Trusted Resource Allocation in Volunteer Edge-Cloud Computing for Scientific Applications

---

DISSERTATION

A Thesis presented to

the Faculty of the Graduate School

at the University of Missouri

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

ASHISH PANDEY

Dr. Prasad Calyam, Thesis Supervisor

DECEMBER, 2022

The undersigned appointed by the Dean of the Graduate School, have examined the dissertation entitled:

TRUSTED RESOURCE ALLOCATION IN VOLUNTEER

EDGE-CLOUD COMPUTING FOR SCIENTIFIC APPLICATIONS

presented by Ashish Pandey.

a candidate for the degree of Doctor of Philosophy and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Prasad Calyam

Dr. Praveen Rao

Dr. Dong Xu

Dr. Matthew Maschmann

ACKNOWLEDGEMENT

I would like to acknowledge the faculty members and other students who contributed to this work and thank them for their support: Songjie Wang, Zhen Lyu, Dmitrii Chemodanov, Mauro Lemus Alarcon, Dr. Trupti Joshi and all VIMAN Lab students. Thanks for all your encouragement!

A very special gratitude goes out to my research advisor – Dr. Prasad Calyam for giving me the opportunity to work with him and guiding me throughout my research. He has been a great role model for me not just in research but also in life. His dedication to science and his work ethic is very inspiring to me.

And finally, last but by no means least, I am grateful to my parents Ramesh Chandra Pandey and Sunita Pandey who have despite their financial hardships provided me with means to pursue my dreams and have been a constant source of moral and emotional support in my life. I am also grateful to my other family members and incredible friends who have supported me along the way and have guided me in the right direction.

TABLE OF CONTENTS

LIST OF PUBLICATIONS

1. A. Pandey, Z. Lyu, T. Joshi and P. Calyam, "OnTimeURB: Multi-Cloud Resource Brokering for Bioinformatics Workflows," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2019, pp. 466-473, doi: 10.1109/BIBM47256.2019.8983386. *Acceptance rate is 18%*

2. A. Pandey, S. Wang and P. Calyam, "Data-intensive Workflow Execution using Distributed Compute Resources," 2019 IEEE 27th International Conference on Network Protocols (ICNP), 2019, pp. 1-2, doi: 10.1109/ICNP.2019.8888119.

3. A. Pandey, P. Calyam, Z. Lyu and T. Joshi,"Fuzzy-Engineered Multi-Cloud Resource Brokering for Data-intensive Applications," 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2021, pp. 257-266, doi: 10.1109/CCGrid51090.2021.00035. *Acceptance rate is 26%*

4. A. Pandey, P. Calyam, S. Debroy*, S. Wang, M. L. 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. *Acceptance Rate is 33.8%*

5. K. Singh et al., "A Formative Usability Study to Improve Prescriptive Systems for Bioinformatics Big Data," 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020, pp. 735-742, doi: 10.1109/BIBM 49941.2020.9313097. *Acceptance rate is 19%*

6. K. Vekaria, P. Calyam, S. Sivarathri, S. Wang, Y. Zhang, A. Pandey, C. Chen, D. Xu, T. Joshi and S. S. Nair, (2020). "Recommender-as-a-service with chatbot guided domain-science knowledge discovery in a science gateway". Concurrency

and Computation: Practice and Experience. 33. 10.1002/cpe.6080. *Impact Factor: 1.447*

7. S.S. Sivarathri, P. Calyam, Y. Zhang, A. Pandey, C. Chen, D. Xu, T. Joshi and S.S. Nair (2019). "Chatbot Guided Domain-science Knowledge Discovery in a Science Gateway Application". 14th Gateway Computing Environments Conference (Gateways), 2019.

8. M. L. Alarcon, R. Oruche, P. Calyam, A. Pandey "Cloud-based data pipeline orchestration platform for covid-19 evidence-based analytics",Book Chapter for 'Novel AI and Data Science Advancements for Sustainability in the Era of COVID-19', Elsevier book publication, 2021.

9. A. Pandey, R. Surya*, M. Maschmann*, P. Calyam "Reinforcement Learning based Carbon Nano Tube Growth Automation". IEEE AIPR Workshop 2021

10. A. Pandey, P. Calyam, Z. Lyu, S. Wang, D. Chemodanov, T. Joshi "Knowledge-Engineered Multi-Cloud Resource Brokering for Application Workflow Optimization". IEEE Transactions on Network and Service Management (TNSM) 2022 (Minor Revision) *Impact Factor: 4.195*

11. R. Oruche, E. Milman, M. Lemus, X. Cheng, A. Pandey, S. Wang, P. Calyam, K. Kee, "Chatbot-assisted Plug-in Management Middleware for Programmable Science Gateways", Wiley Concurrency and Computation: Practice and Experience (CCPE), 2022. *Impact Factor: 1.447*

12. M. L. Alarcon, M. Nguyen, A. Pandey, S. Debroy and P. Calyam. 2022 "VECFlex: Reconfigurability and Scalability for Trustworthy Volunteer Edge-Cloud supporting Data-intensive Scientific Computing", 2022 IEEE/ACM 14th International Conference on Utility and Cloud Computing (UCC'22), 2022 (Accepted)

13. S. Wang, R. Neupane, A. Pandey, X. Cheng and P. Calyam, "Online Learning Platform for Application-Inspired Cloud and DevOps Curriculum," 2021 IEEE 28th International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW), 2021, pp. 35-42, doi: 10.1109/HiPCW54834.2021.00012.

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Data-intensive science applications in fields such as e.g., bioinformatics, health sciences, and material discovery are becoming increasingly dynamic and demanding with resource requirements. Researchers using these applications which are based on advanced scientific workflows frequently require a diverse set of resources that are often not available within private servers or a single Cloud Service Provider (CSP). For example, a user working with Precision Medicine applications would prefer only those CSPs who follow guidelines from HIPAA (Health Insurance Portability and Accountability Act) for implementing their data services and might want services from other CSPs for economic viability. With the generation of more and more data these workflows often require deployment and dynamic scaling of multi-cloud resources in an efficient and high-performance manner (e.g., quick setup, reduced computation time, and increased application throughput). At the same time, users seek to minimize the costs of configuring the related multi-cloud resources. While performance and cost are among the key factors to decide upon CSP resource selection, the scientific workflows often process proprietary/confidential data that introduces additional constraints of security postures. Thus, users have to make an informed decision on the selection of resources that are most suited for their applications while trading off between the key factors of resource selection which are performance, agility, cost, and security (PACS). Furthermore, even with the most efficient resource allocation across multi-cloud, the cost to solution might not be economical for all users which have led to the development of new paradigms of computing such as volunteer computing where users utilize volunteered cyber resources to meet their computing requirements. For economical and readily available resources, it is essential that such volunteered resources can integrate well with cloud resources for providing the most efficient computing infrastructure for users.

In this dissertation, individual stages such as user requirement collection, user's resource preferences, resource brokering and task scheduling, in lifecycle of resource brokering for users are tackled. For collection of user requirements, a novel approach through an iterative design interface is proposed. In addition, fuzzy interference-based approach is proposed to capture users' biases and expertise for guiding their resource selection for their applications. The results showed improvement in performance i.e. time to execute in 98% of the studied applications. The data collected on user's requirements and preferences is later used by optimizer engine and machine learning algorithms for resource brokering. For resource brokering, a new integer linear programming based solution (OnTimeURB) is proposed which creates multi-cloud template solutions for resource allocation while also optimizing performance, agility, cost, and security. The solution was further improved by the addition of a machine learning model based on naive bayes classifier which captures the true QoS of cloud resources for guiding tem-

plate solution creation. The proposed solution was able to improve the time to execute for as much as 96% of the largest applications.

As discussed above, to fulfill necessity of economical computing resources, a new paradigm of computing viz-a-viz Volunteer Edge Computing (VEC) is proposed which reduces cost and improves performance and security by creating edge clusters comprising of volunteered computing resources close to users. The initial results have shown improved time of execution for application workflows against state-of-the-art solutions while utilizing only the most secure VEC resources. Consequently, we have utilized reinforcement learning based solutions to characterize volunteered resources for their availability and flexibility towards implementation of security policies. The characterization of volunteered resources facilitates efficient allocation of resources and scheduling of workflows tasks which improves performance and throughput of workflow executions. VEC architecture is further validated with state-of-the-art bioinformatics workflows and manufacturing workflows.

Keywords: *Multi-Cloud Resource Brokering*, *Scientific Workflow Applications*, *Volunteer Edge Computing*, *Trusted Resource Allocation*.

CHAPTER 1

Introduction

1.1 Data-intensive and Compute-intensive Applications

Data-intensive science applications in fields such as e.g., bioinformatics, health sciences, and high-energy physics are becoming increasingly dynamic with resource requirements. These applications often require specialized instruments and computing, networking, and storage resources (e.g., scientific instruments, supercomputers, federated data repositories, public clouds [1], [2]). Researchers using these applications which are based on advanced scientific workflows frequently require a diverse set of resources that is often not available within a single CSP and demands synergy between multiple CSPs. They seek to create these analytics workflows to frequently utilize cloud solutions easily, efficiently, and with high performance (e.g., high throughput), while containing costs and time for configuring the necessary resources. Moreover, workflows for bioinformatics often process proprietary, private, and confidential pre-publication data within scaling pipelines. Moreover, most distributed resource management approaches and tools do not provide user-friendly interfaces and reusable templates for easy resource provisioning.

1.2 Scientific Workflows

A scientific workflow is the description of a process for accomplishing a scientific objective, usually expressed in terms of tasks and their dependencies. Typically, scientific workflow tasks are computational steps for scientific simulations or data analysis steps. Common elements or stages in scientific workflows are acquisition, integration,

reduction, visualization, and publication (e.g., in a shared database) of scientific data. The tasks of a scientific workflow are organized (at design time) and orchestrated (at runtime) according to dataflow and possibly other dependencies as specified by the workflow designer. Workflows can be designed visually, e.g., using block diagrams, or textually using a domain-specific language.

1.2.1 Science Gateways

Recent science and engineering research tasks are increasingly becoming data-intensive and thus relying on workflows to automate integration and analysis of voluminous data to test hypotheses. For example, research and training in neural science and engineering increasingly deal with diverse and voluminous multi-parameter data 1, posing unique challenges outlined in an NSF iNeuro report 2 as limited access to: multi-omics data archives 3, heterogeneous software 4 and computing resources (Neuroscience Gateway5, Amazon Web Services (AWS)), and multi-site interdisciplinary expertise (e.g., engineering, biology, and psychology). Existing distributed high-performance computing resources (HPC) and other cyberinfrastructure (CI) tools for data management support the related data analysis and visualization capabilities. However, to fully utilize such capabilities, neuroscientists (often with limited CI skills) are required to take valuable time away from the focus of knowledge discovery in neuroscience, in order to learn about how to use the various technologies. To tackle with these research bottlenecks scientific gateways have been developed which collects raw data from research group and process them as per pre-defined workflows. Thus scientific workflows are an integral part of science gateways.

1.2.2 Bioinformatics Workflows Case Study

In recent years, next-generation sequencing (NGS) technology has improved dramatically, with costs dropping and the number and range of sequencing applications increasing exponentially. A wide variety of types of high-throughput sequencing can be generated from RNA or DNA molecules through NGS library construction and sequencing including techniques such as whole genome sequencing (WGS), RNA-Seq,

Table 1.1 Bioinformatics Workflow Applications.

Workflow Name	Workflow Description
FastQC	FastQC Quality Check workflow is used to conduct the quality control checks on raw sequencing data so that we can remove some low-score data before the next step of the analysis.
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.
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.
PGen	PGen workflow allow users to identify the single nucleotide polymorphisms (SNPs: a substitution of a single nucleotide that occurs at a specific position in the genome) and insertion-deletion (indels: an insertion or deletion of nucleotides from a sequence) and perform SNP annotation.
Copy Number Variation (CNV)	Copy number variation analysis helps detect the chromosomal copy number variation (CNV: is a phenomenon in which sections of the genome are repeated and the number of repeats in the genome varies between experimental groups/conditions) that may cause or may increase risks of various critical disorders.
Methylation	Methylation analysis helps estimated the methylation levels of each genomics cytosine and identified the differentially methylated regions between the experimental groups/ conditions.
Single Cell RNA-Seq	Single-cell RNA-Seq analysis allows users to align single-cell RNA-Seq read, perform clustering cells and then assign the cell type identity to clusters via biomarks.

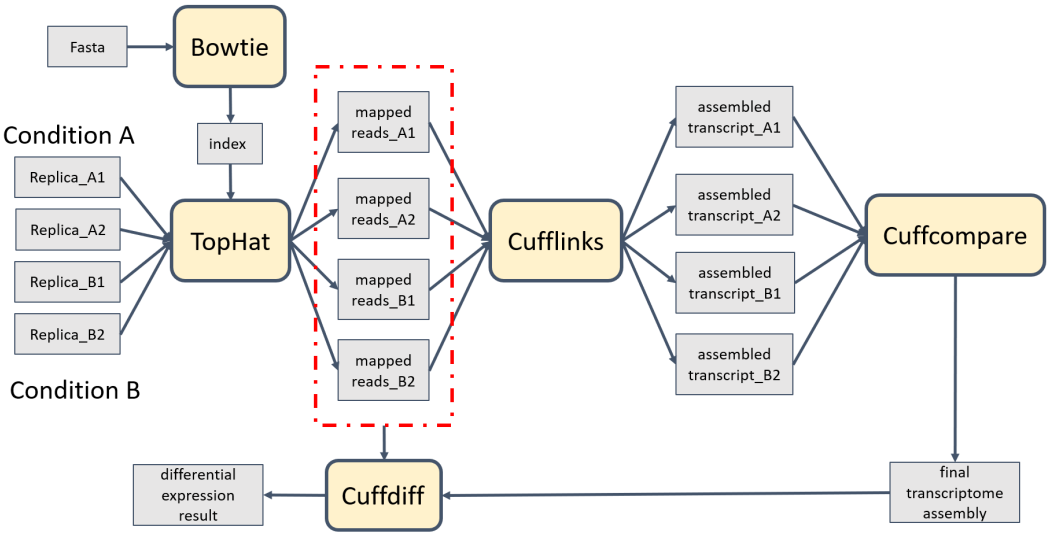


Fig. 1.1 Exemplar RNA-Seq analysis workflow allow us to identify the differential expressed genes by performing the pair-wise comparison of experimental groups/ conditions of sequencing data.

ChIP-Seq, DAP-Seq, RIP-Seq, methylation, and more. To efficiently utilize the large-scale NGS data for analysis, we have developed six bioinformatics workflows as shown in Table 3.6, and we have further centralized the input data for these workflows using CyVerse [19]. An exemplar workflow i.e. RNASeq is shown in Figure. 1.1.

1.2.3 Manufacturing Workflow Case Study

Researchers in the manufacturing industry who are working on areas concerning material discovery such Carbon Nano Tubes often need assistance when selecting parameters in the event of non-productive growth states. To ensure that synthesis processing conditions are found in a deliberate manner within time (i.e., Performance factor) or technology (i.e., Agility factor) or financial (i.e., Cost factor) (PAC) constraints, fundamental investigations to develop scientific approaches for progressive scale-up of the materials discovery is required. In this context, agility refers to the agent performing increased automation of the cyber-physical components i.e., high agility implies high levels of control automation. The research goal in this task addresses the current lack of systematic methods necessary to guide the user in terms of the 'demand' factors and the system's 'supply' operations to sustain a productive system state. It is entirely possible that a researcher will have little basis for deciding processing parameters in the event of non-productive system states. In fact, a review of the literature for CNT growth will find that most research groups find a synthesis procedure that "works" and then minimally vary these parameters, even for diverse applications. Such users can greatly benefit from access to a catalog of parameter set templates, predictive models as well as system resource configurations that were obtained manually. Some of the open research questions for (re)generation of a custom synthesis parameter template as shown in Figure. 1.2: (i) How can we set up interactions between the users and a catalog for composing an 'initial state' template that: (a) leverages the various synthesis system components, and (b) best matches requirements? (ii) How can we organize the catalog with related unstructured data sets or image data sets from successful synthesis configurations, related to user preferences, and benchmarks for future repeatable/reusable use as custom templates? (iii) What are the relevant heuristics to simplify and speed-up

candidate custom template solution prescriptions, and also build materials-community tailored catalog(s) of CNT materials behavior based on analysis of successful synthesis configurations?

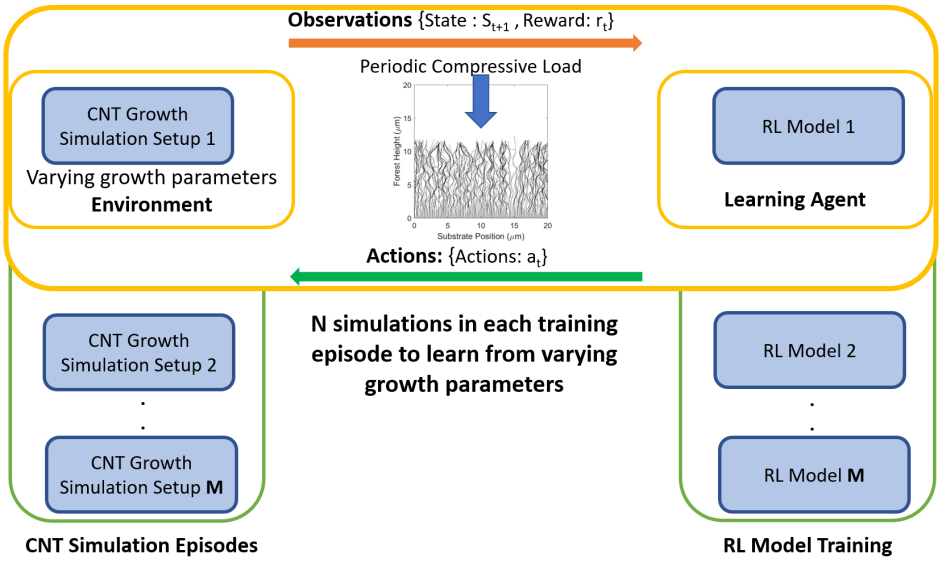


Fig. 1.2 RL agent training methodology followed in simulated growth of CNT tubes in episodes. At each iteration, the tubes are compressed and a reward is generated as feedback for the model to learn optimum actions. The model aims to increase the overall reward in the training.

Two major research thrusts for such manufacturing workflow is: (i) algorithms/-tools to create a catalog that can store and retrieve custom synthesis parameter templates with component combinations that match the CNT material goals and consider user preferences related to performance, agility and cost (PAC) factors, and (ii) Resource recommender algorithms that are suitable for a prescriptive catalog that requires low maintenance and is relevant to a broad range of materials discovery scenarios. To create the catalog, we first need to address cases where no suitable custom template solutions exist, and when a solution should be composed for future re-use. A 'cold start' issue exists in this context, whereby early catalogs have few solutions. Existing knowledge from the users and scientific literature will guide the early formation of parameter templates.

1.3 Cloud Computing Requirements

Scientific and technological applications are becoming increasingly data and compute-intensive. An exemplar case in scientific workflows can be seen in bioinformatics such as gene sequencing/analytics, where applications frequently require diverse multi-cloud resources to execute job flows. Researchers who create these workflow pipelines seek to use large computing and memory resources on a routine basis in an iterative and repeatable manner. Since the nature of research is often iterative and requires repeated workflow execution with varying data, they need to rely on open/commercial cloud resources for their workflows under budget limitations. To optimally fulfill user requirements, they seek to seamlessly interoperate with any compute resources they can access. A potential scenario could involve the utilization of small-scale resources in-house in conjunction with community cloud resources such as GENI [5] and commercial CSPs such as Amazon Web Services. However, the diversity in their sources offered from different commercial and community cloud providers and sparsely documented in-house compute resource configurations can overwhelm users who are not well versed in cyberinfrastructure or are not cloud experts.

1.3.1 Multi-Cloud Computing

Users (researchers and educators) use scientific workflows that require a diverse set of resources available across multiple cloud service providers (CSPs) such as Amazon Web Services (AWS) [1], GENI [2], Google Cloud Platform [4] and Microsoft Azure [5]. They create workflows that frequently require deployment and dynamic scaling of multi-cloud resources in an efficient and high-performance manner (e.g., quick setup, reduced computation time, and increased application throughput). At the same time, users seek to minimize the costs of configuring the related multi-cloud resources. While *performance* and *cost* are among the key factors to decide upon CSP resources selection, the scientific workflows often process proprietary/confidential data that introduces additional constraints of *security* postures e.g., compliance with HIPAA (Health Insurance Portability and Accountability Act) or FISMA (Federal Information Security

Management Act) guidelines. Further, *agility* or the ability of a CSP to auto-manage a diverse set of computing, networking, and storage resources is yet another important factor to deploy scientific workflows i.e., a higher level of automation capability offered by a CSP implies higher agility for the users.

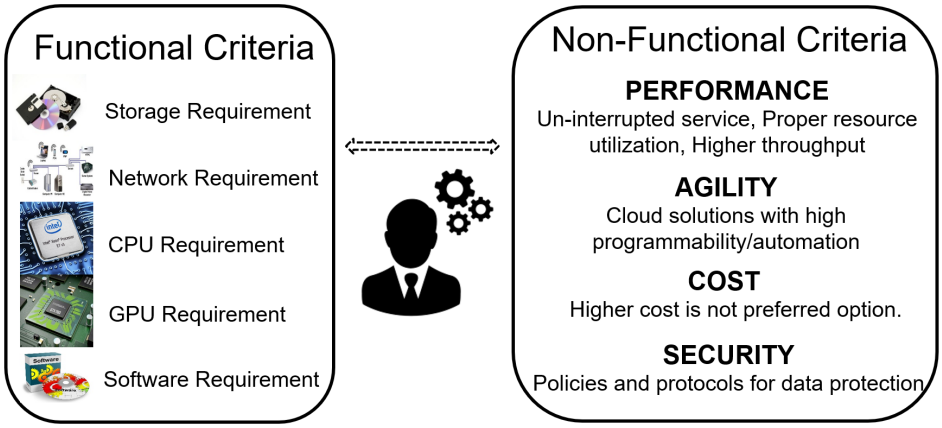


Fig. 1.3 Key functional and non-functional criteria for selection of cloud computing resources. Variations in non-functional criteria preferences from users' and providers' perspectives create the problem of excessive choice.

Thus, selection and configuration of multi-cloud resources for modern applications have become a necessity but it requires careful handling of objective factors such as *performance*, *agility*, *cost*, *security* (i.e., PACS factors) as shown in Figure 1.3. This multi-cloud resource brokering can be formulated as a multi-dimensional optimization problem in resource selection that is contextually dependent on user preferences and their application requirements. Moreover, the diversity in the resources and capabilities offered by different CSPs creates a situation of excessive/overwhelming choice for users. The choice issue is especially significant for users who are not cloud platform experts and require relevant guidance for multi-cloud resource selection. Often, a CSP provides a cloud-specific resource configuration and management service suite (e.g., AWS OpsWorks [6]) but the users often lack relevant tools to work with multi-cloud resources. Additional challenges arise due to the fact that the subjective experience of users can widely fluctuate depending on the network bandwidth, type of applications, capacity load on the servers, and location of the servers to name a few [7].

1.3.2 Multi-Edge Computing

Edge computing is computing that's done at or near the source of the data, instead of relying on the cloud at one of a dozen data centers to do all the work. It means the cloud is coming near the end users where all the computing will be performed. Multi-edge computing (MEC) is becoming a key technology toward "full 5G." However, as it gets widely used, a fundamental problem is how to support as many service requests as possible under stringent Quality-of-Service (QoS) requirements and limited communications and computing resources. Computation offloading and service migration are two major research hotspots in the multi-edge computing (MEC) environment. However, in the existing MEC architecture, as shown in Figure. 1.4, the idle computing resources of offsite edge servers as well as commercial cloud platforms are not fully utilized, which leads to the problem of high overall system time and energy costs.

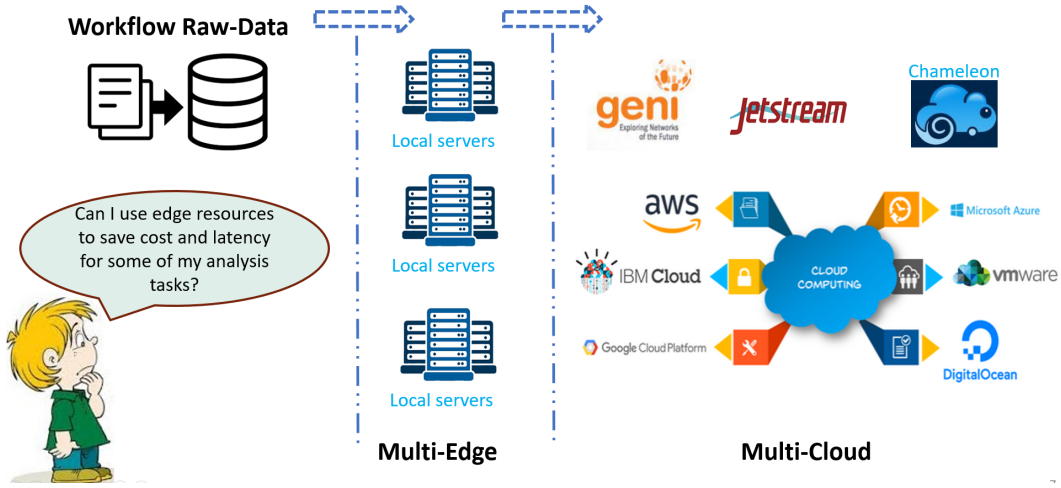


Fig. 1.4 Data processing requirement on multi-edge sites with heterogeneous resources in collaboration with multi-cloud resource platforms.

1.3.3 Trusted Computing on Edge-Cloud Resources

Edge computing promises to reshape the centralized nature of today's cloud-based applications by bringing computing resources, at least in part, closer to the user. Reasons include the increasing need for real-time (short-delay, reliably-connected) computing and resource-demanding artificial intelligence (AI) algorithms that overstrain mobile devices' batteries or compute power but are too bandwidth-demanding to be offloaded to a distant cloud. However, users may need to run their protected use case logic on

(untrusted) third-party edge devices, which can lead to serious issues due to weaker security measures than in cloud environments [8].

1.4 Dissertation Outline

1.4.1 Significance

Users who rely on multi-cloud resources have to deal with end-to-end resource optimization themselves. Additional challenges arise since the subjective experience of users can widely fluctuate depending on the network bandwidth, type of applications, capacity load on the servers, and location of the servers to name a few [7]. Many researchers have proposed approaches to address these challenges in [9] [10] [11]. However, these works either focus on the optimization of only a single CSP's resource(s) or focus on only the cost and/or performance requirements. Several prior works have addressed related problems in this field of study. The fundamental problem has a scope of optimization for users as well as providers since the users have their unique requirements and resources at different cloud providers are heterogeneous in nature and finite. For optimizations in the context of the users, approaches proposed in [10] [12] are based on using cloud platform knowledge bases in recommending solutions. Specifically, the solutions proposed assume that a knowledge base of resources is already available and resources are recommended from the knowledge base to optimally fulfill user requirements. Although optimization of different criteria such as performance, agility, cost, and security are crucial individually, joint optimization of these criteria simultaneously, adds further complexity and variables for optimization. There are multiple factors such as VM configurations, application workload, network bandwidth, and computation capabilities that govern the selection of resources, and these factors can vary in real time within an application deployment. This necessitates leveraging learning-based solutions for improving real-time resource selection. There have also been prior works that have validated the inconsistencies in Quality of Service (QoS) from resources [13] [14]. With all its benefits, however, cloud computing costs can become a barrier to 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) [74] are emerging to leverage volunteer contributions of large-scale cloud resources to reduce costs. 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 [62], 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 lightweight inference on edge nodes. Ensuring trust in VEC allows fulfillment of our goal of utilizing all available cyber resources for any application workflow while maintaining trust based on Performance, Agility, Cost, and Security (PACS) which is the significant outcome of this thesis.

1.4.2 Approach

In order to achieve our goal of democratized use of any available computing resource in a PACS efficient manner as shown in Figure. 1.5, several smaller but crucial sub-problems have to be solved i.e., optimized orchestration of diverse ever-growing resources on edge and cloud platforms needs to tackle several challenges which are

- **Problem 1:** Understanding user's applications requirements
- **Problem 2:** Understanding and utilizing user's expertise in orchestrating cyber resources
- **Problem 3:** ML driven Performance, Agility, Cost and Security (PACS) aware resource allocations for optimized orchestration of resources from multiple cloud platforms and edge resources

- **Problem 4:** AI-driven trusted resource allocation for non-benchmarked voluntary edge resources and cloud platforms

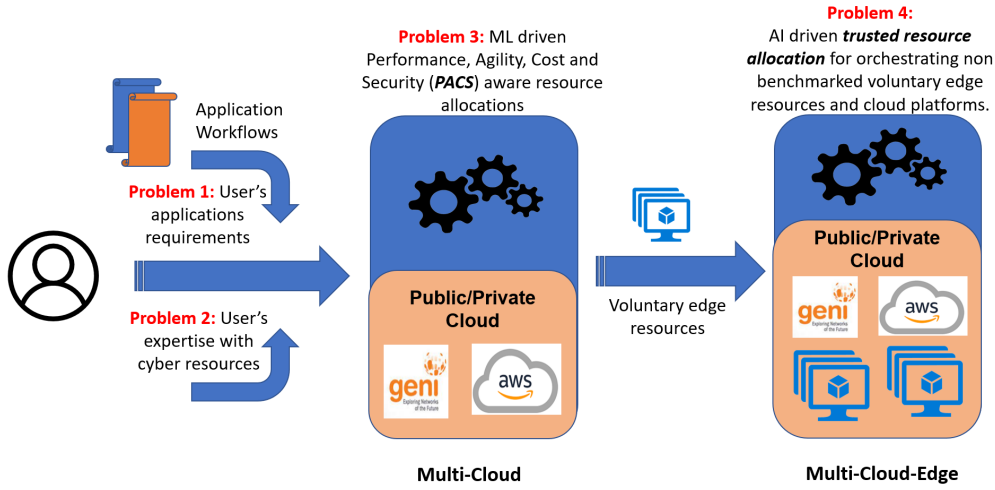


Fig. 1.5 Workflow of key research problems to be solved towards the goal of utilizing all available cyber resources for any user-specific application workflow while maintaining trust based on Performance, Agility, Cost, and Security (PACS).

1.4.2.1 User Engagement

For the ease of non-expert cloud users, who often struggle to deploy cyberinfrastructure in an efficient manner for data analytics and to gain from the experiences of cyber expert users, we divide our proposed solutions into four subsections of i) Collection which comprises a) Knowledge interface system (KIS) and Fuzzy Interface System (FIS), ii) Composition iii) Consumption iv) Fuzzy Engineering. Detailed discussions on the proposed solutions are discussed in Chapter 2.

1.4.2.2 PACS optimization in Multi-Cloud

User specification collection is succeeded by composition wherein the requirements are passed to an optimizer engine (OnTimeURB) to compose template solutions. Custom template composition essentially can be formulated as a selection of machine configurations from a distributed set of diverse cloud instances under specific constraints, which is an NP-hard problem. To effectively formulate the problem into a solvable model, we use CPLEX [17] optimizer to create a relevant optimization model. The objective of the model is defined in a manner to reduce the cost of the template solution for user

requirements. The optimization model is constrained by user specification of required resources and resource thresholds. For example, a user can specify a requirement as {*cpu:8, ram:16Gb, storage: 60Gb*} with a threshold of 25% which acts as a constraint in the model. The below classifications identify potential template responses from the OnTimeURB middleware

High Performance and Cost: All user-defined resources are amplified in step sizes up to a user-defined threshold limit. Each step gives an amplified resource constraint which results in a corresponding solution. Prospective provisioning: minimal configuration {*cpu:8, ram:16Gb, storage: 60Gb*}, maximum configuration {*cpu:**10**, ram:**20Gb**, storage: **75Gb***} etc.

