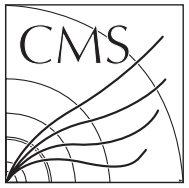
Available on CMS information server

CMS CR -2009/070



The Compact Muon Solenoid Experiment

Conference Report

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

14 May 2009

Grid Interoperation with ARC Middleware for the CMS Experiment

Erik Edelmann, Laurence Field, Jaime Frey, Michael Gronager, Kalle Happonen, Daniel Johansson, Josva Kleist, Jukka Klem, Jesper Koivumäki, Tomas Linden, Antti Pirinen, Di Qing

Abstract

The Compact Muon Solenoid (CMS) is one of the general purpose experiments at the CERN Large Hadron Collider (LHC). CMS computing relies on different grid infrastructures to provide computational and storage resources. The major grid middleware stacks used for CMS computing are gLite, Open Science Grid (OSG) and ARC (Advanced Resource Connector). Helsinki Institute of Physics (HIP) hosts one of the Tier-2 centers for CMS computing. CMS Tier-2 centers operate software systems for data transfers (PhEDEx), Monte Carlo production (ProdAgent) and data analysis (CRAB). In order to provide the Tier-2 services for CMS, HIP uses tools and components from both ARC and gLite grid middleware stacks. Interoperation between grid systems is a challenging problem and HIP uses two different solutions to provide the needed services. The first solution is based on gLite-ARC grid level interoperability. This allows to use ARC resources in CMS without modifying the CMS application software. The second solution is based on developing specific ARC plugins in CMS software.

Presented at *17th International Conference on Computing in High Energy and Nuclear Physics (CHEP09),21 - 27 March 2009,Prague,Czech Republic,15/05/2009*

Grid Interoperation with ARC Middleware for the CMS Experiment

Erik Edelmann[2], Laurence Field[3], Jaime Frey[4], Michael Grønager[2], Kalle Happonen[1], Daniel Johansson[2], Josva Kleist[2], Jukka Klem[1], Jesper Koivumäki[1], Tomas Lindén[1], Antti Pirinen[1], Di Qing[3]

1 Helsinki Institute of Physics, P.O. Box 64, FIN-00014 University of Helsinki, Finland

2 Nordic DataGrid Facility, Kastruplundgade 22, 1., DK-2770 Kastrup, Denmark

3 CERN, CH-1211 Genève 23, Switzerland

4 University of Wisconsin-Madison, 1210 W. Dayton St., Madison, WI, USA

Corresponding author E-mail: Jukka.Klem@cern.ch

Abstract. The Compact Muon Solenoid (CMS) is one of the general purpose experiments at the CERN Large Hadron Collider (LHC). CMS computing relies on different grid infrastructures to provide computational and storage resources. The major grid middleware stacks used for CMS computing are gLite, Open Science Grid (OSG) and ARC (Advanced Resource Connector). Helsinki Institute of Physics (HIP) hosts one of the Tier-2 centers for CMS computing. CMS Tier-2 centers operate software systems for data transfers (PhEDEx), Monte Carlo production (ProdAgent) and data analysis (CRAB). In order to provide the Tier-2 services for CMS, HIP uses tools and components from both ARC and gLite grid middleware stacks. Interoperation between grid systems is a challenging problem and HIP uses two different solutions to provide the needed services. The first solution is based on gLite-ARC grid level interoperability. This allows to use ARC resources in CMS without modifying the CMS application software. The second solution is based on developing specific ARC plugins in CMS software.

1. Computing in CMS experiment

CMS experiment faces a very important data analysis task, which is solved by a worldwide distributed computing system using grid technology in a heterogeneous grid environment. The Worldwide Large Hadron Computing Grid (WLCG) [1] is the entity operating, monitoring and supporting the grid sites for the LHC-experiments. The CMS computing system needs to operate on sites using ARC [2] and gLite middleware as well as on Open Science Grid (OSG) [3]. The details of this computing system have been described in references [4][5]. The computing system consists of a hierarchy of sites with the CERN Tier-0 being responsible for data taking and raw data tape archival. The Tier-1 sites are responsible for calibration and reconstruction of the data and have custodial responsibility of the second copy of the raw data. The Tier-2 sites host user analysis and generate Monte Carlo archived at the corresponding Tier-1 site. The CMS Tier-2 main applications are CMS Remote Analysis Builder (CRAB) [6], which is a tool to create, submit, and retrieve grid analysis jobs and ProdAgent which is a tool for Monte Carlo generation [7][8].

