Письма в ЭЧАЯ. 2016. Т. 13, № 5(203). С. 905–912

THE NUCLOTRON AND NICA CONTROL SYSTEM DEVELOPMENT STATUS

E. V. Gorbachev[1], *V. A. Andreev, A. E. Kirichenko, D. V. Monakhov, S. V. Romanov, T. V. Rukoyatkina, G. S. Sedykh, V. I. Volkov*

Joint Institute for Nuclear Research, Dubna

The Nuclotron is a 6 GeV/nucleon superconducting proton synchrotron operating at JINR, Dubna, since 1993. It will be the core of the future accelerator complex NICA which is under construction now. The TANGO based control system of the accelerator complex is under development now. The report describes its structure, main features and present status.

PACS: 07.05.Bx; 07.05.Dz; 07.05.Hd

INTRODUCTION

The NICA complex will consist of heavy-ion and polarized particles sources, RFQ injector, heavy- and light-ion linear accelerators, superconducting booster synchrotron, the Nuclotron and two superconducting collider rings [1].

The control system of the NICA complex aims at accomplishing a few main tasks:

1) Management of large amount of equipment which is distributed on the accelerator complex area.

2) Realization of different regimes of the accelerator complex working cycle — colliding or fixed target experiments, various ion types and energy.

3) Strict synchronization of accelerators in the chain.

4) Comprehensive beam diagnostics during the entire cycle.

5) Providing protection and safety measures.

We can specify a few important features that the NICA control system has to provide:

1) Centralized administration and monitoring of the control system components.

2) Reliable operation, quick recovery after possible failures.

3) Access to equipment has to be restricted for certain personnel with rights limitation according to user roles.

4) Ease of support, modification and scaling during a long life-time of the accelerator complex which will operate until 2045.

5) Rapid development and easy deployment of the control system, taking limited time and man-power into account.

6) Possibility to integrate third-party control systems as some components of the accelerator complex will be designed and constructed by external organizations.

[1]E-mail: gorbe@sunse.jinr.ru

CONTROL SYSTEM LAYOUT

The control system is a distributed network of computers which communicate by means of transport protocol over TCP/IP. The NICA control system uses the TANGO controls [2] as the middleware providing such a communication protocol.

TANGO is the modern distributed control system framework based on CORBA. The fundamental unit of TANGO is a device which is an abstraction hiding real equipment or program component behind the standard interface. TANGO provides high-level client application interface which has necessary programming classes to implement client–server communications — synchronously or asynchronously execute commands, read or write attributes, or use events to acquire the data from the TANGO devices. TANGO incorporates a number of tools to build efficient control system environment including centralized administration and monitoring, access control, logging system, data archiving and code generation for rapid development of the TANGO device servers using C++, Java and Python.

Three layers of the NICA control system components can be distinguished (Fig. 1):

1) Front-end layer consists of industrial computers, intellectual controllers, crates that directly control equipment and acquire data from sensors. Front-end computers run low-level TANGO programs that realize data acquisition, equipment handling and hide protocol and connection details from higher-layer components.

2) Service layer consists of high-level TANGO devices representing entire subsystems. They collect data from front-end TANGO devices, process it and realize algorithms to control some large subsystems. Those high-level programs provide a standard TANGO interface for entire subsystems allowing client software to execute commands, read and write attributes without knowledge of subsystems structure. Besides, the service layer provides a set of services that are necessary for efficient control system functioning — administration, con-