Low Performance and Cost: The template is the closest match to user resource specification with minimal overprovisioning. Prospective provisioning: {*cpu:8, ram:16Gb, storage: 60Gb*} OR {{*cpu:4, ram:8Gb, storage: 30Gb*} *and* {*cpu:4, ram:8Gb, storage: 30Gb*}} etc.

Further, the optimizer engine is enhanced with capabilities to recommend cloud template solutions which enhances the agility requirements of the users. OnTimeURB is also enhanced with a knowledge engineering method that uses an ML model to aid the optimizer in OnTimeURB to improve the selection of CSPs for better performance e.g., the reduced execution time of application workflows while maintaining optimal cost, security, and agility requirements are implemented. The ML model utilizes a Naive Bayes classifier to recommend optimal cloud template solutions by weighting performance, agility, cost, and security (PACS) factors. Validation of OnTimeURB benefits using a catalog of bioinformatics application workflows integrated within the KBCommons [18] science gateway. Detailed discussions on the proposed solutions are discussed in Chapter 3.

1.4.2.3 PACS based Trust in voluntary edge cloud

A VEC system is comprised of multiple geographically distributed clusters, with each cluster having a set of co-located voluntary diverse edge resources. However, a major challenge for wider adoption of the VEC computing paradigm in scientific applica-

tion 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. The proposed solution to tackle trust in VEC builds on prior probability-based trust models and presents a novel probability-based trust modeling scheme for resource allocation in a VEC system for scientific application workflow management. Specifically, the approach uses a Dirichlet-based probability distribution which can model multiple variables simultaneously. A novel 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. In order to improve the PACS satisfaction of users on the volunteered cloud, the VEC system is further improvised and integrated with reinforcement learning algorithms which are oriented towards characterizing volunteered resources for their performance and reliability. Detailed discussions on the proposed solutions are discussed in Chapter 3.

1.4.2.4 VEC Resource Brokering Guided Applications

VEC architecture of computing can be integrated with various workflows and applications which can leverage edge resources to offload computing requirements. In this thesis two case studies are done wherein VEC can potentially improve the performance of the application. These studies are

- Carbon Nano Tube (CNT) growth automation through reinforcement learning
- Data Analytics pipeline for real-time image analytics

1.4.3 Conclusion

In this thesis, several of the smaller challenges have been tackled towards the final goal of utilizing any available computing resources in a PACS efficient manner. Towards this goal firstly, a novel middleware (i.e., OnTimeFLC) based on fuzzy engineering is proposed to utilize the expertise of users for better cloud resource selection. The method is composed of a multi-level fuzzy model based on factors of performance, agility, cost, and security (PACS) which aids a resource broker based on integer linear programming

(i.e., OnTimeURB) in composing multi-cloud solution templates. We validated that the proposed fuzzy engineering approach by simulating decision-making and utilization of expertise from users in improving the selection of appropriate CSPs. We also showed how OnTimeFLC and OnTimeURB can help with resource management in a science gateway deployment viz., KBCommons to help bioinformatics researchers/educators. Secondly, a knowledge-engineered approach is proposed which used machine learning models to understand workflow and application requirements to guide the optimizer engine of OnTimeURB for improved performance. The applied ML model is designed to learn the bias of expert users towards cloud platforms for different requirements of functional criteria and workflow sizes. Finally, a new architecture of computing i.e., VEC is proposed which aims to utilize voluntary edge resources to reduce cost and improve the performance of applications by reducing latency for data transfers. Further, the challenges of trust based on PACS within VEC computing are addressed through probabilistic models. Artificial intelligence methods such as reinforcement learning models are developed to learn the behavior of volunteer edge clouds which is used to fulfill PACS requirements of user application workflows.

CHAPTER 2

User Engagement with Cloud Platforms

2.1 Overview

To mitigate the problem of ill-advised resource allocation certain key factors can be identified that strongly govern the selection of optimal cloud resources for maximum resource usage and user satisfaction. These factors are performance, agility, cost, and security offered by the CSPs [20]. Since selection and configuration of multi-cloud resources for modern applications requires handling objective factors such as *performance*, *agility*, *cost*, and *security* (i.e., PACS factors), the multi-cloud resource brokering involves a multi-dimensional optimization problem in resource selection. Apart from these functional and objective factors, the users often gain expertise towards certain CSPs creating inherent biases in CSP selection depending on the functional requirements of PACS, business, or geographical constraints to name a few. Moreover, PACS factors are subjective factors that can have varying metrics for evaluation for different cloud service users (CSUs). For example, performance can be evaluated based on sub-factors of availability, reliability, response time, and throughput. The measurement of these sub-factors influences the evaluation of PACS factors and ultimately the CSP selection.

CSPs often provide cloud-specific resource configuration and management service suite (e.g., AWS OpsWorks [6]) which gives insight into trade-offs among its services in the context of PACS criteria. But since these tools focus on a single cloud platform, it lacks the vision of services from other CSPs and thus fails to give insights and trade-off with services from other platforms. Moreover, the subjective factors arising from users' bias toward certain cloud platforms or perceived performance of the cloud resource by

users also need to be considered in the multi-cloud resource brokering which are not addressed with these tools.

2.2 User Engagement

For the ease of non-expert cloud users, who often struggle to deploy cyberinfrastructure in an efficient manner for data analytics and to gain from the experiences of cyber expert users, we divide our OnTimeURB into four subsections of i) Collection a) knowledge interface system (KIS) and Fuzzy Interface System (FIS), ii) Composition iii) Consumption iv) Fuzzy Engineering.

2.2.1 Graphical user interface for resource selection

2.2.1.1 Knowledge Interface System

We leverage KBCommons [18] web portal to collect user resource requirements via a KIS module. KBCommons portal provides a comprehensive web resource for storing, sharing, analyzing, exploring, and visualizing multi-organism genomics and multi-omics data. It provides information for several entities including genes/proteins, microRNAs/sRNAs, metabolites, SNP, traits, etc. A suite of interactive web-based tools for bioinformatics analysis and data visualization is available via this dedicated web resource. It also has a suite of tools for differential expression analysis of transcriptomics data, genomics variation single nucleotide polymorphism (SNP) and insertion-deletion (Indel) data, as well as other multi-omics datasets providing access to Venn diagrams, volcano plots, function enrichment, and gene modules for easy user interpretation.

We implemented and integrated our KIS module as shown in Figure. 2.1 with bioinformatics analytics workflows user interface (UI) deployed within the KBCommons portal to collect user-specific resource constraints and criteria to execute user workflows. The KIS presents users with a set of questions (e.g.,vCPU/RAM required) to capture their requirements. Additionally, users can provide a threshold maximum on the requested resources, for example, a 10% threshold on 10Gb RAM memory requirement will allow the OnTimeURB optimizer to compose solutions with RAM memory up to 11Gb.

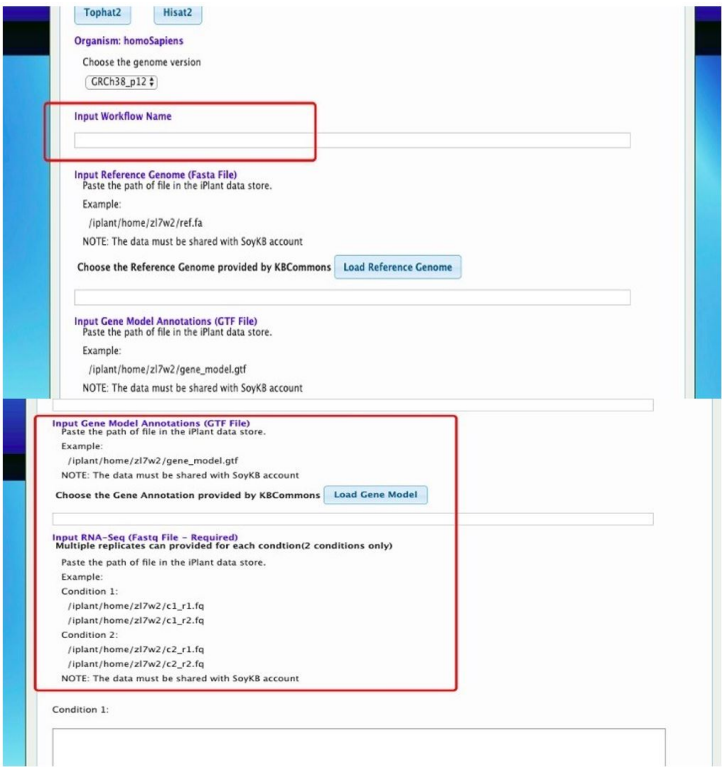


Fig. 2.1 Screenshot of the webpage of creating workflow request in Prototype 1 of KB Commons (Improved System) which shows layout issues and too many instructions for users to read

2.2.1.2 Fuzzy Interface System

The fuzzy interface system is designed to collect and understand users' experience with using CSPs based on PACS factors. The user is provided with an interface to rate their experiences towards different CSPs on a specified scale. The users are also given the option to create rules to quantify their experiences. The format of rule creation is governed by If, Then, PACS, and scaling tags as shown in Table. 2.5.

Below is a sample rule.

```
[{IF Performance is Good AND IF Agility is Good
AND Security is Good THEN CSP is AWS},
{IF Performance is Bad AND IF Agility is Bad
AND Security is Good THEN CSP is MU},
{IF Performance is Good AND IF Agility is Bad
AND Security is Good THEN CSP is GENI}]
```

Listing 2.1: Sample fuzzy rule format

2.2.2 Significance and Related Works

While the challenge of scalability and response time for visualizing big data persists [21], there are also usability issues with data analytic systems [23]. Data visualization is created or delivered by machine learning models [24]. Usability studies need to address whether the critical user population understands the data visualization as well as the complexity of underlying analytic process i.e., how the visualized outcome was created with its underlying calculations [23]. With prescriptive systems, the issue becomes further complicated. Advancement of tools and applications for big data analytics have focused more on providing scalable techniques and less on facilitating easy access of these applications for users [22]. For prescriptive systems, efficient decision-making involves users choosing right parameters for computing their data, which if not performed in a well-informed manner may lead to misinterpretation of results or failed experiments [22].

2.2.3 Approach

We applied multiple research methods. For iteration 1 conducted in the summer 2019, we applied methods to understand the work processes of users with the Baseline System and potential challenges in completing the tasks. We adopted the Sociotechnical Walkthrough (STWT) methodology [25] that is useful for an integrated view of the multiple perspectives of human work procedures related to the technical system. We developed a semi-structured interview protocol [26] using open-ended questions and prompts for elicitation of details organized in four parts: a) current user interaction and task flow, b) user experience, c) participant opinion and reflection, and d) additional comments. We also observed the interaction of the participants with the Baseline System. Each session lasted for 40 to 60 minutes and was conducted by a lead interviewer in presence of an observer.

For iterations 2, 3, and 4, we conducted usability studies with the three prototypes of the Improved System [27]. Methods included a semi-structured interview protocol, a usability task-based performance test [27] with Think-Aloud [28], Single Ease Questionnaire (SEQ) [29], participant observation (recorded data and observer notes), and

the System Usability Scale (SUS) survey [30]. The interviews gathered demographics, experience with genomic data analysis and prompts such as overall experience with the system, user thoughts about the specific functions, and likes/dislikes. Users chose one of the predicted recommendations for cloud solutions, and we asked them to justify their choice. For the usability test, the participants were asked to think aloud [28], while independently completing the tasks with the prototypes. The tasks included: logging into the system and opening the data request form, creating a data workflow request, and reviewing the status of the submitted request. Sessions were administered using Morae software [31], recorded and transcribed via Zoom. Each session was conducted in the presence of at least two researchers and lasted for 60 to 75 minutes. With multiple methods and multiple data sets, a robust set of triangulated data was used to identify maximum usability problems with certainty.

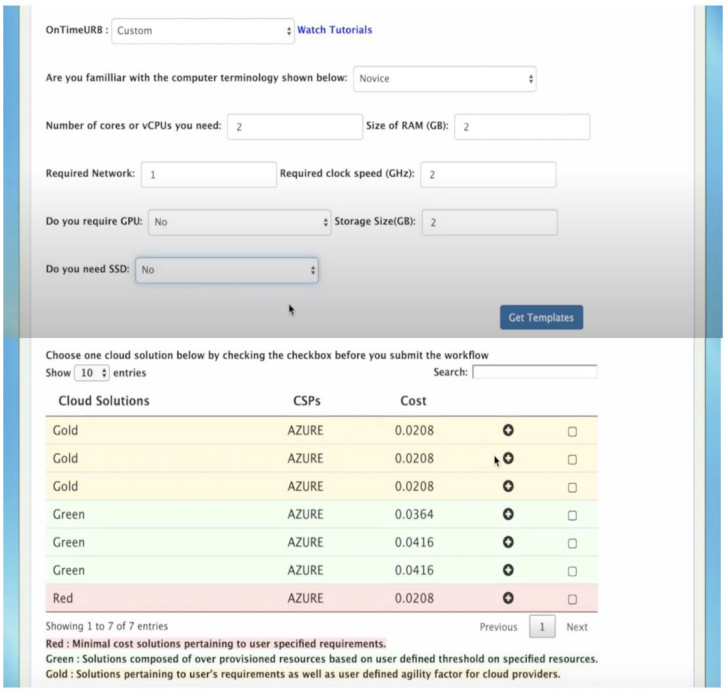


Fig. 2.2 Screenshot of optimum cloud solution provider in Prototype 2. Users enter their preferences in the form and get Gold/Green/Red Templates of cloud solutions. Users are expected to choose a predicted template based on the given cost and available cloud service providers.

We measured usability in terms of its effectiveness i.e., users being able to complete the task with or without difficulty, and efficiency i.e., time taken by the users to complete the task [32]. Descriptive statistics were used to calculate mean and range values.

We analyzed quantitative data for each prototype separately as there were significant changes in functions and the user interface (UI). We performed a two-tailed *t-test* to compare the SUS scores of the three prototypes.

Qualitative data collected in iterations 2, 3, and 4 included observation data and participants' responses to open-ended questions. We used the van den Haak et al. [28] method to thematically analyze qualitative data and identify layout, terminology, data entry, and comprehensive feedback problems. The final interpretation of qualitative results was reviewed by the research method expert to ensure interrater reliability. For iteration 1, interview data for the Baseline System was analyzed through a semi-formal modeling notation, known as SeeMe [25]. SeeMe was used to graphically represent the communication and coordination between the multiple stakeholders and the technical system by visualizing human work procedures, related technology functionalities, and social structures of their interaction [25]. Interview transcripts were analyzed by two researchers, including a research method expert. Final results were reviewed through Intermittent meetings between the researchers until a consensus was reached.

We used a multiple-method approach in each iteration to address the validity and reliability of the results. Qualitative data from the think-aloud protocol, participant observation, and interviews were supported with quantitative data collected from surveys, questionnaires, and task performances.

2.2.4 Evaluation

User satisfaction: With an average SUS score of 73.86, Prototype 3 was rated 'acceptable' [30]. In all, 64 percent (7/11) rated Prototype 3 as 'acceptable' (SUS range: 97.5 to 75.0), 27 percent (3/11) as 'marginally acceptable' (SUS range: 67.5 to 60), and 9 percent (1/11) as 'not acceptable' (SUS score: 40). The two-tailed *t-test* for SUS scores obtained for the prototypes compared Prototype 1 and Prototype 3, and SUS significantly improved from 58.2 to 73.86 ($p < .05$). Participants appreciated the usefulness of Prototype 3 for novice and expert biological researchers. According to the participants, the instructions were easy to follow, and they were able to understand the data analysis process as well as the working of the application. Participants found the

Gold/Silver/Bronze color categorization of cloud solution templates to be more intuitive and spent time reading the instructions to make an informed decision.

Table 2.1 User needs and choices they have made

Participant responses in Iteration 4: Prototype 3 of KBC			
Sr. No.	***Preferences for big data analysis)***	***Option selected by user***	***User rationale***
P1	Shorter execution time; Low cost or free	Gold	*"I choose Gold for relatively lesser time (15 days) and high computing power"*
P2	Ease of use; Shorter execution time; Free of cost	Silver	*"I choose Silver because it matches my choice of estimated time. Given storage is same for all so it does not make a difference."*
P3	Shorter execution time; Low cost	Bronze	*"Because my input data is not very big and I don't want to spend much money on this, I will choose Bronze because the pipeline I am working on, I can wait for a day. But if I do not have to pay then I will choose Gold because of the shortest amount of time it takes."*
P4	System efficiency; Shorter execution time; Free of cost	Gold	*"Because I am not an expert, I will choose Gold - I don't have to put the time into it. I am a beginner and it is fully automatic. It is quick and easy. If I have some experience, then I would want to cut down costs and choose Silver. Or Bronze if I am an expert."*

The central feature of Prototype 3 as shown in Figure. 2.3 was the prescriptive system recommending the optimum set of cloud solutions to the users based on their preference of time, cost, and agility. Users chose a Gold/Silver/Bronze solution to compute their data analysis. While Prototype 2 shown in Figure 2.2 still had usability issues with the predicted recommendations, Prototype 3 indicates the users' preferences and the decision they made were aligned, meaning they picked the decision that was best suited for their needs. This shows the improved accuracy of the newly developed prescriptive system. Table 2.1 lists the data sample of users' preferences and the option they picked based on the prescriptive system.

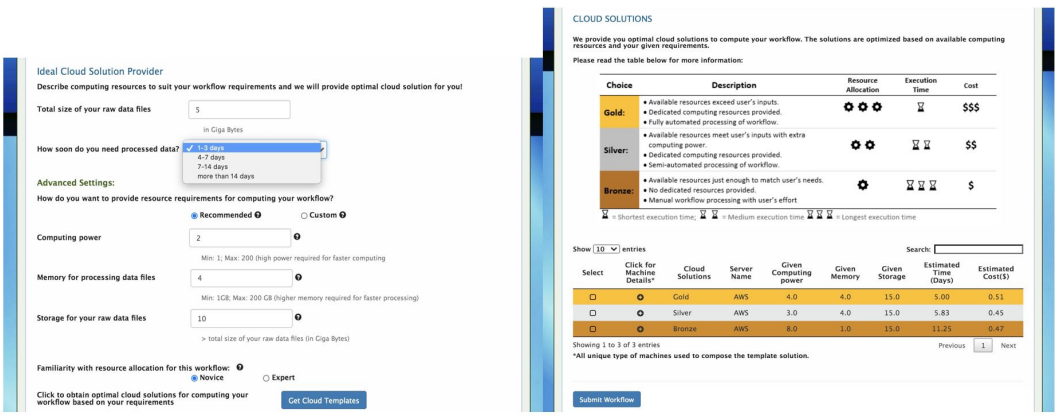


Fig. 2.3 Screenshot of optimum cloud solution provider in Prototype 3 named as 'Ideal Cloud Solution Provider.' Users enter the size of their raw data files and how soon they need the results. New users have the option to select 'Recommended' settings. Users are expected to choose one of the predicted cloud solution templates categorized as Gold/Silver/Bronze based on time, cost, and agility.

2.3 User Preferences

Resource configuration and management service suites such as AWS OpsWorks [6] give insight into trade-offs among its services in the context of PACS criteria, but these tools generally focus on brokering a single cloud platform pool of resources. Moreover, the tools do not consider subjective factors arising from users' bias toward certain cloud platforms or perceived performance of the cloud resource by users, which also needs to be considered in multi-cloud resource brokering.

2.3.1 Fuzzy Engineering for User Preferences

The subjective experience of users towards cloud services can fluctuate depending on the quality of service (QoS), type of applications, capacity load on the servers, and location of the servers to name a few, and thus can not be quantified. As shown in Figure. 2.4 to gain meaningful insights into the bias towards cloud providers which is affected by PACS criteria, fuzzy logic can potentially be promising [33]. Fuzzy logic theory gives tools and methodologies to study uncertainty in a system or a situation and provides flexibility in reasoning. The idea behind fuzzy logic is to imitate human behavior and logical reasoning for deducing conclusions for vague problems in a non-linearly weighted manner [34].

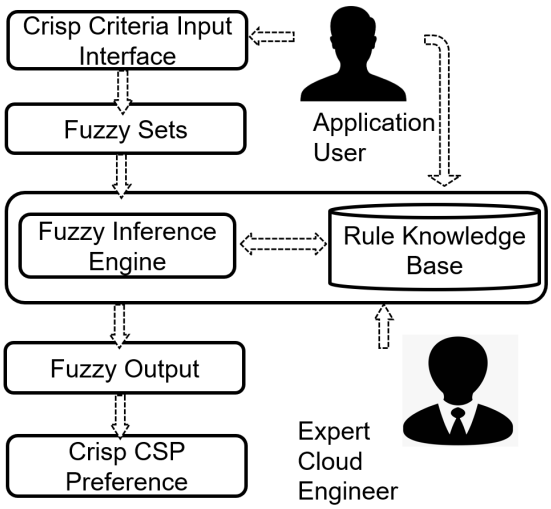


Fig. 2.4 OnTimeFLC's core optimizer engine uses an integer linear programming based optimizer and fuzzy engineering augmented with a knowledge base from different cloud providers: (i) Amazon Web Services (AWS) [1] - public cloud, (ii) GENI - community cloud [2], (iii) MU Data Center - private cloud [3].

2.3.2 Significance and Related Works

Fuzzy logic in user cloud selection: There have been comprehensive studies on the behavior and utilities of fuzzy models in simulating decision making [35] [36] which can be simulated for cloud service selection decision. Specifically, fuzzy logic can be used to complement multi-cloud resource brokering methods that take into account quantitative user resource specifications [20]. These methods can leverage fuzzy logic modeling to consider the user's expertise and biases in cloud selection using the integer linear optimization approaches. Since fuzzy logic can be used in decision-making, it has also been used by researchers in modeling optimal scheduling of resources on cloud infrastructure on data centers.

Fuzzy logic has also been used in the domain of quantifying resource selection. Exemplar efforts in characterizing cloud resources can be seen in [33]. The authors emphasize that there is a need of measuring and evaluating cloud performance to help users in making their decisions. The researcher primarily aimed to develop a model to evaluate the performance of the cloud based on factors such as workload, storage, hypervisor, and network devices. Their proposed model clarifies how the infrastructure and applications on a cloud platform can play an important role in application performance delivery. However, the study is focused only on performance benchmarking for a single cloud platform.

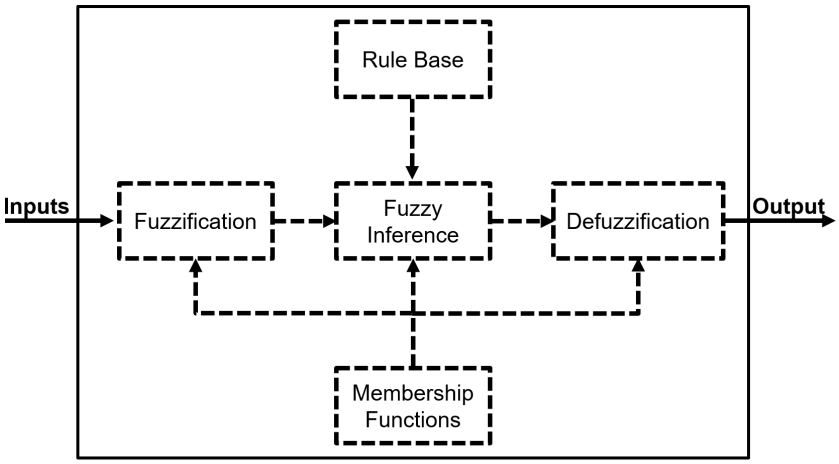


Fig. 2.5 A standard fuzzy inference system comprising of: (a) Input/Output variables, (b) Rule Base, (c) Inference Engine and, (d) Membership functions for variables.

Fuzzy logic in Cloud platforms: Many researchers have attempted in using fuzzy logic for scheduling resources [37] [38] [39]. Amin et. al. [37] have presented and

evaluated a new scheduling algorithm that is an efficient technique for scheduling virtual machines between data centers using fuzzy logic. A similar work [38] focuses on the allocation and scheduling of resources as well as improving the reliability of cloud computing. The typical fuzzy model as shown in Figure. 2.5 is customized to be based on the governing rules which consider factors of cost, trust, length of processes, and priority. Further, the authors showed that their output resulted in lower cost and increased reliability of cloud resources compared to the scheduling techniques such as FIFO (first in first out) and Min-Max (min requirement task first). Toosi et. al. [40] extended fuzzy logic into a load-balancing algorithm. Their approach helps cloud providers with multiple geo-distributed data centers in a region by evaluating the temporal variations in on-site power and grid power price. It then optimizes by routing the demand to a suitable data center in order to reduce cost and improve energy utilization. A similar work [39] proposes a multi-objective best-fit-decreasing (BFD) solution to the virtual machine reallocation problem. The authors consider a multi-objective formulation accounting for power costs and resource utilization. Although the methodologies followed in the above works give better insights into the research domain and utility, the user's experience with cloud service providers has not been considered in the decision-making progress of resource selection as we do in our work in this paper.

A recent work by Rizvi et al. [41] sought to evaluate the security of CSPs through a fuzzy inference system. The authors used several sub-factors modeled with a fuzzy inference model for measuring the security readiness of cloud providers from a cloud service users' (CSUs) perspective. Our work is inspired by this prior effort and we further characterize cloud platforms based on performance, agility, cost, and security (PACS) factors from a user's perspective of non-functional requirements. Our proposed model is unique compared to other related works because we consider the input from CSUs and their convoluted definition of PACS to then synthesize the information into a quantitative form, and finally evaluate cloud service provider selection. We further use the results to create optimal cloud templates customized to user preferences through a novel OnTimeFLC optimizer augmented with a fuzzy logic model.

2.3.3 Approach

To ease the process of multi-cloud resource brokering for non-expert cloud users we have proposed OnTimeFLC which is further divided into four steps: (i) Collection, (ii) Composition, (iii) Consumption, and (iv) Fuzzy Engineering.

To guide non-expert users with resource allocation for application workflows, we deploy a fuzzy engineering model and utilize user's expertise along with a knowledge base of benchmark rules for assessing PACS criteria of CSPs. Fuzzy logic is used to model uncertainty in unquantifiable variables of a system which are the true PACS offered from different CSP resources. The architecture of our implementation is shown in Figure. 2.6.

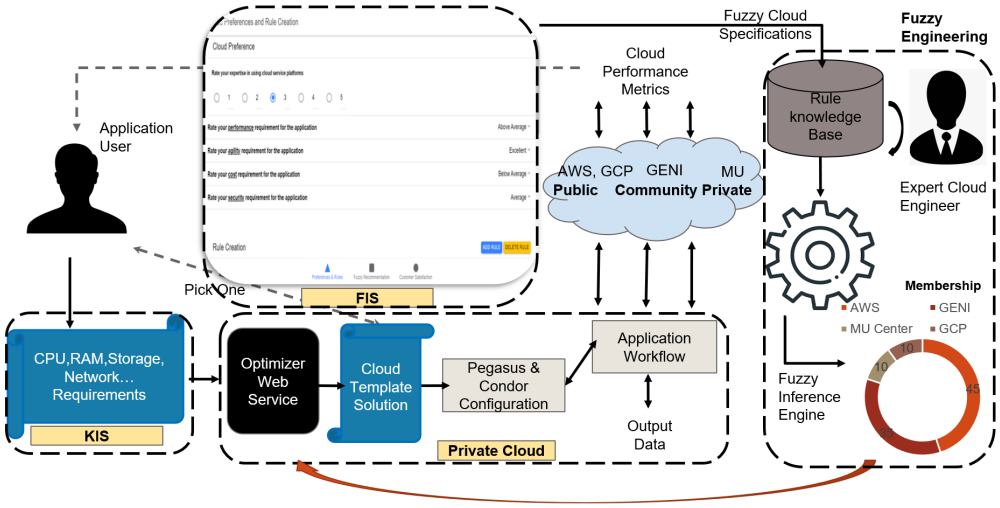


Fig. 2.6 Brokering lifecycle comprising of a) Collection - Knowledge Interface System (KIS) and Fuzzy Interface System (FIS) deployed in KBCommons portal [18] to collect user specification; b) Composition - Templates are composed and classified by optimizer into Red, Green, and Gold templates which are presented in the KIS for selection; c) Consumption: Pegasus, HTCondor, and workflows are configured in a GENI [2] node machine, and HTCondor as per cloud template schedules tasks in resources of Amazon web services (AWS), GENI or MU and d) Fuzzy Engineering - Rules and ratings on PACS criteria are collected and inferred, the results are passed to optimizer.

2.3.3.1 Fuzzification:

This is a process of converting "crisp values" (i.e., a user-specified input or approval on a pre-defined scale) into linguistic fuzzy values. The crisp data provided by the CSUs are mapped to a fuzzy set that contains the membership functions and linguistic values corresponding to the input as shown in Table 2.2. A fuzzy set is defined by the members

it contains as shown in x ϵ X where x is the element. In particular, a fuzzy set is defined by ordered pairs as shown below:

$$A = \{(x, A(x)) \mid x \epsilon U\} \tag{2.1}$$

where U is the universe of discourse which contains all of the elements that may be used in fuzzy set A. The membership functions are read as A(x) wherein a membership value ranges between interval [0,1] to each element in U. In our proposed approach, the linguistic values represent the fuzzy sets $fs\ (i = b, p, a, g, e)$ which consists of bad, poor, average, great, and exceptional. Each of these variables can be interpreted differently by CSUs. Therefore, to standardize the fuzzification the membership functions are used within in the fuzzy sets to define the variation in interpretation. In our approach, CSUs can utilize linguistic values as shown in Table 2.2 in order to make a decision as to which CSP would better fit their needs.

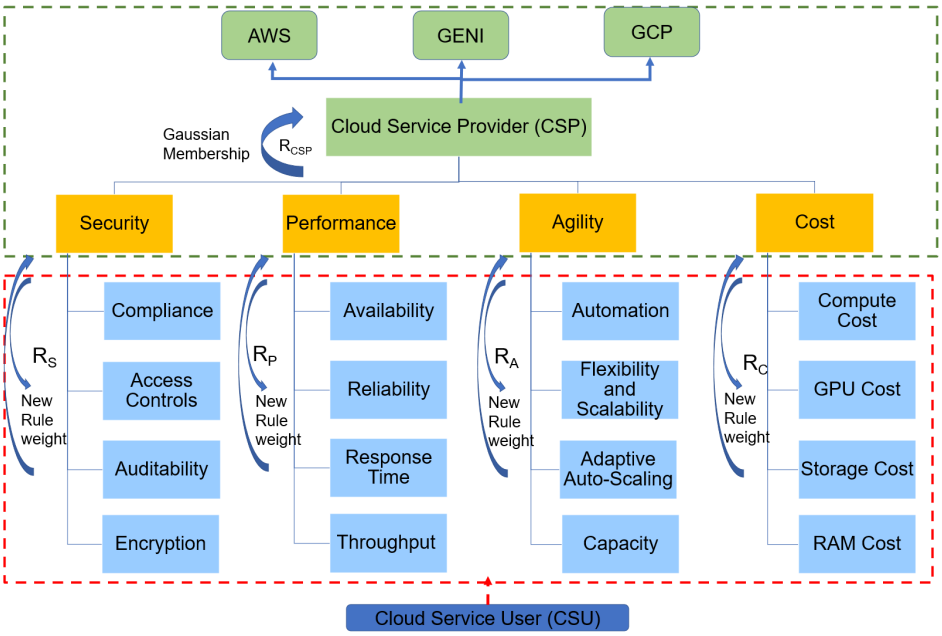


Fig. 2.7 Multi-Level fuzzy inference model with: (i) Red-box representing base fuzzy inference model for measuring PACS from sub-factors, and (ii) Green box representing fuzzy inference model for evaluating CSPs from PACS factors.

