Available on CMS information server

CMS NOTE 2002/015

GriPhyN 2002-3

The Compact Muon Solenoid Experiment

CMS Note

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

March 15, 2002

CMS Grid Implementation Plan - 2002

C. Grandi[1], L.Bauerdick[2], R.Cavanough[3], P.Capiluppi[1], C.Charlot[4], I.Fisk[5],
K.Holtman[6], G.Graham[2], O.Kodolova[7], V.Lefebure[8], H. Newman[6]
[1]*Univ. of Bologna / INFN* , [2]*Fermilab* ,[3]*Univ. of Florida,* [4]*IN2P3-Lyon,*
[5]*Univ. of S.Diego,* [6]*Caltech,* [7]*SINP-Moscow,* [8]*HIP-Helsinki*

Abstract

This document describes the plans for integrating grid tools in CMS computing environment during 2002. The document includes: the description of the CMS software and computing environment; the analysis of the tools already produced by the Grid Projects; the summary of the tools already in use by CMS; the implementation plan for the existing tools and those foreseen by the Grid Projects within 2002, we consider useful to CMS. No effort is done here to build a *long-term* scenario of the CMS-Grid.

1 Introduction

The CMS collaboration has a long term need to perform large-scale simulation efforts, in which physics events are generated and their manifestations in the CMS detector are simulated. These simulation efforts support detector design and the design of the real-time event filtering algorithms that will be used when CMS is running. Furthermore they provide a way for designing the reconstruction and analysis frameworks needed to process large amounts of events that will be available when the detector will start collecting data. To date CMS Regional Centers, which are distributed in Europe and in the USA, have produced approximately 20 million simulated data events, which are being analyzed by CMS physicists. These simulation efforts will continue, and will grow in size, up to 2005 and then throughout the lifetime of the experiment. Given the huge amount of computing resources required and the discrete nature of event simulation, a distributed solution is favored.

CMS software is already highly functional but at the same time is still in rapid evolution. On the other hand experiment milestones impose strict time schedules for data production and analysis. For these reasons the integration in the CMS environment of grid tools and the adoption of the grid paradigm is a particularly challenging task. At the same time the grid projects hold extraordinary promise in providing functionality in the form of tools that are developed and supported by groups outside of the CMS collaboration that support distributed computing. Our goal in this document is then to provide a sense of the current functionality of these tools, to gain a sense of the most fruitful directions of future research done by the Grid groups, and provide a plan for tighter integration with future CMS software.

During 2001 several Grid Projects started in Europe, in the USA and in Asia. Albeit the base software used by all the projects is basically the same (e.g. Globus, Condor-G, ClassAds), the middleware developed by the different Grid Projects addresses different problematic. While it is a strong CMS request that the Grid Projects provide interoperable grid tools, the test of the tools will continue in parallel using the tools developed by the different projects and the integration will be done in the CMS-Grid framework whenever possible.

The plans for integration of grid tools in CMS environment in 2002 are discussed here. Some longer term CMS-Grid scenario is presented in [1].

2 CMS current Applications

The following is a brief description of the applications comprising the core infrastructure of CMS offline Monte Carlo production and analysis. This infrastructure has elements in common with other offline concerns as well, such as the CMS persistency solution and data storage techniques for example.

2.1 Generation of physics channels and simulation of the CMS detector

This step has two sub steps:

1. generation of physics events using CMKIN (currently based on PYTHIA, though other generators should be supported)
2. simulation of tracking in the detector using CMSIM (a GEANT3 based program)

CMKIN needs as input an ASCII file of data-cards describing the physics channel to be simulated (including a run number and the input random number generator seed) and writes out two files: a random access ZEBRA file with the 4-momenta of the generated events (*PYTHIA ntuple*) and the standard output file which includes the summary of the generation information (x-sections, etc…). Since this information is essential for the analysis of the produced data, it is saved persistently on a relational database (*RefDB*, see below) at production time.

CMSIM needs as input an ASCII file of data-cards describing the parameters to be used in the detector simulation, a few files describing the simulation conditions (e.g. CMS geometry, magnetic field, etc…) and the *PYTHIA ntuple* and writes a single formatted ZEBRA file with the simulated events (*FZ file*), a random access ZEBRA file with diagnostic histograms, and a log file and standard output.

Both CMSIM and CMKIN are FORTRAN programs which are linked statically and which don't need any special environment to run but the correct Operating System. No meta-data are recorded anywhere and thus the jobs are only loosely coupled: merging of the data of the same dataset produced by different jobs is always possible provided the run numbers and random number generator seeds are independent.

A typical simulation of 500 events takes from 8 to 24 hours on a 1GHz processor and writes a 0.6 to 1 GB file, depending on the physics channel, which is simulated.

In the future the CMSIM step will be done using OSCAR, a GEANT4 based C++ program, which will store data to an OO database directly (see next section).

2.2 Transfer of data to an OO database and digitization

Further steps of the simulation and reconstruction chain rely on an OO database for data storage. Thus the next operation is the translation of the information of the *FZ files* into an Objectivity/DB database, which is the current choice of CMS for persistency. This is done using ORCA/COBRA using as input an ASCII file of data cards (*.orcarc* file), the CMS geometry file and an *FZ file*. The output of a set of these jobs executed in a single location which read all the *FZ files* of a single dataset is a collection of tightly coupled Objectivity/DB databases which are seen by ORCA as a unique event collection (*hits* databases). Each *FZ file* corresponds to a different run number inside the *hits* database. Meta-data are stored in a set of databases, which are integral part of the dataset (but many datasets can use the same meta-data databases).

These jobs are I/O bound, being the CPU required negligible. The size of the *hits* databases is comparable with that of the *FZ files*.