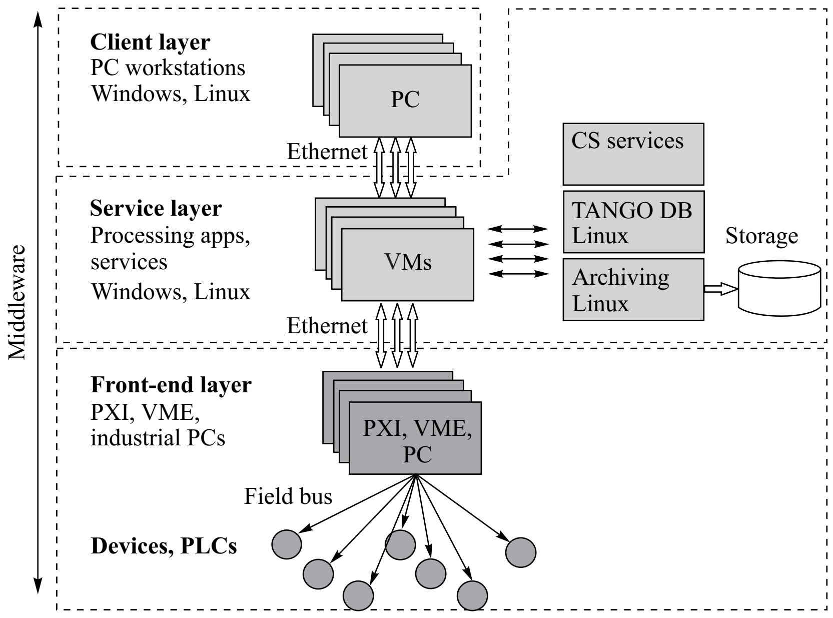


Fig. 1. NICA control system layout

trol system hardware and software management, monitoring, data archiving, development services.

3) Client layer represents the accelerator complex state to operator, visualizes the acquired data and allows operator to perform control tasks. We would like to provide global interface to allow operator to access the entire accelerator complex control with possibility to navigate to its components.

Front-End Layer. Primary requirements for the front-end layer hardware are easy and rapid development, good reliability and performance, easy maintenance during the accelerator complex operation. The development and support of custom hardware for the large accelerator complex is a very resource-consuming task. It is decided to use commercial hardware for most of the control systems tasks. The base of the NICA controls front-end layer is National Instruments PXI, which is a modular Eurocard packaging platform with PCI and PCI express busses with additional synchronization and trigger lines.

The PXI platform offers a wide range of modular instruments of many types including digital and analog I/O, industrial interfaces, digitizers and scopes, signal generators and many others. National Instruments provides Windows and Linux drivers for most of the hardware and supports LabVIEW as well as text-based languages.

While most of the control system tasks can be implemented by modular instruments, some unique signal generation, processing and synchronization tasks can be realized using programmable logic devices. National Instruments provides a few FPGA solutions to implement custom hardware. One of them is FlexRIO product family, which consists of PXI and PXI express modules with Xilinx FPGA, large amount of on-board memory and standard or custom I/O modules that are attached to the front panel of FlexRIO and have access to pins of FPGA. A developer can implement custom FPGA firmware as well as custom input and output signal processing, conditioning and conversion.

Another solution is CompactRIO controller, which is FPGA- and processor-based platform with standard or custom I/O modules. It is running real-time OS and controlled over Ethernet.

The TANGO device servers for a range of National Instruments hardware performing common acquisition tasks were developed. They include drivers for digitizers and scopes, analog and digital input and output, timers and counters, digital multimeters, temperature sensors acquisition modules.

Software for FPGA based boards consists of FPGA firmware and FPGA interface program performing data exchange between the controller and FPGA. LabVIEW provides rapid firmware development using graphical language and VHDL. Interface programs can be developed as TANGO devices using FPGA Interface C API. Those TANGO devices are running on PXI or CompactRIO controllers providing an easy way to integrate the FPGA-based solutions into the TANGO environment.

Developed drivers allow quick deployment of almost any National Instruments acquisition board. Each driver can perform a single input or output task so it is necessary to run a few TANGO devices to realize all features of an acquisition board. The acquisition properties such as sample frequency, samples number, trigger source, and others are configured as the TANGO devices properties.

Client Layer. TANGO provides a set of graphic user interface toolkits for rapid development of graphical client applications in all supported programming languages — C++, Java and Python. There are also LabVIEW TANGO bindings that allow developing full featured TANGO client applications using rich LabVIEW visualization features. Most of the existing desktop client applications for the NICA control system are developed in LabVIEW with TANGO bindings.

Few auxiliary TANGO devices were developed to simplify creation of web client applications by providing access to TANGO from web browser:

1) TANGO device realizes REST protocol. It contains embedded http server providing access to the TANGO devices attributes and commands at URL combined from host, port, device and attribute names. The reply is JSON string consisting of attribute name, value, timestamp and quality. POST method can be used to execute command with input arguments.

2) TANGO device for accessing attributes via WebSocket protocol. The device name and attributes names are specified in properties, and the data update rate is configured via polling period.

3) TANGO device intended for authentication of users to restrict access of web applications to control system components. The access rights are configured in database.

These auxiliary TANGO devices allow creation of web application in JavaScript without any intermediate application servers.

Service Layer. Service layer contains high-level TANGO devices that collect data from the front-end TANGO devices, process it and realize algorithms to control large subsystems. High-level applications of the service layer do not interface with any hardware, so they can run in any place of the control system and can be virtualized. Virtualization is an effective way to deploy control system applications providing easier management of virtual machines (VM), better tasks isolation, fine tuning of CPU time or disk space allocation and more efficient usage of hosts' resources. Virtualization also makes it possible to obtain high availability features — any virtual machine can be quickly restored or migrated from failed physical server.

