfd6,lemusm}@umsystem.edu;{wangso,calyamp}@missouri.edu;sd1998@hunter.cuny.edu*

Abstract

The unprecedented growth in edge resources (e.g., scientific instruments, edge servers, sensors) and related data sources has caused a data deluge in scientific application communities. The data processing is increasingly relying on algorithms that utilize machine learning to cope with the heterogeneity, scale, and velocity of the data. At the same time, there is an abundance of low-cost computation resources that can be used for edge-cloud collaborative computing viz., "volunteer edge-cloud (VEC) computing". However, lack of trust in terms of performance, agility, cost, and security (PACS) factors in edge resources is proving to be a barrier for wider adoption of VEC. In this paper, we propose a novel "VECTrust" model for support of trusted resource allocation algorithms in VEC computing environments for scientific data-intensive workflows. Our VECTrust features a two-stage probabilistic model that defines trust of VEC computing cluster resources by considering trustworthiness in metrics relevant to PACS factors. We evaluate our VECTrust model's ability to provide dynamic resource allocation based on PACS factors, while also enhancing edge-cloud trust in a VEC computing testbed. Further, we show that VECTrust is able to create a uniform and robust probability distribution of salient PACS factor related metrics within diverse bioinformatics workflows execution over batches of workflows.

CCS Concepts

• **Applied computing** → **Bioinformatics**; **Enterprise resource planning**; **Bioinformatics**.

Keywords

Volunteer edge computing, multi-cloud brokering, resource allocation, trust, performance, agility, cost, security.

ACM Reference Format:
Ashish Pandey, Prasad Calyam, Saptarshi Debroy*, Songjie Wang, Mauro Lemus Alarcon. 2021. VECTrust: Trusted Resource Allocation in Volunteer Edge-Cloud Computing Workflows . In *2021 IEEE/ACM 14th International Conference on Utility and Cloud Computing (UCC'21), December 6–9, 2021, Leicester, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3468737.3494099

1 Introduction

Application workflows today tend to handle huge data generated through networked smart edge devices (e.g., biosensors, imaging instruments) and rely on real-time processing capabilities in cloud platforms. However, cloud computing costs can become a barrier for handling scientific application workflows, especially when there is a significant amount of data-related computation tasks involved. New paradigms of cloud computing such as volunteer cloud computing (VCC) [1] are emerging to leverage volunteer contributions of large-scale cloud resources to reduce costs.

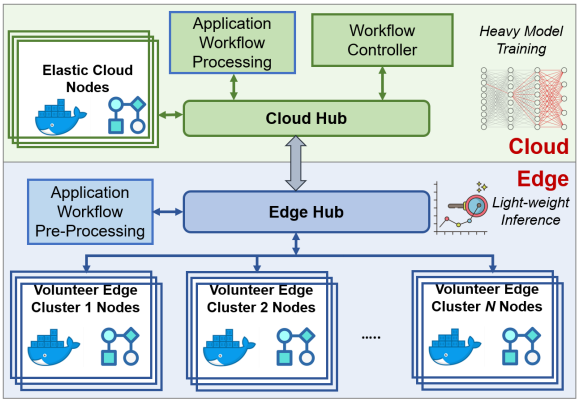


Figure 1: **KubeEdge-based architecture for the orchestration of edge and cloud nodes, where the cloud nodes are used for ML model training and edge nodes are used for ML model inference in scientific application data processing.**

Although VCC provides benefits in terms of cost, the proprietary/sensitive nature of certain scientific applications necessitates partial data processing to be performed at the edge resources i.e., closer to the application user sites. Leveraging emerging technologies such as KubeEdge [2], VCC solutions can be extended to on-demand provision and use an abundance of low-cost edge resources through edge-cloud collaborative computing on a best-effort basis viz., "volunteer edge-cloud (VEC) computing". In addition to providing cost benefits similar to VCC, the VEC paradigm is suitable for the latest generation of compute/data intensive workflows that use machine learning (ML) models to perform heavy training on cloud nodes and light-weight inference on edge nodes.

Figure 1 shows how a KubeEdge-based architecture can be used for the orchestration of edge-cloud resources for compute/data intensive applications that use ML models. Specifically, KubeEdge [2] can help in efficient automation of tasks at volunteer edge nodes and cloud nodes using Docker containers [3], similar to how Kubernetes [4] clusters are managed in predominantly cloud nodes. Integration of VEC with KubeEdge cluster management technology particularly supports small and low-latency computation tasks (including data pre-processing, data quality/privacy checks, ML model inferencing for workflow predictions) at edge nodes that can save burden on network bandwidth connectivity to cloud nodes, while also performing collaborative use of large cloud node resources for heavy computation tasks (e.g., parallel processing for ML model training, huge data movement).

However, a major challenge for wider adoption of the VEC computing paradigm in scientific application workflows relates to ensuring that the volunteer edge resources can be trusted in terms of the performance, agility, cost, and security (PACS) factors, on par with nodes within public clouds. A suitable VEC architecture

UCC'21, December 6–9, 2021, Leicester, United Kingdom

ACM ISBN 978-1-4503-8564-0/21/12…$15.00
https://doi.org/10.1145/3468737.3494099

should present well-defined security protocols, policies, and orchestration mechanisms similar to those in public cloud resources to meet application requirements. It also has to deal with the dynamic and unwarranted nature (i.e., volunteer edge resources can be withdrawn or face availability issues routinely) of VEC resources that disrupt the trust (e.g., in terms of the cost or scalability of edge resources) in the execution of time-critical scientific applications. While trust modeling in cloud computing is a well-researched topic [5] [6], there are limited works on trust modeling in a VEC context. In the VCC context, reputation-based trust is considered assuming capable/homogeneous resources, and resources characterization is performed for extended time periods. In contrast, an effective VEC trust model uniquely needs to: (a) obtain a wide range of data about the containerized edge resources or virtual machines (VMs), and (b) use model-based decisions based on a quick analysis of edge nodes (that can sometimes be austere) resource characterization data – to select trustworthy VEC resources and optimize edge-cloud configurations based on PACS factors.

In this paper, we address the above VEC trust model development challenges by proposing a "VECTrust model" for enabling trusted and optimized resource allocation algorithms in VEC computing for data/compute-intensive scientific application workflows. Our VECTrust features a two-stage probabilistic model that defines trust of VEC computing resources in terms of PACS factors. A probabilistic model is an ideal choice for modeling trust because edge resources can have dynamic behavior towards PACS factors given their voluntary nature and may not exclusively be dedicated to VEC cluster computing. Since the edge resources are voluntary, the edge node provider can alter configurations (increase/decrease capacity or remove availability) on resources by chance. Moreover, geographically distributed VEC resources can fail due to latency issues which can be random in nature and can only be characterized probabilistically. Our two-stage model helps in capturing randomness in PACS factors at geographically co-located VEC resources and also helps in comparing different VEC clusters. We leverage Dirichlet [7] distribution because we need multivariate probabilistic distribution of trust towards PACS at the local intra-cluster stage (i.e., when selecting edge nodes in a single location) and a global inter-cluster stage (i.e., when selecting edge nodes at same or different locations) for dynamic selection of most suited edge/cloud resources to increase the trust levels.

