ACAT 2021

29 Nov. to 3 Dec. 2021
Virtual and IBS Science Culture Center, Daejeon, South Korea

DEVELOPMENT OF THE CONFIGURATION DATABASE AND ASSOCIATED INFORMATION SERVICES FOR THE NICA EXPERIMENTS

E. Alexandrov[a], I. Alexandrov[a], A. Chebotov[a], K. Gertsenberger[a], I. Filozova[a], D. Priakhina[a], G. Shestakova[a]

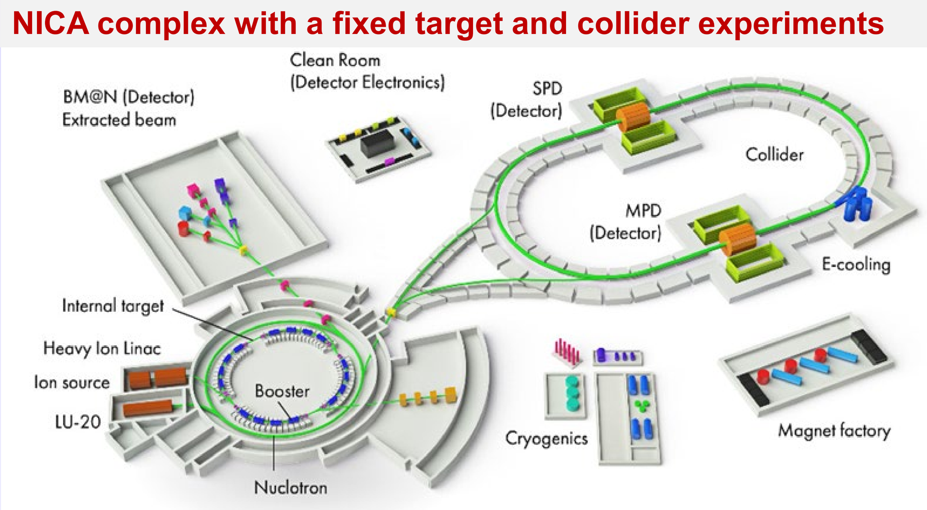
a: Joint Institute for Nuclear Research – JINR (RU)

1. Introduction

One of the priority tasks of the **Joint Institute for Nuclear Research** (**JINR**) for the next years is the creation of the **NICA** accelerator-storage complex [1] for the study of heavy ion collisions with high luminosity at interaction energies in the center of mass of 4 - 11 GeV. The complex of information systems [2] for the NICA experiments has been developed on the basis of the study of general functioning systems for data collection and processing, information systems and databases that are used in high-energy physics experiments, their classification, and the possibility and necessity of creating similar systems or individual components in the experiments of the NICA project. The developed Configuration Database is an essential part of the complex.

The **Configuration Information System** is used to store and provide data on the configuration of experiment hardware and software systems when collecting data from the detectors in online mode. The developed database stores both a set of various configuration parameters, such as those required for setting the detectors into operation modes, for instance, working voltage, and descriptions of a sequence of software tasks (processes), including online raw data digitization, online histogramming, fast event reconstruction and event monitoring, to be started and run during experiment sessions.

