ATL-SOFT-PROC-2015-055
22 May 2015

Bringing ATLAS production to HPC resources - A use case with the Hydra supercomputer of the Max-Planck-Society

J A Kennedy[1], **S Kluth**[2], **L Mazzaferro**[2], **R Walker**[3], **on behalf of the ATLAS Collaboration**

[1] Rechenzentrum Garching, Giessenbachstraße 2, 85748 Garching
[2] Max-Planck-Institut für Physik, Föhringer Ring 6, 80805 München
[3] Ludwig Maximilians Universität München, Am Coulombwall 1, 85748 Garching

Abstract. In recent years the architecture of HPC systems has evolved, moving away from specialized monolithic systems, to a more generic Linux type platform. This change means that the deployment of non-HPC specific codes has become much easier. The timing of this evolution perfectly suits the needs of ATLAS and opens a new window of opportunity.

The ATLAS experiment at CERN will begin a period of high luminosity data taking in 2015. This high luminosity phase will be accompanied by a need for increasing amounts of simulated data which is expected to exceed the capabilities of the current Grid infrastructure.

One approach of ATLAS to address this problem is to develop opportunistic access to resources such as cloud and HPC systems. This paper presents the results of a pilot project undertaken by ATLAS and the Max-Planck-Institut für Physik (MPP) and the Rechenzentrum Garching (RZG) to get access to the Hydra supercomputer facility. Hydra is the supercomputer of the Max-Planck-Society, consists of over 80000 cores and 4000 physical nodes, is located at the RZG near Munich, and uses the Linux OS.

This paper describes the work undertaken to integrate Hydra into the ATLAS production system by using the NorduGrid ARC-CE and other standard Grid components. The customization of these components and the strategies for HPC usage are discussed as well as possibilities for future directions.

1. Introduction

After a two year technical stop for system upgrades the Large Hadron Collider (LHC) started commissioning again in spring 2015. In this second phase, called Run 2, the LHC is expected to achieve a center of mass energy of 13 to 14 TeV, a luminosity of $10^{34}\text{cm}^{-1}\text{s}^{-2}$ with a bunch crossing every 25 ns. The pile-up, for the two major experiments ATLAS [1] and CMS [2], is also expected to grow during the data taking period $2015 - 2018$. This scenario will lead to an increase of the computing resources needed by the experiments [3].

During LHC Run 1 from March 2010 to December 2012, the ATLAS experiment continuously used more CPU resources than made available via the pledges from partner centers. This can be seen by looking at figures 1 and 2, and similar behavior was observed for the other LHC experiments.

To be able to satisfy the increased computing requests, ATLAS benefited from non-dedicated resources: mainly overprovisioning by existing Tier-1 and Tier-2 centres, domestic and commercial Cloud providers, and already now about 5% is provided by HPC systems [4]. Exploitation of HPC can potentially provide very large CPU resources and thus ATLAS is highly interested in this option.

This document describes the technical aspects and the strategies adopted for integrating the MPG Hydra Supercomputer in the ATLAS Grid infrastructure, using standard Grid tools provided by the NorduGrid [5] community.

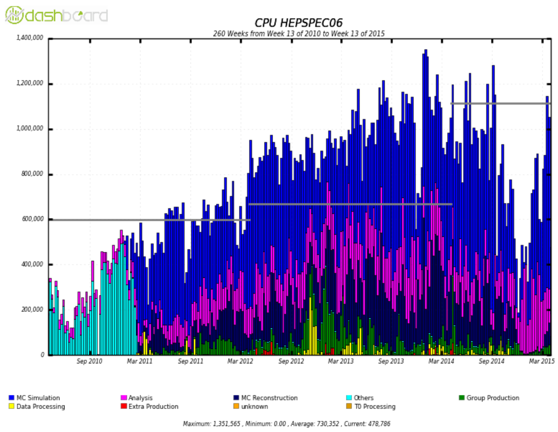

Figure 1. The plot shows the HEPSPEC06 weighted CPU hours used by the different types of ATLAS jobs, in the Run 1 period from April 2010 to March 2015. The gray line represents the resources pledged to the ATLAS collaboration for the same period.