Proxmox VE is used as a virtualization solution. It is a free server platform for creation and management of virtual infrastructure including virtual machines, local and shared storage and high-availability clusters. It supports both Kernel-based Virtual Machines (KVM) and Linux container-based virtualization which provides Linux containers with no performance impact.

Proxmox VE provides Web-based interface that allows performing role-based administration of virtual infrastructure. It supports a few types of shared storage, has integrated backup tool that allows one to make instant or scheduled reserve copies of running virtual machines without any downtime.

Shared storage is the vital part of virtualization to provide high availability of virtual machines. There are a few requirements for the shared storage:

1) Good performance as we need to run many VM images on it.

2) Redundancy to allow using VM images even if some disks or servers failed.

3) Scalability — storage must be expandable without losing performance to provide the data storage for future control system tasks and data acquisition.

The NICA control system uses a distributed object storage cluster Ceph [3] as shared storage solution. Ceph is an open source, reliable and easy to manage solution. It can be built on commodity hardware and has low operation cost. Ceph storage cluster consists of nodes which communicate with each other via network and distribute and replicate data dynamically. Ceph's RADOS block device (RBD) is an ideal way to store VM images that are stripped and replicated across the entire storage cluster. Ceph RBD is thin provisioned so the unused image space can be reclaimed.

The Ceph cluster requires at least one Ceph monitor daemon (MON) and at least two object storage daemons (OSD) to provide two copies of data. A Ceph monitor maintains a map of cluster state, while the object storage daemons store data, handle replication, recovery, rebalancing and provide some monitoring information to the Ceph monitors by checking other OSD daemons via heartbeat.

Data location is synchronized across the cluster using the CRUSH algorithm. Storage clients and OSDs use CRUSH to compute data placement without depending on a central lookup table. The CRUSH algorithm enables the Ceph storage cluster to scale, rebalance and recover dynamically.

The network architecture is very important for the highly scalable fault tolerant storage cluster. The network must have the capacity to handle the expected number of clients and per-client bandwidth. The network must also handle the Ceph OSD heartbeat, data replication, cluster rebalancing and recovery traffic. In normal operation, a single write to the primary OSD results in additional writes to secondary daemons, based on the replication factor set. So, the cluster traffic significantly exceeds public traffic under normal operating conditions. We use two separate networks — one for the client traffic and the other for storage cluster traffic.

Each Ceph node can contain one or a few physical hard disks. It runs one OSD daemon per disk and one journal device per object storage daemon. We use enterprise class SSD disks as journal devices as they allow accelerating random write operations and accommodating bursts. We also use battery backed caching controller, all disks are connected via controller in RAID0 mode with single disk in RAID configuration. It helps to increase random IO by using RAID controller cache to optimize data writes.

It is possible to run the Ceph services directly on Proxmox VE nodes, so the same hosts are used to provide both distributed storage and virtualization cluster. The cluster presently consists of four nodes each with dual eight core E5-2600 Intel Xeon processors, 32 GB of RAM, five reliable 15k RPM SAS2 disks and enterprise class solid-state drive (SSD) for the operating system and the Ceph journal devices. The network equipment includes two 1 Gbit network switches to separate the public and Ceph cluster network. One of them will be upgraded to 10 Gbit network switch to provide necessary network capacity.

One of the most important goals is to minimize the accelerator complex downtime due to control system failures and provide quick failover in such cases. The high availability of the control system is achieved by hardware solutions which include using of uninterruptable power supplies (UPS), redundant power supplies for servers, enterprise class long-lifetime drives and data redundancy for virtual machines and databases.

All virtual machines run on the Proxmox VE cluster with images stored on the Ceph storage with replication factor of three. Hard drives failures are handled by Ceph and not affecting the operation of the control system at all. All virtual machines are periodically backed up to an external Network File System (NFS) storage located on redundant array of disks (RAID). The uninterruptable power supplies are constantly monitored by using Network UPS Tools (NUT) [4] to ensure proper VMs shutdown on low charge.

