

# SkyLab : A Web Application for vCluster

Vincent Paul L. Carpio and Katrina Joy H. Magno and Joseph Anthony C. Hermocilla

## I. INTRODUCTION

### A. Background

Cloud computing has marked significant developments and possibilities in the industry. It focuses on offering services for the different needs of the modern society. There are three categories of cloud computing services namely, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Organizations provide SaaS depending on the demand. Google Apps is one example of SaaS that can be used to manage email and create documents, etc. PaaS offers developers a platform where they can build and deploy applications. IaaS provides storage, servers and handling computer clusters. These tools are primarily created to serve computational needs [1]. A cloud computing platform dynamically allocates, configures, reconfigures and deallocates servers as requested or on demand. This approach ensures the elasticity of cloud computing [2].

Most scientific applications require high performance computing (HPC) which needs CPU intensive computations and large data storage. To be able to host such applications, several computers interconnected in a network such as clusters are needed. This makes scientific computing very costly. Fortunately, with the advancement in cloud computing, these tools can be deployed in the cloud without worrying about the costs of hardware purchase and maintenance [1]. To determine the cost of hosting scientific computing in cloud, the Scientific Computing Cloud (SciCloud) project is conducted in the University of Tartu. Results have shown that transmission delays are the major concern in pursuing HPC applications in the cloud [2]. To address this problem, a study is conducted by Hermocilla in Peak-Two Cloud (P2C). One of the features introduced by P2C is vCluster. vCluster is a tool that enables a user to deploy a working MPI cluster on demand and to terminate it after execution. It uses a master-slave design implementation [3]. However, vCluster is a command-line based application which has limited capacities for data manipulation, analysis, and presentation. Creating a web application that would host vCluster with additional features like graphical representation to visualize trends, and a Graphical User Interface (GUI) to offer convenient access. This paper will present SkyLab, a web application that aims to serve and to improve vCluster functionalities.

### B. Significance of the Study

This study will help us visualize how HPC applications can be hosted in the cloud and understand why is this technology

timely relevant. Even with capable hardware, computations might take hours or even days which makes it a hassle for users. Thus, there is a need for a more convenient way of access for HPC applications. Furthermore, this would encourage students to explore and research on HPC applications. HPC applications usually have a plain numerical output which needs manual effort to be able to clearly visualize trends. A means of presenting data graphically would make the interpretation of data more efficient.

The study would contribute to research on hosting HPC applications in the cloud. Problems encountered and solutions offered throughout the study will give insights to future researchers. Furthermore, the output application of the study can be used by the academe.

### C. Scope and Limitations

The study focuses on vCluster and is not mainly concerned on P2C as a whole. The study initially bases on the current implementation of vCluster and eventually improve it and offer additional features [3].

### D. Objectives

This study aims to develop SkyLab, a web application that would function as a front-end for vCluster. Specifically, it should be able to:

- 1) allow users to select tools and execute them via web interface;
- 2) create an extensible platform that would accommodate additional tools; and
- 3) display output data of tools.

## II. REVIEW OF RELATED LITERATURE

### A. Performance Concerns for HPC Applications in the Cloud

Shajulin [4] has enumerated five performance concerns for HPC applications in the cloud. First concern is resource provisioning issues which includes memory management, energy management, response time, and resource outages. Hardware virtualization for multiple guest operating systems running concurrently on a host computer creates a concern for huge overheads, delays, or memory leakages caused by the process of allocating memory, remapping it to virtual machines, and cleaning the memory. The scenario of multiple guest OS can also cause an OS to fail to make hardware use energy-efficient due to lack of control over hardware resources. Configuration failures, update failures, or even unknown issues cause resource outages. Second, most programming models used in the cloud have inconsistencies between the fault recovery mechanisms in execution and

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

storage processes. Inconsistencies can result to poor performance. Third, enforcing terms of Service Level Agreement (SLA) is bound to performance issues due to loss of control over resource provisioning or deploying cloud services. Fourth, lack of best practices or standards has hindered the utilization of the cloud. If there is no standard interoperable cloud interfaces, users might need to modify their application accordingly with the cloud provider. Fifth, security measures challenges performance since increasing security protocols decreases response time [4].

