Optimisation of the usage of LHC and local computing resources in a multidisciplinary physics department hosting a WLCG Tier-2 centre

Optimisation of the usage of LHC and local computing resources in a multidisciplinary physics department hosting a WLCG Tier-2 centre

Stefano BARBERIS, Leonardo CARMINATI, Franco LEVERARO, Simone Michele MAZZA, Laura PERINI, Francesco PRELZ, David REBATTO, Ruggero TURRA, Luca VACCAROSSA & Miguel VILLAPLANA

Università degli Studi e INFN Milano, via Celoria 16, Milano, 20133, Italy

E-mail: `Miguel.Villaplana@mi.infn.it`

Abstract. We present the approach of the University of Milan Physics Department and the local unit of INFN to allow and encourage the sharing among different research areas of computing, storage and networking resources (the largest ones being those composing the Milan WLCG Tier-2 centre and tailored to the needs of the ATLAS experiment). Computing resources are organised as independent HTCondor pools, with a global master in charge of monitoring them and optimising their usage. The configuration has to provide satisfactory throughput for both serial and parallel (multicore, MPI) jobs. A combination of local, remote and cloud storage options are available. The experience of users from different research areas operating on this shared infrastructure is discussed. The promising direction of improving scientific computing throughput by federating access to distributed computing and storage also seems to fit very well with the objectives listed in the European Horizon 2020 framework for research and development.

1. Introduction

Currently hosting around 400 active users, the University of Milan Physics Department and the local unit of INFN form one of the largest conglomerations of physics research groups in Italy. The local computing infrastructure has already surpassed 2 PB of storage and has a CPU power of nearly 30000 HS06. The distribution of CPU resources among research groups can be seen in Figure 1.

The ATLAS computing model [1] groups the different types of computing centers of the ATLAS Collaboration in a tiered hierarchy that ranges from Tier-0 at CERN, down to the 11 Tier-1 centers and the nearly 80 Tier-2 centres distributed world wide. University of Milan hosts an ATLAS Tier-2 and a Tier-3 center, which provide 90% of the storage and 67% of the computing power the facility maintains.

With the support of a scientific research program considered of relevant national interest by the Italian government (PRIN), we aim at enabling efficient interactive analysis, suitable for locally sharing the same resources used for Grid work on the Tier-2 center, and implementing methods for the best use of the storage and data available for each group.

A further important objective of this project is the transfer of computing technology developed for the LHC experiments to physics areas outside the high energy physics (HEP)

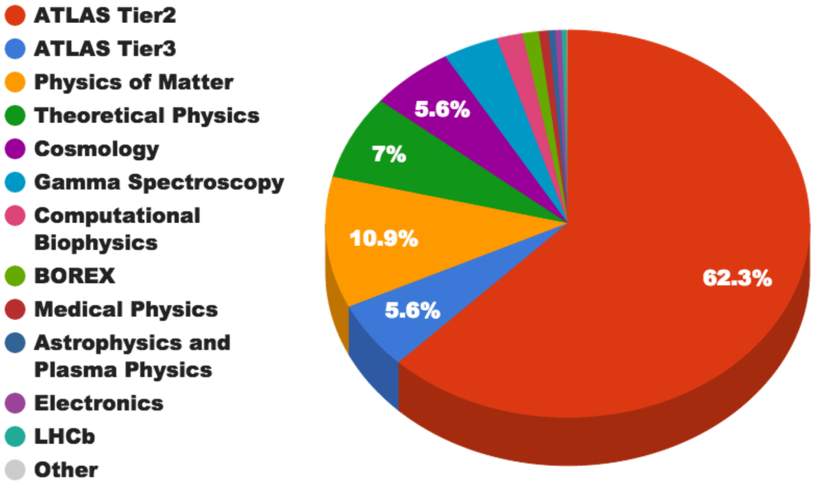

Figure 1. Distribution of CPU resources among the different research groups measured according to the HEP-SPEC06 benchmark.

domain, consistently with the objectives of the Horizon 2020 ”Excellent Science” plan. In this way we aim at creating user communities with uniform computing services at the departmental level and ability to use distributed computing resources. In perspective these aggregations should be able to extend beyond physics to other sciences.

2. Share resources across groups

Our approach relies on HTCondor [2], a specialised workload management system for compute-intensive jobs, for several reasons. It is open source software maintained by the Center for High Throughput Computing at UW-Madison. It provides a job queueing mechanism, a scheduling policy, a priority scheme, resource monitoring, and resource management. In many circumstances, it can checkpoint and migrate a job to a different machine when needed, and flocking technology allows multiple HTCondor pools to work together. Since there is a user community which cannot rely on a local Tier-2 or Tier-3 computing facility, we use HTCondor to uniform and standardise the way interactive resources of any local deployment might be accessed by every user.

