

MANAGEMENT TOOLS FOR DISTRIBUTED CONTROL SYSTEM IN KSTAR

Sangil Lee, Jinseop Park, Jaesic Hong, Mikyung Park and Sangwon Yun

National Fusion Research Institute, Gwahangno 113, Yuseong-Gu, Daejeon 305-333, KOREA

Abstract

The integrated control system of the Korea Superconducting Tokamak Advanced Research (KSTAR) has been developed with distributed control systems based on Experimental Physics and Industrial Control System (EPICS) middleware. It has the essential role of remote operation, supervising of tokamak device and conducting of plasma experiments without any interruption. Therefore, the availability of the control system directly impacts on the entire device performance. For the non-interrupted operation of the KSTAR control system, we have developed a tool named as Control System Monitoring (CSM) to monitor the resources of EPICS Input/Output Controller (IOC) servers (utilization of memory, cpu, disk, network, user-defined process and system-defined process), the soundness of storage systems (storage utilization, storage status), the status of network switches using Simple Network Management Protocol (SNMP), the network connection status of every local control sever using Internet Control Message Protocol (ICMP), and the operation environment of the main control room and the computer room (temperature, humidity, electricity) in real time. When abnormal conditions or faults are detected by the CSM, it alerts abnormal or fault alarms to operators. Especially, if critical fault related to the data storage occurs, the CSM sends the simple messages to operator's mobile phone. In addition to the CSM, other tools, which are subversion for software version control and vmware for the virtualized IT infrastructure, will be introduced.

INTRODUCTION

EPICS [1] was adopted as a middleware of the KSTAR Integrated Control System (KICS) [2,3] to integrate and control a lot of network-based distributed systems for tokamak operation and plasma experiment. As well known, EPICS is a set of open source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific facilities such as particle accelerators, telescopes and other large scientific experiments. Also, Qt-SDK [4] was used as a tool for developing user interface in KSTAR. The KSTAR Widget Toolkit (KWT) [5,6], which is a name of libraries developed on Qt-SDK, had been considered in the view of the following functions:

- Easiness of development
- Maximum effectiveness for user interface
- Performance of visibility
- Software reusability
- Flexibility for requirements of operators
- Independence from OS Platform

CONTROL SYSTEM MONITORING

The KSTAR control system is composed of a lot of local control systems distributed in the vicinity of the KSTAR to operate tokamak, perform experiments and obtain data. The availability of the control system directly results in success or failure of tokamak operation and plasma experiments whereby the Control System Monitoring (CSM) was developed for the purpose of monitoring and supervising the entire control system and IT infrastructure. The CSM consists of two KWT widgets; "BlinkLine" and "CABlinkLabel". The two widgets among many KSTAR widgets were developed only for use in the CSM. Figure 1 shows the configuration of software module configuration for the CSM.

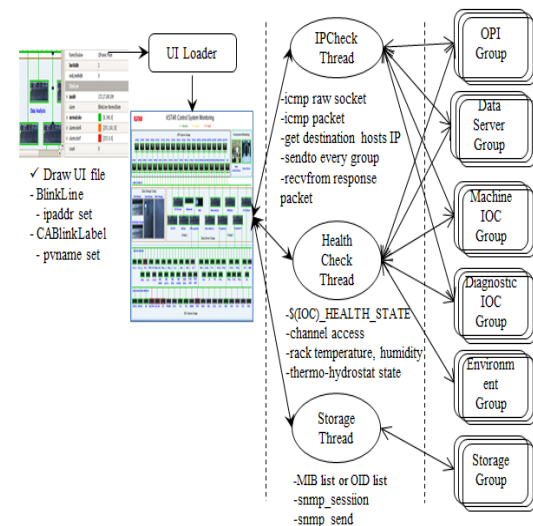


Figure 1: Control System Monitoring software module.

Network Monitoring

"BlinkLine" widget has "ipaddr" property to write IP address for monitoring the network status of IOC server. If "BlinkLine" widget flashes in red, it means that the IOC server has some network problems or has gone into shutdown state. In Figure 1, the "IPCheck Thread" gets IP address from "ipaddr" property of all "BlinkLine" widgets and sends ICMP messages to all host destinations using the "sendto" UDP socket at the same time. The ICMP messages are typically generated for responding to errors in IP datagrams, for diagnosis or routing purposes. In addition, ICMP errors are always reported to the source IP address of the originating datagram. After sending ICMP messages, "IPCheck Thread" receives the response messages using "recvfrom" UDP socket. The request IP list and the response IP list are the same in the normal

state of all networks and all “BlinkLine” widgets display in green. If a problem happens in the connections of some IP addresses, the “BlinkLine” widgets related to those IP addresses will blink in red or orange color.

