PROCEEDINGS OF SCIENCE

Experience with distributed analysis

Ulrik Egede*
Imperial College London, UK
E-mail: U.Egede@imperial.ac.uk

Analysis of large amounts of data from the LHC will start very soon. I give an overview here of the challenges faced by this analysis in an environment where the coumputing resources are spread across the world. An outline is given of the tools used by individual physicists, the performance of the systems and the monitoring performed. I use this information to collect a set of best practise recommendations for the future.

XXth Hadron Collider Physics Symposium
November 16 – 20, 2009
Evian, France

*Speaker.

PoS(HCP2009)052

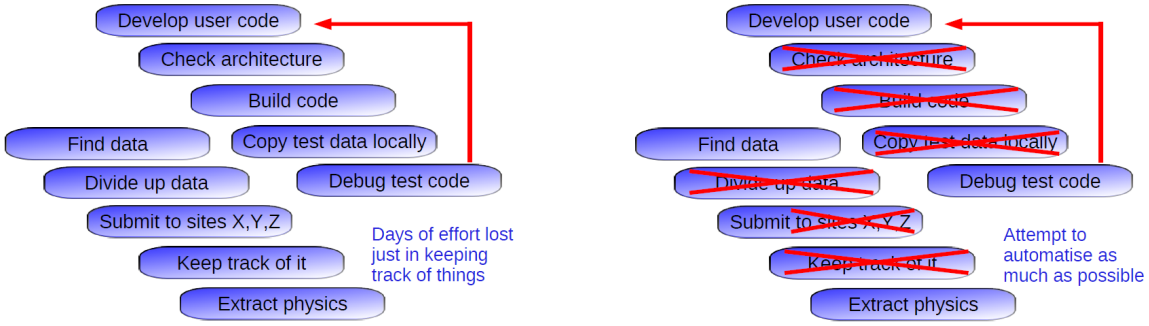

Figure 1: A typical workflow for performing a data analysis in HEP. The number of steps involved is very large and many of them related to error prone manual bookkeeping. To the left is illustrated the situation without a User Interface to help the physicist; to the right the tasks taken over by the User Interface are crossed out.

1. Introduction

The term distributed analysis requires a definition to make any sense. This is best done through a short description of how a new result is typically obtained in High Energy Physics.

Collisions at a collider produce very large amounts of data. The fraction of the collisions that are of interest for a given physics results is typically minuscule meaning that the analysis is a large filtering exercise. The filtering is done in a multi-stage process, often starting at the trigger level, then going through some kind of selection shared among a large number of physicists before datasets are made available to individual physicists for the final processing. The amount of data involved means that all processing can't take place at a single computational site; for the LHC a large number of sites participates in the processing using the EGEE [1], OSG [2] and ARC [3] Grid interfaces.

The process of distributed analysis is thus when individual physicists process data, often located across many sites.

2. User interfaces

To analyse data in a distributed environment can be a daunting task. As illustrated in Fig. 1, the number of steps involved range from ensuring that the user analysis code is built for the correct architecture supported on a Grid site, to keeping track of the which parts of large datasets failed processing.

In order to assist this complex task, several user interfaces such as CRAB[4] used within the CMS collaboration, GANGA [5] used with ATLAS and LHC*b* and ALIEN[6] used by Alice have been developed. While these user interfaces in their initial forms were very different, the exchange of ideas between them mean that they now are very similar in functionality.

As an example GANGA allows for the specification, submission, bookkeeping and post-processing of computational tasks on a wide set of local and distributed resources. In GANGA it is possible to develop analysis code on your local PC where it can be tested on small datasets in an interactive manner, move to a batch system for intermediate testing and finally process the complete dataset

Figure 2: An overview of all the different platforms that GANGA allows submission to. A typical analysis will start using local resources and when everything is working will process all the data on the Grid.

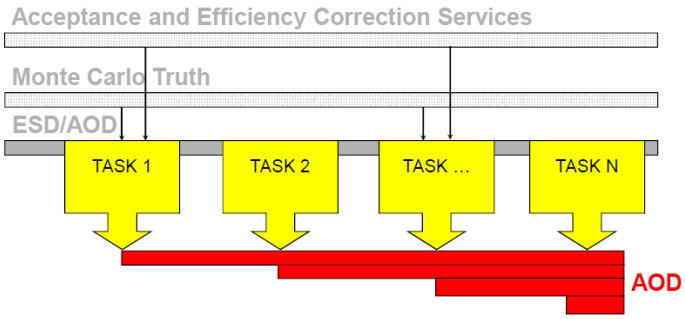

Figure 3: An illustration of an analysis train in Alice. Users submit *tasks* to be carried out, which are then included in a weekly processing of the data.

on the data distributed across multiple Grid sites. The different platforms that GANGA allows for processing on are illustrated in Fig. 2.

