Database System for ATF Control

Nobuhiro TERUNUMA, Timo KORHONEN and Junji URAKAWA
Accelerator Laboratory, High Energy Accelerator Organization, KEK
Oho 1-1, Tsukuba, Ibaraki, 305 Japan

Abstract

Control system for the ATF accelerators has been developed by using two CAMAC networks and the process-control-database system running on the VMS cluster. We succeeded in creating the control system in a limited time and staff with the help of this database application. In this paper, we describe the database configurations.

1 Introduction

The accelerator test facility (ATF) in KEK has a 1.54 GeV S-band linac and a damping ring [1]. Aim of this facility is to study the low-emittance-beam phenomena and to develop technologies for a future linear collider [2]. Studies to understand the behavior of the hardware devices and that of the beam itself have been performed since January 1997, the start of the beam commissioning of the damping ring.

Control system is based on CAMAC serial-highway networks handled by the Open-VMS Cluster system. Control software is integrated by using a commercial database application, Vsystem [3] and with our developed device-driver software. This database application is based on the mechanism of distributing databases across the networks and has a graphical user interfaces on the X-window system. Recently, we change the network configurations and operating system to improve the global performance of our control system.

2 System Configurations

2.1 Hardware

The basic system configuration is shown in Figure 1. We use two VAX servers to control the CAMAC networks. One server controls the CAMAC system for the linac and the other controls that for damping ring. Each CAMAC network consists of the enhanced serial highway driver (2160-Z1F) [4] and many CAMAC crates (up to 64) located near the hardware devices. This highway has a capability of 5 Mbytes/sec of data transmission. Main part of this network was connected with optical fibers to avoid an irregular noise from power supplies. Most of the control tasks are running on the Alpha server, because it has the highest CPU performance. Six VAX stations are used as the user terminals for the accelerator control and for the software development. These machines are connected with a switching HUB (DEC VN900EF) and they make a cluster system running on the Open-VMS V7.1.

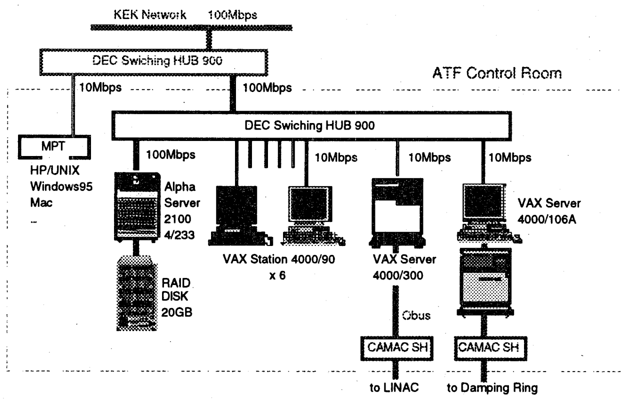


Fig. 1 Cluster configurations in the ATF control system.

2.2 Software Structure

Vsystem feature

The Vsystem was used in the core part of the control system as shown in Figure 2. It could easily include many device libraries written by ourselves. Since the complex part of the data handling in the cluster such as the signaling of database is supported by Vsystem itself, we can concentrate to develop the accelerator control parts such as device drivers and the analysis routines. Vsystem has many tools convenient to the accelerator control such as automatic alarm monitoring, channel data logging and so on. One of them, VDRAW is a very convenient display-and-control tool based on the Motif. It has control buttons, many graph options, various display modes and so on. We made operator interfaces using this tool.

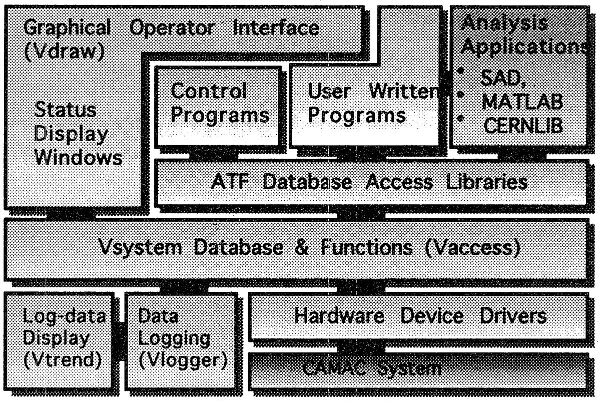


Fig. 2 Software configurations. Vdraw, Vaccess, Vtrend and Vlogger are Vsystem components.