We further validate our VECTrust model in terms of success in workflow executions on trusted resources while simultaneously performing optimization considering PACS factors. Towards this aim, we conduct experiments in a VEC computing testbed featuring diverse bioinformatics workflows (in terms of computing and memory requirements) as well as KubeEdge and Docker technologies. We choose bioinformatics workflows for evaluation because they are popular in the scientific community given the recent rapid advancements in next-generation sequencing (NGS) technologies. Moreover, resource requirements for the bioinformatics workflows vary depending on the type of analysis required on raw NGS data and involve small-scale data quality analysis and privacy-preservation steps, which can be done at edge resources before large computations in the cloud resources. We compare our VECTrust model against three baseline models: (i) reliability-based trust-aware resource allocator algorithm [8] on voluntary cloud resources, (ii) K-nearest neighbor resource selection in voluntary edge resources, and (iii) random selection of resources on voluntary edge resources. By periodically measuring the resource utilization probability distribution during the execution of diverse bioinformatics workflows, we observe whether eligible resources are utilized effectively at the local intra-cluster and global inter-cluster stages. In other words, we show that the VECTrust model is able to assess trusted resources and use them continuously as the workload on the cluster increases, which in effect creates a uniform resource utilization pattern on resources with relatively higher trust.

The remainder of this paper is organized as follows: In Section II, we describe related works. Section III describes the VEC computing trust problem and our VECTrust solution approach. Section IV presents details of our VECTrust model components and their implementation. Section V presents the performance evaluation. Section VI concludes the paper.

2 Related Work

Trust Models in Cloud Computing: Resource trust modeling in cloud computing has been an area of keen interest among researchers. Miscellaneous trust models have been proposed which are based on e.g., auditing, static policies. Surveys such as [9] discuss and compare comprehensive cloud-to-cloud trust paradigms from a federation perspective. In [10], authors address the auditability in clouds via technical and policy-based approaches. Whereas in [11], authors propose a trust-based auditing method where users provide their own security preferences aligned with the cloud provider policies. In [12], authors propose a trust-aware framework to verify and evaluate the security controls established by cloud providers. These works provide important foundations to identify the factors needed for trust assessment and subsequent trust-based classification within a federated cloud environment. Towards trust quantification within a cloud environment, there have been works such as [5] [6] that consider multiple variables and ML models for trust prediction by conducting trust data mining and knowledge discovery. To avoid training requirements and cold-start issues around sufficient data availability in previous ML model-based approaches, there have been alternate and state-of-the-art trust modeling approaches proposed that can be organized either under the reputation-based category or the probability-based category.

Reputation-based Trust Models: In cloud environments, trust between individual entities is typically facilitated by reputation management based on various parameters such as e.g., history, context, collection, representation, and aggregation [10, 13]. Focus typically is on single-source trust quantification, where policies such as QoS parameters and audit assessment and/or accountability factors such as security, reliability, and availability are used as fundamental variables. Other studies such as [6] used ML models to predict reputation primarily based on QoS, and use QoS-based trust values in cloud resource brokering. Trust and reputation modeling under scenarios with incomplete information has also been an area of research. For example, authors in [14] developed a multi-dimensional framework viz., PeerTrust for classifying and comparing trust and reputation systems, and their suitability for a given application even when adequate information is unavailable about a target peer.

A state-of-the-art work on reputation-based trust determination particularly relates to VCC and uses the trust values within a knapsack problem formulation for resource allocation [8]. This work proposes three algorithms for task scheduling on voluntary nodes while leveraging a semi-markov [15] process for trust prediction of voluntary nodes. Their approach focuses on selecting the most trusted resources within a cloud environment while optimizing resource allocations for task execution. However, their trust determination approach requires a long history of underlying task executions and considers stable resources on a cluster for optimization based only on the performance factor amongst PACS factors. In comparison, our VECTrust approach relies on relatively small

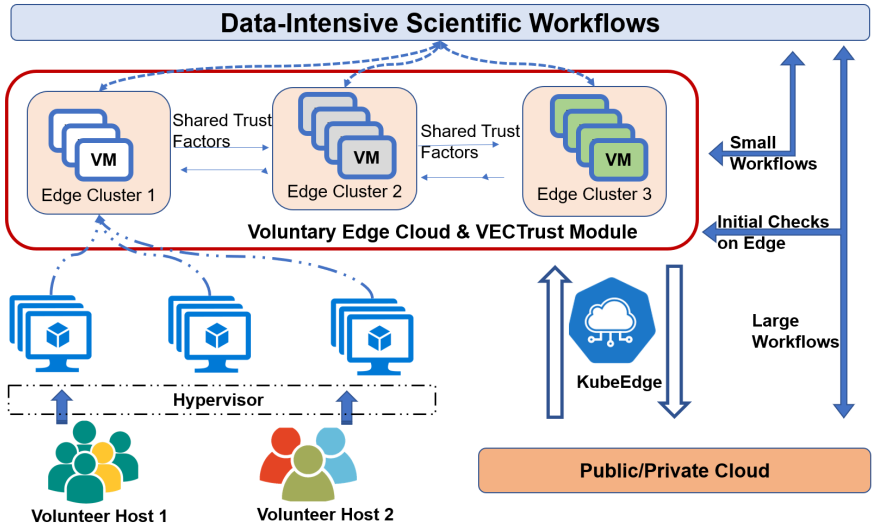


Figure 2: **VEC system overview showing scientific workflows being submitted to a cluster of edge-cloud resources. Edge clusters are created using resources from individual edge node servers. Workflows are transferred to public or private cloud nodes as per the trade-off between requirement of trust and compute resources on the edge and cloud node sides.**

(recent task execution) data amongst volatile voluntary edge resources and determines a trust function by considering all of the PACS factors.

Probability-based Trust Models: For reputation values to hold for a longer time, probabilistic-based trust models have been explored in the context of data-intensive workload management within cloud federations [16, 17]. Among these works, the work in [16] is notable because it is applicable for efficient allocation of federated resources without considering the trust/reputation of the underlying resources. However, such trust-agnostic resource management are not suitable for a VEC system due to the unique challenges such as heterogeneity of volunteer edge resources, and uncertainty of their resource availability due to possible alternations in configurations (increase/decrease capacity or remove availability) on resources by chance. The work in [17] assumes that resource trustworthiness is subjective and proposes a framework based on Analytic Hierarchy Process (AHP) and Fuzzy Simple Additive Weighting (FSAW) to determine trust between service providers and users based on the users' perception. However, their approach relies entirely on the application requirement parameters considered by the users and thus ignores the uncertain or unknown factors of the resource provider, which is commonly the case in a VEC system.

Other notable works such as [18] and [19] propose a collaborative trust model based on the Beta distribution [20] and compare their model against random and deterministic resource selection strategies. Other probabilistic approaches such as [1] propose VCC trust by using the priority of tasks and behavioral changes as trustworthiness metrics. In addition, prior works [7] and [20] have shown that both conservative and optimistic probabilistic strategies are efficient for trust assessment in VCC, respectively. Thus, probabilistic trust models are promising for efficient VEC resource provisioning with an edge-cloud federated environment. Our work builds on these prior probability-based trust models and we present a novel probability-based trust modeling scheme for resource allocation in a VEC system for scientific application workflow management. Specifically, our approach uses a Dirichlet-based probability distribution which can model multiple variables simultaneously. Our two-stage distribution model allows us to characterize edge node metrics such as CPU/RAM utilization, network interfaces, TCP/FTP connections dynamically as well as provide a framework to use both intra-cluster and inter-cluster PACS factors for trust assessment.

