GPPD–PCAD
HPC Resources Management Infraestruture Description and 10-month Statistics

Lucas Leandro Nesi, Matheus S. Serpa, Lucas Mello Schnorr, Philippe Olivier Alexandre Navaux
Institute of Informatics, Federal University of Rio Grande do Sul - UFRGS, Porto Alegre, Brazil

***Abstract*—Managing HPC resources for sharing its computational power among different users is a complex task and requires different software to assist its control. This paper describes the GPPD–PCAD cluster located at the Informatics Institute of the Federal University of Rio Grande do Sul. We present hardware infrastructure of the computational resources, including the interconnection among them. Moreover, we describe most of the software used to manage, control, and report the utilization of the resources, including the specific configurations used, and the versions used. We also present utilization statistics of the cluster using data provided by the resource manager. This information includes the resources utilization over time, most daytime use of the resources and the distribution of jobs sizes. We conclude our paper with some ideas to surpass the limitations we have identified for the current software infrastructure.**

I. Introduction

High-performance computing (HPC) has been responsible for a scientific revolution. The evolution of computer architectures improved the computational power, increasing the number of problems and quality of solutions solved in the required time, e.g., weather forecast. Furthermore, the industry has shifted its focus to parallel and heterogeneous architectures, which nowadays are part of all supercomputers.

The ability to share resources of a supercomputer or a cluster among many different users is a complex task that requires to establish some goals. Questions that may appear on the configuration of such environment are: should the users be able to choose the operating system of the machines? Should the users be able to install and control the installed software? How to effectively and fairly share the resources among the involved people? How to provide an efficient and global authentication system? There are different software with alternative goals to control such systems. Some influential examples are OAR [1], of the French Grid5000 [2], and Slurm [3], present in some Top500 supercomputers [4].

We present GPPD–PCAD[1], which stands for *Parque Computacional de Alto Desempenho* of the Grupo de Processamento Paralelo e Distribuído (GPPD) [2] of INF-UFRGS to share the knowledge gathered voluntarily during its configuration and to familiarize future users about the cluster structure. The main contributions of this paper are as follows. (a) We detail the resources infrastructure, including each node description and the intercommunication. (b) We present the software stack used in the system to manage, control, and supervise all the resources and users. (c) We also present some utilization statistics since the adoption of the current software solution and accountability.

Section II presents the hardware and software infrastructure. Section III details the 10-month utilization statistics metrics. Finally, Section IV concludes the paper with general perspectives about resource management with some future ideas for the software infrastructure of the cluster.

II. Infrastructure

A. Hardware

The resources available on GPPD–PCAD are presented on Table I, with the description of its CPU, Total memory RAM, the storage device, and the accelerator. The machines were gathered across different hardware generations and via a different number of projects and partnerships. To summarize the resources, the machines have Intel CPUs, and an NVIDIA GPU as an accelerator if present.

Multiple switches are used for two purposes. 1 – Internet connection. 2 – Internal services and experiments. One that connects all the resources to the building internet back-end, proving the DHCP service and connection across the campus machines and external users. Another switch, with 1Gb bandwidth, is used to isolate the communication of the machines. All the essential services use this secondary channel, including Slurm, NFS, and LDAP. When executing jobs, users can select one of the network interfaces to be used, and it is strongly suggested to use the dedicated switch to transfer and communicate computational data among the nodes.

B. Software

This Section describes the software stack used though the environment. It is essential to notice that some software choices and configurations are arbitrary and exploratory, without an extensive and exhaustive examination over it.

The operating system used across the cluster is Ubuntu [5], more specifically, Ubuntu server 18.04.02 LTS. Ubuntu is an open-source Operating System developed by Canonical Ltd that used the Linux kernel and is based on the Debian system. The choice regarding its utilization comes from its widespread adoption, where many users are familiar with the system.