There are multiple factors that affect individual criteria of PACS evaluation of a CSP. We identify four sub-factors for each of the PACS criteria used in our multi-stage fuzzy-logic approach as shown in Figure 2.7.

Table 2.2 Fuzzy model linguistic terms descriptions.

Linguistic Terms	**Membership Degree**	**Membership Description**
Exceptional (e)	80-100	Near Flawless (Generally Cheap)
Great (g)	60-80	Better than most (Often Cheap)
Average (a)	40-60	Not Sure (Infrequently Cheap)
Poor (p)	20-40	Sometime fails (Frequently Costly)
Bad (b)	0-20	Frequent failures (Generally Costly)

2.3.3.2 Membership Function:

Membership function grades the association of a value to a set. Different criteria or variables can follow different type of membership depending on their actual distribution of effectiveness in a metric scale.

$$\mu_{A^i}(x) = e^{-(x-\mu)^2/2\sigma^2} \tag{2.2}$$

where μ and σ are center and width of the i^{th} fuzzy set A^{i}

For our problem domain, we assumed that each of the PACS criteria and sub-factors will have a gaussian membership function. We also control the gaussian curve parameters depending on CSUs feedback to create a custom membership function.

Table 2.3 Centers and Sigma values of membership functions for sub-factors of PACS.

Criteria	μ	σ
Exceptional(e)	83.3	5
Great(g)	66.64	5
Average(a)	49.98	5
Poor(p)	33.32	5
Bad(b)	16.66	5

2.3.3.3 Inference Engines:

The inference engine is the core of fuzzy logic where the inference rules are applied to the fuzzy input in order to generate the fuzzy output. Essentially, the inference rules are used to evaluate the linguistic values generated from crisp fuzzy input and map them to a fuzzy set. These fuzzy sets are then transformed into resulting output crisp values using a defuzzification process. Inference engines are superficially classified into two types: *(i) Mamdani inference system:* This inference system is intuitive and well-suited

to human input and is based on an interpretable rule base. Due to its application in simulating human-like thoughts based on constraints, we have utilized it to learn about user biases towards different CSPs. Each of these inference rules is composed of if-then statements wrapped around linguistic terms (Table 2.4). The if-then rules contain the antecedents (i.e., linguistic input terms) and the consequence(i.e., linguistic output). When fabricating an inference rule, operators such as "and," "or," and sometimes "not" are used [41]. For our proposed model, we have used primarily the "and" operator which is defined as below.

$$\mu A \cap B(x) = min[\mu A(x), \mu B(x)] \tag{2.3}$$

This rule extracts the minimum number of the membership values of the fuzzy sets to compute the "and" operation

(ii) Sugeno inference system: This inference uses singleton output membership functions that are either constant or a linear function of the input values. The defuzzification process for a Sugeno system is more computationally efficient compared to that of a Mamdani system since it uses a weighted average or weighted sum of a few data points rather than computing a centroid or center of gravity of a two-dimensional area. We used the Mamdani inference system since we primarily aim to assess CSU's cloud platform preferences by translating and validating their intuition and human reasoning with the knowledge of experts as discussed in [34][42].

2.3.3.4 Defuzzification

During defuzzification, the fuzzy output from the inference engine is mapped to a crisp value that provides the most accurate representation of the fuzzy set [34]. The fuzzy outputs are represented as F_{o1} $(o1 = b, p, a, g, e)$ for the first level and F_{o2} $(o2 = AWS, GENI, MU)$ for the second level and, they share the same gaussian membership function. The system involves five types of defuzzification methods for interpretations of rules namely: a) centroid, b) center of gravity, c) bisector of the area, d) largest of maximum, and e) smallest of maximum. We use the centroid method to obtain the crisp outputs from our fuzzy inference engine [43] which is represented in

the below equation -

$$Crisp\ Fuzzy\ Output = \frac{\sum_{p=1}^{P} z_p \cdot u_c(z_p)}{\sum_{p=1}^{P} u_c(z_p)} \tag{2.4}$$

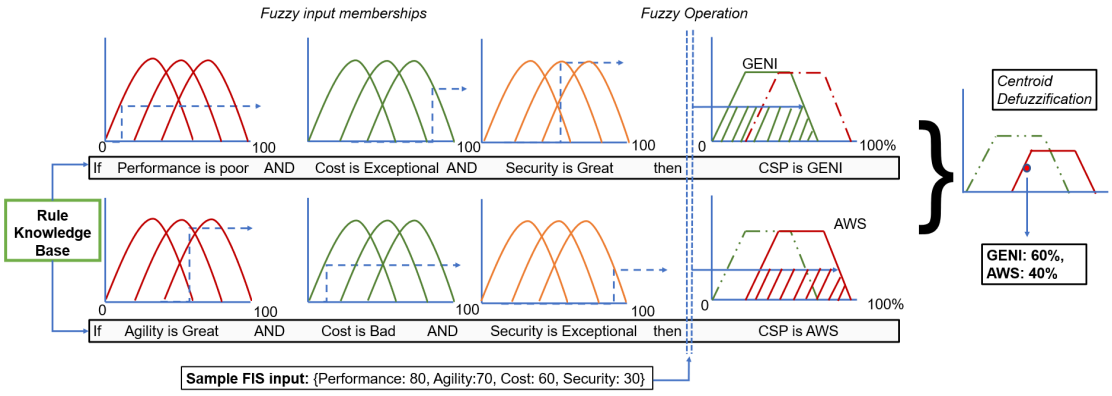


Fig. 2.8 Sample process showing multiple rule interpretations using mamdani inference system; The input PACS variables are assumed to follow gaussian membership function and the CSPs membership function is shown on a trapezoidal function.

The centroid method as shown in Equation 2.4 and Figure **??** uses the center of mass, which is represented as z, in a fuzzy output distribution to determine a single scalar value. The membership of the fuzzy sets is presented in u_c, whereas the value of the membership is represented as z_p. Finally, the crisp output from the defuzzifier is an approximation that is used to represent the PACS index of a CSP based on the evaluation of the first-level factors by the CSU. The CSUs can then use these indexes of a CSP to review if the PACS of the CSP is sufficient enough for their needs through the second level of fuzzy inference.

2.3.3.5 Creating Rule knowledge Base:

Fuzzy logic inference works in synchronization with rules given by users as well as a base set of rules. To collect a base set of rules, we created an online real-time data collection approach following below methodologies:

- User identifies themselves as expert or non-expert workflow users.
- Rule data is collected only for expert users.
- Rules are created to assess different available CSPs on a predefined scale e.g., 1-100, for non-functional PACS criteria as well as sub-factors affecting them.

Two sets of rules used in our evaluation of cloud PACS in linguistic terms are listed in Table 2.5. One of these rule sets is applied at the first level of inference, while the second set is applied at the second level of inference. The full abbreviation of each variable used in established rules is presented in Table 2.4. Figure. 2.8 shows the process of inference of rules for creating rankings of CSPs in terms of PACS criteria.

Table 2.4 Abbreviations used in fuzzy rules.

Parameter	**Abbreviation**	**Parameter**	**Abbreviation**
Performance	P	Auto-Scaling	AS
Agility	A	Capacity	CT
Cost	C	Compute	CC
Security	S	GPU	GC
Availability	AV	Storage	SC
Reliability	RL	RAM	RC
Response Time	RT	Compliance	CE
Throughput	TP	Access Controls	AC
Automation	AN	Auditability	AY
Flexibility	FY	Encryption	EN

2.3.3.6 Validating Gaussian Membership with CSUs Rule Base:

Through validation from multiple iterations from users, we consider that the membership function for CSPs follows the Gaussian membership function. For simplicity, we assume that the sub-factors contributing to PACS criteria of any CSP follow gaussian membership which is justified as most of the human intuitions when validated for a large number of people follow gaussian distribution [45]. We then aggregate the distribution of the fuzzy output from 1000 CSUs from PACS subfactors to extract the performance, agility, cost, and security index score that we term as "PACS-index" for each CSP. This simulation process for 1000 CSUs generates PACS-index values for each of the candidate CSPs and approximately fits into gaussian models. Thus, we get gaussian membership functions for each of the PACS criteria with different c_{i} and σ_{i} distribute on a scale from 1-100. We repeat the simulation again in the second stage of the Fuzzy modeling where the output is a CSP. The membership function for the input variable in the second stage i.e., PACS is aggregated from the previous step.

Table 2.5 Rules for evaluating the CSPs' PACS using a Linguistic Form.

Rule No.	**Rules description for CSP inference**
1	IF (P $\approx$ e) $\rightarrow$ CSP $\cong$ AWS
2	IF (P $\approx$ g) $\rightarrow$ CSP $\cong$ GENI
3	IF (P $\approx$ p) $\rightarrow$ CSP $\cong$ MU
4	IF (A $\approx$ e) $\rightarrow$ CSP $\cong$ AWS
5	IF (A $\approx$ g) $\rightarrow$ CSP $\cong$ GENI
6	IF (A $\approx$ p) $\rightarrow$ CSP $\cong$ MU
7	IF (C $\approx$ p) $\rightarrow$ CSP $\cong$ AWS
8	IF (C $\approx$ a) $\rightarrow$ CSP $\cong$ GENI
9	IF (C $\approx$ e) $\rightarrow$ CSP $\cong$ MU
10	IF (S $\approx$ e) $\rightarrow$ CSP $\cong$ AWS
11	IF (S $\approx$ a) $\rightarrow$ CSP $\cong$ MU
12	IF (S $\approx$ a) $\rightarrow$ CSP $\cong$ GENI
Rule No.	**Rules description for CSPs PACS inference from subfactors**
1	IF (AV $\approx$ e) $\wedge$ (TP $\approx$ g) $\rightarrow$ P $\cong$ e
2	IF (RL $\approx$ e) $\wedge$ (RT $\approx$ g) $\rightarrow$ P $\cong$ e
3	IF (RT $\approx$ p) $\wedge$ (AV $\approx$ e) $\rightarrow$ P $\cong$ g
4	IF (AN $\approx$ e) $\wedge$ (FY $\approx$ g) $\rightarrow$ A $\cong$ e
5	IF (FY $\approx$ g) $\wedge$ (AS $\approx$ g $\rightarrow$ A $\cong$ g
5	IF (AS $\approx$ p) $\wedge$ (CT $\approx$ p) $\rightarrow$ A $\cong$ p
6	IF (CC $\approx$ p) $\wedge$ (GC $\approx$ a) $\rightarrow$ C $\cong$ e
7	IF (GC $\approx$ a) $\wedge$ (RC $\approx$ a) $\rightarrow$ C $\cong$ a
8	IF (SC $\approx$ g) $\wedge$ (CC $\approx$ b) $\rightarrow$ C $\cong$ p
9	IF (CE $\approx$ e) $\wedge$ (AY $\approx$ a) $\rightarrow$ S $\cong$ g
10	IF (AC $\approx$ a) $\wedge$ (EN $\approx$ a) $\rightarrow$ S $\cong$ a
11	IF (AY $\approx$ b) $\wedge$ (EN $\approx$ p) $\rightarrow$ S $\cong$ b

2.3.3.7 Membership distribution

Once the fuzzy engineering model shown in Figure 2.5 is trained, the model returns a membership distribution for the selection of CSPs for specific expert user inputs. To bias the optimizer's objective function toward a CSP using membership distribution, we formulate the below formula -

$$Membership\ Factor\ (m^p) = (1 - M^p) \tag{2.5}$$

where, M^{p} is the membership value of p^{th} platform obtained from the fuzzy engineering model. This formulation ensures that the effective cost of instances from a platform with higher membership distribution is reduced in the objective function represented in Equation 3.6. Since the maximum value of membership distribution for

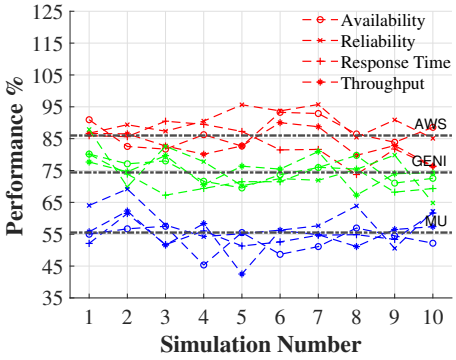
any CSP is 1, hence CSPs with lower membership value will have a lower reduction in effective price, and thus this formulation holds validity for different distributions.

Table 2.6 Application workflows used in evaluation experiments.

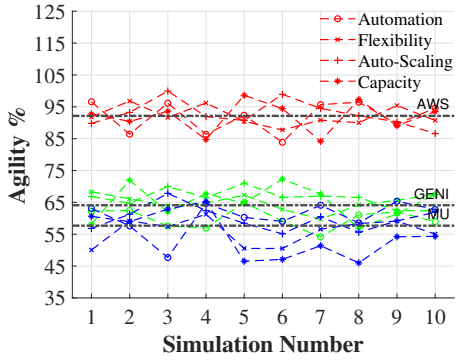
Workflow Name	**Workflow Description**	**Resources Data(Gb), vCPU, RAM(Gb), Network(Gbps), Clock**
FastQC	FastQC Quality Check workflow is used to conduct the quality control checks on raw sequencing data so that we can remove some low-score data before the next step of analysis.	{3, 4, 5, 0.5, 1}; {4, 8, 10, 1, 1.2}
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.	{10, 12, 20, 2, 1.2}; {10, 16, 25, 4, 1.4}
PGen	PGen workflow [44] allow users to identify the single nucleotide polymorphisms (SNPs: substitution of a single nucleotide that occurs at a specific position in the genome) and insertion-deletion (indels: insertion or deletion of nucleotides from a sequence) and perform SNP annotation.	{20, 20, 50, 8, 1.4}; {20, 24, 100, 12, 2}; {30, 28, 120, 16, 2.2}

2.3.4 Evaluation

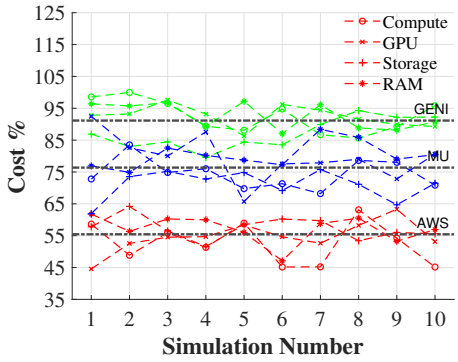
We evaluate the accuracy of the multi-level fuzzy engineering model and its ability in evaluating cloud platforms from the user's perspective by validations. For the evaluation of the fuzzy model, we assume that the user is an expert user. We create the fuzzy model system and create rules defining behaviors of our three base cloud service providers i.e., MU (private cloud), GENI (community cloud), AWS (public cloud). The PACS index of CSPs can be calculated based on the level of satisfaction a CSU receives from a given CSP. More specifically, we ensure the validity for each of these PACS-index for CSPs by applying the theorem as given by [41]. As per the theorem suggested in [41], we sample inputs from only those CSUs who are experts i.e., their crisp input for cloud platform services aligns with true service level agreements from cloud platforms. Such a process of identifying expert CSUs needs external independent third-party validation. For our solution, we verify expert users by cross-checking their choices to align with a large number of CSUs. To validate the effectivity of our proposed solution, we simulate 1000 CSUs iteratively 10 times with inherent biases for subfactors as shown in Table 2.4 towards three CSPs namely AWS, GENI, and MU.



(a) Average performance measurement for CSPs using the fuzzy model with simulation on 1000 CSUs for 10 iterations



(b) Average agility measurement for CSPs using the fuzzy model with simulation on 1000 CSUs for 10 iterations



(c) Average cost measurement for CSPs using the fuzzy model with simulation on 1000 CSUs for 10 iterations

Fig. 2.9 Efficiency of our fuzzy inference engine in quantifying the performance , agility and cost metric for various CSPs

Figures 2.9a, 2.9b, and 2.9c show our simulation towards Performance, Agility, and Cost benchmarking of three concerned CSPs. Each data point in Figures 2.9a, 2.9b, and 2.9c represent the average approval in % towards a specific sub-factor of the CSPs. We simulate the process 10 times, and we find the fuzzy output distribution toward PACS of CSPs from the fuzzy inputs in each iteration. For example, considering only performance, each CSU in each iteration results in one fuzzy output so we get 1000 performance fuzzy values using centroid defuzzification (note that the membership functions are gaussian) for each CSP. We remark that we have created the fuzzy models to simulate defuzzification methods using the Matlab Fuzzy Toolbox [46]. When this process is iterated 10 times, we get 10000 performance fuzzy values for each CSP. The average of these fuzzy values for these three CSPs is shown in Figures 2.9a, 2.9b, and 2.9c as straight horizontal lines. Note that these fuzzy values have a range of 1 to 100. When

this scale is divided into 5 subdivisions as mentioned in Table 2.3, it provides intuition for rule creation in the second level of fuzzy inference as shown in Table 2.5. Based on these results, we prove the validity of our proposed fuzzy inference system which can guide a new user towards CSPs based on their preferences towards sub-factors.

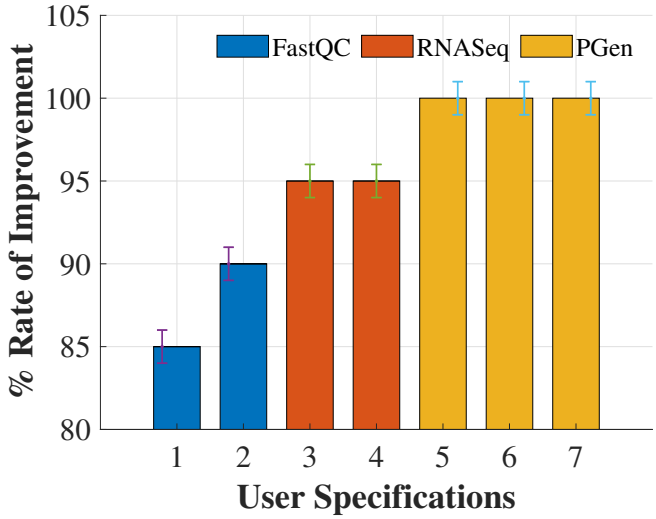


Fig. 2.10 Percentage of workflows with improved execution time (i.e., lesser time) when fuzzy engineering was used 'with' OnTimeFLC resource broker vs. OnTimeFLC was used 'without' fuzzy inference inputs.

Performance Evaluation: We now describe the evaluation results of our proposed framework OnTimeFLC's efficiency in composing cost-effective template solutions for any given user requirements and preferences. We evaluate the efficiency of the fuzzy model to improve application workflow execution time performance at different resource specifications as shown in Table 3.6. We compose, allocate, and compare template solution performance for the user specifications for two cases:(i) Fuzzy model with the expert user was considered to improve the execution performance of workflows by selection of CSPs using OnTimeFLC and, (ii) Only OnTimeFLC's core ILP optimizer engine. The experiments were repeated (10 times) iteratively for each specification to calculate the average execution time for workflows. Figure 3.12 corresponds to our experimental results that show the time to execute the workflow reduced in 98% of the cases with OnTimeFLC including fuzzy engineering. We remark that the rate of improvement in efficiency is highly dependent on the expertise of the user as the fuzzy model suggests CSPs considering users' PACS rating of CSPs.

2.4 Summary

In this chapter we discuss our work done toward understanding and capturing effectively user requirements for their applications through intuitive GUI designs and studying the relevance of the structure of the data required to understand true QoS expected by the use. We also discuss potential approaches to effectively capture users' biases, experiences, and expertise with cloud service providers to improve resource recommendations for their applications. Through this combination of capturing user engagement and user preferences, we were able to truly capture user's perspectives and needs from the cyberinfrastructure which is then transferred to the recommender engines of the resource broker in a structured format so as to create the best resource recommendation for the users.

CHAPTER 3

Multi-Cloud Optimization for Resource Selection

3.1 Overview

Data-intensive science applications in fields such as e.g., bioinformatics, health sciences and high-energy physics are becoming increasingly dynamic with resource requirements. These applications often require specialized instruments and computing, networking and storage resources (e.g., scientific instruments, supercomputers, federated data repositories, public clouds [1], [2]). Researchers using these applications which are based on advanced scientific workflows frequently require a diverse set of resources that are often not available within a single CSP. They also demand synergistic multiple CSP resources. They seek to create analytics workflows to utilize cloud solutions easily, efficiently and with high performance, while containing costs and time for configuring the necessary resources. As a result, the selection and configuration of multi-cloud resources for modern applications requires handling of cost-performance trade-offs as well as intra-CSPs operability which could be based on factors of CSP usability, policies, and security guidelines. Thus, selection and configuration of multi-cloud resources for modern applications requires handling objective factors such as *performance*, *agility*, *cost*, *security* (i.e., PACS factors) as discussed in Chapter 1.

3.2 Optimization for Resource Selection

We propose a novel multi-objective optimization model integrated within a novel resource brokering middleware viz., *OnTimeURB* in order to meet KBCommons biological user-defined constraints of bioinformatics application workflow performance, cost,

and CSPs interoperabilities. The research problem we are trying to address is to ease the process of maximizing cloud resource utilization for non-cloud expert users while at the same time respecting constraints from these users. Our target user being a non-cloud expert, we prescriptively recommend an intelligent set of custom solutions optimized based on different criteria of performance and cost so as to facilitate users to compare and select the best-suited solution template to deploy a given workflow. For evaluation of our OnTimeURB middleware implementation, we conduct experiments with a catalog of bioinformatics application workflows developed within KBCommons framework, over varying next-generation sequencing (NGS) data types and sizes of datasets for various organisms. We consider four CSP resources (i.e., Amazon Web Services [1], GENI [2], XSEDE [47] and local MU [3]) featuring more than 300 different machine configuration instances in our experiments. User data to be processed with the workflow is stored in CyVerse [19] infrastructure. We use the Pegasus workflow management system [48] for the creation and maintenance of the workflow's pipelines. Further, HTCondor [49] and pyGlidein [50] tools are used to automate the distribution of sub-tasks from the workflows to resources distributed across CSPs deployed using cloud template solutions generated from our middleware. We compare our OnTimeURB middleware results with a state-of-the-art k-nearest neighbors (k-NN) approach [12] in order to evaluate OnTimeURB's ability to consistently create cost and performance-efficient optimal solutions. Further, we show the detrimental effect of increasing cloud interoperability constraints on the cost of a cloud solution template composition.

3.2.1 Significance and Related Works

In this section, we discuss some of the related works for cloud solution composition based on single and multiple CSPs.

Deelman et al [9] highlights a comprehensive description of the Pegasus Workflow Management System which helps in the execution of large-scale, multi-stage simulation and data analysis pipelines to enable the study of complex systems. Pegasus can jointly be used with HT Condor [49] a framework to set up high throughput computing system, for the creation of scientific workflows. However neither Pegasus nor HTCondor

helps users in cloud resource allocation, rather they focus on scheduling and handling communication within subtasks of workflows.

Mireslami et. al. [10] propose an optimizer based on geometric programming which minimizes deployment costs while fulfilling user requirements and Quality of Service (QoS). In follow-on work, they propose an algorithm based on the Branch-and-Bound technique which refines the optimizer by adding constraints. Given that the CSPs in reality offer a discrete number of machine instances, their proposed optimizer based on geometric programming disregards the granularity of instances. Moreover, their optimization considers only a single CSP.

Zou et. al. [51] has proposed an Artificial Intelligence (AI) based cloud template composition. Their work differentiates CSPs into different domains each having subdomains such as computing or storage, with each subdomain identifying a set of service files. Further OWLS-Xplan [52] is used for composing services based on user requirements. However, their work assumes that inter-cloud communication is time-consuming, and costly and tries to minimize the number of CSPs in the solution without considering performance, cost, and inter-cloud incompatibility factors. Kurdi et. al. [53] again assume similar inter-cloud communication limitations, and propose a COM2 algorithm that creates multi-CSPs based solutions. This algorithm ensures that the clouds with maximum services are selected, thus increasing service robustness but again does not consider performance, cost, or inter-cloud incompatibility. An algorithm to solve the multi-objective task scheduling problem is proposed by Panda et. al. [54] by considering the minimization of makespan and the total cost in a multi-cloud environment. However, their algorithm assumes unlimited resources on all CSPs and focuses on scheduling tasks of the application, without considering user-defined constraints on resources and multi-cloud compatibilities.

Antequera et. al. [12] provide users with custom template solution recommendations based on heterogeneous CSPs. The recommended templates are based on the well-recognized k-nearest neighbor's algorithm (k-NN) algorithm. Predefined template solutions are created and stored in a catalog, and a template solution with the closest match to user specification is suggested to the user. Nevertheless, the approach fails

to consider inter-cloud operability and is less cost-efficient compared to our proposed OnTimeURB middleware.

3.2.2 Multi-Cloud Resource Allocation

We divide our middleware's control flow into three logical sub-divisions to help non-expert cloud users, namely: i) Knowledge Interface System (KIS), ii) Template Composition iii) Template Deployment and Utilization.

3.2.2.1 Knowledge Interface System

We leverage KBCommons [18] web portal to collect user resource requirements via a KIS module. KBCommons portal provides a comprehensive web resource for storing, sharing, analyzing, exploring, and visualizing multi-organism genomics and multi-omics data. We implemented and integrated our KIS module with bioinformatics analytics workflows user interface (UI) deployed within the KBCommons portal to collect user-specific resource constraints and criteria to execute user workflows. The KIS presents users with a set of questions (e.g.,vCPU/RAM required) to capture their requirements. Additionally, users can provide a threshold maximum on the requested resources, for example, a 10% threshold on 10Gb RAM memory requirement will allow the OnTimeURB optimizer to compose solutions with RAM memory up to 11Gb.

3.2.2.2 Template Composition

User specification collection is succeeded by composition wherein the requirements are passed to the OnTimeURB optimizer engine to compose template solutions similar to the composition by Antequera et. al. [12]. Each template solution in the catalog is formatted as a JSON object comprising of a set of distinct machine node instances. Machine node instance refers to machine configuration in terms of available processing units(CPUs), memory, bandwidth, etc. For example, a1.medium is one of the machine instances provided by AWS [1].

3.2.2.3 Template Deployment and Utilization

Once the user selects a recommended template solution by OnTimeURB, the resources are deployed following the template specification. Fig 3.1 shows the control flow between OnTimeURB components that we have designed and implemented for biological researchers. The bioinformatics workflows are created based on the Pegasus workflow management system [48]. Pegasus, HTCondor, pyGlidein, and iRODS [19] tools are configured to run the workflows on the deployed cloud resources. Pegasus breaks the workflow's pipeline into subtasks which are further scheduled on different cloud machines by HTCondor and pyGlidein. Subsequently, the performance monitoring of applications and deployed resources can be done via the KBCommons portal, which helps users in making better resource specifications in future interactions with OnTimeURB.

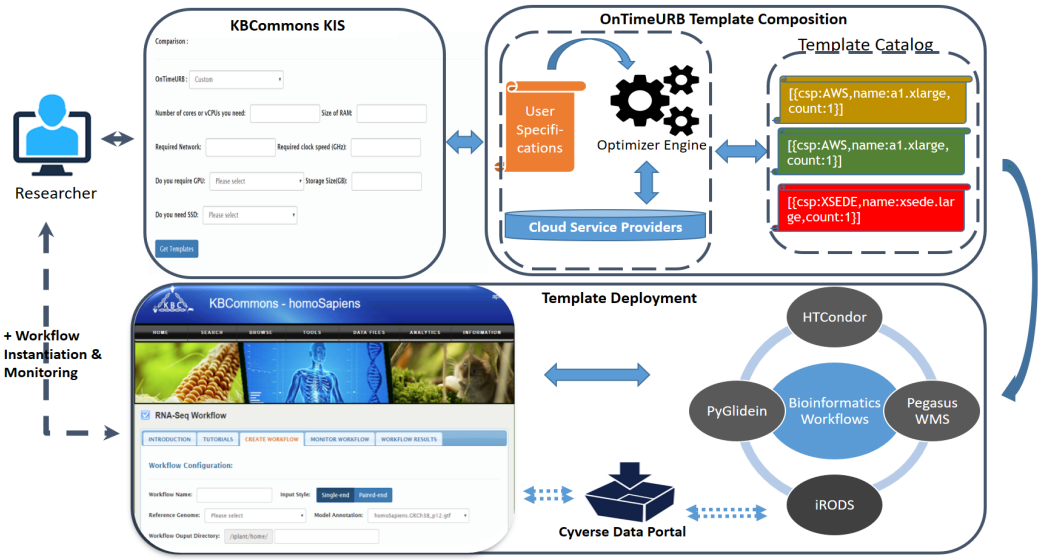


Fig. 3.1 Application control flow comprising of a) Collection: KBCommons Knowledge Interface System (KIS) to collect user specification. CyVerse data portal is used wherein user can keep their data to be processed b) Template Composition: multiple templates are composed and classified by the optimizer, c) Deployment: Tools namely, Pegasus WMS, HTCondor, Pyglidein, iRODS configured and integrated with KBCommons, utilize the selected template for execution of bioinformatics workflows.

3.2.3 Approach

3.2.3.1 Integer Linear Programming Based Optimization

Custom template composition essentially can be formulated as a selection of machine configurations from a distributed set of diverse cloud instances under specific constraints, which is an NP-hard problem. To effectively formulate the problem into a solvable model, we use CPLEX [17] optimizer to create a relevant optimization model. The objective of the model is defined in a manner to reduce the cost of the template solution for user requirements. The optimization model is constrained by user specification of required resources and resource thresholds. For example, a user can specify a requirement as {*cpu:8, ram:16Gb, storage: 60Gb*} with a threshold of 25% which acts as a constraint in the model. The below classifications identify potential template responses from the OnTimeURB middleware

High Performance and Cost: All user-defined resources are amplified in step sizes up to a user-defined threshold limit. Each step gives an amplified resource constraint which results in a corresponding solution. Prospective provisioning: minimal configuration {*cpu:8, ram:16Gb, storage: 60Gb*}, maximum configuration {*cpu:**10**, ram:**20Gb**, storage: **75Gb***} etc.

Low Performance and Cost: The template is the closest match to user resource specification with minimal overprovisioning. Prospective provisioning: {*cpu:8, ram:16Gb, storage: 60Gb*} OR {{*cpu:4, ram:8Gb, storage: 30Gb*} *and* {*cpu:4, ram:8Gb, storage: 30Gb*}} etc.

We base our model on only the resources that can be considered as a joint selection from multiple cloud instances such as e.g., the number of CPU cores, RAM memory, Storage, etc. for optimization. The allocation of such resources can be optimized using our OnTimeURB engine. Our OnTimeURB optimizer engine uses combinatorial optimization with Integer Programming (IP) to return an integral number of machine instances from CSPs.