Database Feature

We can define multiple databases individually on local and remote node in the cluster. This network-distributing mechanism is supported by Vsystem. The identification of the databases is done by defining the logicals based on the VMS system service. For databases on a local node, logicals should be defined as

define <db_name> <database object filename>,

while for the remote databases, filename should be changed as <node>::"task=vserver". Vserver is one of the Vsystem component to access a remote node by using the DECNET objects. The database logical name <db_name> is transparent in the cluster so that the access method to the database is same both in local and remote node.

A basic element of a database is called "channel" and it should be defined one of the various data types; integer, real, binary, strings and arrays for integer and real. Channel has a mechanism to execute some actions automatically when the channel access has occurred. These actions are defined by user written functions that are called "handler". For example, a read access to a channel that shows the hardware status executes the CAMAC action of the target hardware. The index name used to access a channel is defined with its database name as <db_name>::<channel_name>. Table 1 shows the examples to change the magnet current and to read the monitored current. In this case, the magnet is ZV1R and the database logical name is db2_magnet.

Table 1
Typical functions to access the channel

```
vatf_rput("db2_magnet::zv1r:current.write", current)
vatf_rget("db2_magnet::zv1r:current.monitor", current)
```

3 ATF Database Structure

3.1 Database format

In the cluster system, we use a lot of logicals to simplify the accesses for other data. Therefore we set a prefix to the database logical name; DB_ for the Alpha, DB1_ for the linac and DB2_ for the ring. For the channels, they are named with the hardware-device name as <device_name>:<generic_type>, where the generic type describes the data element of the hardware. Table 2 shows the definitions of the channels to build the binary objects for the magnet database.

3.2 Device Access

There are two ways for writing and reading the CAMAC on a remote server from a local machine. One is a direct CAMAC function call that connected to a network process and the other is a Vserver method as described before.

At the first commissioning of the linac, we needed to have a CAMAC access method from Alpha server. We prepared the network-CAMAC processes between Alpha and VAX servers by using the system services [5]. To keep the compatibility between Alpha and VAX servers, the format of the CAMAC functions is unified as that of the IEEE CAMAC standard[6]. In this method, we identify the target server by specifying the CAMAC branch number. Device libraries such as the early developed linac part use these IEEE functions. This method is quite simple and convenient, but this leads to heavy traffic on the network.

Table 2
Example of channel definitions for the magnet

```
$ZV1R:SWITCH
    integer value 0
    branch 1 crate 26 slot 2 subaddress 0 hparam5 4
     out automatic handler H_KIKUSUI5A_CTRL
    label "Control switch for ON/OFF/RESET"
$ZV1R:STATUS.READ
    integer value 1
    branch 1 crate 26 slot 8 subaddress 0 hparam5 8
    in  automatic handler H_KIKUSUI5A_STATUS
    label "PS status bits"
$ZV1R:CURRENT.MONITOR
   real format "(f5.2)" value 0.0
    branch 1 crate 26 slot 16 subaddress 10 hparam5 0
   hparam6 1
     in automatic handler H_KIKUSUI5A_ADC
    label "monitored current by ADC"
$ZV1R:CURRENT.READ
   real format "(f5.2)" value 0.0
    branch 1 crate 26 slot 13 subaddress 5 hparam5 0
     in automatic handler H_KIKUSUI5A_DATA
    hfunction HWREAD
    label "setting current stored in DAC"
$ZV1R:CURRENT.WRITE
   real format "(f6.2)" value 0.0
    branch 1 crate 26 slot 13 subaddress 5 hparam5 0
     out automatic handler H_KIKUSUI5A_DATA
    hfunction HWWRITE
    label "setting current to be stored in DAC"
```

An average time to execute a CAMAC single action from Alpha to VAX server 4000/300 for linac is 5 ms. For VAX server 4000/106A for damping ring, it is only 1 ms. The access speed in the remote VAX is 2 ms and 0.5 ms. In this situation, the performance of the CAMAC access is limited by the network overhead. Therefore, we prepared a function that executes many single actions in a network call.

Handler method

Another method is a channel access on a remote database by using the Vserver with the handler mechanism as described before. Databases and related access routines are exist in the remote machines. Further, Vsystem has a good access method for the remote databases called 'list mode'. This accesses multiple channels in a function call. Results of the list-mode performance measured in the cluster is summarized in Table 3. The channel accesses were performed from the Alpha server to remote the VAX server, and they were done during the beam experiments. Also we show the influence of the CAMAC access in the handler.