Server Group Monitoring

The “CABlinkLabel” widget has two alarm states. One is the alarm state blinking in yellow for reporting abnormal status in IOC server’s resource. The other is the alarm state blinking in red for IOC server problem. Left-clicking on the “CABlinkLabel” widget generates a resource dialog window to inform a user about the resource status of the IOC server in detail. All servers have their own resource information in the form of EPICS record as follows:

- CPU Load (Idle, Nice, System, User)
- Memory (Available, Used, Free, Swap)
- Heartbeat Count
- System Information (IP address, Up time, Disk, etc)
- Network Packet (In-bound, Out-bound)

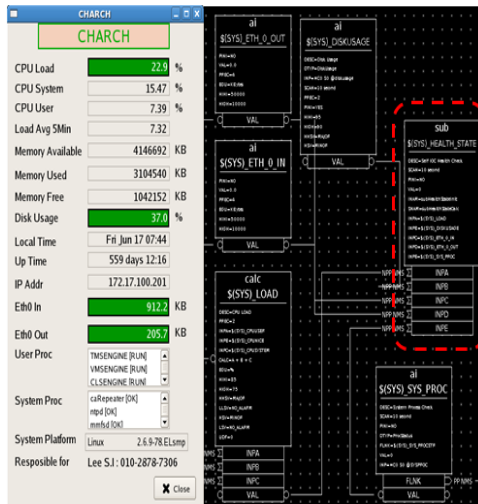


Figure 2: Heath state logic EPICS database.

IOC Health State Logic: \$(IOC)_HEALTH_STATE

- Alarm:
 - CPU Load $\geq 85\%$ or
 - Disk Usage $\geq 95\%$ or
 - Ethernet Packet ≥ 20 Mbytes or
 - User-defined System Process bit count ≥ 1
- Normal:
 - CPU Load $< 85\%$ and
 - Disk Usage $< 95\%$ and
 - Ethernet Packet < 20 Mbytes and
 - User-defined System Process bit count $= 0$

Environment Monitoring

In order to use stable IT resources, it is important to maintain an optimal environment in the control room and the computer room. We have used National Instrument (NI) DAQ modules (cFP-2120, NI cFP-RTD-124 and cFP-DI-304) to monitor the temperature of server racks, the relay signals of thermo-hydrostat and the water leak

sensing signals of High Performance Computer (HPC). As shown in Fig. 2, EPICS channel access can be achieved through Ethernet communication between EPIC IOC driver support module and NI module.

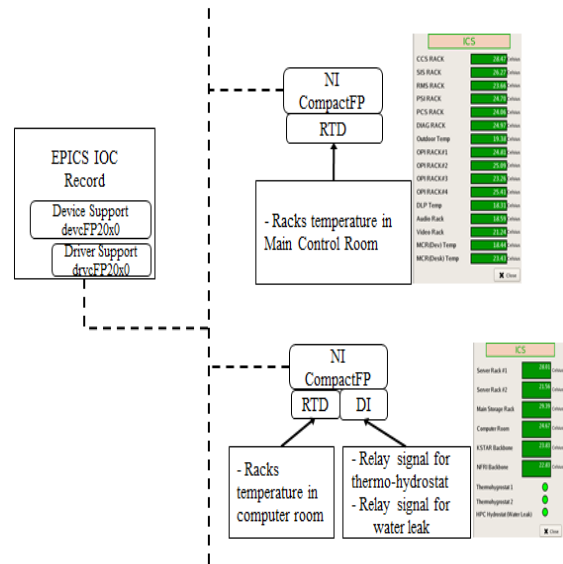


Figure 3: Environment monitoring using NI cFP.

The two “CABlinkLabel” assigned to “Main Control Room” and “Server Room” in Figure 3 blink according to the following alarm logic.