In addition to these applications several services need to be available as well for CMS and WLCG. CMS sites need a Storage Element (SE) with a SRM interface (often dCache), a PhEDEx server for data transfers between sites and a Frontier open source squid web cache for calibration data. CMS site monitoring is done using CMS SAM CE, CMS SAM SRM and JobRobot [9]jobs. The JobRobot uses CRAB to submit analysis jobs to sites every four hours. CMS stores site information in SiteDB. The CMS software framework CMSSW is mostly installed by grid jobs. WLCG expects site information in the GOCDB and monitors sites using WLCG SAM jobs. Job accounting information needs to be extracted and stored in the APEL database.

2. ARC grid middleware

The Advanced Resource Connector (ARC) middleware [10] has initially been developed in the Nordic countries by the NorduGrid collaboration [11]. It is in use in more than ten countries with a CPU core count of more than 30 000. The NorduGrid infrastructure has been available 24/7 since 2002. The Nordic Data Grid Facility (NDGF), the distributed Tier-1 computing facility uses ARC for LHC- and other VOs. NDGF is funded by Denmark, Finland, Norway and Sweden. NDGF has mostly manpower and very little hardware, since the computing resources are hosted and owned by the participating institutions.

ARC is developed by the NorduGrid collaboration and by the EU funded (6 MEUR) KnowARC project as well as by research projects using ARC. There are two branches of ARC, the ARC1 development branch by KnowARC and the stable production branch used by NDGF. ARC binaries are available for Debian, Fedora, OpenSuse, RedHat and Ubuntu Linux distributions and 64-bit support has been available since at least 2005. The ARC User Interface (UI) is trivial to install, since it only requires a wget to get the tarball, tar to unpack it and then to source the initialization script.

ARC has been designed to avoid single points of failure. Brokering is done directly by the UI to avoid scaling problems of a centralized brokering service. Usually no middleware is required on Worker Nodes (WN) since the GridManager handles the file I/O from Storage Elements to the WNs. Jobs run only when the GridManager has transferred the input files, so the CPU efficiency can be higher than in the gLite model [12].

The ARC GridManager supports a caching mechanism for input files, which can reduce file I/O significantly. Several GridManager servers can be used at large sites to increase the bandwidth as

needed to match the number of WNs. The jobs session directories can be served by several (NFS) servers to increase the available internal bandwitdth to the WNs and to ensure stability.

The European Grid Initiative (EGI) [13] will develop the Universal Middleware Distribution [14] ARC components have been selected to be part UMD together with gLite and UNICORE.

3. ARC and CMS Computing in Finland

The Finnish grid infrastructure M-grid is built on ARC middleware and the Compute Elements (CE) used by the Finnish CMS Tier-2 are been part of M-grid. Future Finnish grid projects are likely to use ARC. Knowledge and support is mostly available for ARC, since ARC is the most widely used middleware in Finland. ALICE and CMS in Finland share the same CPU and disk hardware. ALICE in Finland is part of Nordic Data Grid Facility (NDGF), the distributed Tier-1 resource, which uses ARC middleware, so using ARC for CMS as well reduces middleware maintenance work, since only one middleware flavor is needed. ARC development has mainly been driven by a bottom up approach where user and experiments needs are taken into account and bugs are fixed in a timely manner.

To use ARC for CMS, the CMS grid applications need to be adapted for ARC middleware. The goals of this task is to enable CMS to run on ARC transparently with ProdAgent and CRAB as well as to enable local users to use CMS tools natively with ARC.

3.1. Description of Tier-2 center used in interoperation tests and development

Development and testing of interoperation solutions was carried out at the Finnish CMS Tier2 center, called T2_FI_HIP using two ARC Compute Elements (CE) and a SRM dCache Storage Element (SE). On the ARC CEs the Rocks 4.1 Linux distribution and SGE batch system have been installed as well as the software required for interoperation tests (gLite runtime environment, etc).

