SOFTWARE AND COMPUTATIONAL INFRASTRUCTURE OF LIA-20 CONTROL SYSTEM

A. Senchenko[#], G. Fatkin, S. Serednyakov, BINP SB RAS and Novosibirsk State University, Novosibirsk, Russia
P. Selivanov, BINP SB RAS, Novosibirsk, Russia

Abstract

The linear induction accelerator LIA-20 for radiography is currently under construction at Budker Institute of Nuclear Physics.

This paper presents software architecture and computational infrastructure for the accelerator controls. System and application software are described. Linux operating system is used on PC and embedded controllers. Application software is based on TANGO. Overall data transfer rate estimations are provided.

LIA-20 PROJECT

Linear Inductor Accelerator LIA-20 is designed to provide three electron bunches with energy up to 20 MeV, current up to 2 kA and lateral size after focusing on the target less than 1 mm. It is planned to provide three consecutive bunches, with one of them divided into 9 angles. The accelerator will be used for the flash X-Ray radiography.

LIA-20 consists of the injector, 30 "short" accelerating modules (SAM) and 12 "long" ones (LAM). Injector generates beam with the energy up to 2 MeV. SAM increases the energy by 0.33 MeV and LAM increases the energy by 0.66 MeV. The total length is about 75 meters. Structure is described in detail in [1].

Control units are placed along the installation. All units are based on uniform VME crate and connected via Ethernet.

DATA RATES

All channels could be divided into following groups:

Fast. All measurements faster than 10 us: voltage on inductor, currents on lenses and beam position monitor.

Slow. This group includes measurements with duration up to several milliseconds (charging device, degaussing current).

Timing. These channels provide all devices with proper start pulse.

Interlock. These channels belong to subsystem that prohibit experiment in case of component malfunction or failure.

Technological controls. This group incorporates vacuum controls, optical system alignment [4], control of power supplies.

First four groups are bound to machine cycles, while the last one is continuous.

Tables 1 present the summary of channels and data rates. Estimations are provided for one-bunch cycle.

COMPUTATIONAL INFRASTRUCTURE

Computational infrastructure components are distribute across two areas: control room and experimental hall (see Fig.1). Control rack and operators PC are located in control room. Control rack is equipped with two server, UPS and 24 port switch. The switch provides connectivity between server, UPS, operator's PC and experimental hall switch. All VME crates are located in experimental hall and connected via two switches. Description of components are provided below.

Server:
- CPU 2.0 GHz, Cores 4
- Intel x86-64
- RAM 32GB
- Gigabit Ethernet
- 4TB SCSI
- RAID 5

VME Crate Controller:
- PowerPC - based
- Diskless network boot

Operator's PC:
- CPU 2.2, Cores 4
- Intel x86-64
- RAM 4GB
- Up to 4 monitors

Table 1: Data Rates Estimation

Channel type	Number of channels		Data rate	
	whole system	per VME crate	whole system	per VME crate
Fast	594	22	5.7 MB/cycle	214 KB/cycle
Slow	1485	55	13.5 MB/cycle	0.5 MB/cycle
Timing	1485	55	13.5 KB/cycle	0.5 KB/cycle
Interlock	1485	55	13.5 KB/cycle	0.5 KB/cycle
Technological control	1000	~40	513 KB/min	19 KB/min
	6000	~280	19.3 MB/cycle + 540 KB/min	3.5 MB/cycle + 19.5 KB/min