Environment Alarm Logic:

- Main Control Room: ICS_CTRLROOM_TEMP
 - Alarm: 1 rack’s temp among 16 racks’ temp $\geq 32\text{ }^\circ\text{C}$
 - Normal: every rack temp $< 29\text{ }^\circ\text{C}$
- Computer Room: ICS_SERVERROOM
 - Alarm: 1 rack’s temp among 6 racks’ temp $\geq 32\text{ }^\circ\text{C}$ or 1 thermo-hydrostat of 2 thermo-hydrostats == alarm or water leak for HPC cooling line == alarm
 - Normal: every rack temp $< 29\text{ }^\circ\text{C}$ and 1 thermo-hydrostat of 2 thermo-hydrostats == alarm and water leak for HPC cooling line == alarm

Storage System Monitoring

The storage system monitoring has not been fully implemented so far. Some modules were developing using SNMP which is the standard operations and maintenance protocol for the Internet. The SNMP-based management not only produces management solutions for systems, applications, complex devices, and environment control systems, but also provides the Internet management solutions supporting web services. The storage system of KSTAR is also in the form of a management system using SNMP. “Storage Thread” in Figure 1 is being implemented the storage system monitoring module by using the Management Information Base (MIB) that is provided by storage manufacturer. Table 1 shows all IT resources being monitored by the CSM for KSTAR operation.

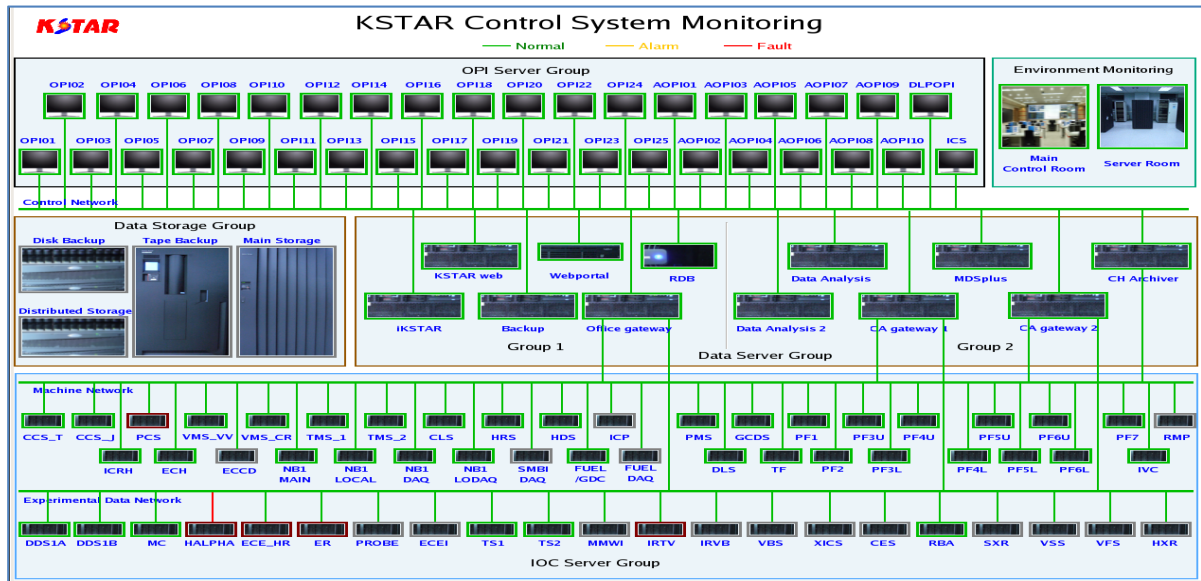


Figure 4: KSTAR Control System Monitoring user interface.

Table 1: IT Resources for KSTAR Operation

Group	Number of server	OS Platform	Network
OPI	35 Servers	Linux	Control
Data	12 Servers	Linux/Window	Control
Plant IOC	37 Servers	Linux/Window	Machine
Diagnostic IOC	21 Servers	Linux/Window	Experiment
Storage	4 Storages		Local
Environment			Control

The Figure 4 shows that the CSM is supervising the entire control system and IT infrastructure for KSTAR operation.

Short Message Service (SMS) System

SMS is a text messaging service component of mobile communication using standardized communication protocols that allow the exchange of short messages between mobile phone devices (<http://en.wikipedia.org/wiki/SMS>). Basically the KSTAR Supervisory Interlock System (SIS) manages all the alarms received from the 13 local interlock subsystems and transmits simple alarm messages to operators in charge of the systems which operate for 24 hours a day, such as vacuum pumping system, cryogenic systems, control systems and so on. When the CSM detects any serious problem in main servers like a system shutdown, it notifies the problem to the SIS at once via hard-wired signals. Next, the SIS transmits the alarm messages to a SMS server through RS232 communication and the SMS server relays the alarm messages to CDMA devices. Finally, the operators received the messages correspond to the problem according to emergency procedures.

