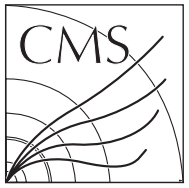
Available on CMS information server

CMS CR -2011/017



The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

11 January 2011 (v2)

Running CMS Remote Analysis Builder jobs on Advanced Resource Connector middleware

Erik Edelmann, Kalle Happonen, Jesper Koivumäki, Tomas Lindén, and Joni Välimaa

Abstract

CMS user analysis jobs are submitted to the grid with the CMS Remote Analysis Builder application (CRAB). According to the CMS computing model the applications should run transparently on the different grid flavours in use. In CRAB this is handled with different plugins that are able to to submit to different grids. Recently a CRAB plugin for submitting to the Advanced Resource Connector (ARC) middleware has been developed. The CRAB ARC plugin enables simpler and faster job submission with full job status information available. CRAB can be used with a server which manages and monitors the grid jobs on behalf of the user. In the presentation we will report on the CRAB ARC plugin and on the status of integrating it with the CRAB server and compare this with using the gLite ARC interoperability method for job submission.

Presented at *CHEP2010: International Conference on Computing in High Energy and Nuclear Physics 2010*

Running CMS remote analysis builder jobs on advanced resource connector middleware

E Edelmann[1], K Happonen[2], J Koivumäki[2], T Lindén[2] and J Välimaa[2]

[1] Nordic Data Grid Facility, Kastruplundgade 22, DK-2770 Kastrup, Denmark

[2] Helsinki Institute of Physics, P.O.B. 64, FIN-00014 University of Helsinki, Finland

E-mail: tlinden@cc.helsinki.fi

Abstract. CMS user analysis jobs are distributed over the grid with the CMS Remote Analysis Builder application (CRAB). According to the CMS computing model the applications should run transparently on the different grid flavours in use. In CRAB this is handled with different plugins that are able to to submit to different grids. Recently a CRAB plugin for submitting to the Advanced Resource Connector (ARC) middleware has been developed. The CRAB ARC plugin enables simple and fast job submission with full job status information available. CRAB can be used with a server which manages and monitors the grid jobs on behalf of the user. In the presentation we will report on the CRAB ARC plugin and on the status of integrating it with the CRAB server and compare this with using the gLite ARC interoperability method for job submission.

1. Advanced Resource Connector

The Nordugrid Advanced Resource Connector (ARC) middleware [1, 2] is the main middleware in the Nordic countries and several other countries, see Figure 1. The NorduGrid infrastructure has been up since 2002, presently there is $\mathcal{O}(55000)$ cores at $\mathcal{O}(60)$ sites with the largest CEs having $\mathcal{O}(6000)$ cores. The Nordic Data Grid Facility (NDGF), the distributed Nordic Tier-1 computing facility, uses ARC for LHC- and other VOs. NDGF is funded by Denmark, Finland, Norway and Sweden, mostly for manpower with only little hardware. The actual computing resources are provided by the participating institutes. ARC is available from the standard repositories of the major Linux distributions and it has had 64-bit support since at least 2005.

ARC has been designed to be a high performance, scalable, and reliable infrastructure while avoiding single points of failure. It is easy to install and maintain. It is portable and **available on Red Hat, Fedora, Debian and Ubuntu Linux, Windows, Mac OS and Solaris**. Resource brokering is made by the ARC User Interface (UI) at job submission time to avoid scaling problems.

A unique feature of ARC middleware is the grid monitor web interface showing detailed Compute Element, Storage Element and user status [3]. It is very useful for troubleshooting and monitoring both from the user and site perspectives. ARC supports different customized grid monitors for different countries, VOs or sites. A picture of the NDGF ARC grid monitor page for Finland is shown in Figure 2.

ARC is developed by the NorduGrid collaboration, NDGF, and within the European Middleware Initiative (EMI), in turn part of the European Grid Initiative (EGI). KnowARC

