

Medium-Speed Feedback Software Based on the Existing Control System

A. Taketani, T. Fukui, K. Kobayashi, T. Masuda, T. Ohshima, R. Tanaka, T. Wada and A. Yamashita
 SPring-8, Kamigori, Ako-gun, Hyogo 678-12, Japan

Abstract

The Equipment Manager(EM) model has been already employed by the control system of the SPring-8 storage ring. It is implemented on the client/server software structure, and it accepts human-oriented commands from clients on the operator consoles. The EM can access the VME I/O modules with reasonable execution time, however, there is the application field to develop much faster software on the CPU board by using the existing EM framework. A new software process - EM agent(EMA) is developed as a virtual device which can be used for the feedback purpose. This paper describes the development of the EMA, and its performance is reported.

1 Introduction

The Equipment Manager(EM) is a RPC server process which accepts the human-oriented command from a client process and accesses the machine component through the VME modules[1]. It is general that a message has to be sent from the operator console to the remote running cpu over the network. It means that all message transactions are suffering from the network overhead anytime. It concludes that there is a limit of the performance coming from the network overhead. The performance of the EM is measured to be 1msec typically for one device access. It takes 0.8msec for resolving of the human-oriented command to that of the corresponding hardware command, and 0.2msec for device driver calling. The execution time between the MS and the EM is found to be 11msec for the round trip. It is indicating that most of the time is spent for the network communication. It is good enough for most of the accelerator operation, however, there is the application field to develop much faster software on the CPU board by using the existing EM framework.

1.1 Equipment manager agent

The basic idea of the EM Agent(EMA) is to reduce the network overhead. The EMA is developed as a virtual device which can be controlled by the EM equivalently as that of the real hardware device. One of the application field of the EMA can be a software feedback system with the medium speed, 10~20msec per one transaction. If we try to make a feedback loop, the feedback process needs to read the equipment status, calculate the feedback values, and manipulate the equipments according to the given algorithm and parameters. The EMA is using the same SVOC commands and calling the EM as the local call. The similar sequence as of the GUI is implemented on the EMA. This allows the easy migration from the slower control by GUI's to the faster scheme with the EMA procedures.

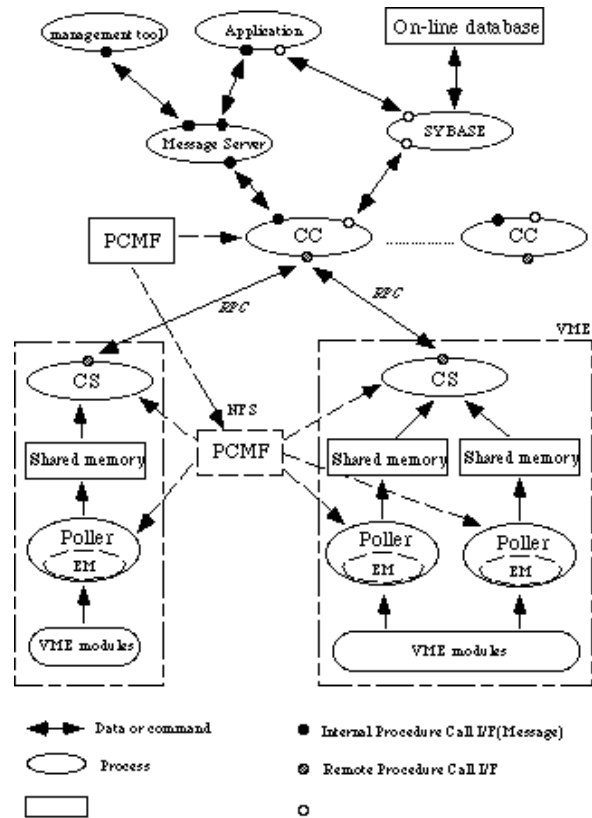


Figure 1. Schematic diagram of EMA.

2 Software structure

The EMA is developed with the same framework as that of the EM's. It is using the same function set and device configuration tables. There are two parts in the EMA. One is the common frame part which manages the EMA process control such as start/stop and the feedback parameter setting (parameters are kept inside EMA as the internal static values). The other is the feedback algorithm part which accesses and manipulates the VME I/O modules through the existing EM frame with the same device configuration file.

2.1 EMA frame

When the EM receives the create command, the EMA is created by the EM as the child process with the same device configuration file. The EM sends a start command to the EMA, after starting the EMA process, the EM can make communications by sending the command to and getting the reply from the EMA through the message queue

mechanism. The commands and replies are described in the same message syntax as that of other EM commands. The EMA is implemented as the virtual device which has the feedback control by the parent EM.

The EMA frame checks the message queue at the beginning of every feedback loop. If there is a command from the EM on the queue, the EMA reads it and replies immediately to the EM. The parameters for the EMA is downloaded as the command strings without stopping the feedback procedure. The EMA continues to run until it is stopped by the EM.

2.2 Algorithm

The algorithm implementation is very similar as that of the GUI. The sequence is made from a set of V/O/C and coded into user functions. The data replied from the VME I/O modules are stored into the local memory space and used for the feedback procedure.

3 EMA performance

For the EM, typically one I/O modules access takes 1msec for the resolving the human-oriented command. It is the same for the EMA. The EMA needs another 1msec for the frame part to handle the message queue. If one tries to make 10 I/O accesses for reading and manipulating the machine components, it takes about 11msec in total instead of about 100msec comparing to that of through the network.

3.1 Application to the klystron control

We applied the EMA scheme for the klystron control as the real case. The control process which ramps up the klystron power needs many feedback loops by reading vacuum gauges and tuning the klystron power. During the ramping up the klystron power, an arc may be caused in the cavity due to the non-flatness of the surface. It happens with the certain applied voltage into the cavity. The arc hits

the surface and kicks out the gases. Then the vacuum pressures would be worse and the reflection power to the klystron is increased. Finally the klystron becomes to be not operational.

The feedback loop reads the vacuum pressure gauges of the RF cavities and controls the klystron power. The response time of the vacuum gauge is about 20msec, it is the slowest part of the feedback system. This feedback procedure needs to monitor the 10 signals, that is, three for the vacuum gauges, six for monitoring the klystron status and one for the power controller. Additionally, these modules had been installed on the two different VME's.

At the beginning we attempted this feedback loop with the EM scheme by sending the control messages from RF GUI to the EM's over the network step by step. One loop took 300msec including the GUI's event loop. It was too slow to catch up the changing the vacuum pressure.

We modified the hardware configuration such that all I/O modules were hooked up to the single VMEbus in order to be controlled by the same CPU. And the feedback system was ported by using the EMA scheme. It provided good performance, stable operation, and smooth ramp up of the power.

This scheme has been newly introduced and working since the beginning of the SPring-8 storage ring commissioning.

4 Conclusion

We developed the EMA software feedback system. It has been working as the part of the SPring-8 control system. The applied field of the EMA can be not only the feedback but also any fast looping procedures as the beam orbit correction.

Reference

- [1] A.Taketani et al., Proc. of ICALEPCS'95, Chicago, USA, 1995, p.625.