Slurm [3] (Simple Linux Utility for Resource Management) is an Open-source job scheduler for Linux-based systems

[1]The GPPD–PCAD website: http://gppd-hpc.inf.ufrgs.br/
[2]The GPPD website: http://www.inf.ufrgs.br/gppd/site/

TABLE I
COMPUTATIONAL RESOURCES AVAILABLE ON GPPD–PCAD

Machine	CPU	RAM	Storage	Accelerator
gppd-hpc	2 x Intel Xeon E5-2630	16 GB DDR3	11 TB	
knl[1-4]	Intel Xeon Phi 7250	96 GB DDR4	440 GB	
blaise	2 x Intel Xeon E5-2699 v4	256 GB DDR4	1.6 TB	4 x NVIDIA Tesla P100
tupi1	Intel Xeon E5-2620 v4	64 GB DDR4	440 GB	
tupi2	Intel Xeon E5-2620 v4	80 GB DDR4	3.6 TB	2 x NVIDIA GeForce GTX 1080Ti
hype[1-3]	2 x Intel Xeon E5-2650 v3	128 GB DDR4	550 GB	
hype[4-5]	2 x Intel Xeon E5-2650 v3	128 GB DDR4	550 GB	2 x NVIDIA Tesla K80
orion1	2 x Intel Xeon E5-2640 v2	48 GB DDR3	916 GB	1 x NVIDIA Tesla K20m
orion2	2 x Intel Xeon E5-2640 v2	32 GB DDR3	916 GB	2 x NVIDIA Tesla K20m
draco[1-6]	2 x Intel Xeon E5-2630	64 GB DDR3	1.8 TB	1 x NVIDIA Tesla K20m
draco7	2 x Intel Xeon E5-2630	128 GB DDR3	1.8 TB	2 x NVIDIA Tesla K20m
bali[1-2]	2 x Intel Xeon E5-2650	32 GB DDR3	916 GB	
beagle	2 x Intel Xeon E5-2650	32 GB DDR3	916 GB	
turing	4 x Intel Xeon X7550	128 GB DDR3	3.6 TB	

created by LLNL (Lawrence Livermore National Laboratory). Currently, Slurm is deployed at many supercomputers around the world including the Brazilian super computer SDumont[3]. Slurm permits users to allocate and deploy batch jobs to a series of resources guaranteeing that specific resources, like CPUs, memory, or entire nodes, are correctly shared among users and jobs without erroneous interaction. Slurm is adopted on PCAD with version 18.08.5-2. Primarily, each set of equal resources (Draco machines, for example) are aggregated on a partition of resources that have an individual queue for submissions of jobs. Users can send jobs and allocate entire nodes of different partitions. PCAD also have a `shared` queue for allocation of CPUs instead of entire nodes. Multiple jobs can then be executed at the same time on the same node.

LDAP [6] (Lightweight Directory Access Protocol) is an Open-source application protocol for accessing and maintaining distributed directory information services over an IP network. A common use of LDAP is to provide a central place to store usernames and passwords. This allows many different applications and services to connect to the LDAP server to validate users. LDAP is used on PCAD as the central authentication system. The users are not individually registered on each resource, but, when authentication is required, the node will communicate to the LDAP server on the PCAD front-end node to validate the authentications credentials.

PAM [7] (Pluggable authentication modules) is a mechanism to integrate multiple low-level authentication schemes into a high-level application programming interface (API). It allows programs that rely on authentication to be written independently of the underlying authentication scheme. In PCAD, PAM enables Slurm to guarantee the exclusive access on the resources for only users that have jobs executing on it. It is also used to validate the authentication with private keys instead of passwords (The default approach of LDAP).

Ganglia [8] is a distributed monitoring tool for high-performance computing systems, clusters, and networks. The software is used to view either live or recorded statistics covering metrics such as CPU and network utilization for many nodes. Every PCAD resource has the ganglia client. The overall monitor is publicly available on PCAD's website [4].

NFS [9] (Network File System) is a distributed file system protocol originally developed by Sun Microsystems, allowing a user on a client computer to access files over a computer network much like local storage is accessed. PCAD uses NFS to provide the users a directory on all nodes. The NFS server is on the PCAD front-end and has a total of 12TB for all users.

Another critical aspect of managing resources is security. PCAD uses two software to deal with this aspect. The Snoopy library [10] capable to log all commands and arguments on a system. And the Fail2ban [11], that is an intrusion prevention software framework that protects computer servers from brute-force attacks. Written in the Python programming language, Fail2ban can run on POSIX systems that have an interface to a packet-control system or iptables.

The time of the different machines is synchronized in PCAD using NTP [12] (Network Time Protocol). It is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. In operation since before 1985, NTP is one of the oldest Internet protocols in current use.