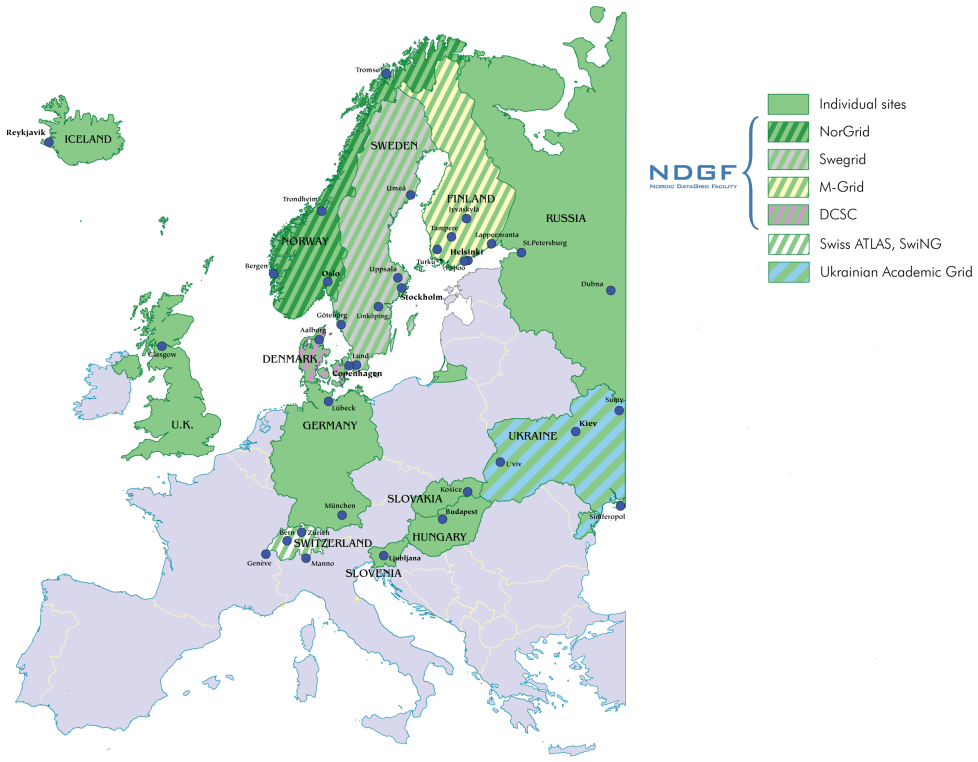


Figure 1. A map of ARC middleware deployment.

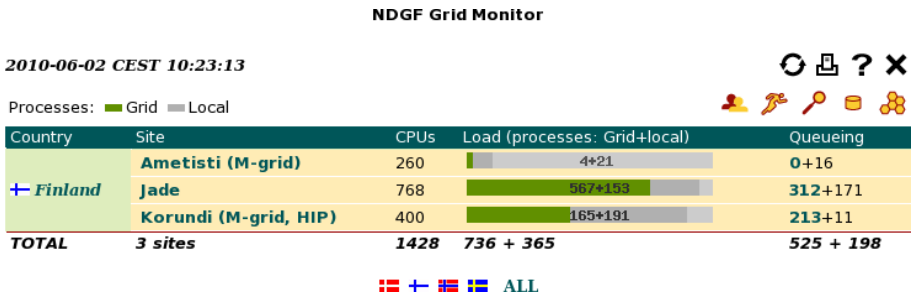


Figure 2. The ARC NDGF grid monitor page for Finland showing CMS analysis jobs. The grid monitor is a webinterface to the ARC information system allowing the extraction of detailed information for jobs in the system.

which was a 6 M€ EU-project has also developed ARC. The Unified Middleware Distribution (UMD) developed by EGI will be based on ARC, gLite and UNICORE.

The ARC UI is easy to install in binary format or as a tarball, because the configuration is defined by a supplied script and no configuration files needs to be edited. Installing the UI tarball is trivial and can be made with three commands as shown by this example:

```
wget http://download.nordugrid.org/software/nordugrid-arc/releases/\
0.8.3/redhat/el5/x86_64/nordugrid-arc-standalone-0.8.3-1.el5.x86_64.tgz
tar xvzf nordugrid-arc-standalone-0.8.3-1.el5.x86_64.tgz
source nordugrid-arc-standalone-0.8.3-1/setup.sh \
nordugrid-arc-standalone-0.8.3-1
```

2. ARC in Finland for CMS

The current Finnish grid infrastructure M-grid is built on ARC middleware. The Finnish Grid Infrastructure project will use ARC as well. Knowledge and support is mostly available for ARC in Finland. The CEs used by the Finnish CMS Tier-2 are part of M-grid, so ARC was chosen for CMS usage as well. CMS resources in Finland use services of the Nordic Data Grid Facility (NDGF), the distributed Tier-1 resource, which uses ARC as middleware.

