Workload Characterization of the Shared/Buy-in Computing Cluster at Boston University

Yonatan Klausner* and Christopher Liao*, David Starobinski, Eran Simhon, and Azer Bestavros
*Both authors contributed equally
Boston University
yklausner@gmail.com, {cliao25, staro, simhon, best}@bu.edu

***Abstract*—Computing clusters provide a complete environment for computational research, including bio-informatics, machine learning, and image processing. The Shared Computing Cluster (SCC) at Boston University is based on a shared/buy-in architecture that combines shared computers, which are free to be used by all users, and buy-in computers, which are computers purchased by users for semi-exclusive use. Although there exists significant work on characterizing the performance of computing clusters, little is known about shared/buy-in architectures. Using data traces, we statistically analyze the performance of the SCC. Our results show that the average waiting time of a buy-in job is 16.1% shorter than that of a shared job. Furthermore, we identify parameters that have a major impact on the performance experienced by shared and buy-in jobs. These parameters include the type of parallel environment and the run time limit (i.e., the maximum time during which a job can use a resource). Finally, we show that the semi-exclusive paradigm, which allows any SCC user to use idle buy-in resources for a limited time, increases the utilization of buy-in resources by 17.4%, thus significantly improving the performance of the system as a whole.**

I. Introduction

Computing clusters, connected computers that work together, are used by researchers in various fields. Due to the high demand for computing resources, characterizing the performance of computing clusters is essential. Statistical characterization of computing clusters helps assess the system's efficiency and improve the service provided to users of the cluster.

The *Shared Computing Cluster (SCC)* at Boston University (BU) [9] is of interest due to its implementation of a *shared/buy-in* architecture. Shared users access the cluster for free, while buy-in users purchase their own resources to which they get prioritized access. In addition, the SCC allows any user to utilize buy-in resources when they are idle, for a limited amount of time.

Our goals in this work are two-fold: (i) inform users in their decisions whether to buy-in resources or not; (ii) quantify the gains achieved by the shared/buy-in architecture.

Toward achieving the first goal, we perform a detailed statistical comparison of the waiting time experienced by shared and buy-in jobs. We identify several factors that play a key role, such as the run time limit, the type of parallel environment, and the amount of resource used by a given user or group of users. Toward achieving the second goal, we perform a detailed analysis of the utilization of the system as a whole, and of its different components. Our results show that the architecture balances out the workload more evenly than if idle buy-in resources were not shared.

The rest of this paper is organized as follows. We first introduce basic terminology and discuss related work. We then provide an overview of the architecture of the SCC. Next, we present a statistical characterization of the SCC, focusing on the waiting time of jobs and utilization of resources, before drawing our conclusions.

II. Background and Related Work

Before discussing related work, we introduce general terminology that applies to the SCC:

- A *job* is a programming task.
- The number of *slots* is the number of Central Processing Units (CPU) required for a job to run.
- A *queue* "is a logical abstraction that aggregates" a set of slots across one or more nodes [2]. In this context, a queue is not a waiting list.
- A *user* is an individual executing a job.
- A *project* is a group consisting of one or more users.
- The *run time limit* is the maximum amount of time a job can take to run before it is killed.

Many papers focus on *workload characterization*, "the science that observes, identifies and explains the phenomena of work in a manner that simplifies your understanding of how the network is being used" [9]. Ref. [1] asserts that workload studies typically focus on the usage of resources, characterization of the arrival process, and identification of system patterns. Computing clusters are grouped into two main categories: *grids*, heterogeneous clusters generally used in scientific and academic settings, and *clouds*, large homogeneous clusters generally used in commercial settings [3]. The Grid Workloads Archive (GWA) contains 12 workload traces of grid clusters [6]. In 2011, Google published a cloud workload trace, which is studied in [3, 4,5]. In relation to other clusters, the SCC is considered a grid-like computing cluster. Different theoretical methods of characterizing workloads to predict future usage and adjust existing scheduling policies are proposed in [7,8].