NICA complex with a fixed target and collider experiments

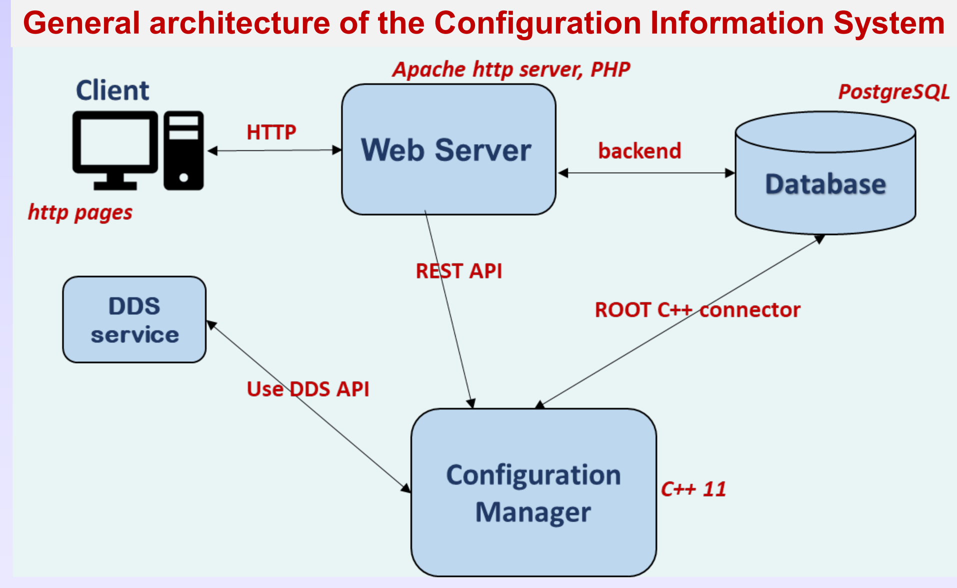


The possibility of the partial use of existing configuration systems [3] has been considered, but in this case a large amount of functionality would be redundant for the **NICA** experiments. It has been decided to develop its own system. The **Dynamic Deployment System** (DDS) [4] has been chosen for online process management since the system is easy to deploy and provides all the necessary capabilities with an application programming interface (API).

2. General architecture of the Configuration Information System

The general architecture of the **Configuration Information System** is illustrated in the figure below. The core of the system is the Configuration Database developed in PostgreSQL. The database stores all configuration data, both hardware and software, that are needed for online event processing. The client, implemented as a web interface that contains http pages viewable in any browser, communicates with the Configuration Database through the Apache HTTP Web Server, which in turn is connected with the database as a backend. The Web Server allows sending commands to the Configuration Manager using the REST API service.

General architecture of the Configuration Information System



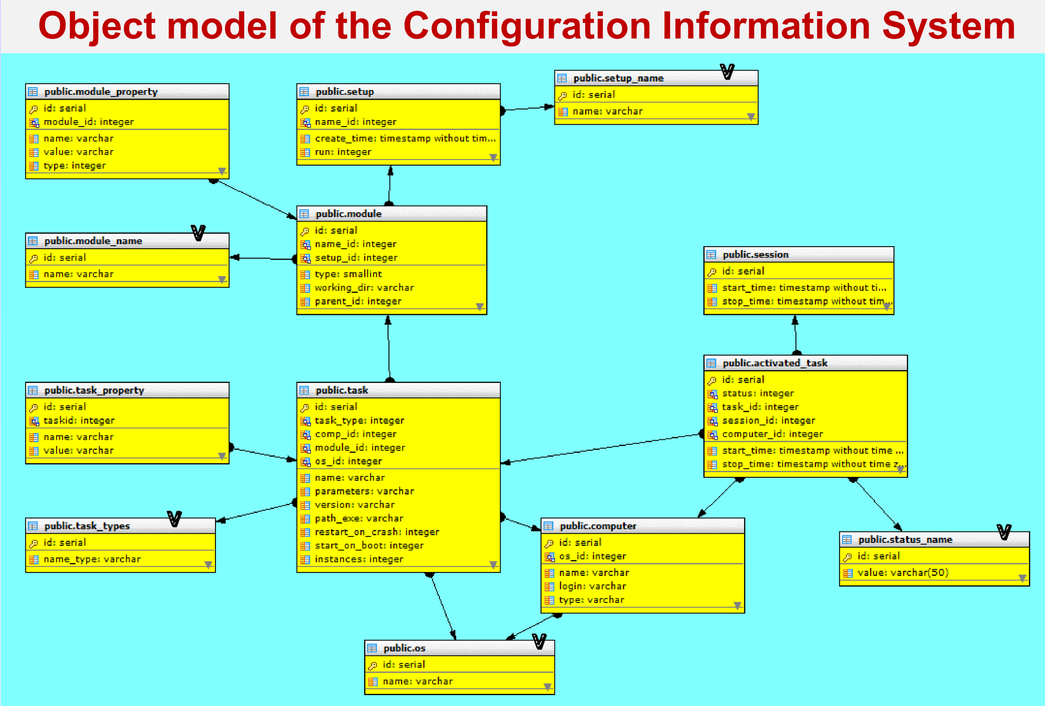
The **Configuration Manager** is responsible for starting, stopping and supervising tasks. It runs as a daemon process and performs obtained commands from the Web server using the **DDS** service that communicates with the **DDS** server via the C++ DDS API. Furthermore, the **Configuration Manager** reads data from the database and writes the necessary information concerning started, crashed or stopped tasks using the ROOT C++ connector.

3. Database object model

The object model of the Configuration Information System is presented in the figure below. The top object of the system is a ***setup*** that contains a list of the corresponding ***module*** objects. The ***module*** represents a detector or software system responsible for solving tasks of a common objective. The **Configuration System** contains a list of catalogues for a number of attributes. These objects are marked with "V" in the figure.

