Available on CMS information server **CMS CR 2001/006**

CMS Conference Report

HEPGRID2001: A Model of a Virtual Data Grid Application

Koen Holtman

Published in Proc. of HPCN Europe 2001, Amsterdam, p. 711-720, Springer LNCS 2110. (c) Springer-Verlag LNCS.

HEPGRID2001: A Model of a Virtual Data Grid Application

Koen Holtman

California Institute of Technology,
Mail code 256-48, 1200 E. California Blvd. Pasadena, CA 91125, USA.
koen@hep.caltech.edu

Abstract. Future high energy physics experiments will require huge distributed computational infrastructures, called data grids, to satisfy their data processing and analysis needs. This paper records the current understanding of the demands that will be put on a data grid around 2006, by the hundreds of physicists working with data from the CMS experiment. The current understanding is recorded by defining a model of this CMS physics analysis application running on a 'virtual data grid' as proposed by the GriPhyN project. The complete model consists of a hardware model, a data model, and an application workload model. The main utility of the HEPGRID2001 model is that it encodes high energy physics (HEP) application domain knowledge and makes it available in a form that is understandable for the CS community, so that architectural and performance requirements for data grid middleware components can be derived.

©Springer-Verlag. To be published in Proc. of HPCN Europe 2001.

1 Introduction

In several areas of science, the growth in scale of the data generation and processing activities out-paces Moore's law. One example is the CMS high energy physics (HEP) experiment at the CERN laboratory [1] which plans to generate 1000 TB of raw data per year from 2006 on, with an estimated 10,000 CPUs all over the world being used around the clock in data analysis activities [2]. In the past, large physics experiments often developed their data processing facilities using in-house knowledge. To cope with the increase in scale and complexity, several experiments have recently joined up with computer science groups in research projects like GriPhyN [3] and the European DataGrid [4], which are tasked with researching and developing the necessary future software infrastructure. For these projects to be successful, it is now necessary that the experimentalists communicate their requirements in the language of computer science, rather than internally using language of their own scientific domain. The model in this paper was created as part of this communication effort to computer scientists.

The HEPGRID2001 model reflects the current understanding of the demands that will be put on a the CMS data grid system around 2006, by the hundreds of physicists working with data from the CMS experiment. The main reference

source used is [2], which records the state of knowledge about the quantitative aspects of the requirements in early 2001. The model is composed of three parts. The first part is a model of the peta-scale distributed grid hardware configuration expected to be available to CMS in 2006. The second part is a CMS data model using the concept of virtual data as proposed by the GriPhyN project [3]. The third part is a workload model, which is complex enough to capture the essential challenges and opportunities faced by the virtual data grid catalogs, schedulers and optimizers.

The 'HEP' in the name 'HEPGRID2001' is the common abbreviation for high energy physics, the number 2001 signifies that it encodes the current understanding of a future application, an understanding that will evolve over time.

2 Hardware model

The CMS experiment is run by a 'virtual organization', the CMS collaboration, in which over 140 institutions world-wide participate. The funding and manpower constraints involved have the result that the CMS hardware will be a world-wide distributed system, not a centralized one. The hardware consists of a central site called the 'tier 0' center, 5 regional sites called 'tier 1' centers, and 25 more local sites called 'tier 2' centers. The tier 0 center is located at the CERN laboratory, the location of the CMS experiment detector. The tier 1 centers each have a 2.5 Gbit/s network link to the tier 0. Each tier 1 center has 5 tier 2 centers connected to it. Tier 1-2 connections are also 2.5 Gbit/s network links. It should be stressed that the actual link capacity available to CMS in 2006 cannot be estimated very accurately, mainly because of uncertainties about long-term developments in the international telecom market. In any case, it is expected that the effective throughput for grid application level data transport will be only about half of the raw link capacity. This is due to protocol overheads and some other traffic on the same link (interactive sessions, videoconferencing), but also because the need for reasonable round trip times (reasonably short router queues) implies that the links cannot be saturated to full capacity.

