International Conference on Computing in High Energy and Nuclear Physics (CHEP'07) IOP Publishing
Journal of Physics: Conference Series **119** (2008) 062041 doi:10.1088/1742-6596/119/6/062041

Complete Distributed Computing Environment for a HEP Experiment: Experience with ARC-Connected Infrastructure for ATLAS

A. Read[1], A. Taga[1], F. Ould-Saada[1], K. Pajchel[1], B. H. Samset[1], D. Cameron[1,2]

[1] Department of Physics, University of Oslo, P.b. 1048 Blindern, N-0316 Oslo, Norway
[2] Nordic Data Grid Facility, Kastruplundgade 22, DK-2770 Kastrup, Denmark

E-mail: a.l.read@fys.uio.no

Abstract.
Computing and storage resources connected by the Nordugrid ARC middleware in the Nordic countries, Switzerland and Slovenia are a part of the ATLAS computing Grid. This infrastructure is being commissioned with the ongoing ATLAS Monte Carlo simulation production in preparation for the commencement of data taking in 2008. The unique non-intrusive architecture of ARC, its straightforward interplay with the ATLAS Production System via the Dulcinea executor, and its performance during the commissioning exercise is described. ARC support for flexible and powerful end-user analysis within the GANGA distributed analysis framework is also shown. Whereas the storage solution for this Grid was earlier based on a large, distributed collection of GridFTP-servers, the ATLAS computing design includes a structured SRM-based system with a limited number of storage endpoints. The characteristics, integration and performance of the old and new storage solutions are presented. Although the hardware resources in this Grid are quite modest, it has provided more than double the agreed contribution to the ATLAS production with an efficiency above 95% during long periods of stable operation.

1. Introduction
The path from recording particle collision events in a detector to discovering new physics is long and complicated. For the Large Hadron Collider (LHC) experiments at CERN, the use of Grid technologies to distribute computational work and data storage makes constructing the process more complicated at first, but in the end makes things more efficient. The ATLAS experiment, which has the largest data output of all the LHC experiments, exports data from CERN and runs production of simulated Monte Carlo data on 3 Grids. At computing centres in Scandinavia, Switzerland and Slovenia, ARC middleware is used to connect resources to NorduGrid for the use by ATLAS physicists in these countries. The design of this middleware makes it easy to construct applications which harness NorduGrid resources for both production and analysis of physics data. In this paper we describe the "full chain" from receiving data into NorduGrid from the detector or by simulation production through to end user analysis and discovery of important physics phenomena.

Firstly, in Section 2 we explain the concepts of NorduGrid and how it enables large-scale physics computation with no expert knowledge required by users. The unique solution to data storage and how it fits the experiments' requirements is also shown in this section. Then in Section 3 we describe the ATLAS distributed data management system which is used for transfer and bookkeeping of all ATLAS data. We explain how this sytem implements the ATLAS computing model to enable the data required by physicists to be in the right place at the right time. The system used in NorduGrid for producing simulated data is then discussed in Section 4, and we present some results showing the efficiency of the system. In Section 5 we show the GANGA software which is used as an interface for analysis job submission to NorduGrid. Some examples of physics results obtained using GANGA and NorduGrid are presented in Section 6 and we summarise and conclude in Section 7.

2. Nordugrid and ARC

NorduGrid [1] is a collaboration which develops and maintains the Advanced Resource Connector (ARC), an open source software solution enabling production quality computational and data Grids. ARC software is currently deployed across 44 sites in 18 countries, shown in Figure 1. In total there are almost 7000 CPUs available across NorduGrid for scientific researchers.