3 Background and Terminology

In this section, we first present an overview of a VEC system for data-intensive scientific workflows. Following this, we provide a background on the assumptions and underlying theoretical models relevant for the purposes of performing trusted resource allocation in VEC environments.

3.1 VEC System Overview

A VEC system is comprised of multiple geographically distributed clusters, with each cluster having a set of co-located voluntary diverse edge resources. As shown in Figure 2, the VEC clusters interact with public cloud resources through KubeEdge [2] for management of resource-intensive stages of the scientific workflows. The system leverages VEC edge resources for execution of small workflows and for initial data quality check or privacy-preservation stages of large workflows that are to be scheduled over cloud nodes to incorporate user workflow preferences. KubeEdge is an ideal technology for integrating edge resources and in particular VEC resources which are

- voluntary in nature i.e., the availability of resources is not guaranteed.
- heterogeneous in nature.
- limited in size.
- can be geographically distributed.
- can change behavior in respect to PACS dynamically.

The emergence of latest technologies such as KubeEdge and Docker [3] have created opportunities for novel resource management strategies in VEC systems that work collaboratively with cloud platforms to support data/compute-intensive scientific workflows in domains such as bioinformatics. Specifically, VEC systems equipped with KubeEdge and Docker can leverage (a) resource allocation for ML model training and analytics based on heavy

computation operations of workflows in cloud node resources, and (b) resource allocation model inference and execution of critical decisions about the quality and privacy of data sources at edge node resources.

Similar to ML-based workflows, the execution of many of compute-intensive bioinformatics workflows can be based on results of smaller analytics workflows such as e.g., screening to evaluate if bigger computation tasks are required to be executed. For example, as shown in Figure 2, a large bioinformatics workflow such as PGen [21] [22] which analyzes large next-generation sequencing (NGS) data will not be executed on the cloud node resource if a smaller subset of its initial pipeline for example FastQC does not produce good results at the VEC edge node resources.

Table 1 provides details on the various exemplar bioinformatics workflows used in our VEC system implementation. We remark that - although we use bioinformatics workflows that are exemplar, our work is broadly applicable to any other scientific workflows that can be executed in a VEC system using container technology such as Docker. In the following, we provide additional background on the bioinformatics workflows used in our work. Scientific workflows typically involve a pipeline of processes for accomplishing a scientific objective usually expressed as smaller computation tasks that are dependent on output from previous tasks in the hierarchy. These workflows have a pre-defined structure with stages such as data acquisition, refinement, pre-processing, processing, and storage. The primary motive of these workflows is to provide an easy-to-use environment for individual researchers to expedite the research insights by automating the analytics pipelines within computation workflows. Given that NGS technology has improved dramatically in recent years, costs have dropped and the number and range of sequencing applications have increased exponentially. A wide variety of high-throughput sequencing can be generated from RNA or DNA molecules through NGS library construction. In order to efficiently utilize the large-scale NGS data for analysis, many diverse (in terms of computing and memory requirements) workflows can be created. Often these workflows are data/compute-intensive and might need several days to finish execution even with considerable scale of VEC resources. However, many of these workflows have common initial stages which are relatively lightweight and can be used to assess if the larger/heavy workloads should be executed or not.

An exemplar scientific workflow that is enabled by data from NGS is the PGen [21] workflow. The workflow allows users to identify single nucleotide polymorphisms (SNPs) and insertion-deletions (indels), perform SNP annotations and conduct copy number variation analyses on multiple resequencing datasets. The workflow has initial common processing stages which can be assessed using FastQC and Alignment workflow as noted in Table 1. These smaller workflows are essentially composed of e.g., data quality check tools to assess data being analyzed in PGen workflow and typically take very small time and resources for execution. Since these smaller workflows can be used in a disjoint manner with the larger workflow, they are ideal to demonstrate VECTrust capabilities.

For the purposes of our work, we leverage three small bioinformatics workflows as shown in Table 1 using the Pegasus WMS [23] and have further centralized the input data location using CyVerse [24]. The analysis pipelines for these workflows are split into multi-steps and parallel processes to gain the most efficiency. The workflow describes the dependencies among the tasks as a directed acyclic graph (DAG), where nodes are tasks and edges are the task dependencies. Each computing task can optionally be assigned a number of cores and memory that they can efficiently utilize to run on the edge node or cloud node computation sites.

3.2 Assumptions

The VEC system model assumptions for the purposes of our work are as follows:

- Small virtual machines at different geographic locations can be considered as volunteer edge resources.
- Only small workflows which can be executed within a reasonable time period (i.e., $<$ 1 hours in a VEC system) are considered for this work. Bigger workflows that adopt VCC can benefit from the scope of our VEC work in cases where they support functions for lightweight steps such as data quality checks and pre-processing for privacy-preservation of datasets.
- Time for the flow of controls within a VEC system is assumed to be negligible because the time to execute the considered workflows are on the order of minutes while times of control flows are typically on the order of seconds.
- We assume varying degrees of trustworthiness arises when VMs inside individual edge nodes could be hosted on the same or different physical server resources.

3.3 Dirichlet Distribution

The Dirichlet distribution defines a probability density function (i.e., PDF) for a vector input having the same characteristics as the multinomial parameter θ of which the density needs to be computed. The distribution has the support (the set of points where it has non-zero values) over the parameters of θ i.e., x_k such that the elements are:

$$x_1 ..... x_k, \text{ where } x_i \in (0,1) \text{ and } \sum_{i=1}^{K} x_i = 1$$

where K is the number of elements in θ. Its probability density function is defined as:

$$Dir(\theta|\alpha) = \frac{1}{Beta(\alpha)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

$$\text{where } Beta(\alpha) = \frac{\prod_{i=1}^{K}(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \text{ and } \alpha = (\alpha_1, ...\alpha_k)$$

In the context of our work, θ and its elements correspond to either the number of VMs within a cluster or the number of clusters within a VEC depending on the stage of the solution. The Dirichlet distribution is parameterized by a vector α, which has the same number of K elements as our multinomial parameter θ. So Dir($\theta|\alpha$) can be interpreted as the PDF associated with multinomial distribution θ, given that the distribution has parameter α. For example, as shown in Figure 3, the distribution is uniform (represented by color shades) across the defined three parameters $\{x_1, x_2, x_3\}$ when the α parameters for the distribution are uniform and comparable. It can also be verified that the distribution tends to concentrate towards the parameters having a higher α value. The probability density (PD) is concentrated and larger when the α values are larger, which indicates a more probabilistically deterministic solution. The distributions in Figure 3 showcase the spread of PD getting concentrated as α navigates from {1,1,1} through {5, 5, 5} to {10, 10, 10} suggesting that $[x_1, x_2, x_3]$ are increasingly having the same probability of occurrence. While α of {1,2,4} suggests the probability biasing towards x_3. Hence if θ multinomial shown in Figure 3 represented by $[x_1, x_2, x_3]$ is considered as three virtual machines in a VEC cluster, then α of {10, 10, 10} suggests highest confidence that the machines show equal PD; while α of {1, 1, 1} also means