To effectively consider different machine configurations from different CSPs, we formatted the configurations into JSON objects as detailed in Listing 3.1. Real machine

instance configurations were extracted from more than 300 physical instance types from different CSPs. Since our middleware focuses on non-frequent cloud service users hence cost for a pay-as-you-go basis was referred from the CSPs. This formatting allows users to add more instances from varying CSPs in the knowledge base as per availability and thus can effectively be used by the optimizer for template composition.

```
[{ "csp":"AWS",
    "OS":"LINUX",
    "name":"a1.large",
    "vCPU":"2",
    "ram":"4",
    "price":"0.051",
    "network":"10",
    "clock":"2.3",
    "pricing_ssd":"0.10",
    },{...} ]
```

Listing 3.1: JSON formatted machine instances provided to optimizer engine

To ensure hardware granularities are available as instances from CSPs, the optimization problem is formulated as an integer linear programming (ILP) model which is convex and guarantees the best possible solution. Equation 3.6 ensures the objective to minimize the cost of template solution creation. The number of instances (x^p_i) are calculated subject to the constraints deployed in the model referenced in Equation 3.7. Constraints are created on instances such that the resources contributed from the deployed instances should satisfy user-specified requirements (S_t). Templates are obtained by varying thresholds (S^{th}_t) on resources. Equation 3.8 adds constraints on instances that belong to incompatible CSPs using Δ_{p1p2}.

Table 3.5 summarizes parameters and variables pertaining to our modeling Equations 3.6,3.7,3.8.

$$minimize \sum_{p=1}^{P} \sum_{i=1}^{I^p} \frac{c_i^p}{w_i^p} \cdot x_i^p \tag{3.1}$$

Table 3.1 Parameters, Sets and Variables for Problem Formulation

Parameter Symbol	**Parameter Description**
c_i^p	cost of renting i_{th} instance at P_{th} service provider
w^p	user defined factor ranging (0-1] for P_{th} provider to assign preference to CSPs
A_i^p	max number of instances of type i available at provider p
R_{it}^p	resource t_{th} available with i_{th} instance of P_{th} resource
S_t	specification requirement of type t
S_t^{th}	threshold on resource of type t
Δ_{p1p2}	binary number indicating compatibility between CSPs
M	large integer number
Set Symbol	**Set Description**
I^P	total number of instances in P_{th} provider
P	total number of providers
T	total type of resources
Variable Symbol	**Variable Description**
$x_i^p \in [0, A_i^p]$	is an integer variable denoting the number of instances of type i at provider p
$\delta_{P_1P_2} \in \{0, 1\}$	is a binary variable ensuring only one of incompatible P1, P2 is selected

subject to:

$$\sum_{p=1}^{P} \sum_{i=1}^{I^p} R_{it}^p \cdot x_i^p \geq S_t + S_t^{th}, \forall t \in T \tag{3.2}$$

$$\begin{aligned} &\sum_{i=1}^{I^{p_1}} x_i^{p_1} \leq M \cdot \delta_{p_1p_2} \\ &\qquad\qquad\qquad\qquad , \forall p_1, p_2 \in P : \Delta_{p_1p_2} = 0 \\ &\sum_{i=1}^{I^{p_2}} x_i^{p_2} \leq M \cdot (1 - \delta_{p_1p_2}) \end{aligned} \tag{3.3}$$

3.2.3.2 Interoperability

Researchers may have a personal bias towards certain CSPs due to specific services, policies, or security requirements unique to users' workflow applications. Since these requirements vary with users, we implemented an interoperability matrix within On-TimeURB middleware to allow users to be able to custom-define CSP selection as per

their expertise and preferences. The matrix is illustrated in Table 3.4, where "1" indicates that corresponding row and column CSPs are operable with each other, while "0" indicates that the CSPs are not operable simultaneously. As per the implementation of the matrix, diagonal elements are always "1" because we assume instances of a cloud provider will always be operable with each other. As an example Table 3.4 suggests AWS [1] and GENI [2] can not operate together. Hence, machine instances from both of these CSPs will not be taken in the template composition simultaneously, however, these CSPs can compose templates by combining with other CSPs.

Table 3.2 Interoperability matrix

$CSPs$	**PLSCI2**	**AWS**	**GENI**	**XSEDE**	**local MU**
PLSCI2	1	1	1	1	1
AWS	1	1	0	1	1
GENI	1	0	1	1	1
XSEDE	1	1	1	1	1
local MU	1	1	1	1	1

NOTE: Rows and column represent CSPs. PLSCI2 is the local host machine containing the workflow application.

3.2.4 Evaluation

In this section, we show how the proposed OnTimeURB performs consistently better in composing cost-effective template solutions for user requirements as compared to the state-of-the-art k-NN approach [12] for custom template composition. We also evaluate the impact of the interoperability matrix on cost-to-solution for templates.

3.2.4.1 Tools and Configurations

OnTimeURB template recommendations are evaluated using the implemented control flow as shown in Fig. 4.4. KIS module was integrated into the KBCommons science gateway portal to collect user specifications. The OnTimeURB optimizer engine was hosted on an independent GENI [2] node machine as a web service. Pegasus [48], HTCondor [49], pyGlidein [50], and CyVerse iCommand [19] were installed and configured on the independent local machine (PLSCI2) and they interact with the optimizer via REST web service calls. Bioinformatics application workflows were created using Pegasus and configured in the node machine (PLSCI2). Workflow-specific tools are

packaged along with the workflow for ease of distribution on different cloud platforms. The workflow accesses user data located at CyVerse using the iCommand in real-time for the processing steps.

3.2.4.2 Bioinformatics Workflow Description

In recent years, next-generation sequencing (NGS) technology has improved dramatically, with costs dropping and the number and range of sequencing applications increasing exponentially. A wide variety of types of high-throughput sequencing can be generated from RNA or DNA molecules through NGS library construction and sequencing including techniques such as whole genome sequencing (WGS), RNA-Seq, ChIP-Seq, DAP-Seq, RIP-Seq, methylation and more. To efficiently utilize the large-scale NGS data for analysis, we have developed six bioinformatics workflows, and we have further centralized the input data for these workflows using CyVerse [19]. CyVerse data storage was used as the cloud storage infrastructure and all raw data and final results are stored and managed therein. The Pegasus workflow system is used to define and control the required computational tasks of workflows. These tasks include user-defined tasks as well as Pegasus-added tasks such as data staging between the CyVerse data storage and cloud computing infrastructure's file system such as XSEDE [47], PLSCI2 and AWS [1]. Pegasus also adds data cleanup tasks to maintain and minimize the workflow footprint on the file system as the workflow progresses.

We have split the analysis pipeline into multi-step and parallel processes to gain the most efficiency. The workflow describes the dependencies among the tasks as a directed acyclic graph (DAG), where the nodes are tasks and the edges denote the task dependencies. And each computing task can optionally be assigned a proper number of cores and memory that they can efficiently utilize to run on the cloud computing site. We made the bioinformatics workflows available via a web-based implementation integrated with KBCommons. The workflow submission within KBCommons is mainly intended for biological researchers and guides them through five steps for workflow creation and submission, which allows them to access the results within the KBCommons as well.

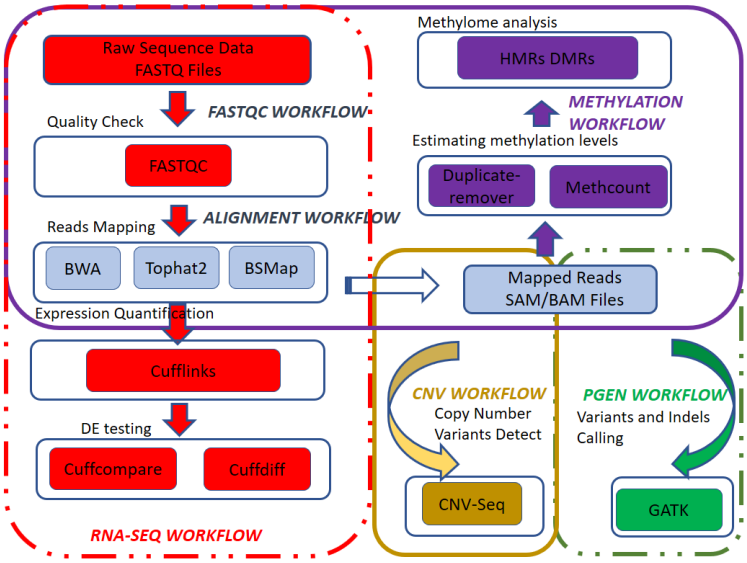


Fig. 3.2 Bioinformatics Workflow which shares tasks and tools for their execution.

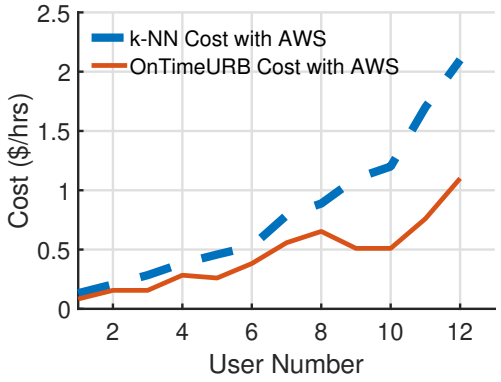
Table 3.3 Increasing User Specifications for the Workflows

User Cases	Data Size (Gb)	vCPUs	RAM(Gb)	Network(Gbps)	Clock(Ghz)
Fastqc					
User_1	3	4	8	0.5	1
User_2	4	4	15	2	1.2
User_3	6	6	15	4	1.4
User_4	8	6	25	8	1.4
RNASeq					
User_5	10	12	25	4	1.4
User_6	10	14	30	10	1.4
User_7	20	14	50	10	1.4
User_8	20	16	60	10	2
Pgen					
User_9	20	20	30	10	1.4
User_10	20	22	40	10	1.4
User_11	30	24	60	12	1.4
User_12	35	26	60	16	2

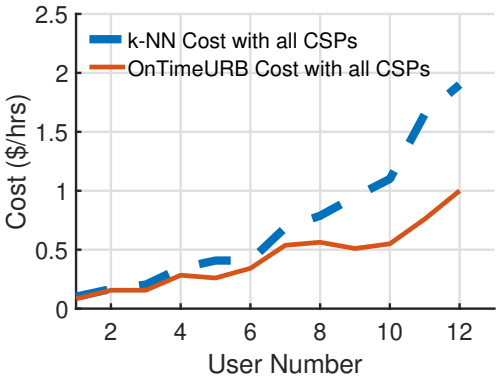
3.2.4.3 Cost to Custom Template Efficiency

To evaluate the efficiency of OnTimeURB to suggest cost-effective solutions, we considered three of the implemented workflows based on their resource requirement from lower to higher scale (size of NGS data and workflow pipeline), namely, Fastqc (low scale), RNASeq (medium scale), PGen (high scale). Note that the Alignment workflow is a sub-workflow of RNASeq and Pgen refer Fig. 3.2. Further, we assessed resource requirements for these workflows for different sizes of data, based on our perceived performance requirements. Table 3.10 summarizes these user specifications for

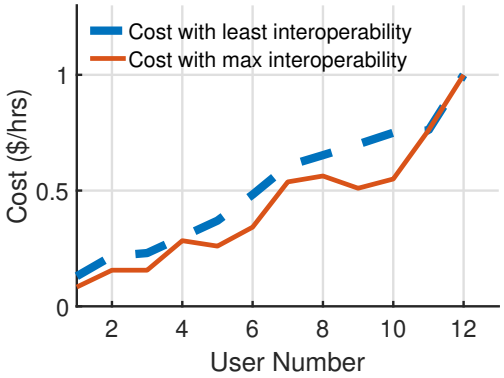
the workflows. To assess the benchmark efficiency of the optimizer we did not apply interoperability constraints between the four candidate CSPs-that is to say AWS [1], GENI [2], XSEDE [47], local MU [3].



(a) Comparison of cost to templates with OnTimeURB optimizer and k-NN approach with only AWS



(b) Comparison of cost to templates with OnTimeURB optimizer and k-NN approach with all four cloud service provider



(c) Comparison of cost to templates with least and maximum interoperability between CSPs

Fig. 3.3 Template resource suggestions comparison using k-NN and OnTimeURB, when partial (only CPU) and full specifications (all requirements) are provided with partial (only AWS) and full (i.e., all CSPs) knowledge base. The resource allocations closest to the specified specifications are better.

Fig. 3.3a compares OnTimeURB's cost to the template solution for the user specifications from Table 3.10 with only AWS [1] as candidate CSP against the well-recognized k-NN approach suggested in [12], while Fig 3.3b compares the cost to a solution considering all four CSPs. From the cost comparisons, it is evident that OnTimeURB outperforms kNN for all user's requirements specified in Table 3.10 for composing template solution in a single CSP as well as the multi-cloud base. We acknowledge that the cost to template largely depends on the size of CSPs instance knowledge base considered within the optimizer, nevertheless, OnTimeURB is expected to outperform kNN with

all size of instance knowledge base because kNN searches for a single machine instance matching closest to user requirement while OnTimeURB composes template solution by combining multiple machine instances, thus OnTimeURB is leveraging the instance granularities from CSPs in a better manner.

3.2.4.4 Interoperability cost repercussions

The interoperability matrix is incorporated within OnTimeURB to allow users to add restrictions on simultaneous CSP selection of incompatible CSPs. Since the matrix can be filled in numerous ways by users, we identify the matrix representing the least and maximal compatibility between CSPs. Further, we evaluate the maximum cost to template repercussions caused due to user-identified interoperability matrix. Fig. 3.3c summarizes the cost of template composition for user specifications from Table 3.10, for minimum and maximum user-defined intra-CSPs compatibility. The results suggest an increase in cost for template composition when intra-CSPs constraints are applied. However at higher scales of user specification cost to compose solutions is the same, this is due to the fact that at higher scales OnTimeURB composed templates consisted of only a single CSP thus nullifying the effect of intra-CSP constraints.

3.3 Machine Learning Based Resource Allocation

Selection and configuration of multi-cloud resources for modern applications requires handling objective factors such as *performance*, *agility*, *cost*, *security* (i.e., PACS factors) as shown in Figure 1.3. This multi-cloud resource brokering can be formulated as a multi-dimensional optimization problem in resource selection that is contextually dependent on user preferences and their application requirements. Moreover, the diversity in the resources and capabilities offered by different CSPs creates a situation of excessive/overwhelming choice for users. The choice issue is especially significant for users who are not cloud platform experts and require relevant guidance for multi-cloud resource selection. In this paper, we present a novel multi-objective optimization algorithm that powers a multi-cloud resource broker middleware viz., *OnTimeURB* that

considers objective PACS factors in the brokering process. The algorithm is aided with an ML model in order to additionally account for user-centered subjective factors for workflow executions. The ML model is designed to learn the bias of expert users towards cloud platforms for different requirements of functional criteria and workflow sizes.

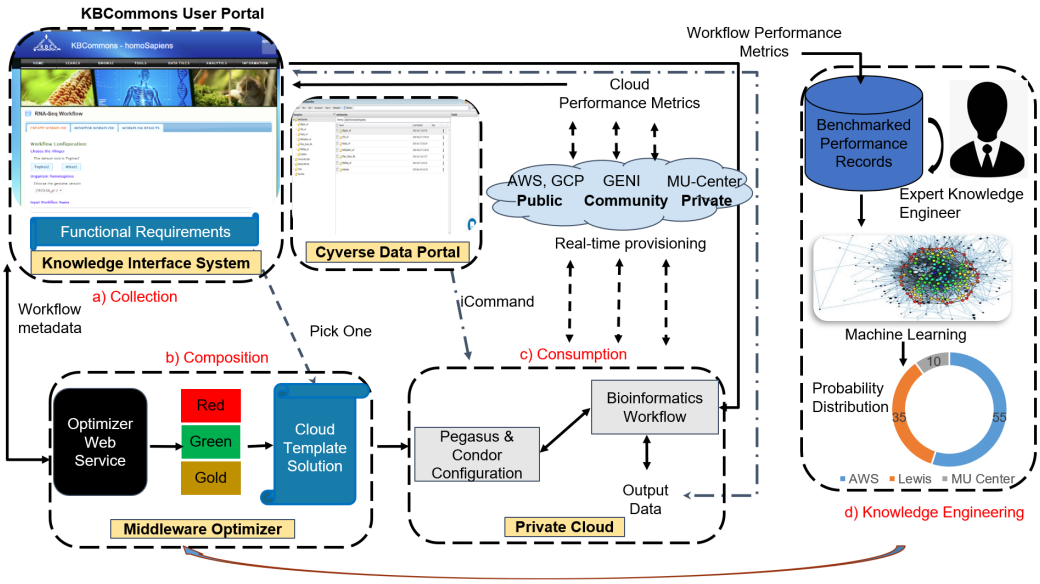


Fig. 3.4 Brokering lifecycle comprising of a) Collection - Knowledge Interface System (KIS) deployed in KBCommons portal [18] to collect user specification; b) Composition - Templates are composed and classified by optimizer into Red, Green, and Gold templates which are presented in the KIS for selection; c) Consumption: Pegasus, HTCondor and workflows are configured in a GENI [2] node machine, and HTCondor as per cloud template schedules tasks in resources of AWS, GENI or GCP and d) Knowledge Engineering - Performance metadata such as execution time, success rate are collected from the consumption step and stored in a central repository, which is used to train a machine learning model.

3.3.1 Significance and Related Works

For the providers, the placement of heterogeneous VMs on their infrastructure poses a key optimization challenge. One of the contributions given in [55] involves a heuristic backtracking algorithm for VM placement while considering inter-VM interactions in heterogeneous data centers. Their algorithm aims to increase the performance of a chosen infrastructure by optimizing the placement of VMs. In a similar work, authors in [56] consider a service level agreement-based resource allocation problem for multi-tier cloud-hosted applications. Their work is mainly focused on the optimization of multi-dimensional resource allocations in data centers using an algorithm based on force-directed search. In contrast, our work focuses on using the context of users' func-

tional and non-functional requirements. In another recent related work [57], authors proposed a multicriteria optimization approach for CSP selection in a multi-cloud setting. The analytic hierarchy process (AHP) was used for the assessment and assignment of priorities to the CSPs, which was further optimized by using three metaheuristic algorithms viz., simulated annealing, genetic algorithm, and particle swarm optimization. Note that their optimizations were focused on the selection of a single CSP for fulfilling a service requirement. We have used a similar AHP-based solution to compare and evaluate our ML-guided template solution.

As discussed previously, provisioned cloud resources do not always meet QoS expectations. There have been attempts to understand this fluctuation of QoS levels through learning-based approaches, which has motivated our use of a knowledge engineering approach in OnTimeURB.

There are multiple factors such as VM configurations, application workload, network bandwidth, and computation capabilities that govern the selection of resources, and these factors can vary in real time within an application deployment. This necessitates leveraging learning-based solutions for improving real-time resource selection. An interesting result can be seen in the observation from [58] which proposed PARIS, a Performance-Aware Resource Inference System. They noted that bigger machine configurations are not always better, and similar machine configurations can provide different performances. They used opaque workload resource requirements to benchmark testbed machines (VMs) and a random forest model was found to be effective for allocating VMs. Although the approach considers the subjectively perceived performance of machines and assumes a static system of VMs, it can be further generalized so that dynamic resource allocations can be fulfilled. While the approach in [58] successfully studies the QoS fluctuations, large applications such as bioinformatics workflows feature diverse pipelines with different processes. Typically, every workflow process has unique resource requirements, hence resource optimization has to be effective to fulfill requirements from all these processes. Consequently, the problem also needs to optimize cloud resource allocations while also ensuring optimal resource scheduling. For optimizations of task scheduling and management, learning-based models have been

proposed. In an exemplar work on optimal task scheduling, a learning model viz., Decima based on graphical neural network (GNN) and reinforcement learning (RNN) [59] was detailed. GNN is used for organizing the application job metadata, while RNN is used for learning the parallelism level of job execution. The objective is on improving average job completion time over state-of-the-art scheduling algorithms. Given that these approaches use learning-based algorithms, they are dependent on training methodologies which can pose adaptability problems for changing user resource requirements of applications. Nonetheless, the above works motivate our approach to utilize ML models for aiding optimization through knowledge engineering.

In our previous work [20], we provided a brokering approach that takes into account user PACS factors for the allocation of multi-cloud resources using ILP. In this work, we extend our prior work by modeling agility and its implications for multi-cloud resource brokering. More importantly, we introduce a novel knowledge engineering approach to better understand and utilize true QoS values from cloud resources contextually for applications. The knowledge engineering approach involves iterative execution of applications at varying workloads, metadata data collections for successful executions, and performance benchmarking of data using expertise from expert knowledge engineers. This information is used for creating training data for an ML model to learn the QoS behavior of cloud resources. This information also helps in the further integration of results from knowledge engineering with the ILP model for cloud template solution recommendations that improve the performance of data-intensive scientific workflows.

3.3.2 LifeCycle for Resource Allocation

Figure 3.4 shows the brokering lifecycle of OnTimeURB for KBCommons that includes: i) Collection, ii) Composition, iii) Consumption, and iv) Knowledge Engineering. These steps are designed to help non-expert cloud users of KBCommons working on bioinformatics workflow applications to leverage experiences from expert users and increase the efficiency of their infrastructure resources. The steps specifically leverage the OnTimeURB orchestrated components of ILP-based optimizer, ML-based knowledge engineering, and multi-cloud resource task schedulers.

3.3.2.1 Collection

The resource brokering involves collecting user resource requirements through a Knowledge Interface System (KIS) integrated in the KBCommons web portal [18] as shown in Figure 3.4. KIS purpose in KBCommons is to collect user-specific resource constraints and criteria to execute bioinformatics workflows hosted on private cloud resources. The KIS presents KBCommons users with a set of questions (e.g., number of vCPU/RAM required) organized into functional groups such as storage, networking, computation, and software requirements. Additionally, users can provide a upper bound on the requested resources e.g., 20% threshold on 200 Mbps of network bandwidth. This input indicates to the optimizer that up to 240 Mbps of network bandwidth may be required by the user, and thus the optimizer may also generate templates with 240 Mbps network interfaces. The collection process also requires users to identify themselves as expert/non-expert cyber users. This data allows us to record the choices and feedback of the expert knowledge users on the performance and efficacy of recommended cloud template solutions. The data is further used to train an ML model, which can help in making better and more intelligent choices for non-expert users.

3.3.2.2 Composition

Once the user-specified requirements for the application workflow are collected, they are input to the OnTimeURB optimizer to create template solutions forming a catalog. Each template solution in the catalog is formatted as a JSON object comprising of a set of distinct machine node instances. Each node instance represents a machine configuration in terms of the available processing units (i.e., CPUs, memory, network bandwidth, storage). For example, $a1.medium$ is one of the machine instances available from AWS. A sample representation of the template is provided as shown in Listing 3.2. We can see that three distinct types of AWS instances are used to compose the template. We categorize the compositions of templates into three distinct types: i) Red, ii) Green and iii) Gold template solutions.

```
[{csp:AWS,name:t3.nano,count:2},
{csp:AWS,name:t3.micro,count:1},
```

```
{csp:AWS,name:t3.small,count:1}]
```

Listing 3.2: Sample template format consisting of three type of instance machines from AWS.

Red Solution: Strict user-defined resource constraints are considered. This is the most cost-effective optimal solution. The template is the closest match to the user resource specification with minimal over provisioning. Potential resource provisioning: {*cpu:4, ram:8Gb, network: 200Mbps, storage: 30Gb*} OR {{*cpu:2, ram:4Gb, network: 200Mbps, storage: 15Gb*} *and* {*cpu:2, ram:4Gb, network: 200Mbps, storage: 15Gb*}}.

Green Solution: All user-defined resources are amplified in step sizes up to a user-defined threshold limit. The agility of all the CSPs is considered to be the same for this recommendation. Each step gives an amplified resource constraint which results in a corresponding over-provisioned solution. The number of steps up to the threshold decides the number of solutions generated in this template solution recommendation category. Potential resource provisioning: a) minimal {*cpu:4, ram:8Gb, network: 200Mbps, storage: 30Gb*}, b) maximum {*cpu:**5**, ram:**10Gb**, network: **250Mbps**, storage: **37Gb***}.

Gold Solution: All user-defined resources are amplified in step sizes up to a set threshold limit. Hence the templates lie between a minimal and maximum configuration similar to the Green solution, however, the agility of the CSPs is applied as per the user preferences. Considering the user will give more preference to agile platforms while defining agility factors, OnTimeURB will prefer more agile CSPs to compose template solutions. We provide more details about the agility factor in Section IV-A.

3.3.2.3 Consumption

Once a cloud template is selected and initialized, the resources as per the template specification are automatically deployed. Figure 4.4 illustrates the detailed steps of the template consumption and monitoring that we have designed and implemented for a set of custom scientific workflows in KBCommons based on the Pegasus workflow management system [48] and HTCondor job scheduler [49]. Pegasus and HTCondor are configured to run a given application workflow on the deployed cloud resources

recommended by a solution template. Pegasus breaks the workflow tasks into subtasks framed into a directed acyclic graph with edges representing dependencies, which are further scheduled on different cloud instances by HTCondor scheduler and pyGlidein glidein service [50].

3.3.2.4 Knowledge Engineering

To guide non-expert users with resource allocation for application workflows, we deploy an ML model for knowledge engineering i.e., to learn from the resource allocation patterns of expert users for varying applications. The essence of the ML model lies in the process of learning about performances in terms of the time metric of sub-tasks within workflows. The iterative execution of varied workflows at different workloads by expert users and their feedback helps in creating the benchmark records shown in Figure 3.4.

3.3.3 Approach: Key Optimizations

3.3.3.1 Security

We consider security policy incompatibilities between CSPs as a constraint to multi-cloud resource selection for users. Security requirements are unique to groups of users and can range from authentication, access control, sensitive data storage, administrative privileges, location of CSP and communication link between nodes, and more. For example, GCP [4] provides encrypted storage by default while AWS [1] and Azure [5] do not support default support. Consequently, the users processing data that requires file encryption by default would want to choose processing resources from GCP [4]. To mitigate such security policy issues, OnTimeURB allows users to be able to custom-define an interoperability matrix as per their expertise and preferences. For the matrix illustrated in Table 3.4, “1” indicates that corresponding row and column CSPs are compatible with each other, while “0” indicates that the corresponding CSPs are not compatible.

Table 3.4 Interoperability matrix.

CSP	**LOCAL**	**AWS**	**GENI**	**GCP**	**AZURE**
LOCAL	1	1	1	1	1
AWS	-	1	1	0	1
GENI	-	-	1	1	1
GCP	-	-	-	1	0
AZURE	-	-	-	-	1

NOTE: Rows and column represent CSPs. Matrix suggests GCP [4] and AZURE [5] are incompatible. Consequently, machine instances from both of these CSPs will not be taken in the template composition simultaneously. LOCAL is the user host machine containing the application.

3.3.3.2 Agility

Since our proposed optimizer design assumes multi-cloud resources, it considers the availability of diverse resources from compatible CSPs while optimizing the cost. Moreover, to ensure the agility of the template solutions, we use a parameter w^{p} that includes a weight on the cost of instances of the CSPs. The w^{p} will be a number between 1 to 10, and represents a global view of services and features provided from individual CSPs. A smaller w^{p} suggests a fewer number of services from a CSP. The optimizer will also track the increase in cost-to-agility ratio in the inclusion of instances from a CSP as per Equation 3.6. Weights w^{p} for the agility of cloud providers can be assessed by the number of agile services provided by the provider (f^{p}) against a combined set of all agile services (F) offered from all providers assessed by the user as formulated in Equation 3.4, and normalized between 1 to 10. These services can be e.g., 'Locations of servers', 'Available Security Protocol', 'Available Networking Options', and 'Reliability and Failover'. User discretion is used to identify agile services from CSPs as users will have varying standards in terms of agility expectations. We use Cloudarado [99] as a reference data source to identify a set of agile services required by a specific user. Some of the agile features can be recognized as cloud features and management, cloud servers, images and licenses, networking, security, and reliability and failover. We further define an *Agility Index* to compare agility of individual template solutions generated from the middleware optimizer as given by –

$$AgilityFactor(w^p) = \frac{f^p}{F} \tag{3.4}$$

$AgilityIndex$: Compute power (u^{p}) weighted mean of agility factor (w^{p}) of CSPs involved in template composition. These parameters, as summarized in the definitions Table 3.5, are inputs for Equation 3.5. We assume the compute power as the primary criteria to assign weights due to the fact that - performance for data processing workflows highly depends on compute power and network interface bandwidth of a resource node. A CSP which is assigned a larger fraction of computing power and network interface bandwidth requirement from a user will have a larger impact on the overall agility of the template. Hence, we define –

$$AgilityIndex = \frac{\sum_{p=1}^{P} u^p \cdot w^p}{\sum_{p=1}^{P} u^p} \tag{3.5}$$

3.3.3.3 Optimizer

To ensure that granular instances from CSPs are considered, the optimization problem in Figure 3.4 is formulated as an integer linear programming (ILP) model, which is convex and guarantees the best possible solution. Equation 3.6 ensures the objective to minimize the cost of template solution creation, while also selecting better performing CSPs represented by the probability factor d^{p}. The value of w^{p} for CSPs is normalized between 1 to 10 for our study. The selection of the normalization scale is dependent on the user preferences and related resource requirements in absolute values. We choose this normalization scale because the individual resources in our evaluation from different CSPs have resources e.g., CPU cores and RAM on a scale of 10s. This selection of the normalization scale prevents unbalanced normalization of cost values in Equation 3.6. The probability factor is elaborated more in the next section. The number of instances ($x^{\mathrm{p}}{}_{\mathrm{i}}$) are calculated subject to the constraints deployed in the model referenced in Equation 3.7. Constraints are created on instances such that the resources contributed from the deployed instances should satisfy user-specified requirements (S_{t}). Equation 3.8 adds constraints on instances that belong to incompatible CSPs using Δ_{p1p2}.