The digitization step, which is also done using ORCA, reads events from two datasets: the *signal* dataset and the *pile-up* dataset. The *pile-up* dataset is a special collection of *hits* databases, which is exported to all Regional Centers. All the databases including the Meta-data ones of the *pile-up* dataset need to be registered in read-only mode to the local Objectivity/DB federation at the time the jobs run.

This process is also critical for the I/O, not only for the CPU. About 200 events from the *pile-up* dataset, 300 KB in size each on average, have to be randomly dispatched to a given processor digitizing an event from the *signal* dataset. Given the CPU time to perform the digitization of a complete piled-up event (i.e. of a signal event with all its pile-up events superimposed), 10 s on a 1 GHz PC processor, this implies about 6 MB/s of input data per *data crunching* processor, named *client* in the following. This limits to about 10 the number of clients to be efficiently served with the needed input data by a given data server. The output of the digitization step is a collection of tightly coupled Objectivity/DB databases, which are seen by ORCA as a unique event collection (*digi* databases). *Digi* databases contain links to events in the *hits* databases, which allow navigation to the data used to produce them.

A pre-requisite to perform this step, as well as any other operation which writes Objectivity/DB databases, is currently the need of a strong coordination among the sites participating to the effort. A pre-allocation of *database's ID*'s in a database *Federation*[1] has to be defined, assigning the ID numbers to the different sites in order to be able to combine all the objects produced into a single, consistently accessible set of uniform simulated data. It's a requirement that has to be applied to every database system, being the database either relational or object oriented. Furthermore a strong coordination is needed in the allocation of the *Owner name*, which is the mean by which the Meta-data of datasets being produced in different independent sites can be later on merged in the same Objectivity/DB federation.

ORCA is a dynamically linked program, thus the correct shared libraries must be available on the computing nodes and the correct environment has to be set as well as the correct OS and *glibc* versions.

2.3 Reconstruction and user analysis

2.3.1 Short-term scenario for user analysis

The output of the previous step is a collection of events that is of the same kind of the data that will be collected by the real CMS detector (*raw data*). The activities of reconstruction and analysis imply read access to the *hits* and *digi* databases and write access to new databases (*user collections*) or production of external files (e.g. ntuples, root files, etc…).

Albeit Objectivity/DB allows registering all databases to a unique federation (thus having all datasets available to a user job), this is not used in CMS because of data access efficiency. Input data location is done manually and the user jobs are submitted to the sites which have the data stored locally.

[1] Databases are organized in *Federations* within Objectivity/DB. The federation concept imposes some practical constrints; for example applications can only access data from one federation at a time and the federations themselves are limited in the number of internal files it can index, up to 64K. This index is called the *database ID*.

2.3.1 Long-term scenario for analysis environment

In order to better understand the future developments of CMS computing environment, the long-term scenario for user analysis is described here. Such analyses search massive numbers of events for signature patterns offering evidence for various proposed theories of nature. Once CMS is online and recording live data from the detector, analysis activities will be the dominant use of computing resources.

In a typical analysis, a physicist selects events from a large database consisting of small sized event classification information, or TAG data. Using the TAG data, the physicist gathers the full reconstructed event from various sources including, if necessary the creation of fully reconstructed versions of those events if they do not already exist on some "convenient" storage system. A typical analysis might look something like the following:

- Search through 10^9 TAG events (~100 GB), select 10^6 of those events and fetch the Event Summary Data (ESD) data for the selected events (~1 TB)
- Invoke a user defined reconstruction of the ESD data and make a user defined set of Analysis Object Data (~100 GB) and Tags (~100 MB)
- Analyze the user defined Analysis Object Data (AOD) and TAG datasets interactively, extracting a few hundred candidate signal events (~10 MB).
- Histogram the results and visualize some of the "interesting" events. We seek to create virtual data techniques to track data dependencies for the files and/or objects in this process from TAG schemas and TAG databases (or tables) back to the reconstructed event sets and possibly back to the raw data.

2.4 Monte Carlo production environment

The Monte Carlo production environment is distinguished mainly by the fact that it involves production processing on a large scale while at the same time minimizing the amount of direct human intervention. Therefore, a Monte Carlo production environment will need to have automated subsystems dealing with the following areas: input parameter management, robust and distributed request and production accounting, preparation of executables, local management of production resources, distributed management of production resources, local access to mass storage, and distributed file storage and replica management. In the long term, the production system must be capable of fault detection and tolerance. Monte Carlo production is therefore a leading candidate in the integration of grid tools since many of the problems being addressed by grid researchers address the above problems. The current tools are described below.

All Monte Carlo production requests are stored in a reference database (*RefDB*) at CERN. Each request has all the input parameters needed to create the data. The RefDB uses the World Wide Web (wget) to distribute parameters asynchronously to Regional Centers located worldwide. The RefDB also recieves accounting information asynchronously at the finish of each local production job via an HTTP GET query. In addition to the RefDB, accounting information is stored locally at each Regional Center in a variety of formats, but most importantly the production log files are kept. The tracking information that define the status of the request is joined and summarized at the RefDB.

The request is dispatched to Regional Centers by e-mail. A set of scripts (IMPALA) has been developed to automate production job creation and submission for the different steps of the production chain in a Regional Center, but this process is also initiated by hand. IMPALA also monitors job execution and in some cases is able to re-create and re-submit crashed jobs. IMPALA uses BOSS as interface to local batch facilities and also supports legacy direct interfaces to local batch facilities. BOSS is a system that is able to do bookkeeping of the relevant information produced by the different types of jobs synchronously with job execution. BOSS does this by wrapping each production batch job with a process that monitors standard output and standard error in real time. BOSS also is able to register application level tracking schema with each job type and update this information in real time or at least synchronously with the end of the job. BOSS uses MySQL database as a backend. When configured propoerly, the real scheduler is hidden to IMPALA by BOSS. The summary of job tracking done in a RC by IMPALA is sent back to the *RefDB*. Distributed management of compute resources is currently done by hand by the overall CMS production coordinator. Access to local Mass Storage facilities is typically provided by the IT divisions of hosting institutions, such as Enstore at FNAL or CASTOR at CERN. Distributed access to mass storage has yet to be worked out. File replication is typically achieved with off the shelf tools such as *scp* and *bbcp*. Finally, none of the systems in place is particularly fault tolerant at this stage and requires significant expenditures of manpower to make it all work.

