

CONSOLE SYSTEM USING THIN CLIENT FOR THE J-PARC ACCELERATORS

Susumu Yoshida*, Makoto Takagi, Shigenobu Motohashi, Takao Iituka
Kanto Information Service (KIS) 8-21 Bunkyouchou Tsuchiura, Ibaraki, 300-0045

Norihiko Kamikubota, Noboru Yamamoto, Tadahiko Kato, Hidetoshi Nakagawa,
Jun-ichi Odagiri, Takahiro Matumoto
High Energy Accelerator Research Organization (KEK) 1-1 Oho, Tsukuba, Ibaraki, 305-0801

Abstract

We are interested in using commercial Thin Client as accelerator console terminals in J-PARC. We configured Thin Client system to work as Linux-based workstations with X-server. Evaluation of Thin Client has been carried out during J-PARC Linac commissioning and software development.

INTRODUCTION

In the accelerator controls, a console system is one of important components. Recently the J-PARC accelerator complex started beam commissioning [1]. The present console terminals for J-PARC Linac are standard Personal Computers (PCs) with Linux Operating System (OS). As the number of PCs increases, maintenance works for them, both software configurations and hardware setups, become more time-consuming and troublesome. In the near future, we expect nearly 100 displays for operation, maintenance, and software development in J-PARC. Thus, we need some ideas to simplify maintenance works of console terminals.

Taking into account discussion in early studies of console system [2], we have interested in commercial Thin Client (TC) systems. In general, TC is used with a remote server, where all the applications and file managements are executed. TC is designed to receive screen-copy of a remote server in real time. Thus, TC hardware is customized only in man-machine interfaces (screen display, keyboard, mouse, and so on). Nowadays in the commercial market, TC systems are understood to guarantee higher security-level, and have been provided by many companies.

Our idea is to use TC terminals as accelerator consoles in J-PARC. After booting from a boot-server, a TC works as a Linux-based workstation with X-server. Management of configurations is made at a boot-server side. Moreover, we use single configuration for all of the TC terminals, thus, maintenance becomes much easier than that with multiple PCs.

Since TC is disk-less and fan-less, we expect longer life-cycle and easier hardware maintenance than standard PCs over several years. In addition, we could suppress total cost to purchase console terminals.

We introduced a TC system and used both for software development and for J-PARC Linac beam commissioning.

*e-Mail: <syoshida@post.kek.jp>

In this article, we report current usage status. Experienced problems and solutions are also given.

MODEL SELECTION

In the past years, we discussed some companies feasibilities to run Linux OS in commercial TC terminals. Among possible candidates, we selected HP Compaq t5720, since it is the only model to allow an extension PCI card. General specifications of t5720 are shown in Table 1.

We also selected Matrox Epic-A, a multi-display card for a PCI slot. Matrox supports Linux drivers for Epic-A. As a result, each of TC terminal has 2 (or 4) displays (see Figure 2 and Figure 3). This feature is preferable for us, since we can decrease total number of TC main bodies by less than half.

Table 1: Specifications of the Thin Client.

HP Compaq t5720	
Processor Type	AMD Geode NX-1500
Processor Speed	1GHz
Amount of flash memory	512MB
Memory size	512MB or 1GB

NETWORK-BOOT CONFIGURATIONS

Network-Boot Method

Network-boot is a popular method to download firmware from a server into network devices. The same method can be used to distribute customized OS images (for example, Windows CE, Linux, and others) into many PCs. In order to utilize network-boot method, a PC must have PXE (Preboot eXecution Environment) feature. Recently most of commercial products with a network port have this PXE feature.

We decided to use network-boot technique to centralized management of customized Linux OS for TC terminals. We configured and started necessary network services (DHCP, TFTP, and NFS) at a file server to enable network-boot feature. At each TC terminal, PXE configuration was made to use network-boot at its start-up time.

We decided to follow so-called "NFS-root" style: (a) to make download images (Linux kernel and initrd which is needed for booting) as small as possible, (b) after booting,

a TC mounts root-disk areas (/etc, /usr, /bin, etc.) of remote server using NFS. Thus, all the TC terminals share the same root files. The procedures how a TC terminal starts up in our environment are summarized and shown in Figure 1.

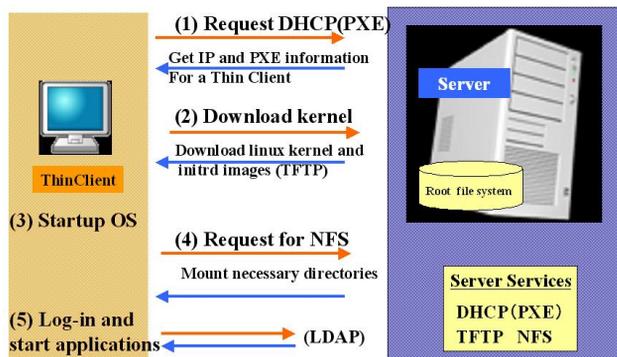


Figure 1: Start-up of TC with network booting.

Making Root-Disk Images

In order to make root-disk areas for TC terminals, we introduced “LiveCD” technique. The LiveCD is a general name of a bootable CD/DVD, which can start a PC as a disk-less client. We followed descriptions on a tutorial web site [3], where our target OS (Scientific Linux 4.4, the main OS for J-PARC controls) is supported.