[#]a.i.senchenko@inp.nsk.su

ISBN 978-3-95450-181-6

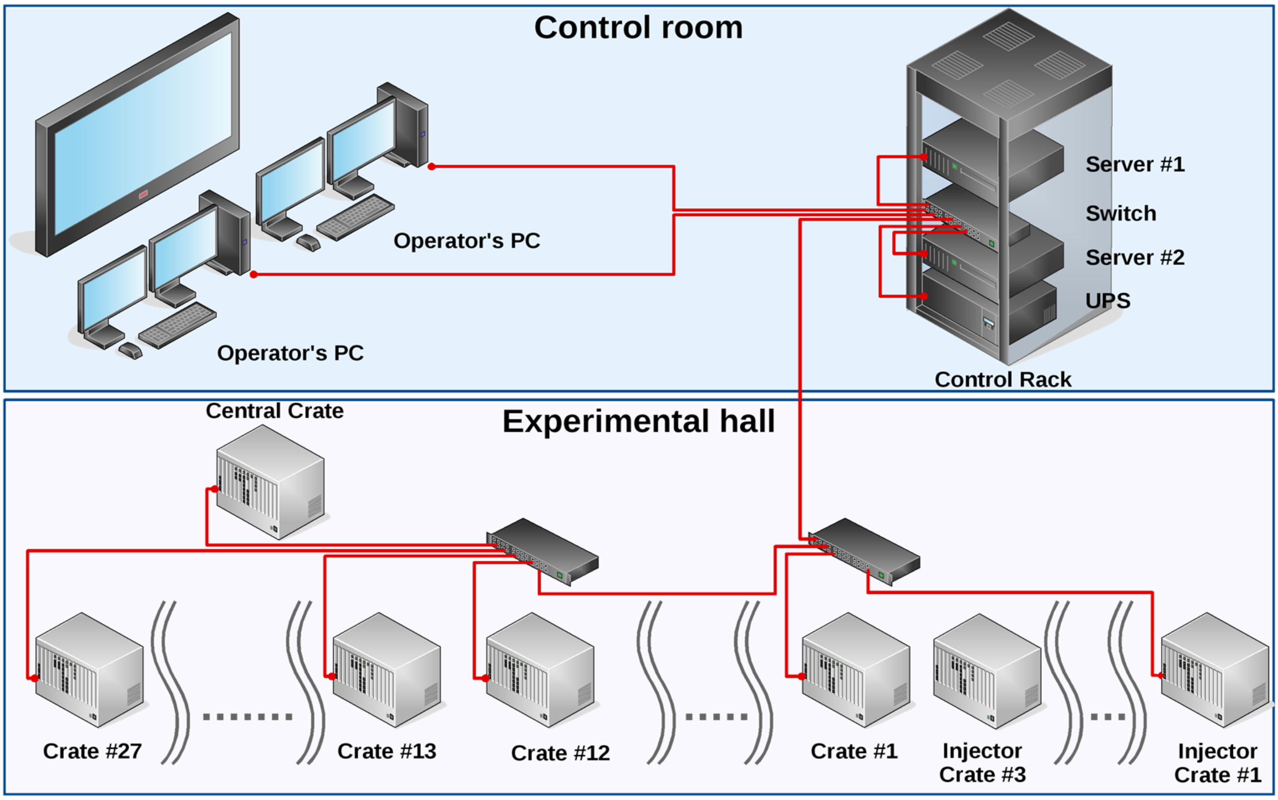

Figure 1: LIA-20 computational infrastructure layout.

SYSTEM AND APPLICATION SOFTWARE

System Software

Experience of BINP facilities shows that use of isolated virtual environment facilitate maintenance while keeping resources overhead at acceptable level. It was decided to provide dedicated virtual machines(VMs) for different services and subsystems.

VM#1 is used for network boot. Kernel image is served via TFTP. Controller' file systems exported on NFS.

VM#2 hosts Tango(see below) specific services: TANGODB and Sardana.

Additional VM could be easily provided on demand.

It is necessary to reduce number of upgrades during development phase and to freeze changes before commission. That why it seems rational to use OS distribution with long term support (LTS).

Server:
Ubuntu LTS
VM:
Ubuntu LTS
Operator's PC:
Ubuntu LTS
Controller:
Debian

Application Software

Experience of LIA-2[2] shows that use of widely-used control system software could reduce costs. Taking into account that LIA-20 is more sophisticated than LIA-2 it could crucial.

After some studies it was decided to use TANGO controls. A lot tools (like rapid UI prototyping, Archive service, macros) are available out-of-the-box or in the form external library.

Application software consists of following layers:

IO abstraction layer. The bottom layer. There are libraries that provide access to CANBus, VME and hide implementation details.

Device Driver layer. This layer is a set of libraries (user-space drivers) that implement interaction with particular device and facilitate re-use.

Low-Level TangoDevices. This layer consists of TangoDevices that wrap Device Drivers and expose them to Tango. They are arranged in tango servers by underlying bus type.

High-level TANGO devices. The topmost layer. Tango Devices and user applications that control subsystems rather than particular device.

There are two types of user application: “engineer” and “operator's”. The first one provides access to “raw” tango-device. It is designed for developing and testing purposes. The second one is high-level application that interacts with multiple tango-devices and hides implementation details. The software scheme is illustrated in Fig. 2.

