Вісник Харківського національного університету
Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»
№ 703, 2005, с.74-86

УДК 004.7; 539.12

LCG Middleware at the KIPT CMS Linux Cluster

M. V. Voronko, S. S. Zub, L. G. Levchuk, D. V. Soroka
NSC Kharkov Institute of Physics and Technology, Ukraine

Problems associated with storage, processing and analysis of huge data samples expected in experiments planned at the Large Hadron Collider (LHC) are discussed. Current status of the KIPT CMS Linux Cluster (KCLC), which is a part of the Moscow distributed regional center for the LHC data analysis, is outlined. Configuration of the LHC computing Grid middleware at the KCLC is described. Participation of the KCLC in the CMS Monte-Carlo event production is outlined.

1. Introduction

The revolutionary development of computing facilities and networks that the society has witnessed for the last decade opens new opportunities for solving long-standing problems in science. This progress has already boosted [1] research activities in molecular biology (molecular modeling for drug design), neuroscience (brain activity analysis), geosciences (earthquake simulations and predictions), theoretical physics (non-perturbative treatment of strong interactions based on the lattice quantum chromodynamics (QCD)), and other fields on frontiers of the contemporary science. The complexity of these problems predetermines the needs of huge computing resources to be involved in the investigations and calls for a broad cooperation of scientists on an international or even global scale. Furthermore, such cooperation implies not only joint intellectual efforts aimed at solution of a scientific problem itself, but also establishment of principally new concepts of computation with possibility to allocate world-widely distributed resources to an application job. The latter aspect is addressed by the Grid technology, which was put forward in late ninetiess (see, e.g., Ref. [2] and references therein) and emerges in the beginning of the new century as a novel universal computational environment enabling integration of processors and storage elements of systems belonging to diverse organizations in widespread locations.

One of the major challenges that promote fast development and implementation of the Grid infrastructures comes from the high-energy physics (HEP), where experimentalists encounter exorbitant amounts of data which have to be securely stored, quickly processed and thoroughly analyzed. Nowadays, the bulk of HEP data is provided by experiments performed on particle accelerators at FNAL, SLAC and BNL (USA), DESY (Germany), and KEK (Japan). However, the situation is going to be dramatically harder in case of experiments to be carried on at the Large Hadron Collider (LHC), world's largest accelerator, which will be put into operation at CERN (Switzerland) in 2007.

The LHC nominal luminosity of $10^{34}\text{cm}^{-2}\text{s}^{-1}$ corresponds to 10^9 proton-proton collisions per second. Four detectors installed at the beam crossing points will register particles produced in these interactions. In case of the Compact Muon Solenoid

(CMS) detector [3], about 10^{-7} of the total event flow will be selected by a multi-level trigger for the off-line event processing and analysis. Thus, the data should be archived in a high performance storage system (SS) with the rate of ~100 Hz. Since the size of one CMS event written to the SS is supposed to be ~1 Mbyte, about 100 Mbyte of the information per second (or more than 1 Pbyte annually) has to be transferred to the SS.

The CMS physics program includes search for the Higgs boson, supersymmetry partners to known elementary particles and rare decays of B-mesons. A discovery of signals manifesting an evidence for such "new physics" requires an immense amount of data to be processed and thoroughly analyzed. Typically, the integral luminosity of $\sim 10^5$ pb^{-1} is needed in order to separate those manifestations against a huge background (see, e.g., Ref. [3]). It means that 10^9 CMS events (or 10^{15} bytes of information) have to be processed and analyzed. All this sets hard requirements upon data acquisition and storage systems, terms of computing and networking.

Furthermore, in preparation of the experiment, a lot of computational work is needed in order to optimize the trigger system and provide its physical contents, i.e., justify selection of given data samples and rejection of others. This activity implies "complete" simulation of the experiment starting from event generators (Monte-Carlo tools realizing existing theoretical models for description of high-energy particle interactions), through detailed simulation of the detector response, to the final stage which is event reconstruction and analysis. Certainly, to accomplish this work, considerable computing power is required.