The different units are encouraged to organise their resources under HTCondor pools, where execute machines report to the central manager of the corresponding pool. In addition to this, we add a new central manager to which all execute machines also report to as it is shown in Figure 2. This provides usage accounting across all the resources together and serves as a top-level pool to submit jobs to when users want to access all possible resources. Group pools remain the default pool for job submission, but with the super-pool added to their *FLOCK_TO* list. Flocking is only allowed via the super-pool.This way, users get the quality of service they were already enjoying from their own central manager, but excess jobs may be conveniently sent to the other resources. We give the group’s negotiator priority over super-pool’s to guarantee high priority to group users on their own machines.

In order to maximise the usage of our resources, we are working to find a solution that allows a nice coexistence of parallel and batch jobs. HTCondor provides several runtime environments (called a universe) [3]. Of the universes available, we have studied three cases: standard, vanilla, and the parallel universe.

Jobs which use more than one core in parallel, can also run in the standard or vanilla universes,

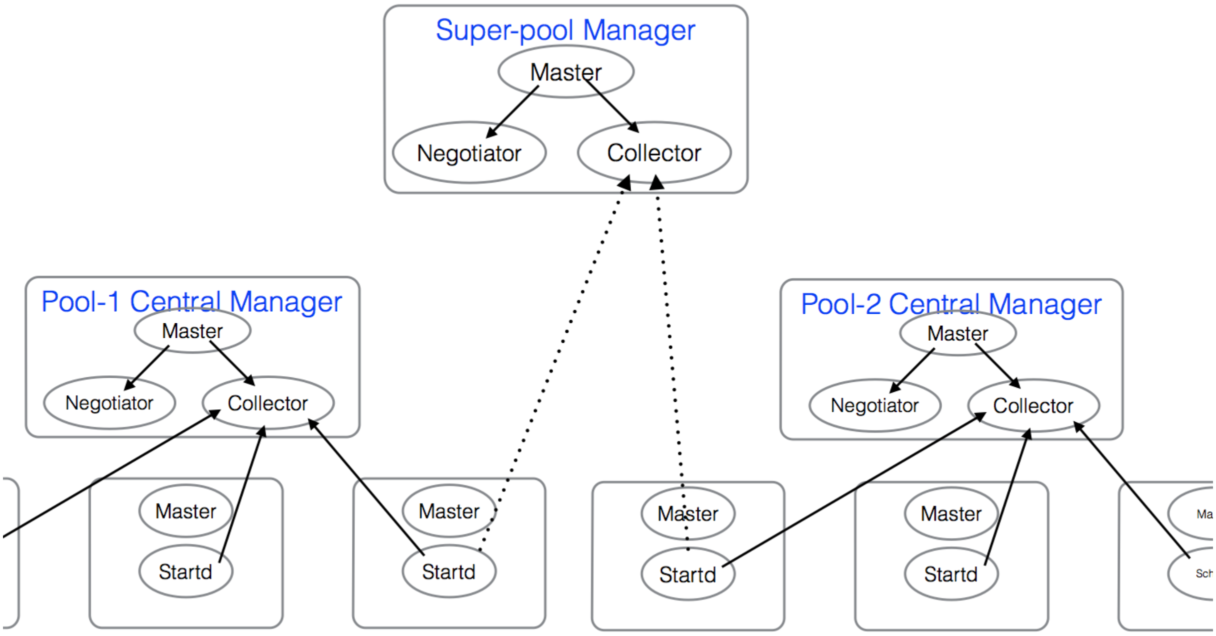

Figure 2. This diagram illustrates the way resources are organised in Milan, where execute machines of every independent HTCondor pool also reports to the collector of the additional central manager in charge of optimisation and monitoring of the resources.

although they can only use resources which belong to the same physical machine. This can be accomplished by using partitionable or dynamic slots. The whole physical machine is presented as a single slot, from which resources are provisioned dynamically when they are needed. By partitioning the slot, the use of these resources becomes more flexible. This design has been successfully implemented and tested. Users submit their analysis to the standard or vanilla universes and their jobs can run on resources external to their pool via flocking.

One known issue of this design is that idle resources tend to be occupied by single-core jobs, preventing multi-core jobs from finding enough idle resources at the same time in the same machine, even if the latter has better priority. This is addressed in our local ATLAS Tier-2 by periodically draining selected resources with the *condor_defrag* daemon. The possibility of extending the use of *condor_defrag* to the whole system is still under study since we still don't have a realistic model of the workload.

Of course the parallel universe supports a wide variety of parallel programming environments, and it encompasses the execution of MPI jobs. It supports jobs which have more than one process that must be running at the same time on different machines to work correctly. While the parallel universe is the preferred way to run parallel jobs, a mechanism that allows for this type of jobs to be executed on external resources via flocking is still work in progress.