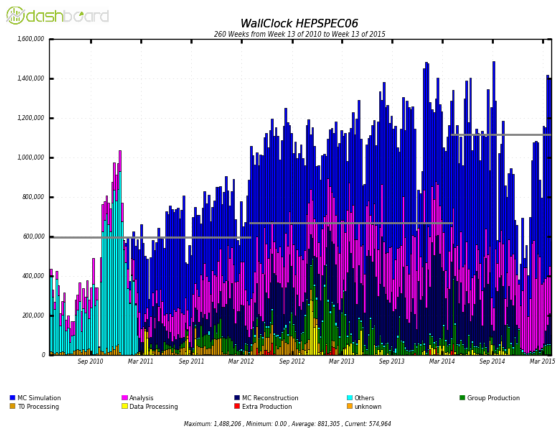

Figure 2. The plot shows the HEPSPEC06 weighted wallclock hours consumed by the different types of ATLAS jobs, in the Run 1 period from April 2010 to March 2015. The gray line represents the resources pledged to ATLAS Collaboration for the same period.

2. The RZG Hydra Supercomputer

The *IBM iDataPlex HPC system Hydra* consists of about 3500 nodes equipped with Intel Ivy Bridge processors with 20 cores per node at 2.8 GHz, and 610 nodes with Sandy Bridge-EP processors with 16 cores per node at 2.6 GHz. Finally, 350 nodes of the Ivy Bridge type are equipped with accelerator cards. The majority of the nodes have 64 GB of main memory, while 200 Ivy Bridge plus 20 Sandy Bridge have 128 GB. The higher memory allows these machines to run jobs with extremely high memory requirement. The jobs are managed by the IBM LoadLeveler V5.1 batch system on the Linux OS [7].

The nodes are organized in five domains: three with 628 nodes, one with 1800 nodes and another one with nodes equipped with the accelerator cards. In each domain the networking is based on InfiniBand FDR14 technology, which ensures fast communications. On the other side the connection between the different domains is much weaker and for this reason the batch jobs are confined to run within one domain. In common with many other HPC systems, Hydra nodes are not equipped with local disks and instead storage is based on the IBM gpfs shared filesystem: 26 I/O nodes provide more than 5 PB of disk storage. This space is used for storing data, hosting jobs binaries and saving the results. The system can reach a peak performance of about 1.7 PFlop/s while the accelerator part alone provides up to 1 PFlop/s.

The Hydra system is designed to run tasks which are CPU intensive, highly parallelized and have low I/O. These requirements contrast somewhat with several types of ATLAS jobs, for instance data analysis, which is I/O intensive. Following this, the types of ATLAS jobs which can run on Hydra are limited to those which best suit the system, as described in section 4.

The Hydra system can be accessed only from within the Max-Plank-Gesellschaft (MPG) network and all connectivity from the nodes with the internet is completely blocked including the ATLAS Grid. The users interact with the system via several login nodes using standard communication protocols such as ssh. There are no Grid services provided by default and the data management and the job submission is made directly by the users from the login nodes.

The OS installed on the nodes is SLES 11 sp3 for the x86_64 platform [6]: this is the enterprise version of SUSE Linux OS. As a consequence the ATLAS software, based on Scientific Linux 6 (SL6), can be easily integrated.

Due to the high resource requirements needed by classic HPC jobs, the Hydra system is unlikely to reach a slot coverage of 100 %. The unused slots, although temporary in nature, can be filled opportunistically by lighter and short running jobs. With more than 83000 cores and even with a mean slots coverage of 98 %, in principle ATLAS could potentially use more than 1500 slots.

The next section will describe how the Hydra system was integrated with the ATLAS Grid using the NorduGrid ARC computing element. The solutions presented can be easily generalized to other HPC systems and experiments using the Grid standard protocols.

3. The integration between Hydra and the Grid

As already introduced in the previous sections, Hydra is neither accessible from outside the MPG network nor is there any kind of Grid interface. As a consequence, a full integration with the ATLAS Grid needs:

- an interface to the ATLAS Workload management system, PanDA [8], for job scheduling;
- a service which manages staging in and out the data to and from the ATLAS Grid;
- an interface to interact with the Hydra local batch system, the IBM LoadLeveler;
- a local web proxy (SQUID) which allows ATLAS jobs to retrieve detector conditions data etc.