Major challenges for the scientific community involved in the LHC experiments are supposed to be met via communication and collaboration at a distance, network-distributed computing and software development, remote data resource management and physics analysis. It suggests development of a Grid infrastructure, which would take into account mentioned above peculiarities and purposes of these experiments. The work on this project known as LHC Computing Grid (LCG) [4] is currently under way. Within this project, a complex multi-tier architecture of regional centers (RC's) is created [5] for data storage, processing and analysis (see Fig. 1). Interaction of processes running in different RC's is supposed to occur through the network, what is provided by a sort of connectivity software called middleware. Installation and proper configuration of the LCG middleware at a RC is not a trivial task, since this software is a prototype version subjected to permanent modification, rather than a completed product. Then, there are no universal recipes of tuning the Grid services, which would take into account the variety of system environments. Instead, one has to go on in the configuration process step by step, in order to avoid possible system conflicts. Moreover, participation of remote systems is needed to assure that the LCG middleware is correctly installed and configured at a given RC.

In this paper, we outline the procedure of LCG middleware configuration at the KIPT CMS Linux Cluster (KCLC). The KCLC is a specialized PC farm devised for conducting activities on Monte-Carlo simulations within the CMS physics program and CMS data analysis. The first stage of the KCLC (see Ref. [6]), which now enters the Moscow distributed RC (see Fig. 1), was constructed in 2001. Information about its current performances is given in the next section. Configuration of the LCG middleware elements at the KCLC is described in Sect. 3. Some use of these elements

is concerned in Sect. 4, where our CMS simulation activity is briefly described. Sect. 5 contains concluding remarks.

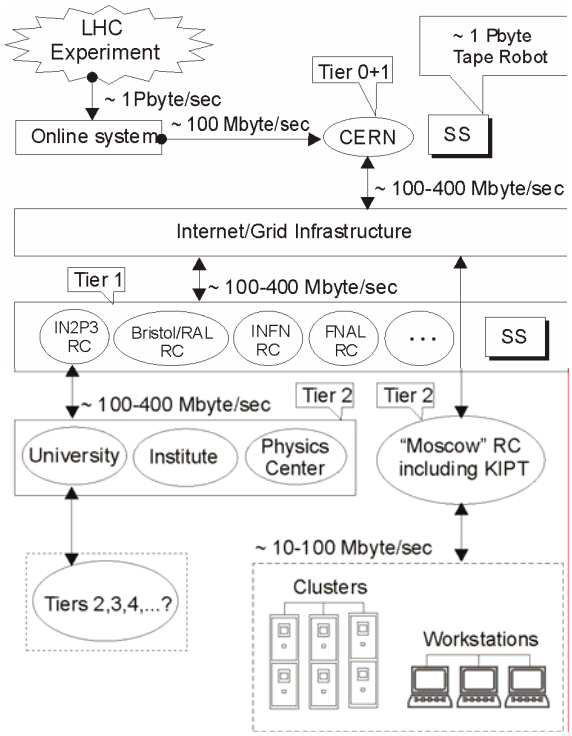

Fig.1. Hierarchy of Grid structures for LHC experiments.

2. KIPT CMS Linux Cluster

The KCLC (see Fig. 2) is a specialized PC farm for computing within the CMS physics program. Eleven dual nodes, viz., 4x800 MHz, 4x1000 MHz, 10x1400 MHz Pentium III, 2x2.0 GHz Xeon and 2x2.4 GHz Xeon processors are allocated for interactive or batch jobs providing the total PC farm CPU power of ~38 Gflops. The overall hard disk drive (HDD) storage is 2.5 Tbyte including two (0.4 Tbyte and 0.9 Tbyte) RAID arrays. System RAM is 9 Gbyte.

The nodes run Linux as the operation system (OS) and the Portable Batch System (PBS) (see, e.g., Ref. [7]) as a batch system. Other important components of our cluster are the network file system (NFS) and network information service (NIS). They provide the joint access to such resources as CERNLIB (including PYTHIA and GEANT), ROOT, PYTHIA, GEANT and LHC++ and to specific CMS software such as CMSIM (a GEANT-based package for simulation of the CMS detector response), ORCA (an object oriented tool for CMS event reconstruction). Versions of the programs are permanently refreshed according to CMS collaboration current demands.

