Advances in GPPD–PCAD Management with 12-months Analysis and Perspectives

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*—High-Performance computing resources are the basis for many projects, and a shared utilization is the only way to be efficient in the allocation by the users. Managing these resources and enabling a shared use is complex and requires configurations in many levels of the hardware and software stacks. Although different modern tools exist to assist in this task, it remains complex and requires many infrastructure decisions that must cope with the max possible utilization cases. This paper presents the last advances in GPPD–PCAD resource management over the past year. We report the new machines, software changes, and the adoption of new tools to enable users and administrators a better experience. We also present an analysis of the 12 months from August/2019 to August/2020 of resource utilization that focuses on compute nodes and end-users. In this analysis, we offer a general user behavior that includes infrastructure utilization during the COVID-19 pandemic.**

I. Introduction

High-Performance computing resources enable many users to execute their resource-intensive applications [1]. The management of these resources is complex and requires many configurations, sometimes tailored for specific vendor components. Technologies do exist to share these resources across many users, and effectively guarantee that everyone can use them. Examples of such tools are Slurm [2] and OAR [3], of the French Grid5000 [4]. Moreover, the software stack of these platforms is complex, and tools that automatize configuration are desirable. Many configuration management (CM) software exist, such as Ansible [5], Jenkins [6], and Puppet [7].

We present the last year's advances in GPPD–PCAD[1], which stands for *Parque Computacional de Alto Desempenho* of the Grupo de Processamento Paralelo e Distribuído (GPPD)[2] of INF-UFRGS. The last GPPD–PCAD report [8] presented the cluster and initial configuration. This paper focuses on the new advancement related to that publication until today. The significant contributions are as follows: (i) the extension of the cluster with six new machines; (ii) changes in the software stack to enable a better experience, by including virtualization with Docker [9] and Singularity [10]; utilization of Ansible and IPMI (Intelligent Platform Management Interface) [11] to configure and maintain the cluster updates; and employment of BTRFS (B-tree file system) [12] on the main NFS partition; (iii) the analysis of the last 12-months, discussing how it is the general job submission configuration, resource utilization, and the behavior during the COVID-19 pandemic.

Section II presents the hardware highlighting the new machines. Section III details the alterations made in the software stack from the last report [8]. Section IV analyses the 12-months (August 2019 to August 2020) resource utilization with different statistics. Finally, Section V concludes the paper with future perspectives and ideas to be implemented.

Also, we highlight that all the knowledge gathered in the process of building and maintaining GPPD–PCAD is registered in an internal repository for future aspirants. We find relevant the know-how of managing HPC resources and think that it may be useful for prospective students that will maintain it. Students interested in participating may contact the team [3].

II. Hardware Infrastructure

The GPPD–PCAD infrastructure is compost of 33 nodes, four racks, and is interconnected by two networks. First, a Gigabit Ethernet for internal communication (Slurm, NFS, ssh). Second, a 100 Megabit Ethernet that is accessible via the main site (UFRGS-INF) network. Each node has at least one disk split into three partitions, 100GB for the system, 2GB for swap and the rest for scratch user space. Also, most of the machines have IPMI interfaces that vary per vendor. In some cases, the IPMI interface is also connected to the site or internal network depending on individual characteristics.

All the resources are present in Table I with machine name, CPU, RAM, Disk space, and Accelerators. The new additions are marked with a star symbol (*). These new machines comprehend new vendor architectures (`apolo` and `sirius` with AMD CPU, and `tsubasa` with NEC SX-Aurora Vector Engine). Also, GPPD–PCAD provides its managing system to machines of other INF-UFRGS research groups, including `saude`, `cidia`, and `cei[1-2]` machines.