While our statistical characterization of the SCC includes similar analysis, our study contains the following novel aspects:

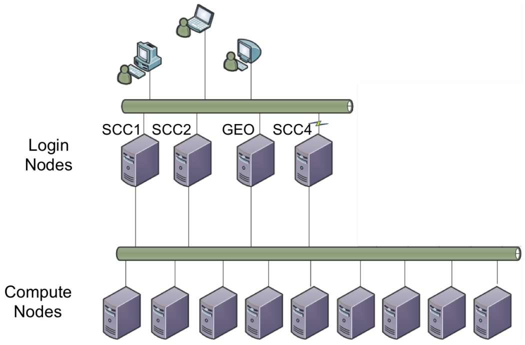

Figure 1. The SCC Architecture consisting of 4 login nodes and 467 compute nodes [10].

1) *Shared/buy-in*. To our knowledge, there have been no previous studies of a cluster that include shared and buy-in resources.
2) *Focus on waiting time*. We focus on the waiting time, an endogenous parameter, while most of the prior work above focus on exogenous parameters.

III. Cluster Architecture

The SCC at BU provides a complete environment for computational research and features a system that has shared and buy-in resources. Users using the shared part of the system do not pay for resources. In order to become a buy-in project, a project owner buys new buy-in nodes for the SCC and is granted priority to these nodes. Buy-in users benefit because they experience shorter waiting times on their own nodes as seen in section IV-A. Shared users also benefit since buy-in resources are reclaimed when they are not being utilized by its owner as explained in section III-A.

A user logs on to the SCC by accessing one of the 4 login nodes through a secure shell. *Login nodes* are used for light computing tasks. These nodes are not included in our analysis. Instead, we analyze the *compute nodes*, which allow users to submit computationally intensive jobs that generally run for a longer amount of time and require more resources. Figure 1 shows the relationship between the login nodes and compute nodes in the SCC.

A. System Overview

In the SCC, compute nodes are divided into two categories: shared and buy-in. In general, each buy-in node has two queues: a *buy-in queue* and a *public queue*. The buy-in queue only accepts jobs from a user of that specific buy-in project. The public queue accepts jobs from any user as long as the run time limit is 12 hours or less. The public queue for a node is disabled when a buy-in user submits a job on the SCC for that project. This policy prevents buy-in users from waiting for a long time for their own resources, but allows their resources to be utilized when idle. Shared resources only have one type of queue: *shared queues*.

Upon submitting a job, users are able to specify certain attributes for their jobs such as the run time limit, the number of slots, the parallel environment, and the number of GPUs. In section IV-A, we investigate how these requests affect the waiting time.

B. Scheduler

The SCC operates using the *Open Grid Scheduler*, an open-source scheduling software [2]. A scheduling round occurs every 15 seconds by default, but is also triggered by events such as job submissions and job completions. During each round, the scheduler assigns priorities to waiting jobs, sorts them accordingly, and allocates resources to jobs with the highest priorities. Priority is primarily affected by the number of slots requested and by the recent usage of the user submitting the job and other users belonging to the same project.

IV. Statistical Analysis

The SCC has been completely operational since July 2013. We analyze a 4.2 GB data trace collected from January 1, 2015 through July 20, 2016, using R and Python. In our analysis, *shared jobs* are defined as jobs that run on shared nodes, while *buy-in jobs* are defined as jobs that users of buy-in projects run on their own buy-in nodes.

A. Waiting Time

We define the waiting time of a job as the time elapsing from its submission till its start. We use the mean, median, and standard deviation as the main statistical measures of the waiting time. The median is not affected by outliers and is the purest measure of central tendency. The mean, on the other hand, takes into account the deviation of large outliers, which may be skewed for reasons listed in the sequel. We use the standard deviation as a means to quantify the variation of the distribution around the mean.

Table I shows that buy-in jobs tend to wait less than shared jobs, as the median waiting time for buy-in jobs is 0.56 hours, while the median waiting time for shared jobs is 1.2 hours. Additionally, for buy-in jobs, 8.8% of jobs wait more than 12 hours and 2.7% of jobs wait more than one week, while for shared jobs, 12.0% of jobs wait more than 12 hours and 3.0% of jobs wait more than one week. Thus, the waiting time distribution of shared jobs has a heavier tail than that of buy-in jobs.