Table 1: **Exemplar bioinformatics workflows used in our VEC system implementation.**

Workflow Name	Workflow Description	{vCPU, RAM(GB), Data(GB)} Requirements
FactQC	FastQC Quality Check workflow is used conduct the quality control checks on raw sequencing data so that we can remove some low score data before the next step analysis.	{1, 0.8, 5}
Alignment	Alignment workflow is used to arrange the reads of DNA or RNA to a reference genome so that we can know which genes expressed or discovery of new, unannotated transcripts.	{1, 1.2, 10}
RNA-Seq	RNA-Seq analysis allow us to identify the differential expressed genes by performing the pair-wise comparison of experimental groups/ conditions of sequencing data.	{2, 1.5, 10}

the machines have equal PD but there is less confidence on this equal distribution. In contrast, α of {1, 2, 4} suggests x_3 has highest probability distribution with a good level of confidence.

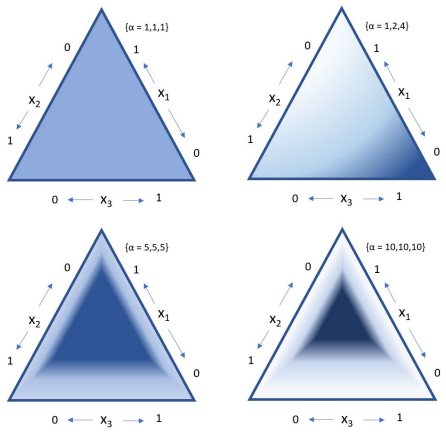


Figure 3: **Sample Dirichlet distribution for a system with three parameters $\{x_1, x_2, x_3\}$ at different parameterized values of α. The distribution in the figures show variation in probability distribution across the parameters as their corresponding α varies.**

4 VECTrust Model

In this section, we first give an overview of our VECTrust solution components. We then discuss details about our approach of intra-cluster and inter-cluster trust modeling. Lastly, we detail the integration of our VECTrust model with a resource allocator algorithm to optimally orchestrate workflows on trusted VEC resources considering PACS factors.

4.1 Overview

Our proposed VECTrust uses a cloud resource broker model that orchestrates VEC resources based on a centralized probability-based trust propagation model. We utilize the Dirichlet distribution across the edge computing resources to create PDFs while considering PACS factors. As shown in Figure 4, the PDFs are generated at the intra-cluster stage and then again at the inter-cluster stage to guide a resource allocator for scheduling workflows at the most suitable VMs. Note that we adapt the popular K nearest neighbor (KNN) based resource allocation in VECTrust for mapping tasks to optimal resources. At the intra-cluster stage, the VMs within a cluster act as the parameter (θ) of that cluster for PDF generation. At the inter-cluster stage, the number of clusters in the VEC resources acts as the primary parameter for the PDF generation.

Figure 5 shows the first stage of our proposed VECTrust model that evaluates the trustworthiness of resources (i.e., VMs within a cluster) through direct interactions between VMs and the dedicated local cluster management server (LCMS). Each of the LCMS contributes to the cluster's PACS factors data that is collected at a central cluster management server (CCMS). Hence, the trustworthiness of resources is assessed at a local cluster level as well as at a more global level when compared with other clusters amongst VEC resources. This approach in VECTrust is different from the traditional trust frameworks [18], where reputation and trustworthiness are evaluated through direct and indirect interactions between two volunteer hosts within a community. In VECTrust, the interaction or feedback between any VM and the LCMS is direct. The LCMS collects data about a VM only from that VM, unlike other methods. However, the collected data from each VM informs the LCMS about its interactions with other VMs based on their shared bandwidth. This process consolidates the responsibility of trust assessment at the LCMS. It then shares the generated PDFs and raw data of VMs within its cluster with the CCMS for further processing at the global level. This multi-stage hierarchical architecture of the VECTrust model allows easy integration of new layers of edge devices such that any LCMS can act as the CCMS for the layers below, thus providing seamless scalability to our solution.

4.2 Intra-cluster Trust Modeling

For each VEC cluster governed within the VECTrust model, one of the VMs act as the LCMS for that cluster. As shown in Figure 5, the LCMS monitors each of the VMs in the VEC cluster to collect data from them for various parameters such as CPU/RAM utilization, firmware, network controllers, sensors, etc. These parameters contribute towards trust assessment based on PACS factors of the VEC resources.

4.2.1 Trust based on performance We assume that the performance of a VM cannot be assessed based on the executed workload because a large workflow expects a relatively larger computation power. Hence, the execution time of a workflow cannot be a benchmark for a VM's performance. We remark that a VM with a better configuration of firmware should be more suited at executing a workload i.e., a better configured VM will execute the same workflow in smaller time compared to a poorly configured VM. Hence, we argue that the optimal use of resources is a better indicator of performance. Authors in [25] indicate that the optimal upper thresholds for CPU utilization for a given set of workloads are between 70% and 90%. We utilize these limits when ranking different VMs for performance at different active workloads. Additionally, while monitoring a VEC cluster, the proposed model considers only those VMs that have active workloads. For the sake of PDF calculation, inactive VMs in VEC resources are given the average percentage value of CPU and RAM utilization of the active VMs. Thus, all VMs in VEC resources have at least comparable CPU and RAM utilization, and thus inactive VMs are also considered when the performance is

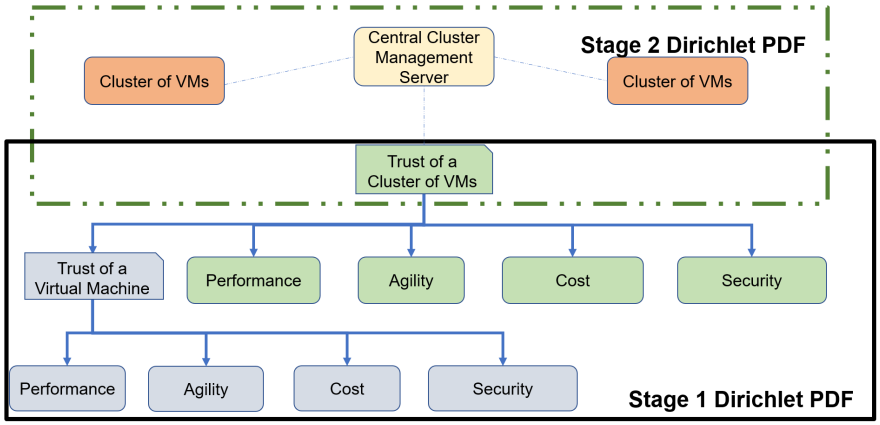


Figure 4: **Logical stages of Dirichlet probability distribution applied at intra-cluster and inter-cluster stages. Blocks in black color identify with factors at intra-cluster stage, while blocks in green relate to factors for inter-cluster stage for probabilistic distribution.**

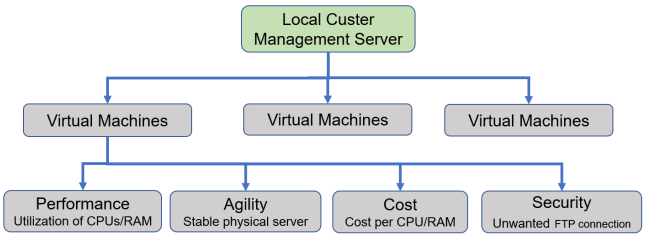


Figure 5: **At intra-cluster stage LCMS collects data from individual VMs geographically collocated and hosted within an edge cluster.**

evaluated. With these values acting as the guiding parameters, we create Dirichlet distribution (PDF) for the cluster with respect to performance to identify best-performing VMs.