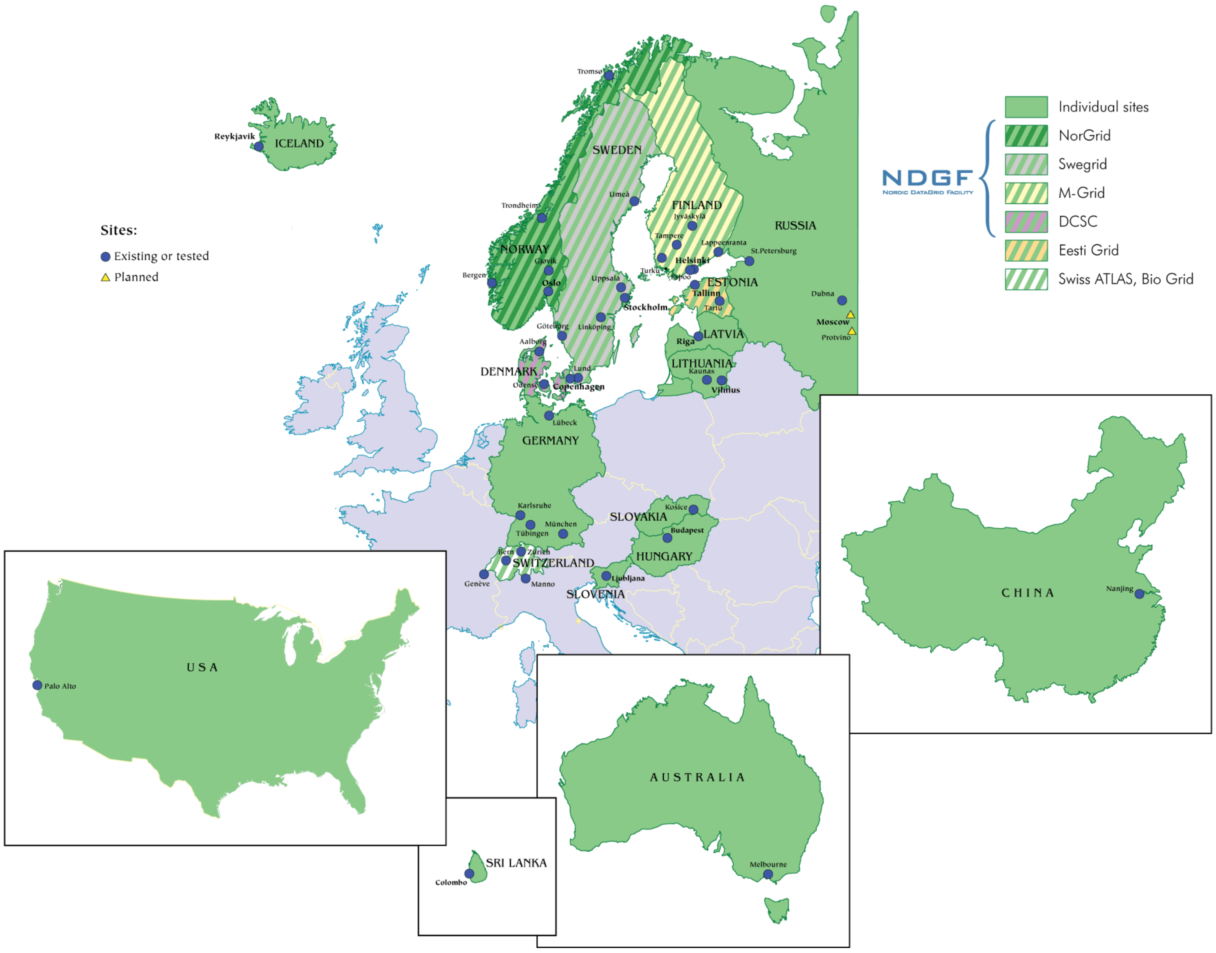


Figure 1. ARC enabled Grid sites.

ARC provides a reliable implementation of the fundamental Grid services, such as information services, resource discovery and monitoring, job submission and management, brokering and data management and resource management. The middleware builds upon standard Open Source solutions like the OpenLDAP, OpenSSL and Globus Toolkit libraries.

There are several components of ARC. Firstly, there are Grid services which run on computing resources: the Grid Manager, the ARC GridFTP server and the Local Information Service. The Grid Manager is a service running on a resource which takes care of jobs and manages a cache area to which input and output data is staged. Job submission and pre- and post-job data staging are performed through the GridFTP server. Information services populate the information database stored in the Globus-modified OpenLDAP back-ends.

Indexing services for the resources are constructed from a special setup of the Globus GIIS backend which allows building a hierarchical mesh of Grid-connected sites. ARC middleware can use a variety of data indexing services, such as the Globus RC, RLS and the LCG File Catalog (LFC). ARC client tools and the Grid Manager daemon are interfaced to these catalogs. All ATLAS data is indexed using a Globus RLS file catalog [2].

ARC provides a light-weight command line interface to submit, monitor and manage jobs, move data and obtain resource information. This interface has a built-in broker, which is able to select the best matching resource for a job. This client-based resource brokering eliminates the need for a resource broker service and is unique among the Grids used by ATLAS.

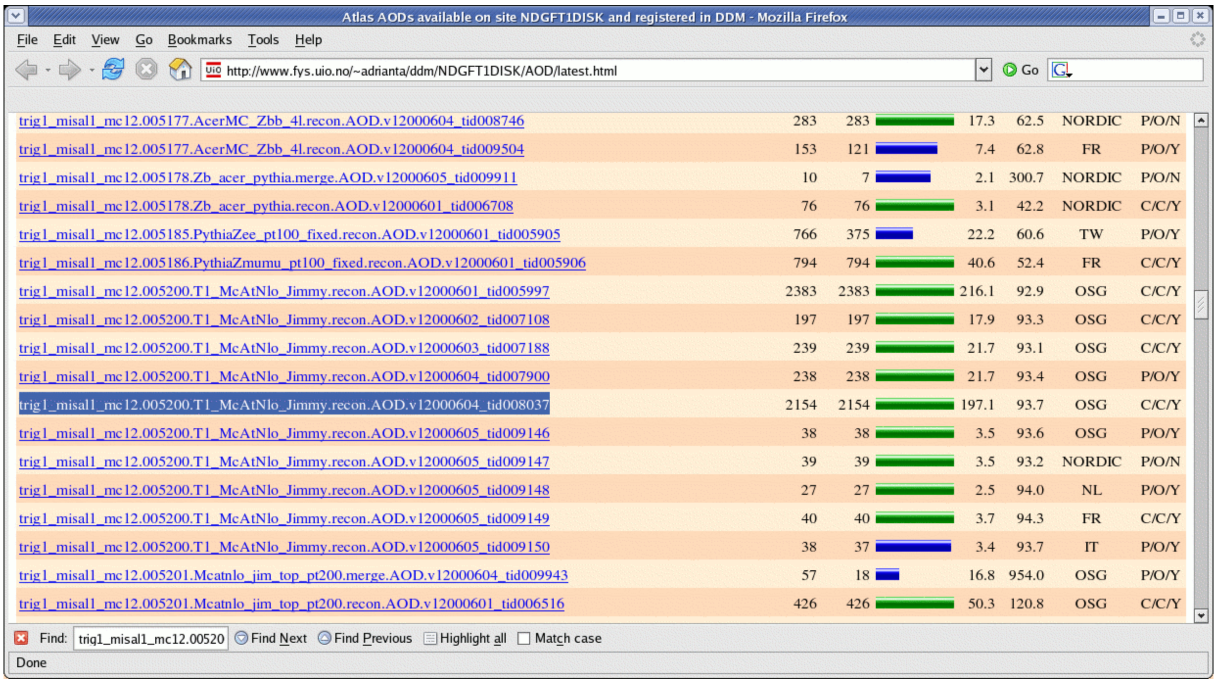
Storage Elements offer Grid-enabled secure access to disk-based storage capacity. Historically, ARC was based on a loose collection of GridFTP servers, however ATLAS and other experiments require that data be exposed via the Storage Resource Manager (SRM) interface, and that each Tier 1 centre provide a single endpoint to receive data from CERN. Since no single site in NorduGrid could provide the resources to constitute a Tier 1 centre, it was decided to adopt a distributed storage solution [3]. The Nordic DataGrid Facility (NDGF) project is charged with creating this system and offers a distributed storage system to its users based on dCache [4]. In this system the storage pools are distributed between various computing centres in the Nordic countries but they are all reachable through a common SRM endpoint at srm.ndgf.org. All incoming data from CERN and other Grids is stored at the NDGF Tier 1 centre. This system introduces a storage hierarchy into the previously flat storage resources controlled by ARC, and the modifications that were necessary for the ATLAS MC production system to deal with this are described later in Section 4.