3 Grid tools currently used by CMS

GDMP (Grid Data Mirroring Project) [2] was first developed in the framework of CMS to enable easy transfer of the Objectivity/DB databases produced in the different Regional Centers. GDMP version 1.2 has been used for data transfer by most of the CMS Regional Centers in 2001. GDMP has since been extended to be able to handle flat files as well as Objectivity files.

Job submission to remote resources using Condor-G has been tried in several sites. In particular a small production has been completely done using Condor-G in Caltech, UCSD and Wisconsin Univ. Condor-G is a gateway to resources managed by Globus GRAM. The use of Condor-G presupposes that Globus is installed and working at the regional centers and that the Globus gateway is functioning with correct and up-to-date certificates.

4 Grid tools available now and foreseen in 2002

4.1 Areas of integration of grid middleware

The main areas where currently available grid tools can help CMS computing environment are:

- Regional Centers management
  - Installation
  - Configuration
  - Monitoring
- Distribution of Monte Carlo production, reconstruction and other *scheduled* tasks:
  - Submission of jobs to grid resources
    - Fault tolerance
    - Increase efficiency
  - Transfer of produced files to final destinations
  - Replica of files to Regional Centers
- Analysis of distributed data
  - Automatic location of input files
  - Automatic allocation of resources
  - Transfer of output files to the user
  - Remote Analysis Servers

4.2 Software

The grid tools available to date or that will be available in the first part of 2002 are:

- Virtual Data Toolkit 1.0 from GriPhyN
- Release 1 from EU-DataGrid

4.3 Security

Access to distributed resources is done basically using GSI authentication. The following Certification Authorities are already available and inter-operating:

- CERN CA
- Czech Republic – CESNET CA
- France - CNRS Datagrid-fr CA
- Ireland - Grid-Ireland CA
- Italy – INFN CA

- Netherlands – NIKHEF CA
- Nordic countries – NorduGrid CA
- Portugal – LIP CA
- Russia - Russian DataGRID CA
- Spanish - DATAGRID-ES CA
- United Kingdom – GridPP CA

In addition:

- Globus CA

is available. In April 2002 is also foreseen:

- Esnet CA

which will become the default in the USA.

4.4 Platforms

Most of the grid middleware has been made available for Linux platforms. CMS integration activities will be done using Linux.

Even though CMS software is proved to be working properly with any 2.2.X linux kernel, the only versions officially supported by CMS at the time being are Red Hat 6.1 (standard distribution, with 2.2.12-20 kernel) and Red Hat 6.1.1 (CERN distribution, with 2.2.19-6 kernel). These two platforms, as well as RH 6.2, are widely used by CMS Regional Centers (including CERN). Porting of the grid middleware to RH 6.1, 6.1.1 and 6.2 is mandatory for inclusion in the official CMS production system.

4.5 Software installation tools

The set up of the correct software environment on each node that will run CMS code is mandatory. Two systems are currently available to standardize software installation:

- LCFG developed by EU-Datagrid is based on **`rpm`**
- DAR developed by FNAL is based on *tar-balls*

4.6 Data management tools

Both VDT 1.0 and DataGrid release 1.1.1 toolkits use GDMP 2.1 as file manager. GDMP 2 introduces the notion of a Replica Catalog. The Replica Catalog maps logical filenames to one or more physical file locations. While GDMP initially implemented its own replica catalogue, this functionality may be taken into the Globus project at some later date. GDMP contains within it the capability to transfer files between sites. The semantic of using GDMP involve "publishing" the file and its physical location to a central replica manager under some logical filename. The recieving site then "pulls" the published file. It is an open question whether this semantics satisfies all CMS requirements. GDMP 3, which will be ready by the end of march 2002, will include the API for a *Replica Manager* and will be adopted both by VDT and DataGrid.

In the longer term the use of Virtual Data can help data production and analysis in CMS. The extension of the concept of the Replica Catalog to a Virtual Data Product Catalog (VDPC) goes in this direction. A prototype of a VDPC has been developed by the GriPhyN project which tracks the dependencies of data files and transformation between data files. As such, it is able to regenerate any (missing or deleted) derived data file on demand. Integration with GDMP and its extension is foreseen during 2002.

4.7 Job submission tools

Albeit submission to the grid is done using Globus and Condor-G both in the VDT and in the DataGrid toolkits, the two projects are today addressing different aspects of the problem.

In the DataGrid release 1 toolkit the development of a Resource Broker service interfaced with an Information Index and with a Replica Catalog addresses the problematic of automatic resource allocation (see figure 1)[3]. The consultation of the Replica Catalog when doing the choice of the resource where to submit the job optimizes the usage of the network. The current interface allows easy file transfer from the user to the execution node and *vice versa* as well as translation of file names from logical to physical in a rather transparent way.