The main CE is a HP Proliant DL 145 G2 cluster called sepeli.csc.fi, containing 128 nodes and 512 2.2GHz AMD Opteron cores, with a HP Proliant DL 585 front-end and four HP DL 385 disk servers. The other CE, ametisti.grid.helsinki.fi is a HP DL 145 G1 cluster with 260 cores shared with other university departments. It has been described in reference [15].

The storage system used in the interoperation tests is build on Hitachi Data Systems TagmaStore AMS1000 and it has 280 times 750 GB SATA hard disks. During the tests, 100 TB of storage space was reserved for CMS experiment. The storage system provides a standard SRM interface using dCache software.

4. Grid Interoperation

Interoperability means that grid systems are able to exchange information and that grid jobs can move from one grid system to another. As an example, grid jobs can be submitted with gLite tools and executed in ARC computing resources. Interoperation is the actual use of interoperable systems.

There are several ways in which users can profit from resources that run different grid middlewares. One simple model is that two or more grid middlewares are installed in parallel at one computing cluster. This is rather simple to achieve but does not provide real grid interoperation. In addition, system administration load is increased. Another model is to build bridges (also called gateways) between different grid systems. The gateway build between gLite and ARC systems is explained in the following sections. A user oriented approach to use different grids is building interfaces or plugins in user applications so that the applications can send jobs to several grid systems. This has been done for many CMS applications like CRAB and ProdAgent where special sets of plugins take care of interaction with grids (LCG, OSG and NorduGrid/ARC) or with batch systems. Another goal is that grid system developers find common standards in e.g. job submission and in information systems so that grid systems can directly exchange information. Grid standardization work is done in e.g. OGF-GIN group and some progress has been made concerning information systems schemas (Glue version 2), job description language (JSDL) and in job submission (OGSA-BES). Some of these are not yet complete standards for production grids and it takes time before common standards are implemented in gLite and ARC middlewares.

4.1. Requirements for grid interoperation in Finland

In order for HIP to support CMS computing, there are various CMS applications that have to be able to use HIP resources. These applications already have support for the gLite and OSG grid middleware but they lack the support for submitting jobs to sites running the ARC middleware.

The main CMS applications that need to work with ARC middleware include

- ProdAgent for the execution of CMS Monte Carlo simulation jobs.
- CRAB for the preparation and execution of CMS grid analysis jobs.
- WLCG and CMS SAM tests in order to check that sites and services are available and working

To achieve ARC support in CMS applications there are basically two different methods. CRAB and ProdAgent are modular programs, and they interface to grid middleware using plugins. This means that adding ARC support requires a new set of grid interface plugins for the software. This option provides application level interoperability. The second option is to have the interoperability at the grid middleware level. In this option the different grid middlewares operate together transparently, and jobs submitted using gLite can end up on ARC computing element. The CMS applications can continue using the existing gLite plugins without any changes. This option provides grid level interoperability.

Supporting application level interoperability is simpler in terms of amount of development and software that have to work together. However, it has to be implemented separately for each CMS application. Grid level interoperability requires a rather complicated gateway system, but it gives application independent grid level interoperability, and only has to be implemented once.

ARC support in CMS applications is implemented both using application and grid level interoperability. This choice was made due to time constraints and available previous work. The ProdAgent software users ARC resources with ARC interface plugins. CRAB software and the CMS SAM CE jobs currently rely on grid level interoperability between gLite and ARC.

4.2. WMS gateway scheme/grid level interoperation

The principles of WMS gateway scheme for gLite-ARC interoperation are shown in Figure 1.

Grid level interoperability between gLite and ARC has already been a goal during earlier EGEE projects. In the EGEE-II project this reached a stage where test jobs could successfully be submitted from gLite to ARC [17]. These were simple test jobs which tested the interoperability only superficially.

To have basic interoperability, i.e. submitting jobs from gLite to ARC, four requirements need to fulfilled.

- gLite must
  1) be aware of resources using ARC
  2) be able so submit jobs to ARC in a way that ARC can receive them and receive the output or report error conditions
- ARC must
  3) understand the user authorization rules of gLite
  4) be able to execute the jobs that gLite sent

These four points are solved separately. For the first point, there needs to be translation from the ARC information system to the gLite information system. The second point is solved in the gLite Workload Management Service, which is responsible for job brokering and submission in gLite. ARC supports the VOMS proxy certificate authorization scheme used in gLite, which addresses point three. For the last point, the gLite Worker Node software stack needed to be installed on the ARC worker nodes.

