Software and Computational Infrastructure of LIA-20 Control System

A. Senchenko, G. Fatkin, S. Serednyakov (1,2), P. Selivanov (1) 1 - BINP SB RAS, 2 - NSU

Abstract

The linear induction accelerator LIA20 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.

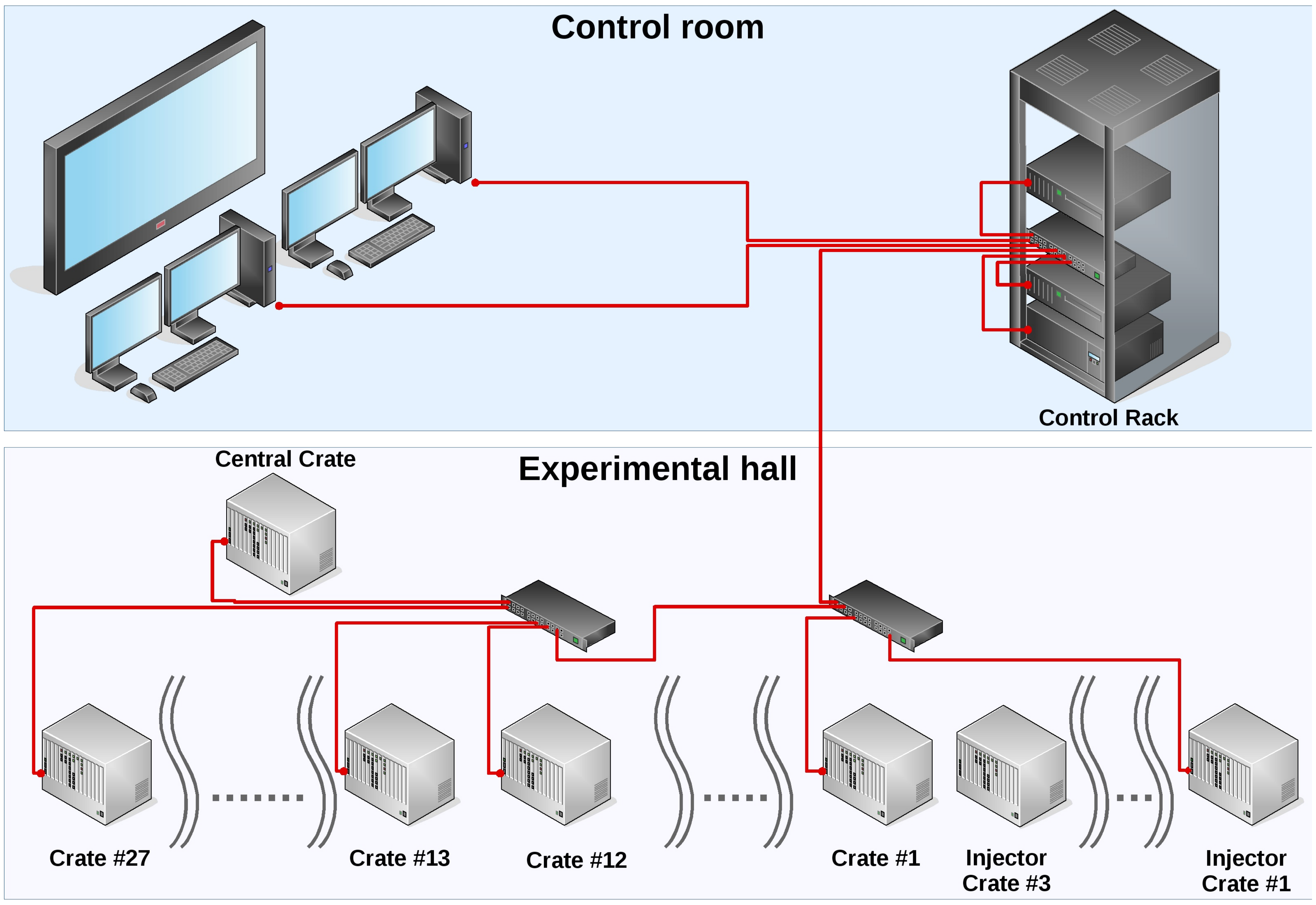
Regimes

1. Cycle interval 10-15 sec (4-6 cycles/min)
2. 1/2/3 - bunches per cycle
3. Test regimes

Data Rates

Channel type	Number of channels		Data rate (1-bunch)	
	whole system	per VME crate	whole system	per VME crate
Fast (shorter than 10 us) U,I of inductors	594	22	5.7 MB/cycle	214 KB/cycle
Slow Degaussing, lense PS	1485	55	13.5 MB/cycle	0.5 MB/cycle
Timing Delay Lines, Timers	1485	55	13.5 KB/cycle	0.5 KB/cycle
Interlocks	1485	55	13.5 KB/cycle	0.5 KB/cycle
Technological controls Temperature, vaccuum	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