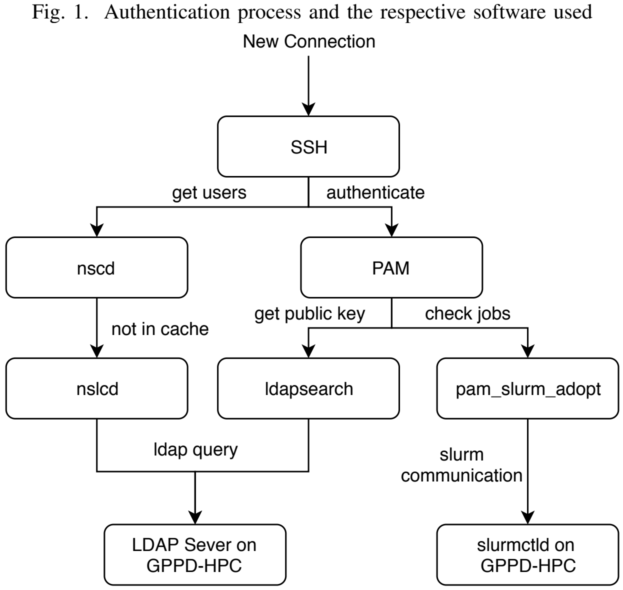
User Interaction and Resource Reservation: The job submission is carried out accessing the PCAD front-end using `SSH` and either using `salloc` or `sbatch` Slurm commands. The commands require to specify a machine's partition and the total expected time (maximum execution time) for the job. The majority of the partitions have a time limit of three days. The global view of some of the used software's interaction can be viewed when inspecting the authentication process. The authentication flow is presented in Figure 1. Basically, when a new connection is made to a resource node or the front end via (and only via) `ssh` a series of software are consulted to validate the authentication. First, SSH needs to check if the user exists and get common Linux information, including user's groups and location folder. This is made thought name service requests cache process `nscd`. In case the requested user is not present in the cache, `nscd` will consult `nslcd`

[3]The SDumont website: https://sdumont.lncc.br/

[4]The PCAD glanglia link: http://gppd-hpc.inf.ufrgs.br/ganglia/

that will query the LDAP server on GPPD–PCAD. If the user is found, the SSH now needs to confirm the credentials using the user public key. To get this information, SSH uses PAM to execute a bash script and query the LDAP server to return the key. In case the key is valid, the user is authenticated, and now PAM needs to confirm if the user can access the machine, i.e., if there is a Slurm job running on it (This step is only executed on the resource nodes and not in the front-end). Slurm provided a PAM module, pam_slurm_adopt, that will consult if the user has jobs running on the machine. In case the user has a job, PAM will validate the user and grant the access. If in any part of this process an invalid operation is done the connection is closed.

Fig. 1. Authentication process and the respective software used



III. Utilization Statistics

This Section presents resource utilization since the implementation of the current managing approach. The data of jobs executed was collected using Slurm, starting at 2018-10-23 and ending on 2019-08-26. There were a total number of jobs of 292836 by 55 different users. Different aspects can be used to present the resources utilization. Figure 2 presents the density of the size of the jobs in hours, where on the X-axis there is the size of the job in hours and the Y-axis the total density computed by `stat_density` R function. It is possible to check the majority of jobs are smaller than three hours, with a slight pick on 12 hours. Also, the number of jobs with size greater than 24 hours is small.

Figure 3 presents the daytime utilization of the partitions. Where the X-axis is the hour of daytime (24 hours) and in the Y-axis there is the total number of jobs that were executed in that time instant. The data show that there is a pick of utilization during work hours (8:00-17:00) when the count of executing jobs is increasing. Moreover, during the night, the resources utilization is gradually decreasing, probably justified because the jobs begin to end.

Fig. 2. Density of jobs size in hours (of jobs with duration greater than 2 minutes) in different partitions

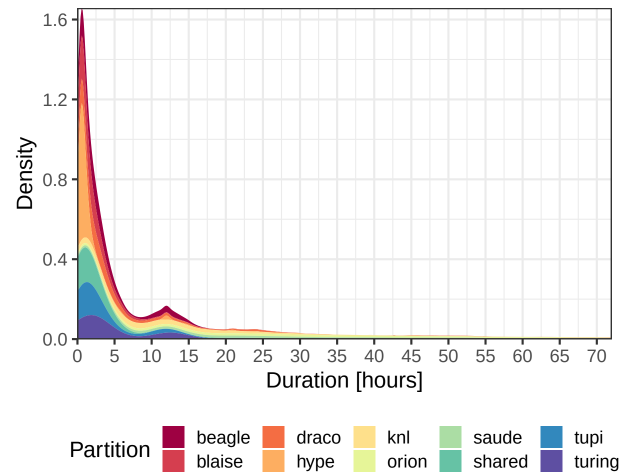


Fig. 3. Daytime utilization of the resources partition (of jobs with duration greater than 2 minutes)

