Multi-Cloud Performance and Security-driven Brokering for Bioinformatics Workflows

Minh Nguyen, Saptarshi Debroy, Prasad Calyam, Zhen Lyu, Trupti Joshi
City University of New York, University of Missouri-Columbia
Emails: {*minh.nguyen, saptarshi.debroy*}*@hunter.cuny.edu,* {*calyamp, zl7w2, joshitr*}*@missouri.edu*

***Abstract*—Data-intensive bioinformatics applications often use federated multi-cloud infrastructures to support compute-intensive processing needs. In this paper, we propose a Multi-Cloud Performance and Security (MCPS) Brokering framework within such federated multi-cloud infrastructures to allocate cloud resources to applications by satisfying their performance and security requirements.**

I. Introduction

Data-intensive bioinformatics applications often require specialized compute/networking/storage resources that are not always available locally on-site and need to use resources in remote cloud domains for processing. Thus, researchers are increasingly adopting federated multi-cloud infrastructures (e.g., CyVerse [1]) to support compute-intensive or data-intensive science collaborations. Allocation of such federated multi-cloud resources is typically based on applications' performance considerations (e.g., data throughput, execution time). However, such one-dimensional resource brokering fails to consider scenarios where applications' security requirements across different life-cycle stages (Low, Moderate, and High) contradict with remote domains' diverse security policies (ranging from very strict to very relaxed).

In this paper, we propose a Multi-Cloud Performance and Security (MCPS) Brokering framework for resource allocation of a set of SoyKB [2] bioinformatics application workflows across federated multi-cloud infrastructures. The proposed framework builds upon formalized performance specifications or *QSpecs* and security specifications or *SSpecs* of exemplar SoyKB workflows [3]. The framework also facilitates a end-to-end workflow security design that formalizes and complies with diverse domain security policies or *RSpecs* used by the application relating to local and remote cloud domains. The MCPS Broker performs a security-aware global scheduling to choose the optimal cloud domain, and a local scheduling to choose the optimal server/core within the chosen cloud domain. Using real SoyKB application workflows, we implement the proposed MCPS Broker framework in the GENI Cloud and demonstrate its utility through a National Institue of Standards and Technology (NIST) guided risk assessment [4].

II. SoyKB Workflow Specifications

For the purposes of this work, we consider the implementations of two high-throughput cloud-based bioinformatics data analysis workflows in the SoyKB [2] science gateway developed for soybean and other related organisms. These workflows provide biological users with an avenue to analyze their in-house generated datasets using multi-step workflows and conduct analysis in high performance computing environments that support the necessary security levels to handle Health Insurance Portability and Accountability Act (HIPPA) compliance. The complex PGen workflow is used to efficiently facilitate analysis of large-scale next generation sequencing (NGS) data for genomic variations. Whereas a comparatively simpler RNA-Seq analysis workflow is used to perform quantization of gene expression from transcriptomics data and statistical analysis to discover differential expressed gen/isoform between experimental groups.

TABLE I: RNA-Seq workflow *QSpecs*

Stages	Compute (cores)	Storage (GB)	Network (Mbps)
1. Pre-processing	1	4	NA
2. Alignment	1	20	NA
3. Sort BAM	1	20	NA
4. Annotation	1	20	NA
5. Post-process	1	10	NA
6. Expression	14	10	NA
7. Variants	14	10	NA

Tables I and II express the *QSpecs* and *SSpecs* of RNA-Seq workflows for different stages of the workflow processing life-cycle in tabular forms. The *QSpecs* expresses the number of compute cores, memory storage in GBs, and network bandwidth in Mbps specifications for each stage of the workflow life-cycle. The *SSpecs* is a formal data structure to describe the minimum security requirements against confidentiality, integrity, and availability threats and represented in terms of 'Data' and 'Auxiliary' security requirements. 'Data' requirements are divided into Compute, Storage, and Network requirements, i.e., resources that deal with the data unlike 'Auxiliary' requirements.

III. Algorithms and Service Design

For the proposed framework, SoyKB workflows are represented as Directed Acyclic Graphs (DAGs) with vertices representing individual life-cycle stages and edges representing stage transition. The objective is to schedule each such DAG vertex to one or more computing cores within an individual multi-cloud domain that satisfies the workflow *QSpecs* and *SSpecs*. To achieve this, the MCPS broker employs two algorithms: i) a global scheduling algorithm to allocate DAG vertices to domains that will achieve DAG scheduling with *SSpecs* satisfaction and ii) a local scheduling algorithm to choose optimal computing core within the chosen domain for *QSpecs* satisfaction. As such a multi-constrained scheduling optimization problem is NP-complete, MCPS Broker use a modified version of Fast Critical Path (FCP) [5] heuristic algorithm due to its near-optimal performance yet maintaining a relatively low running time.

This material is based upon work supported by the the National Science Foundation under Award Number: OAC-1827177. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

TABLE II: RNA-Seq workflow *SSpecs*; L: Low, M: Moderate, and H: High.

Stages	Compute							Storage							Network							Auxiliary									
	AC	AU	CA	IA	SA	SC	SI	AC	AU	CA	IA	SA	SC	SI	AC	AU	CA	IA	SA	SC	SI	AT	CM	CD	IR	MA	MP	PE	PL	PM	PS	RA
1	H	L	H	H	L	L	L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
2	H	L	H	H	L	L	L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
3	H	L	H	H	L	L	L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
4	H	L	H	H	L	L	L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
5	H	L	H	H	L	L	L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
6	H	L	H	H	L	L	L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
7	H	L	H	H	L	L	L	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

Fig. 1 illustrates the overall security-driven resource brokering services design and underlying components. The *Workflow Manager* gathers DAG information of input workflows and generates the *QSpecs* and *SSpecs* that are used by the *MCPS Broker* for resource scheduling. The MCPS Broker uses resource availability and domain policy (*RSpecs*) information from *Resource Manager* for resource brokering. The *Execution Controller* manages the overall operation and message passing among the different components.