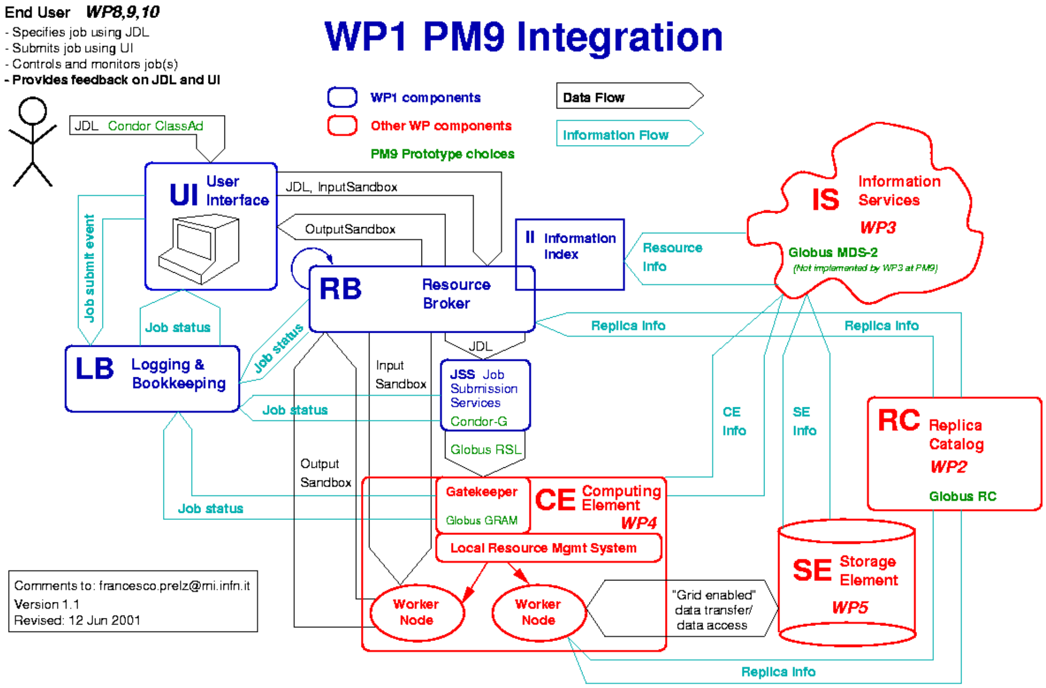

Figure 1: EDG Grid Scheduler Architecture

MOP is a tool developed by the PPDG group that couples IMPALA with Condor-G and GDMP thus allowing production job submission to the grid and automatic transfer of produced data to the final destination. Furthermore MOP uses DAGMan (included in VDT) to manage the dependencies of the production jobs performing the different steps of simulation. The inclusion of BOSS in MOP is being addressed now. MOP is a very thin packacging layer that essentially wraps jobs in both DAGMAN language to describe dependencies and in Condor description files to describe the atomic parts of the jobs. It also has some knowledge of the internals of IMPALA of how to transfer input files and output log files to/from the execution site. Data files are transferred using GDMP.

Future developments include the use of DAGMan and of a job decomposition mechanism in the DataGrid toolkit and the use of Virtual Data Catalogs in MOP.

Extending the Virtual Data approach, GriPhyN/PPDG will build active components, the *planners*, which will build DAG’s using IMPALA and the parameters from the production requests (see figure 2). In this approach the functionalities of the *RefDB* are incorporated in the VDPC. Any data request (either an official production or a user analysis job) triggers the production of any data on which it relies if the data are not available yet. More details can be found in [4].