The PBS is used as the cluster batch job and system resource management package. It accepts (see details in Ref. [7]) a batch job (a shell script with some control

attributes) preserves and protects the job until running, runs the job and delivers output to the submitter. The PBS allows one to administer flexibly the system resources while carrying on the computing and may be configured to support jobs run on a single system, or many systems grouped together. It can load processors of the cluster nodes in an optimal way (in accordance with an administrator policy) and select, e.g., the highest-priority execution jobs.

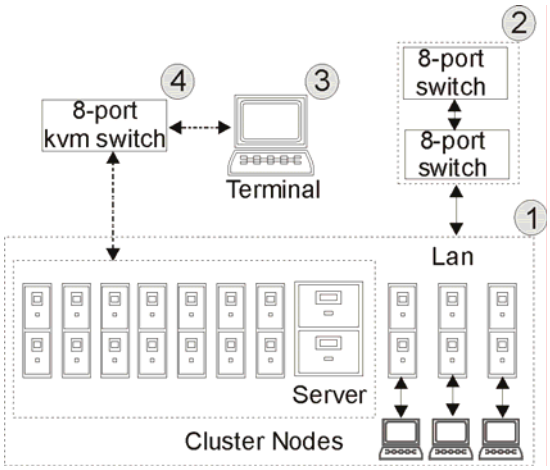

Fig.2. KCLC layout: 1 – server and nodes; 2 – two ethernet switches; 3 – terminal; 4 – kvm switch.

The configuration of the PBS at the NSC KIPT CMS Linux cluster is presented in Fig. 3.

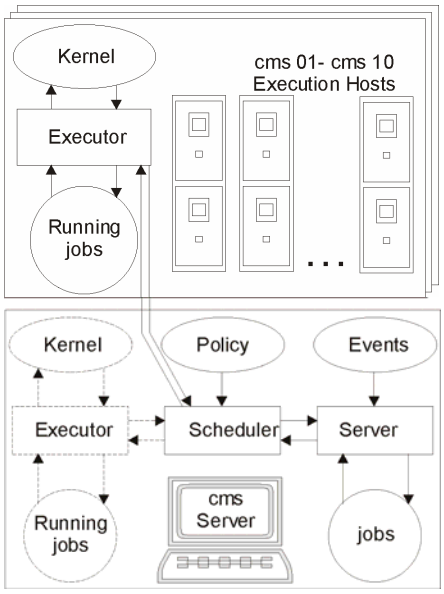

Fig.3. Use of PBS at the KCLC.

The batch system consists (see Ref. [7]) of a command shell and three daemons: the job server, the job scheduler and the job executor, with the latter being activated on every host allocated for execution. The commands are used to submit, monitor, modify and delete jobs and are available at each of the 11 nodes of the cluster. They communicate through the network with the job server. The server main function is to provide proper processing of the "events", i.e., such services as receiving/creating a batch job, modifying the job, protecting the job against system crashes and placing the job into execution. The job scheduler is a daemon, which contains a "policy" controlling which job has to be chosen for execution, and where and when it has to be submitted. The scheduler communicates with the server to get information about the availability of jobs to execute. To learn about the state of system resources, it addresses the job executors. (The daemon-to-daemon interface occurs via the network.) The job executor is the daemon, which actually places the job into execution. It also takes the responsibility for returning the job output to the user. Once a new job to be executed is found by the scheduler, and free resources are available in the system, the job is submitted to an execution host least loaded at the moment as estimated by the batch system. At present, the maximum number of non-parallel jobs executed on the cluster simultaneously is 22. The dual Intel Pentium III 800 MHz computer, performs the server tasks and does not participate in batch executions by default, though can be allocated to a batch job by a special request. If there are no free nodes (i.e., all 22 execution processors are busy), new submitted jobs are put (depending on computing resources requested) into one of 5 queues. When a free processor becomes available, it is immediately allocated to a job from the queue corresponding to the least amount of requested resources.

3. Grid Middleware at the KCLC

As mentioned above, the LCG middleware closely interacts with the system environment. For that reason, to avoid possible system conflicts, we performed a manual (step-by-step) LCG software deployment. Such a procedure helped us to provide a proper interface between LCG services and packages needed for CMS computational tasks such as, e.g., CMS Monte-Carlo event production (see Sect. 4). On the other hand, it allowed us to overcome difficulties associated with low external network bandwidth.