3.1. The ARC Computing Element

A solution satisfying the requirements is the ARC Computing Element (ARC-CE), developed by NorduGrid. Through its interface, see figure 3, the ARC-CE:

- provides a standard Grid interface for the local batch system for jobs submission, management and information retrieval about computing resources;
- is able to stage in and cache the input files, to avoid wasting resources, and to stage out the results to a remote storage service, i.e. the ATLAS DDM;
- monitors the status of the jobs and makes this information available to ATLAS.

Only users authenticated via X.509 certificates can access the ARC-CE. Some adjustment to the ARC-CE code has been made to enable a complete and optimized integration with the Hydra local batch system. The ARC-CE has been installed in a dedicated virtual machine with the Scientific Linux OS, a very light hardware configuration (2 cores, 8 GB of RAM) and inbound and outbound internet connectivity.

The ATLAS software is provided via the CernVM filesystem [10] (cvmfs), which is a read-only POSIX filesystem designed to deliver scientific software on virtual and physical machines. It is built on top of the *filesystem in user space* (fuse) Linux kernel module and uses the standard HTTP protocol for file distribution.

Some customizations have been applied to realize the full integration between Hydra and ATLAS:

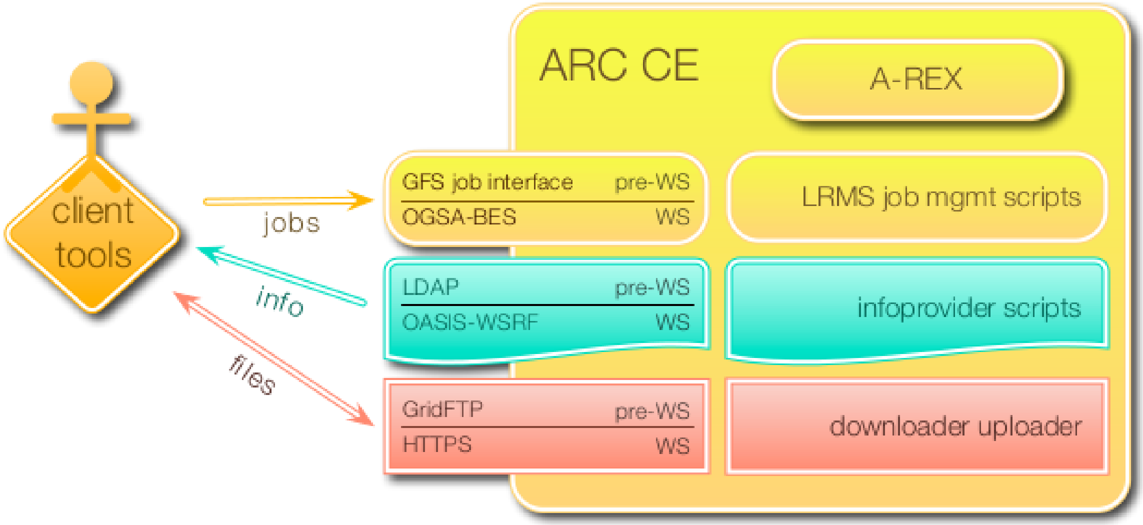

Figure 3. The ARC-CE interfaces and components [9]

- a new unix user has been created in the ARC-CE and mapped with the unix user granted rights to run jobs on Hydra;
- the Hydra gpfs file system was exported to the ARC-CE node and the area owned by the unix user was mounted in read and write mode. This area, shared between the compute nodes and the ARC-CE node, contains several directories, as shown in figure 4. The `cache`, `session` and `tmp` directories are used for data staging, caching and monitoring. The `binary` and `lib` areas contain the software required by ATLAS jobs but neither present on the compute nodes nor provided by cvmfs. The `runtime` directory is filled with configuration files needed by the jobs when they start on the compute nodes. These files set up the environment on each node, including the required ATLAS software, to run the jobs. Finally the `software` area is used to share the ATLAS software coming from cvmfs.
- the ATLAS software needed by the jobs is manually synchronized, via `rsync`, in the shared `software` directory. This is due to limitations of the `cvmfs` technology which prevent mounting it in the gpfs shared filesystem. As a consequence, Hydra can run only predefined tasks with the software present in the `software` directory.