The most complex part of this interoperability scheme is WMS job handling. The EGEE-II project has developed an interoperability module for the WMS. Condor handles the actual job scheduling and submission when it has been accepted by the WMS. A Condor module, NorduGrid GAHP, handles the job submission to ARC clusters. The ARC clusters are identified by their contact string in the information system. The NorduGrid GAHP module creates an ARC xrsl job description file, and submits the job to the ARC site using a generic gridftp interface, using the original user's proxy certificate. It then monitors the status of the job, and retrieves the output of the job once it is finished.

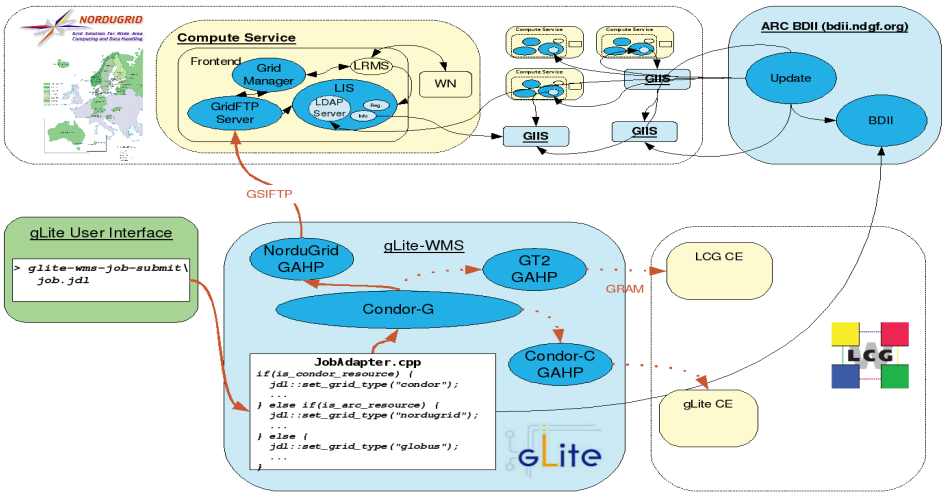


Figure 1: Illustration of WMS gateway scheme (gLite to ARC job submission).

The WMS module had been successfully tested with proof-of-concept test jobs in the end of the EGEE-II project. For production usage, this implementation needed improvement. Initially with the existing interoperability module, only 5 jobs could be submitted successfully out of 10 jobs. Submitting a larger amount of jobs would only decrease the success rate to about 20 % for 1000 jobs after the Condor-G configuration had been improved. The NorduGrid GAHP module was also improved. The ARC server's job handling and acceptance was improved, which eliminated some error conditions seen during submission and slightly improved the job submission success rate. For jobs that run over 20 minutes, the success rate was 0%. This problem was due to a check in the gLite WMS that expected jobstatus output in a certain amount of time. Removing the test resulted in a job submission success rate of 99 % or better for 2000 submitted jobs [12].

4.2.1. Information system translation

Both ARC and gLite information systems rely on LDAP for information sharing. This means that the main difference between them is the LDAP schemas they use. In earlier EGEE-II interoperability work, a basic ARC to gLite information system translation service was produced [17]. This service reads the information from an ARC GRIS, translates the relevant attributes, and publishes this information to a gLite site BDII service. This translation is not complete, and not all attributes have a direct translation.

For CMS analysis work, several improvements in the translation had to be made. The main change was adding local storage element advertising. There was also significant effort in correcting the format of published information. The original format of many translated attributes was syntactically correct, but did not correspond to the undocumented semantics of the information system. Even though the information system translation does not consider the full set of available attributes at the moment, the current subset of translated information should be sufficient for almost all grid use-cases.

4.2.2. Interoperation requirements for ARC Computing Elements

ARC computing elements are not able to run gLite jobs without additional configuration. The first issue is user authorization. ARC mainly uses gridmap files for authorization. This means that authorized users' certificate subjects are collected, and their mapping to local user groups is decided in advance. The WLCG's use of VOMS to manage VOs complicates this. The VOMS includes additional features, including mapping of users' proxy certificates to different roles on request. VOMS does provide an interface where you can retrieve the users' certificate subjects, and compile gridmap files. VOMS even supports retrieving a list of users who are mapped to a certain role. With this scheme you can even have partial support for VOMS roles on ARC using traditional gridmap files. However, since the user role information is retrieved in advance, users with multiple roles are unable to choose their role at run-time, which is the main benefit of using roles. This means that they will be mapped to which ever role is configured for them on the target system.