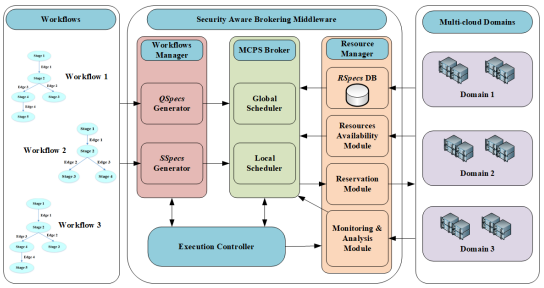


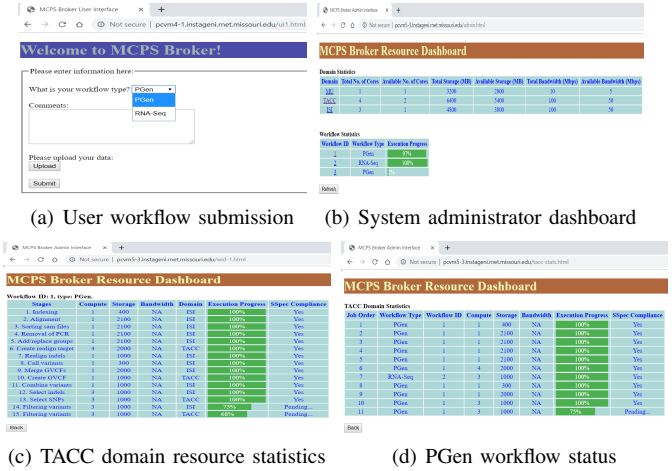
Fig. 1: MCPS Brokering service design components

IV. Implementation and Evaluation

Here we illustrate the initial testbed setup on GENI infrastructure that is used for our evaluations. We geographically distribute the multi-cloud resource domains approximately based on the real computing centers used for SoyKB workflows: a local University of Missouri (MU) domain, as well as remote cloud domains, such as Texas Advanced Computing Center (TACC), and Information Sciences Institute (ISI). We create compute capability and network bandwidth mismatches in terms of number of cores, their speed, and Mbps values mimicking real-life SoyKB implementation. The testbed also replicates security policies of TACC, ISI, and MU domains as well as dynamic resource utilization levels. For the experiments, workflows are sent from the MU domain users through the MCPS Broker, which decides whether the workflows are processed locally at MU or remotely at TACC or ISI based on the global and local algorithm outcomes discussed with the results ultimately sent to CyVerse upon processing.

Fig. 2 describes the MCPS Broker user interfaces we developed for the purposes of the testbed experiments and data collection. Fig. 2(a) shows how users can select the particular type of workflow being uploaded to the system. Fig. 2(b) shows the admin dashboard of MCPS broker where the administrator can monitor the resources available and the working status of each domain as well as the working status of each workflow. The admin can view details of each domain resources or the details of the running workflows by selecting the relevant menu options (as shown in Fig. 2(c)). The admin is further provided the option to view the detailed statistics of each workflow using the corresponding Workflow ID (as shown in Fig. 2(d)).

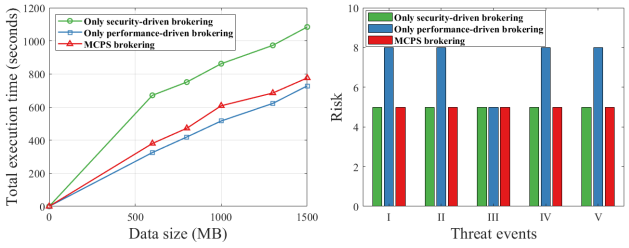
Fig. 3(a) shows that for different data sizes, MCPS brokering performs almost as good as 'only performance-driven brokering' in terms of choosing domains for processing that optimize total execution time. Whereas, 'only security-driven brokering' performs poorly as it always chooses ISI for processing irrespective of the ISI domain's resource availability for ISI being most secured.



(a) User workflow submission (b) System administrator dashboard

(c) TACC domain resource statistics (d) PGen workflow status

Fig. 2: MCPS Broker User Interface



(a) Performance comparison (b) Security evaluation

Fig. 3: Brokering scheme comparison

Finally, the security compliance comparison results shown in Fig. 3(b) use the NIST [4] based risk assessment method. This risk assessment study allows us to compare the security compliance in terms of domain selection for processing against 5 threats (of potentially 'High' to 'Moderate' impact) for different brokering techniques. The figure shows that the overall risk of different threats are similar for 'only security-driven brokering' and our proposed 'MCPS brokering' as these schemes almost always choose ISI or TACC over MU regardless of the formers' resource availability. This is because the TACC and ISI have clearly laid out policies. However, 'only performance-driven brokering' sometime chooses MU over ISI or TACC if MU has much higher resource availability in comparison to ISI or TACC, thus compromising security.

References

[1] CyVerse - https://www.cyverse.org.

[2] T. Joshi et. al., "Soybean Knowledge Base (SoyKB): A Web Resource For Soybean Translational Genomics", *BMC Genomics*, Vol. 13, No. 1, 2012.

[3] M. Dickinson et. al., "Multi-cloud Performance and Security Driven Federated Workflow Management," *IEEE Transactions on Cloud Computing (TCC)*, 2018.

[4] R. S. Ross, "Guide for Conducting Risk Assessments", *NIST SP800-30- Rev1 Technical Report*, 2012.

[5] A. Radulescu and A. J. C. van Gemun, "On the complexity of list scheduling algorithms for distributed memory systems", *Proc. of ACM Supercomputing*, 1999.