The ***module_property*** and ***task_property*** objects represent the necessary module and task properties respectively with parameters defining its name and value.

Object model of the Configuration Information System

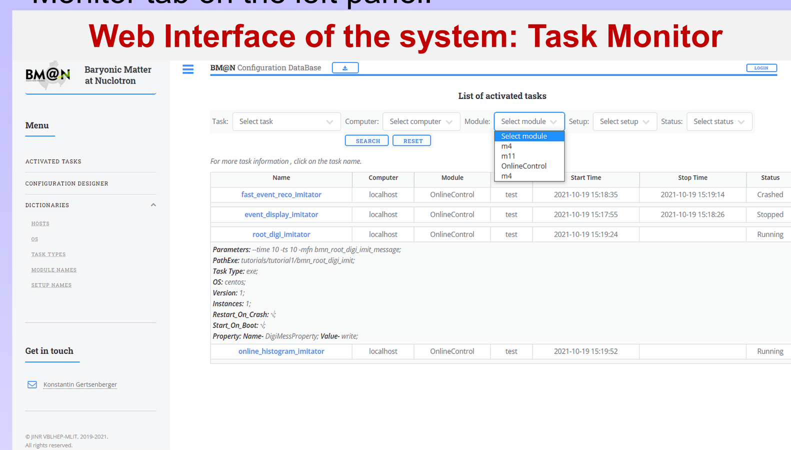


The ***task*** object of the ***module*** comprises all the attributes sufficient to start the task with the required parameters. The ***task*** also has a reference to the operating system object (***os***) to ensure that the task can be launched on a given computer. The reference to the ***computer*** object sets a host, on which the task will be started. The operating system installed in the computer is defined by a reference to the ***os*** object.

4. Web interface

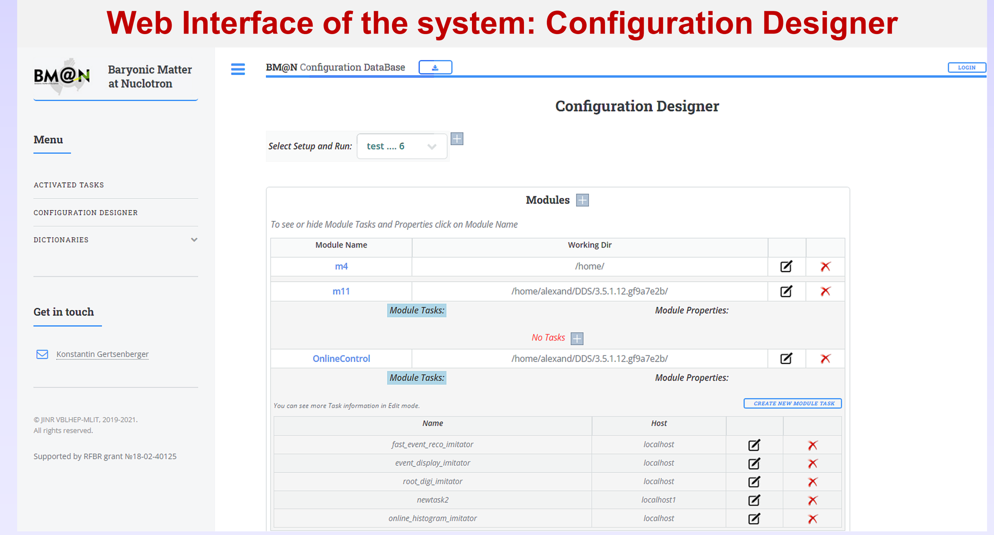
A set of necessary functions for monitoring active tasks, viewing, searching and managing objects of the Configuration Database has been implemented in the Web Interface. The user can select the Task Monitor tab on the left panel.

Web Interface of the system: Task Monitor



The Task Monitor tab displays a list of all running tasks, and collaboration members can use filters located at the top of the panel. The table contains only the most frequently used task parameters. All details of the task can be seen by clicking on the task name. An example of task monitoring is shown in the figure on the left.