Of course, any Grid middleware is assumed to perform a variety of functions, and, in general, there is no single completed set of packages that would provide all these tools. This is typical of systems subjected to fast development and modification. However, there exist a number of common packages, which are included in the majority of Grid-based software. The LCG middleware discussed here uses the Globus tool kit [8] and the most part of the services developed within the European Data Grid (EDG) project [9].

These packages provide the work of main services, *viz*., GridFTP, GateKeeper, Globus Job Manager, monitoring and discovery service (MDS), locallogger, Broker, MyProxy, etc. The main set of packages is supplemented by software, which is specific for a given collaboration (CMS, ATLAS, ALICE or LHCb) participating in the LCG project.

Generally speaking, Grid components can be either single computers or clusters. Cluster option is supported by the LCG, in particular, via inclusion of the LCG-

specific interface to the Open PBS [7] called *lcgpbs* in the middleware. Accordingly, earlier PBS configuration described in Ref. [6] has been altered at the KCLC by the *lcgpbs* wrapper in order to provide a due operation of the LCG services.

Furthermore, a system in the LCG may include nodes having different functions. In case of the KCLC, we have nodes of the Computing Element (CE), Worker Node (WN), Storage Element (SE), and User Interface (UI) types.

The CE is based on 5 main services supported by the Globus package [8]. At the top level, the Globus provides services for organization of metacomputer infrastructure: Adaptive Wide Area Resource Environment (AWARE), version MPICH of the Message Passing Interface (MPI), Constructing Virtual Environment (CAVE), Common Object Request Broker Architecture (CORBA), etc. Principles and libraries of earlier systems, such as the Parallel Virtual Machine (PVM), the "plane" MPI, Condor, and their expansions, have been put into the basis of the package. The detailed description of the concept and realization of parallel programming with MPICH methods is outlined in the package documentation and in Ref. [10].

The Globus service that provides access to a CE from the Grid is called *gatekeeper*. It is the "frontend" of the CE. Usually, Gatekeeper is installed on a computer with a real IP address. It is this node that accepts jobs from an LCG user, distributes them over the WN's of a cluster and returns the output to the user. In case of the KCLC (see Fig. 4), we have one CE (node 'cms') and 9 WN's installed on all nodes of the farm, except for the SE ('cms05'). As shown on Fig. 4, the WN services can be configured also on a node running the CE. At the KCLC, the WN installed on the CE node can be allocated to a job through a special request.

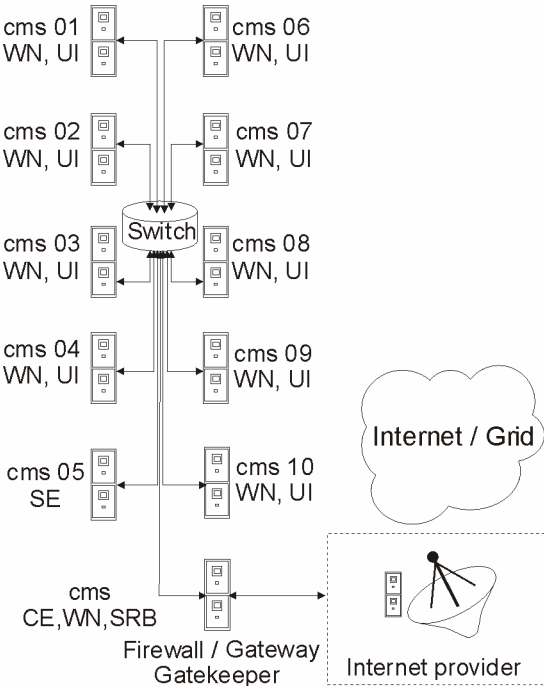

Fig.4. KCLC with Grid elements.

The next Globus service needed for the CE configuration is the MDS based on the *ldap* protocol. It provides an information support to the local batch system and corresponds to the external LCG structures about the system status. The system data are collected through the so-called Berkley data-based information index (BDII) or Grid information index (GIIS) server. The information is taken by the server from clients of the Grid resource information service (GRIS). The GRIS has been configured on all of the KCLC nodes.