In new versions of ARC, an early version of support for dynamic VOMS usage is available. This feature allowed the usage of dynamic mapping of roles on the ARC computing element. Even if the support was largely undocumented, and in early stages, once set up, it has worked without problems.

The second issue with executing gLite jobs on ARC clusters, is that unlike the ARC middleware, gLite jobs expects to find a gLite worker node environment when the jobs are being executed. The

gLite middleware stack does provide separate easily installable worker node installations. By combining these with specific gLite runtime environments for ARC, the ARC CEs could be made to support the execution of gLite jobs.

4.3. Interoperation with plugins

CMS applications have been developed in such a way that interaction with different grid systems is only coded in plugin components. General grid related tasks like job submission, job monitoring and results gathering are programmed in a set of plugins for each grid flavour. The plugin approach makes it easier to add support for new grid systems in CMS applications. However, a good knowledge of CMS applications is still required when developing and testing new plugins. The work done in plugin development for ARC middleware is explained in the following sections about CMS data analysis and CMS Monte-Carlo production.

5. CMS data management and data transfers with ARC

Most of CMS data management tasks can be carried out by using ARC, gLite or even dCache client tools. The storage systems used in CMS (dCache, CASTOR, DPM etc.) provide standard SRM interface which makes interoperation easier.

For data transfers CMS uses the Physics Experiment Data Export (PhEDEx) [16] system. CMS data transfers are initiated by PhEDEx agents running in different CMS sites. The distributed agents communicate using a central Transfer management database (TMDB). Figure 2 shows the cumulative data transfer volume during one year from all CMS sites to an ARC based site. Figure 3 shows an example of data transfer quality (green means no errors, white means no transfer attempts) from all the seven CMS Tier-1 sites and from CERN to a Tier-2 site running ARC middleware. These transfers are done in PhEDEx debug instance and they are very useful in monitoring data transfer quality [18].

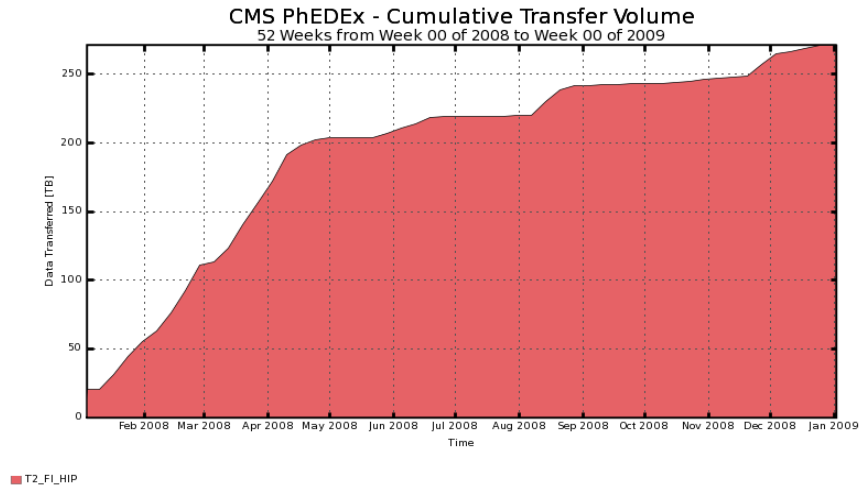


Figure 2: Volume of CMS data transfers to an ARC based site (PhEDEx Debug instance).