3.2. Workflow

Following the schema on figure 4, the entire workflow can be summarized like this:

(i) the ATLAS jobs ready to be processed are sent via the ARC Control Tower [11] (aCT) to the ARC-CE. This service takes care of all the communications with the ATLAS job management system PanDA;

(ii) the ARC-CE, using the information contained in the job description file, downloads the input data into the shared `cache` and submits the jobs to the Hydra batch system;

(iii) once the job has been submitted, the ARC-CE, interacting with the batch system, monitors the jobs status and provides this information to PanDA via the aCT;

(iv) the jobs running on Hydra will find the required and pre-staged software and data directly in the shared areas `software`, `cache` and `binary` and `lib`;

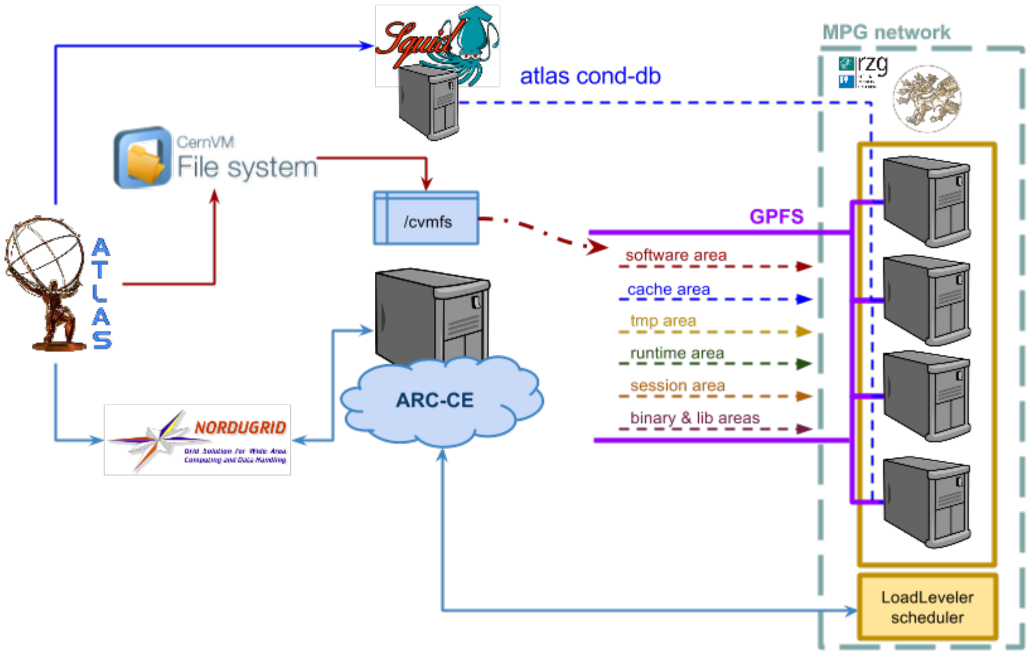

Figure 4. The Hydra-ATLAS Grid integration schema

(v) when a job finishes successfully the ARC-CE transfers the results directly to the ATLAS DDM system.

Some customizations to the ARC-CE core code have been applied for a better integration with the Hydra Supercomputer. These involve especially the interface with the local batch system LoadLever where, in some places, a rewrite of the code was necessary. Another area of development was the so called `cache cleaning` script. This ARC-CE script was in conflict with the Hydra gpfs filesystem and thus was not able to respect the user quota. The script was modified to solve this issue. All these changes have been communicated to the NorduGrid developers.

4. Results

In this section statistics and results about the Hydra usage for the ATLAS jobs will be presented. All the information has been retrieved using the ATLAS Dashboard [12], for the period 01 July 2014 to 01 March 2015.