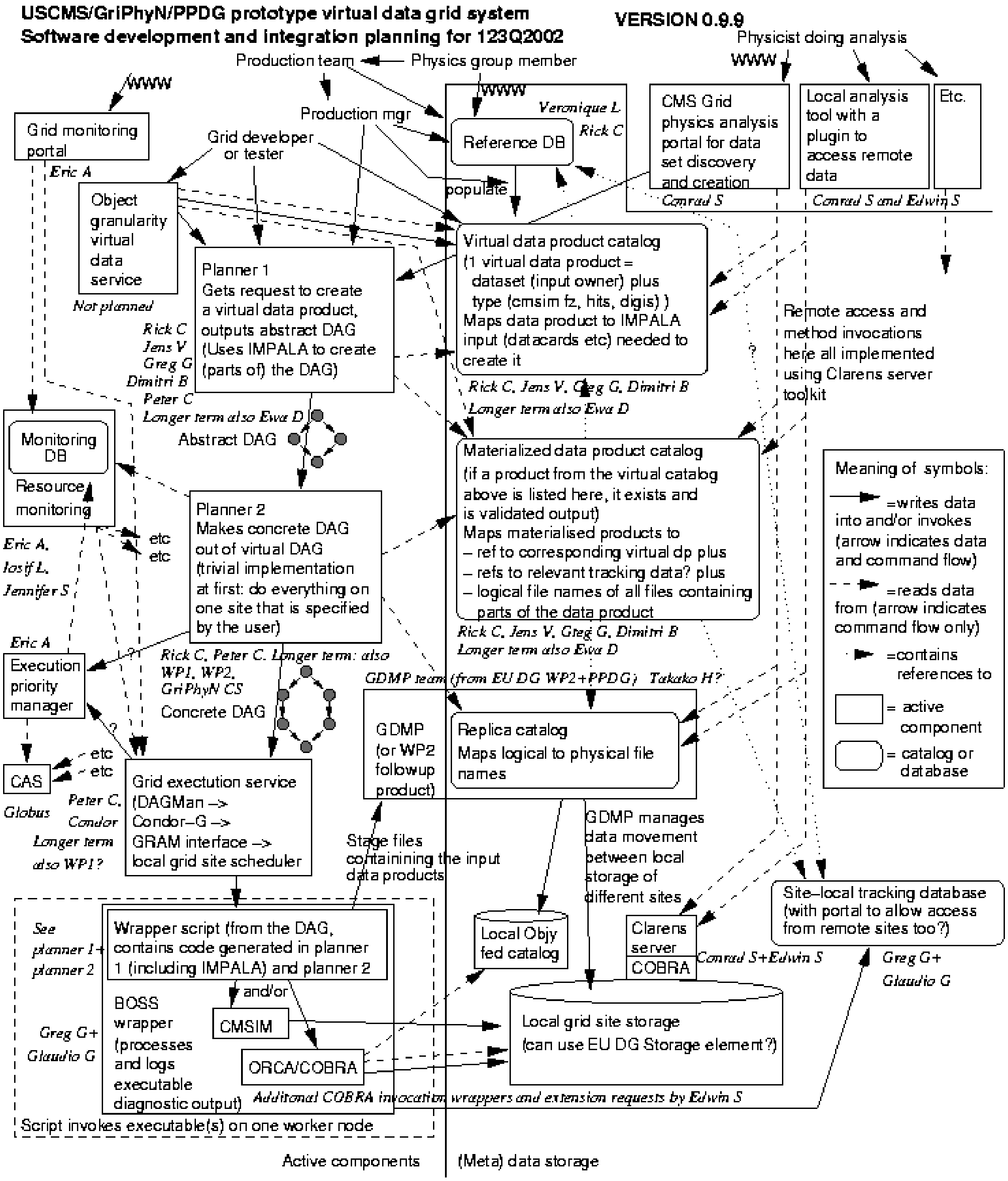
USCMS/GriPhyN/PPDG prototype virtual data grid system
Software development and integration planning for 123Q2002
VERSION 0.9.9
Physicist doing analysis
WWW
Production team
Physics group member
WWW
Grid monitoring portal
Production mgr
Grid developer or tester
WWW
Veronique L
Rick C
Reference DB
populate
CMS Grid physics analysis portal for data set discovery and creation
Conrad S
Local analysis tool with a plugin to access remote data
Conrad S and Edwin S
Etc.
Eric A
Object granularity virtual data service
Not planned
Planner 1
Gets request to create a virtual data product, outputs abstract DAG (Uses IMPALA to create (parts of) the DAG)
Rick C
Jens V
Greg G
Dimitri B
Peter C
Longer term also Ewa D
Virtual data product catalog
(1 virtual data product = dataset (input owner) plus type (cmsim fz, hits, digis))
Maps data product to IMPALA input (datacards etc) needed to create it
Rick C, Jens V, Greg G, Dimitri B
Longer term also Ewa D
Remote access and method invocations here all implemented using Clarens server toolkit
?
Monitoring DB
Resource monitoring
Abstract DAG
Materialized data product catalog
(if a product from the virtual catalog above is listed here, it exists and is validated output)
Maps materialised products to
– ref to corresponding virtual dp plus
– refs to relevant tracking data? plus
– logical file names of all files containing parts of the data product
Rick C, Jens V, Greg G, Dimitri B
Longer term also Ewa D
Meaning of symbols:
=writes data into and/or invokes (arrow indicates data and command flow)
=reads data from (arrow indicates command flow only)
=contains references to
= active component
= catalog or database
Planner 2
Makes concrete DAG out of virtual DAG (trivial implementation at first: do everything on one site that is specified by the user)
Eric A,
Iosif L,
Jennifer S
etc
etc
Eric A
Execution priority manager
?
Rick C, Peter C. Longer term: also WP1, WP2, GriPhyN CS
Concrete DAG
GDMP team (from EU DG WP2+PPDG) Takako H?
GDMP (or WP2 followup product)
Replica catalog
Maps logical to physical file names
CAS
Globus
etc
etc
?
Peter C,
Condor
Longer term also WP1?
Grid exectution service
(DAGMan –>
Condor–G –>
GRAM interface –>
local grid site scheduler
Stage files containining the input data products
GDMP manages data movement between local storage of different sites
Site–local tracking database (with portal to allow access from remote sites too?)
See planner 1+ planner 2
Wrapper script (from the DAG, contains code generated in planner 1 (including IMPALA) and planner 2
Local Objy fed catalog
Clarens server
COBRA
Conrad S+Edwin S
Greg G+
Glaudio G
and/or
BOSS wrapper (processes and logs executable diagnostic output)
CMSIM
ORCA/COBRA
Greg G+
Glaudio G
Local grid site storage
(can use EU DG Storage element?)
Additional COBRA invocation wrappers and extension requests by Edwin S
Script invokes executable(s) on one worker node
Active components
(Meta) data storage

Figure 2: GriPhyN/PPDG software development and integration plan using Virtual Data approach

4.8 Interactive analysis tools

A remote data server (Clarens) is being developed by GriPhyN to enable analysis of CMS data distributed over a Wide Area Network. Clarens is based on a Client/Server approach and provides a framework for remote data analysis. Communication between the client and server is conducted via XML-RPC over a scalable SSL-encrypted HTTP transport provided by the Apache web server. Currently the server is only linked to the standard CMS analysis C++ libraries, but various server functions are envisioned. The client can be end-user specific and several implementations already exist (e.g. C++, Python, JAS, PHP, etc…). It is foreseen to interface Clarens to VDT (in particular using GSI authentication) and to interface it to the virtual data catalog [4].

The EU-DataGrid scheduler will be extended in 2002 to support interactive analysis.

5 Grid Integration Plans for 2002

Since the official grid-integration activities just started in CMS, most of the activities reported here are more testing plans rather than a real integration plan: any decisions on integration of grid tools into official CMS software environment will depend on the results of these tests.

This section is divided in four sub-sections according to the structure of the CMS Grid-Integration Task [5 and modifications].

5.1 Architecture sub-task

In the short and medium term, the CMS framework (COBRA) will not be responsible for interacting with the grid services directly. Any access to distributed data will be responsibility of the underlying database service. Long-term solutions will depend on the baseline choice for persistency in CMS. In particular if the final database service will be Oracle or a similar database, the file concept will disappear. Data location and replication will be done at collection level, or even at object level. In this case it will be responsibility of the CMS framework, i.e. COBRA, to interface with the grid services (see [1]). Long-term solutions are not discussed here.