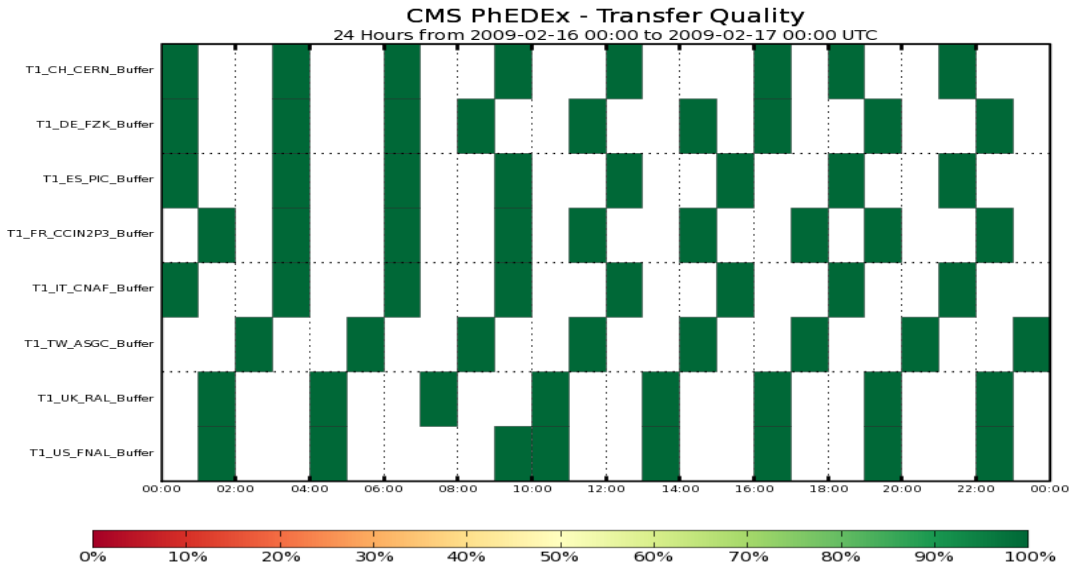


Figure 3: Data transfer quality from CMS Tier-1 sites to an ARC based site in PhEDEx debug instance.

Other CMS systems like the Dataset Bookkeeping System (DBS) and conditions data access using FroNTier do not depend on grid middleware flavour and can be easily used in an ARC site.

6. ARC in CMS data analysis

CMS data analysis can be carried out on ARC resources using the WMS gateway scheme described in earlier chapters. This scheme allows users to send standard CRAB jobs to ARC CEs without any job modifications provided that a gLite Runtime Environment (RE) is installed on the CE.

During this work, only the patched gLite WMS called arcwms.cern.ch was able to submit jobs to ARC resources. When the required patches are deployed, all the gLite WMS's are able to submit to ARC.

ARC specific plugins for CRAB are developed so that CRAB can directly submit jobs to ARC resources without depending on gateway between gLite and ARC. Presently most of the plugins have been developed and are being tested with job submission.

ARC middleware features like prestaging of the input files by ARC GridManager may improve CPU efficiency because CPUs do not have wait idle when input files are copied from a storage element. The copying of job output files to a remote storage element is also done by ARC GridManager and this may eliminate many of the problems encountered by CMS Analysis Support Task Force [19].

6.1. *CMS JobRobot*

The CMS Job Robot [9] is an automated system to automatically submit and manage fake analysis jobs using CRAB. It is used as a commissioning tool to test if a site is capable to run certain CMS workflows at the required scale. The Job Robot run in an ARC site (T2_FI_HIP) using the gateway based grid interoperation.

6.2. *CMS analysis on ARC resources using glideinWMS*

The glideinWMS [20] allows to send CMS jobs to many kinds of CEs including ARC CEs using Condor-G. This was demonstrated for the first time in large scale during CMS Computing Software and Analysis challenge of 2008 (CSA08) .

7. ARC in CMS Monte-Carlo production

Within CMS there is a need to generate at least initially as much Monte Carlo events as real events, a task that can easily be distributed over many contributing sites. To automate the task, a tiered workload management system has been designed [7], where the workload is distributed over a bunch of "ProdAgent" instances. Each ProdAgent instance has some resources in the form of computing elements (CEs) and storage elements (SEs), typically a Tier-1 or Tier-2 site. Production jobs are generated from workflows, that are either pulled from a central server called the ProdMgr, or injected by the local operator. The data being produced is initially stored at a local SE, and registered in the CMS Dataset Bookkeeping System (DBS), after which it can be transferred with PhEDEx to the hosting Tier-1 site.