The parameters, as summarized in definitions Table 3.5, are inputs for our model

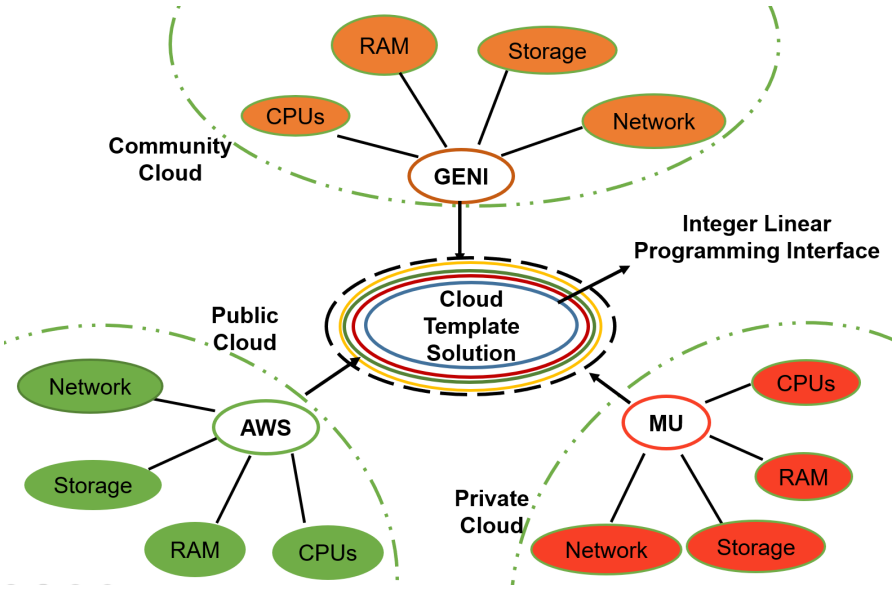


Fig. 3.5 OnTimeURB's core optimizer engine is based on integer linear programming and a knowledge base featuring three different cloud providers: (a) AWS (public cloud), (b) GENI (community cloud), and (c) MU Data Center (private cloud).

described with Equations 3.6, 3.7, and 3.8.

$$minimize \sum_{p=1}^{P} \sum_{i=1}^{I^p} \frac{c_i^p}{w_i^p} \cdot x_i^p \cdot d^p \tag{3.6}$$

subject to:

$$\sum_{p=1}^{P} \sum_{i=1}^{I^p} R_{it}^p \cdot x_i^p \geq S_t + S_t^{th}, \forall t \in T \tag{3.7}$$

$$\begin{aligned} &\sum_{i=1}^{I^{p_1}} x_i^{p_1} \leq M \cdot \delta_{p_1 p_2} \\ &\qquad\qquad , \forall p_1, p_2 \in P : \Delta_{p_1 p_2} = 0 \\ &\sum_{i=1}^{I^{p_2}} x_i^{p_2} \leq M \cdot (1 - \delta_{p_1 p_2}) \end{aligned} \tag{3.8}$$

3.3.3.4 ML Model Algorithms:

We design our data model to collect workflow execution metadata and label the iteration with a CSP, based on user feedback for the execution. Inherently, every application execution will have its own configuration footprint feature which uniquely defines it. Depending on user feedback, application workflow executions with the same features might be tagged with different CSP labels, while executions with different features can share same label. To effectively acknowledge this observation for guiding resource

Table 3.5 List of Parameters, Sets and Variables.

Parameter Symbol	**Parameter Description**
f^p	Number of agile services from the P_{th} CSP
c_i^p	cost of renting i_{th} instance at P_{th} CSP
w^p	agility factor of P_{th} CSP
u^p	number of CPUs assigned with P_{th} CSP by optimizer in a template composition
d^p	probability factor for p^{th} platform
F	Super set of all agile services identified by the user
A_i^p	max number of instances of type i available at CSP p
R_{it}^p	resource t_{th} available with i_{th} instance of P_{th} resource
S_t	specification requirement of type t
S_t^{th}	threshold on resource of type t
Δ_{p1p2}	binary number indicating compatibility between CSPs
M	large integer number
Set Symbol	**Set Description**
I^P	total number of instances in P_{th} provider
P	total number of provider
T	total type of resources
Variable Symbol	**Variable Description**
$x_i^p \in [0, A_i^p]$	is an integer variable denoting the number of instances of type i at provider p
$\delta_{P_1P_2} \in \{0, 1\}$	is a binary variable ensuring only one of incompatible P1, P2 is selected

allocation optimization, we use a probabilistic classification technique because absolute classification cannot model subjective user preferences. Towards this goal, we employ a Naive Bayes classification algorithm as shown in Figure 3.6. We specifically use Naive Bayes classifier as it converges quicker than other machine learning models such as logistic regression and requires fewer data points for convergence. Moreover, we use the ML model as a guide for CSP selection in the ILP model due to which, a highly precise/overfitted ML model is not required. The fundamental Naive Bayes assumption is that each feature should be: (a) independent, and (b) equal. We ensure that both these prerequisites are met while we train the model.

Generating Training Data: It is a well-established fact [60] that data modeling for machine learning is a very crucial step for avoiding under-fitting or over-fitting. Figure 3.6 presents our workflow orchestration process that involves training an ML model

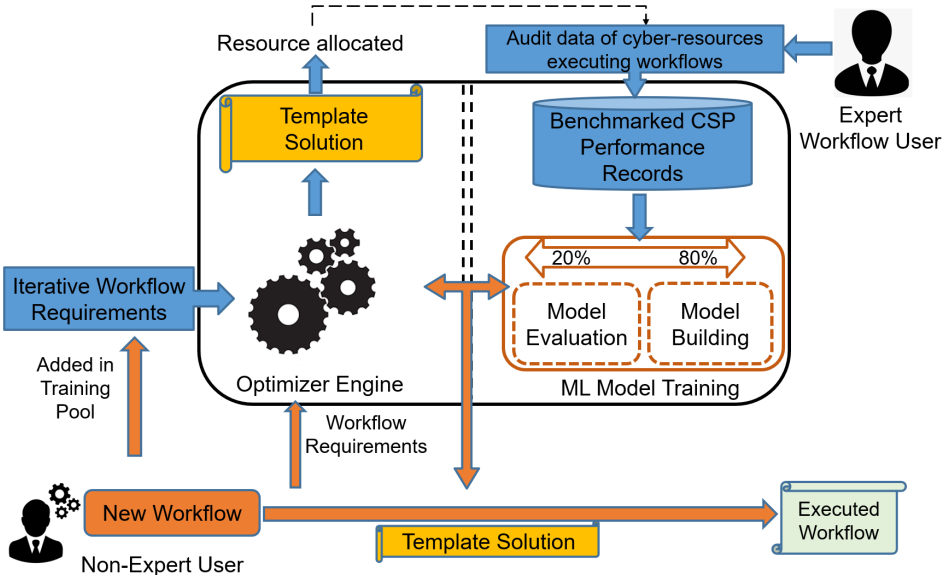


Fig. 3.6 OnTimeURB's core components: a) OnTimeURB optimizer engine, b) Naive bayes based machine learning model.

using diverse workflows and integration of expert user feedback. The optimizer engine uses the pre-trained ML model in the process to inform non-expert users with recommended template solutions for their workflow execution demands. Details of the ML model training are as follows: expert user provides feedback on the execution of the workflows which is further used to evaluate the performance of a template solution in the workflow execution. The ML model is trained with the workflow requirements, template solution, and feedback provided by the expert user. The new workflows submitted by non-expert users are added to the workflow training pool as well for improving the ML model training. For the new workflow, the optimizer engine takes workflow requirements as input and uses CSP recommendations from the pre-trained ML model to create a customized template solution for the non-expert user's workflow execution. To avoid related pitfalls and to collect workflow execution data to train the ML model in Figure 3.6, we created an online real-time data collection approach following the below steps:

- Users identify themselves as expert or non-expert workflow users.
- Metadata is collected only for expert users for considering positive cases of cloud platform selection.
- Workflow execution iteration is done by varying potential decisive factors such

as application workflow, type of data, quantity, and fragmentation of data, time taken by sub-tasks of the applications, and type of computing resources suggested in template solutions.

- Feedback from expert users is collected for the overall performance of the execution of application workflow as either: $satisfied$ or $unsatisfied$.
- For positive feedback which is represented by success and lesser time for execution, workflow configuration is tagged with the cloud platform having the highest ratio of computing power in the selected template solution. Workflow configuration is represented by the type of workflow, size of data to process, and fragmentation of data files.
- Iteration of workflow execution is done to collect data.
- Model is trained with all the application workflows.
- Model is trained with an equal number of executions of each application workflow; for our experiments, we trained the model with 20 executions of each workflow for a given data size. Since we executed each workflow with 5 different input data sizes, there were cumulative 100 executions performed for each workflow.
- Model is trained with a similar size of input data for each application workflow.

Probability Factor Calculation: Once the ML model shown in Figure 3.6 is trained, the model returns probabilistic distribution of the selection of CSPs for a specific workflow execution configuration. To bias the optimizer's objective function toward a CSP that has higher probabilistic classification, we formulate the below formula –

$$ProbabilityFactor(d^p) = (1 - P^p) \tag{3.9}$$

where P^p is the probabilistic classification of p^{th} platform obtained from the ML model. This formulation ensures that the effective cost of instances from the platform with higher probabilistic classification is reduced in the objective function represented in Equation 3.6. Given that the summation of probabilities for all CSPs is 1, other

CSPs with lower probabilistic classification values will have a lower reduction in effective price. Consequently, this formulation holds validity in all cases of probability distribution among CSPs.

3.3.4 Evaluation

In this section, we evaluate our OnTimeURB multi-cloud resource broker's efficiency in composing cost-effective template solutions for user requirements as compared to the state-of-the-art k-NN approach used by Antequera et. al. [12] for custom template composition. We also show the impact of security considerations (i.e., the interoperability matrix) on the cost to solution for the recommended templates and the ability of our OnTimeURB middleware in composing more agile solutions. Lastly, we describe our evaluations to show the ability to use an ML model in enhancing the ability of OnTimeURB for recommending template solutions with better runtime performance.

3.3.4.1 Bioinformatics Workflow Applications

To demonstrate OnTimeURB's ability to efficiently utilize the large-scale NGS data for analysis, we use seven bioinformatics workflows as described in Table 3.6. Elaborate details of workflow implementation are available in our previous work [20]. We have centralized the input data for these workflows using CyVerse [19], which is used as the cloud storage infrastructure that manages raw data as well as final results. The Pegasus workflow system is used to define and control the required computational tasks of workflows. These tasks encompass user-defined tasks as well as Pegasus-added tasks such as data staging between the CyVerse data storage and computing infrastructure's file system such as AWS [1] (public cloud), GENI [2] (community cloud) and MU Data Center [3] (private cloud). Pegasus also adds data cleanup tasks to maintain and minimize the workflow footprint on the file system as the workflow progresses.

3.3.4.2 Evaluation of Cost to Template Solution

We consider three of the implemented workflows based on their resource requirement from lower to higher scale (size of NGS data and workflow pipeline complexity in terms

Table 3.6 Bioinformatics Workflow Applications.

Workflow Name	Workflow Description
FactQC	FastQC Quality Check workflow is used to conduct the quality control checks on raw sequencing data so that we can remove some low-score data before the next step of analysis.
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 discover of new, unannotated transcripts.
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.
PGen	PGen workflow allow users to identify the single nucleotide polymorphisms (SNPs: substitution of a single nucleotide that occurs at a specific position in the genome) and insertion-deletion (indels: insertion or deletion of nucleotides from a sequence) and perform SNP annotation.
Copy Number Variation (CNV)	Copy number variation analysis helps detect the chromosomal copy number variation (CNV: is a phenomenon in which sections of the genome are repeated and the number of repeats in the genome varies between experimental groups/conditions) that may cause or may increase risks of various critical disorders.
Methylation	Methylation analysis helps estimated the methylation levels of each genomics cytosine and identified the differentially methylated regions between the experimental groups/ conditions.
Single Cell RNA-Seq	Single-cell RNA-Seq analysis allows users to align single-cell RNA-Seq read, perform clustering cells and then assign the cell type identity to clusters via biomarks.

of the number of tools used), namely, FastQC (low scale), RNASeq (medium scale), PGen (high scale). In addition, we assess resource requirements for these workflows for different sizes of data, based on common performance requirements summarized in Table 3.10. Only the Red category template from the optimizer is compared because of its design to perform a resource allocation that is closest to a user's requirement. For multiple CSP cases, we do not consider an interoperability constraint i.e., all elements in the interoperability matrix are set to 1, and all CSPs are considered as being equally agile.

Below are the descriptions of base categories considered for evaluation results in Figures 3.7 and 3.8:

- **AWS-KNN:** Knowledge base consisting of resources from only AWS [1], and

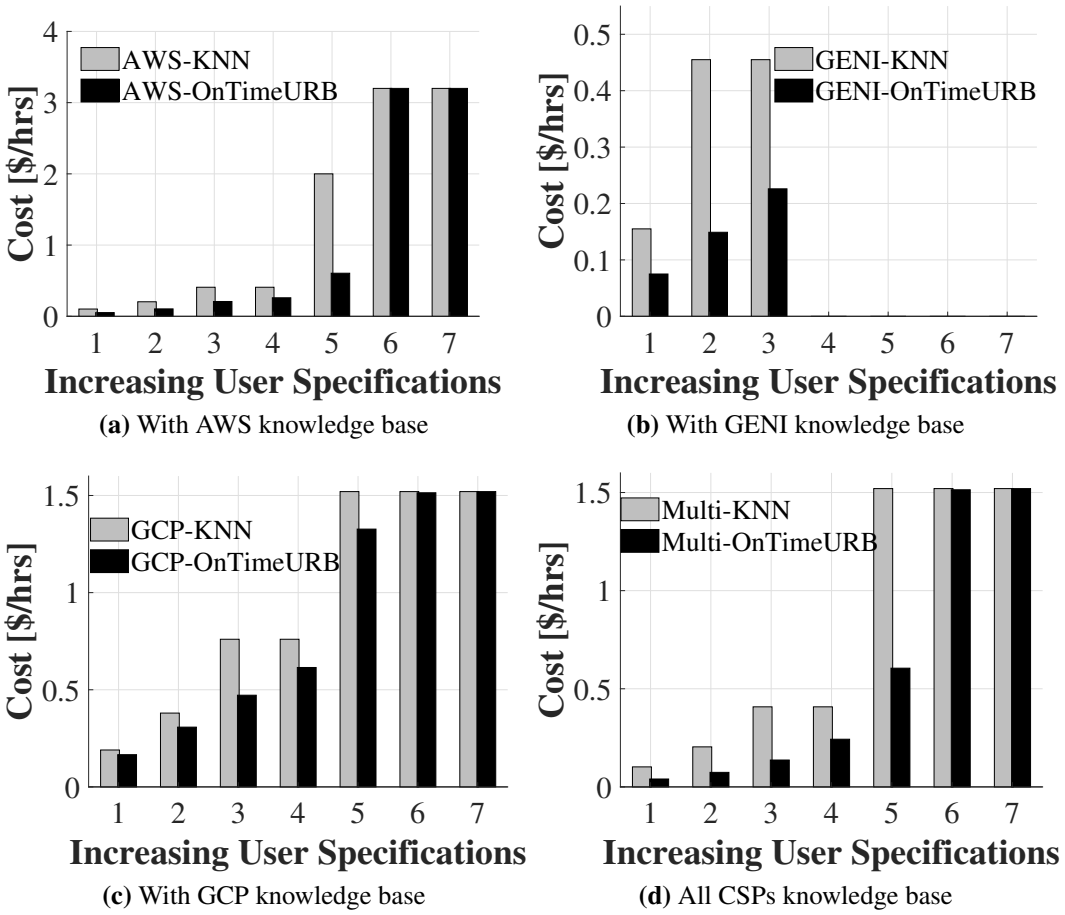
Table 3.7 Increasing User Specifications for the Workflows.

User Cases	Data Size (Gb)	vCPUs	RAM (Gb)	Network (Gbps)	Clock (Ghz)
FastQC					
Case_1	3	4	5	0.5	1
Case_2	4	8	10	1	1.2
RNASeq					
Case_3	10	12	20	2	1.2
Case_4	10	16	25	4	1.4
PGen					
Case_5	20	20	50	8	1.4
Case_6	20	24	100	12	2
Case_7	30	28	120	16	2.2

k-NN was used to get the template.

- **AWS-OnTimeURB:** Knowledge base consisting of resources from only AWS [1], and OnTimeURB was used to get the template.
- **GENI-KNN:** Knowledge base consisting of resources from only GENI [2], and k-NN was used to get the template.
- **GENI-OnTimeURB:** Knowledge base consisting of resources from only GENI [2], and OnTimeURB was used to get the template.
- **GCP-KNN:** Knowledge base consisting of resources from only GCP [4], and k-NN was used to get the template.
- **GCP-OnTimeURB:** Knowledge base consisting of resources from only GCP [4], and OnTimeURB was used to get the template.
- **Multi-KNN:** Knowledge base consisting of resources from all four CSPs, and k-NN was used to get the template.
- **Multi-OnTimeURB:** Knowledge base consisting of resources from all four CSPs, and OnTimeURB was used to get the template.

In Figure 3.8, we compare resource allocations with k-NN and our OnTimeURB optimizer for Cases 4 and 7, when either partial (i.e., novice user case) or full user (i.e., expert user case) specifications are provided. For both Case 4 and Case 7, when full specifications are provided, our OnTimeURB optimizer with multiple CSP resource options gives better solutions (i.e., provides a closer match to use case specifications).



(a) With AWS knowledge base

(b) With GENI knowledge base

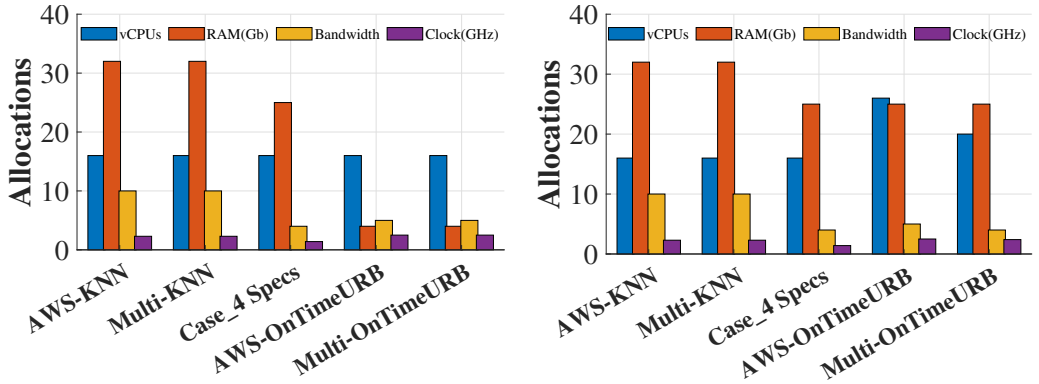
(c) With GCP knowledge base

(d) All CSPs knowledge base

Fig. 3.7 Cost analysis of template solutions showing cost benefits of OnTimeURB compared to k-NN over different cloud providers: *(top, left)* Amazon Web Services (AWS), *(top, right)* GENI community cloud, *(bottom, left)* Google Cloud Platform (GCP), and *(bottom, right)* all of four CSPs.

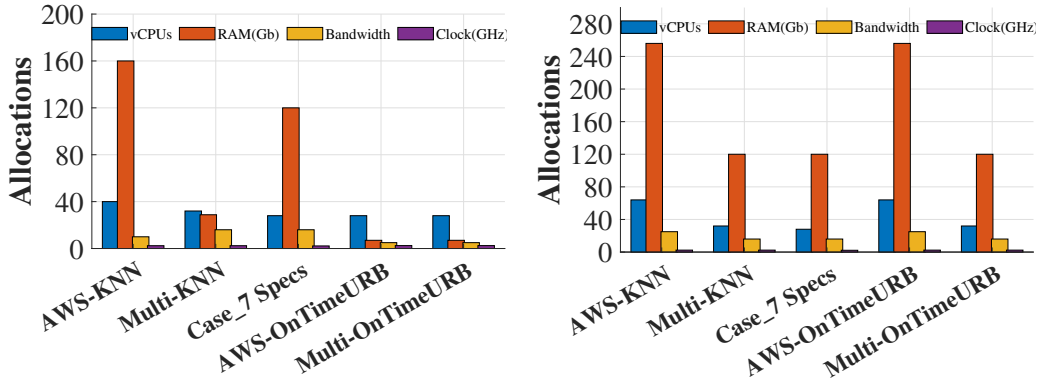
We remark that similar behavior seen in Cases 4 and 7 is also observed in other cases, and the benefits of OnTimeURB become more evident depending on the size and diversity of the knowledge base and user specifications. In contrast, the k-NN recommends an over-provisioned template solution for the same case. The results also illustrate how we can obtain better solutions with multiple CSP resource options compared to e.g., only AWS [1] dataset owing to more cloud resource diversity. Both k-NN and our optimizer perform poorly when only the CPU is specified. This can be attributed to the fact that both methods optimize cost by considering only CPU while disregarding other resources.

The above finding from Figure 3.8 shows the ability of our OnTimeURB optimizer in harnessing the multi-cloud dataset for generating better-matched solutions given a set of user preferences, as opposed to the k-NN solution approach. We remark that our



(a) Case_4 and only CPU specification

(b) Case_4 and all specification



(c) Case_7 and only CPU specification

(d) Case_7 and all specification

Fig. 3.8 Template resource suggestions comparison using k-NN and OnTimeURB, when partial (only CPU) and full specifications (all requirements) are provided with partial (only AWS) and full (i.e., all CSPs) knowledge base. The resource allocations closest to the specified specifications are better.

OnTimeURB optimizer performs better for template composition as it fulfills user requirements by composing templates featuring instances from multiple providers, while k-NN searches for a single instance amongst the providers to match the user requirements. Essentially, our approach optimizes at a more granular level in contrast to k-NN, and thus gives more cost-optimized solutions.

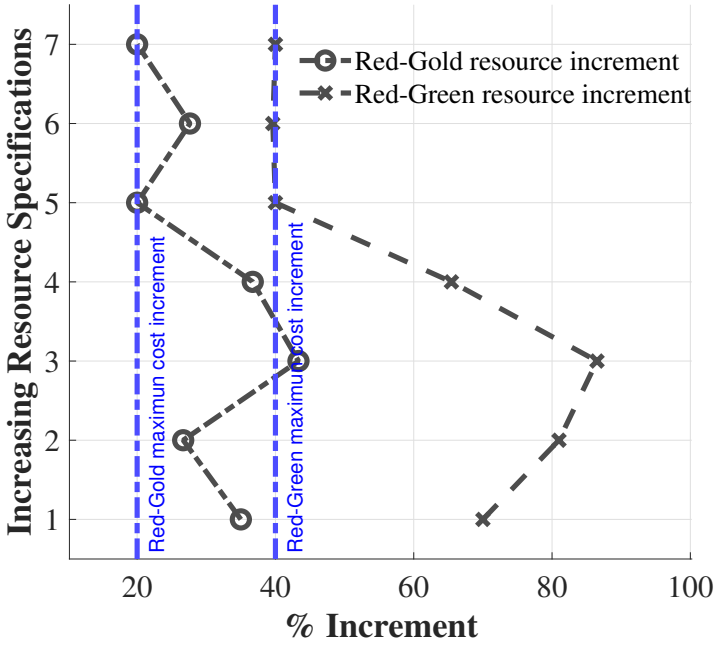


Fig. 3.9 Comparison of gold and green templates against red templates when comparing increment in cost to templates and increment in resource allocations.

3.3.4.3 Evaluation of Template Solution Choices

We compare Green and Gold template solutions with Red for % increment in cost vs. the % increment in threshold on resources. In the following, we describe the base categories considered for evaluation results in Figure 3.9:

- **Red-Gold maximum cost increment:** maximum cost increment observed from red to gold templates for all seven specifications.
- **Red-Green maximum cost increment:** maximum cost increment observed from red to green templates for all seven specifications.
- **Red-Gold resource increment:** average resource increment observed from red to gold templates for each specification.

- **Red-Green resource increment:** average resource increment observed from red to green templates for each specification.

3.3.4.4 Evaluation of User Security Requirement Overhead

We evaluate the effect of the interoperability matrix on template solution composition by considering both the minimum and maximum compatibility matrices. As per our definition of interoperability, a minimum compatibility matrix will have only diagonal elements of the matrix as '1' while a maximum compatibility matrix will have all elements of the matrix as '1'. A diagonal element with value '0' implies that machine instances from that CSP are not compatible amongst themselves. In such cases, that corresponding CSP will be auto-removed from consideration of our OnTimeURB optimizer. Table 3.8 summarizes the cost of template solution composition for user specifications from Table 3.10, considering the minimum and maximum user-defined intra-CSPs compatibility constraints.

Table 3.8 Cost comparison of templates generated with minimum and maximum compatibility.

Templates	Max. Matrix Cost ($/hrs)	Min. Matrix Cost ($/hrs)	% increase
Case_1	0.0403	0.052	29.03
Case_2	0.0741	0.104	40.35
Case_3	0.1377	0.2076	50.76
Case_4	0.1768	0.26	47.05
Case_5	0.60549	0.60549	0
Case_6	1.5144	1.5144	0
Case_7	1.7072	1.7072	0

3.3.4.5 Evaluation of User Agility Requirement Overhead

For evaluation of the impact of the agility factor on the composition of templates, we use OnTimeURB to create red and green templates for user specifications from Table 3.10. Equation 3.5 is used to calculate the Agility Index for the templates. The templates are generated by: (a) considering no agility factor i.e., all CSPs have equal agility factor (w^{p}) of '1', (b) considering an agility factor, for which we refer to the Cloudarado [99] to assess the value of w^{p} for CSPs, normalized between 1 to 10. Figure 3.10 summarizes

agility index values for templates generated for user specifications from Table 3.10. It can be observed from the Figure 3.10, that the agility index is maximum for solution templates when an agility factor is considered.

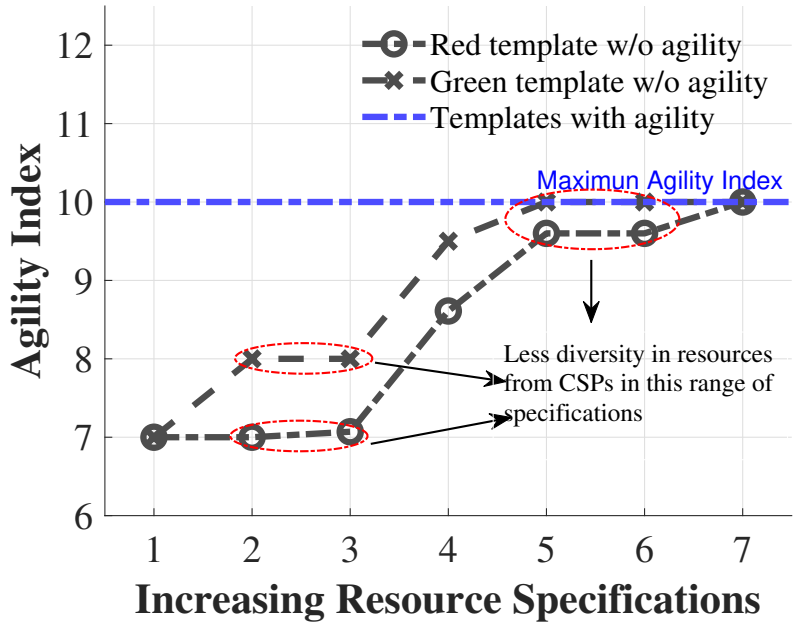


Fig. 3.10 Agility index comparison of red and green templates generated for increasing user specifications a) without agility factor, b) with agility factor.

3.3.4.6 Evaluation of Model-Based Template Solutions

To measure the efficacy of our ML model in guiding the OnTimeURB optimizer engine to create optimum cloud template solutions, we use the time to complete workflow as the user satisfaction metric. We assume that successful and lower workflow execution time implies better performance and higher user satisfaction. Towards this goal, we execute workflows with red cloud solution templates 'with' and 'without' with the aid of the ML model in our experiments.

Using the above methodology, the average time taken without the ML model (t_1) when greater than the average time taken with the ML model (t_2) implies that the ML model helps in better cloud solution template selections. We further validate this assumption with experimental results shown in Figure 3.12. For the experiments, we considered four workflows with increasing complexity involving: FastQC, Alignment, RNASeq and, PGen. The workflows were executed with different sizes of raw data in-

put ‘with’ and ‘without’ with the aid of the ML model. We assess the accuracy of the OnTimeURB optimizer using ML methods in suggesting resources that result in a better time of execution for similar configurations of workflows. As shown in Figure 3.12 we can notice improved accuracy in workflow executions for larger data sizes. This observation can be attributed to the fact that larger data sizes and complex workflows have more intermediate steps. Consequently, they require more time to execute and compensate for minor fluctuations in the performance of nodes in terms of computational resources available to the workflows.

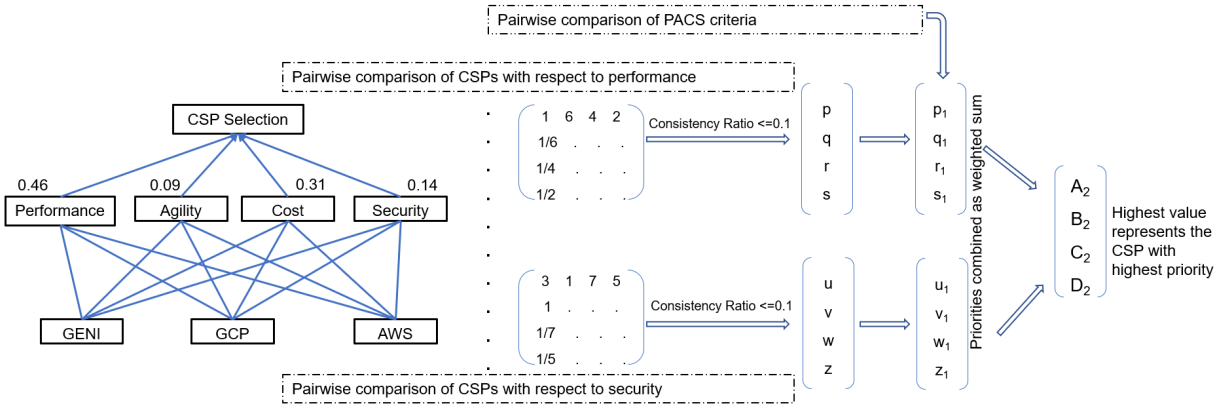


Fig. 3.11 Decision model for creating a priority list for CSP selection using the Analytic Hierarchy Process. The decision model on the left represents the priority order for each of PACS criteria contributing towards the CSP selection for the user application workflows shown in Table 3.6.

The PACS criteria based on CSP rankings derived using analytical hierarchy process[61] as shown in Figure.3.11 are further used as the d^p factor along with our ILP optimizer (ILPAHP) for governing the creation of template solutions. A direct comparison of our OnTimeURB with ILPAHP shows the true potential of our learning-based resource characterization of CSPs in comparison to resource characterization based on QoS metrics as claimed by each of the CSPs.

To compare OnTimeURB with ILPAHP, the average time taken without the ML model (i.e., only considering the ILP optimizer) was compared with the time taken when the ILP optimizer was guided with the AHP. Comparison results of ILPAHP and OnTimeURB for the different workflow executions are shown in Table 3.9. We can observe that the improvement in performance is seen more with OnTimeURB at two different raw data sizes. However, it can be noticed that ILPAHP shows very similar

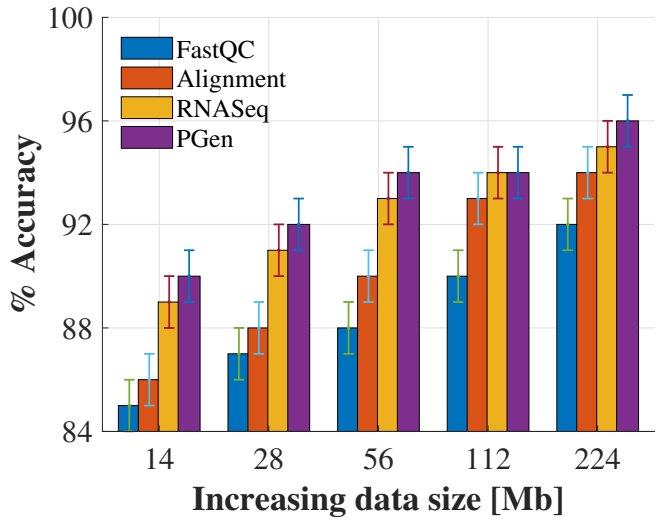


Fig. 3.12 Percentage of workflow with improved execution time performance with aid of machine learning model.

results at different raw data sizes for different workflows, while performance improves more with OnTimeURB for larger workflows involving relatively larger data processing. These results demonstrate the ability of our OnTimeURB to assess the actual performance of the CSP infrastructure dynamically to select a better template solution and obtain better performance in comparison with ILPAHP, which is based on using static priorities for a given set of CSPs.