The preparation procedure is easy. We made copies of necessary files: kernel and initrd into a predefined directory of the boot-server. Then, we customized it to add our site-specific control components, such as desk-top environment (XFCE) and printer services.

EXPERIENCED PROBLEMS

Successful Result with Software Development

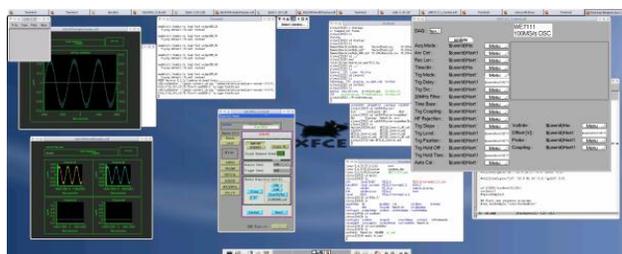


Figure 2: Software Development with a 2-display TC

During 2006-2007, we introduced 19 TC terminals with 2 (or 4) displays each. Firstly, they have been used for software development by control staff members, and showed good enough reliability. Figure 2 shows a demonstrative screenshot of a 2-display TC terminal. Some of them have been used also for simple status-display of J-PARC accelerators successfully.

Problems Found in Linac Commissioning

As a next step, we intended to use TC terminals in the J-PARC Linac commissioning. It means they were used in real operational activities of an accelerator by non-control

members (i.e. physicists). Two TC terminals were installed and used together with standard PCs. Figure 3 shows one of the TC terminals in the field at J-PARC Linac Klystron Gallery. The result was serious: both TC terminals caused fatal stops every night.

After some investigations, we found that some of Java applications, which communicated accelerator devices frequently using EPICS Channel Access (CA) protocol, contain memory-leak. In a typical case, a TC terminal stops 5 hours after it’s booting, which corresponds that a TC spent all the available memory.

In the middle of 2007, we updated libraries related to Java-CA communication to the latest versions [4]. The updates showed an improvement in memory consumption rate, but the problem still remained.

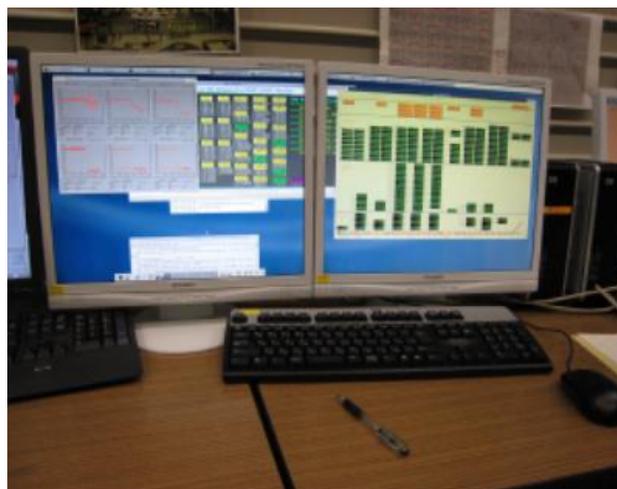


Figure 3: A TC terminal in Linac Commissioning.

Another problem was found in waveform-monitoring applications. During early beam studies at a repetition rate 1-5 Hz, monitoring waveforms was satisfactory. Later as the number of applications increased, slow responses over a whole TC terminal became serious. The 100 % CPU-loading was observed. It was understood that the CPU power of TC is not enough to run several applications at the same time.

Countermeasures Against Problems

Possible countermeasures against two problems were discussed. For the 1st problem, we keep to survey related source codes where memory-leaks implemented. In addition, a trial to add an external USB-disk as a Linux SWAP area is in progress.

For both problems, a practical solution to avoid fatal stops is that (1) prepare other standard PCs for running applications, then (2) use a TC terminal as a simple X-terminal (i.e. not to run troublesome applications on it).

SUMMARY

We introduced commercial Thin Client terminals as an accelerator console system for J-PARC. We configured a Thin Client terminal to be a Linux-based workstation with X-server. Each terminal is equipped with 2 (or 4) displays.

We developed easy procedures to maintain Thin Client terminals at a server-side.

During 2006-2007, 19 Thin Client terminals were introduced. They were used for software development successfully. Some of them were used in J-PARC Linac commissioning, and showed two problems: (a) fatal stops caused by memory-leak applications, and (b) slow responses by CPU-intense applications such as waveform monitoring. Possible countermeasures against problems were discussed.

We would like to thank the J-PARC Linac commissioning members, especially Dr. A. Ueno and Dr. M. Ikegami, for their help at J-PARC Linac Klystron Gallery.

REFERENCES

- [1] K.Hasegawa, *et al.*, “Commissioning of the J-PARC LINAC”, Proc. PAC2007, p.2619-2623
see also URL: <http://j-parc.jp/Acc/en/index.html>
- [2] N.Kamikubota, *et al.*, “Study of PC-based Console for the J-PARC Control System”, Proc. PCaPAC2005
- [3] “Scientific Linux Live CD and DVD - Diskless Client”, URL: <http://linux.web.psi.ch/livecd/>
- [4] M.Sekoranja, “Java CA Client Library - Update”, EPICS Collaboration Meeting in DESY-Hamburg, April 2007