The individual center characteristics are as follows:

For each single center at	CPU capacity	Nr of CPUs	Storage space
Tier 0	600,000 SI95	3000	2300 TB
Tier 1	150,000 SI95	750	900 TB
Tier 2	30,000 SI95	150	70 TB

The above numbers were taken from [2], they reflect estimates for 2006. The capacity of a single CPU used in 2006 is estimated to be 200 SI95.

3 Data model

Figure 1 shows the HEPGRID2001 data model, which is defined in terms of a data flow arrangement inside the grid system.

Any piece of data in the grid is called a 'data product' in this model. A data product is the smallest piece of data that the grid needs to handle. Figure 1

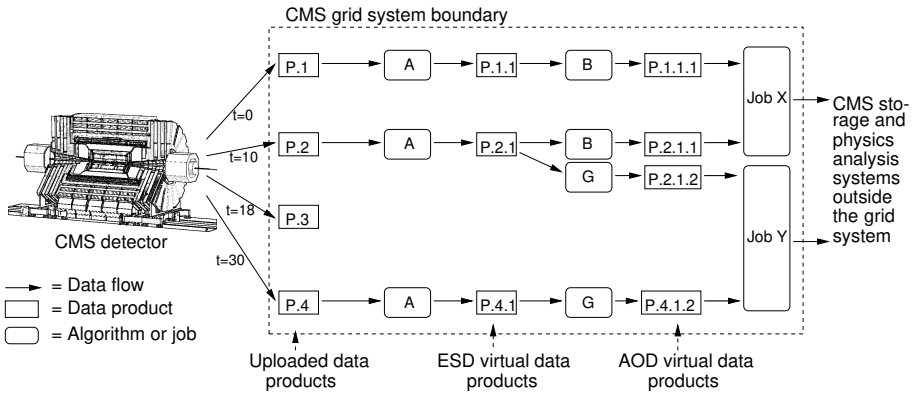

Fig. 1. HEPGRID2001 Data model, in terms a data flow inside the CMS grid

shows how products (the boxes with square corners) are generated and used. There are two types of data product: uploaded data products and virtual data products.

Each *uploaded data product* contains the CMS detector output for a single *event*. An event is defined here as a set of particle collisions occurring simultaneously inside the CMS detector. Events will occur at a rate of 40 Mhz inside the detector: a real-time filter selects only some 100 events per second for storage, for uploading into the grid. All uploads happen at CERN, the location at the tier 0 site. Given the event size and the duty cycle of the detector, this yields 1000 TB of uploaded data products per year.

A *virtual data product* is the output of some data processing algorithm that has been registered with the grid. By having the algorithm registered with the grid, the virtual data product value can be computed on demand. The algorithm will use the value of another (virtual or uploaded) data product as its input. Therefore, each uploaded product $P.1, P.2, \ldots$ is the top node of a tree of virtual data products which can be derived from it by applying algorithms (A, B, G). These tree structures are modeled in more detail further below. Each product has a unique identifier (UID), for example $P.1, P.4.1.2$. From the UID of a virtual data product, the grid can determine which algorithms and inputs are needed to compute (derive) the product value.

The data flow in figure 1 is a somewhat simplified representation of the several processing steps that physicists will perform to extract information from the detector measurements. The timing of these steps, and the feedback loop that occurs while making them, is discussed in section 4. The virtual data products obtained directly from the uploaded (raw) data products are generally called ESD (event summary data) products by CMS physicists, those obtained from ESD products are generally called AOD (analysis object data) products. Arrangements of algorithms more complicated than this 2-stage chain are also possible, but will occur less often and are not accounted for in this model.

The central idea that underlies virtual data products is that they can be defined first, by registering algorithms with the grid, and then need only be computed when needed as the input for a job (for example job X in figure 1). Management tasks related to intermediate results are thus offloaded from the grid users onto the grid itself. This makes the users more productive, and also gives the grid complete freedom in using advanced scheduling techniques to optimize the (pre)computation, storage, and replication of data product values. The ability of the grid to always re-compute a virtual data product value can also be used to achieve higher levels of fault tolerance. The virtual data idea is not new, for example spreadsheets also embody this concept. The major innovation of the virtual data grid work of the GriPhyN project is the scale at which these services will be provided.