4.2.2 Trust based on agility Measuring agility of individual VMs is a convoluted function of many factors that characterize the VMs configuration such as performance and robustness to workload. Specific to a single VM, the robustness can also depend on its behavior measured by sensors (e.g., temperature, fan speeds, voltage, battery state, power usage statistics) when the workload is varying over time. Primary factors to assess the agility of the VMs can be: (i) the ability of a VM to process multiple workflows simultaneously, (ii) the number of failed attempts to assign new workflows to a VM within a cluster on request, and (iii) quality of processors used, which effects task parallelism. All these factors contribute towards the calculation of the α values as discussed in Section III. More specifically, we estimate the agility metric (α) of the nodes using gamma function with the above-mentioned factors as described in [19], where the reliability metric of an individual node is predicted by using a beta distribution on the success and failure rates for weighted execution of high, medium and low priority tasks.

4.2.3 Trust based on cost In order to incorporate cost as a factor for trusted computation, we reduce the cost of execution for workflows when executed on the VEC cluster. Since there is heterogeneity of resources within different clusters, different resources can be available at different prices. We remark that VEC resources are more economically viable compared to commercial cloud providers as the former do not follow a standard cost structure. For cost-based trust computation, we normalize the cost of resources within each cluster on a scale from 1 to 10 and then subtract this value from 10 (the inverse is used since we do not want to use a VM which has a high price). This normalized cost is then used as the α parameter in the trust PDF computation. This ensures that the VMs with lower costs have the highest α value, and consequently have higher trust.

4.2.4 Trust based on security Finally, we create discrete security levels that provide a measure of security based on different parameters. Some of these parameters are computer system, firmware, operating system, version/build, network interfaces (IPs, bandwidth in/out), network parameters, TCP/UDP/FTP connections, etc. For example, a large number of FTP connections in a VM might reflect an unsecure system and thus will belong to a lower security level. We measure the trust of VMs as a function of these levels that contribute to the α values in generating the PDFs of the cluster. More specifically, we categorize security for all individual nodes as either 'good' or 'bad' based on a threshold of TCP/FTP connections over a fixed period of time immediately before the time of trust calculation for scheduling tasks. We use the frequency of good and bad categorization of a node over time as a core factor for its trust estimation. We calculate the security metric (α) for each of the nodes by using the gamma function with the above-mentioned factors of good and bad, and our approach is similar to the reliability metric calculation in [19]. The metrics for PACS factors for each of the VMs are collected into the LCMS on an on-demand basis and their linear function is used to calculate the normalized α values (1 to 10) for different PACS factors. The parameters are then used to generate PDF distributions for the VEC cluster in LCMS and subsequently communicated to the CCMS as needed.

4.3 Inter-cluster Trust Modeling

The VECTrust model generates Dirichlet distribution for the PACS factors at an inter-cluster stage as shown in Figure 6. For generating the PDFs, θ parameters are represented by the number of clusters, and α parameter used for the PACS factors is discussed below.

4.3.1 Trust based on performance The CCMS collects Dirichlet distributions from each of the LCMS of the VEC clusters. A performance normalization factor is calculated by a weighted average of the performance PDFs by using the number of VMs in the individual VEC clusters as weights. The normalized PDFs act as the α value for the second stage of PDF using the Dirichlet distribution for performance.

4.3.2 Trust based on agility At the inter-cluster stage, the CCMS aims to identify the most agile VEC cluster in its set of clusters. The CCMS collects PDFs for agility from all the LCMSs and then transforms them based on the number of probabilistically agile VMs in each of the clusters. The time to add a new VM from a cluster is also one of the factors to assess agility. These factors contribute towards the α values for the clusters to generate PDFs at the inter-cluster stage. In addition, trust in terms of the agility of a VEC resource can change dynamically over its lifecycle. This is because the maximum value of PDs (a discrete number generated when creating PDFs) of a cluster can change over time which depends on a number of factors, such as (a) addition time of a new VM, (b) unexpected failures arising on adding new VMs in a cluster, and (c) resource availability amongst the volunteers themselves. At any time, the velocity of the addition of VMs from the cluster is a major factor towards agility. The CCMS has time-series data of these PDFs for agility, and a higher mean for agility at the cluster level and a lower standard deviation in the PDFs over time indicates an agile and robust cluster towards workloads. Similarly, a large standard deviation indicates the cluster is not robust to cases with workload increase or decrease.

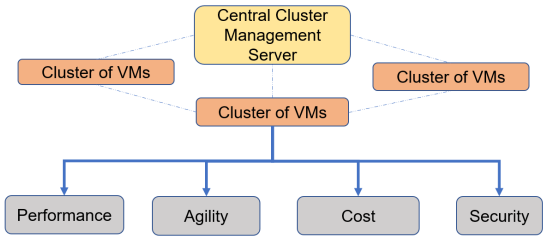


Figure 6: **At inter-cluster stage CCMS collects data from geographically distributed LCMSs of clusters and performs assessment based on PACS factors.**

4.3.3 Trust based on cost The trust factors at the inter-cluster stage are essentially based on the PDs generated from the LCMS servers in the VEC resources. The PDs for cost from each of the clusters are collected and normalized from 1 to 10. The normalized values are then used as α for calculating the cost PDs at the inter-cluster stage.

4.3.4 Trust based on security For security-based on trust at the inter-cluster stage, the VECTrust model first identifies the most secure clusters in a given set. The CCMS collects security PDFs from all LCMSs and transforms them based on the number of probabilistically secure VMs in each of the clusters. These numbers are weighted by the degree of security which is assessed using NIST guidelines [26], and act as the α values for the clusters to generate security PDF at the inter-cluster stage.

4.4 Resource Allocation Methodology

The approach of assessing VEC resources using probabilistic modeling using Dirichlet distribution can dynamically capture dynamic resource behavior. However, different workflows might have their individual functional requirements specified by the user in terms of e.g., the number of vCPUs, RAM, and disk drive space. Such requirements need to be met explicitly for the workflows to perform on par with user expectations. This is an optimization problem of resource selection from available resources in order to fulfill workflow requirements and foster execution of the maximum number of workflows possible. One of our prior works [22] towards such an optimization uses integer linear programming and the popular KNN for heuristic optimization of resource selection for executing workflows. We have thus used the KNN algorithm to select specific VM from the clusters which matche the requirements of a given scientific workflow to be executed. The application of this algorithm ensures that the quantitative requirements for the workflows as stated by the users are necessarily fulfilled. Following this, trusted scheduling of these workflows is performed on the VEC clusters based on the user PACS factor priorities as explained in Algorithm 1.