Table 3.9 Percentage of workflows with improved execution time. ILPAHP and OnTimeURB are compared when workflows are executed with different sizes of raw data.

Workflow *Data Size*	**ILPAHP** *28 MB*	**OnTimeURB** *28 MB*	**ILPAHP** *224 MB*	**OnTimeURB** *224 MB*
FactQC	79%	87%	81%	92%
Alignment	78%	88%	79%	94%
RNA-Seq	84%	91%	84%	95%
PGen	83%	92%	86%	96%

We further characterize the utility of our OnTimeURB in generating optimal resource allocations using image processing workflows. These workflows are listed below in Table 3.10 and are created using the 'ImageMagick' command line tools [100]. The workflows are executed on the same testbed as KBCommons. Our machine learning model is first trained with these workflows. The workflows are then scheduled on the testbed similar to the bioinformatics workflows. As shown in Figure 3.13, the workflow execution is improved when using OnTimeURB similar to the case of the bioinformatics workflows. The workflows that involve the processing of larger data sets perform

well with OnTimeURB, even without machine learning similar to bioinformatics workflows. Based on the above observations, we can generalize that our OnTimeURB can learn features of tasks within new workflows and can improve the execution time of the workflows by allocating optimal resources for them, especially for larger tasks with larger data sets.

Table 3.10 User Specifications with increasing resource requirements in the image processing workflows.

Workflow Name	Workflow Description	vCPUs	RAM (Gb)
IM-1	Single intermediate processing step	2	4
IM-2	Two intermediate processing steps	3	6
IM-3	Three intermediate processing steps	4	6

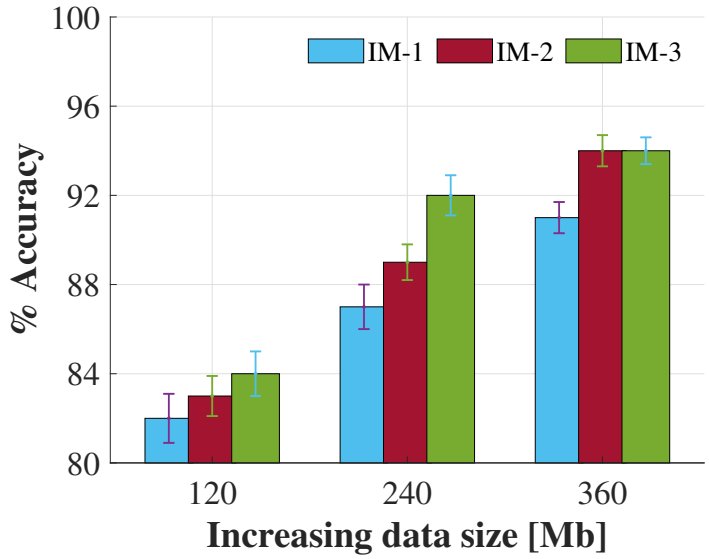


Fig. 3.13 Percentage of workflows with improved execution time performance with aid of machine learning model.

3.4 Summary

In this chapter, we have discussed architectures recommender engines for cloud resource brokers. We have first developed a recommender engine based on integer linear programming i.e., OnTimeURB which can effectively pick resources to create cloud template solutions from multi-cloud platforms while optimizing PACS of the overall solutions. The recommender also allocates resources for the user based on their selection of cloud template solutions. We further improve the recommender engine by guiding it with improved security and agility considerations as well as naive bayes based machine

learning model that characterizes the performance of underlying multi-cloud resources based on expert user feedback. The machine learning model trained with expert user inputs is unbiased from the claims made by individual cloud resource providers claims and relies on expert users' recommendations for the composition of cloud templates for non-expert cloud users. The implemented algorithms improve the performance of as much as 96% workflow applications.

CHAPTER 4

Trusted Resource Selection in Volunteer Edge Computing

4.1 Overview

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. 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 to the wider adoption of VEC. In this chapter, we propose a novel "VECTrust" model [94, 95] 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 the 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 workflow execution over batches of workflows.

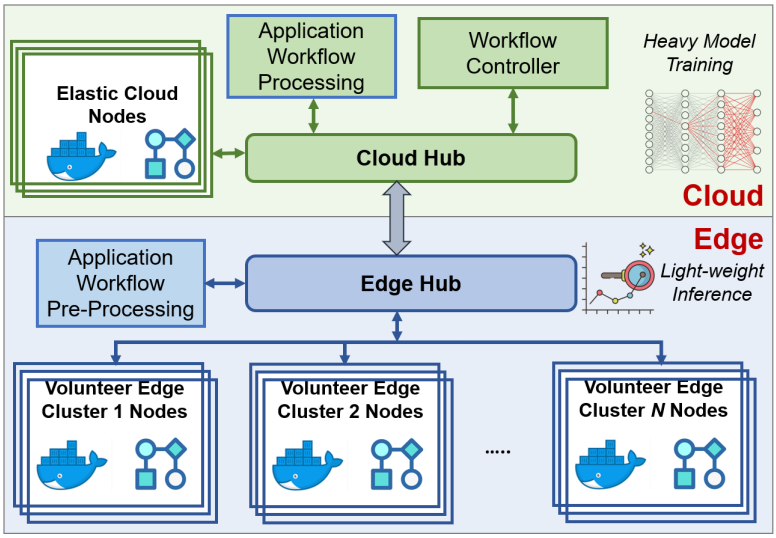


Fig. 4.1 KubeEdge-based architecture for the orchestration of edge and cloud nodes, where the cloud nodes are used for ML model training and edge nodes are used for ML model inference in the scientific application data processing.

4.2 Background and Relevance

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 4.2, the VEC clusters interact with public cloud resources through KubeEdge [62] for the management of resource-intensive stages of the scientific workflows. The system leverages VEC edge resources for the execution of small workflows and for initial data quality checks or privacy-preservation stages of large workflows that are to be scheduled overcloud nodes to incorporate user workflow preferences. KubeEdge is an ideal technology for integrating edge resources and in particular VEC resources. Characteristics of VEC resources that distinguishes them from commercial cloud platform resources 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.

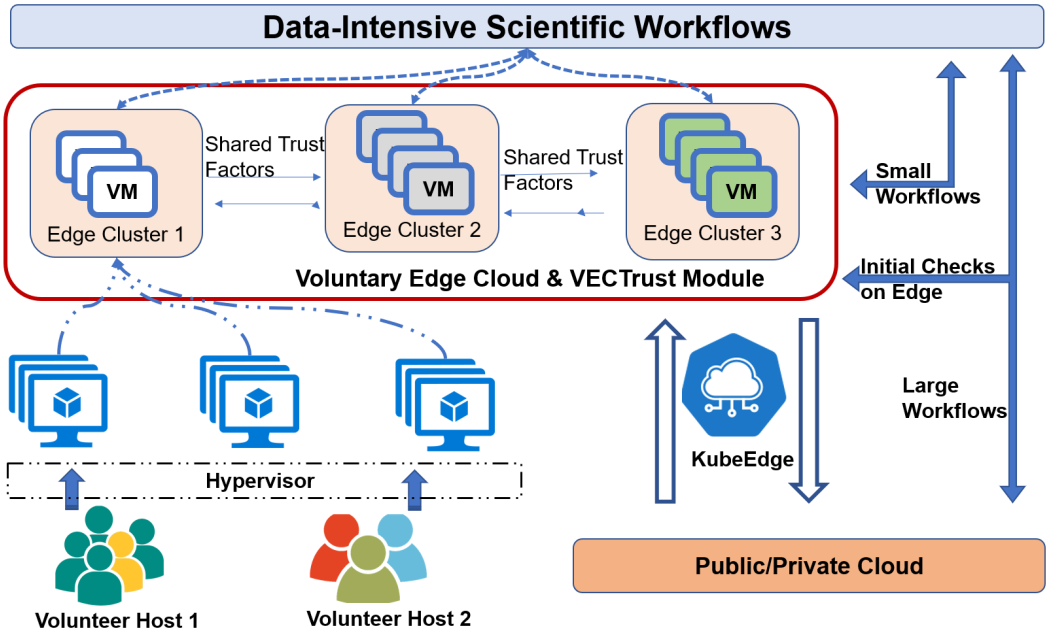


Fig. 4.2 VEC system overview showing scientific workflows being submitted to a cluster of edge-cloud resources. Edge clusters are created using resources from individual edge node servers. Workflows are transferred to public or private cloud nodes as per the trade-off between the requirement of trust and compute resources on the edge and cloud node sides.

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 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 [63] [64], there are limited works on trust modeling in a VEC context. In the VCC context, reputation-based trust is considered to assume capable/homogeneous resources, and resource 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.

4.3 Probabilistic Models for Trusted Resource Selection in VEC

We address the above VEC trust model development challenges by proposing a "VEC-Trust 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 the 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 [65] 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.

4.3.1 Significance and Related Works

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 [66, 67]. 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 [64] 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 [68] 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.

However, 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 [69, 70]. Among these works, the work in [69] is notable because it is applicable for the efficient allocation of federated resources without considering the trust/reputation of the underlying resources. However, such trust-agnostic resource management is 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 [70] 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 [71] and [72] propose a collaborative trust model based on the Beta distribution [73] and compare their model against random and deterministic resource selection strategies. Other probabilistic approaches such as [74] propose VCC trust by using the priority of tasks and behavioral changes as trustworthiness metrics. In addition, prior works [65] and [73] 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.

4.3.2 Approach

Dirichlet Distribution Background: 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:

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

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

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

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

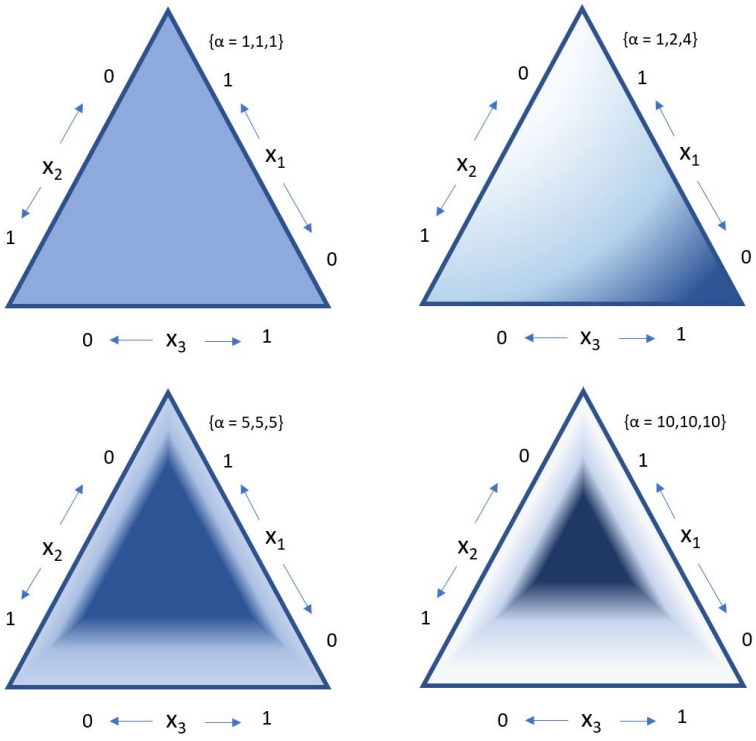


Fig. 4.3 Sample Dirichlet distribution for a system with three parameters $\{x_1, x_2, x_3\}$ at different parameterized values of α. The distribution in the figures shows variation in probability distribution across the parameters as their corresponding α varies.

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 4.3, 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 toward 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 4.3 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 4.3 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 the highest confidence that the machines show equal PD; while α of $\{1, 1, 1\}$ also means 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 the highest probability distribution with a good level of confidence.

Solution Approach: 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 4.4, 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.

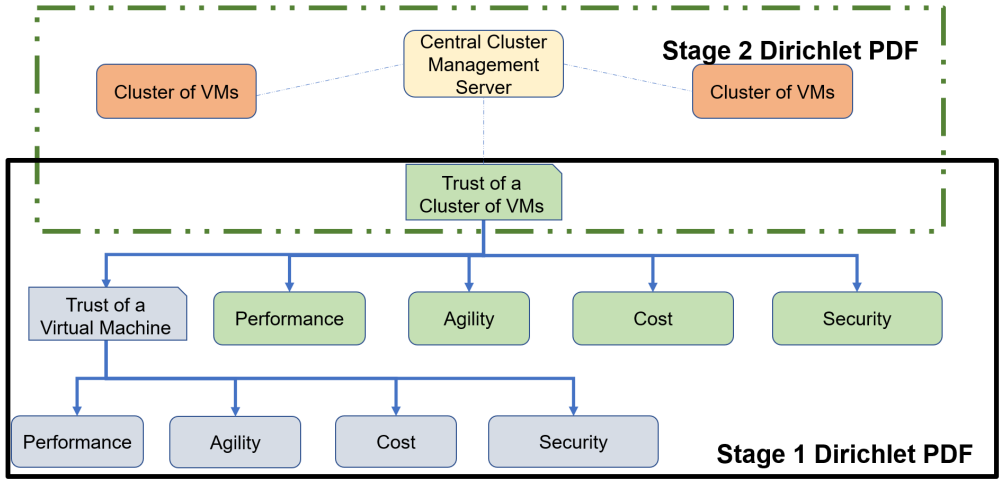


Fig. 4.4 Logical stages of Dirichlet probability distribution applied at intra-cluster and inter-cluster stages. Blocks in black color identify with factors at intra-cluster stage, while blocks in green relate to factors for inter-cluster stage for probabilistic distribution.

The first stage of our proposed VECTrust model 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 [71], 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.

The approach of assessing VEC resources using probabilistic modeling using Dirich-