In a study by Jackson et al. [5], performance of a set of benchmarks designed to represent a typical HPC workload run on Amazon EC2 have been quantitatively examined. Results of the study show that the percentage of the communication time of the application is strongly correlated to its overall performance in EC2. The more communication, the worse it performs. In addition, applications with significant global communication perform relatively worse than those with less. Lastly, variability can be significant in EC2. Variability is introduced by multiple virtual machines, by the network and by underlying non-virtualized hardware [5].

Another study conducted by [6], analyzed running scientific workflows in EC2. It has been found that the primary cost is in getting resources to execute workflow tasks, and storage costs are relatively cheaper. Storing data in the cloud rather than transferring it to each workflow effectively reduces the high cost of data transfer. These results show that the cloud is a good alternative for running scientific workflow applications but for cloud providers to be able compete with existing physical systems in terms of performance of HPC applications would need high-speed networks and parallel file systems [6] [7]. While current cloud computing services are insufficient for large-scale scientific computing, it may still be a viable option for the scientists who need resources instantly and temporarily [8]. Despite the performance tradeoff, EC2 offers ease-of-use and cheaper costs which are marginal factors in consideration for academic purposes [9].

### B. Related Work

The Yabi system [10] targets the non-technical audience with an intuitive drag-and-drop scalable web-based workflow environment that allows HPC to be appreciated a broader audience. Yabi also enables other workflow systems that have limited access to multiple existing HPC to utilize its data and access models.

Another related project is WImpiBLAST [11], an open-source web interface for parallel BLAST searches. BLAST is the most extensively used gene sequence analysis program for sequence homology similarity search in large databases of sequences. mpiBLAST is an open-source parallelization of BLAST that can speedup large-scale annotation by using supercomputers and HPC clusters. WImpiBLAST is created to help researchers who lack expertise in using Message Passing Interface (MPI) benefit from its advantages.

## III. METHODOLOGY

### A. Materials

Programming Language: Python

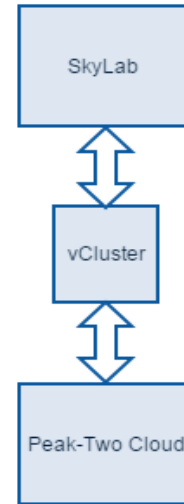


Fig. 1. System Architecture of SkyLab

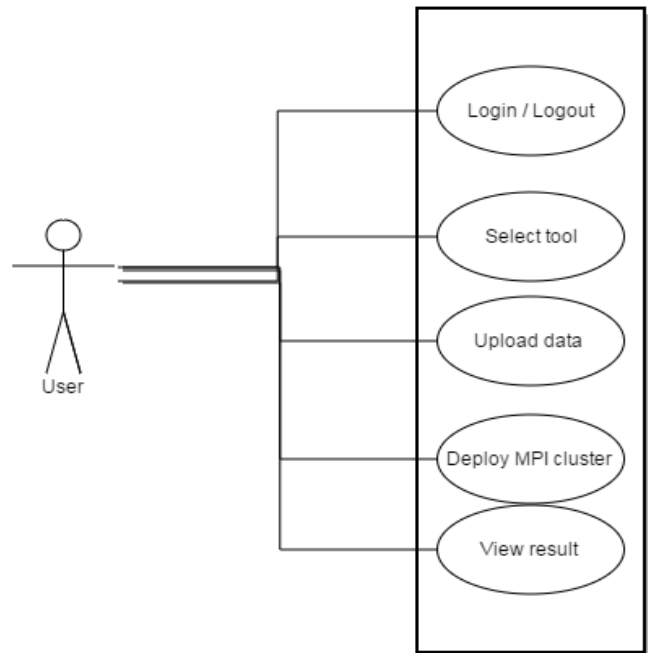


Fig. 2. Use Case Diagram of SkyLab

Web Framework: Django

DBMS: SQLite

Cloud Infrastructure: Peak-Two Cloud

### B. System Architecture

SkyLab functions as a web frontend for vCluster which is built on top of Peak-Two Cloud.