5.2 Regional Centers sub-task

Since CMS goal is to exploit distributed resources regardless the country where they're located, one important step is to identify the components needed on machines configured by the different Grid Projects to be visible by tools developed by the other Grid Projects. In this task we expect a strong support by the Grid Projects themselves and by LCG.

Another important task is to allow easy installation, configuration and monitoring of computing resources at CMS Regional Centers. DAR is today used for CMS software installation by most of the RC. LCFG in principle can take care of the installation and maintenance of all software (including the OS) but doesn't offer today enough functionalities and a large amount of manual work is required. The possibility to use LCFG for installation of packages on pre-installed farms will be investigated soon at INFN and in Russian centers. In release 2 (September 2002) the system will be more robust and complete and it will be investigated for full farm installation and maintenance.

The support to CMS test-bed activities is another activity included in this sub-task.

5.3 Productions sub-task

Data production and analysis are targeted to CMS detector design, so the integration of any new tool in the "production" environment must be driven more by efficiency consideration than on innovation. On the other hand, the use of new tools in real "production" activities can boost their development bringing an increase of efficiency at the end. For these reasons it is foreseen that some limited number of Regional Centers will try to do an early deployment of some of the new tools in their "production" environment.

It must be stressed that CMS production environment has been developed in order to grant high efficiency and quality control of produced data. It will be hard to reach such a performance using grid tools available today and in the short term. Nevertheless we believe that it is worth to build a prototype based on grid tools in order to identify components that can increase automation (and decrease the manpower needed to drive the production itself). Furthermore adopting the grid paradigm in the production framework that is already highly structured, will help us in building an efficient analysis framework that is where the use of the grid can really help in hiding to the final user details on data and resource location.

Our goal is to use GDMP 2 for data transfer between Regional Centers during "spring 2002" productions. Since a stand-alone installation kit is not provided by GDMP authors, such a kit will be prepared inside CMS as soon as the needed middleware will be considered stable by the authors (March 2002 ?).

The EU-DataGrid scheduler will be interfaced to BOSS allowing easy coupling of IMPALA with the EU-DataGrid scheduler (February 2002). A test production distributing jobs to remote sites will be tested in a few pilot sites in Europe. It needs the set-up of the services (II, RB, LB, JSS, see figure 1) at CERN (April 2002).

At the time being the production requests are assigned by the production manager to the Regional Centers manually (typically by e-mail). The possibility to use the EU-DataGrid scheduler to dispatch production requests to the RC's will also be investigated (June 2002). This will also automate the update of the *RefDB* with the summary of the production.

MOP will be used for limited CMS productions in sites in the USA participating in the USCMS Grid Testbed during spring 2002 (February 2002).

Virtual Data Catalog (see figure 2) will be integrated with a prototype of a grid-enabled IMPALA (march 2002). Materialized Data Catalog prototype will be available in April 2002. Planner 1 and 2 will be ready in March 2002. A test production using a VDPC with the functionalities of the *RefDB* will be tested (May 2002). Possible integration in the CMS production environment will be evaluated at the end of the “spring 2002” production (June 2002).

As said in the previous sub-section, interoperability between tools developed by the US and European Grid Projects is essential. Albeit development and testing of tools continues independently on the two sides of the Atlantic to increase efficiency, the goal is to build a unique CMS grid environment. In the following we discuss the scenario where the Virtual Data Grid System is integrated with EU-DataGrid middleware.

At the time being the EU-DataGrid scheduler doesn’t manage DAG’s. This functionality is expected with EDG release 2 (September 2002). CMS will try to have an early prototype of the EDG software to test the possibility to use the EU-DataGrid scheduler with part of the functionalities of *Planner 2*, *Execution Priority Manager* and *Grid Execution Service* (see figure 2). In figure 3 we report the structure of a prototype to be built in the fall 2002, where EDG Scheduler and *Replica Manager* are *Virtual Data*-enabled.