let 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 the execution of the maximum number of workflows possible. One of our prior works [20] 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 match 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
```

4.3.3 Evaluation

Testbed: 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 4.1. 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 [2] infrastructure and the GCP [4], similar to the architecture described in Figure 4.1. As shown in Figure 4.5, three clusters of resources are created across three geographically different locations (RENCI, NCSU and Texas AandM), 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 shown in Figure 4.5. 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.

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 orders 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 VEC-

Trust 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 [20], and (iii) Unobtrusive utilization-reliability aware scheduling algorithm (U^2Trust) [75], which uses a semi-markov process [76] for reliability-based trust prediction of voluntary nodes and is based on formulating the resource scheduling problem as a knapsack problem.

Table 4.1 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 to conduct the quality control checks on raw sequencing data so that we can remove some low score data before the next step of 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 discovered 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}

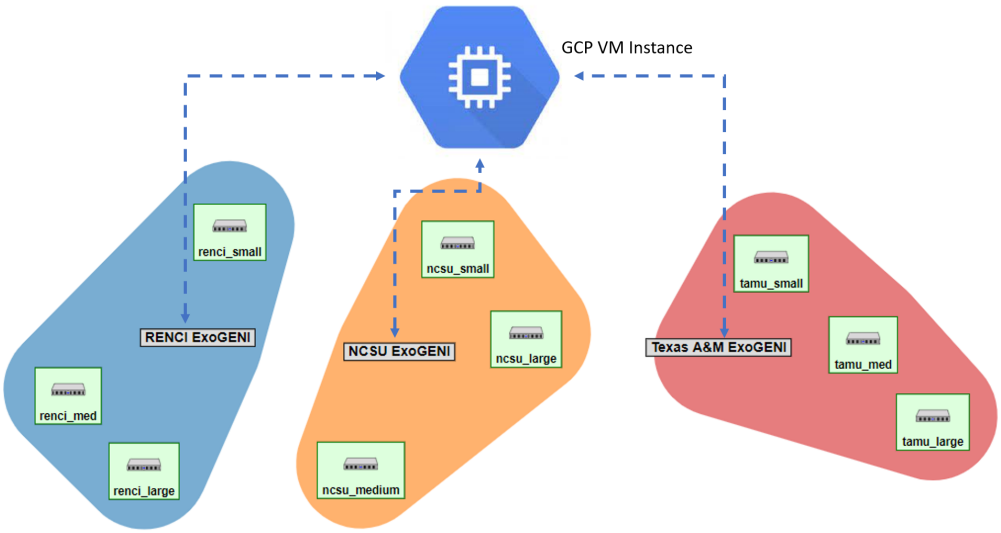
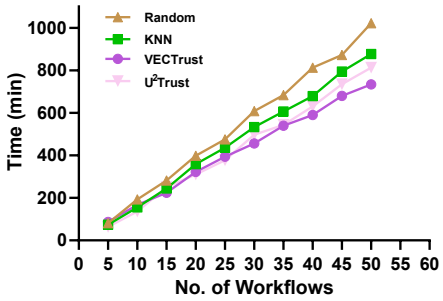


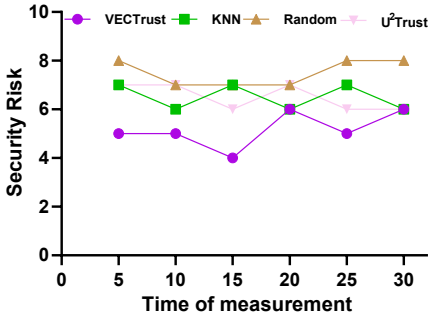
Fig. 4.5 VECTrust testbed using a GCPs' VM instance acting as the CCMS, and GENI nodes acting as smaller remote edge clusters.

For evaluating trusted workflow execution based on performance in VEC resources, we create 12 workflow batches as shown in Figure 4.6 with a different number of work-

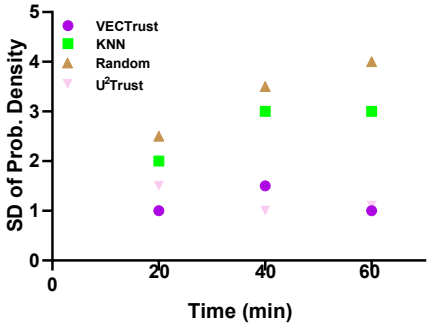
flows mentioned in Table 4.1. Each batch comprises of a set of workflows (e.g., 5, 10, 15, and so on) in a fixed order. This implies that - when a batch is executed, the same workflow is added to the testbed at any time t, irrespective of the scheduling algorithm. This ensures the same arrival pattern 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 4.6(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 resources required compared with resources available in nodes.



(a) Performance comparison



(b) Security risk comparison



(c) Cluster utilization efficiency

Fig. 4.6 (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 the standard deviation of resource utilization PDs for the scenarios: (i) VECTrust, (ii) KNN (iii) Random selection (iv) and U^2Trust .

In order to compare security based on trusted execution between the competing algorithms, we use the NIST guidelines [77] 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 workflows is 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 4.6(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.

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 4.6(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.

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 4.2, 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 the 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.

Table 4.2 Agility comparison between different trust models for different workflows with new VM additions, averaged over 5 repetitions.

Workflow	**VECTrust**	**KNN**	**Random**	***U*^{2}*Trust***	**% 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%

4.4 Learning-based Models for Trusted Resource Selection in VEC

Recent technological improvements have modernized research methodologies. These improvements range from autonomous to automation of data capturing and analytics processes. Often these applications require all the processes to be accomplished locally because of security and delay constraints. To accomplish the goal, edge/fog is a possible middleware between a cloud and a local research environment. Edge nodes provided through VEC architecture can provide processing with acceptable security and latency to robots, sensors, actuators, etc. The applications need to configure various services at the edge nodes to enhance and automate the performance of the system. Our current work [95] considers an important but less investigated service hosting problem, where the edge nodes are dynamically reconfigured through a learning-based reconfigurable security framework to host possibly the most recently requested workflow execution services from the application users. Because of the limited storage and computational resource at the edge nodes along with the dynamic and volatile nature of VEC resources, the problem of reconfiguration is important, which can increase the number and types of workflow tasks hosted by the edge node as well as improve the trust on the VEC architecture. We extend our earlier work on trusted VEC environment [94] by proposing the "VECFlex" framework that employs flexible security reconfiguration, behavioral trust modeling, and scalable scheduling mechanisms. This framework addresses the aforementioned trust, security reconfiguration, and scalability challenges by adopting a

two-tier mechanism that: i) addresses the trust and security configuration issues by applying a Reinforcement Learning (RL), viz., Q-learning driven trust rank classifier that dynamically updates the trust rank of VENs based on their behavior and ii) addresses the scalability issues by applying a security-aware modified Particle Swarm Optimization (PSO) scheduler to allocate the most suitable resources to execute incoming and outstanding tasks.

4.4.1 Significance and Related Works

One of the recent works [85] has used a similar approach of reconfigurable security to create an edge computing framework for IoT devices. The proposed framework is designed to overcome the challenges including high computation costs, low flexibility in key management, and low compatibility in deploying new security algorithms in IoT, especially when adopting advanced cryptographic primitives. A similar work [86] also focuses on reconfiguring edge nodes in industrial settings. The work primarily focuses on scheduling when a sensor node is within the transmission range of multiple fog nodes, to efficiently select the most appropriate fog node for data transmission, different types of fog node selection methods, random selection, shortest estimated latency first, and shortest estimated buffer first, have been considered and evaluated in this paper with satisfactory results. Characterizing the behavior of dynamic environments has been challenging in edge systems. Authors in [96] address the trust issue in mobile edge computing (MEC) environments by using RL-based CPU resource allocation in conjunction with a blockchain-based trust mechanism. Authors in [97] propose an RL-based trust model to observe and learn about the behavior of user nodes within a cluster to optimize the usage of resources. To establish trust established between edge environments with mobile ad-hoc network (MANET) nodes authors in [98] propose a self-adaptive trust-based associative based routing protocol using Q-Learning. The results from these works suggest that applying RL-based modeling on resource behavior could effectively predict the reliability of such resources.

4.4.2 Approach

To improve the trust on VEC resources performance, agility, affordability, and security (PACS) features need to improve so that the user's confidence in volunteer edge nodes increases. This information is analyzed, and if any PACS factors exceed a threshold, we can conclude that the node is unsafe. To decide if a node is safe or not, we plan to utilize a reinforcement model called Q learning. As a result, any node with a particular security configuration at the time of the scheduling of the decision will be rewarded. That reward is determined by several parameters, including whether a node is accessible for executing a workflow and, if not, how long it takes to complete a workflow on average. If a node is under attack i.e., if it takes a longer time for executing workflows, then it will adversely affect the reward value. Other criteria such as the amount of Dos attacks and the performance of a node for a specific workflow are also considered. This model is always being refined, with the reward function values for each node changing with each workflow execution. As a result, it checks for every node that has the same security configuration as a given workflow. Only the node with the greatest reward value for that configuration is assigned the workflow. The potential trust model based on

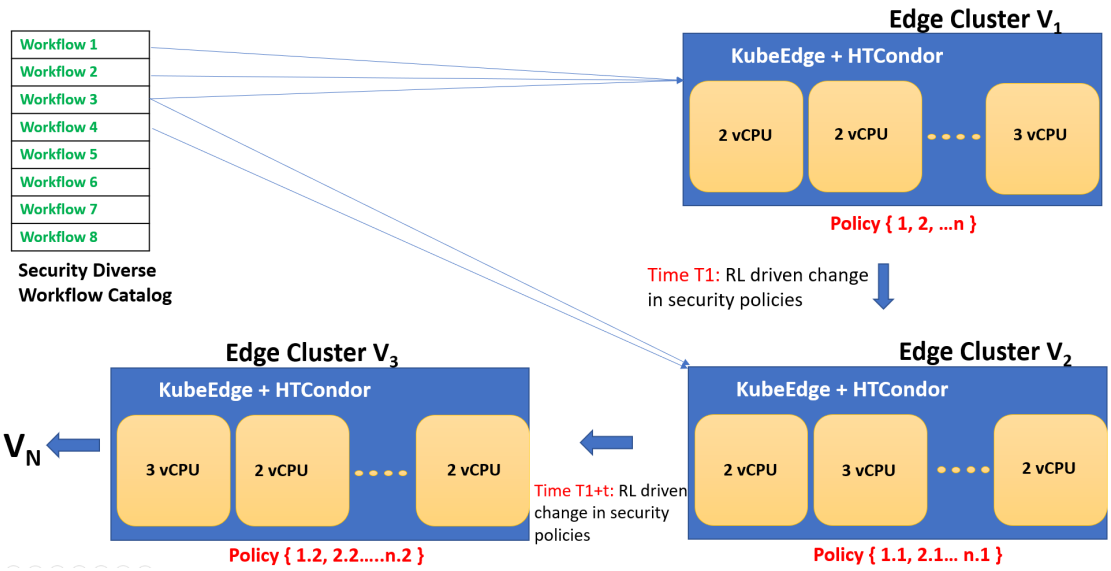


Fig. 4.7 Reconfigurable security on VEC resources based on reinforcement learning model built to improve trust on the resources by improving PACS on the edge resources.

reinforcement learning as shown in Figure. 4.7 will use change in pre-defined security levels of the edge nodes as action space.

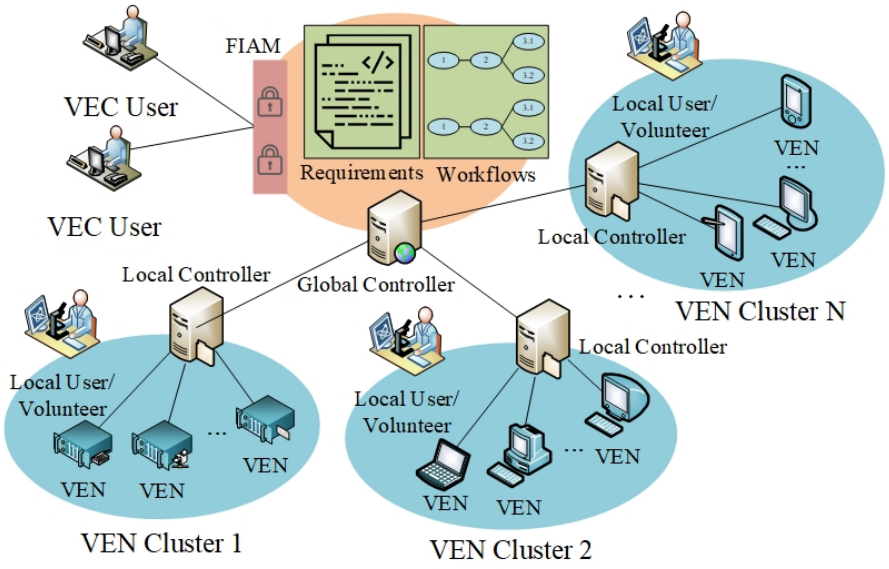


Fig. 4.8 VEC system model with VEC clusters, VENs, and VEC users

4.4.2.1 Behavioral Modeling And Optimization

Figure 4.8 shows a typical VEC system model divided into two parts, i.e., VEC users and VEC infrastructure. As illustrated, the VEC users, i.e., researchers in the need for on-demand and low-cost computation and storage resources are connected to the VEC ecosystem through Federated Identity and Access Management (FIAM) where they submit their pool of workflows and corresponding performance and security requirements. Meanwhile, the VEC infrastructure consists of volunteer edge clusters; each cluster consists of a supernode or local controller and multiple VENs for processing workflows. These VENs can vary from rack servers to desktop computers, to IoT devices based on the contributing researchers' (i.e., volunteers') lab hardware.

Resource behavioral analysis model:

Our behavioral model uses the historical availability of resources and a set of minimal required security policies to decide an initial trustworthiness rank of the VENs.The initial "Assigned Trust Rank" indicated in the *Resource Parameters Table* is based merely on an offline evaluation of resources and expected performance. With the initial trust rank, the current status of resources, and expected performance, an on-demand analysis is applied by the RL-driven agent to evaluate the reliability of the node based on previous results under similar conditions. The target of the RL-based approach is to measure the consistency of the VENs on executing previous tasks and then to determine the ex-

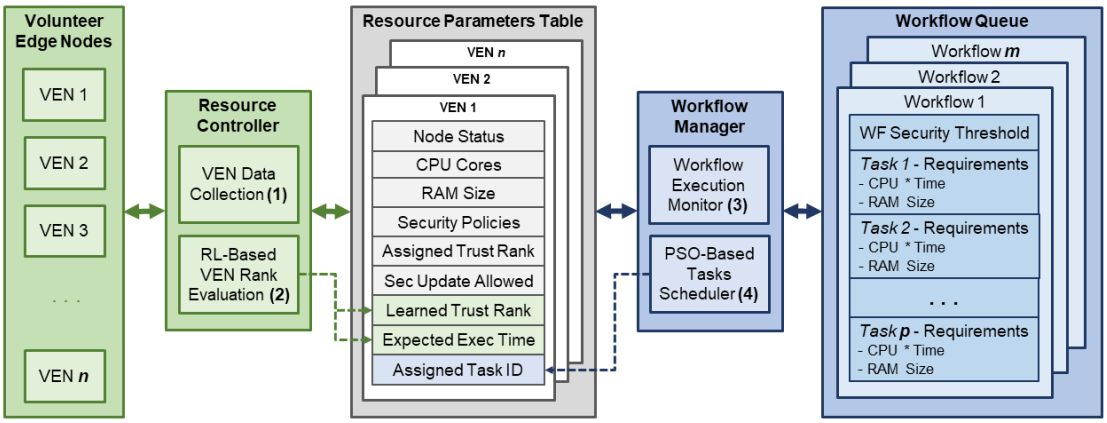


Fig. 4.9 VEC Architecture: The Resource Controller implements the RL-based resource behavioral analysis model with modules (1), and (2). The Workflow Manager implements the PSO-based workflow scheduling model with modules (3) and (4).

pected execution time of the next task. Identifying VENs whose execution times are consistent with the predicted execution times gives a level of certainty to those nodes on coming task assignments, and a new "Learned Trust Rank" is then determined. Using the most trusted nodes, determined by our RL approach, will make good use of resources and hence minimize the overall execution time. Our proposed RL-driven approach uses a Q-Learning model with a learning agent

Our Q-learning-based behavioral analysis follows the process executed by modules (1) and (2) under 'Recourse Controller' as described in Figure 4.9. In the initialization phase, VEN information is collected by the global controller (through local controllers), and an initial trust rank is assigned based on the number of CPUs, RAM Size, and the number of security policies configured. All parameters are stored in the *Resources Parameters Table*, which represents the state space of the VEC environment. Then, any update on the state space triggers a new evaluation of the Q value for all idle VEN nodes, which is a measure of the level of trust for each node. Busy VEN nodes are not updated as a new Q value makes sense only when the nodes have completed a task and an actual execution time spent on processing the last task has been provided. The new Q values calculated for the relevant VENs define the action to update the trust rank of each node. The learned rank will be used by our PSO-based optimization to perform resource allocation and task scheduling, as explained by Algo. 2.

Workflow scheduling model:

Algorithm 2: VEN rank update based on the learnt Q-value

Input: Table $state\_parameters$, each record contains the parameters for each VEN
Output: Table $state\_parameters$ with the $learned\_rank$ field updated based on the result of the learned Q-value
for *each idle node* n *in* $state\_parameters$ **do**
    Calculate actual execution time X_n^a expended on the last task;
    Calculate the variance between the predicted execution time X_n^p and the actual execution time X_n^a;
    Calculate the reward value R_n;
    Calculate the new Q-value Update the $learned\_rank$ field for node n in the $state\_parameters$ table;
return $state\_parameters$;

For workflow scheduling, we use a security-aware modified PSO algorithm, and follow the process executed by modules (3) and (4) as described in Figure 4.9. PSO is a heuristic algorithm that improves the optimal solution within its large population (called a swarm) of candidate solutions (called particles) over time. The biggest advantage of PSO is that it can help find the near-optimal solution for a problem within a large solution space. In our problem, we have a large swarm of workflows and a large pool of volunteer nodes; the objective is to find the best node/VEN (position) for each workflow (particle). Each workflow is an object that includes CPU requirement (cpu), memory requirement (mem), security requirement (sec), best-known position ($best$), and velocity (vel) of the swarm's members; best-known position and velocity will be initialized in a uniformly distributed random manner beforehand. Meanwhile, the object consists of CPU availability (cpu), memory availability (mem), security level (sec), security flag (sec_flag, a Boolean variable indicating whether that node can adjust its security level or not), and node's rank ($rank$, based on the outcome of the RL-driven behavioral modeling algorithm and used for a final decision if there are multiple satisfied nodes for one workflow). These member variables will help define the $cost$ function of assigning a task to a particular VEN within a VEC cluster for the main PSO algorithm. Firstly, we need to check if the node's security level satisfies the workflow's minimum set of security requirements or if the node can adjust/reconfigure its security level to match that requirement. Next, it will check if the VEN has enough resources to process the workflow by comparing the node's CPU and memory availability with the workflow's CPU and memory requirements.

In the proposed security-aware modified PSO algorithm (Algo. 3), we scan the entire workflow space and VEN space to compute the optimal node for all workflows. For each workflow, we update its velocity based on the current velocity, current posi-

Algorithm 3: Security-aware modified PSO

Input: Workflow list $w\_list = \{w_i\}$ and VEN list $n\_list = \{n_j\}$
Output: A matrix $node\_mat$, each row is a list of near-optimal nodes for each workflow
$weight = 0.3$; $cp = 1.5$; $cg = 1.3$;
$node\_mat = []$;
for $i \in [0, len(w\_list))$ **do**
    $node\_list = []$;
    $curr\_pos = 0$;
    for $j \in [0, len(n\_list))$ **do**
        $rp = rand(0, 1)$; $rg = rand(0, 1)$;
        $w\_list[i].vel =$
          $weight * w\_list[i].vel + cp * rp * (w\_list[i].best - curr\_pos) + cg * rg * (swarm\_best - curr\_pos)$;
        $curr\_pos \mathrel{+}= w\_list[i].vel$;
    if $n\_list[curr\_pos]$ *is available or almost available* **then**
        if *cost*$(w\_list[i], n\_list[curr\_pos]) >$ *cost*$(w\_list[i], n\_list[w\_list[i].best])$ **then**
            $w\_list[i].best = curr\_pos$;
        if *cost*$(w\_list[i], n\_list[swarm\_best]) <$ *cost*$(w\_list[i], n\_list[w\_list[i].best])$ **then**
            $swarm\_best = w\_list[i].best$;
        $node\_list.append(curr\_pos)$;
    $node\_mat.append(node\_list)$;
return $node\_mat$;

tion, best known node of that workflow, known swarm's best, and parameters including rp, rg, cp, cg and $weight$. Parameters rp and rg are uniformly distributed random variables between $[0, 1]$. Parameters cp (cognitive coefficient) and cg (social coefficient) are acceleration coefficients and are selected based on each scenario with typical values ranging between $[1, 3]$. The parameter $weight$ is also selected based on each design but must be smaller than 1 to prevent divergence. Overall, the selection of cp, cg, and $weight$ is usually based on multiple runs for the best estimation. After the new velocity is calculated, a new position of the workflow is updated based on the new velocity. Moreover, we need to make sure that the new position (i.e., VEN) is available to process the workflow. Upon that, we start updating the workflow's best position and swarm's best if this new position is more optimal and append the feasible positions to a list. When the inner for loop is completed, we append the feasible positions list to the matrix. Finally, a matrix that contains the lists of feasible positions for each workflow is returned. Based on this matrix and the RL-based rank of nodes (calculated in Algo. 2), we can choose the most optimal node for each workflow.

The overall resource optimization follows the process described in Figure 4.10. The steps in green correspond to the behavioral analysis process and the steps in blue correspond to the modified PSO-based task scheduling process.

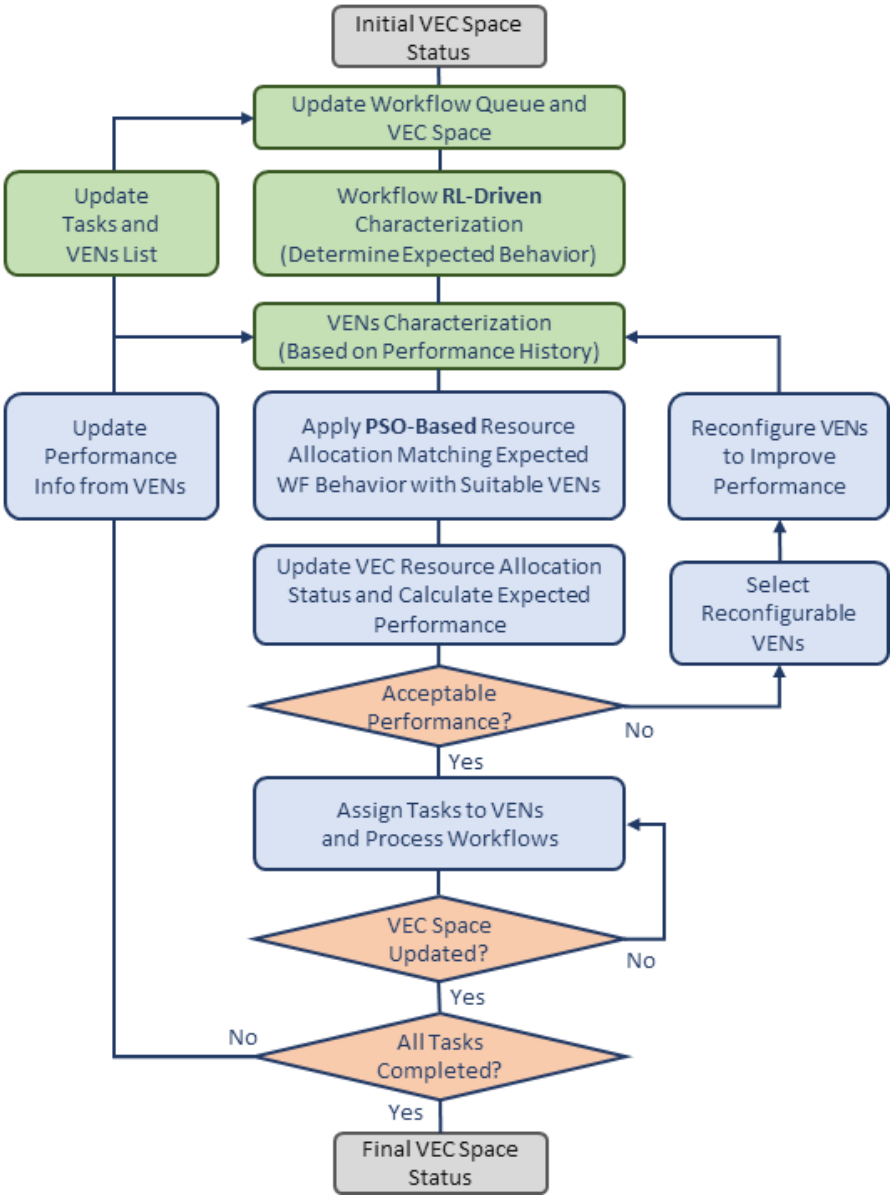


Fig. 4.10 Overall resource optimization process flowchart

4.4.3 Evaluation

We discuss below the testbed design, implementation, experiment setup, and results to evaluate the performance of our proposed VEC*Flex* framework.

4.4.3.1 Testbed setup and VEC*Flex* implementation

For VEC*Flex* framework evaluation, we set up a virtual VEC testbed topology using Amazon Web Service (AWS) [1] followed the model presented in Fig. 4.8. We create a virtual machine (VM) to be configured as the VEC global controller that hosts all work-

flows, related data, tasks, and their requirements. The global controller site communicates with individual VEC cluster sites via public IP addresses. The global controller also hosts the RL-driven behavioral modeling and modified PSO-based scheduling algorithms (i.e., Algo. 2 and Algo. 3). The algorithms are run with data collected from local controllers belonging to individual VEC cluster sites. To simulate the VEC cluster sites, we deploy 10 VMs. Each of these VMs also play roles of local controller and reports directly to the global controller with necessary information. On each VM, we run 5 Docker containers to mimic the VENs. Thus, overall the testbed setup simulates 10 VEC cluster sites, each with 5 containers as VENs for a total of 50 VENs. The resource configuration such as memory and processor speed assigned to each VEN is different to mimic a heterogeneous resource environment.

4.4.3.2 Evaluating the impact of PSO and RL

Here we assess the impact of our proposed modified PSO and RL-driven methods by demonstrating their improvement over Random selection. First, to illustrate the effectiveness of the proposed PSO algorithm in handling workflow scheduling, we compare the performance of our proposed VEC*Flex* which combines PSO-based resource selection with RL-driven modeling (RL+PSO) against a Random resource selection scheme with RL-driven behavioral modeling (RL+Random). The comparison is carried out for four different scalability scenarios in terms of the number of workflows. The metric we use for this evaluation is 'Average workflow processing time'. Here, the processing time of a workflow depends on the allocated VEN resources, e.g., for a workflow that requires 2 CPUs and 2 Gb of memory, a VEN resource of 2 CPUs (up to 3.3 GHz) and 8 GB of memory can process in 2 minutes 11 seconds while a VEN resource of 1 CPU and 1 GB memory takes 4 minutes 34 seconds to process. As mentioned before, the VENs are assigned different amounts of resources to mimic a heterogeneous environment. We run the experiment 15 times with a different set of workflows and VENs to calculate the average processing time. As shown in Fig. 4.11, we observe that for different workflow scenarios, VEC*Flex* outperforms the RL+Random scheme by allocating close to optimal resources from VENs that help quicker processing of the workflows.

For evaluating the utility of RL-driven behavioral modeling, we conduct a similar comparison of our proposed VEC*Flex* scheme against only the Security-driven scheme (i.e., without any behavioral modeling) for four different scalability scenarios in terms of the number of VENs. As RL updates the rank of the VENs periodically based on the historical behavior of the node and then selects the nodes with the highest rank, we want to identify if by using RL we will have more VENs available than only the Security-driven scheme. As for the only Security-driven scheme, it selects the top VENs with the highest number of security policies enabled regardless of the CPU and memory. For each case, we determine the total amount of CPU and memory resources that the selected VENs provide, i.e., we determine the total computation power that each option provides. The idea is to determine how effective is VEC*Flex* in selecting the required top VENs with the highest computation capacity while keeping the environment safe. For this evaluation, we run the experiment 15 times and plot the mean value. As illustrated in Fig. 4.11, VEC*Flex* is more effective in selecting the best VENs, with the highest computation capacity and satisfactory security, in comparison with the just security-based approach.

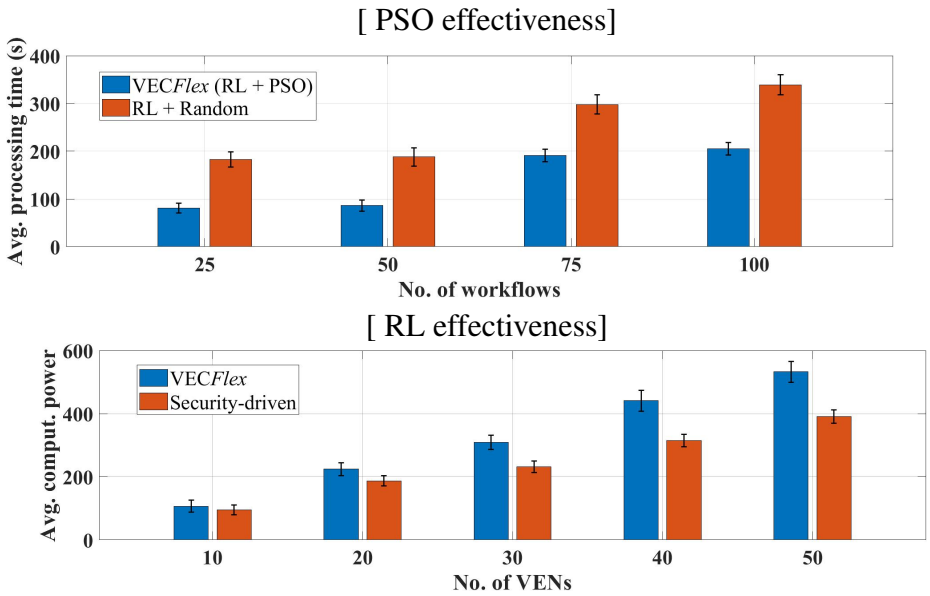


Fig. 4.11 Evaluation of the impact of modified PSO and RL-driven approaches

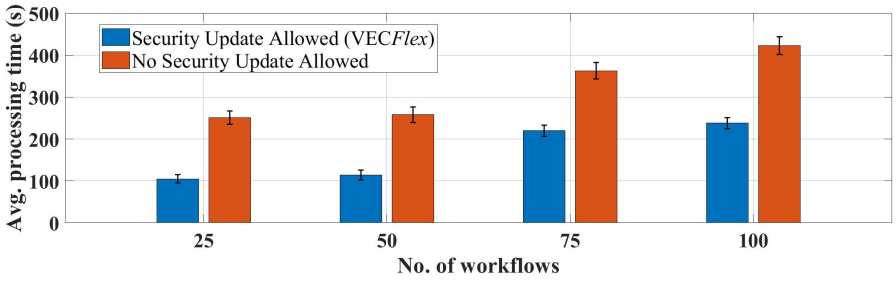


Fig. 4.12 Security update allowance evaluation

4.4.3.3 Security Reconfigurability

In order to satisfy the minimum security requirements of tasks during scheduling, a task can only be executed on a VEN that has the minimum set of security policies implemented. As a solution to this problem, our approach relies on VENs that allow security policies to be reconfigured. In our testbed implementation that can happen only if that VEN has the 'Security Update Allowed?' field of the VEN set to 'YES'. If so, we change the 'Security Policies' parameter to meet the needs of the workflow. In this final evaluation, we disable that option in order to stop security reconfiguration. This triggers the scheduling mechanism to suspend task scheduling until VENs with the minimal required security policies become available. As can be seen from Fig. 4.12, with this modified setting, the scheduling takes on average $1.5 - 2$x the processing time for the workflows to finish (in comparison to VEC*Flex*), hence demonstrating the utility of VEC*Flex*'s proposed security reconfiguration.

4.5 Summary

In this chapter, we have built upon the last chapter on multi-cloud resource selection and have further proposed a novel VEC architecture for computing. This architecture focuses on improving the PACS for the workflow applications which helps in reducing cost and improves performance for applications. The proposed solution aims to utilize any available edge resource for the execution of any workflow, we further delineate the fundamental trust problems arising from using any edge resources for the execution of the workflow and suggest solutions to improve trust in edge resources. With a better trust model in place, the proposed solution allows using of methodologies of utilizing edge resources more effectively which has been the main bottleneck in their widespread adoption. We have also shown the application of reinforcement learning algorithms in the assessment and improvement of trust in the context of PACS factors in VEC resources by characterizing volunteered resources and their dynamic behavior to workloads. The initial results show significant improvement in trust factors of performance, agility, cost, and security while utilizing the VEC cloud.

CHAPTER 5

VEC Resource Brokering Guided Applications

5.1 Introduction

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. 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". These VEC resources in collaboration with cloud platforms can be potential cyber resources that could reduce the cost as well as potentially improve performance for these applications by reducing the latency of data transfers. Below we discuss case studies of applications that can use VEC architecture for workflow execution and data analytics pipelines.

5.2 Manufacturing: Carbon Nanotube Case Study

Experimental research such as cell cultures and carbon nanotube (CNT) growth are largely governed by following predefined execution protocols with fine-tuned control of parameters. There are promising opportunities to apply reinforcement learning (RL), which is an established learning technique in the area of general artificial intelligence, in order to automate the CNT growth process and accelerate related scientific breakthroughs in material discovery. Although there are RL techniques' strengths in exploration and exploitation methodologies, there are challenges in developing relevant

learning policies in experimental research settings. In this paper, we present a novel data-driven reinforcement learning solution for assisting CNT growth. Our approach is focused on developing RL models to learn the characteristics of CNT growth temporally while varying parameters that affect growth. Enabling automation through our RL model in CNT growth experiments allows for exploring a wider range of growth conditions. The ultimate goal of our RL model is to reach desired CNT growth while dynamically controlling growth parameters throughout a sequence of experiments.

5.2.1 Significance and Related Works

CNT growth under electron microscopy by different groups suggests that the mechanism is extremely sensitive to each parameter such as carbon precursor, metal catalyst, particle size, temperature, and pressure. Even a minor change in any of these parameters leads the growth in critically different directions. Since there are so many variables involved in synthesizing CNT forests reliably, it is our belief that automating the process of CNT growth would help constrain them. Not only would automating help provide reliable and reproducible growth by controlling variables it would also reduce the chance of human errors

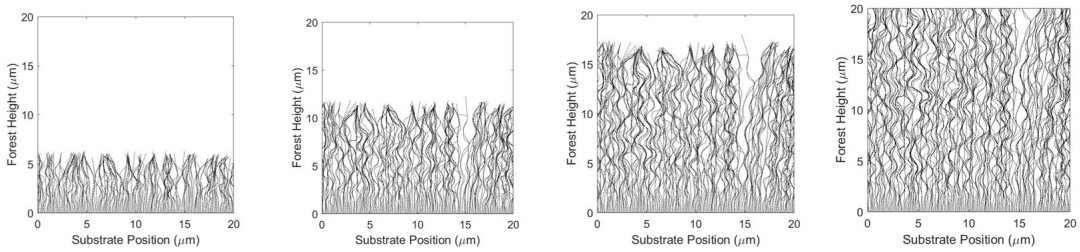
Researchers have recently constructed a high-throughput, autonomous research robot to synthesize isolated single walled nanotubes (SWNTs), to efficiently navigate within a high-dimensional synthesis space. The Autonomous Research Systems (ARES) uses a Raman spectrometer laser to provide heat and simultaneously characterize growing SWNTs in situ [78], [79]. The ARES system utilizes machine learning algorithms to learn from previous experimental results and uses a planner to conduct experiments to reach an objective. In one demonstration, the ARES system autonomously determined experimental parameter sets required to achieve specified SWNT growth rates in fewer than 100 iterations [79]. The ARES approach shows that integrating artificial intelligence, with high-throughput experimentation can allow experimental parameter space navigation within a process–structure domain that is not well understood by researchers. The interactions between large populations of concurrently growing CNTs in forests add complexity, which is not present during the synthesis of isolated

SWNTs. In this study, a time-resolved finite element method (FEM) CNT forest simulation tool [80], [81] is used as a high-throughput virtual laboratory to examine the synthesis–structure–property design loop of CNT forests. Images of each CNT forest morphology were obtained at the end of their simulated synthesis. A mechanical compression simulation was used to obtain mechanical properties [81], [82], [83]. One of the recent works [84] has shown an exemplary study where RL was leveraged. In the applied approach authors have trained RL models in simulation and then implemented it successfully into the physical world scenarios. Data for RL is collected via running an agent in the desired environment, but for applications like robotics, running a robot in the real world may be extremely costly and time-consuming. In this paper, authors have introduced the RL-scene consistency loss for image translation, which ensures that the translation operation is invariant with respect to the Q-values associated with the image. This allows the model to learn a task-aware translation. Incorporating this loss into unsupervised domain translation, the RL-CycleGAN which is based of the deep neural network, a new approach for simulation-to-real-world transfer for reinforcement learning was proposed.

5.2.2 Approach

Experimental data often takes days to be produced and relies on the skill of the operator to produce SEM images with high detail and low noise, this often causes a deficit of physical CNT forest experimental data. To supplement this lack of data, a physics-based finite-element simulation is used to obtain CNT forest images at different stages of growth. Simulation parameters such as density, plot area, and standard deviation of rate and angle will be used for the study, these parameters can be modified to produce images analogous to physical CNT forest growths. A test agent will be validated using the synthetic data from the simulation and will then be applied to the experimental data to test the CNT forests' physical properties. Another simulation was developed to measure the compressive strength of individual CNTs and is used to obtain physical properties that can be associated with the synthetic images being generated by the. Figure. 5.1 shows the images representing the sequential growth of the CNTs being

developed through our simulation framework.



(a) Initial growth **(b)** Half of the growth **(c)** CNT growth last stage **(d)** Final CNT growth

Fig. 5.1 Our simulated CNT forest growths as shown at different stages of their growth (a) growth pattern at height of 5uM, (b) growth pattern at height of 10uM, (c) growth pattern at height of 15uM, (d) growth pattern at height of 20uM

Drawing inspiration from the recent works with reinforcement learning which has the potential to create end-to-end autonomy. We have developed first of it's kind smart RL agent to regulate CNT growth by regulating synthesis parameters so as to achieve a growth that maps to a desired mechanical property. Through a simulation-based study, our RL agent helps to overcome difficulty in controlling parameters of growth in real experimental setup as well as reduces manual time thus enabling faster and better sequential growth of CNT tubes with regulated parameters for desired properties. In our proposed RL model the agent learns from CNT growth variation in a simulation-based environment where the critical parameters of density, growth rate, tube radius, tube stiffness as well as Van der Waals forces are used as control parameters. Enabling automation through our RL model in CNT growth experiments allows for exploring a wider range of growth conditions. We have specifically used two kinds of actions to control simulated CNT growths which are a)change in standard deviation in angular deviation of CNT tubes while they are growing and, b) change in the rate of growth of tubes. Consequently, two models are generated based on these action controls. However, the goal of both agents is to learn optimal growth policies to improve the maximum compression load capacity of the CNT tubes. A simulation-based learning environment is necessary to utilize RL because all learning-based methods need,

- large iterations of similar and dis-similar situations to learn.
- it is time and resource-consuming to perform CNT growth experiments for all

permutations of the value of controlling parameters.

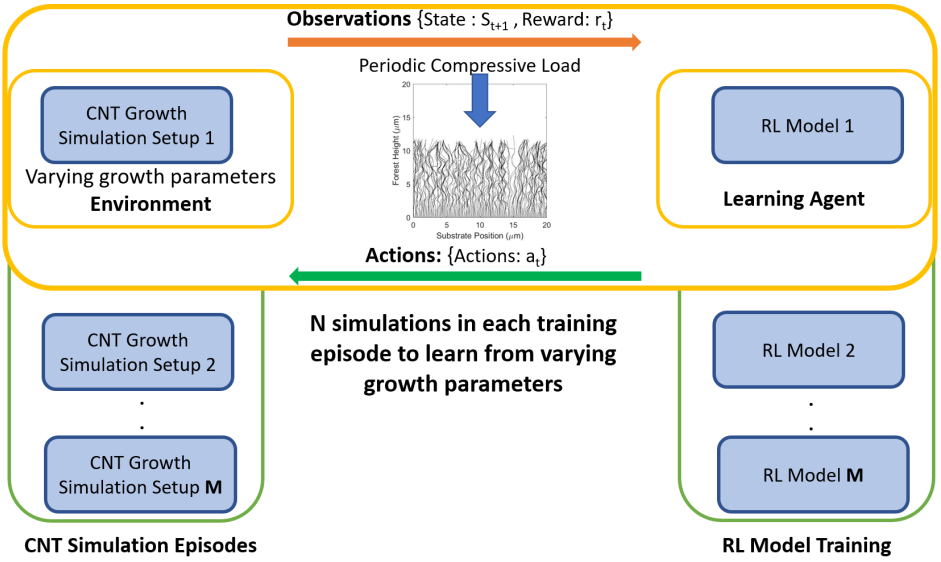


Fig. 5.2 RL agent training methodology followed in simulated growth of CNT tubes in episodes. Each episode contains N=30 simulation growth. A total of 10 episodes was used for learning the growth policies. At each iteration, the tubes are compressed and a reward is generated as feedback for the model to learn optimum actions. The model aims to increase the overall reward in the training.

More specifically the goal of our learning base solution is to create a smart learning agent which can dynamically regulate parameters affecting CNT growth to improve its ability to take the compressive load. We detail below the definition of the key sub-components which are used to create our RL model.

- **Agent:** The 'q' in q-learning stands for quality. Quality in this case represents how useful a given action is in gaining some future reward. When q-learning is performed we create what's called a q-table or matrix that follows the shape of [state, action] and we initialize our values to zero. We then update and store our q-values after an episode. This q-table becomes a reference table for our agent to select the best action based on the q-value. It does not require a model of the environment (hence "model-free"), and it can handle problems with stochastic transitions and rewards without requiring adaptations.

- **Actions:** We have developed two different models based on the actions performed on the growth environment to characterize CNT compression. These actions are i) regulating the waviness of the tube getting generated. That is the average angular deviation of the tubes from straight growth. It is essentially the degree of

tortuousness in the CNT tubes. ii) The average rate of tube growth which can lead to adding defects in the tube growth due to the breaking of the tubes.

- **Observations:** Total maximum compression load that can be withstood by tubes at different heights of compression is measured and compared to assess improvement in the load capacity of the grown tubes. These observations are used to generate suitable rewards for the model at different steps of measurement.

- **Reward:** The agent is rewarded for the selection of growth parameters which leads to improvement of the maximum capacity of compression load of the CNTs and it is penalized when the compression load capacity decreases after a certain parameter of growth is applied in the growth environment. The action applied at each step is governed by the assessed reward measured by the model.

5.2.3 Evaluation

In this section, we evaluate our RL model in it's ability to regulate growth parameters to improve the maximum compression load capacity of the CNT tubes.

Evaluation of model based on angular deviation: Our Q-learning-based RL agent was able to learn the growth parameter "wave" temporally to create CNT tubes with the ability to withstand more compressive forces. The maximum compressive load of 3.59e-05 N was achieved at the "wave" parameter configuration of [3,4,5] in steps of growth within 1-33, 33-66, and 66-100 steps when the CNTs were compressed to 95% of their initial height. As shown in Table 5.1 maximum compression load is observed at 90% of the maximum height of the tubes.