The most important service responsible for the data transfer between LCG nodes is GridFTP. At present, the GridFTP meets all the requirements established for contemporary global networks.

As mentioned above, the LCG middleware contains a Globus wrapped version of the OpenPBS needed for interaction of the CE with the WN's. The Globus service that accomplishes this interface is the Globus resource allocation manager (GRAM).

At last, one needs to configure a very important EDG service called *locallogger*, which provides authorization of users on the CE and secure operation of the system.

In general, CE installation at the KCLC required deployment of 132 system packages and adaptation of 21 configuration files.

The WN's are the nodes, where jobs, actually, run. However, a user does not interact directly with them. All work of the WN's occurs behind the CE and is hidden from the user. The WN's should have access to user's packages (e.g., software specific for a given experiment), while the CE manages their operation via the *gatekeeper* and the batch system. In case of the WN, no special Globus services (like the ones used by the CE) are needed. Nevertheless, Globus utilities (GRIS, GRAM, etc.) remotely collect information about WN resources for the CE. The WN setup at the KCLC caused installation of 135 system packages with adaptation of 8 configuration files.

First of all, the WN services have been installed and configured on one of the KCLC nodes (*viz*., cms06). While doing this, a shell script automating this procedure was developed and tested and then used for subsequent WN setting up on other nodes.

The SE is a Grid node that has to fulfill an access to data disposed on an SS (large disk array, tape (DLT) robot, etc.) for other nodes. The SE provides a uniformity of such an access, with allowing a remote user to be unaware of the SS technical details, number and internal interaction.

The SE functions are determined by peculiarities of the HEP tasks. Information about data location on an SE can be directly obtained from the corresponding CE name. This makes it possible to avoid ineffective exchange with big data arrays between different RC's for processing or analysis, because the principle of data, computing resources and software encapsulation can be readily implemented in this scheme.

LCG file access management occurs by means of the so-called Virtual Organizations (VO's). In case of the KCLC, the VO's are the CMS and the CERN Grid deployment team (DTEAM). The VO's have special local accounts at the SE. Any LCG user belongs to a VO and possesses a personal certificate mentioned in the *grid-mapfile* at the SE. This certificate is put into correspondence with a local user with name that coincides with the VO name.

The SE is burdened with more system services than the CE or WN. In addition, it actively uses such Globus services as *gatekeeper*, GridFTP, Globus Job Manager,

MDS, *tomcat*, etc. Adaptation of 14 configuration files was needed for SE installation at the KCLC.

The UI is any node, which runs software providing direct access to Grid resources. It is that machine that gives an opportunity for a user to communicate with the Grid through the LCG command interface. The UI is used to submit jobs and get results of computations.

To run the UI functions, *openssh* must be installed and properly configured on the system. In case of a cluster, it is important that the *ssh* configuration would support batch job submission and processing. At the KCLC, this is achieved by a passwordless node-to-node access based on the host-based authorization provided by a special *sshd* daemon configuration.

One of the main Grid problems is security. It is solved not only by special protocols, but also through a package of measures, which include obligatory use of personal and other certificates, authorization with a password and ordinary restriction of access to ports of an element performed by a system of special package filters called *firewall*. All these measures are consistent with the current LCG middleware version installed at the KCLC, which is LCG_2_2_0.

In addition to the LCG middleware described above, we configured at the KCLC also the storage resource broker (SRB) [11], which is a separate (non-LCG) Grid element. The SRB is client-server middleware that enables a uniform interface for connecting to heterogeneous data resources over a network and data replication. At the KCLC, both SRB server and client services have been running since 2002.

Information about all data accumulated on world-spread SRB servers is contained on so-called metadata catalogue (MCAT) servers. An MCAT server stores and synchronously refreshes data catalogue images of SRB servers registered at this MCAT server. The KCLC SRB server has been registered at the MCAT server of the Bristol/RAL RC (United Kingdom).

An access to SRB data occurs by means of a command interface resembling "classical" file transfer protocol (FTP) shell, though allowing a user to manipulate pseudo-locally with data physically stored on different remote SRB servers. Similar to the SE, this scheme provides a uniform way of data treatment and hides technical details of the SS hardware.