The TANGO database is essential for the NICA control system operation. This is a MySQL database that runs in Linux container on fast local SSD for performance reasons. The high availability of the TANGO database is achieved by means of MySQL replication (Fig. 2). Semi-synchronous MySQL replication with two master MySQL servers with fully symmetrical configuration is deployed. Each master MySQL server is a slave of the other

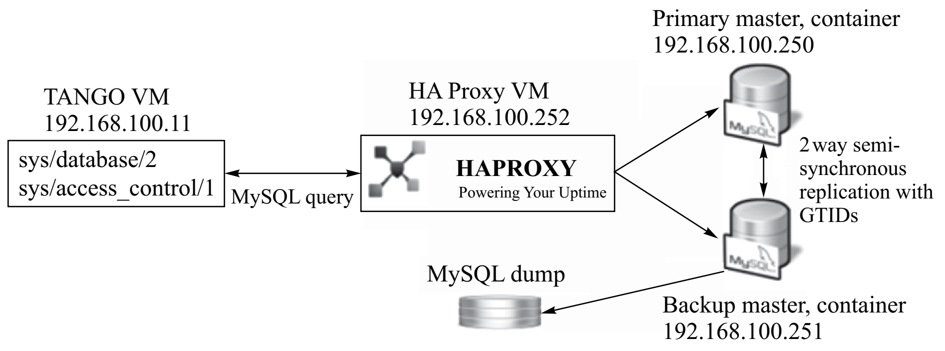


Fig. 2. TANGO database high-availability solution

MySQL server, but only one of them is getting external connections to avoid possible conflicts. A proxy solution for the TCP-based applications HAProxy [5] is used to periodically check health of MySQL servers and route external MySQL traffic to primary or backup MySQL backend server. HAProxy runs on high-available VM to ensure its permanent availability. Periodic dump of the TANGO database from MySQL slave server is used as an additional measure to provide hourly TANGO database backups for two weeks of operation.

The storage space scalability is provided by Ceph which retains or improves its characteristics when adding more disks and more nodes.

Storage space and throughput, CPU cores number and amount of RAM can be increased by adding more nodes to the control system cluster. As a result, more data and VM images can be stored and more virtual machines can be run. The only drawback is a short burst of the network traffic during data rebalancing to new storage daemons.

CONTROL SYSTEM ADMINISTRATION

The TANGO device servers are running on many distributed computers, hence the ability to control all the control system components remotely is necessary. TANGO provides a special service Starter which is running on all control system computers. Its main task is to start all necessary TANGO device servers in proper order at the system startup and keep them running.

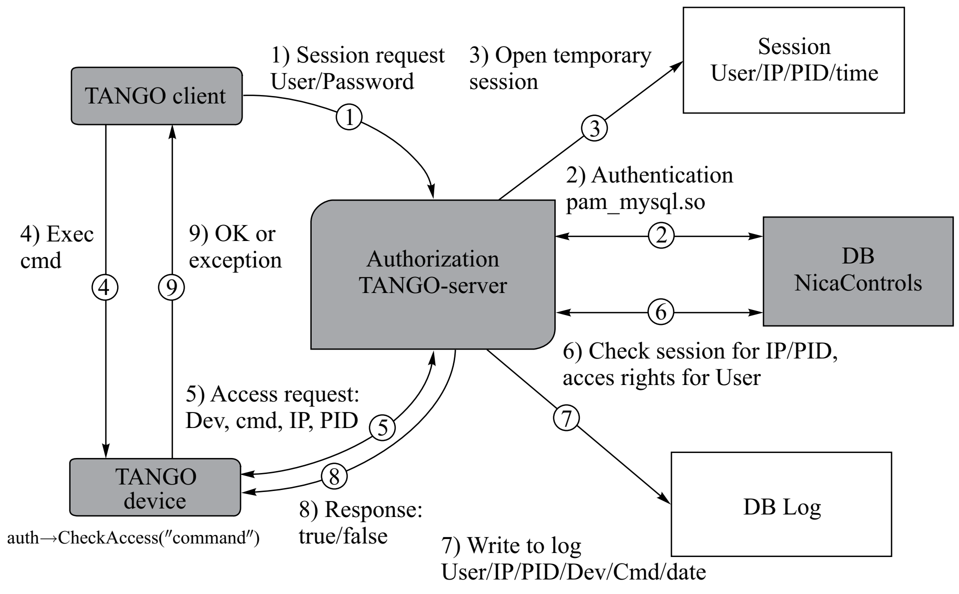
The TANGO manager Astor is used to visualize the state of entire control system and provide tools to start or stop the TANGO device servers remotely, access their logs and settings in the TANGO database. An operator can see if computers and the TANGO device servers are running or not. However, a computer can have performance problems or the TANGO device can be in fault state which cannot be seen using Astor.