3. ATLAS Distributed Data Managment

The ATLAS experiment will generate tens of PetaBytes of data per year, from the detector itself and through simulation production. This data will be distributed globally according to the ATLAS computing model [5]. After a first pass reconstruction at CERN, raw data from the detector and reconstructed data products are replicated to 10 Tier 1 centres according to the data type and the amount of data the centre has agreed to store. In addition to the data from the detector, simulated data is produced on Grid resources worldwide, and this data must also be replicated.

DQ2 [6] is the ATLAS distributed data management system, designed to implement the ATLAS computing model, and by interacting with the underlying Grid middleware, provides a single entry point for ATLAS users requiring access to data. The basic unit of ATLAS data is a file which contains a certain number of physics events (of order 100). In DQ2 files are aggregated into datasets, a collection of files plus associated metadata descriptive of the dataset itself and its constituent files. A set of catalogs store information on the location of datasets, their constituent files and associated system metadata. All managed data movement in the system is based on the unit of datasets and automated using a subscription system. The idea is that a site subscribes to a dataset, and DQ2 services resident at the site act to keep the site's copy of the dataset up to date with respect to any changes that might be made to the dataset over time. One such set of services is used to manage data flow into and within NorduGrid. The integration between DQ2 and ARC is explained in detail in [7].

The majority of simulated data in ATLAS is produced outside NorduGrid, but the end data products which are useful for end users wishing to perform physics analysis are replicated to the Tier 1 centre by DQ2. In order to run analysis on NorduGrid resources, the dataset must be present in NorduGrid storage, hence a web based monitoring service was created to track the progress of this replication. An example screenshot is shown in Figure 2 and shows a number of datasets typically used for analysis. Some datasets have been completely replicated (green bars) and some have been partially replicated (blue bars). The first column of numbers shows the number of files in the dataset as reported by DQ2, and the second column shows the number of files in the dataset present on NorduGrid. This page is frequently updated by examining the content of the DQ2 dataset and the files registered in the RLS file catalog.


Atlas AODs available on site NDGFT1DISK and registered in DDM - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.fys.uio.no/~adrianta/ddm/NDGFT1DISK/AOD/latest.html
Go
Find: trig1_misal1_mc12.00520
Find Next
Find Previous
Highlight all
Match case
Done

trig1_misal1_mc12.005177.AcerMC_Zbb_4l.recon.AOD.v12000604_tid008746	283	283	🟩	17.3	62.5	NORDIC	P/O/N
trig1_misal1_mc12.005177.AcerMC_Zbb_4l.recon.AOD.v12000604_tid009504	153	121	🟦	7.4	62.8	FR	P/O/Y
trig1_misal1_mc12.005178.Zb_acer_pythia.merge.AOD.v12000605_tid009911	10	7	🟦	2.1	300.7	NORDIC	P/O/N
trig1_misal1_mc12.005178.Zb_acer_pythia.recon.AOD.v12000601_tid006708	76	76	🟩	3.1	42.2	NORDIC	C/C/Y
trig1_misal1_mc12.005185.PythiaZee_pt100_fixed.recon.AOD.v12000601_tid005905	766	375	🟦	22.2	60.6	TW	P/O/Y
trig1_misal1_mc12.005186.PythiaZmumu_pt100_fixed.recon.AOD.v12000601_tid005906	794	794	🟩	40.6	52.4	FR	C/C/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000601_tid005997	2383	2383	🟩	216.1	92.9	OSG	C/C/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000602_tid007108	197	197	🟩	17.9	93.3	OSG	C/C/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000603_tid007188	239	239	🟩	21.7	93.1	OSG	C/C/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid007900	238	238	🟩	21.7	93.4	OSG	P/O/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037	2154	2154	🟩	197.1	93.7	OSG	C/C/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000605_tid009146	38	38	🟩	3.5	93.6	OSG	P/O/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000605_tid009147	39	39	🟩	3.5	93.2	NORDIC	P/O/N
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000605_tid009148	27	27	🟩	2.5	94.0	NL	P/O/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000605_tid009149	40	40	🟩	3.7	94.3	FR	C/C/Y
trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000605_tid009150	38	37	🟦	3.4	93.7	IT	P/O/Y
trig1_misal1_mc12.005201.Mcatnlo_jim_top_pt200.merge.AOD.v12000604_tid009943	57	18	🟦	16.8	954.0	OSG	P/O/Y
trig1_misal1_mc12.005201.Mcatnlo_jim_top_pt200.recon.AOD.v12000601_tid006516	426	426	🟩	50.3	120.8	OSG	C/C/Y

Figure 2. DQ2 dataset availability monitor.