1) Run Time Limit (RTL): As mentioned in Section III-A, users set the RTL when submitting a job. Table I shows that raising the RTL of a buy-in job over the default 12 hours has negligible effect on the median and mean waiting times. However, raising the RTL of a shared job over 12 hours increases the median waiting time from 1.1 to 1.7 hours, the mean from 4.6 to 10.6 hours, and the standard deviation from 10.1 to 36.2 hours. Hence the difference in performance between shared and buy-in jobs is much more significant once the RTL exceeds 12 hours.

2) Parallel Environment: There are two types of parallel jobs: *Open Multi-Processing (OMP)*, parallel jobs that run on one node, and *Message Passing Interface (MPI)* jobs, jobs that run on multiple nodes. 98.5% of parallel jobs are OMP jobs.

Table I
WAITING TIME COMPARISON (HOURS)

	Shared			Buy-in		
	Mean	Median	SD	Mean	Median	SD
Overall	6.21	1.21	20.90	4.99	0.56	18.27
RTL $\leq$ 12 hr	4.58	1.08	10.07	4.40	0.73	13.94
RTL $>$ 12 hr	10.60	1.71	36.18	4.67	0.60	18.07
Single-slot	6.12	1.28	20.83	4.80	0.79	16.03
OMP	6.53	0.74	18.90	5.28	0.12	21.81
MPI	15.62	0.58	53.70	20.33	0.02	72.85
Non-GPU	6.22	1.21	20.91	4.99	0.56	18.27
GPU	4.74	0.03	11.37	0.01	0.01	0.01

As shown in Table I, the difference in the median waiting time among single-slot, OMP, and MPI jobs is within one hour for both shared and buy-in jobs. However, the waiting time standard deviation of single-slot jobs is 20.8 hours for shared jobs and 16.0 hours for buy-in jobs, while the standard deviation of MPI jobs is 53.7 hours for shared jobs and 72.9 for buy-in jobs. Hence, submitting an MPI job increases the probability of waiting for a long time, especially for buy-in jobs.

3) GPU Requests: While GPU requests only account for 0.05% of the total number of jobs, they tend to have significantly shorter waiting time and is an example of why the need for a specific resource would be an incentive to become a buy-in user. For shared jobs, the median waiting time decreases from 1.2 to 0.03 hours when requesting GPUs. The standard deviation also drops from 20.9 to 11.4 hours, indicating a shorter tail. However, for buy-in jobs, the median waiting time decreases from 0.56 to 0.01 hours, while the standard deviation decreases from 18.3 to 0.01 hours. These statistics show that requesting GPU decreases the waiting time much more dramatically for buy-in jobs than for shared jobs.

4) User Usage: Figure 2 shows a positive correlation between a user's mean waiting time and the user's total usage of the SCC. The correlation coefficient is 0.39. We hypothesize that this correlation is due to the priority assigned by the scheduler. As alluded in Section III-C, the priority of a job is lowered if the user's recent usage is high, resulting in a longer waiting time. Therefore, if a user uses the SCC more frequently, his or her average waiting time would increase, which would also be an incentive to become a buy-in user.

We conclude this section by noting the following factors that may skew the waiting time data trace:

- If a user submits multiple jobs at once, the user may wait for his own jobs due to the 512 slot limit for users on shared resources.
- Users have the option of holding their own jobs after submitting them. There is no way to extract the time that users hold their own jobs from the waiting time.

B. Utilization and Workload

In this section, we analyze the utilization of different resources. We define the *workload* as the product of the number of slots used by a job and the job running time (in hours). The *workload capacity* is the workload achieved if all nodes are completely utilized, and the *actual workload* is the sum of the workload of all jobs that actually ran. The *utilization* is the fraction of the workload capacity taken up by the actual workload over a period of time.