The goals of the ARC middleware development for CMS are twofold:

- Enable CMS to submit transparently analysis jobs (CRAB) and Monte Carlo production jobs (ProdAgent) to ARC resources from other middlewares
- Enable local users to use CMS tools natively with ARC middleware

3. Interoperability

CMS, as the other LHC experiments, needs to run applications on several grid flavours, which is a non trivial problem. Interoperability is the ability to exchange information and use resources. This can be achieved on several levels:

- User joins several grids
- Site joins several grids
- Application supports several grids (ProdAgent, CRAB)
- Gateway connects grids (NDGF gLite-ARC gateway)
- Middleware interoperability [4, 5] (gLite WMS or glideinWMS + Condor-G)
- Standardized middleware APIs with interchangeable components (SRM protocol, implementations BeStMan, CASTOR, dCache, DPM, StoRM)

The problem of submitting CMS jobs to ARC resources has been solved both by middleware interoperability and by application level interoperability.

4. CMS Remote Analysis Builder

CMS Remote Analysis Builder (CRAB) is a user tool for submitting CMS software framework (CMSSW) analysis jobs to the grid [6, 7]. When submitting analysis jobs, it is important that the jobs go to a site where the data to be analyzed resides. CRAB will automatically submit the jobs to the correct place by using the CMS data location system Dataset Bookkeeping service (DBS) to find the data. The output data will be copied to a site specified by the user. CRAB can be used in standalone mode, see Figure 3 where user jobs are submitted directly to the grid or with an intermediate server, see Figure 7 that manages the jobs on the grid on behalf of the user.

4.1. Standalone CRAB

Standalone CRAB jobs can be submitted to ARC resources using:

- gLite WMS or CMS glideinWMS + Condor-G ARC interoperability
- CRAB ARC scheduler

4.1.1. gLite WMS + Condor-G ARC interoperability Submitting jobs using the gLite WMS + Condor-G ARC interoperability is **application independent**, so it is also used for CMS Site Availabilty Monitoring (SAM) jobs and jobs that install different CMSSW versions. This could be used by any VO and application that needs to submit jobs from gLite to ARC resources. All gLite WMSs have been patched to be able to submit to ARC CEs and detailed ARC job

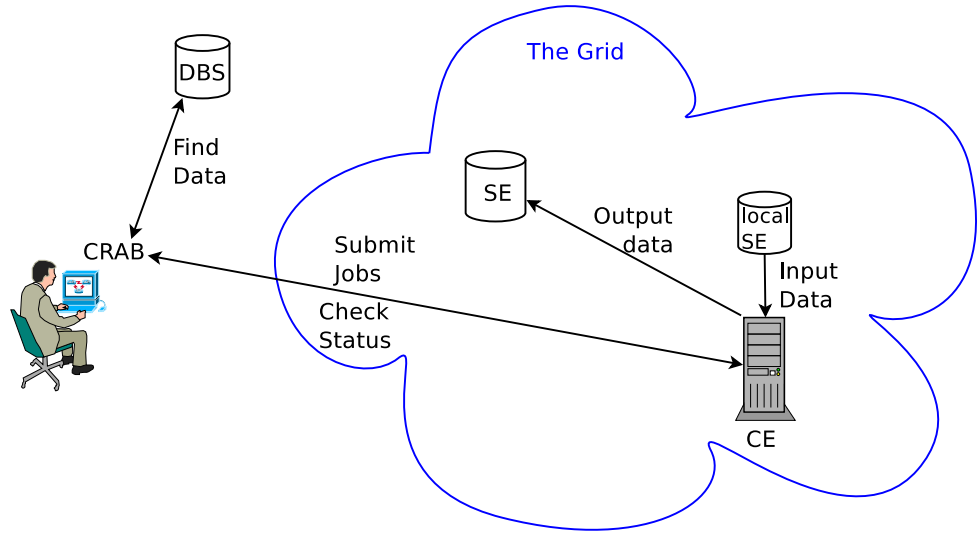


Figure 3. A simplified schematic picture of standalone CRAB.

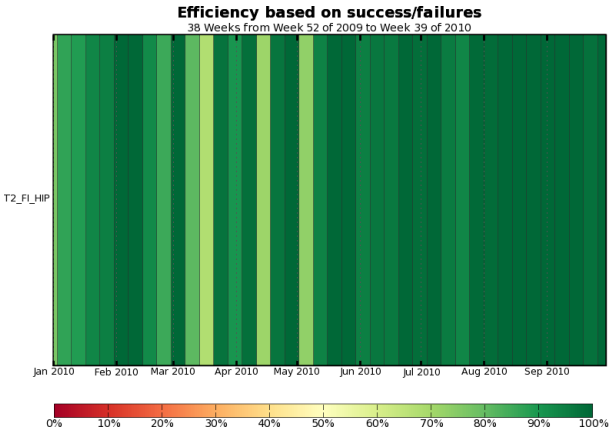


Figure 4. CMS JobRobot efficiency of ARC jobs at the HIP Tier-2 site during January-September 2010.