The key to the success of GANGA is the ability to move between different processing environments as outlined above without making any changes to the workflow of the analysis.

While chaotic data processing by individuals is very attractive as a physicists gets to analyse their data when they want it, it carries many disadvantages such as: the same data might be analysed by different users for the same purpose, scheduling might lead to bottlenecks in the system and code not strictly versioned causing trouble with reproducibility. For these reasons as much as possible of analysis should be done in a shared way using a production environment. Often this will take the shape of data reduction passes on the full dataset coordinated by different physics groups.The Alice collaboration has an interesting concept of *analysis trains* as illustrated in Fig. 3. The idea is to have a very easy way for user code to become part of a centralised processing.

3. Current performance

The obvious questions to ask about the current analysis systems are if they work. Reasons for failure are numerous. A few examples are: aborted Grid jobs, unscheduled downtime, misleading exit codes, wrong documentation, lack of debugging information, and data not available as described. While a user might be able to debug these problems themselves in a local environment, they simply do not have the tools or indeed the access rights to do so in a distributed environment.

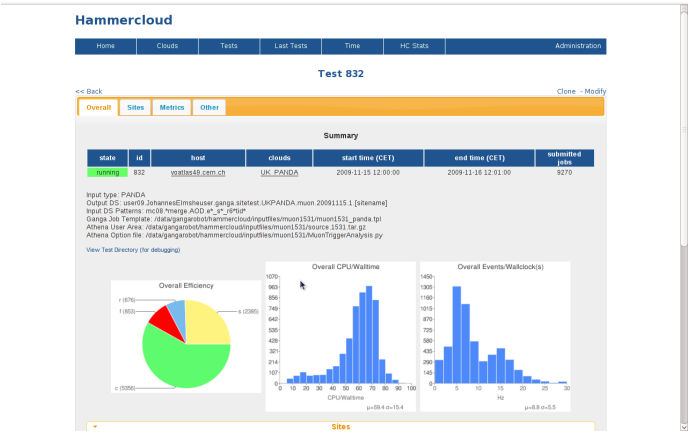

Figure 4: Results from HammerCloud testing in ATLAS on 15 Nov. 2009.

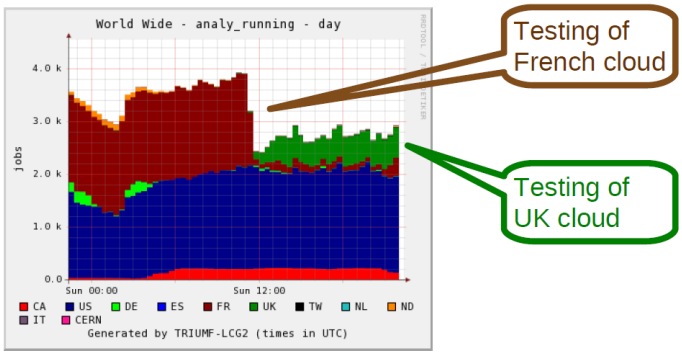

Figure 5: Overview of all the analysis jobs running in ATLAS as seen by the PANDA system. The performance testing from HammerCloud as illustrated in the previous figure can clearly be seen.

The development of testing systems that emulate user analysis in a controlled manner and the monitoring of all user jobs and their behaviour has made these problems visible to the developers of the systems. Most of these issues can now be dealt with automatically or at least hidden from the user. A few examples of this are given below.

Testing of the distributed analysis is done by creating analysis tasks in an artificial way. It has two purposes. One is to stress test the system by creating jobs at a level that is anticipated for analysis in the future; the other is to be able to catch failures and general behaviour issues of the system in a programmatic way, rather than relying on emails and bug reports from users.

The HammerCloud system developed by ATLAS is a very comprehensive system for such testing. In Fig. 4 the status of the system can be seen from 15 Nov. 2009. In Fig. 5 all the analysis jobs running on the same day is shown. The rotating tests of the French grid cloud and subsequently the UK grid cloud from the HammerCloud testing can clearly be seen on top of genuine analysis activities.