Regarding storage, users are encouraged to use CEPH Object Storage [4] via RESTful protocols like Amazon S3. A Ceph installation is available in the department for this use too. Each pool also has local scratch disk for those who are not yet ready to use Object Storage access directly with their compiled programs and they want to use the local disk as buffer and send data to the Object Storage with an external script when the job is finished. External access to existing local ATLAS storage can be handled via XrootD [5]. Another initiative contemplates extending GPFS [6], used now by ATLAS, to the other clusters.

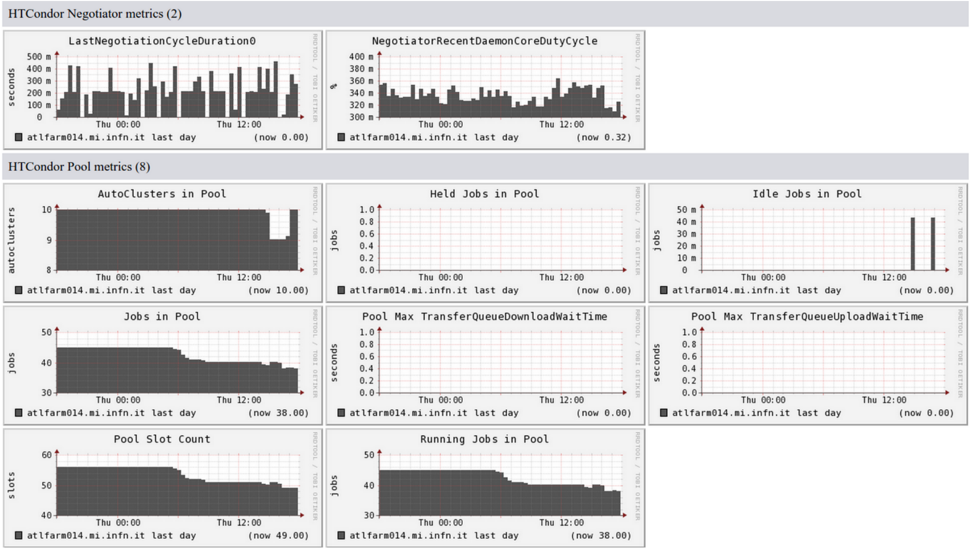

Figure 3. This Ganglia charts show the performance of one of the HTCondor pools active in Milan.

3. Authentication, authorisation

Users' authentication and authorisation are centralised and performed through server and proxy radius, LDAP and Kerberos. Different categories of users (University, INFN, guest) are redirected to different authentication servers, also external to the Physics Department, while authorisations are managed internally by LDAP. Local authentication and authorisation servers are built in high availability configuration.

In the current setup, authentication among HTCondor daemons, both in the super-pool and the local clusters, is handled via host name, because all the clusters are under our administrative control. More robust solutions will be studied in the future, when fully independent pools join the system.

4. Monitoring

The system is monitored using Ganglia [7], a scalable distributed monitoring system for high-performance computing. It is based on a hierarchical design targeted at federations of clusters, which makes it ideal for our purpose. In our design, metrics are aggregated first for each cluster independently, and are sent later to the top-level manager, which also acts as a Ganglia server. Figure 3 shows an example of the typical HTCondor metrics monitored with Ganglia.

5. Conclusion

University of Milan Physics Department and the local unit of INFN can now allow and encourage the sharing among different research areas of computing, storage and networking resources. Our approach, based on HTCondor, has been tested and shows good behaviour. Parallel and batch jobs successfully coexist in the standard universe, and can be run in resources external to the pool via flocking upon need. Flocking in the parallel universe is still work in progress.

Authentication and authorisation through LDAP and Kerberos, and monitoring with Ganglia are fully functional.

Technologies like virtualisation and cloud computing are being tested in order to maximise availability and reliability. We aim to allow for a dynamic expansion of resources upon need, in a way that is transparent to the user, where cloud resources are organised under HTCondor.

Acknowledgments

We acknowledge the support of the Milano University Physics Department and INFN computing center. The work of M.V. is partially supported by an Italian PRIN2010 grant (20108T4XTM).

References

[1] R Jones and D Barberis, The ATLAS computing model, 2008 J. Phys.: Conf. Ser. 119 072020

[2] Douglas Thain, Todd Tannenbaum, and Miron Livny, Distributed Computing in Practice: The Condor Experience, Concurrency and Computation: Practice and Experience, Vol. 17, No. 2-4, pages 323-356, February-April, 2005.

[3] http://research.cs.wisc.edu/htcondor/manual/v7.8/2_4Road_map_Running.html

[4] http://ceph.com

[5] http://xrootd.org

[6] http://www-01.ibm.com/support/knowledgecenter/SSFKCN/gpfs_welcome.html

[7] Matthew L. Massie and Brent N. Chun and David E. Culler, The Ganglia Distributed Monitoring System: Design, Implementation And Experience, Parallel Computing, 2003, Vol. 30, pages 2004.