4. ATLAS Monte Carlo Simulation Production

Monte-Carlo generation and reconstruction of simulated data for ATLAS is performed in order to prepare and test both the ATLAS software and the Grid infrastructure for the real data which are expected from 2008. Physics tasks consisting of up to tens of thousands of Monte Carlo simulation or reconstruction jobs are defined by physics teams and assigned to one of the 3 Grids used by ATLAS roughly according their capacities (the Nordic countries have agreed to provide 5.6% of the required computing and storage resources for the ATLAS experiment, but have consistently provided more than 10% during the last 2 years of production).

Individual job definitions, including job status and output metadata, are stored in an Oracle database at CERN (prodDB). The complete life-cycle of the Grid jobs is controlled by a supervisor (Eowyn, written in Python) which polls the prodDB database periodically for new jobs designated for execution on NorduGrid and updates the database as the status of the Grid jobs change from SUBMITTED to one of the terminal states FINISHED, FAILED or KILLED. Jobs are given different priorities and those with highest priority are picked up first.

All the interactions between the supervisor and jobs on NorduGrid are handled by the executor (Dulcinea) which performs a set of call-outs to prepare jobs, to check their status, and to post-process the results. The executor is written in Python and interacts via the Python API of the NorduGrid ARC middleware.

The NorduGrid computing resources which support the ATLAS virtual organization are collected in a dedicated information server. The information presented by this server is used by the job broker in the ARC client to direct submitted jobs to clusters with the necessary resources (e.g. sufficient memory on the compute node, the required release of ATLAS software installed) and preferentially to the most powerful clusters with unused compute nodes or short execution queues. In addition, a configuration file can be used to supress clusters which for whatever reason are not functioning correctly.

Both the input and output data are stored permanently in the distributed SRM service of the NDGF Tier 1 centre. Input datasets which are not already stored at the Tier 1 centre are replicated from other ATLAS storage sites by DQ2 (although direct access of files stored at other SRM-based ATLAS storage sites is possible, this is not encouraged by the ATLAS computing model). The transfer of input and output files to and from each Grid job is handled by the Grid Manager process running on each of the computing clusters. Each Grid Manager has its own cache thus dynamically reducing the load of transferring files from SRM multiple times, especially in the case of frequently accessed files such as detector database releases. The number of parallel file transfers by the Grid Manager is configurable thus providing a method to limit the input/output load on the front-end server. The fact that all file transfers are between a limited number of Grid Managers and the SRM endpoint (rather than a much larger number of compute nodes, as in the model for EGEE) puts a natural limit on the number of parallel accesses to the SRM endpoint. All the files in SRM are indexed in the RLS file catalog.

The ATLAS software itself is a signficant source of errors (anywhere from 10 to 90% of the total errors, depending on the type of physics task and the maturity of the ATLAS software release). The next most significant sources of error are cluster or batch system failures (full disks, filesystem overloads, power failures, etc.) and errors downloading input files or uploading output files. The up/downloading errors can most often be traced to temporary problems with either the RLS file catalog or the SRM storage service. Jobs suffering from up/downloading errors are resumed up to three times by Dulcinea before they are declared FAILED - this is especially important for jobs which fail during uploading in order to waste as little processing time as possible for otherwise succesful jobs.

The post-processing of successful jobs in the executor includes the collection of metadata (e.g. number of events, version of the detector geometry, output file characteristics, obtained by downloading an XML file produced by the ATLAS job), and resource usage (e.g. CPU and/or wall time provided by the NorduGrid information system) to be passed back to the supervisor for insertion in the production database, registering file transfer addresses (location in SRM) and file metadata in the RLS and cleaning the jobs away. The additional registration of output files in DQ2 is done by a standard module of the supervisor.

If a Grid job fails, the ATLAS and Grid logfiles are automatically uploaded to a local directory which is directly accessible to both the executor and a web-server, and a summary of the error information is passed back to the supervisor for storage in the production database. Failed jobs are retried and if they repeatedly fail due to ATLAS software problems, the problem is diagnosed and a bug report is submitted to the ATLAS bug-tracking system.

During long periods of stable operation job efficiencies over 90% (ratio of successful ATLAS jobs including uploading output files to submitted Grid jobs) and walltime efficiencies over 95% have been acheived (see Figure 3), with the majority of the errors and wasted walltime due to ATLAS software failures. In order to maximize the efficiency a new task is started by releasing only four jobs. These so called "scout jobs" test if the jobs are possible to run and if the job