ISBN 978-3-95450-181-6

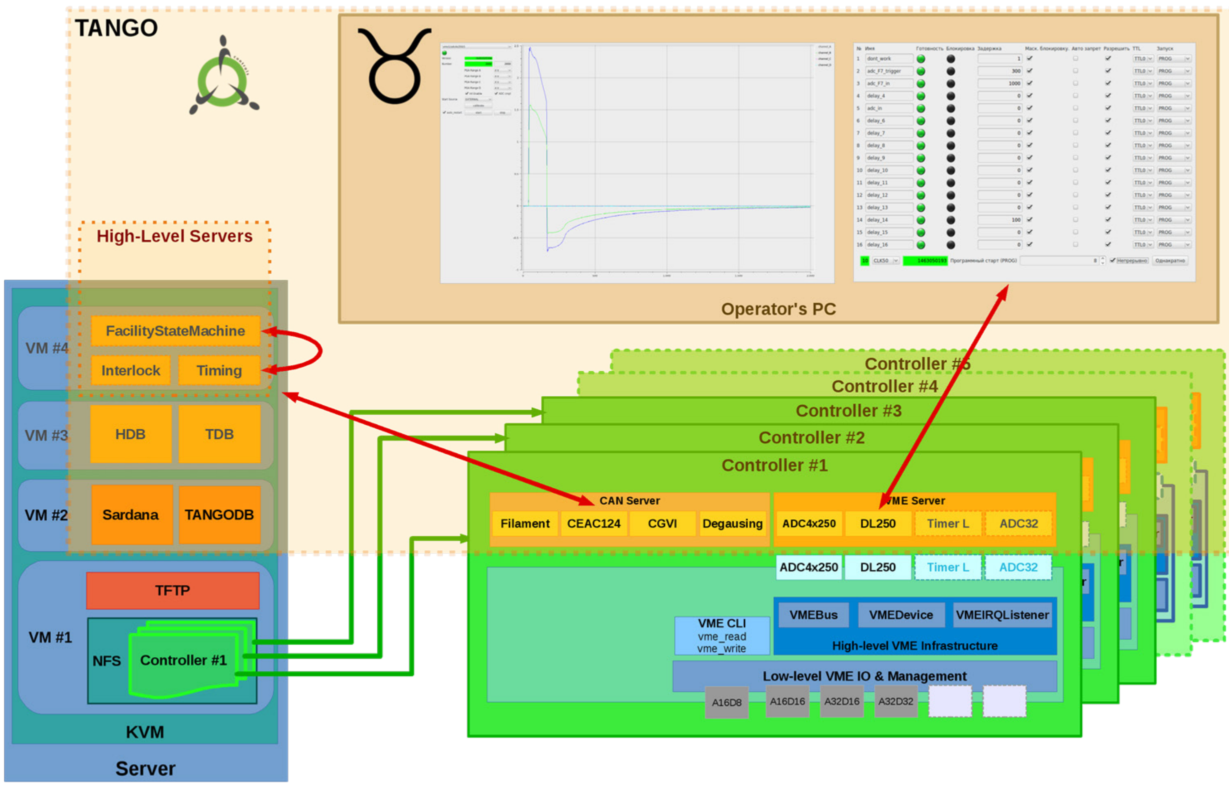

Figure 2: LIA-20 software scheme.

To date, tango-devices (with underlying drivers) and engineering user application are provided for ADC4x250VME [3], DL-250VME, and for majority of CANBus devices.

CONCLUSION

Next step is assembling of LIA-5 – the 5 MeV linear accelerator with all control system elements. It requires to not only minimal viable software but high-level tango-devices also. Other important tasks arise – infrastructure management software (cabling database), machine modelling software and integration of proven LIA-2 data analysis system [5].

REFERENCES

[1] G. Fatkin et al. “Structure and hardware of LIA-20 control system” THCBSH02, these proceedings.

[2] Bak P. A. et al., “Control system of a linear induction accelerator of an X-ray complex: Structure, hardware, and test performance“, Optoelecron. Instrument. Proc., 2011, Vol. 47, No. 3, pp. 303-312.

[3] E. Kotov et al. “VME Based Digitizers for Waveform Monitoring System of Linear Induction Accelerator (LIA-20)”, THPSC081, these proceedings.

[4] A. Chupyra and E. Bekhtenev “System of Geodetic Measurements for LIA-20” THPSC083, these proceedings.

[5] Data analysis system of a radiographic complex based on a LIA-2 accelerator Fatkin, G.A. 2015 Optoelectronics, Instrumentation and Data Processing.

ISBN 978-3-95450-181-6