Within the CMS collaboration, data transfers required for the subsequent analysis have achieved much attention. The data transfers achieved with the PHEDEX [7] system is illustrated in Fig. 6. As it can be seen, data transfer rates approaching the levels expected with early data taking in LHCare achieved. Again this was only possible by introducing artificial transfers for about 60% of the

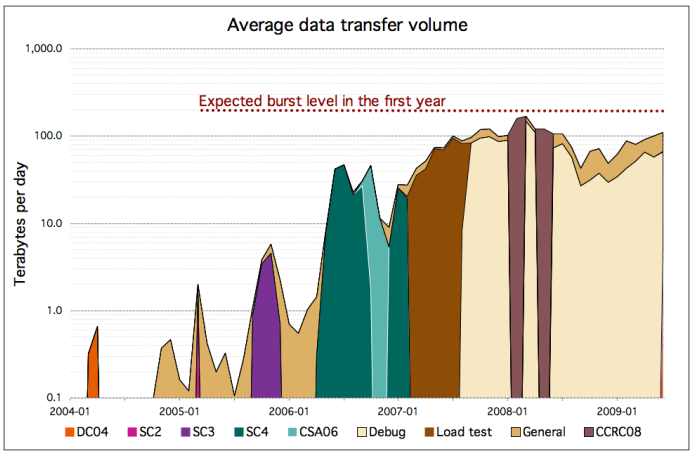

Figure 6: Data transfer rates in CMS using the PhEDEx system.

overall rate.

While User Interfaces are designed to be easy to use and the monitoring taken place in the background will catch many problems before they become visible, there is still a need for user support.In ATLAS they have recognised the need to isolate developers from the daily support and have created the *Distributed Analysis Support Team* which provides user support through mailing lists and instant messaging.

4. Conclusion

The distributed analysis systems of the LHC experiments have achieved a mature state where they are used by the order of a thousand people every day. Monitoring and testing of the systems has been developed in parallel to provide a better performance for users. In general the systems are working very well and are ready for the challenges that the analysis of vast amounts of LHC data will provide.

Many different computing paradigms have been tried out in the development of the distributed analysis systems. Some of these have proved successful while others failed, mostly due to that they did not work in systems where the behaviour can be very erratic. In terms of best practice the following recommendations can be made.

- Experiments should use of private Workload Management Systems. Shared workload management systems such as gLite does not have the knowledge to implement the most efficient use of resources by experiments. A private WMS system sending jobs to the Grid on behalf of individual users provide this flexibility.

- Planned rather than demand driven data transfers. Data transfers can take a long time and it is often not obvious to an end user what amounts of data are involved. For this reason systems, where load of individual sites is monitored and data moved to balance this, works better than requests driven by individual users that more leads to an “arms race” situation where each user try to get their data everywhere.

- Pull mode of operation, where skeleton Grid jobs are started which, when started, pull a real job. In a system where failures often occur, a pull mode operation dramatically improves reliability by performing some testing before the actual job is pulled to a site and also allows for easy resubmission to another site in case of a later failure.
- The use of User Interfaces. Learning the many different interfaces of different Grid and Batch systems is highly time consuming. The use of User Interfaces such as GANGA hides this complexity behind a programmable interface.
- Provide centralised processing where at all possible.
- The creation of dedicated support teams and training sessions for individual physicists.

Acknowledgments

In the preparation for this talk, I had many contacts with the ATLAS, CMS, LHC*b*, and Alice collaborations. In particular I would like to thanks Dan Van der Ster, Latchezar Betev, Frank Wuerthwein, Joe Incandela, Raja Nandakumar, Roberto Tenchin and Andreas Morsch.

References

[1] E. Laure *et. al.*, *Enabling Grids for e-Science: The EGEE Project*, EGEE-PUB-2009-001.

[2] Open Science Grid, *A Blueprint for the Open Science Grid*, OSG DocDB Document 18-v8

[3] M.Ellert *et al.*, *Advanced Resource Connector middleware for lightweight computational Grids*, Future Generation Computer Systems **23**, (219), 2007.

[4] D. Spiga *et. al.*, *CRAB: the CMS distributed analysis tool development and design*, Nucl. Phys. B Proc. Suppl. **177-178**, (267), 2008.

[5] J. T. Mościcki *et. al.*, *Ganga: A tool for computational-task management and easy access to Grid resources*, *Computer Physics Communications* **180** (2303) 11, 2009, [arXiv:0902.2685].

[6] P. Buncic *et. al. The architecture of the AliEn system*, CHEP 2004.

[7] J. Rehn *et. al*, *PhEDEx high-throughput data transfer management system*, CHEP 2006.