Web Interface of the system: Configuration Designer



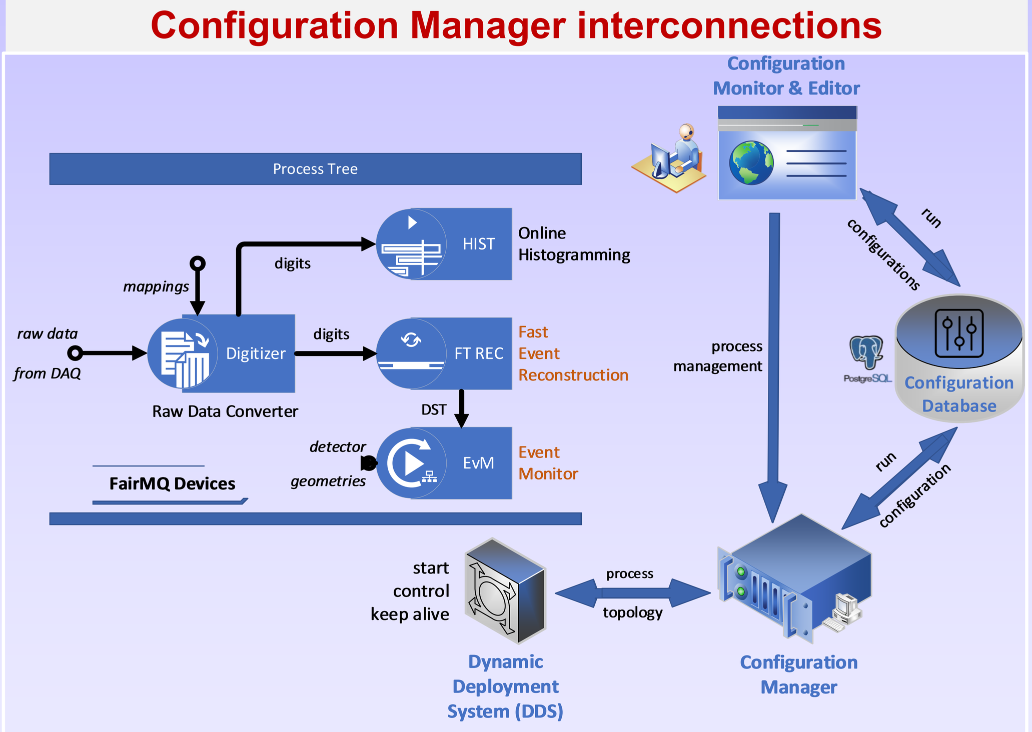
The Configuration Designer tab offers important features such as adding new configurations and managing the current ones for the selected setups of the experiment. The configurations are added using a top-down approach, i.e., one first selects a setup object, then adds or selects modules to be connected to the setup in this configuration and then the corresponding tasks are added for each module. One can see an example of the Configuration Designer in the figure on the right.

5. Configuration Manager

The **Configuration Manager** of the system is responsible for managing processes according to the active configuration and updating the database with information on running tasks. The interconnection of the **Configuration Manager** with other systems is shown in the figure below. The manager acts as a daemon and communicates with the Configuration Database, Web Server and DDS system.

The **Configuration Manager** listens for remote commands such as starting all setup tasks, stopping a set of tasks, starting or stopping a single task. It obtains detailed information on the selected setup and parameters of child tasks from the database. The **Configuration Manager** uses the **DDS** server to execute the above commands. It converts the task configuration into a dedicated DDS topology and sends the appropriate requests to the DDS server using the C++ DDS API. The **Configuration Manager** uses callback messages with information on the statuses of running tasks to store the information in the database.

Configuration Manager interconnections



6. Conclusions

The Configuration Information System for the NICA experiments has been designed and is at the final stage of the development.

The object model of the Configuration Database has been described, and the database has been implemented in PostgreSQL.

The Configuration Manager has been developed as a daemon that processes client commands related to task management and monitors the results through the database.

The DDS system has been chosen and is used by the Configuration Manager to start and manage processes.

The Web Interface has been developed and is constantly being improved.

Acknowledgments

The work was funded by the Russian Foundation for Basic Research according to the research project 18-02-40125.

References

1. NICA Collaboration. Searching for a QCD Mixed Phase at the Nuclotron-Based Ion Collider Facility, 2014.
2. E. Alexandrov, I. Alexandrov, K. Gertsenberger, et al., "Information systems for online and offline data processing in modern high-energy physics experiments," Mod. Inf. Technol. IT-Educ., No. 15, 654–671 (2019).
3. Almeida J., Dobson M., Kazarov A., Miotto G.L., Sloper J.E., Soloviev I., Torres R. The ATLAS DAQ system online configurations database service challenge // Journal of Physics: Conference Series. 2008. V. 119(2): 022004.
4. Lebedev A., Manafov A. DDS: The Dynamic Deployment System // EPJ Web of Conferences. 2019. V. 214. P. 01011.