At present, the KCLC SRB server contains CMS data generated on the cluster within the CMS Monte-Carlo event production program starting from 2002.

4. Participation in the CERN/CMS Monte-Carlo Event Production

It has become a common HEP practice to perform complete computer simulation of an experiment as a preliminary step of its accomplishment. Certainly, in case of such a complex system as the CMS, this task requires a great amount of computation. It is caused by both the complexity of physics processes to be simulated and the elaborate design of the CMS detector, response of which should be adequately reproduced. Such computer experiment is needed to optimize the detector trigger system. Then, it gives an opportunity to tune and check the integrity of the LCG middleware and other software developed for CMS data distributed storage and analysis. Also, an important objective of this activity is studying software tools developed by the collaboration for event reconstruction and training of physicists intending to participate in data processing and analysis.

Participation of the RC's in the complete computer simulation of the CMS experiment occurs via the so-called CMS Monte-Carlo event production. Computing resources of universities and research centers from more than 30 countries are involved in solution of this problem.

The scheme of the complete event production chain is displayed in Fig. 5. The computer experiment starts from the event generation (module CMKIN), performed by the PYTHIA simulation tool (see Ref. [12]) that implements the Lund version of the quark-parton model of proton-proton interactions. Detector response simulation (block CMSIM/OSCAR) is introduced by the Fortran-based package CMSIM performing complete GEANT [13] (version 3) simulation of passing particles through the CMS detector. Optionally, this step can be fulfilled with the OSCAR [14] package, the detector simulation object-oriented software, utilizing the C++-based GEANT_4 toolkit. The CMSIM/OSCAR simulation is the most CPU-time-consuming part of the production cycle. The average multiplicity of particles produced in proton-proton collisions at the LHC energy reaches several hundreds, and the history of each of such particles and its possible decay products should be followed up to the particle disappearance (due to decay, stopping or escaping the detector volume). A typical production job generating 250 CMS events takes more than 24 hours of a 1 GHz Pentium III processor. The result of this step is information about tracks of all particles passed through the detector. Based on this information, the signals in the CMS compartments are evaluated (step HITS). Digitization of these signals (stage DIGI) and reconstruction of particle tracks (RECO) with (optionally) taking into account possible pile-up events are accomplished by the ORCA tool [15].

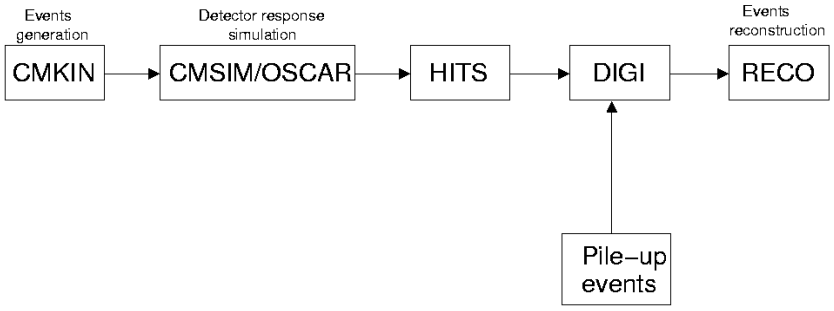

Fig.5. Complete cycle of the CMS event generation/reconstruction.

The KIPT/CMS Group has been participating in the event production runs since 2002 employing the KCLC. A production center has to have a cluster of computers capable of running batch jobs and a special CMS simulation software. Apart from the mentioned above physics software, the latter includes a number of special production tools.

An RC receives a request to simulate the events in the form of an assignment with a unique identification number (ID). This process is maintained with the help of a reference database (RefDB) located at CERN, which generates requests, distributes assignments and traces their status. When an assignment is received, it should be processed with a complex of Unix shell, Python and Perl scripts called McRunjob. Of course, McRunjob has to be properly configured with taking into account the system environment of the PC farm and the peculiarities of the given assignment. It takes the

assignment ID as an input and downloads the relevant data from the RefDB creating a number of jobs, with each of them being a complete entity capable of running independently. It also keeps the track of jobs by updating their status in the RefDB and using the BOSS software [16], which a wrapper for the cluster batch system providing an interface to the MySQL database.