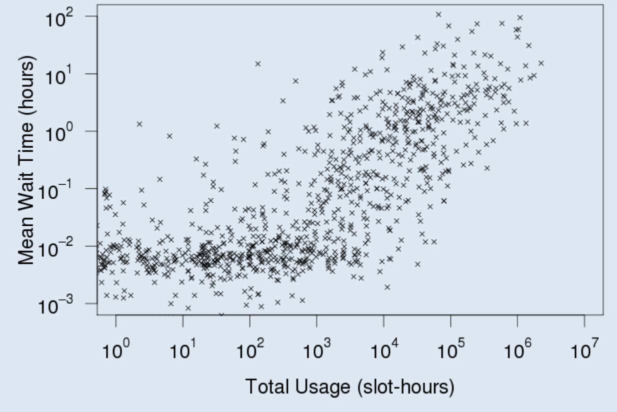

Figure 2. The mean waiting time of a user's job vs. the user's usage of the SCC.

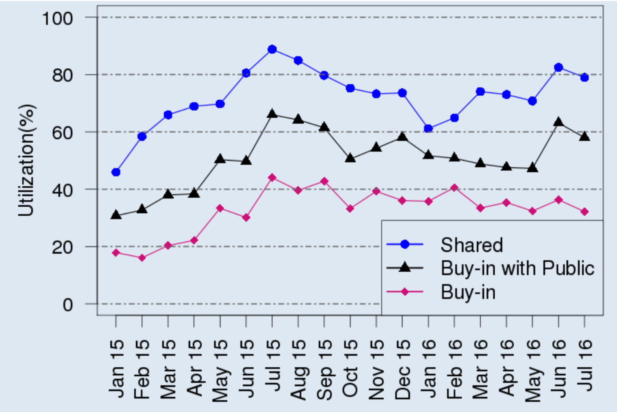

Figure 3. The average monthly utilization trend comparison between shared and buy-in parts of the SCC.

Over the analyzed period, the workload capacity is 3.45×10^7 slot-hours for shared nodes and 4.32×10^7 slot-hours for buy-in nodes. The actual workload on shared nodes is 2.43×10^7 slot-hours. For buy-in nodes, buy-in queues account for 1.42×10^7 slot-hours, while public queues account for 7.51×16^6 slot-hours. This shows that the workload capacity of shared nodes over the period is 79.9% of buy-in nodes, while their actual workload is 111.9% of buy-in nodes. Thus, workload is distributed more heavily on shared nodes than on buy-in nodes. Out of the 1,033 SCC users, 188 users account for 95.0% of the workload.

1) Public Queue: Figure 3 shows the monthly mean utilization. This shows that the shared utilization is above the buy-in utilization, even with public queues included. Figure 3 illustrates the benefits of implementing public queues on buy-in nodes. Without public queues, buy-in nodes would

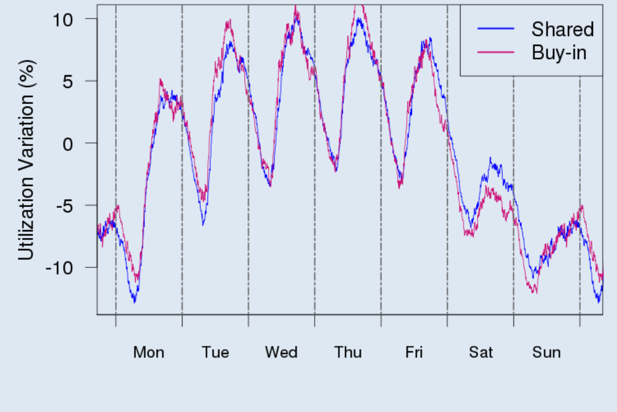

Figure 4. Weekly utilization pattern. Note that buy-in here refers to buy-in utilization with public queues included.

only have an average utilization of 32.9%. Public queues reclaim 17.4% of the workload capacity of buy-in nodes, resulting in an average utilization of 50.3% on buy-in nodes. By utilizing buy-in nodes when they are idle, public queues prevent demand for shared resources from exceeding their capacity and balance out the workload more evenly between shared and buy-in nodes.