In figures 5 and 6 Hydra's contribution is compared to the established MPPMU ATLAS Tier-2, which is also located at RZG. The parameters used are the CPU consumption for *good jobs* (jobs successfully finished), figure 5, and the total number of completed jobs, figure 6. In the first case the Hydra fraction of CPU time is about 24 %, while considering the total number of jobs completed it is about 17 % compared to the Tier-2. This difference is due to the characteristics of the jobs which can run on Hydra, and in general on modern Supercomputers: highly parallelized, low I/O and CPU intensive.

In figures 7 and 8, the *job efficiency*, defined as the ratio between the CPU time and the full walltime to run the job, is presented. Low values can be an indicator of a high I/O load, which would generally not be suitable for HPC systems. In figure 7 it is also possible to note the effect of organizing jobs in campaigns for Hydra, which leads to intermittent usage, while for the

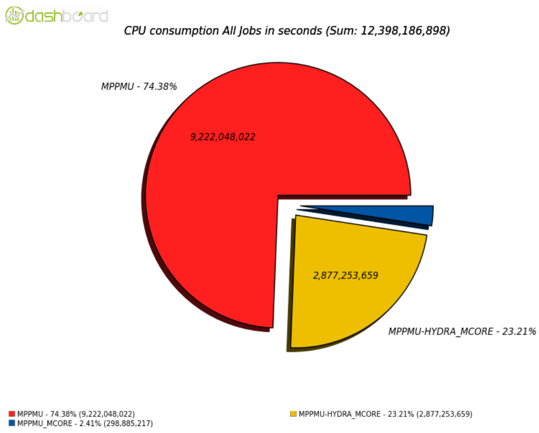

Figure 5. CPU consumption for successful jobs in seconds. Hydra's contribution is about 24 %

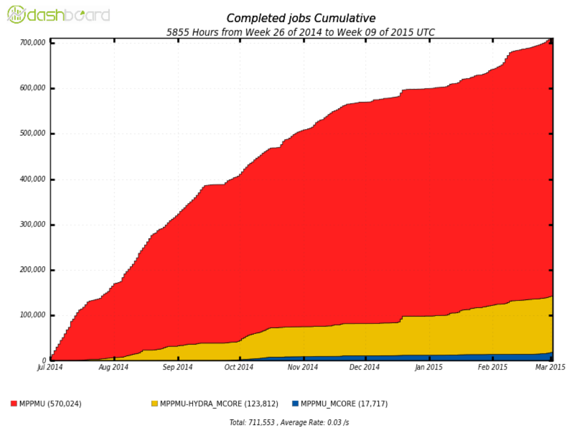

Figure 6. The total number of completed jobs processed at the MPPMU T2 and Hydra. Hydra's contribution is about 17 %

MPPMU Tier-2 a continuous flow can be seen. This is due to the fact that the most suitable jobs for Hydra are Monte Carlo simulations. Monte Carlo jobs are ideal for HPC systems since they support parallelization; have low I/O and are CPU intensive. However they are only one part of the ATLAS production chain and this has led to a campaign-like deployment with periods of high and low usage. Moreover only the most recent ATLAS code releases have the optimized code for running in such HPC systems.

The results to date are very encouraging, however areas where improvements can be made are under consideration. Figure 7 shows that improvements in resource usage can be made by ensuring that a continuous flow of jobs is available, however this is also influenced by the availability of free slots on the HPC systems. Additionally, as the usage of HPC systems progresses, the efficiency of the HPC multi-core jobs as shown in figure 8 should also improve.

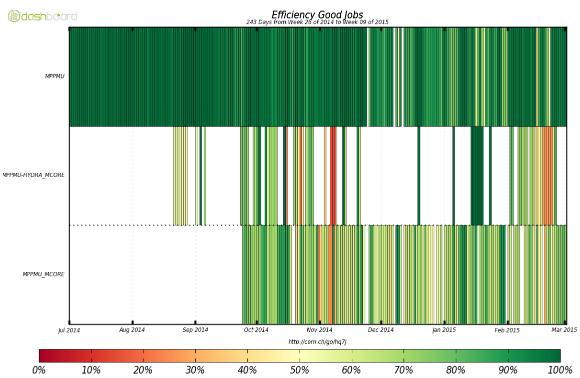

Figure 7. Efficiency of ATLAS good jobs, in the time period July 2014 to Mar 2015). Top=MPPMU, Middle=Hydra-Multicore, Bottom=MPPMU-Multicore.