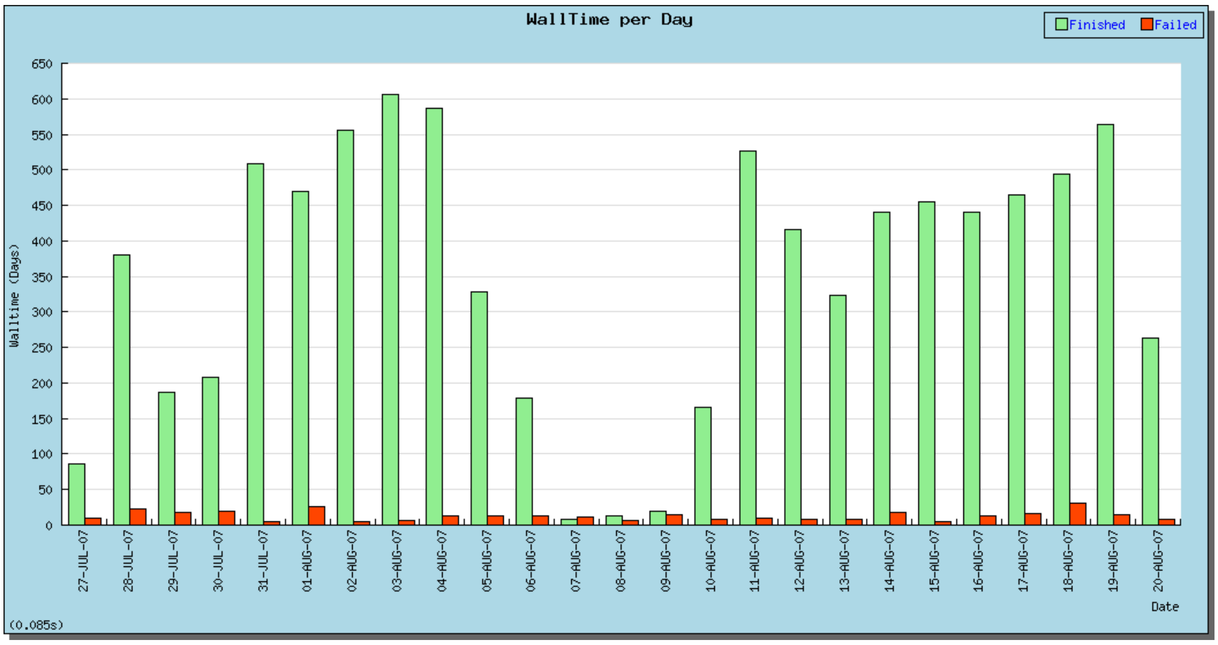


Figure 3. Wall time used per day by successful (green) and failed (red) jobs for a 25-day period in summer 2007.

parameters like requested CPU time or requested memory are suitable. The rest of the jobs are released only after successful completion of at least three scout jobs. This strategy has recently been implemented by the central job management and is now used for all three Grids.

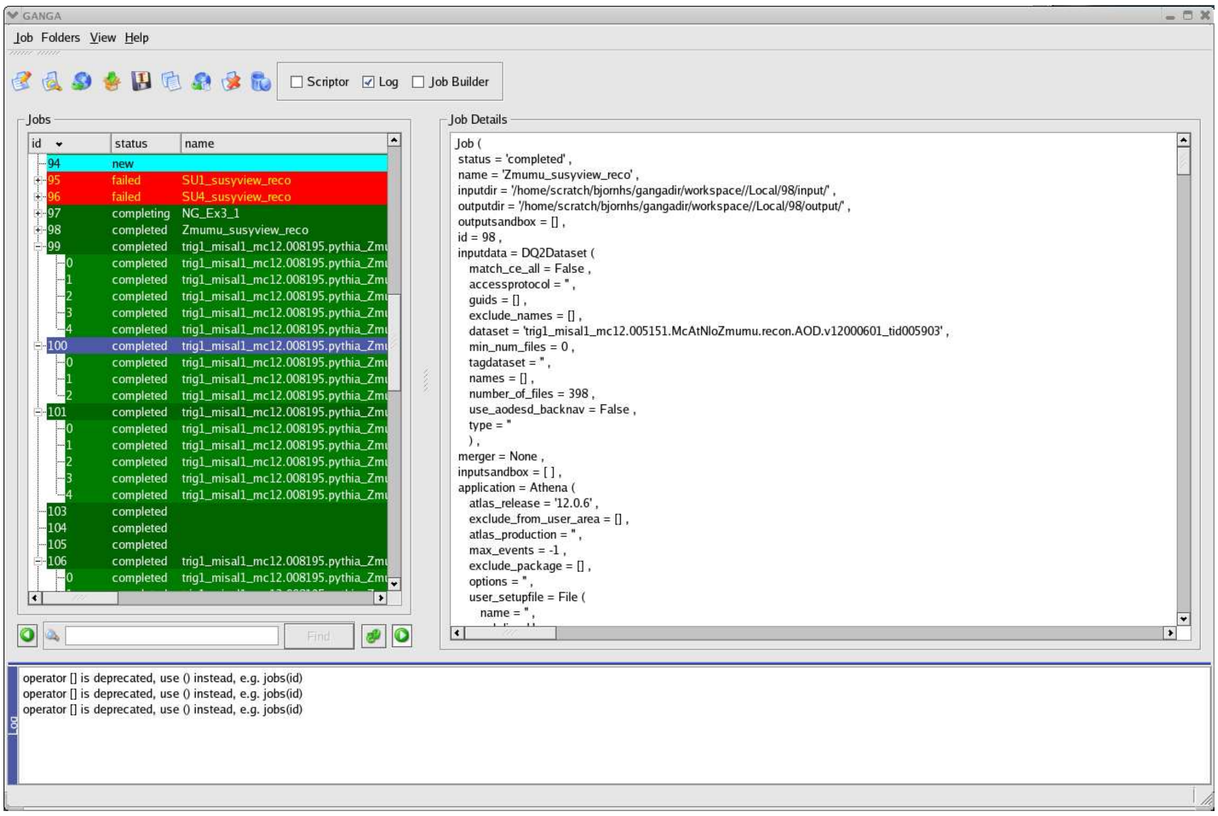
In 2006 the ATLAS production on ARC-enabled clusters processed 239k jobs taking 55k CPU-days (150 CPU-days/day averaged over the year) and during 2007 this increased to 703k jobs in total and an average of 330 CPU-days/day. At the time of writing, approximately 2.5 million files are registered in the ATLAS RLS file catalog, including both the production on NorduGrid and replication of datasets from other ATLAS storage sites.

There have been between 13 and 17 clusters spread across Scandanavia, Switzerland and Slovenia participating in the production with a total of between 2 and 4 thousand CPUs of which we manage to use 10-20% depending on the fraction of the resources configured to be available to Grid usage and competition among the Grid users. The number of clusters and CPUs has steadily increased during 2006-7. The record for the peak production integrated over a day is almost 1000 CPU days and the peak number of ATLAS jobs running concurrently is around 1200.