Algorithm 1: Trust-driven resource selection algorithm based on PACS factors

```
Result: VEC edge node resource on a trusted cluster
while true do
    Periodic incoming workflow execution request;
    factor ← Get workflow factor optimization;
    i is the required VM configuration;
    if factor then
        i ← KNN to find eligible VMs;
        if i ≠ null then
            PDFs at LCMSs for factor;
            PDFs transfer from LCMSs to CCMS;
            cPDF ← PDF at CCMS for factor;
            cluster ← highest distribution from cPDF;
            if i ϵ cluster then
                Execute the worklfow on cluster at i VM node
            end
        else
            PDFs at LCMSs for factor;
            PDFs transfer from LCMSs to CCMS;
            cPDF ← PDF at CCMS for factor;
            cluster ← highest distribution from cPDF;
            Request new VM with i configuration on cluster;
        end
    else
        defaults to 'Performance' factor based trust;
    end
end
```

Intra-cluster Trust computation: VECTrust takes the task outcomes as input for a certain period of time from T_s to T_e, where T_s is the start time and T_e is the end time, and estimates the trust score for each volunteer host within a cluster. Essentially LCMS monitors each of the VMs in the edge node cluster and collects PACS factors data through Docker containers running as web services. LCMS also calculates multiple Dirichlet distributions for each of PACS factors for all VMs within a given cluster by using α values that are normalized from 0 to 5 for a fair evaluation of the best performing VMs in the cluster.

Inter-cluster Trust computation: Such a trust computation is used for behavior detection. The VECTrust model continuously monitors new workflows to be scheduled. It tries to schedule workflows on the best performing cluster with a suitable matching VM by estimating the trust score of each of the clusters dynamically before making a decision of scheduling. When a new workflow execution request comes into the queue with demand for any of PACS factors, then the CCMS uses KNN to find the eligible VM resources that fulfill the workflows' CPU and RAM requirements. The CCMS collects data from each of LCMS in the VEC system and normalizes maximum PD from each of LCMS for PACS factors in the range of 0 to 10. Subsequently, it uses these normalized densities as α values for the second level of Dirichlet for the clusters. The VEC cluster with the highest PD value represents the most ideal cluster for assigning the workflow and the VM with the highest PD within that cluster is the best suited VM for executing the workflow. The CCMS ranks these clusters and eligible VMs inside them according to their PD values. The selected VM is then assigned the workflow using KubeEdge. While the workflows are executing, the CCMS and LCMS periodically collect Dirichlet distribution data to assess the uniformity of probability distribution with performance as the

Table 2: **VM/Worker nodes description at the GENI Edge Nodes.**

Resource Name	Core	RAM	Disk
XOSmall	1	1GB	10GB
XOMedium	1	3GB	25GB
XOLarge	2	6GB	50GB

primary factor. Lesser value of the standard deviation of *alpha* values suggests better utilization of resources, while a higher standard deviation suggests certain resources are under-performing or idle.

5 VECTrust Evaluation

In this section, we first describe the testbed setup to evaluate our VECTrust model in terms of PACS factors. Following this, we show benefits of our VECTrust approach in comparison with state-of-the-art approaches in the context of resource selection, and their impact on the execution times of the application workflows.

5.1 Testbed Setup

For the scientific workload on the VEC computing resources, we have used the three bioinformatics workflows: (a) FastQC, (b) Alignment, and (c) RNA-Seq workflows listed in Table 1. Each workflow requires a diverse amount of resources, which are provisioned in the testbed environment for the corresponding expected execution. The testbed resources for evaluation of our proposed VECTrust model are obtained from the NSF-funded GENI [27] infrastructure and the GCP [28], similar to the architecture described in Figure 1. As shown in Figure 7, three clusters of resources are created across three geographically different locations (RENCI, NCSU & Texas A&M), which are controlled via KubeEdge instance in GCP. Each edge cluster contains three different VM configurations namely: (a) XOLarge, (b) XOMeduim and (c) XOSmall as shown in Table 2. All VMs which are part of the VEC testbed have public IPs as they host RESTful web services via Docker containers for communication and data transfer tasks.

The XOLarge VMs within the cluster act as LCMS for the cluster. The CCMS is created on an independent GCP VM instance having 2 cores and 4GB RAM. KubeEdge is installed on all the nodes such that the CCMS acts as the master node and the 9 VMs across the GENI cloud infrastructure act as worker nodes. Each VM has a web service that reports PACS factors information relevant to a VM to its LCMS and CCMS. This information is utilized to create Dirichlet distributions which govern trust calculations that direct the scheduling of workflows on a worker node over VEC computing resources.

5.2 Evaluation Methodology

Our experiments' goal is to evaluate the VECTrust model's effectiveness and efficiency by measuring its impact in terms of PACS factors during repeated and pre-determined order of execution of workflows on the VEC testbed. Towards this goal, workflows are submitted with one of the PACS factor intents related to trust. The trusted resource allocation for cost is not considered for the purposes of this work because: (a) we consider voluntary edge resources that do not incur any cost, and (b) the cost quantitatively depends on the price of cloud resources, which can be easily determined and varies depending on the cloud platform our VECTrust is deployed. We compare VECTrust's efficiency in trusted resource allocation for workflow execution on VEC resources by executing different number of workflows against three competing resource provisioning strategies: (i) Random selection of VM resources, which represents the most common resource selection as per availability, (ii) KNN algorithm used for selection, which is a popular algorithm for recommendation systems [22], and (iii) Unobtrusive utilization-reliability aware scheduling algorithm (U^2Trust) [8], which uses a semi-markov process [15] for reliability-based trust prediction of voluntary nodes and is based on formulating the resource scheduling problem as a knapsack problem.

5.2.1 Trust based on Performance For evaluating trusted workflow execution based on performance in VEC resources, we create 12 workflow batches as shown in Figure 8 with different number of workflows mentioned in Table 1. Each batch comprises of a set of workflows (e.g., 5, 10, 15, and so on) in a fixed order. This implies that - when a batch is executed, the same workflow is added to the testbed at any time t, irrespective of the scheduling algorithm. This ensures the same arrival pattern or intervals between workloads on the testbed for a fair comparison between the resource allocation algorithms. A new workflow is added in 5 minute intervals, which is the average execution time for our slowest workflow i.e., FastQC.

The results in Figure 8(a) show that U^2Trust performs better than KNN since it uses reliability as well as knapsack formulation to find nodes, while KNN only uses mapping of resource required compared with resources available in nodes. However, our proposed VECTrust model performs better than both the state-of-the-art approaches as our trust model identifies higher performing VEC clusters as well as higher performing VMs in those clusters while considering the reliability of nodes. Although U^2Trust performance is comparable with VECTrust, recall that our VECTrust model relies only on a short history of underlying volunteer edge resources to capture volatile fluctuations in resource performance during the allocation process.

5.2.2 Trust based on Agility In order to evaluate trusted workflow execution based on agility, we add workflows with relatively higher resource requirements which trigger the CCMS to add new VMs in one of the VEC clusters. We iterate this process multiple times (i.e., 5 times) for each of the workflows using VECTrust and three other competing approaches: (i) KNN (cluster with VMs closest to the workflow requirement is selected for adding a new VM), (ii) Random and, (iii) U^2Trust.