In ATF, there are many devices with different interface even if the roles of them are same. For example, we have many types of magnet power supplies. This comes from the difference of the manufacturer and historical reasons. To make the channel structure common for the same device group, these differences should be handled in the handler. Therefore we prepared handlers for every device type.

Table 3
Performance of channel action (channels/second).

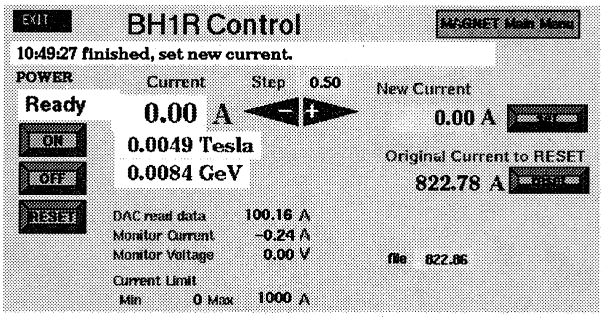
	mode	no CAMAC	with CAMAC
VAX(linac)	normal	60	50
	list	1000	344
VAX(ring)	normal	198	196
	list	5520	1428

Standalone process

Some channels have no "automatic" handler options to avoid the heavy concentration of the hardware reading accesses. For these channels, the background reading processes are prepared. These processes enable the "automatic" options in themselves and then update the channel values. Other accesses of these channels will only get values without executing the handler. We use this method to read the hardware status and the BPM data.

3.3 User interface

Figure 3 shows one of the magnet control windows. Monitored values are displayed with various colors associated with the device status. There are two ways for executing the user written routines from the VDRAW window. One is the "user-link" option and it is a switch to submit a direct command into operating system. The other is a direct channel access from the display windows.



BH1R Control

10:49:27 finished, set new current.

POWER Ready

Current 0.00 A

Step 0.50

0.0049 Tesla

0.0084 GeV

New Current 0.00 A

Original Current to RESET 822.78 A

DAC read data 100.16 A
Monitor Current -0.24 A
Monitor Voltage 0.00 V

file 822.86

Current Limit
Min 0 Max 1000 A

Fig. 3 Magnet control window. Allow switches increase or decrease the setting current.

3.4 Present status

At present, august 1997, we use 28 databases for the accelerator control. They have 12667 channels in total; 4639 integer and 5454 real and other are binary and strings. Largest database is for magnets and it has 7520 channels. The number of channels that use the automatic handler is only 384. We are adding the handler options into more channels to improve the overall performance.

Most of the status readers, background processes, are updating the channel data with specified time intervals; about ten seconds. For BPMs, data update is handled by the trigger distributed from the beam injection signal and now its rate is 0.75 Hz. Graphical plots of the closed orbit on the operator window are updating at this rate. Since the maximum injection rate is 25 Hz, we have capability to operate such rate if the CAMAC operations of BPM are completed under the condition that other requests are running.

4 Summary

The control system for the ATF accelerators has been developed by using the process-control-database system, Vsystem. We succeeded in creating the control system in a limited time and staff with the help of this database application.

We are in the half way to establish the control system in the style that mentioned in this paper. Now we are changing the databases for linac part because they were created in early phase of the ATF construction and they did not use the distributed feature. ATF group plans to install many monitor devices and plan to develop the on-line analysis tools. Although the data handling in the system will become more complicated one, we will be able to optimize the control system by using this database system.

Acknowledgments

We would like to express our thanks to Professor K. Takata for his encouragement. We wish to thank Dr. H. Hayano and Mr. T. Naito for their efforts of making the first linac control system. We also wish to thank other ATF members for their support.

References

[1] F.Hinode, et al., KEK Internal 95-4 (1995).
[2] JLC Design Study, KEK Report 97-1 (1997).
[3] VSYSTEM is a commercial product of Vista-Control Systems, Inc., Los Alamos, New Mexico.
[4] developed by Kinetic Co., Ltd.
[5] T. Korhonen et al., Proc. of the 21th Linear Accelerator Meeting in Japan, Tokyo, p174.
[6] ANSI/IEEE Std 758-1979, IEEE Standard Subroutines for CAMAC.