5. Distributed User Analysis with GANGA

While initial reconstruction and distribution of ATLAS data from the detector will be performed centrally by the collaboration, at a certain point in the chain experimenters need to be given access to the data. Though much reduced, the data amounts are still such that it is not practical to store it all at each participating institution. High energy physics Grid systems must therefore allow for personalized distributed analysis. This means that an experimenter must be able to write his or her own analysis code, submit the code to the Grid and have it meet up with the data on a worker node. The results can then either be copied back to the experimenter, or be put directly on the Grid for common access.

In ATLAS, the most versatile tool for doing this is the GANGA job definition system [8]. It



id	status	name
94	new	
95	failed	SU1_susyview_reco
96	failed	SU4_susyview_reco
97	completing	NG_Ex3_1
98	completed	Zmumu_susyview_reco
99	completed	trig1_misal1_mc12.008195.pythia_Zmu
0	completed	trig1_misal1_mc12.008195.pythia_Zmu
1	completed	trig1_misal1_mc12.008195.pythia_Zmu
2	completed	trig1_misal1_mc12.008195.pythia_Zmu
3	completed	trig1_misal1_mc12.008195.pythia_Zmu
4	completed	trig1_misal1_mc12.008195.pythia_Zmu
100	completed	trig1_misal1_mc12.008195.pythia_Zmu
0	completed	trig1_misal1_mc12.008195.pythia_Zmu
1	completed	trig1_misal1_mc12.008195.pythia_Zmu
2	completed	trig1_misal1_mc12.008195.pythia_Zmu
101	completed	trig1_misal1_mc12.008195.pythia_Zmu
0	completed	trig1_misal1_mc12.008195.pythia_Zmu
1	completed	trig1_misal1_mc12.008195.pythia_Zmu
2	completed	trig1_misal1_mc12.008195.pythia_Zmu
3	completed	trig1_misal1_mc12.008195.pythia_Zmu
4	completed	trig1_misal1_mc12.008195.pythia_Zmu
103	completed	
104	completed	
105	completed	
106	completed	trig1_misal1_mc12.008195.pythia_Zmu
0	completed	trig1_misal1_mc12.008195.pythia_Zmu

```
Job (
 status = 'completed' ,
 name = 'Zmumu_susyview_reco' ,
 inputdir = '/home/scratch/bjornhs/gangadir/workspace//Local/98/input/' ,
 outputdir = '/home/scratch/bjornhs/gangadir/workspace//Local/98/output/' ,
 outputsandbox = [] ,
 id = 98 ,
 inputdata = DQ2Dataset (
   match_ce_all = False ,
   accessprotocol = '' ,
   guids = [] ,
   exclude_names = [] ,
   dataset = 'trig1_misal1_mc12.005151.McAtNloZmumu.recon.AOD.v12000601_tid005903' ,
   min_num_files = 0 ,
   tagdataset = '' ,
   names = [] ,
   number_of_files = 398 ,
   use_aodesd_backnav = False ,
   type = ''
   ) ,
 merger = None ,
 inputsandbox = [ ] ,
 application = Athena (
   atlas_release = '12.0.6' ,
   exclude_from_user_area = [] ,
   atlas_production = '' ,
   max_events = -1 ,
   exclude_package = [] ,
   options = '' ,
   user_setupfile = File (
     name = '' ,
```

```
operator [] is deprecated, use () instead, e.g. jobs(id)
operator [] is deprecated, use () instead, e.g. jobs(id)
operator [] is deprecated, use () instead, e.g. jobs(id)
```

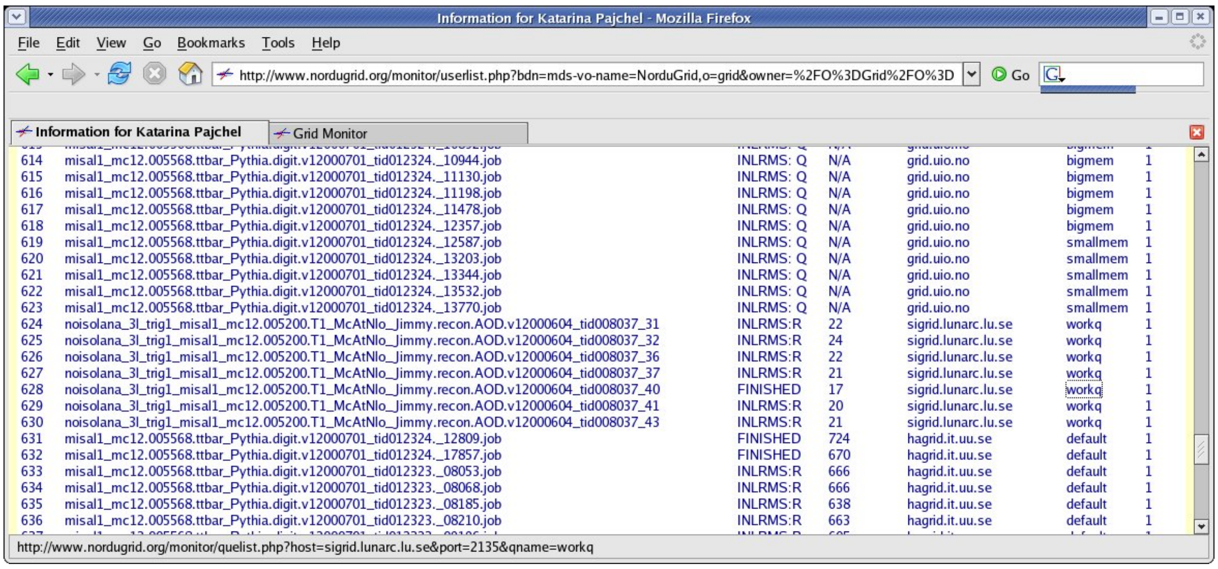
Figure 4. The GANGA graphical user interface. On the left is a list of new and completed jobs with various status flags, and on the right is part of the job definition for a selected job.

can submit jobs to all three Grids used in ATLAS, and in particular it is NorduGrid–aware as it comes bundled with a pre–compiled version of the ARC middleware.