Network



Computational infrastructure

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

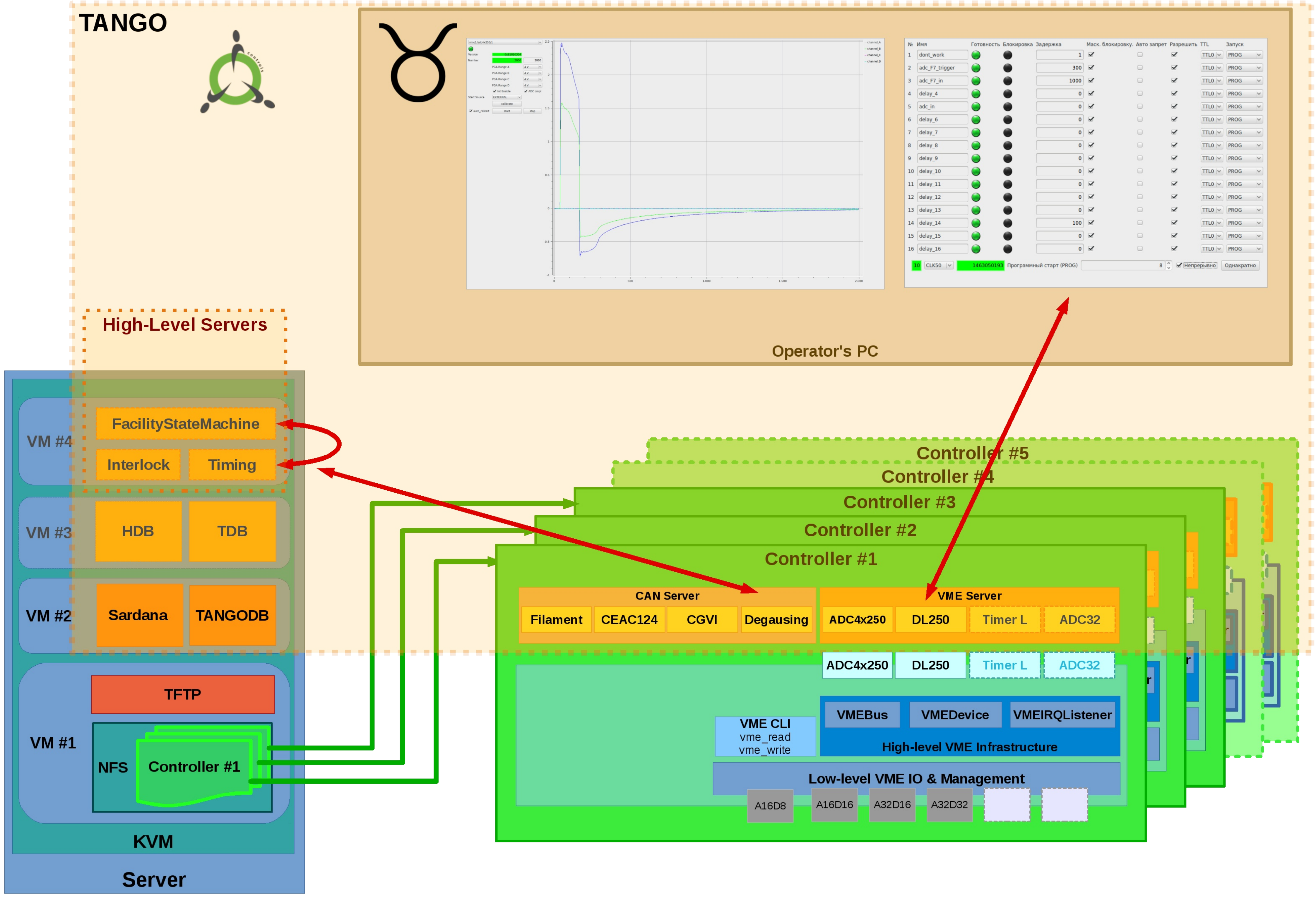
VME Crate Controller:
PowerPC - based
Diskless network boot

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

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

Network:
3 * 24 port switches

System and Application Software



System software

The use of virtualization facilitates maintenance operations and provides dedicated OS for group of services.

VM#1 is used for network boot. Kernel image is served via TFTP. Root file system of controllers are exported via NFS.

VM#2 contains Tango specific servicers: TANGODB and Sardana.

Application software

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 (userspace 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. Timing server will provide proper delays and start times to underlying Timer-L and DL250.