### C. Requirements Specification

#### 1) Functional Requirements:

- Users can browse and select tools with an interface. It includes, but not limited to:

#### AutoDock

It is a suite of automated docking tools designed to predict how small molecules bind to a receptor of known 3D structure. [12]

#### AutoDock Vina

It achieves significant improvements in the average accuracy of the binding mode predictions, while also being up to two orders of magnitude faster than AutoDock 4. [12]

#### DOCK

It is used to predict the small molecule-target interactions. [13]

#### Quantum Espresso

It is an integrated suite of Open-Source computer codes for electronic-structure calculations and materials modeling at the nanoscale. [14]

#### GAMESS

GAMESS is a program for ab initio molecular quantum chemistry. [15]

#### Ray

It uses parallel genome assemblies for parallel DNA sequencing [16]

- Users can upload input data to the server.
- Users can view the results of the used tools.
- The system is integrated with vCluster functionalities.

#### 2) Non-functional Requirements:

- Users must be authenticated to be able to use the system.
- The system must be easy to use and user friendly.
- The system must be able to accommodate a maximum of 100 users logged on and using functionalities of the system.
- The system must be highly maintainable for future development.

#### D. Evaluation of Methodology

The system requirements specified and use cases satisfy the given objectives of the study. The created software will be subjected to user acceptance test for further evaluation. A load tester will be created to measure the scalability of the program.