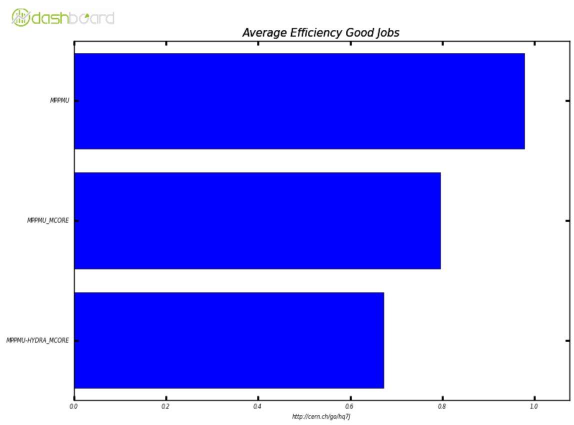

Figure 8. The average efficiency of ATLAS good jobs. Top=MPPMU, Middle=MPPMU-Multicore, Bottom=Hydra-Multicore.

Moreover, if we wish to understand the efficiency of the resource usage, it is worth comparing the usage of the Hydra system with the MPPMU Tier-2 w.r.t. successful and failed jobs. Figures 9 and 10 show the successful, failed and deleted jobs which ran on Hydra and the MPPMU Tier-2. The percentage of failed jobs on Hydra is slightly higher, by approximately 4 %, than on the MPPMU Tier-2. This is an encouraging result, since the failure rate on Hydra also contains failed jobs from the initial test phases, and thus production jobs on Hydra run with approximately the same failure rate as similar jobs on the Grid. The result shows both the suitability of ATLAS simulation codes for HPC systems and the reliability of this solution on HPC resources.

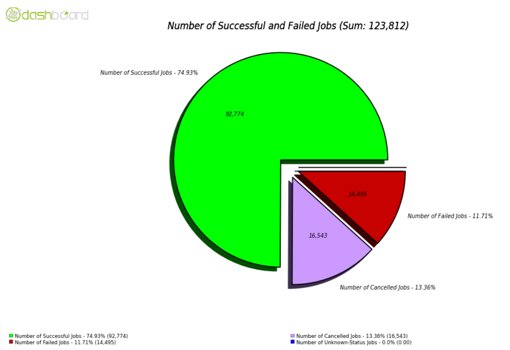

Figure 9. Hydra: Percentage of successful (75%), failed (12%) and deleted (13%) jobs. The deleted jobs were canceled directly by ATLAS.

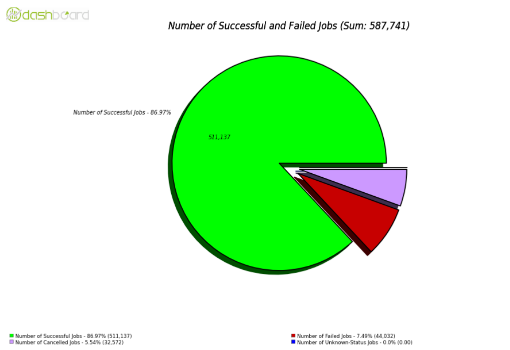

Figure 10. MPPMU Tier-2: Percentage of successful (87%), failed (8%) and deleted (5%) jobs. The deleted jobs were canceled directly by ATLAS.

Finally figure 11 shows the types of jobs which ran on Hydra. While this confirms the bias towards the more suitable MC simulations it also shows some fraction of reconstruction jobs can be performed on the HPC systems. Monte Carlo simulation and reconstruction jobs required together about 2/3 of the compute resources of the ATLAS Grid in 2014 [4] and we expect MC production to dominate ATLAS CPU resource consumption in Run 2. Moving these jobs onto HPC systems as opportunistic resources, ATLAS could free up capacities of the generic ATLAS Grid for analysis and other types of jobs.