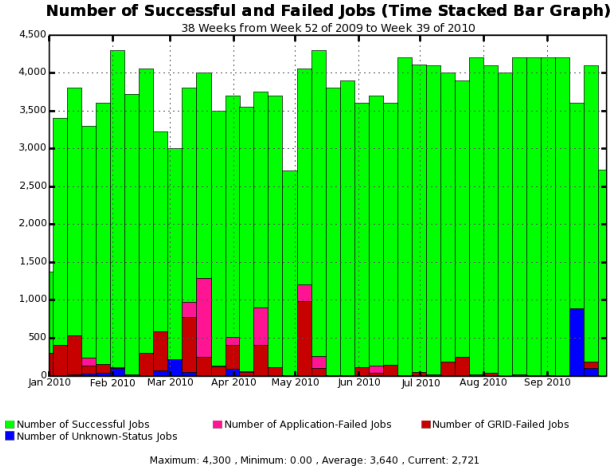


Figure 5. ARC JobRobot jobs during January-September 2010. The success rate of 149 241 jobs was 93 %.

status information is available from the gLite WMS servers. The gLite WMS + Condor-G ARC interoperability works very well in production for user analysis jobs and for site monitoring with the CRAB based CMS JobRobot, see Figures 4 and 5. The majority of CRAB jobs are submitted to ARC in this way.

The drawbacks of the gLite WMS + Condor-G ARC interoperability are that an ARC site needs the gLite worker node package installed, it requires a gLite-UI on the user's submission computer, the information system needs to be translated and grid operations (submitting, checking status, etc.) are slower compared to accessing ARC directly. This leads to increased complexity with twice as many things that can go wrong and a dependency on the way that the gLite WMS servers are working.

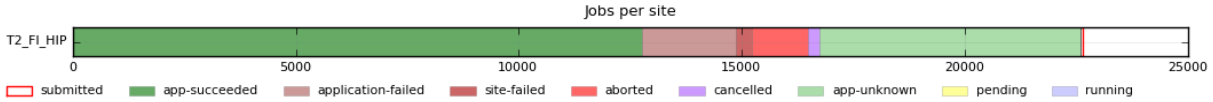


Figure 6. Jobs submitted with the ARC scheduler until the end of September 2010.

4.1.2. CRAB ARC scheduler CRAB supports different grid middlewares through schedulers or plugins. A new scheduler for ARC has been implemented in CRAB [8]. The ARC scheduler is part of the standard CRAB release and it enables using an ARC UI to submit CRAB standalone jobs to ARC resources. The CRAB ARC Runtime Environment on the CE has no gLite dependency, so the native CRAB ARC scheduler simplifies software dependencies, because no gLite UI is needed nor is a gLite RE needed on the CE. Reporting of the ARC job status to the CMS logging service, the Dashboard, has been implemented. ARC uses a shared file system (NFS) for GridManager and worker node communication. The job directory was changed to reside on the local disk for the ARC scheduler and the gLite Runtime Environment instead of on the shared file system, which improved the CPU efficiency by a small amount.

A major benefit of the CRAB ARC scheduler is that job submission can be significantly faster compared to gLite WMS + Condor-G ARC interoperability. On several occasions the gLite WMS servers used by CMS have been overloaded so that only the job submission step might take several hours. Since ARC has been designed not to use any central resource brokering this problem does not exists when submitting jobs from an ARC UI using the CRAB ARC scheduler. Another benefit is that job queries are faster since they can be submitted directly to ARC with the CRAB ARC scheduler compared to using gLite WMS + Condor-G ARC interoperability.

The schedulers ARC, gLite and Condor are mutually exclusive in the CRAB configuration file, so only one type of job can be submitted at a time, which limits the usability of the ARC scheduler, but this limitation could be resolved by using a CRAB server instead.

4.2. CRAB server

A CRAB server automates some of the tasks of the standalone CRAB by providing services such as job monitoring, automatic resubmission and notification to ease the task of running CMSSW jobs on the grid. CMS is planning to use only the CRAB server mode of operation and to phase out standalone mode CRAB.

The resource matchmaking for jobs is done by the CRAB server. CMS is developing the CRAB server so that the scheduler is not selected statically in the CRAB client configuration file, but rather on a site basis, which will make the CRAB ARC scheduler much more usable. The CRAB server at the HIP Tier-2 site is being used for testing and developing the configuration and integration of the CRAB ARC scheduler. Job submission to ARC through the HIP CRAB server has been demonstrated using gLite WMS + Condor-G ARC interoperability for the production CRAB server version. Job submission from an ARC UI to a CRAB server needs some development work.