The uploaded and virtual data products (figure 1) are modeled more formally as being the nodes in 10^9 trees. Each tree has, as its top node, one uploaded data product with the UID $P.e$, for $1 \leq e \leq 10^9$. The e here is an event identifier, a number which uniquely identifies the corresponding event in the detector. The branching structure of the trees reflects the use of different (improved) versions of derivation algorithms over time. Algorithm development is iterative: the next version is developed based on experience using the current version. The branching structure is as follows. Below the top node in each tree are 5 ESD virtual data products, these get UIDs $P.e.x$ with $1 \leq x \leq 5$. Below each ESD there are 20 AOD virtual data products, these get UIDs $P.e.x.y$ with $1 \leq y \leq 20$. Each virtual data product can only be derived by applying some specific algorithm to the value of the product that is its parent in its tree.

Product characteristics are as follows:

CMS		Level		CPU power needed to		CPU time needed to	
name	Type	in tree	Size	derive	analyze	derive	analyze
RAW	uploaded	1	1 MB	–	3000 SI95s	–	15 s
ESD	virtual	2	500 KB	3000 SI95s	25 SI95s	15 s	0.125 s
AOD	virtual	3	10 KB	25 SI95s	10 SI95s	0.125 s	0.05 s

Here, the 'time to derive' is the runtime of the algorithm that computes the product value, and the 'time to analyze' is the time spent in a job to analyze the product value. The numbers and terminology were taken from [2].

The structure, size, and final destination of the job output in figure 1 are not captured in this model. They are not captured both because the output is currently less well understood, and because optimizations on the output side are considered less crucial to the successful operation of the physics grid system. Additional work on modeling the job output will likely be done in the next few years. In general the job output is smaller, often significantly smaller, than the job input. Some jobs compute one single output value (with a size below a few KB) based on their input, others will, for each event in the request set, compute and output a derived physics data (DPD) structure with a size as large as 10 KB per event. It is therefore more important to run (sub)jobs close to their input datasets, than to run them close to the destination of their output. Moving both the input datasets and the (sub)jobs close to the destination is expected to be an

interesting optimization, but not one that is crucial to the successful operation of the grid.

4 Workload model

Multi-user physics analysis workloads have a complicated structure. They can be modeled at various levels of detail: the level of detail for HEPGRID2001 is very high, in order to capture the essential challenges and opportunities faced by the virtual data grid catalogs, schedulers and optimizers. This work takes a two-step approach to modeling the workload. As the first step, this paper gives a high-level overview of the shape of the workload, and discusses the factors that determine its shape. For reasons of style and space however, the high-level overview does not contain all statistical and morphological details that would be needed by researchers who want to use this workload model in simulations. Therefore, as the second step, a workload generator has been defined that can be used in such simulations. The workload generator encodes some additional domain knowledge needed to generate a properly stochastic workload. The generator, available at [6], computes a one-year grid workload, containing 124695 jobs and 12565 'hints', following the tree structures defined below. The output of the workload generator is an ASCII file with each line describing a single job or hint.

4.1 Job model

Physicists get work done from the CMS virtual data grid by submitting jobs to it, see figure 1. A HEPGRID2001 job definition consists of two things: the *job request set*, which is a set of data product UIDs, and the *job code*, which is a parallel program that can be executed by the grid. The grid needs to execute the job code, and deliver the values of all data products in the request set to this code for further analysis. The job code will use grid services to deliver its output to the user. An example of a job request set (for job Y in figure 1) is the set with the product UIDs $P.2.1.2$ and $P.4.1.2$, this set can also be written as

$$\bigcup_{e \in \{2,4\}} P.e.1.2$$

In the HEPGRID2001 model, job request sets always have the general form

$$\bigcup_{e \in E} P.e.X$$

where E is a set of event identifiers and X is a (possibly empty) sequence of integers.