### IV. INITIAL RESULTS

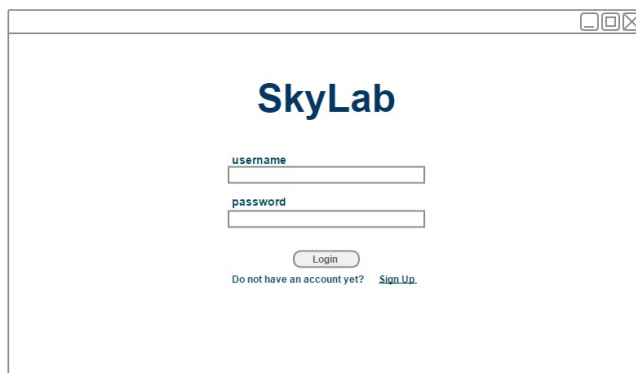


Fig. 3. Prototype Design of Login Page

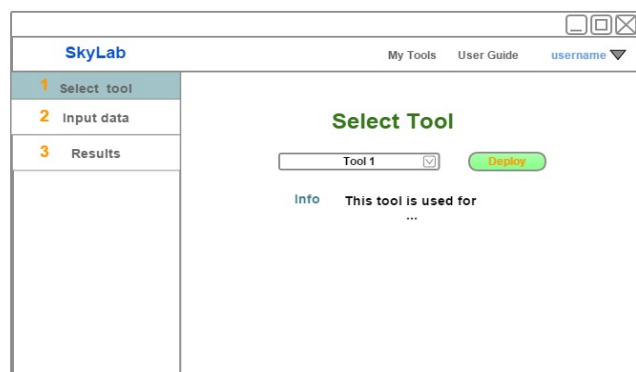


Fig. 4. Prototype design of Tool Selection Page

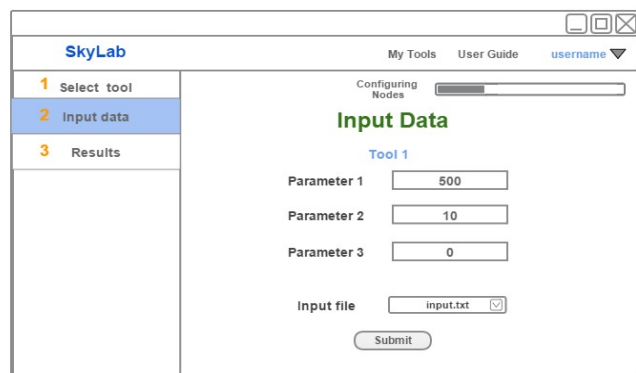


Fig. 5. Prototype Design of Data Input Page

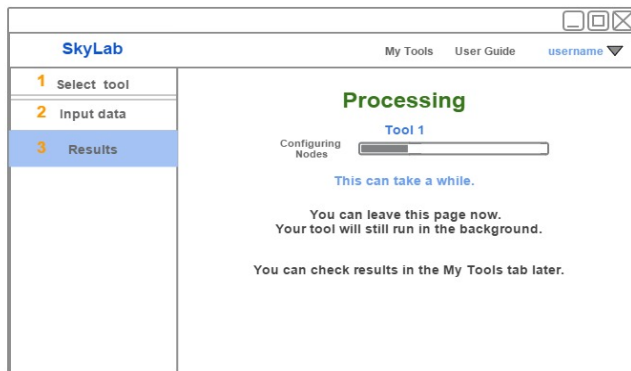


Fig. 6. Prototype Design of Processing Results Page

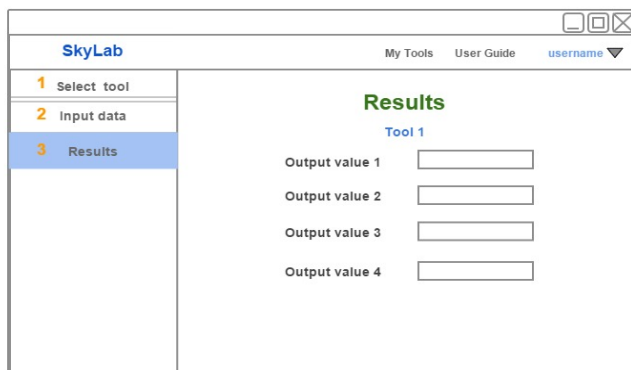


Fig. 7. Prototype Design of Results Page

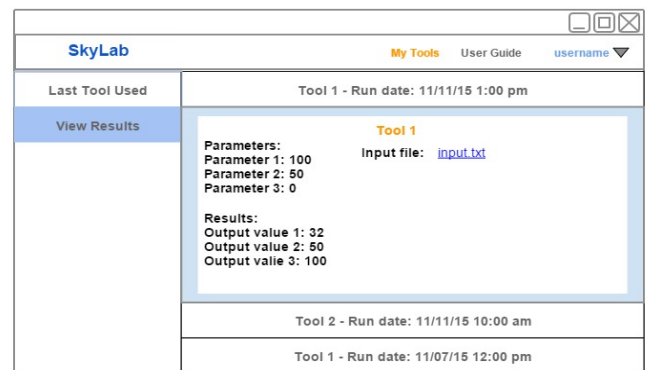


Fig. 9. Prototype Design of Sample Results Segment of My Tools Page

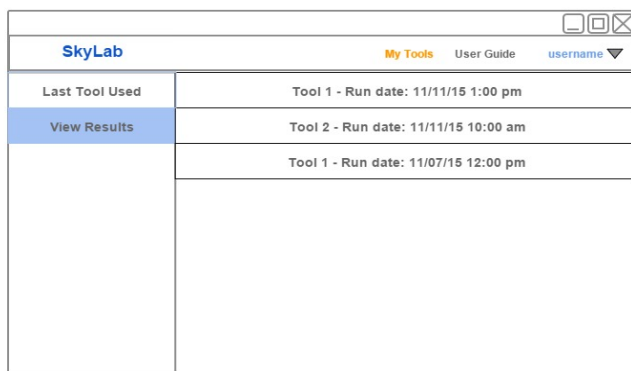


Fig. 8. Prototype Design of Results Segment of My Tools Page

## REFERENCES

- [1] S. P. Ahuja and S. Mani, "The state of high performance computing in the cloud," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 2, pp. 263–266, Feb. 2012. [Online]. Available: <http://www.chinacloud.cn/upload/2012-03/12031713036456.pdf>
- [2] I. Brandic, I. Raicu, S. N. Srirama, O. Batrashev, P. Jakovits, and E. Vainikko, "Scalability of parallel scientific applications on the cloud," *Scientific Programming*, vol. 19, no. 2/3, pp. 91 – 105.
- [3] J. A. C. Hermocilla, "P2c: Towards scientific computing on private clouds," in *Proceedings of the National Conference on Information Technology Education (NCITE 2014)*, 2014, pp. 162–167.
- [4] S. Benedict, "Performance issues and performance analysis tools for hpc cloud applications: a survey," *Computing*, vol. 95, no. 2, pp. 89 – 108.
- [5] K. Jackson, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. Wasserman, and N. Wright, "Performance analysis of high performance computing applications on the amazon web services cloud," 2010. [Online]. Available: <https://www.nersc.gov/assets/NERSC-Staff-Publications/2010/CloudCom.pdf>
- [6] G. Juva, E. Deelman, K. Vahi, G. Mehta, B. Berman, B. Berriman, and P. Maechling, "Scientific workflow applications on amazon ec2," [Online]. Available: <http://www.isi.edu/gideon/publications/JuveG-EC2.pdf>