GANGA (see Figure 4) is a python–based framework for defining 'jobs' and submitting them to various backends. A 'job' in this case is a task to be performed, typically run a program (like 'echo') with a specific input ('Hello world'). Input data can be specified, either from the user's local computer or from the Grid through specifying a DQ2 dataset. Jobs can be automatically split into subjobs and their outputs merged after completion. Output data can also either be downloaded, or put directly onto the Grid for easy common access. Finally, GANGA acts as a common, mostly seamless interface for submitting the configured job to either a local computer, a batch system (e.g. Condor or LSF), or a Grid. The interaction with the middleware is hidden from the user, and switching from a local test to a full–scale Grid submission is simply a matter of switching the back-end definition of the job.

To allow submission to NorduGrid, GANGA is shipped with a recent stable version of the ARC middleware. At job submission time, the backend handler in GANGA writes a correctly formatted XRSL file (used to specify job requirements and parameters), and submits it using the regular ARC command line interface. Resource availability is determined by the users Grid certificate. As GANGA is a python–based package, the arclib python bindings could also easily have been used for this purpose. The light–weight nature of the ARC standalone client, with a total size of less than 20 MB, made integrating it fully into a third party distribution like GANGA a natural and easy thing to do.

After submission, user jobs can be tracked through GANGA's own status monitoring. The NorduGrid monitor, shown in Figure 5, is a web based tool that can give a detailed description


Information for Katarina Pajchel - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.nordugrid.org/monitor/userlist.php?bdn=mds-vo-name=NorduGrid,o=grid&owner=%2FO%3DGrid%2FO%3D
Go
Information for Katarina Pajchel
Grid Monitor

614	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._10944.job	INLRMS: Q	N/A	grid.uio.no	bigmem	1
615	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._11130.job	INLRMS: Q	N/A	grid.uio.no	bigmem	1
616	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._11198.job	INLRMS: Q	N/A	grid.uio.no	bigmem	1
617	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._11478.job	INLRMS: Q	N/A	grid.uio.no	bigmem	1
618	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._12357.job	INLRMS: Q	N/A	grid.uio.no	bigmem	1
619	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._12587.job	INLRMS: Q	N/A	grid.uio.no	smallmem	1
620	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._13203.job	INLRMS: Q	N/A	grid.uio.no	smallmem	1
621	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._13344.job	INLRMS: Q	N/A	grid.uio.no	smallmem	1
622	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._13532.job	INLRMS: Q	N/A	grid.uio.no	smallmem	1
623	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._13770.job	INLRMS: Q	N/A	grid.uio.no	smallmem	1
624	noisolana_3l_trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037_31	INLRMS:R	22	sigrid.lunarc.lu.se	workq	1
625	noisolana_3l_trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037_32	INLRMS:R	24	sigrid.lunarc.lu.se	workq	1
626	noisolana_3l_trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037_36	INLRMS:R	22	sigrid.lunarc.lu.se	workq	1
627	noisolana_3l_trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037_37	INLRMS:R	21	sigrid.lunarc.lu.se	workq	1
628	noisolana_3l_trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037_40	FINISHED	17	sigrid.lunarc.lu.se	workq	1
629	noisolana_3l_trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037_41	INLRMS:R	20	sigrid.lunarc.lu.se	workq	1
630	noisolana_3l_trig1_misal1_mc12.005200.T1_McAtNlo_Jimmy.recon.AOD.v12000604_tid008037_43	INLRMS:R	21	sigrid.lunarc.lu.se	workq	1
631	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._12809.job	FINISHED	724	hagrid.it.uu.se	default	1
632	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012324._17857.job	FINISHED	670	hagrid.it.uu.se	default	1
633	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012323._08053.job	INLRMS:R	666	hagrid.it.uu.se	default	1
634	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012323._08068.job	INLRMS:R	666	hagrid.it.uu.se	default	1
635	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012323._08185.job	INLRMS:R	638	hagrid.it.uu.se	default	1
636	misal1_mc12.005568.ttbar_Pythia.digit.v12000701_tid012323._08210.job	INLRMS:R	663	hagrid.it.uu.se	default	1

http://www.nordugrid.org/monitor/quelist.php?host=sigrid.lunarc.lu.se&port=2135&qname=workq

Figure 5. NorduGrid monitor showing jobs belonging to a particular user.

on both the status of the Grid and down to the level of individual jobs. The set of running jobs belonging to a particular user is shown in Figure 5. Jobs 624 to 630 are physics analysis jobs running on a cluster in Sweden, and correspond to analysis of the dataset highlighted in Figure 2.

6. SUSY Searches on the Grid

In preparation for data taking with ATLAS, a large number of physics studies are being performed using highly detailed simulated data. These data, produced as discussed in Section 4 above, are presently being analyzed using the distributed analysis tools shown in the previous section.

As an example, Figure 6 shows preliminary results from a search for supersymmetric particles. Supersymmetry (SUSY) (see e.g. [9] for a recent review) is a hypothesis that states that for each known elementary particle there exists a heavier superpartner, i.e. that we have so far only found (at most) half the particles that exist in nature. SUSY can solve a number of known problems in high energy particle physics, but there is as yet no experimental evidence for it. If it exists, there is however a good chance that it can be discovered at the LHC over the coming years.