III. Software Stack and Managing Tools

The GPPD–PCAD infrastructure uses standard and modern tools to manage its resources [8]. Though, over time, it is expected that alternatives or extra tools start to be utilized as we intend to keep the system updated with the most recent and capable tools. This Section presents the software stack changes of GPPD–PCAD and the new tools employed in this last year to solve and automatize many tasks. First, the operational system of the computing nodes was changed to Debian 10. We believe that this system is more aligned with general users' Linux knowledge and does not force the presence of extra software

[1]The GPPD–PCAD website: http://gppd-hpc.inf.ufrgs.br/
[2]The GPPD website: http://www.inf.ufrgs.br/gppd/site/
[3]GPPD–PCAD administration team e-mail: gt-cluster-l@inf.ufrgs.br

TABLE I
COMPUTATIONAL RESOURCES AVAILABLE ON GPPD–PCAD

Machine	CPU	RAM	Storage	Accelerator
gppd-hpc	2 x Intel Xeon E5-2630	16 GB DDR3	22 TB	
apolo*	AMD Ryzen 5 3400G	48 GB DDR4	480 GB	AMD Radeon Vega 11
bali1	2 x Intel Xeon E5-2650	32 GB DDR3	1 TB	
bali2	2 x Intel Xeon E5-2650	32 GB DDR3	4 TB	
beagle	2 x Intel Xeon E5-2650	32 GB DDR3	1 TB	
blaise	2 x Intel Xeon E5-2699 v4	256 GB DDR4	4 TB	4 x NVIDIA Tesla P100-SXM2
cei[1-2]*	2 x Intel Xeon Silver 4116	96 GB DDR4	13 TB	
cidia*	2 x Intel Xeon Silver 4208	320 GB DDR4	10 TB	2 x NVIDIA GeForce RTX 2080 Ti
draco[1-6]	2 x Intel Xeon E5-2640 v2	64 GB DDR3	2 TB	1 x NVIDIA Tesla K20m
draco7	2 x Intel Xeon E5-2640 v2	128 GB DDR3	1 TB	2 x NVIDIA Tesla K20m
hype[1-3]	2 x Intel Xeon E5-2650 v3	128 GB DDR4	600 GB	
hype[4-5]	2 x Intel Xeon E5-2650 v3	128 GB DDR4	600 GB	2 x NVIDIA Tesla K80
knl[1-4]	Intel Xeon Phi 7250	96 GB DDR4	480 GB	
orion1	2 x Intel Xeon E5-2630	48 GB DDR3	1 TB	NVIDIA Tesla K20m
orion2	2 x Intel Xeon E5-2630	32 GB DDR3	1 TB	NVIDIA Tesla K20m
saude	Intel Xeon CPU E5-2620 v4	128 GB DDR4	8 TB	4 x NVIDIA GeForce GTX 1080Ti
sirius*	AMD Ryzen 9 3950X	64 GB DDR4	240 GB	NVIDIA GeForce GT 1030
tsubasa*	2 x Intel Xeon Gold 6226	192 GB DDR4	2 TB	4 x NEC SX-Aurora TSUBASA
tupi1	Intel Xeon E5-2620 v4	64 GB DDR4	2.5 TB	2 x NVIDIA GeForce GTX 1080Ti
tupi2	Intel Xeon E5-2620 v4	80 GB DDR4	5.2 TB	
turing	4 x Intel Xeon X7550	128 GB DDR3	5 TB	

of the distribution's organization. Another difference is the revisited Slurm job's priority configuration. Now, we utilize the global and partition components to better schedule tasks by various users (switching faster) and not by submission order.

Moreover, the management of the resources became more accessible with the utilization of IPMI. This feature also highlights the heterogeneity of the resources not only in computational hardware but in this interface. The nodes are from at least three different major vendors with their customs management system that integrates with IPMI. We took a close look at each one to configure them better for our needs. This configuration showed significant importance during the COVID-19 pandemic as the physical access to the infrastructure site was severely limited.

Furthermore, the NFS partition file system present on the front-end node was changed from ext4 to BTRFS [12]. BTRFS (B-tree file system) is a copy-on-write file system currently being developed in the mainline Linux kernel that presents many desirable features. For us, the most important one is that it enables transparent compression for files and can be configured with different algorithms (We use `lzo`). It automatically detects which files would benefit from compression and applies it. Currently, in the 22TB RAID NFS partition present on the front-end, there is a total of 16TB data but only 10TB physical space being used, a saved of 6TB by BTRFS compression. Although BTRFS compression may present a little overhead with some workloads [13], the NFS partition should not be used by time-sensitive applications, as it already has the network overhead. The NFS partition's main objective is to enable a global file-system across all nodes for medium/long period files, typical present in the applications' start or end. Temporary files present during execution should use the intra-node storage (`/scratch`). Other exciting features that BTRFS presents are system snapshots, space-efficient packing of small files, and sub-volumes.