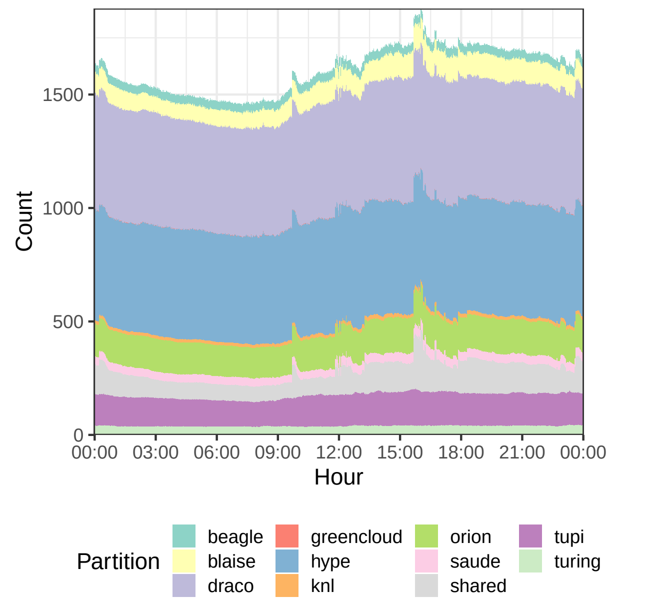
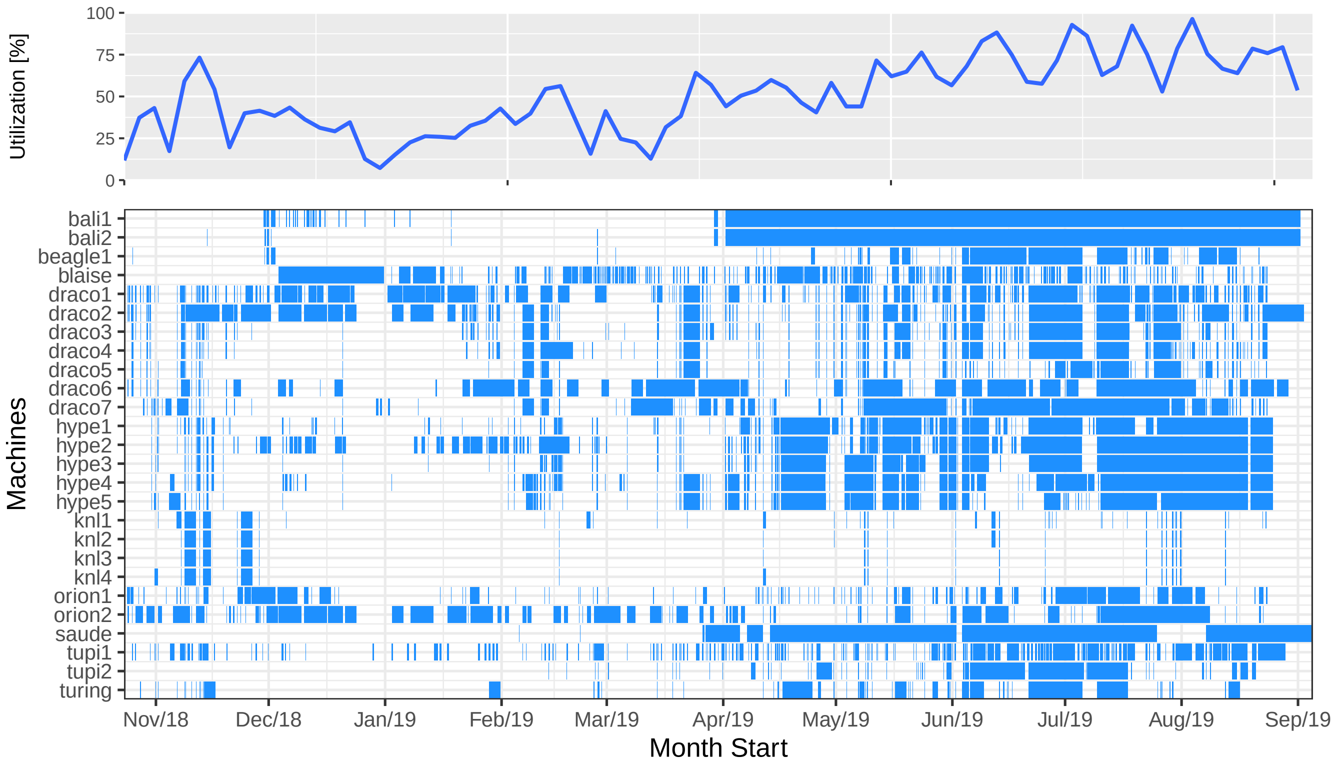