Particle collisions that carry information on SUSY will be very rare compared to other types. Making such a discovery therefore entails searching through a large number of events and looking for specific striking signatures. These signatures will have backgrounds coming from known standard model processes, so one needs to model both the expected signal and backgrounds. The left panel of Figure 6 shows a comparison of signal (red) and background (black) for one event topology in a specific supersymmetry model, using as a signature the Cambridge Mt2 variable [10, 11]. At a certain value of Mt2 the signal goes above the background, showing that a discovery is possible should this model prove to be correct. The right hand side of Figure 6 shows a scan of various supersymmetry parameters for 500 SUSY models, within the mSUGRA SUSY breaking scheme. The masses of three supersymmetric particles are shown (q_R,$\tilde{g}$,χ_0^1), and on the lower right is the cross section for the process that was under study in the left–hand plot ($qq \rightarrow \tilde{q}_R\tilde{q}_R$).

Both the SUSY model scan and the search analysis were performed using Grid infrastructure,

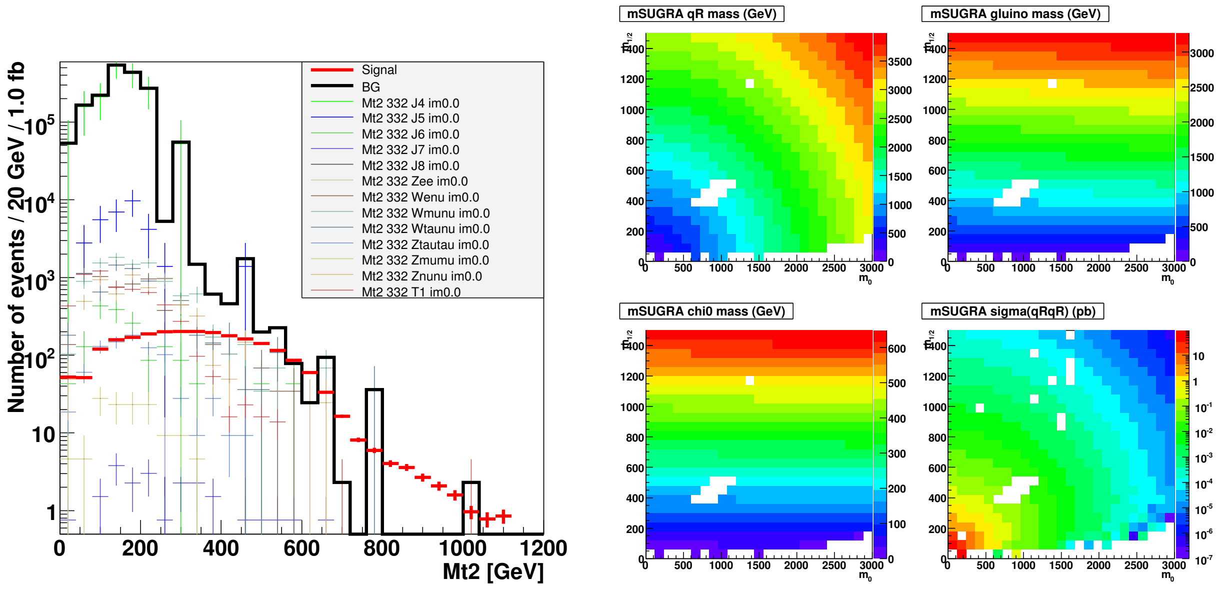


Figure 6. Searching for supersymmetry using grid infrastructure. Left: A signal–to–background comparison for a specific SUSY model. Right: Properties of various mSUGRA realisations of SUSY.

as well as data produced and distributed through the methods described above. The final steps were made with personalised user analysis code, submitted as jobs to NorduGrid defined through GANGA and monitored through the NorduGrid monitoring services.

7. Conclusion

In this paper we have shown how members of a single institute can contribute to all aspects of the chain leading to physics analysis and discovery, from collecting data from the detector or simulation production all the way to Grid based analysis of massive numbers of events. We have described how using ARC connected resources and a unique distributed storage solution enables NorduGrid to handle efficiently the large data requirements from the ATLAS experiment. At the time of writing the MC simulation production system has managed to utilise almost 1000 CPU days in a single day (the equivalent of keeping 1000 CPUs occupied over a whole day), and this number is expected to go up with the increase in resources available.

The lightweight nature of the ARC middleware leads to simple integration into tools such as GANGA, so that physicists require no Grid knowledge to use Grid resources. Real physics analysis is already being perfomed on NorduGrid, and some example results of these analyses have been presented. The use of Grid resources depends highly on the ease of use and transparency to the end user, whether or not they are a Grid expert, and we are confident that the Grid solutions shown here fit these requirements and will be commonly used by all physicists in the future.

References

[1] M Ellert *et al* 2007 *Future Generation Computer Systems* **23** 219–240
[2] A L Chervenak *et al* 2004 *Performance and Scalability of a Replica Location Service* Proceedings of HPDC 04
[3] G Behrmann *et al* 2007 *A Distributed Storage System with dCache* Proceedings of CHEP 2007 #148
[4] dCache `http://www.dcache.org`
[5] ATLAS Collaboration 2005 Atlas computing technical design report Tech. rep. CERN

[6] M Lassnig *et al* 2007 *Managing ATLAS data on a petabyte-scale with DQ2* Proceedings of CHEP 2007 #64
[7] G Behrmann *et al* 2007 *ATLAS DDM Integration in ARC* Proceedings of CHEP 2007 #149
[8] http://cern.ch/ganga
[9] L Pape and D Treille 2006 *Rept. Prog. Phys.* **69** 2843–3067
[10] C G Lester and D J Summers 1999 *Phys. Lett.* **B463** 99–103 (*Preprint* `hep-ph/9906349`)
[11] A Barr, C Lester and P Stephens 2003 *J. Phys.* **G29** 2343–2363 (*Preprint* `hep-ph/0304226`)