Письма в ЭЧАЯ. 2015. Т. 12, № 3(194). С. 624–627

DEVELOPMENT OF THE DISTRIBUTED COMPUTING SYSTEM FOR THE MPD EXPERIMENT AT THE NICA COLLIDER

K. V. Gertsenberger [1]

Joint Institute for Nuclear Research, Dubna

Experimental data processing and storing are topical issues in modern high energy physics experiments. Development of a distributed cluster based on the server farm of the Laboratory of High Energy Physics was started to accomplish these tasks in the MPD experiment at the NICA accelerator complex. This article describes the design approaches and methods of cluster development for storing and processing of data obtained at the multipurpose detector. The current cluster scheme and structure are presented; software for building data storage and parallelization of the MPD event processing is noted. The presentation introduces two methods to parallelize data processing: using PROOF software tool of the ROOT environment and a scheduling system under development by the author.

Обработка и хранение экспериментальных данных в современных экспериментах в области физики высоких энергий является крайне актуальными проблемами. Для решения этих задач в эксперименте MPD ускорительного комплекса NICA началась разработка распределенного кластера на базе «фермы» серверов Лаборатории физики высоких энергий. В данной статье описываются подходы и методы проектирования такого распределенного кластера для обработки и хранения данных, получаемых с многоцелевого детектора. Приведена схема и состав кластера, развернутого к настоящему времени, а также методы и программное обеспечение для создания хранилища данных и распараллеливания обработки событий эксперимента MPD. В статье представлены два подхода к распараллеливанию обработки данных: использование инструмента PROOF программной среды ROOT и системы планирования, разрабатываемой автором данной статьи.

PACS: 07.05.Kf

INTRODUCTION

According to the Joint Institute for Nuclear Research programme on creation of the ion accelerator facility for the range of colliding energy of nuclei being $\sqrt{S_{NN}} = 4{-}11$ GeV and the multipurpose detector MPD [1] optimized to study the properties of hot and dense baryonic matter in heavy-ion collisions, the Nuclotron-based Ion Collider fAcility (NICA) is constructed. To support the MPD experiment, MPDRoot software is developed. It serves for the MPD event simulation, reconstruction of experimental or simulated data and following physical analysis of heavy-ion collisions registered by the MultiPurpose Detector at the NICA collider.

[1] E-mail: k.gertsenberger@gmail.com

The development of the distributed cluster for the MPD experiment is required primarily for the following reasons: high interaction rate (up to 6 KHz) and particle multiplicity. An event of a central Au + Au collision at the energies of the NICA collider contains up to 1000 charged particles. As a result, one event reconstruction takes tens of seconds in MPDRoot software now, and then sequential processing of one million events can take several months. Furthermore, the total required data storage is estimated at five to ten PB of raw data obtained from detectors per year. There are two main directions of the distributed NICA cluster development: data storage development for the experiment and organization of parallel event processing.

1. DEVELOPMENT OF THE DATA STORAGE FOR THE MPD EXPERIMENT

Creation of the distributed NICA cluster based on the computer farm of the Laboratory of High Energy Physics for the MPD experiment was started this year. Now it consists of one hundred twenty-eight processor cores connected by Infiniband with a network bandwidth up to ten Gb/s (Fig. 1).

A GlusterFS distributed file system [2] was used to organize the data storage at the NICA cluster. It aggregates existing file systems in a common distributed file system. TCP/IP protocol or Infiniband RDMA can be used to interconnect data nodes on GlusterFS. Automatic replication and self-checking services working as background processes make it possible to prevent data loss and to restore files in case of hardware or software failure.

The existing ext3 file systems were joined on the cluster machines to the */nica/mpd* shared volume. The shared partition */nica/user/* was created for home directories so that users connecting to any machine of the NICA cluster are directed to the same home directory. All the amounts are replicated on hard disks of the different machines. The developed GlusterFS data storage is actively used by about sixty MPD experiment members now.

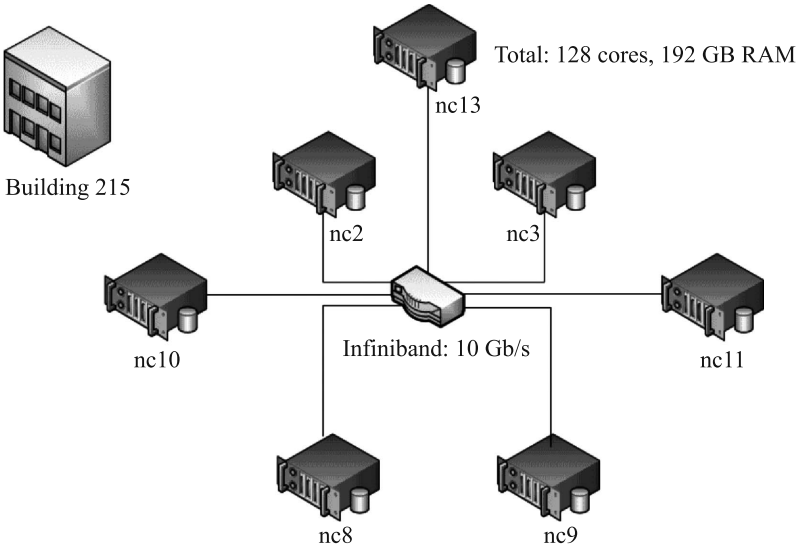

Fig. 1. Current NICA cluster scheme for the MPD experiment at LHEP

2. USING PROOF SYSTEM FOR PARALLEL MPD EVENT PROCESSING

To parallelize the event data processing on the NICA cluster, two methods are implemented: using PROOF [3] software tool to parallel data processing in a ROOT macros on parallel architectures and developing the MPD scheduling system for the task distribution on the cluster nodes.