Figure 4 presents the occupation of the resources by jobs over time since the installation of the system management in October 2018 until the end of August 2019. The top panel presents the total utilization of the resources per day. The bottom panel presents the utilization per resource. The vertical red line represents the day that the data was taken. With the aid of this visualization, it is possible to check the periods of more utilization, in the same way, it gives a notion of the most utilized resources. For example, the `knls` machines are the lesser utilized resources, while the `hypes` are been used extensively during the last months. From the data, it is also possible to notice that the resources are being more utilized in 2019 when compared to 2018.

Fig. 4. Machines Utilization



IV. Conclusion

This paper presented a study case of managing HPC resources thought new and modern software solutions that permit hardware sharing with different users. The hardware infrastructures are presented together with the software stack used. Also, we present some real utilization data of the resources. First, we present the hours that the machines were most utilized, that is the work hours. Second, we present the distribution of jobs size submitted; the majority is smaller than two hours. Third, the Gantt chart diagram of individual resource utilization, showing the increased use of the resources in the last months. Future work on PCAD includes the study of different batch schedulers like OAR and enabling users to deploy their custom operating system images. Also, other solutions to report utilization can be used to show PCAD's user behavior.

Acknowledgements

This study was financed by the "Coordenação de Aperfeiçoamento de Pessoal de Nível Superior" (CAPES) - Finance Code 001, the National Council for Scientific and Technological Development (CNPq), and the projects: FAPERGS MultiGPU (16/354-8), FAPERGS GreenCloud (16/488-9), FAPERGS Internacionalização (19/711-6), the CNPq project 447311/2014-0, the CAPES/Brafitec EcoSud 182/15, the CAPES/Cofecub 899/18, Petrobras (2016/00133-9 and 2018/00263-5), and donations by HPE and Intel under the *Lei da Informática* grants.

References

[1] N. Capit, G. Da Costa, Y. Georgiou, G. Huard, C. Martin, G. Mounié, P. Neyron, and O. Richard, "A batch scheduler with high level components," in *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, 2005.*, vol. 2. IEEE, 2005, pp. 776–783.

[2] D. Balouek, A. Carpen Amarie, G. Charrier, F. Desprez, E. Jeannot, E. Jeanvoine, A. Lèbre, D. Margery, N. Niclausse, L. Nussbaum, O. Richard, C. Pérez, F. Quesnel, C. Rohr, and L. Sarzyniec, "Adding virtualization capabilities to the Grid'5000 testbed," in *Cloud Computing and Services Science*, ser. Communications in Computer and Information Science, I. I. Ivanov, M. van Sinderen, F. Leymann, and T. Shan, Eds. Springer International Publishing, 2013, vol. 367, pp. 3–20.

[3] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer, 2003, pp. 44–60.

[4] TOP500, "TOP500 statistics list," https://www.top500.org/statistics/list/, 2018, accessed: 2019-05-24.

[5] Canonical, "Ubuntu," https://ubuntu.com/, 2019.

[6] K. Zeilenga, "Lightweight directory access protocol (ldap): Technical specification road map," 2006.

[7] P. A. M. PAM, "Unified login with pluggable authentication modules (pam)," 1995.

[8] M. L. Massie, B. N. Chun, and D. E. Culler, "The ganglia distributed monitoring system: design, implementation, and experience," *Parallel Computing*, vol. 30, no. 7, pp. 817–840, 2004.

[9] B. Nowicki, "NFS: Network file system protocol specification," 1989.

[10] M. Eriksen and M. Baker, "Log every executed command to syslog (a.k.a. snoopy logger)," 2010. [Online]. Available: https://github.com/a2o/snoopy

[11] C. Jacquier, "Fail2ban," 2010. [Online]. Available: https://www.fail2ban.org/wiki/index.php/Main_Page

[12] D. Mills, "Network time protocol (version 3) specification, implementation and analysis," 1992.