The job code is a parallel program, that is run as a set of subjobs. In this set there will be several 'worker' subjobs (the number to be decided by the grid schedulers) and one 'aggregation' subjob. Communication between subjobs is very minimal: at the end of its run every worker subjob uses the grid services to

send a single, relatively small, package of information to the aggregation subjob. The aggregation subjob creates the final job output and sends it outside the grid. To execute the job, the grid schedulers may partition the job request set in any way, and feed the different parts to as many worker subjobs that may run in multiple locations. The products in the request set may be delivered to the worker subjobs in any order. This allows for massive parallelism in virtual data product derivation and job execution. The grid also has complete freedom in choosing when to create (derive) and delete virtual data product values, and in replicating and migrating product values over the grid sites.

4.2 Properties of physics analysis workloads

The properties of the grid workloads produced by physics analysis are determined by three major interacting factors: the methodology of high energy physics as an experimental science, the way in which physicists collaborate and divide their work, and the need to maximize the utility of the available computing resources.

The goal in a physics experiment is to observe new physics phenomena, or observe phenomena with new levels of accuracy, in the particle collisions occurring inside the detector. The physics of two colliding particles is highly stochastic. The collision creates a highly localized concentration of energy, in which new particles may be created, with different rates of probability. Most 'interesting' particles will be created with extremely low probabilities: for example in the CMS experiment the (so far only theoretically predicted) creation of a Higgs boson force carrier particle might occur only once in 10^{12} collisions. The most resource and time-consuming task in physics analysis is therefore to recognize and isolate, from all events, only those events with a collision in which a sought-after phenomenon occurred. To decide whether an event e is interesting, whether it fits the sought-after phenomenon, the uploaded (raw) data product of event e is run through a chain of feature extraction algorithms. Examples of features that are extracted are the tracks (trajectories) of any photons and electrons emanating from the collision point). Then, several 'cut predicates' are applied to the extracted features of event e. The cut predicates select for the sought-after phenomenon, only the events which satisfy all cut predicates are left as 'interesting'. An example of a cut predicate is `n_elec==2`, which is an abbreviation for 'the observed number of electrons produced by the collisions in the event, and emanating from the collision point, is 2'. Due to the stochastic nature of collision physics, the probability that an event satisfies a set of cut predicates is uncorrelated with the time at which the event occurred.

Physics analysis, the development of feature extraction algorithms and cut predicates, is an iterative process, in which subsequent versions are refined until their effects are well-understood. The grid jobs run during this process can be compared to the compile-and-run steps in iterative software development. The grid job 'locate the Higgs events and calculate the Higgs mass from them' is highly atypical: it is the final job at the end of a long analysis effort. A much more typical job is 'run this next version of the system I am developing to

locate the Higgs events, and create a plot of these parameters that I will use to determine the properties of this version'.

In an analysis effort to isolate a particular phenomenon, the cut predicates are generally developed one after each other. Each cut predicate is developed and refined by using it inside grid jobs and studying the output of these jobs. The request sets of these jobs always consist of one data product for each of the events that satisfy all cut predicates so developed so far. The cut predicates are developed by individual physicists, with the exception of a first 'group' level cut predicate that defines the 'channel' that a group of physicists is interested in.

The feature extraction algorithms that produce the virtual data products of section 3 are not written by individual physicists, but by groups of specialists. In the HEPGRID2001 workload model, a new ESD derivation algorithm (and associated detector calibration constants) is released 5 times per year, a new AOD derivation algorithm 75 times per year.

4.3 Workload details

The 137260 jobs and hints in the workload, as created by the workload generator, are arranged as the nodes in five trees called the 'workload trees'. Each tree is 5 levels deep. Only the leaf nodes at level 5 represent actual physics analysis jobs. All non-leaf nodes represent 'hints' for the grid scheduler. Physicists can submit these hints to the scheduler to help it anticipate and optimize the future workload. Each hint node encodes a prediction about the job request sets of the jobs at the leaf nodes below. This prediction takes the form of a job request set over a set of events ES, where it is guatanteed that this ES a superset of the sets of events in all job request sets below. In current practice, physicists supply similar hints to their computing system operators and management boards, who use them to allocate resources and to perform 'production efforts', in which large sets of data products are pre-computed and stored for later use.