In case of the KCLC, the jobs are submitted to a PBS queue, which distributes them for execution on different nodes of the cluster. If some of these jobs fail for some reason, they can be resubmitted later. The BOSS provides real time monitoring and book keeping for jobs submitted and running on the cluster and storing the information in the MySQL database located on the KCLC CE node ('cms').

The complexity of installation and configuration of the production software is associated with difficulties of adaptation of these complex tools to a specific platform with the peculiarities and nuances typical of a cluster system. Besides, versions of the physics packages must be agreed upon in order to get results that can be checked and reproduced.

When a production job is running, a summary file for this job is generated by the production software, which contains complete information about the computation process (versions of all physics packages actually used, initial random number, names and sizes of all generated files, execution host name, date and time of the beginning and the end of the execution, resources [RAM and CPU time] used, etc.). Upon completion of the job, the summary file is automatically either e-mailed directly to CERN, or written to the MySQL database to be processed later by a script transferring the relevant information to the CERN RefDB. In any case, information in the RefDB is updated on a regular basis. This provides permanent monitoring of the production activity in the RC's. On the other hand, the summary files give an opportunity to check or reproduce the production results, if some of them appear to be questionable or incomplete.

Reliability of the production results is also achieved through the so-called validation assignments (VA). Before running a production assignment, it is mandatory for a production center to ask for the VA from the RefDB and run it. The VA examines the system environment and correctness of the software versions. The resulting VA summary files submitted to CERN must coincide with sample data. Only in this case the RC is admitted to the physical event generation and can get the production assignment.

Concise statistics of KCLC participation in the CMS Monte-Carlo production runs is presented in Table 1. More than 600000 events have been generated at the KCLC within this program for the two years, with the majority of these data being the results of the most cumbersome production step CMSIM. The CMSIM data occupy ~1 Mbyte/event of the HDD space. There are also generated (CMKIN) and reconstructed (RECO) events in the form of the so-called HEP-standard "ntuples", as well as HITS files, which are comparable in the size per event with the CMSIM data. A physical content of a production assignment is given by the corresponding dataset name. For instance, dataset 'hi02_qcd80_90' means generation of QCD jets in the transverse momentum range p_T=80–90 GeV/c requested by the HI ("heavy ions") physics group in 2002, while 'jm03_Wjets_20_50' refers to production of weak $W^{\pm}$ bosons with p_T=20–50 GeV/c accompanied by QCD jets in proton-proton collisions at the LHC

energy requested by the JetMET ("jets and missing transverse energy") group in 2003. More detailed information about each of the datasets listed in Table 1 can be found in the CERN RefDB.

In conclusion, we would like to mention that, in addition to participation in the CMS event production program, the KCLC is intensively used in our studies of possibilities to observe a heavy Standard Model Higgs boson in decays $H \to ZZ \to ll\nu\nu$ with the CMS detector. Details concerning this activity can be found, e.g., in Ref. [17].

Table 1. Results of KCLC participation in CMS Monte-Carlo event production in 2002–2004.

Data Set	Assignment ID	Number of events generated	Data type
Hi02_photon100_110	1829	1000	CMSIM
Hi02_qcd80_90	1644	5000	CMKIN, CMSIM
Hi02_qcd100_110	1645	1000	CMKIN, CMSIM
eg02_BigJets	1958, 1959, 1960, 1961	85250	CMKIN+CMSIM, HITS, DIGI, RECO
Bt03_qcd120_170tth	2245	80000	CMKIN
mu03_tt2mu_Mll1000	2549	500	CMKIN
eg03_Wenu_calibration	2840	99500	CMSIM
Jm03_Wjets_20_50	3024	100000	CMKIN
eg03_Zee	3047	100000	CMSIM
Tt_ch_140_tb20	5344, 5345, 5481	150000	CMKIN, CMSIM, HITS

5. Conclusion

An enormous data flow expected in the LHC experiments forces the HEP community to resort to the LCG, a Grid-based model of data storage, processing and analysis. Accordingly, a worldwide infrastructure of LCG RC's is created, with the KCLC being a part of the Moscow distributed RC.

The KCLC is a specialized PC farm constructed at the NSC KIPT for computer simulations within the CMS physics program and preparations to the CMS data analysis. Currently, it combines 22 CPU's, and its total HDD storage is 2.5 Tbyte.