The parallel ROOT Facility (PROOF) is a part of the ROOT environment. It uses data-independent parallelism based on the lack of correlation for the MPD events to process different events in parallel. PROOF orients on three parallel architectures: one multiprocessor or multicore machine, a heterogeneous computer cluster, and a GRID system.

PROOF support was added to MPDRoot software and reconstruction code was rewritten according to PROOF rules and classes. The last new parameter of the reconstruction: *run_type* has default value "local" for sequential processing. To parallel MPD event processing on one multicore machine with PROOF-Lite user can use "proof" string value and can limit the threads number by "workers" number, e.g., "proof:workers = 3". To speed up the data processing on the NICA cluster with PROOF On Demand server, the last parameter is used and also can limit the number of the workers by the following value:
"proof:mpd@nc8.jinr.ru:21001:workers = N_{workers}".

PROOF works on the NICA cluster as described below. A client sets the macro parameter to run it on the PROOF cluster in parallel. PROOF sends this task to PROOF On Demand server. The server runs the macro on the worker nodes and each assigned node receives one event from the input file to process. If worker node finishes the processing of the current event, it will receive the next one. When all the events are processed, the result will be merged on the master server and will be sent to the client or to the output file.

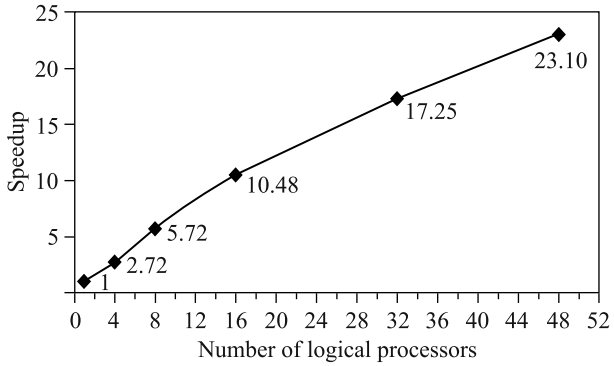

Fig. 2. Speedup of the reconstruction on the NICA cluster

The graph in Fig. 2 presents the reconstruction speedup on the NICA cluster with PROOF On Demand for one thousand MPD events.

3. MPD-SCHEDULER FOR DISTRIBUTED JOB EXECUTION

To distribute users' jobs on the NICA cluster and run it in parallel, the MPD scheduling system was developed. MPD-scheduler was implemented in C++ language with ROOT classes' support. It uses the Sun Grid Engine (SGE) [4] scheduling system to distribute the

MPD jobs on the NICA cluster. SGE combines cluster machines at the LHEP farm into the pool of worker nodes with seventy-eight processor cores.

Jobs for distributed execution on the NICA cluster are described and passed to MPD-scheduler as XML file. The description starts and ends with tag ⟨job⟩. Tag ⟨macro⟩ sets information about macro being executed by MPDRoot, its path and arguments. Tag ⟨file⟩ defines files to process by macro. It can set absolute file path or define a file list from the MPD simulation and production database. Tag ⟨run⟩ describes run parameters and the allocated resources for the job.

To execute user's jobs on the NICA cluster, MPD-scheduler parses the job description and runs shell scripts by *qsub* command of the Sun Grid Engine environment. Sun Grid Engine defines free workers of the cluster and runs the event data processing in parallel. When the worker finishes its part of the processing, the status of this worker will be changed to free value, so it can execute another user job. MPD-scheduler has the possibility to merge the result files in the mode of partial file processing.

CONCLUSIONS

The following conclusions can be noted. The distributed NICA cluster was deployed based on the LHEP farm for the NICA/MPD experiment. The data storage was organized with the GlusterFS distributed file system. PROOF On Demand cluster was implemented to parallelize event data processing for the MPD experiment, PROOF support was added to the MPDRoot software. The system for the distributed job execution — MPD-scheduler — was developed to run MPDRoot macros concurrently on the cluster. The practical values of the speedup for MPD event processing were obtained. The information about described systems is presented on our site *mpd.jinr.ru* in detail.

This work was supported by the 2012 grant for young scientists and specialists of JINR.

REFERENCES

1. *MPD Collab.* The MultiPurpose Detector — MPD. Conceptual Design Report. Dubna: JINR, 2012. 259 p.
2. GlusterFS Developers. Gluster File System 3.3.0. Administration Guide. Gluster, 2012. 134 p.
3. *Hayrapetyan A., Vala M.* ROOT and PROOF Tutorial. Karlsruhe Inst. of Technol., 2012. 58 p.
4. *Haas A.* Sun Grid Engine 6.1 + DRMAA Interface. Sun Microsystems GmbH. 2007. 33 p.