- [7] E. Walker, "Benchmarking amazon ec2 for high-performance scientific computing," Oct 2008. [Online]. Available: <https://www.usenix.org/legacy/publications/login/2008-10/openpdfs/walker.pdf>
- [8] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahrigner, and D. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing."
- [9] Z. Hill and M. Humphrey, "A quantitative analysis of high performance computing with amazons ec2 infrastructure: The death of the local cluster?" in *Proceedings of the 10th IEEE/ ACM International Conference on Grid Computing (Grid 2009)*, Oct 2009. [Online]. Available: <http://www.cs.virginia.edu/humphrey/papers/QuantitativeAnalysisEC2.pdf>
- [10] A. A. Hunter, A. B. Macgregor, T. O. Szabo, C. A. Wellington, and M. I. Bellgard, "Yabi: An online research environment for grid, high performance and cloud computing." *Source Code for Biology and Medicine*, vol. 7, no. 1, pp. 1 – 10.
- [11] P. Sharma and S. S. Mantri, "Wimpiblast: Web interface for mpiblast to help biologists perform large-scale annotation using high performance computing." *PLoS ONE*, vol. 9, no. 6, pp. 1 – 13.
- [12] "AutoDock AutoDock." [Online]. Available: <http://autodock.scripps.edu/>
- [13] "Overview of DOCK." [Online]. Available: [http://dock.compbio.ucsf.edu/Overview\\_of\\_DOCK/index.htm](http://dock.compbio.ucsf.edu/Overview_of_DOCK/index.htm)
- [14] "QUANTUMESPRESSO - Quantum ESPRESSO." [Online]. Available: <http://www.quantum-espresso.org/>
- [15] "Gordon Group/GAMESS Homepage." [Online]. Available: <http://www.msg.ameslab.gov/games/capabilities.html>
- [16] S. Boisvert, F. Raymond, . Godzaridis, F. Laviolette, and J. Corbeil, "Ray Meta: scalable de novo metagenome assembly and profiling." *Genome Biology*, vol. 13, no. 12, p. R122, 2012. [Online]. Available: <http://genomebiology.com/2012/13/12/R122>