A Probabilistic Approach to the Simulation of Data Processing Centers

Vladimir Korenkov[1,2,⋆], *Andrey* Nechaevskiy[1,⋆⋆], *Gennady* Ososkov[1,⋆⋆⋆], *Daria* Priakhina[1,⋆⋆⋆⋆], and *Vladimir* Trofimov[1,†]

[1] *Laboratory of Information Technologies, Joint Institute for Nuclear Research,*
Joliot-Curie 6, 141980 Dubna, Moscow region, Russia
[2] *Plekhahov Russian University of Economics, Stremyanny per. 36, 117997 Moscow, Russia*

Abstract. The simulation of a data center for the storage and processing of data from the NICA detectors is an important step towards the creation of the NICA computing system. A model developed in the frame of the probabilistic approach to the solution enables decisions concerning a lower bound of the necessary resources for full data transfer of the detector records to the storage system.

1 Introduction

The high-energy physics experiments make heavy use of the distributed computing infrastructures for the storage and processing of huge amounts of data. The estimate of the necessary configurations of such infrastructures heavily relies on the preliminary realistic modeling of the data flow among the various components of the computing architecture.

The simulation enables the prediction of the behavior of the data storage modules at all stages from the data acquisition (DAQ) system to the long-term storage. This makes it possible to identify bottlenecks in the system at the design stage and to evaluate the necessary resources. There are many tools for modeling distributed computing infrastructures each of which being subject to restrictions [1]. The analytical methods [2] do not allow estimating real multi-level computing architectures with complex distribution functions and task flow handling. The simulation requires excessive granularity of data flows [3], which increases the complexity of the model implementation.

Given the architecture of a physical facility, it is necessary to create a specialized tool for modeling data flows and tasks. A probabilistic approach was chosen to simulate a distributed data storage and processing system for NICA experiments. This rests on the understanding of the organization of the high-energy physics experiments.

[⋆] e-mail: korenkov@jinr.ru
[⋆⋆] e-mail: nechav@jinr.ru
[⋆⋆⋆] e-mail: ososkov@jinr.ru
[⋆⋆⋆⋆] e-mail: pryahinad@jinr.ru
[†] e-mail: tvv@jinr.ru

2 The chain of the data storage and processing

A data set is obtained during active periods, which are interspersed with both planned interruptions (e.g. to configure the equipment) or accidental ones (in case of equipment failure). The distributed data storage and processing systems include several storage levels. The raw data recorded by the detector are written to the data acquisition buffer. These data are transferred to the intermediate storage for pre-processing. Finally, data are transferred to the long-term storage in a robotized library. All storage devices have limited capacity and the data transfer channels have limited bandwidth. Recording will stop if there is no available space on the storage device. If one of the storage levels cannot receive data, then a higher level will store them if it can. If there is no space on the tape, recording on this drive stops during the time needed to replace the tape. If the data acquisition buffer is full, then data will be lost because the detector has not its own data storage buffer. This is an emergency.

Data processing is carried in a computing farm. A data processing job goes to the computing farm. The job will be launched if there is an available processor element. If not, the job will stay in the queue and wait for an available slot. Data are needed to run the job, so the data will first be transferred from data storage to the farm.

3 Simulation approach

The goal of our simulation is to determine the hardware configuration which will ensure the operation of the data storage and processing system with specified quality. A basic quality-criterion parameter is the data loss (in % of the total data volume from the detector). The amount of data coming from the detector in 1 second is a random variable with a certain probability distribution. The planned shutdown is a deterministic event that starts and ends at specific times. Equipment failures are random events. The probability of occurrence of such events and the duration of recovery are assumed to follow from previous exploitation of similar facilities.

The simulation software includes modules related to the database, the equipment configuration settings, the data transfer and data processing.

The equipment configuration simulation is a cyclic process. At the first step random variables are generated. At the second step data transfer and processing are simulated. At the third step the results are checked. This produces total data loss estimates (in %) which are compared to a threshold.

The developed program was tested by modeling a system of distributed computing structure using abstract data and workflows. 24 hours (86400 s) of the working facility were simulated.

The simulated system includes data generators (triggers), disk servers, tapes, computing farms and data transfer channels (Fig. 1). As long as the equipment descriptions are ready, the next step is the specification of the input parameters (see Table 1). The processing of 200 jobs was simulated, assuming file sizes randomly distributed from 100 GB to 1 TB. Files are stored on the buffers (T1) and data pools with specified I/O speed. Each job was characterized by its processing time at the farm (LIT or LHEP), defined as the time required to complete the job. This time is a number from an interval (bounded by a minimum and a maximum value) defined from the analysis of real experiments.

4 Simulation results

The program was used to evaluate the load of a computing farm (Fig. 2a) and the maximum possible load of the data transfer channels (Fig. 2b).