ProdAgent accesses the resources it has at it's disposal through the grid middleware in use at the site. Because of this, ProdAgent has to support several different grid middlewares and batch queue systems. To achieve this, the parts of ProdAgent that need to interact directly with the grid middleware has been isolated to a small sets of components, each with a set of plugins for the various supported middlewares. For the needs of the Finnish Tier-2 site, a set of plugins for NorduGrid ARC has been developed. With a total of roughly 1600 lines of python code, the plugins are

- ARCCreator: Generates job scripts
- ARCMonitor: Releases jobs from the internal queue when free resources are available.
- ARCSubmitter: Submits jobs
- ARCTracker: Tracks the status of jobs, and fetches them when they are done.
- ARCKiller: Kills jobs

This set of plugins matured during the autumn and winter of 2008 and 2009, and was taken into production during spring 2009. During approximately 1.5 week of production use at the Finnish Tier-2 site, 4 workflows, with close to 400000 jobs, have been processed. The datasets (e.g. /H200_ZZ_2lqq/Summer08_IDEAL_V9_v1/GEN-SIM-RECO) produced by these workflows are available in the CMS Data Bookkeeping System (DBS).

8. Summary and Outlook

CMS software can run on ARC resources as a result of the work done with two grid interoperation approaches: WMS gateway based grid interoperation and the application specific plugins. WMS gateway approach is used for CRAB analysis jobs and specific plugins have been developed for CMS production software.

Many issues with WMS gateway approach are debugged and improved. These include for example applying the required patches on all WMS servers, solving VOMS related problems, better jobstatus information and making it possible to send jobs from CRABserver to ARC. ARC specific plugins for CRAB are developed so that CRAB can send jobs directly to ARC resources.

Building large scale grid interoperation without widely implemented common standards has proven to be time consuming. Hopefully the current efforts with common information system schema (GLUE version 2), common job description language (JSDL) and common job submission interfaces will make grid interoperation easier in the future. The interoperation solutions described in this paper (WMS gateway and ARC plugins in major CMS applications) make sure that CMS data can be analysed also using ARC resources when LHC collisions data becomes available later in 2009.

References

[1] The Worldwide LHC Computing Grid (WLCG), http://lcg.web.cern.ch/LCG/
[2] O. Smirnova et al., “ARC middleware: evolution towards standards-based interoperability”, these proceedings.
[3] The Open Science Grid, http://www.opensciencegrid.org/
[4] CMS Computing TDR, http://cmsdoc.cern.ch/cms/cpt/tdr/
[5] CMS Computing Model, http://cmsdoc.cern.ch/documents/04/note04_031.pdf
[6] CRAB, https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrab
[7] D. Evans et al., “CMS MC Production System Development & Design”, Proceedings of Computing in High Energy Physics 2007, Victoria, British Columbia, Canada, September 2-7, 2007.
[8] ProdAgent, https://twiki.cern.ch/twiki/bin/view/CMS/ProdAgent
[9] CMS JobRobot, https://twiki.cern.ch/twiki/bin/view/CMS/JobRobot.
[10] NorduGrid/ARC middleware, http://www.nordugrid.org/middleware
[11] NorduGrid, http://www.nordugrid.org/
[12] M. Grønager et al. "Interoperability between ARC and gLite – understanding the grid-job life cycle ", in Proceedings of 4th IEEE International Conference on e-Science, Indianapolis, Indiana, USA, December 7-12, 2008.
[13] European Grid Initiative (EGI), http://web.eu-egi.eu/
[14] Universal Middleware Distribution, http://knowledge.eu-egi.eu/knowledge/index.php/UMD
[15] T. Lampén et al., “Testing TMVA software in b-tagging for the search of MSSM Higgs bosons at the LHC”, J. Phys.: Conf. Ser. 119 Volume 119 (2008) 032028 (9pp).
[16] PhEDEx, http://cmsweb.cern.ch/phedex
[17] M. Grønager et al. “LCG and ARC Middleware Interoperability” , in Proceedings of Computing in High Energy Physics 2006, Mumbai, India, February 13-16, 2006.
[18] G. Bagliesi et at., “Debugging Data Transfers in CMS”, these proceedings.
[19] J. Andreeva et al., “CMS Analysis Operations”, these proceedings.
[20] S. Padhi et al., “Use of glide-ins in CMS for production and analysis”, these proceedings.