An additional TANGO monitoring system which consists of two parts was developed. The first part is a TANGO device to monitor computer resources. It runs on each computer of the control system and periodically collects information about the disk, processor and memory usage. The other part is a TANGO device intended for monitoring states of every TANGO device of certain subsystem. The monitoring client application represents states and statuses of all the TANGO devices and computers of the control system. Alerts and notifications are sent to developers in case of problems.

Apart from the TANGO monitoring, monitoring solution Zabbix [6] is used to monitor the network traffic, hypervisors resources, Proxmox cluster, UPS devices, the TANGO database performance, Ceph cluster performance and health, local and shared storage space.

Another important aspect of the control system operation is access control. We want to restrict access to control system software for certain computers and users and give them corresponding rights to perform specific tasks. The network access limitation, including private subnetworks for subsystems and proper firewalls configuration, is one of the possible measures. Another possibility is the TANGO AccessControl service that restricts connections to the TANGO devices to certain users connecting from certain computers. The access checks are performed on the client side before opening connection to the TANGO device.

An additional server-side authorization service was developed to provide more secure access control and more flexible rights restrictions. The server-side authorization algorithm is shown in Fig. 3. The service uses an additional authorization TANGO device. The authorization TANGO device accepts username/password pair from client application and authenticates the user using Linux Pluggable Authentication Modules (PAM) and information from the NicaControls database. If authentication is successful, a temporary session is opened for the specific user, IP address and process ID. The TANGO device which needs to check the user's rights makes request to authorization server from a command execution method with the TANGO device name, command name, client IP and process ID. After that, authorization server checks the IP and process ID pair against opened sessions and checks user rights from the database. The authorization device logs all the access information into the database and return "true" value if the authenticated user is allowed to access the requested command or attribute. It is possible to precisely tune the access rights using MySQL regular expressions. Operator-expert rights separation can also be implemented.



1) Session request User/Password
3) Open temporary session
Session User/IP/PID/time
TANGO client
2) Authentication pam_mysql.so
4) Exec cmd
9) OK or exception
Authorization TANGO-server
DB NicaControls
6) Check session for IP/PID, acces rights for User
5) Access request: Dev, cmd, IP, PID
TANGO device
8) Response: true/false
DB Log
auth→CheckAccess("command")
7) Write to log User/IP/PID/Dev/Cmd/date

Fig. 3. Server-side TANGO authorization

CONTROL SYSTEM MANAGEMENT

The NICA complex consists of a few large facilities, such as accelerators and transfer lines. Each of them has a number of subsystems. Every subsystem consists of large numbers of equipment, TANGO drivers, cables, computers, network equipment which are developed and maintained by certain people.

The NicaControls [7] database was developed to store this information and describe relations between components of control system hardware and software. The top hierarchy is a facility (accelerator or transfer line), which consists of subsystems (RF, vacuum, beam diagnostics, etc.). Each subsystem consists of devices (equipment). Devices are installed in racks, crates or other containers. They are managed by TANGO device drivers and connected to other devices by cables. TANGO devices are produced by TANGO servers, which contain programming classes and are running on certain computers. Every TANGO class is developed by some person and has source code and documentation. Those relations allow one to find documentation for hardware or software, links to source code and executable files, developers names, racks locations, IP addresses of computers and other useful information.

Information in the database is being added during the control system development. The information from the NicaControls database is used by other control system components — TANGO monitoring system and server-side access control.

CONCLUSIONS

Design of the NICA control system infrastructure was presented. It provides rapid development of hardware and software using the TANGO Controls and National Instruments equipment, reliable and scalable operation by using virtualization and the Ceph storage cluster, comprehensive administration and monitoring, efficient management of the control system equipment and software by using the NicaControls database.

Several Nuclotron subsystems and prototypes for the future NICA accelerators were developed using the described approach.

REFERENCES

1. *Trubnikov G. V. et al.* NICA Project at JINR // Proc. of IPAC2013, Shanghai, China, May 2013.
2. TANGO Controls. http://www.tango-controls.org.
3. Ceph. http://www.ceph.com.
4. Network UPS Tools. http://www.networkupstools.org.
5. HAProxy. http://www.haproxy.org.
6. Zabbix. http://www.zabbix.com.
7. *Gorbachev E., Sedykh G.* The Equipment Database for the Control System of the NICA Accelerator Complex // Proc. of ICALEPCS2013, San Francisco, CA, USA, 2013. P. 1111–1113.