Figure 2 shows the temporal evolution of the number of available cores in the two computing farms (left) and of the load of the farm resources (right). Figure 2b shows the load of the data transfer

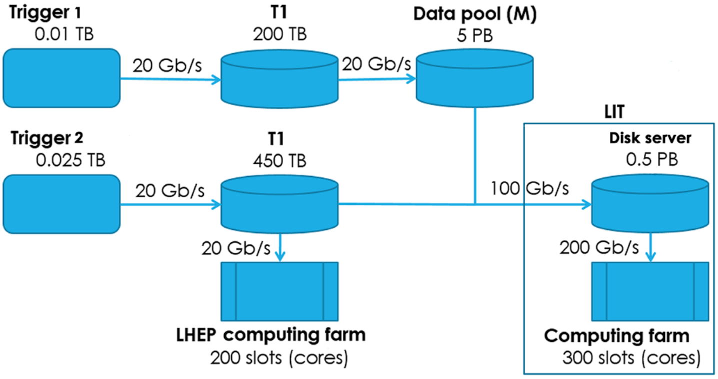

Figure 1. Schematic architecture of the distributed computing structure with specified hardware parameters

Table 1. Additional input parameters for simulation

Parameter	Value
Number of jobs per flow	200
Processing time in the LIT computing farm	Uniform distribution, mean = 10986 ms
minimum value	3677 ms
maximum value	18544 ms
Processing time in the LHEP computing farm	Uniform distribution, mean = 11428 ms
minimum value	3643 ms
maximum value	18544 ms
T1 I/O speed	1.3 GB/s
Data pool (M) I/O speed	2.2 GB/s

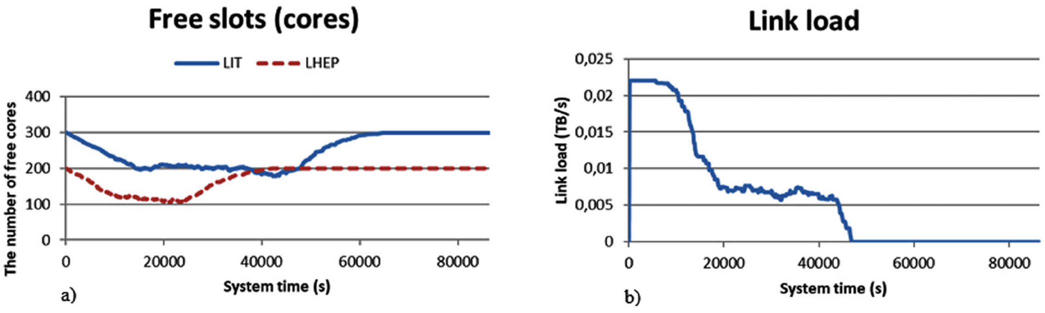

Figure 2. a) Temporal evolution of the number of the available cores in the two computing farms; b) Load of the channel between the LIT data pool and the LIT computing farm

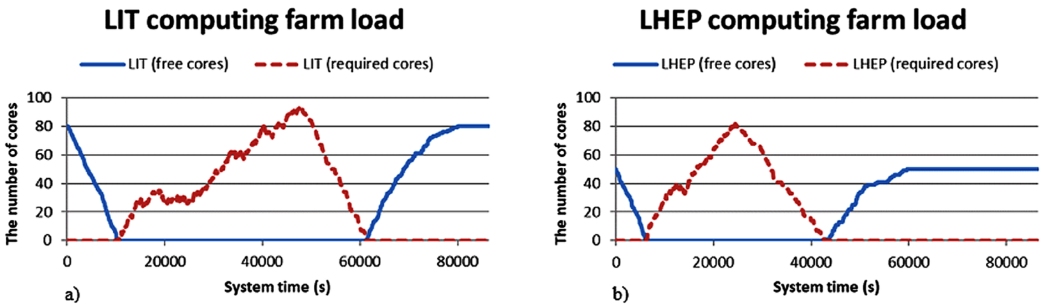

Figure 3. Job queues simulation results (LIT and LHEP computing farms)

channel between the LIT data pool and the LIT computing farm (in Fig. 1, this is the channel between the LIT disk server and the LIT computing farm). A peak of data transfer was registered at the beginning of the simulation. The system loading gradually decreased.

A second simulation instance illustrates the lack of resources. For the same simulation parameters the number of farm slots was changed. The number of slots on computing farms was 80 (LIT) and 50 (LHEP). The simulation results show the growth of the job queues (Fig. 3).

5 Conclusions and outlook

A new simulation program was developed. It can ensure the simulation of operation of the data storage and of the processing system by means of a probabilistic approach. The developed program was tested by modeling a system with distributed computing structure for abstract data and job flows. The simulation results show how to optimize the load of the computing farm. We are going to check the reliability of the model predictions on real data flows in actual BM@N experiment runs.

References

[1] C. Dobre, F. Pop, V. Cristea, Journal of Algorithms and Computational Technology **5**, 2, 221–257 (2011)

[2] Yu.S. Popkov, Management problems **3**, 10–20 (2003)

[3] V.V. Korenkov, A.V. Nechaevskiy, G.A. Ososkov, D.I. Pryahina, V.V. Trofimov, A.V. Uzhinskiy, Computer Research and Modeling **7**, 3, 691–698 (2015)