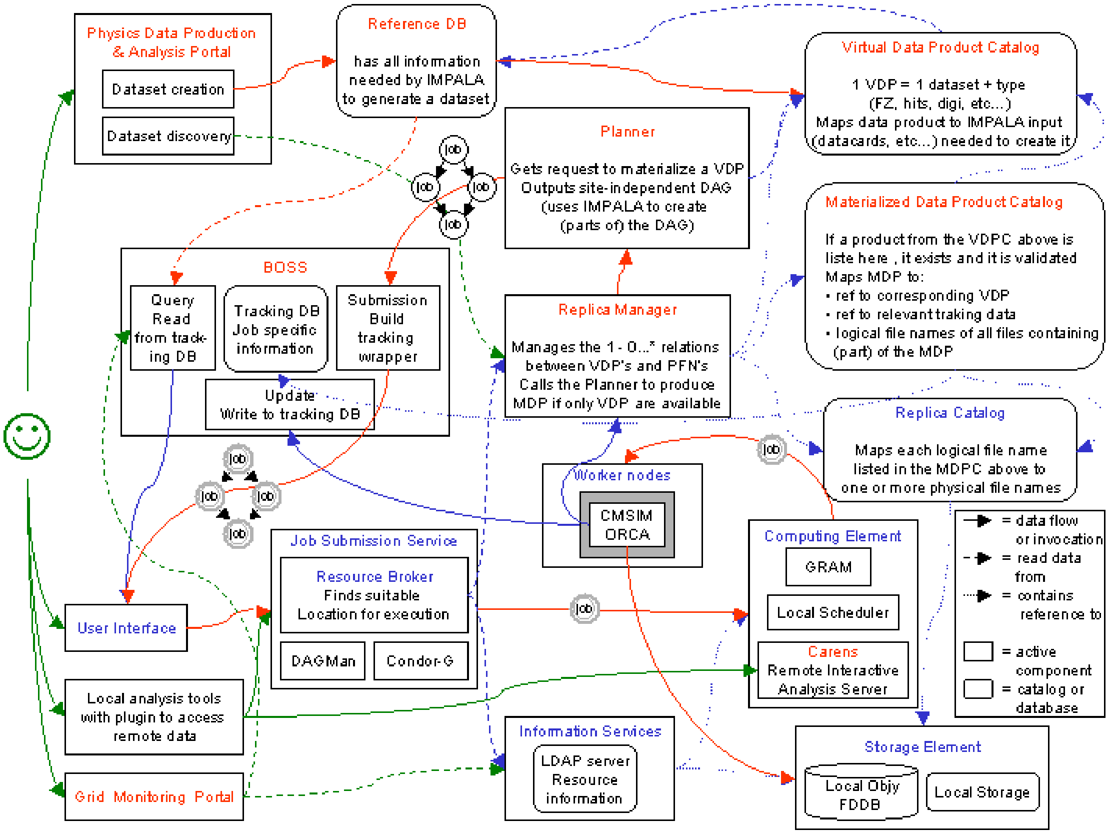
Physics Data Production & Analysis Portal
Dataset creation
Dataset discovery
Reference DB
has all information needed by IMPALA to generate a dataset
Virtual Data Product Catalog
1 VDP = 1 dataset + type (FZ, hits, digi, etc...) Maps data product to IMPALA input (datacards, etc...) needed to create it
Planner
Gets request to materialize a VDP Outputs site-independent DAG (uses IMPALA to create (parts of) the DAG)
Materialized Data Product Catalog
If a product from the VDPC above is liste here , it exists and it is validated
Maps MDP to:
• ref to corresponding VDP
• ref to relevant traking data
• logical file names of all files containing (part) of the MDP
BOSS
Query Read from tracking DB
Tracking DB Job specific information
Submission Build tracking wrapper
Update Write to tracking DB
Replica Manager
Manages the 1 - 0...* relations between VDP's and PFN's Calls the Planner to produce MDP if only VDP are available
Job
Worker nodes
CMSIM ORCA
Replica Catalog
Maps each logical file name listed in the MDPC above to one or more physical file names
Job Submission Service
Resource Broker Finds suitable Location for execution
DAGMan
Condor-G
Computing Element
GRAM
Local Scheduler
Carens Remote Interactive Analysis Server
User Interface
Local analysis tools with plugin to access remote data
Grid Monitoring Portal
Information Services
LDAP server Resource information
Storage Element
Local Objy FDDB
Local Storage
= data flow or invocation
= read data from
= contains reference to
= active component
= catalog or database

Figure 3: Integrated grid environment for CMS

A production request corresponds to an entry in the VDPC. Data are always accessed through a *Virtual Data*-enabled *Replica Manager*. When a data product is requested (regardless how it is requested) and it is not present in the MDPC but is listed in the VDPC, the RM invokes the *Planner*. The Planner checks that the request is coming from a user that is authorized to trigger the production of a dataset. The Planner uses the information in the VDPC and *IMPALA* to create a DAG. The DAG is location-independent, since all of the site related parameters (I/O, etc…) are managed by the *Resource Broker* using the information provided by the *Replica Catalog* and the *Information Services*. The DAG is then submitted to *BOSS*, which builds a wrapper around each node of the DAG to allow runtime job tracking. BOSS submits the job to the RB using the EDG *User Interface*. Each job is sent by the *Job Submission Service*, which also manages the priorities, to a *Computing Element* through the Globus GRAM interface. Running jobs write tracking information to the BOSS database and output data to a suitable *Storage Element* (determined by the RB). The output data are notified to the RM that updates the RC and the MDPC. In all steps user identification is done through its certificate.

5.4 User analysis sub-task

As anticipated in previous sections, massive user analysis is just starting now. Several approaches are being investigated to grant effective access to CMS data and computing resources to the end-users in the short term. It is essential that the interface to access the resource is easy-to-understand and that at the same time the access to the resources is highly controlled. In the longer term data analysis will benefit from the new architecture developed for data production.

The construction of a prototype analysis farm is foreseen at INFN for the analysis of the muon and b-τ groups (February 2002). Interactive access to the farm will not be granted to the users allowing easy management of the cluster. CPU and storage resources will be allocated to user jobs using the DataGrid scheduler. Input data will either be served by a close storage element, or made available by replicating data from the grid. Extension to the use of more than one farm will be straightforward since the scheduler is hiding the concrete resource being used (June 2002).

A *Clarens* prototype will be tested using it to analyze remotely data produced using MOP (march 2002). An analysis portal will be ready in summer 2002; client analysis tools with a plug-in to analyze remote data will be ready before the end of 2002. In the longer term, Clarens will be integrated with the environment showed in figure 3. When the EDG Scheduler will support interactive jobs can be thought as an interactive analysis server configured on a Computing Element. Access to the CE is gained using the Resource Broker. The CE is chosen by the RB in order to minimize the resources needed to read data from the Storage Element where the input data are.

5.5 Comments on the plan

We are aware of the fact that some of the components used in the definition of the prototype shown in figure 3 are not, at the time being, robust enough to fulfill CMS needs not only in the long term but also for managing today's production process. In particular a single Replica Manager (and related catalogs) will represent single-point-of-failure of the system.

References

[1] **CMS Note 2001/037**, K. Holtman et al., *"CMS Data Grid System. Overview and Requirements"*.

[2] http://cmsdoc.cern.ch/cms/grid/

[3] **DataGrid-01-D1.2-0112-0-3**, *"Definition of architecture, technical plan and evaluation criteria for scheduling, resource management, security and job description"*.