Table 5.1 Comparison of maximum load capacity of CNTs at different height with and without RL model when **based on angular deviations** where H is the initial height of tubes before compression

Height after Compression	**Max. Load without model**	**Max. Load with RL model**	**Optimal 'wave' Configuration**
95% of H	01.43e-05	3.59e-05	[3,4,5]
90% of H	4.61e-05	6.43e-05	[4,4,5]
80% of H	2.1e-05	4.2e-05	[3,4,5]

Evaluation of model based on the rate of growth: The maximum compressive load of 6.98e-05 N was achieved at the growth rate parameter configuration of 65e-

9 meters/sec when compression was done to 90% of tubes initial height. The agent improved the rate of growth parameter from 60e-9 to 65e-9 after the final iteration when the angular "wave" parameter was fixed at the value of 3. The rate of growth parameter theoretically can take any value within simulation experiments but the agent suggests values of 65e-9 m/sec for growth which improves compression load capacity. As shown in Table 5.2 the average improvement in the load capacity increased by 183% at different heights of compression.

Table 5.2 Comparison of maximum load capacity of CNTs at different heights with and without RL model where H is the initial height of tubes before compression

Height after Compression	**Max. Load without model**	**Max. Load with RL model**	**Optimal Average Growth Rate**
95% of H	2.56e-05	4.71e-05	65 e-9 m/sec
90% of H	3.58e-05	6.98e-05	65 e-9 m/sec
80% of H	2.81e-05	4.86e-05	60 e-9 m/sec

5.2.4 Manufacturing: Automation pipeline on VEC

For experimental research such as material discovery, a large number of parameters govern the outcome of the experiments. These parameters are tough to control and convolute the final result. In our exemplar study of CNT growths which also have a large number of parameters affecting its growth, essentially NextGen instruments such as for imaging are utilized. With these new generations of instrumentation, it is often difficult to predict the types of data to be collected tailored toward understanding the experimental problem at hand. These instruments are characterized by the generation of large volumes of data which often needs more processing. Regardless, they are in high demand and are available for a limited time.

Hence it becomes critical to use them as efficiently as possible. These requirements necessitate development of an intelligent algorithm and cohesive data pipelines to generate insights faster and correctly as shown in Figure 5.3 Through our current work, we have shown the utility of reinforcement learning for predicting and controlling the growth of CNT tubes with the goal to move toward a fully autonomous intelligent system that can grow CNTs of desired properties by utilizing image data in synergy with

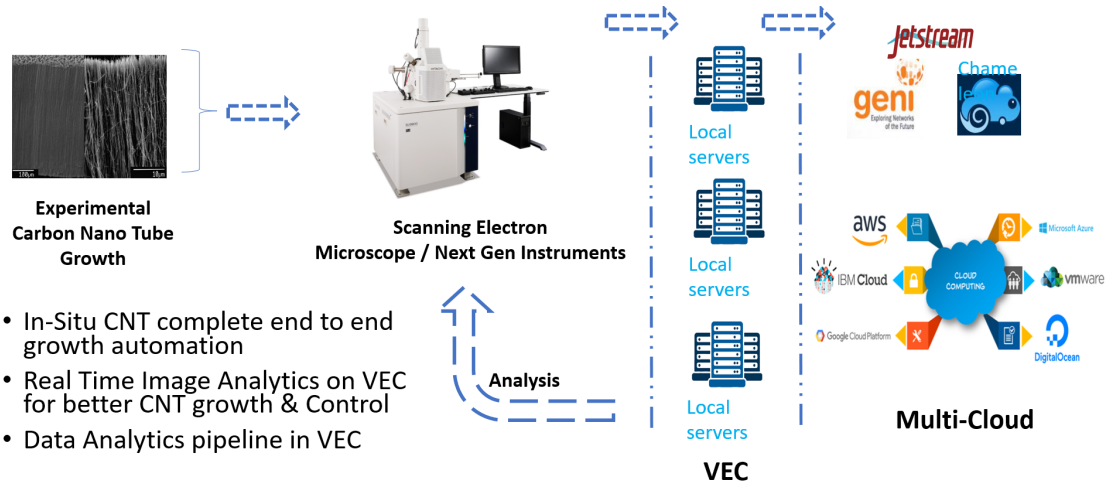


Fig. 5.3 Data analytics pipeline leveraging the VEC architecture and multi-cloud platform for analytics of the raw CNT image data generated from next generation of scientific instruments

other governing control parameters. Note that such fully autonomous system will require extensive data collection and potentially large-scale cloud computing to process huge data quicker so as to make an informed decision on time-sensitive CNT growth and we estimate that these kinds of workflows processes other experimental researchers as well.

5.3 Healthcare: Protected Data Analytics Case Study

Accessing massive collections of prior medical literature and handling the on-going data deluge creates challenges for healthcare data consumers (e.g., clinicians and researchers) who need to make timely data-driven decisions related to the COVID-19 pandemic response. The current practice still heavily relies on time-consuming and onerous manual methods to search, compile and select the articles that are relevant for gaining insights to shape outcomes. The COVID-19 pandemic demands swift actions from researchers and clinicians, and there is a dire need for robust tools to help them manage the data sets in research tasks, and also to enable them to collaborate with other experts based on critical evidence. The tools also need to be integrated within unified data-sharing platforms that increase accessibility to specialized literature and support data analytics automation to expedite e.g., search and analysis processes. Even more importantly, the tools need to be accessible in a flexible and scalable manner by utilizing cloud-based deployments with necessary interfaces to integrate open-source tools

and healthcare social networks.

5.3.1 Significance and Related Works

The OHDSI program is committed to promote the importance of health data analytics through the development and release of open-source data analytics tools (i.e., ATLAS, ACHILLES, ATHENA) [101]. These tools have common features which allow them to interact with a CDM [102] that can be implemented using multiple database management systems (e.g., Postgresql, Redshift). Through proper extraction, transformation, and loading (ETL) processes, disparate structured and unstructured data sources can be integrated into the CDM repository under a well-defined data structure that will allow the analytic tools to utilize templates to run standardized data analytic processes and generate insightful results. Multiple solutions have been developed to store and share health-care data in cloud environments, keep those records secure in such environments, provide analytic services related to health big data, and preserve data privacy. In the context of data accessibility, the work in Health-care Data Gateway [103] aims to securely store Electronic Health Record (EHR) data in a cloud-based platform and uses a Blockchain-based secure storage layer. Data sharing is supported among multiple users (i.e., physicians, researchers, government institutions, private organizations) based on role assignments. Similarly, in their work, [104] proposed a system to store EHR in a public cloud, and their focus was on ensuring data confidentiality and integrity by using an access control mechanism based on the lattice model

5.3.2 Approach

The core component of the data pipeline orchestration in OnTimeEvidence is built on top of the open-source OHDSI on AWS. The AWS CloudFormation is used to deploy OnTimeEvidence along with the data access and process management module, entitlement database, and access admin console. We leverage the JupyterLab included with OHDSI on AWS to facilitate users' data access and interaction, and create extensions such as the user data request forms and the data processing models in order to provide the analytic workspace for health-care data and COVID19 publications analysis

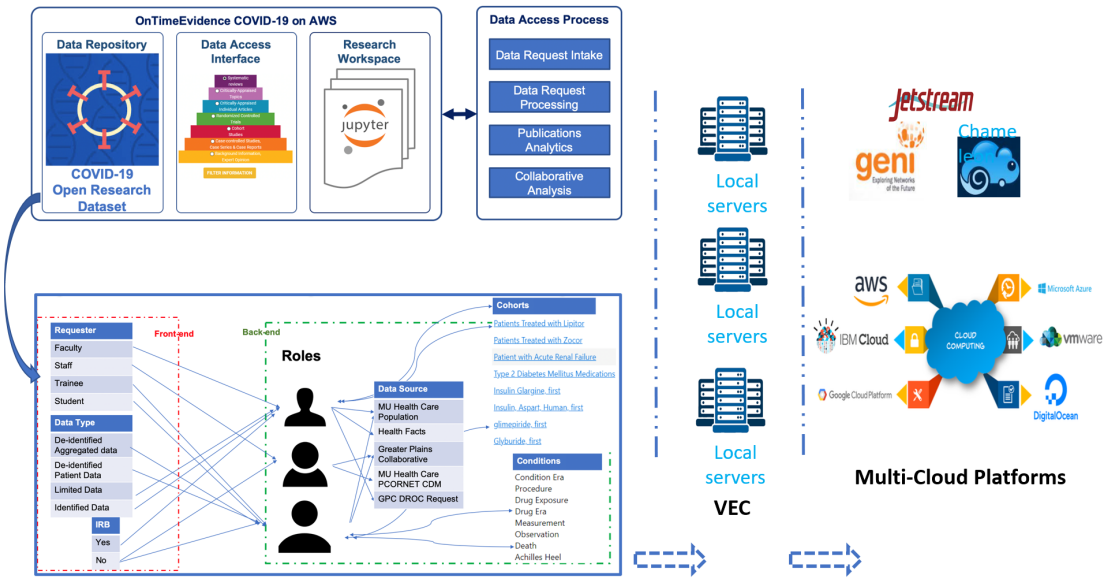


Fig. 5.4 COVID-19 literature selection and analysis workflow process in OnTimeEvidence for knowledge discovery leveraging the VEC architecture and multi-cloud platform.

as well as result sharing. We have uploaded the SynPUF Medicare and the CORD-19 datasets to a relational repository on the OHDSI Red-shift data warehouse service, and the related health-care data and COVID-19-related literature information are available for process testing and validation of user utility.

To facilitate these tools into a unified unit we have developed a data discovery platform viz-a-viz "OnTimeEvidence" as shown in Figure. 5.4. The platforms was developed with the capability to enhance scalability. However, extensive use of cloud computing resources is not economically feasible. For such platforms which need to facilitate smaller applications at a higher frequency and large applications intermittently. A combination of edge and cloud services which is the promise of VEC is the most optimal computing architecture. VEC architecture in these platforms ensures cheaper solutions and reduced maintenance costs while at the same time enabling applications to maintain performance by leveraging multi-cloud platforms if required.

5.4 Summary

In this chapter, we discuss potential applications which can benefit from VEC architecture of computing. Specifically, we discuss the manufacturing use case of carbon nanotube (CNT) growth and ways to improve the CNT growth by learning its optimal growth parameters through a RL model. We further plan on integrating the RL solution and new data analytics pipelines and workflows with VEC clusters and multi-cloud platforms. Such integration will act as an exemplar use case that will guide and motivate the integration of more data analytics pipelines to adapt the VEC architecture of computing. We also describe the utility of VEC architecture in a healthcare data analytics pipeline i.e., "OnTimeEvidence" where smaller analytics and queries can be handled through voluntary edge resources while the larger compute-intensive tasks can be managed through cloud platforms. Moreover, VEC computing architecture can find its utility in many scientific as well as commercial applications such as IoT devices, since it focuses on the economical availability of computing resources. Rapid availability of these volunteered resources is critical for financially constrained projects and users.

CHAPTER 6

Conclusions and Future Works

In this thesis, we have presented a novel solution for improving cloud resource allocation for workflows and applications. We solve the problem of provisioning optimal cyber resources for users by diving the large problem into smaller problems that can be solved sequentially toward the final solution. More specifically we presented a novel multi-cloud resource recommender that focuses on improving PACS criteria for resource selection and allocation. We have also presented a novel architecture of computing i.e. VEC which aims at utilizing all available computing resources for all kinds of workflows and applications.

6.1 Contributions Summary

User Engagement: For the ease of non-expert cloud users, who often struggle to deploy cyberinfrastructure in an efficient manner for data analytics and to gain from the experiences of cyber expert users, we developed intuitive GUIs and structure [87][88] to capture users requirements and expertise effectively within KbCommons portal [18].
User Preferences: To guide non-expert users with resource allocation for application workflows, we deploy a fuzzy engineering model and utilize users' expertise along with a knowledge base of benchmarks rules for assessing PACS criteria of CSPs [89][90].
PACS Optimization: We propose a novel multi-objective (performance, agility, cost and security) optimization model integrated within a novel resource brokering middleware viz., OnTimeURB in order to meet KBCommons biological user-defined constraints of bioinformatics application workflow performance, cost and CSPs interoperabilities [91][92].

ML based Resource Allocation: We present a novel multi-objective optimization algorithm that powers a multi-cloud resource broker middleware viz., OnTimeURB that considers objective PACS factors in the brokering process [93]. The algorithm is aided with an ML model in order to additionally account for user-centered subjective factors for workflow executions. The ML model is designed to learn the bias of expert users towards cloud platforms for different requirements of functional criteria and workflow sizes.

Trusted Volunteer Edge Computing: We propose a novel compute methodology through VEC [94, 95]. A VEC system is comprised of multiple geographically distributed clusters, with each cluster having a set of co-located voluntary diverse edge resources. 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. To solve this problem we have proposed a novel "VECTrust" model for the support of trusted resource allocation algorithms in VEC computing environments [94].

6.2 Future Works

Future work could include improving the OnTimeURB by leveraging the expertise of researchers working with different computing platforms. Specifically, we could take into account user biases towards computing resources arising due to factors such as data protection policies, collaboration needs, and availability. This can facilitate optimizations by taking into consideration unquantifiable biases from users, while still accounting for PACS requirements and QoS fluctuations of computing resources. Further, OnTimeURB was developed only on the services which are available on a pay-as-you-go basis from different CSPs. The solution can be further improved upon to capture more services being offered by different cloud service providers for improving its recommendations. Since the cost to resources for many of the resources provided by cloud service providers does not scale linearly, broadening the scope of optimization to standalone services from cloud providers can improve resource allocation even more.

6.2.1 Autonomous-VEC

As part of the future work towards improving trust in VEC architecture, one can include dynamic capture of characteristics of a given set of workflows 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 workflow requirements and, handle diverse job arrival and cluster scheduling patterns in very large VEC systems. A better and improved architecture to standardize, configure and orchestrate VEC resources can be created for the mass execution of diverse workflows from geographically distributed users. Since VEC architecture is driven by the availability of volunteered resources, an autonomous framework could be developed which allows volunteer users to contribute their resources and earn compute hours which they can use in the future when they need more computing resources. This creates incentives for volunteers to contribute and utilize resources from the VEC pool of resources.

6.2.2 VEC and IoT Devices

VEC architecture of computing is based on volunteered edge resources, therefore IoT devices are ideal to further expand computing resources. With the tremendous growth in IoT devices and their capabilities such as smart watches, Nest Thermostat, Amazon Echo, etc. There is going to be the availability of more and more distributed devices that are capable of computations. These devices themselves might need computation capabilities from nearby edge resources to enrich their functionalities. Easy integration of VEC resources with these IoT devices will foster the growth of both VEC architecture as well as IoT devices.

REFERENCES

[1] Amazon Web Services. Available: https://aws.amazon.com/ [Online][Last accessed: 25th December 2021]

[2] Berman, M., Chase, J.S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., Ricci, R. and Seskar, I., 2014. GENI: A federated testbed for innovative network experiments. Computer Networks, 61, pp.5-23.

[3] University of Missouri Data Center. Available: https://doit.missouri.edu/services/servers-administration/server-housing/mu-data-center/ [Online][Last accessed: 25th December 2021]

[4] Google cloud platform. Available: https://cloud.google.com/ [Online][Last accessed: 25th December 2021]

[5] Microsoft Azure. Available: https://azure.microsoft.com/en-us/ [Online][Last accessed: 25th December 2021]

[6] Amazon OpsWorks. Available: https://aws.amazon.com/opsworks/ [Online][Last accessed: 25th December 2021]

[7] Calyam, Prasad and Rajagopalan, S. and Selvadhurai, Arunprasaath and Mohan, S. and Venkataraman, A. and Berryman, A. and Ramnath, Rajiv. (2013). “Leveraging OpenFlow for resource placement of virtual desktop cloud applications”. 311-319.

[8] Meurisch, Christian. ”The Trusted Edge.” arXiv preprint arXiv:2105.13601 (2021).

[9] Deelman, Ewa and Vahi, Karan and Juve, Gideon and Rynge, Mats and Callaghan, Scott and Maechling, Philip and Mayani, Rajiv and Chen, Weiwei and Ferreira da

Silva, Rafael and Livny, Miron and Wenger, Kent, “Pegasus, a workflow management system for science automation”, *Future Generation Computer Systems*, 46. DOI: 10.1016/j.future.2014.10.008, 2014

[10] Mireslami, Seyedehmehrnaz and Rakai, Logan and Wang, Mea and Homayoun Far, Behrouz., “Minimizing Deployment Cost of Cloud-Based Web Application with Guaranteed QoS”, Pages 1-6, DOI: 10.1109/GLOCOM.2015.7417230, 2015

[11] Mireslami, Seyedehmehrnaz and Rakai, Logan and Homayoun Far, Behrouz and Wang, Mea., “Simultaneous Cost and QoS Optimization for Cloud Resource Allocation”, *IEEE Transactions on Network and Service Management*, PP. 1-1. DOI: 10.1109/ TNSM.2017.2738026, 2017

[12] Antequera, Ronny and Calyam, Prasad and Ankathatti Chandrashekara, Arjun and Mitra, Reshmi, “Recommending heterogeneous resources for science gateway applications based on custom templates composition”, *Future Generation Computer Systems*, 100. DOI: 10.1016/j.future.2019.04.049, 2019

[13] Yadwadkar, Neeraja and Hariharan, Bharath and Gonzalez, Joseph and Smith, Burton and Katz, Randy. (2017). “Selecting the best VM across multiple public clouds: a data-driven performance modeling approach”. 452-465. 10.1145/3127479.3131614.

[14] Wang, Jun-Bo and Wang, Junyuan and Wu, Yongpeng and Wang, Jin-Yuan and Zhu, Huiling and Lin, Min and Wang, Jiangzhou. (2017). “A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing”. IEEE Network. 32. 10.1109/MNET.2018.1700293.

[15] 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.

[16] KubeEdge,“https://kubeedge.io/en/”.[Online].[Last accessed: 25th December 2021]

[17] IBM ILOG CPLEX Optimization Studio. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio [Online][Last accessed: 25th December 2021]

[18] Knowledge Base Commons. Available: http://kbcommons.org/ [Online][Last accessed: 25th December 2021]

[19] CyVerse—Cyberinfrastructure for Data Management and Analysis, "https://www.cyverse.org/". [Online]. [Last accessed: 25th December 2021]

[20] Pandey A., Lyu Z.,Joshi T., and Calyam P., "OnTimeURB: Multi-Cloud Resource Brokering for Bioinformatics Workflows," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 2019, pp. 466-473.

[21] C. L. P. Chen and C. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," Information Sciences, vol. 275, pp.314-347, 2014.

[22] D. Frazzetto, T. D. Nielsen, T. B. Pedersen et al., "Prescriptive analytics: a survey of emerging trends and technologies," The VLDB Journal, vol. 28, issue 4, pp. 575-595, 2019.

[23] C. Chen, "Top 10 unsolved information visualization problems," IEEE Computer Graphics and Applications, vol. 25, issue 4, pp. 12-16, 2005.

[24] R. Jordan Crouser and R. Chang, "An affordance-based framework for human computation and human-computer collaboration," IEEE Transactions on Visualization and Computer Graphics, vol. 18, issue 12, pp. 2859 – 2868, 2012.

[25] T. Herrmann, K. Loser, and I. Jahnke, "Sociotechnical walkthrough: a means for knowledge integration," The Learning Organization. Vol. 14, issue 5, pp.450-464, 2007.

[26] J.W. Creswell, Qualitative Inquiry Research Design: Choosing Among Five Approaches. Los Angeles, CA: SAGE Publications, 2013.

[27] A.W. Kushniruk, V.L. Patel, and J. J. Cimino, “Usability testing in medical informatics: cognitive approaches to evaluation of information systesm and user interfaces”, AMIA Annual Symposium Proceedings, pp. 218-222, 1997.

[28] M. van den Haak, M. De Jong, and P. Jan Schellens, “Retrospective vs. concurrent think-aloud protocpols: testing the usability of an online library catalogue”, Behavior and Information Technology, vol. 22, issue 4, pp. 339-351, 2003.

[29] J. Sauro, 10 things to know about the Single Ease Question (SEQ), Measuring U, October 2012. Available: https://measuringu.com/seq10/ [Online] [Last access: 29th August 2020]

[30] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” Intl. Journal of Human–Computer Interaction, vol. 24, issue 6, pp. 574-594, 2008.

[31] Morae Software Available: https://www.techsmith.com/tutorial-morae-current.html [Online] [Last accessed: 25th December 2021]

[32] J. Preece, H. Sharp, and Y Rogers, Interaction Design: Beyond Human-Computer Interaction. Danvers, MA: John Wiley Sons, 2019.

[33] Akhtar, Samia. (2014). Performance Evaluation In Cloud Computing Using Fuzzy Logic.

[34] Ebrahimnejad, Ali and Verdegay, José Luis, “Fuzzy Set Theory,” Fuzzy Sets-Based Methods and Techniques for Modern Analytics, vol. 364

[35] Bahuti, Marcelo and Abreu, Lucas and Yanagi Junior, Tadayuki and de Lima, Renato and Campos, Alessandro. (2018). Performance of fuzzy inference systems to predict the surface temperature of broiler chickens. Engenharia Agrícola. 38. 813-823. 10.1590/1809-4430-eng.agric.v38n6p813-823/2018.

[36] Hayat, Bashir and Kim, Kyong and Kim, Ki-Il. (2018). A study on fuzzy logic based cloud computing. Cluster Computing. 21. 1-15. 10.1007/s10586-017-0953-x.

[37] Mehranzadeh, Amin and Hashemi, Seyyed Mohsen. (2013). A Novel-Scheduling Algorithm for Cloud Computing based on Fuzzy Logic. International Journal of Applied Information Systems. 5. 28-31. 10.5120/ijais13-450939.

[38] Zavvar, Mohammad and Rezaei, Meysam and Garavand, Shole and Ramezani, Farhad. (2016). Fuzzy Logic-Based Algorithm Resource Scheduling For Improving The Reliability Of Cloud Computing. Asia-Pacific Journal of Information Technology and Multimedia. 05. 39-48. 10.17576/apjitm-2016-0501-04.

[39] Braiki, Khaoula and Youssef, Habib. (2019). Fuzzy-logic-based multi-objective best-fit-decreasing virtual machine reallocation. The Journal of Supercomputing. 10.1007/s11227-019-03029-8.

[40] Toosi, Adel. (2015). A Fuzzy Logic-based Controller for Cost and Energy Efficient Load Balancing in Geo-Distributed Data Centers. 10.1109/UCC.2015.35.

[41] Rizvi, Syed. and Mitchell, John and Razaque, Abdul and R. Rizvi, Mohammad and Williams, Iyonna . “A fuzzy inference system (FIS) to evaluate the security readiness of cloud service providers.” Journal of Cloud Computing 9 (2020): 1-17.

[42] Topaloglu, Fatih and Pehlivan, Hüseyin. (2018). “Comparison of Mamdani type and Sugeno type fuzzy inference systems in wind power plant installations”. 1-4. 10.1109/ISDFS.2018.8355384.

[43] Chakraverty, Snehashish and Sahoo, Deepti Moyi and Mahato, Nisha Rani (2019) Defuzzification. Springer:117–127

[44] Liu, Yang Khan, Saad Wang, Juexin Rynge, Mats Zhang, Yuanxun Zeng, Shuai Chen, Shiyuan Maldonado dos Santos, Joao Vitor Valliyodan, Babu Calyam, Prasad Merchant, Nirav Nguyen, Henry Xu, Dong Joshi, Trupti. (2016). ”PGen: Large-scale genomic variations analysis workflow and browser in SoyKB” *BMC Bioinformatics*, 17. 337. 10.1186/s12859-016-1227-y.

[45] Shen, Yelong and Jin, Ruoming and Dou, Dejing and Chowdhury, Nafisa and Sun, Junfeng and Piniewski, Brigitta and Kil, David. (2012). “Socialized Gaus-

sian Process Model for Human Behavior Prediction in a Health Social Network". Proceedings - IEEE International Conference on Data Mining, ICDM. 1110-1115. 10.1109/ICDM.2012.94.

[46] MathWorks 2017a https://in.mathworks.com/company/newsroom/mathworks-announces-release-2017a-of-the-matlab-and-simulink-pro.html [Online][Last accessed: 25th December 2021]

[47] XSEDE User Portal Available: https://portal.xsede.org//guest [Online][Last accessed: 8th August 2019]

[48] Pegasus workflow management system Available: https://pegasus.isi.edu/ [Online][Last accessed: 25th December 2021]

[49] HTCondor Available: https://research.cs.wisc.edu/htcondor/index.html [Online][Last accessed: 25th December 2021]

[50] WIPACrepo-pyglidein [Online][Last accessed: 25th December 2021] Available: https://github.com/WIPACrepo/pyglideinpyglidein

[51] Zou Guobing, Chen Yixin, Xiang Yang, Huang Ruoyun, Xu You, "AI Planning and Combinatorial Optimization for Web Service Composition in Cloud Computing", *Proceedings of the International Conference on Cloud Computing and Virtualization.*, Pages 28-35,

[52] Klusch M., Gerber A., "Fast composition planning of OWL-S services and application", *In Proceedings of the 4th European Conference on Web Services*, Pages 181-190, 2006

[53] Kurdi Heba, Al-Anazi Abeer, Campbell Carlene, Alfaries Auhood , "A combinatorial optimization algorithm for multiple cloud service composition", *Computers and Electrical Engineering*, 42. Pages 107-113,

[54] Panda Sanjaya, Jana Prasanta , "Efficient task scheduling algorithms for heterogeneous multi-cloud environment", *The Journal of Supercomputing*, 71. Pages 1505-1533,

[55] Su, Kui and Xu, Lei and Chen, Cong and Chen, Wenzhi and Wang, Zonghui., "Affinity and Conflict-Aware Placement of Virtual Machines in Heterogeneous Data Centers," *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems, Taichung*, 2015, pp. 289-294. DOI: 10.1109/ISADS.2015.42

[56] Goudarzi, Hadi and Pedram, Massoud., "Multi-dimensional SLA-Based Resource Allocation for Multi-tier Cloud Computing Systems," *2011 IEEE 4th International Conference on Cloud Computing, Washington, DC*, 2011, pp. 324-331. DOI: 10.1109/CLOUD.2011.106

[57] Mohamed, A.M., Abdelsalam, H.M. (2020). "A multicriteria optimization model for cloud service provider selection in multicloud environments". Software: Practice and Experience, 50, 925 - 947.

[58] Yadwadkar, Neeraja and Hariharan, Bharath and Gonzalez, Joseph and Smith, Burton and Katz, Randy. (2017). "Selecting the best VM across multiple public clouds: a data-driven performance modeling approach". 452-465. 10.1145/3127479.3131614.

[59] Mao, Hongzi and Schwarzkopf, Malte and Venkatakrishnan, Shaileshh and Meng, Zili and Alizadeh, Mohammad. (2018). "Learning Scheduling Algorithms for Data Processing Clusters".

[60] Nasreen, Shamila. (2014). "A Survey Of Feature Selection And Feature Extraction Techniques In Machine Learning",SAI,2014.

[61] Saaty TL. "Decision Making for Leaders: the Analytic Hierarchy Process for Decisions in a ComplexWorld". Third Revised Edition. Pittsburgh, PA: RWS Publications; 2012.

[62] KubeEdge,"https://kubeedge.io/en/".[Online].[Last accessed: 25th December 2021]

[63] X. Li and J. Du, "Adaptive and attribute-based trust model for service-level agree-

ment guarantee in cloud computing," *IET Information Security*, vol. 7, no. 1, pp. 49-50, Mar. 2013. doi: 10.1049/iet-ifs.2012.0232

[64] 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

[65] A. Josang, and J. Haller, "Dirichlet reputation systems," In *2nd International Conference on Availability, Reliability and Security (ARES'07)*, pp. 112-119, Apr. 2007.

[66] 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

[67] 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

[68] 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.

[69] 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

[70] 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

[71] 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

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] Joint Task Force Transformation Initiative, "Guide for Conducting Risk Assessments", *NIST Special Publication 800-30*, 2012.

[78] Rao, R., Liptak, D., Cherukuri, T., Yakobson, B. I. and Maruyama, B. "In situ evidence for chirality-dependent growth rates of individual carbon nanotubes". Nat. Mater. 11, 213–216 (2012).

[79] Nikolaev, P. et al. Autonomy in materials research: a case study in carbon nanotube growth. npj Comput. Mater. 2, 1–6 (2016).

[80] Hajilounezhad, T., Ajiboye, D. M. and Maschmann, M. R. "Evaluating the forces

generated during carbon nanotube forest growth and self-assembly". Materialia 7, 100371 (2019).

[81] Maschmann, M. R. Integrated simulation of active carbon nanotube forest growth and mechanical compression. Carbon 86, 26–37 (2015).

[82] Brown, J. et al. Delamination mechanics of carbon nanotube micropillars. ACS Appl. Mater. Interfaces 11, 35221–35227 (2019).

[83] Hines, R., Hajilounezhad, T., Love-Baker, C., Koerner, G. and Maschmann, M. R. 'Growth and mechanics of heterogeneous, 3D carbon nanotube forest microstructures formed by sequential selective-area synthesis. ACS Appl. Mater. Interfaces 12, 17893–17900 (2020).

[84] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz and M. Khansari, "RL-CycleGAN: Reinforcement Learning Aware Simulation-to-Real," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11154-11163, doi: 10.1109/CVPR42600.2020.01117.

[85] R. Hsu, J. Lee, T. Q. S. Quek and J. Chen, "Reconfigurable Security: Edge-Computing-Based Framework for IoT," in IEEE Network, vol. 32, no. 5, pp. 92-99, September/October 2018, doi: 10.1109/MNET.2018.1700284.

[86] T. Rahman, X. Yao, G. Tao, H. Ning and Z. Zhou, "Efficient Edge Nodes Reconfiguration and Selection for the Internet of Things," in IEEE Sensors Journal, vol. 19, no. 12, pp. 4672-4679, 15 June15, 2019, doi: 10.1109/JSEN.2019.2895119.

[87] K. Singh et al., "A Formative Usability Study to Improve Prescriptive Systems for Bioinformatics Big Data," 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020, pp. 735-742, doi: 10.1109/BIBM 49941.2020.9313097.

[88] K. Vekaria, P. Calyam, S. Sivarathri, S. Wang, Y. Zhang, A. Pandey, C. Chen, D. Xu, T. Joshi, S. S. Nair, (2020). "Recommender-as-a-service with chatbot

guided domain-science knowledge discovery in a science gateway". Concurrency and Computation: Practice and Experience. 33. 10.1002/cpe.6080.

[89] A. Pandey, P. Calyam, Z. Lyu and T. Joshi, "Fuzzy-Engineered Multi-Cloud Resource Brokering for Data-intensive Applications," 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2021, pp. 257-266, doi: 10.1109/CCGrid51090.2021.00035.

[90] S.S. Sivarathri, P. Calyam, Y. Zhang, A. Pandey, C. Chen, D. Xu, T. Joshi, S.S. Nair (2019). "Chatbot Guided Domain-science Knowledge Discovery in a Science Gateway Application".

[91] A. Pandey, S. Wang and P. Calyam,"Data-intensive Workflow Execution using Distributed Compute Resources," 2019 IEEE 27th International Conference on Network Protocols (ICNP), 2019, pp. 1-2, doi: 10.1109/ICNP.2019.8888119.

[92] A. Pandey, Z. Lyu, T. Joshi and P. Calyam, "OnTimeURB: Multi-Cloud Resource Brokering for Bioinformatics Workflows," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2019, pp. 466-473, doi: 10.1109/BIBM47256.2019.8983386.

[93] A. Pandey, P. Calyam, Z. Lyu, S. Wang, D. Chemodanov, T. Joshi "Knowledge-Engineered Multi-Cloud Resource Brokering for Application Workflow Optimization". IEEE TNSM 2022 (Minor Revision)

[94] A. Pandey, P. Calyam, S. Debroy*, S. Wang, M. L. Alarcon. 2021. "VECTrust: Trusted Resource Allocation in VolunteerEdge-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.

[95] M. L. Alarcon, M. Nguyen, A. Pandey, S. Debroy and P. Calyam. 2022 "VECFlex: Reconfigurability and Scalability for Trustworthy Volunteer Edge-Cloud supporting Data-intensive Scientific Computing" (accepted)

[96] L. Xiao, Y. Ding, D. Jiang, J. Huang, D. Wang, J. Li and H.V. Poor, 2020. "A reinforcement learning and blockchain-based trust mechanism for edge networks". IEEE Transactions on Communications, 68(9), pp.5460-5470.

[97] M.H. Ling, K.L.A. Yau, J. Qadir, and Q. Ni, 2018. A reinforcement learning-based trust model for cluster size adjustment scheme in distributed cognitive radio networks. IEEE Transactions on Cognitive Communications and Networking, 5(1), pp.28-43.

[98] A. Vijaya Kumar and A. Jeyapal, 2014. Self-adaptive trust based ABR protocol for MANETs using Q-learning. The Scientific World Journal, 2014.

[99] Cloud Computing Comparison Engine Available: https://www.cloudorado.com/ [Online][Last accessed: 30th November 2021]

[100] Open-source Image Magick Package. Available: https://imagemagick.org/script/index.php [Online][Last accessed: 13th May 2022]

[101] Hripcsak, G., Duke, J. D., Shah, N. H., Reich, C. G., Huser, V., Schuemie, M. J., et al. (2015). Observational health data sciences and informatics (OHDSI): Opportunities for observational researchers. Studies in Health Technology and Informatics, 216, 574.

[102] R. Makadia, P. B. Ryan,(2014). Transforming the premier perspective® hospital database into the observational medical outcomes partnership (OMOP) common data model. Egems, 2(1), 1110.

[103] X. Yue, H. Wang, D. Jin, M. Li, W. Jiang (2016). Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. Journal of Medical Systems, 40, 1–8.

[104] D. R. Matos, M. L. Pardal, P. Adao, A. R. Silva, M. Correia (2018). Securing electronic health records in the cloud. In Proceedings of the 1st workshop on privacy by design in distributed systems (pp. 1–6).

VITA

Ashish Pandey

PhD student of Computer Science;

University of Missouri-Columbia, USA

I comes from a humble, middle-class family in India. I am a first-generation college student. I am a technology enthusiast and love to read and understand know-how and working mechanisms of varying technologies ranging from computer science, physics, and bioengineering. Coming from a developing country and having seen the struggles of communities with limited resources, I am deeply invested in processes to uplift quality of life and health care. I received my B. Tech degree in Mechanical Engineering from the Indian Institute of Technology, Jodhpur, India. My research interests include cloud computing, recommender systems, artificial intelligence, machine learning, and their integration with bioinformatics. In addition to the aforementioned practice areas, I am an experienced software developer and am skilled in full stack development (MEAN stack), Core Java, and Product Lifecycle Management (PLM) as well as Computer-Aided Design (CAD) technologies.

In my undergrad, I extensively used MATLAB, Ansys, and Solidworks (for prototype modeling and evaluations) for my academic projects. The amazing simulations and modeling softwares inspired me towards the interdisciplinary field of Computer-Aided Design (CAD). I eventually joined the leading CAD software development company, Dassault Systèmes as a Software Engineer where I worked extensively with multiple programming languages such as C++, Java as well as technologies such as Tool Command Language (TCL) and Databases. Despite a challenging and fullfilling career at Dassault, I was unable to pursue my other scientific interests. Continuous conversations on scientific research with one of my closest friends who was completing his Ph.D. in biomedical engineering reminded me of my childhood dreams of working with tech-

nology at the intersection of biotechnology and healthcare which can directly help the community. I decided to pursue doctorate studies in Computer Science. As a graduate student, my continued interactions with faculties eventually led me to join Professor Prasad Calyam's research lab, which specialized in networking, virtualization and cloud computing research. My research is focused on streamlining data and memory-intensive bioinformatics workflow execution using community and commercial cloud computing platforms wherein I have used methods like integer linear programming for optimal cloud resources selection. The challenges for the research prompted me to learn about tools such as Pegasus workflow management system for creating bioinformatics workflows, HTCondor and SLURM workload scheduler for data staging and distribution of subtasks within cloud platforms and, also about several cloud computing platforms such as GENI, Grid HP sites, AWS among others. I also contributed to the creation of bioinformatics workflows such as FastQC, RNA-Seq, Methylation, Single-cell analysis. I am further exploring machine learning techniques for improving cloud resource allocation for workflows from the field of bioniformatics. The research required me to work with large genomic data along with analytical tools such as fastqc, bwa, tophat and, the resulting success through research papers instigated my curiosity further in the areas of translational bioinformatics. The rigor of working on genomics data for scientific workflows encouraged me to learn more about the complexity of biological problems and the immense possibilities understanding and solving these can offer to human health.

I and deeply passionate about science and my career goal is to contribute to extraordinary research and creating technologies that can help in limiting health issues in society.