These performances allow us to participate in the CMS Monte-Carlo event production representing "complete" simulation of the CMS experiment. Such computer experiments require huge computation resources, and their accomplishment is distributed world-widely over the RC's. For the time being, more than 600000 events have been generated on the KCLC in such production runs. The most part of the generated data are stored at (and can be fetched from) the KCLC SRB server.

To proceed further with our preparations for the CMS experiment, the installation and proper configuration of the LCG middleware at the KCLC was needed. This is not a trivial task, since this software is neither universal nor "complete" product, and one has to go on in the configuration process step by step, in order to avoid system conflicts. At present, the LCG middleware that supports the CE, WN, SE, and UI functions for the KCLC nodes has been installed. A number of special system services had to be configured in each case. The employed at the KCLC procedure of manual step-by-step LCG software installation and configuration as well as the scripts

developed for its implementation are applicable for LCG middleware deployment on other computing centers scheduling allocation of their resources for tasks of LHC data processing or analysis.

To our knowledge, the KCLC is, at present, the only PC farm in Ukraine, which has been constructed to meet the HEP computing needs, and which is actively used in computation activities for the LHC experiments. Introduction at the KCLC of the computation concept based on the novel Grid technology is also pioneering for the country. So, we hope that our experience can be useful for construction of other computing systems of the similar type, and, on the other hand, our work is promotional for introduction of the Grid computing technology in Ukraine.

Further development of the KCLC is planned with considerable increase of its capacities and deeper integration into the LCG structures.

6. Acknowledgements

The authors thank Prof. P. V. Sorokin for his permanent interest to this work and helpful discussions.

REFERENCES

1. http://www.gridcomputing.com.
2. Foster I., Kesselman C., and Tuecke S. The Anatomy of the Grid Enabling Scalable Virtual Organization // Int. J. High Perform. Comput. Appl. 15(3), (2001).
3. The Compact Muon Solenoid Technical Proposal, - CERN/LHCC 94-38, 15 December 1994.
4. Lamanna M. The LHC computing grid project at CERN // Nucl. Instr. and Meth. **A**. (in press); http://lcg.web.cern.ch/LCG.
5. Stickland D. CMS Computing and Core-Software Report, - Proc.VIth Annual RDMS CMS Collaboration Meeting, “Physics Program with the CMS Detector”, IHEP, Moscow, Russia, December 19, 2001, p.293-308.
6. Levchuk L.G., Sorokin P.V., Soroka D.V., and Trubnikov V.S. NSC KIPT Linux Cluster for Computing within the CMS Physics Program // Problems of Atomic Science and Technology, Ser. “Nuclear Physics Investigations”. Kharkov 2002. N2(40), p.49-51.
7. http://www.openpbs.com/.
8. http://www.globus.org.
9. EDG User Guide, - IST-2000-25182, 15 January 2003,p.7.
10. MacDonald N., Minty E., Harding T., Brown S. Writing Message Passing Parallel Programs with MPI, - Edinburgh Parallel Computing Centre, The University of Edinburgh.
11. The Storage Resource Broker, - San Diego Supercomputing Center (SDSC), http://www.npaci.edu/DICE/SRB.
12. SjÖstrand T., Eden P., Friberg C., Lönnblad L., Miu G., Mrenna S. and Norrbin E. // Computer Physics Commun. 135 (2001) 238.
13. Brun R., Goossens M., and Zoll L. GEANT Users Guide, - Program Library Long Writeup W5013. CERN, 1993.

14. OSCAR, Object oriented Simulation for CMS Analysis and Reconstruction, http://crnsdoc.cern.ch/oscar/.
15. Object Oriented Reconstruction for CMS Analysis, - CMS IN 199/001.
16. Grandi C., Renzi A. Object Based System for Batch Job Submission and Monitiring (BOSS), - CMS Note 2003/XXX.
17. Levchuk L.G. On the Possibility to Observe with CMS a Moderately Heavy Higgs Decaying via $H \to ZZ \to ll\nu\nu$, - 8th Annual RDMS CMS Collaboration Conference, JINR, Dubna, December, 2003.