[4] http://kholtman.home.cern.ch/kholtman/planv1/plan.html

[5] http://cmsdoc.cern.ch/~grandic/CCSPlenary011108.ppt

Appendix: CMS InterGrid test-bed plans

Inter-Regional Datagrid Testbeds
Information requested by HIJTB

1. Associated experiment (if any)

- CMS

2. Participating sites (if some are involved at the start and others later, indicate times)

- CERN
- Caltech
- Fermilab
- Univ of Calf., San Diego
- Univ. of Florida
- Univ. of Wisc., Madison
- INFN Bologna/CNAF
- INFN Legnaro/Padova
- IN2P3 Lyon
- PPARC RAL
- SINP
- ITEP
- JINR

Start times of the sites have still do be discussed.

3. Principal goals of test (What constitutes success?)

Some of the specified goals are to be considered as long term expectations, while others are short or medium term tests activities.

- Distribution of Monte Carlo production:
  - Submission of jobs to GRID resources
  - Transfer of produced files to "final" destinations
  - Replica of files to Regional Centers
- Analysis of distributed data
  - Automatic location of input files
  - Automatic allocation of resources
  - Transfer of output files to the user
- Probe Certificate Authority issues

Other possible goals include integration the US-CMS Test Grid with the

EU-CMS Test Grid.
The success is the demonstration of the interoperability of the "certified" grid tools developed in the US and in the EU.

4. What applications will be tested?

- CMS core software (COBRA)
- CMS applications software (CMSIM, ORCA)
- Production framework (IMPALA)
- Analysis-specific distributed accesses applications

5. What resources are needed and how will they be provided? (Transatlantic bandwidth, processors, storage,....)

The tests will mostly use grid projects dedicated resources.
Some transatlantic data transfers will be done among Regional Centers participating to official CMS productions. Sustained transfer rate will be of order 100 Mbps, which is of the same order of magnitude of CMS production rate. Some spikes will be possible due to occasional activity. In any case official CMS productions for CMS PRS groups takes priority over testbed plans. Hardware or manpower may not be diverted from production to take part in these tests until the production team have met their commitments.

6. What assistance should be provided by the InterGrid committees?

Focus the critical issues to be tested and provide coordination on common resource requests (e.g. Transatlantic bandwidth).
Provide a common repository of reports on functionality of tested tools offered by DataGrid, GriPhyN/PPDG, and other grid projects.
Provide support for Grid Projects interoperability.

7. Provide major milestones for the project. When will work start? When will active testing start? Other milestones.

Separate tests by DataGrid and GriPhyN/PPDG already started. Common tests can be planned to start in June.
What follows is a tentative schedule. It depends on the time when stable middleware will be available.
Inter-project tests (end June) depend on the support we will get from middleware developers and DataTag on interoperability issues.

- February: US sites have the VDT 1.0, Condor-G 6.3.1, and Objectivity 6.1 installed.
- February: EU sites have DataGrid middleware installed.
- Spring 2002: run the PPDG MOP and the GriPhyN Virtual Data Catalog demonstrations (as exhibited at Supercomputing 2001) on the Test Grid.
- March: First data transfer with GDMP 2.1
- End March: Replica of files to Regional Centers with GDMP
- from April: migration to the ESNet Certificate Authority of US sites
- June: Make some US resources available to EDG Resource Broker
  Make some EU resources available to MOP+VDPC
- End June: Submissions to the EU+US grid with EDG Scheduler:
  automatic location of input files (GDMP)
  automatic allocation of resources (Broker)
  transfer of output files to the user
  Submission of jobs with MOP to the EU+US grid
- End 2002: Start testing the prototypes of the Virtual Data Grid System integrated with the EDG Scheduler

8. Manpower

a) Project Coordinator(s)

Still to be decided. The current coordinators are:

- C. Grandi <Claudio.Grandi@bo.infn.it> (CMS)
- P. Capiluppi <Paolo.Capiluppi@bo.infn.it> (EU)
- J. Amundson <amundson@fnal.gov> and
  R. Cavanaugh <cavanaug@phys.ufl.edu> (US)

b) Principal contacts: at least one in each region involved

- It has to be understood what kind of roles will have the Principal contacts for the regions involved. The choice of the responsible person(s) has to be done on the basis of the planned activities to better match the goals of the Project. Preliminary names are:

- CERN	(?)	
- Caltech	Julian Bunn	<julian@cacr.caltech.edu>
- Fermilab	Jim Amundson	<amundson@fnal.gov>
- Univ. of Calf., San Diego	Ian Fisk	<ifisk@ucsd.edu>
- Univ. of Florida	Rick Cavanaugh	<cavanaug@phys.ufl.edu>
- Univ. of Wisc., Madison	Peter Couvares	<pfc@cs.wisc.edu>
- INFN Bologna/CNAF interim	Claudio Grandi	<Claudio.Grandi@bo.infn.it>

- INFN Legnaro/Padova	interim	Claudio Grandi <Claudio.Grandi@bo.infn.it>
- IN2P3 Lyon		Claude Charlot <charlot@poly.in2p3.fr>
- PPARC RAL		(?)
- SINP	interim	Olga Kodolova <Olga.Kodolova@cern.ch>
- ITEP	interim	Olga Kodolova <Olga.Kodolova@cern.ch>
- JINR	interim	Olga Kodolova <Olga.Kodolova@cern.ch>

c) Estimated effort required. How will it be provided? Are names available now?

- No estimate is available now. However for the grid interoperability only we can envisage 2-3 FTE's.

9. What Grid middleware will be tested?

a) From DataGrid

- LCFG (Farm configuration)
- authentication/authorization system (common with PPDG/GriPhyN)
- Grid Scheduler
- GDMP (common with PPDG/GriPhyN)

b) From PPDG/GriPhyN

- authentication/authorization system (common with EDG)
- Virtual Data Toolkit 1.0:

Condor	6.3.1	
Condor-G	6.3.1	
DAGMan	6.3.1	
ClassAds	0.9	
Globus	2.0 beta	
GDMP	2.1	(common with EDG)

c) Other

- Condor-G 6.3.1