VERSION CONTROL AND SOFTWARE DEPLOY

Overall, software version control is very important. In particular, it is more important in case that many developers must cooperate with each other or development should be carried out under the distributed environment over a long period of time. The first use case of a tool for software version control was Concurrent Version System (CVS). Later, it was changed to Subversion [7] for some merits as follow:

- Commit unit as work had changed, not a file
- Atomic commits
- Binary file support
- File and directory renames with Unicode.
- Directory version control

Source Code Version Control

Software development in a distributed environment means that the developed codes are distributed more widely and possibility of wrong programs may increase. To create an EPICS IOC, we generate many kinds of files such as c/c++, h, db/vdb, dbd, config, template, st/stt, makefile and so on. It is likely to cause some bugs in programs if we modify or change these different types of files after a long time without software version control. In addition, the necessity of software version control becomes more prominent when program development is carried out by outside contractors.

Software Deploy Management using Subversion

Subversion supports the binary file formats for version control. As shown in Table 1, there are 105 servers for KSTAR operation. Deploying and compiling the source code in every server are very difficult and often a large burden. For these reasons, we installed the linux OS as far as possible to maintain the binary file compatibility and

binary files generated by linux machines work properly in all linux servers except for only a few cases. Therefore, it is possible to deploy the binary modules (libraries or execution files) even if they are generated by a virtualized development server. In particular, it is more suitable for deploying OPI programs because they are modified frequently. The following is the procedure for deploying the binary file.

- Software development in virtualized developing server (source code development and compile)
- Software testing in virtualized distributed servers (software verifying in runtime)
- Commits binary software in SVN server
- Deploying binary software (object files or execution files)

VIRTUALIZATION FOR DISTRIBUTED ENVIRONMENT

Recently, green IT has emerged as a buzzword in the IT industrial field, and among the green IT technologies the computer virtualization technology has the lead. Virtualization of computer (server and desktop) within a limited environment increases the effectiveness of IT resources and reduces costs. As hardware performance and virtualization technology have been improved, a virtual machine has been substituted for a stand-alone server. For the purposes of server redundancy and cost saving, KSTAR has also adopted the virtualization in servers being less intensive against CPU utilization.

The virtualized servers are implemented using ESX/ESXi hypervisor of vmware [8].

Servers virtualized in KSTAR:

- Virtualized OPI servers for remote experiment participants (10 remote OPI servers)
- Virtualized servers for software testing
- Virtualized development servers
- Operation of virtualized EPICS gateway servers

To use the reliable and verified software for KSTAR, new software should be tested and verified first in the virtualized environments which are simulating the real environments before application. In the future, this virtualization will be extended to the data service servers after testing and pre-verification.

FUTURE WORKS

As the KSTAR experiment campaigns have continued, the longer pulse plasma has been achieved and the more data has been obtained. It has led to lacks in network bandwidth and server resources; moreover the shortages become more prominent in the 2011 campaign. To mitigate the insufficiency and enhance the performance of the infrastructure, we will perform the upgrades as follows:

- Completion implementation of storage monitoring system using SNMP in the CSM
- Consideration of clustering server system
- Implementation of fault tolerant server

- Load balancing for data service
- Virtualization for data service

CONCLUSIONS

The CSM has been developed using several standard protocols for monitoring and supervising the entire control system and IT infrastructure to assist the non-interruptible tokamak operation and successful experiments of KSTAR. The faults or alarms detected by the CSM are immediately reported to the operators by the helps of the supervisory interlock system and the SMS system. The operators then quickly restored the problems according to the emergency procedure. As a result of this process, the KSTAR was able to perform the continuous operation and experiment without interruption during the four months. The use of subversion for software version control minimized the human errors due to the wrong software management and helped the stable operation of KSTAR. Through the virtualization adopted in the KSTAR, the time and cost for consisting of IT infrastructure was reduced for the sudden demands of operation. Overall, software testing into the virtualized environment before applying to the real environment increased the stability of the KSTAR operation.

REFERENCES

- [1] EPICS (<http://www.aps.anl.gov/epics/>)
- [2] K.H.Kim, et. Al., The KSTAR Integrated control system based on EPICS, Fusion. Eng. Des. Vol.81 (2006), pp 1829-1833.
- [3] Mikyung Park, et. Al. Overview of KSTAR Integrated Control System, Nuclear Engineering and Technology vol.40 (2008), pp 451-458
- [4] Qt-SDK (<http://qt.nokia.com/>)
- [1] S.Baek, et.al., "KSTAR widget toolkit using Qt library for the EPICS-based control system" Proceedings of ICALEPCS (2009).
- [6] KWT (<http://kwt.sourceforge.net/>)
- [7] Subversion (<http://subversion.apache.org/>)
- [8] Virtualization (<http://www.vmware.com/>)