As shown in Table 3, the average time to add a VM in the VEC cluster suggested by VECTrust is lesser compared to the other competing approaches. The best drop in time from the second-best competing solution was for RNASeq at 79%. This is because of the VECTrust use of the PD for agility during identification of the VEC cluster where a new VM can be added faster. Note that the VECTrust consistently outperforms U^2Trust since the latter estimates trust of resources using a large history of workflow execution, which is not available in the voluntary edge cluster resources that can be intermittent in availability, and subject to possible alternations in configurations. In addition, U^2Trust can only assess and add an already known node in a cluster without considering the trust of the cluster itself, whereas VECTrust always adds nodes to trusted clusters.

5.2.3 Trust based on Security In order to compare security based on trusted execution between the competing algorithms, we use the NIST guidelines [26] to evaluate the risk of scheduling workflows. The NIST guidelines help us to quantitatively evaluate the security of individual edge nodes on a scale of 1 to 10 in terms of their vulnerabilities during workflow execution in the VEC system. Using these quantitative values, we correspondingly generate security PDs for each cluster for different baselines and compare them to assess resulting differential security postures. To calculate the security risk in the three competing model cases, a batch of 20

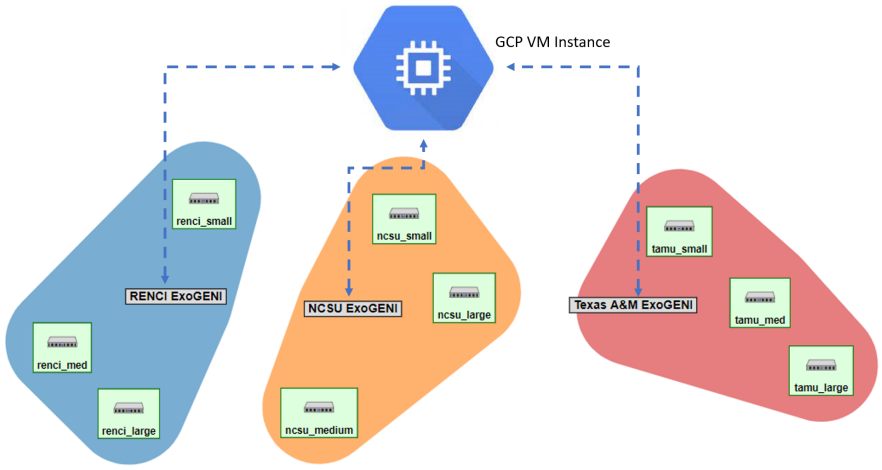
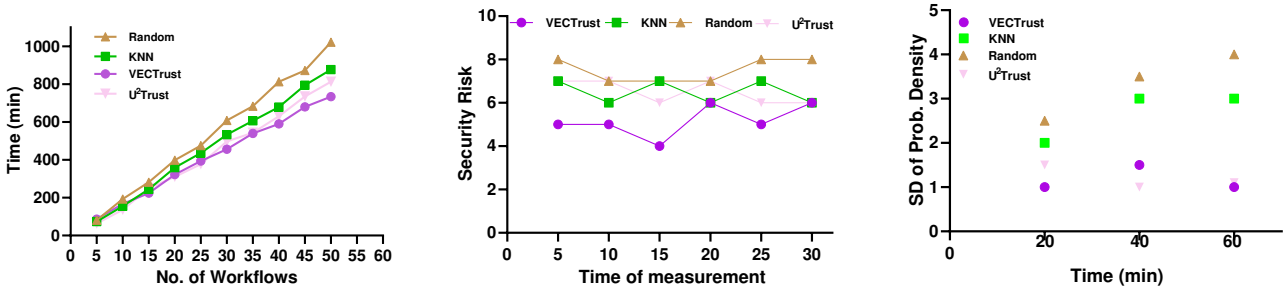


Figure 7: **KubeEdge testbed using a GCP VM instance acting as the CCMS, and GENI nodes acting as smaller remote edge clusters.**



(a) (b) (c)

Figure 8: **(a) Performance comparison when workflows are executed with different batch sizes on the same VEC resources with: (i) VECTrust, (ii) KNN, and (iii) Random selection algorithms; (b) Security risk comparison with a 20 workflow batch that is used to execute on the VEC computing resources; the average security risk is calculated at 5 minute time intervals; (c) Cluster utilization efficiency comparison using standard deviation of resource utilization PDs for the scenarios: (i) VECTrust, (ii) KNN (iii) Random selection (iv) and U^2Trust .**

Table 3: **Agility comparison between different trust models for different workflows with new VM additions, averaged over 5 repetitions.**

Workflow	VECTrust	KNN	Random	U^2Trust	% Drop
FastQC(small) time in secs.	125	144	175	150	86%
Alignment (medium) time in secs.	140	172	185	165	84%
RNASeq (large) time in secs.	154	193	210	197	79%

workflows are executed. When new workflows are added with 5 minutes intervals to the VEC system, the security PD of the identified VM for execution is considered as the security risk for the competing baselines.

As shown in Figure 8(b), the average security risk for scheduling with VECTrust is less compared to other approaches as the VECTrust model assigns workflows on relatively more secure VMs. U^2Trust performs similar to KNN as both methods rely on finding the right node for workflow execution and do not consider security risks. In contrast, our VECTrust performs better than these approaches as it searches for secure nodes based on their PD for resource allocations of application workflows.

5.2.4 Cluster Utilization Efficiency We also measure the efficiency of the effective cluster utilization by comparing the Dirichlet distributions of resource utilization periodically (i.e., at 10 minute intervals) while a batch of 20 workflows is added to the VEC system at 5 minute intervals. As shown in Figure 8(c), the standard deviation of the PDFs for VECTrust is lower and has minor variations at all times measured. In contrast, the random selection algorithm shows the maximum standard deviation and has relatively more variance at all times. This is because the VECTrust model identifies better performing VEC edge nodes and schedules more workflows on those nodes repeatedly, thus reducing risks of failures and delays in workflow execution. For longer time of execution (≈60 minutes), we can see that the U^2Trust performs similar to VECTrust because resource scheduling becomes critical when the number of tasks increase in the queue and U^2Trust focuses on improving utilization of the system resources using knapsack modeled scheduling.

We remark that U^2Trust will perform similar or better than VECTrust in voluntary cloud environments where resources are larger and static in nature with less chance of configuration alteration. Further, larger number of workflows with very diverse resource requirement can be better scheduled using knapsack algorithms used in U^2Trust. However, VECTrust is a better solution for VEC computing environments where resources are generally volatile

in terms of their PACS factors (as shown in results in Table 3 and Figures 8(a) - (c)).