The management of all these resources requires much automatization, and different modern tools exist with this goal, including Jenkins [6], Puppet [7], and Ansible [5]. We select Ansible [5] to use in GPPD–PCAD, as other supercomputers use it for CM (Configuration Management). The basic structure of Ansible compounds hosts (the nodes) and playbooks (the scripts), where a playbook can be performed on a set of nodes. We mainly utilize Ansible to keep machines updated and guarantee the last configuration files on them. These actions are performed recurrently and automatically. Though, if a node is being used (allocated), a Slurm job will be created to respect and not interfere with that user allocation.

We also highlight the availability of two newly installed software, the virtualization tools Docker [9] and Singularity [10]. Because the cluster has permission limitations over some software, including the Linux kernel, the users can virtualize their own system and make modifications as they please, considering their limitations. These changes provide users more freedom and do not require administration interference or software installation requiring admin-level permissions.

IV. RESOURCE UTILIZATION ANALYSIS

This Section presents general resource utilization statistics of GPPD–PCAD from 2019-08-01 to 2020-08-26 with 82713 jobs submitted by 80 users. Currently, GPPD–PCAD already processed 352000 jobs (all durations) with more than 100 registered users. Similar to the last report [8], we present the density of jobs duration in hours in Figure 1. The X-axis is the duration of jobs in hours, and the Y-axis is the total density computed by `stat_density` of the `ggplot2` R package function. The majority of the jobs take less than 3 hours to conclude, with remarkable peaks in half-hour and 2 hours. Also, there are small peaks in 5, 6.5, and 12 hours. Small jobs are a desirable situation as the batch management system can schedule better the jobs of different users across the resources. The duration of the jobs is directly correlated to users' allocation choices and their applications' behavior.

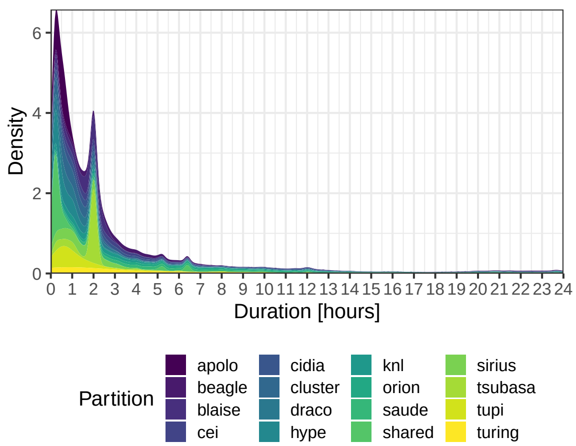

Fig. 1. Jobs duration density in hours (of jobs with a duration greater than 2 minutes) in GPPD–PCAD 16 partitions.

Another interesting metric is the resource utilization across day hours (0-24h). Figure 2 presents this metric (Number of jobs per inner day hours) for the 16 partitions. The X-axis is the daytime (Aggregated per minute), and the Y-axis is the total amount of jobs that were executed during that minute. Like last year's results [8], the day starts (00:01) with the number of jobs decreasing until more or less 8:00 when it starts to grow up again. The number of jobs then reaches a peak at 16:00~17:00 and dropping again until the end of the day. This reflects the regular work hours of the users. Most jobs are submitted between 8:00-17:00 (explaining the increasing). There are few jobs submitted at night (18:00-07:00), while the remaining jobs still are finishing (explaining the decreasing).

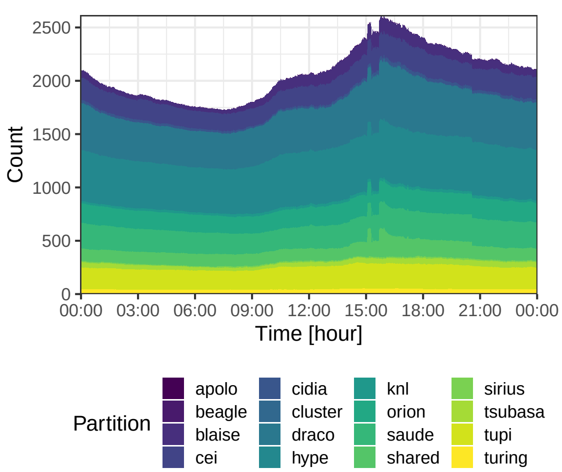

Fig. 2. Utilization of the resources by day minute (of jobs with a duration greater than 2 minutes) in GPPD–PCAD partitions.