The tree properties are summarized by the following table, then discussed in more detail further below.

Level	Type	Fan-out	Interpretation of this (sub)tree in CMS
1	hint	5 trees	Each represents the use of a different ESD
2	hint	20 subtrees	Each represents the actions of a physics group
3	hint	25 subtrees	Each represents an analysis effort of a physicist
4	hint	3-5 subtrees	Each represents a phase in an analysis effort
5	job	5-20 leaf nodes	Each represents a job in an analysis effort

The jobs and hints are not submitted to the grid all at once, but over a period of about a year, following a 'job sequence' order. This job sequence roughly sweeps from left to right through the workload trees as illustrated in the leftmost plot of figure 2.

At level 1, the highest level of the workload model, each of the five trees in the workload corresponds to the joint use by all physicists of new ESD derivation algorithm. A new ESD derivation algorithm (and associated detector calibration

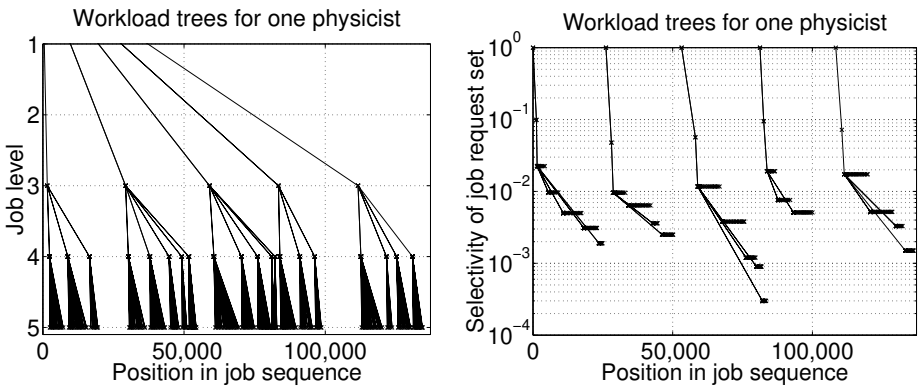

Fig. 2. Partial visualizations of the workload trees, as created by the workload generator [6]. Only the jobs and hints for a single physicist in a single group are shown. Each job or hint is plotted with a small 'X', these are connected by lines to show the tree structure. The left hand plot shows level and position of in the job sequence. The right hand plot shows exactly the same trees, but this time the y axis shows the selectivity (fraction of all events included) of the job and hint request sets.

constants) is released 5 times per year. The release of new AOD derivation algorithms is not reflected directly in the workload trees, but modeled in a different way in the workload generator output.

At level 2 of the trees, the workload model reflects that CMS physicists will organize themselves into 20 groups. Each group will decide on a 'first' cut predicate, with a selectivity from 4%–15%, that represents a very rough selection of the events that the group is interested in. In every group subtree the hint and job request sets are always over sets of events that satisfy at least the first group cut predicate. This group coordination, and the associated level 2 hints, provide important resource saving opportunities for the grid.

At level 3 of the trees, the model reflects that each group has 25 physicists in it, each physicists will perform an independent analysis effort on the events selected by the first group cut predicate. At level 3 every physicist develops a second, private cut predicate with a selectivity of 20%–25%, this predicate is combined with the group predicate to select the events considered in the analysis effort. The level 3 hint notifies the grid of this sub-selection.

At level 4, the different iterative phases in the activity of a physicist are modeled: each subtree represents a phase. A single phase models the development of a single new cut predicate. Going from one phase to the next, the physicist adds the newly developed predicate to the set used for the subsequent hints and jobs, increasing the selectivity with 30%–70%.