2) Pattern: Figure 4 shows the weekly pattern of utilization. It shows that the system's utilization is lowest over the weekend and that the daily utilization peaks in the late afternoon. Shared and buy-in utilization exhibit the same weekly pattern. This is consistent with prior findings in the literature, since other grid workloads exhibit a regular arrival pattern while cloud workloads tend to exhibit a more random pattern [3].

V. Conclusion

Many prior studies focus on the efficiency and architecture of specific computing clusters. The SCC is different from most computing clusters due to its shared/buy-in configuration. From our statistical analysis of the SCC, we show that while the waiting time is generally shorter for buy-in jobs than for shared jobs, there are other factors involved. Requesting resources for the default of 12 hours results in similar mean waiting times for shared and buy-in jobs. However, for requests exceeding 12 hours, the mean waiting time of shared jobs is more than twice longer that of buy-in jobs. On the contrary, while MPI jobs have longer mean waiting times than one-slot jobs, for both shared and buy-in jobs, the increase in the standard deviation is more significant for buy-in jobs. Finally, the mean waiting time of a user's job increases significantly as a function of the users' total usage of the SCC. These results should provide useful guidelines to users whether to purchase buy-in resources or not.

Our analysis of the utilization pattern can help the SCC implement policies that maximize the system's utilization. For example, the SCC could give users a greater incentive to use the system over low utilization periods, such as weekends. One incentive could be to increase the maximum number of slots that a user can use over the weekend while decreasing the maximum number of slots over peak utilization periods. In addition, the SCC could weight a user's usage based on the current system utilization. If the system's utilization is high, the user will be "charged more" and vice versa. The implementation and evaluation of such policies represent interesting areas for future work.

Acknowledgment

This research was supported in part by NSF under grants CNS-1012798, CNS-1117160, and CNS-1414119, and by the Hariri Institute for Computing at BU. The authors would also like to acknowledge the Research Computing Services group at Boston University, including Glenn Bresnahan, Mike Dugan, and Katia Oleinik, for their guidance and technical support.

References

[1] M. Calzarossa, L. Massari, and D. Tessera, "Workload characterization: A survey revisited", *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 48, 2016.

[2] (2010). Beginner's guide to oracle grid engine 6.2, [Online]. Available: http://www.oracle.com/technetwork/oem/host-server-mgmt/twp-gridengine-beginner-167116.pdf.

[3] S. Di, D. Kondo, and W. Cirne, "Characterization and comparison of cloud versus grid workloads", in *Cluster Computing (CLUSTER), 2012 IEEE International Conference on*, IEEE, 2012, pp. 230–238.

[4] P. Garraghan, P. Townend, and J. Xu, "An analysis of the server characteristics and resource utilization in google cloud", in *Cloud Engineering (IC2E), 2013 IEEE International Conference on*, IEEE, 2013, pp. 124–131.

[5] F. Gbaguidi, S. Boumerdassi, E. Renault, and E. Ezin, "Characterizing servers workload in cloud datacenters", in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, IEEE, 2015, pp. 657–661.

[6] A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. H. Epema, "The grid workloads archive", *Future Generation Computer Systems*, vol. 24, no. 7, pp. 672–686, 2008.

[7] L. K. John, P. Vasudevan, and J. Sabarinathan, "Workload characterization: Motivation, goals and methodology", in *Workload Characterization: Methodology and Case Studies, 1999*, IEEE, 1999, pp. 3–14.

[8] A. Khan, X. Yan, S. Tao, and N. Anerousis, "Workload characterization and prediction in the cloud: A multiple time series approach", in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, IEEE, 2012, pp. 1287–1294.

[9] R. Lee, "An introduction to workload characterization", *Novell, http://support. novell. com/techcenter/articles/ana19910503. html*, 1991.

[10] (Jul. 2016). Research computing support, [Online]. Available: http://www.bu.edu/tech/support/research/.