Per-user utilization is also another critical metric. Figure 3 shows the cumulative node-hours utilization for the top 30 users in the analyzed period. The utilization of the first four users outlines from the others. Also, users 3 and 4 only use one partition. From rank five to 30, the utilization smoothly drops with utilization in all partitions. This also shows that most users prefer to use the same partitions over time, in general, not using more than four different ones.

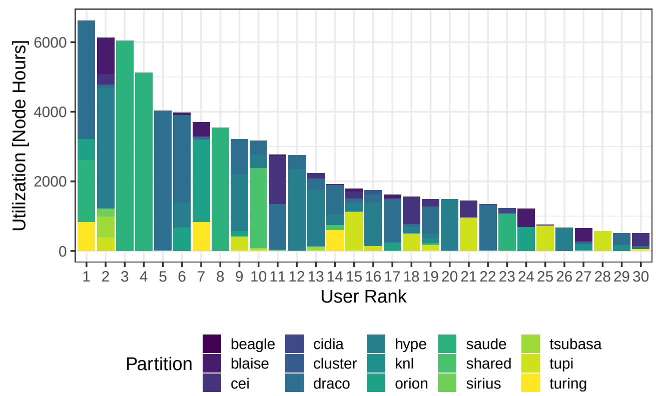

Fig. 3. Resource utilization (Node hours) per partition of the 30th top users.

Figure 4 presents the utilization per machine in the last 12 months (August/2019 to August/2020) in the X-axis. The top panel presents the % of utilization on that day in the cluster. The bottom panel shows a typical Gantt chart per individual resource (Y-axis) where states are running jobs (allocations) at that moment. In this Figure, a resource is utilized if there is at least one job on that day. The black line presents the actual values, while the blue line is the smooth curve computed by `geom_smooth` of the `ggplot2` R package with the `gam` method. Regarding machine utilization, some clusters are much busier than others. For example, `dracos` and `hypes` are intensively utilized, while specific and exotic architectures like the Intel Xeon Phi on `knls` present low utilization. Some machines were integrated into the GPPD–PCAD during the year, so in the plot, there are long non-utilized areas where these machines are absent (`Sirius`, for example).

Moreover, there are two moments highlighted in red areas that change the behavior of GPPD–PCAD utilization that occurred because of external events. First, at the end of January and the beginning of February, along with electrical maintenance, shut down the infrastructure for a few days. Second, in the middle of March, it started the COVID-19 pandemic restrictions, impacting the physical access and utilization of the GPPD–PCAD at our university (UFRGS). A drop in the resources' utilization is perceptive for the first days, but it rapidly returned to normal values.

V. Conclusion

This paper presented the advances in hardware and software of the GPPD–PCAD resource management. The six new compute nodes were shown along with the following improvements in the software stack: (i) remote management and