Installing the ARC UI is trivial and installing the CRAB client is also straight forward, so any CMS physicist could install these on a PC or laptop of their choice. In this way any CMS physicist could use the ARC UI to submit CRAB jobs to a ARC enabled CRAB server accessing all CMS data resources, without the need to login to a machine with a gLite UI installed. Achieving this situation is the goal of the present work in connection with the CRAB server version 3 being developed.

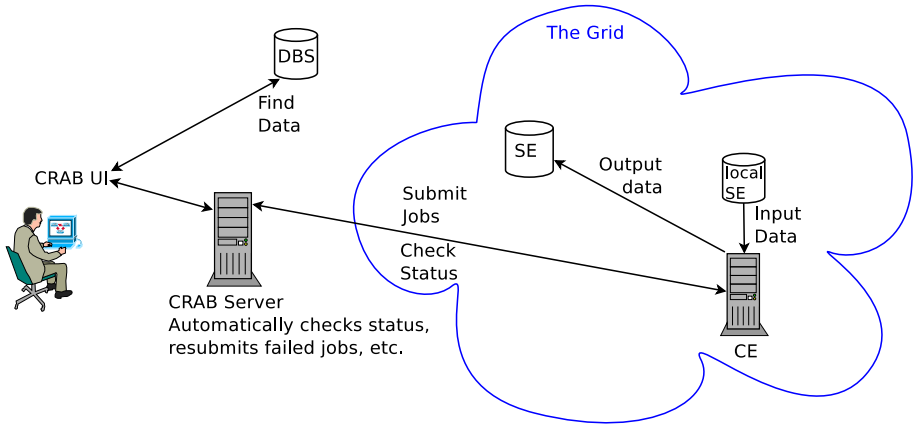


Figure 7. A simplified schematic picture of CRABserver.

5. Summary

ARC is a portable middleware used in several countries. CRAB job submission to ARC is in production use and can be accomplished in several different ways. The major part of the CRAB ARC jobs are submitted using the gLite WMS ARC interoperability, which *works for any grid application*. The gLite WMS ARC interoperability works fully transparent for the CMS users, so most of them are not even aware of that some of their jobs run on ARC and not on gLite or on OSG. The CMS glideinWMS is used to submit pilot CRAB jobs to ARC and this is also in production use. Standalone CRAB jobs can be submitted with an ARC UI to ARC resources using the ARC CRAB scheduler and sometimes this works significantly faster than using the gLite WMS ARC interoperability way. CRAB server test jobs have been submitted to ARC using gLite WMS ARC interoperability. Job submission from an ARC UI to a CRAB server needs some development. Achieveing the goal of submitting CRAB jobs with an ARC UI to a CRAB server to access any CMS grid resource needs more development work too.

Acknowledgements

This project has been funded by the Ministry of Education and Academy of Finland. The CSC personnel is thankfully acknowledged for maintaining CPU and disk resources.

References

[1] Ellert M, Grønager M, Konstantinov A, Kónya B, Lindemann J, Livenson I, Nielsen J L, Niinimäki M, Smirnova O and Wäänänen A 2007 Advanced Resource Connector middleware for lightweight computational Grids *Future Generation Computer Systems* **23** 219–40

[2] Smirnova O et al. 2010 Recent ARC developments: Through modularity to interoperability *J. Phys.: Conf. Ser.* **219** 062027

[3] Pajchel K et al. 2005 Usage statistics and usage patterns on the NorduGrid: Analyzing the logging information collected on one of the largest production Grids of the world *Proc. of CHEP 2004 27th September – 1st October 2004 Interlaken* CERN-2005-002 **2** ed A Aimar et al. pp 711-4

[4] Grønager M, Johansson D, Kleist J, Søttrup C, Wäänänen A, Field L, Qing D, Happonen K, Lindén T 2008 Interoperability between ARC and gLite - understanding the grid-job life cycle, *Proc. of 2008 Fourth IEEE International Conference on eScience* 493–500

[5] Klem J et al. 2010 Grid Interoperation with ARC Middleware for CMS Experiment *J. Phys.: Conf. Ser.* **219** 062016

[6] Fanfani A et al. 2010 Distributed Analysis in CMS *J. Grid Comput.* **8** 159–179

[7] Spiga D et al. 2008 CRAB: the CMS distributed analysis tool development and design *Nucl. Phys. B - Proc. Suppl.* **177-178** 267–268

[8] Koivumäki J 2010 *Development of a Plugin for Transporting Jobs Through a Grid* `http://urn.fi/URN:NBN:fi:amk-2010082512761`