6 Conclusion

In this paper, we proposed a trust modeling approach viz., VECTrust for a new paradigm of voluntary edge cloud (VEC) computing which is ideal for executing scientific workflows that use machine learning for model training in cloud nodes and model inference at edge nodes. These workflows leverage edge node resources that are jointly orchestrated with cloud node resources using emerging cluster management technologies such as KubeEdge. The proposed VECTrust features a two-stage probabilistic model and defines trust in terms of performance, agility, cost, and security (PACS) factors for the execution of large workflows that are commonly seen in scientific application domains such as bioinformatics. Our KubeEdge based testbed for VECTrust evaluation uses resources from GCP and GENI platforms, wherein geographically distributed clusters comprising of at least three VMs each were created to simulate a VEC system. We show through our evaluation experiments that our VECTrust model increases trust based on PACS factors in resource allocation in a VEC system when compared to state-of-the-art approaches, when executing data/compute-intensive workflows. VECTrust also continuously improves resource balancing across a VEC system by allocating resources that create a uniform probabilistic distribution.

Future work can include dynamic capture of a given set of workflows' characteristics in terms of PACS factors for pertinent development of trust models. Particularly, this will help in dynamic machine learning-based optimization within VEC trust models to cater to diverse workflows requirements and, handle diverse job arrival and cluster scheduling patterns in very large VEC systems.

Acknowledgment

This work was supported by the National Science Foundation under award number OAC-1827177. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Y. S. Alsenani, G. V. Crosby, K. R. Ahmed, and T. Velasco, "ProTrust: A Probabilistic Trust Framework for Volunteer Cloud Computing for IoT application," *IEEE Access*, Vol. 8, PP. 135059-135074, 2020.
[2] KubeEdge,"https://kubeedge.io/en/".[Online].[Last accessed: Sep. 2021]
[3] Docker"https://www.docker.com/".[Online].[Last accessed: Sep. 2021]
[4] Kubernetes,"https://kubernetes.io/".[Online].[Last accessed: Sep. 2021]
[5] X. Li and J. Du, "Adaptive and attribute-based trust model for service-level agreement guarantee in cloud computing," *IET Information Security*, vol. 7, no. 1, pp. 49-50, Mar. 2013. doi: 10.1049/iet-ifs.2012.0232
[6] C. Mao, R. Lin, C. Xu, and Q. He, "Towards a trust prediction framework for cloud services based on PSO-driven neural network," *IEEE Access*, vol. 5, pp. 2187-2199, 2017. doi: 10.1109/ACCESS.2017.2654378
[7] A. Josang, and J. Haller, "Dirichlet reputation systems," In *2nd International Conference on Availability, Reliability and Security (ARES'07)*, pp. 112-119, Apr. 2007.
[8] T. Mengistu, D. Che and S. Lu, "Multi-Objective Resource Mapping and Allocation for Volunteer Cloud Computing," 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), 2019, pp. 344-348.
[9] U. Ahmed, I. Raza, and S. A. Hussain, "Trust evaluation in cross-cloud federation: Survey and requirement analysis," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1-37, 2019. doi: 10.1145/3292499
[10] R. K. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "TrustCloud: A framework for accountability and trust in cloud computing," In *2011 IEEE World Congress on Services*, pp. 584-588, Jul. 2011. doi: 10.1109/SERVICES.2011.91
[11] S. Rizvi, K. Karpinski, B. Kelly, and T. Walker, "Utilizing third party auditing to manage trust in the cloud," *Procedia Computer Science*, vol. 61, pp.191-197, 2015. doi: 10.1016/j.procs.2015.09.192
[12] S. M. Habib, V. Varadharajan, and M. Mühlhäuser, "A trust-aware framework for evaluating security controls of service providers in cloud marketplaces," In *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 459-468, Jul. 2013.
[13] M. Nguyen, S. Debroy, P. Calyam, Z. Lyu, and T. Joshi, "Security-aware Resource Brokering for Bioinformatics Workflows across Federated Multi-cloud Infrastructures," In *Proceedings of the 21st International Conference on Distributed Computing and Networking*, pp. 1-10, Jan. 2020. doi: 10.1145/3369740.3369791
[14] Z. Noorian, and M. Ulieru, "The state of the art in trust and reputation systems: a framework for comparison," *Journal of theoretical and applied electronic commerce research*, vol. 5, no. 2, pp. 97-117, 2010. doi: 10.4067/S0718-18762010000200007.
[15] T. M. Mengistu, D. Che, A. Alahmadi and S. Lu, "Semi-Markov Process Based Reliability and Availability Prediction for Volunteer Cloud Systems," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 2018, pp. 359-366, doi: 10.1109/CLOUD.2018.00052.
[16] A. Rezgui, N. Davis, Z. Malik, B. Medjahed, and H. S. Soliman, "CloudFinder: A system for processing big data workloads on volunteered federated clouds," *IEEE Transactions on Big Data*, vol. 6, no. 2, pp. 347-358, 2017. doi: 10.1109/TBDATA.2017.2703830
[17] M. Alhanahnah, P. Bertok, Z. Tari, and S. Alouneh, "Context-aware multifaceted trust framework for evaluating trustworthiness of cloud providers," *Future Generation Computer Systems*, vol. 79, pp. 488-499, 2018. doi: 10.1016/j.future.2017.09.071
[18] A. Celestini, A. L. Lafuente, P. Mayer, S. Sebastio, and F. Tiezzi, "Reputation-based cooperation in the clouds," In *IFIP International Conference on Trust Management*, pp. 213-220, Jul. 2014. doi: 10.1007/978-3-662-43813-8_15
[19] Y. Alsenani, G. Crosby and T. Velasco, "SaRa: A Stochastic Model to Estimate Reliability of Edge Resources in Volunteer Cloud," 2018 IEEE International Conf. on Edge Computing (EDGE), 2018, pp. 121-124.
[20] A. Josang, and R. Ismail, "The beta reputation system," In *Proceedings of the 15th bled electronic commerce conference*, Vol. 5, pp. 2502-2511, Jun. 2002.
[21] Y. Liu, S. M. Khan, J. Wang, M. Rynge, Y. Zhang, S. Zeng, S. Chen, J. V. M. Dos Santos, B. Valliyodan, P. Calyam, and N. Merchant, "PGen: large-scale genomic variations analysis workflow and browser in SoyKB," *BMC bioinformatics*, vol. 17, no. 13, pp. 177-186, Oct. 2016.
[22] A. Pandey, Z. Lyu, T. Joshi, and P. Calyam, "OnTimeURB: Multi-Cloud Resource Brokering for Bioinformatics Workflows," In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 466-473, Nov. 2019.
[23] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, W., R. F. Da Silva,, M. Livny, and K. Wenger, "Pegasus, a workflow management system for science automation," *Future Generation Computer Systems*, vol. 46, pp. 17-35, 2015.
[24] CyVerse|Cyberinfrastructure for Data Management and Analysis, "https://www.cyverse.org/". [Online]. [Last accessed: Sep 2021]
[25] F. Al-Haidari,M. Sqalli, and K. Salah, "Impact of cpu utilization thresholds and scaling size on autoscaling cloud resources," In *2013 IEEE 5th International Conference on Cloud Computing Technology and Science* , vol. 2, pp. 256-261, Dec. 2013. doi: 10.1109/CloudCom.2013.142
[26] Joint Task Force Transformation Initiative, "Guide for Conducting Risk Assessments", *NIST Special Publication 800-30*, 2012.
[27] Geni, Exploring Networks of The Future, "https://www.geni.net/". [Online]. [Last accessed: Sep. 2021]
[28] Google cloud platform,"https://cloud.google.com/". [Online]. [Last accessed: Sep. 2021]