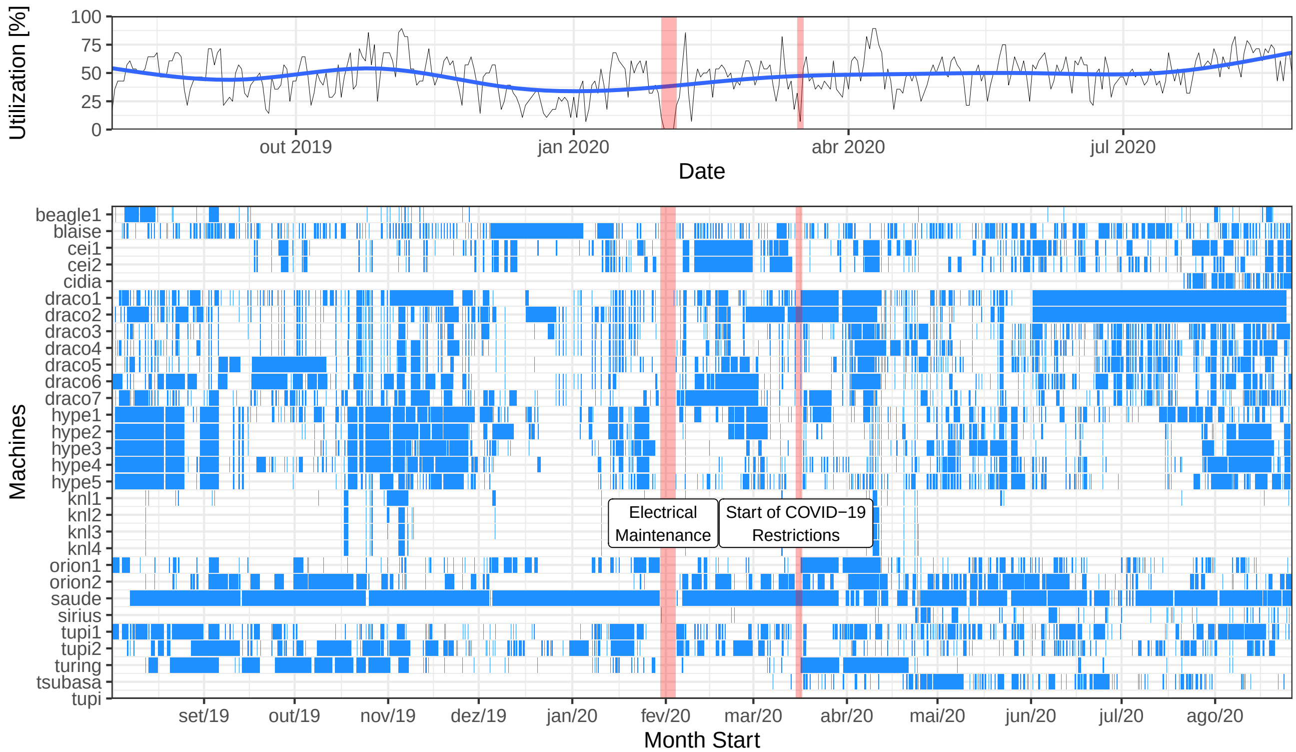

Fig. 4. Cluster aggregated total utilization and per machines Gantt chart jobs.

monitoring using the IPMI; (ii) software provisioning and configuration with Ansible; (iii) virtualization with Docker and Singularity; (iv) B-tree file system, which transparent compress data. Besides, we analyze the utilization during the last year, highlighting the effects throughout the COVID-19 pandemic. Future work includes evaluating new tools and more fine-grained analysis, like the number of jobs that utilize virtualization and energy consumption.

For more information about the GPPD–PCAD, please contact the team at gt-cluster-l@inf.ufrgs.br.

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] J. Dongarra, S. Tomov, P. Luszczek, J. Kurzak, M. Gates, I. Yamazaki, H. Anzt, A. Haidar, and A. Abdelfattah, "With extreme computing, the rules have changed," *Comp. in Sci. Eng.*, vol. 19, no. 3, p. 52, 2017.

[2] 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.

[3] N. Capit *et al.*, "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.

[4] D. Balouek *et al.*, "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.

[5] L. Hochstein and R. Moser, *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*. " O'Reilly Media, Inc.", 2017.

[6] V. Armenise, "Continuous delivery with jenkins: Jenkins solutions to implement continuous delivery," in *2015 IEEE/ACM 3rd International Workshop on Release Engineering*. IEEE, 2015, pp. 24–27.

[7] J. Loope, *Managing infrastructure with puppet: configuration management at scale*. " O'Reilly Media, Inc.", 2011.

[8] L. L. Nesi, M. S. Serpa, L. M. Schnorr, and P. O. A. Navaux, "GPPD – PCAD - HPC resources management infrastructure description and 10-month statistics," in *XVII Workshop de Processamento Paralelo e Distribuido*. GPPD, 2019, pp. 21–24.

[9] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux journal*, vol. 2014, no. 239, p. 2, 2014.

[10] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PloS one*, 2017.

[11] *IPMI-Intelligent Platform Management Interface Specification Second Generation*, Intel, Hewlett-Packard, NEC and DELL, 10 2013, Rev 1.1.

[12] O. Rodeh, J. Bacik, and C. Mason, "BTRFS: The linux B-tree filesystem," *ACM Transactions on Storage*, vol. 9, no. 3, pp. 1–32, 2013.

[13] M. Larabel, "Btrfs Zstd compression benchmarks on linux 4.14 - Phoronix," https://www.phoronix.com/scan.php?page=article&item=btrfs-zstd-compress&num=1, 2017, accessed: 2020-09-14.