5. Conclusions

The Hydra supercomputer is running ATLAS jobs, mainly Monte Carlo simulation, since more than one year. It is fully integrated into the ATLAS Grid infrastructure and as shown in section 4, its contribution, compared with the MPPMU Tier-2, has already reached a significant level.

This type of opportunistic HPC resource usage presents benefits for both ATLAS and the institutes hosting the HPC systems. ATLAS can gain additional resources, allowing traditional Grid resources to be applied to data analysis tasks etc, and the HPC systems themselves can increase their occupancy by making use of these less resource hungry, short running jobs.

One of the major issues met during the integration was related to the usage of the Hydra gpfs file-system: this was addressed by both the ATLAS community and Hydra administrators, mainly by improving file access patterns by reducing the need to access many small files.

The major areas identified for improvements are:

- Monte Carlo multi-core code: a further optimization in the code is needed to increase the job efficiency;

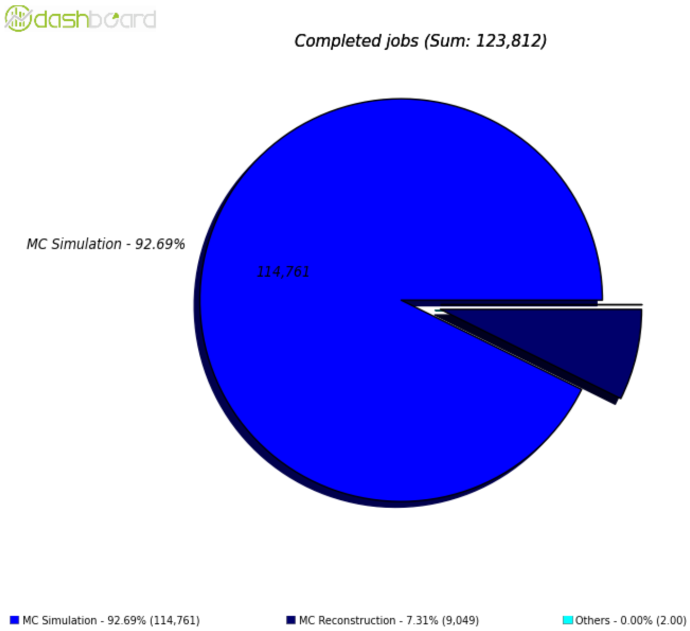

Figure 11. Breakdown of jobs types run on Hydra during the period April 2014 - March 2015

- ATLAS software distribution: moving from a manual synchronization of the ATLAS software in the shared area to an automatic deployment can increase the reliability of the entire system;
- defining strategies for a better usage of the HPC opportunistic resources via backfill and preemptive job deletion.

References

[1] G Aad et al. *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST 3 (2008) S08003
[2] S Chatrchyan et al.,*The CMS experiment at the CERN LHC*, JINST 3 (2008) S08004
[3] I Bird et al, *Update of the Computing Models of the WLCG and the LHC Experiments*, CERN-LHCC-2014-014 / LCG-TDR-002 (2014)
[4] S Campana, *ATLAS Distributed Computing in LHC Run 2*, Proceedings of CHEP 2015 conference, J. Phys.: Conf. Ser.
[5] The NorduGrid Collaboration, `http://www.nordugrid.org`.
[6] *SLES Home Page*, `https://www.suse.com/products/server/`
[7] IBM RedBooks, *Workload Management with LoadLeveler*, `http://www.redbooks.ibm.com/redbooks/pdfs/sg246038.pdf`
[8] T Maeno et al., *Overview of ATLAS PanDA Workload Management*, J. Phys.: Conf. Ser. 331 (2011) 072024
[9] F Paganelli et al, *ARC Computing Element, System Administrator Guide*, NORDUGRID-MANUAL-20 (2014)
[10] J Blomer et al., *The CernVM File System*, Technical Report (2013) CERN
[11] A Filipcic, *arcControlTower: the System for Atlas Production and Analysis on ARC*, J. Phys. Conf.: Ser. 331 (2011) 072013
[12] ATLAS dashboard, `http://dashboard.cern.ch/atlas/`