At level 5, the leaf nodes represent the actual physics analysis jobs run by the physicists. Each set of jobs under a single level 4 parent represents the iterative development of a single predicate. All these jobs will share the same request

set. Job request sets generally contain AOD products. In the later phases of a physicist effort however, when the event set left is relatively small, the physicist is more likely to select larger products (ESD or even RAW products) for the job request sets. This reflects both the decreased runtime penalty of using larger products, and the increased need to use larger products because the information present in the smaller products has been exhausted already, as a means for event selection, by previous cut predicates.

The different groups and physicists all work independently and in parallel: this means that the level 3 physicist subtrees in any workload tree will overlap in time. Each individual physicist has a sequential think-submit-wait cycle. Therefore, in the job sequence, the hints and jobs of each physicist subtree appear in strict tree traversal order.

4.4 Workload statistics

The following table gives some statistical properties of the HEPGRID2001 workload.

Time span covered by workload in the model	1 year
Number of physicists submitting jobs	500
Number of jobs	341/day
Average size of a job request set	10^7 products
Average size of a job request set	1.3 TB
CPU capacity needed to analyze requested products in jobs	960,000 SI95
RAW products requested by all jobs	$4.5*10^9$/year
ESD products requested by all jobs	$3.0*10^{11}$/year
AOD products requested by all jobs	$9.4*10^{11}$/year
Average number of times that a single product is requested	40
CPU capacity needed to derive all requested products once	433,000 SI95
Different virtual data products defined	105/event/year
Different virtual data products derived at least once	31/event/year
ESD products derived if all derived only once	$4.3*10^9$/year
AOD products derived if all derived only once	$2.7*10^{10}$/year
Size of RAW products	1000 TB/year
Size of ESD products derived if all derived only once	2166 TB/year
Size of AOD products derived if all derived only once	269 TB/year

5 Conclusions

This paper records the current understanding of the demands that will be put on a virtual data grid around 2006, by the hundreds of physicists working with data from the CMS experiment. Related work on the modeling and simulation of CMS computing has been done in the MONARC project [5]. In comparison to this work, the MONARC models generally contain more hardware details and less workload details. Also, in stead of hints, the MONARC workloads have explicit 'production jobs' submitted by production managers, jobs that compute

and store large sets of virtual data product values for further analysis. The GriPhyN [3] and European DataGrid [4] projects are currently both going through application requirements gathering cycles, and this work is part of that effort. An important contribution of this work is that it encodes resolutions to many detailed modeling issues, based on domain knowledge in CMS, resolutions that are needed to do simulations with realistic workloads. Encoding this domain knowledge is a necessary step towards collaborating more closely with modern computer science.

The CMS data grid has many requirements in common with other grid applications: for example security and sharing policies for creating a virtual organization, fault tolerance, and the handling of differences between hardware platforms. This paper focuses on those requirements that might be unique to high energy physics: the scale of the problem, the structure of the virtual data products, and the nature of the workload. It is not known currently how unique these high energy physics requirements are. From the standpoint of the CMS experiment, it would be preferable if commonalities and new abstractions could be found that show that the requirements are less unique than thought, so that there can be greater sharing with grid related software development occurring in other efforts.

Acknowledgements

Most quantitative elements of the hardware and data models are due to the MONARC project [5] and the recent LHC computing review efforts in the CMS collaboration [2]. Thanks go to Paolo Capiluppi, Ian Foster, Irwin Gaines, Iosif Legrand, Harvey Newman, and Kurt Stockinger for their comments and feedback in creating the HEPGRID2001 model.

References

1. CMS Computing Technical Proposal. CERN/LHCC 96-45, CMS collaboration, 19 December 1996. See also: http://cmsinfo.cern.ch/Welcome.html
2. S. Bethke et al. Report of the steering group of the LHC computing review. CERN/LHCC/2001-004, CERN/RRB-D 2001-3, 22 February 2001. Available from http://lhc-computing-review-public.web.cern.ch/lhc-computing-review-public/
3. http://www.griphyn.org/
4. http://www.eu-datagrid.org/
5. http://monarc.web.cern.ch/MONARC/
6. http://kholtman.home.cern.ch/kholtman/hepgrid2001/