MULTI-HOST MESSAGE ROUTING IN MADOCA II

T. Matsumoto[#], Y. Furukawa, K. Okada, JASRI/SPring-8, Hyogo, 679-5198, Japan

Abstract

MADOCA II is the next generation of the Message and Database-oriented Control Architecture (MADOCA) that was implemented in the control systems of the SPring-8 and the SPring-8 Angstrom Compact Free Electron Laser (SACLA) data acquisition (DAQ) systems in 2013 [1]. In 2014, SACLA was equipped with a second beamline (BL2) to increase the capacity of experiments. Messaging over a firewall network is required for beamline control, and an ad hoc socket application was used for this because MADOCA II does not have the requisite functionality. However, this caused problems and increased the complexity of the control system. In this paper, we propose multi-host message routing in MADOCA II to get rid of the socket application In SACLA DAQ systems, multi-host message routing is used for sophisticated access control. We introduce a master server that mediates control messages between operator workstations and equipment management servers. Since access control can be centralized to the master server, reliable operation was achieved by avoiding the effects of the accidental modification of DAQ settings by end users. We implemented multi-host message routing in the SACLA DAQ system in September 2014, and it has been stably operating since then.

INTRODUCTION

As the next generation of the Message and Database-oriented Control Architecture (MADOCA), MADOCA II was successfully implemented to control systems for SPring-8 and the SPring-8 Angstrom Compact Free Electron Laser (SACLA) data acquisition (DAQ) system in 2013, as reported at the last meeting of the International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS) [1]. Like its predecessor, MADOCA II is based on a message-oriented control scheme. However, messaging in this architecture has been made more flexible by replacing the messaging protocol with the ZeroMQ socket library [2] and reconfiguring the messaging scheme. In MADOCA II, data of varying length, such as image data can be attached to messages, and control applications can be used in the Windows operating system environment. Such features were utilized in several control applications in SPring-8 [3] [4].

However, MADOCA II needed to be extended when a second beamline (BL2) was equipped to the SACLA system in 2014. There are several network layers in the control system for experimental users, data acquisition, and the beamline equipment. We sometimes need control

over the firewall network, which requires messaging via an intermediate host, as shown in Figure 1.

MADOCA II is designed to facilitate communication between hosts. To implement message routing among more than two hosts, we used an ad hoc socket application to mediate messages. However, we found that the application stalled at times. This was because the application processes messages sequentially, and is affected when a messaging procedure takes a while. Moreover, the application renders the control procedure complex, making it difficult to understand when and why control troubles occurred. To solve these problems, we developed multi-host message routing in MADOCA II.

In the following section, first, we provide an account of the implementation of multi-host routing in MADOCA II and its application to the SACLA DAQ system. We conclude with a summary of our findings.
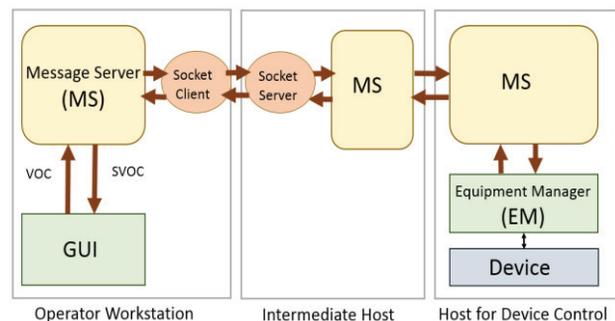

Figure 1: Messaging via an intermediate host in MADOCA II. An ad hoc socket application is used to mediate a message.

IMPLEMENTATION OF MULTI-HOST MESSAGE ROUTING IN MADOCA II

In this section, we describe the implementation of multi-host message routing in MADOCA II. We first explain the messaging procedure between two hosts, followed by an account of how the scheme is extended to multi-host message routing. Some technical issues, such as access control, prevention of messaging loops, and RTT (round-trip time) during messaging, are also addressed.

Message Routing Between Hosts

Figure 2 shows an example of messaging between hosts. A message command is exchanged between a graphical user interface (GUI) on host-b and an equipment manager (EM) on host-a through a Message Server (MS) in each host. In MADOCA II, messages are based on text and composed of a character string in subject/verb/object/complement (S/V/O/C) syntax. In case of messages from a GUI, "put" (V) represents the

control action, "objectA" represents the equipment to be controlled, and "start" (C) represents the value to describe the content of the action. The subject (S) is composed of process number, application name, account name, and host name, and is determined by the control framework as "123_matumot_testgui_host-b." The response is obtained from an EM and read as "objectA/put/123_matumot_testgui_host-b/ok" (in the response, S and O are interchanged).
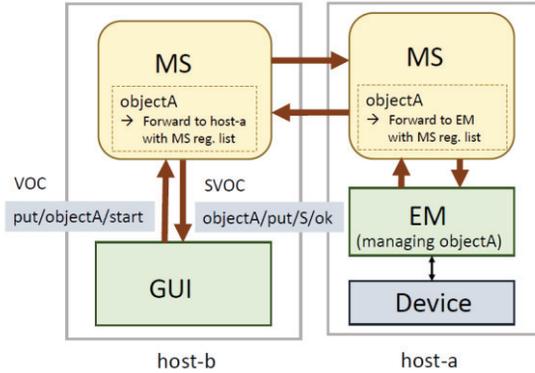

Figure 2: An example of messaging between hosts in MADOCA II. A GUI sends a VOC command, and the response is returned in SVOC format from an EM (equipment manager). Object information registered in an MS (Message Server) is used for message routing.

To implement message routing, we use the object name (O) to determine the route to an EM and the subject name (S) to identify the route back for the message to the GUI because S contains information regarding the GUI, such as application name. In an MS, the following information is registered through related sockets to handle message routing among applications:

1. S defined in the application, such as a GUI or an EM in the same host.
2. Hostnames for connecting or connected MS(s).
3. Object names tagged with the S of an EM in the same host. Object information is also registered to a connected MS.

When an MS on host-b is connected to one on host-a, "objectA" is registered in an MS on host-b. Therefore, a message can be transferred from a GUI to an EM by referring to the registered information in the MSs.

Message Routing Through Multiple Hosts

In MADOCA II, object information is used to determine the route of the message. However, this information is only shared directly with the connected MSs. Therefore, multi-host messaging involving more than two hosts is not possible in MADOCA II. To overcome this limitation, we developed an extension in MADOCA II to forward a message to other hosts even if the object in the SVOC message is not registered in an MS.

Table 1: An example of the Access Control List (ACL) in MS

Object name	Management method	account@hostname
sr_ms_serve	MS	*@*
sr_ms_manage	MS	control@localhost
object_fwd1	MS:host1	*@*
object_fwd2	MS:host1,host2	*@*
object_ip1	MS	*@172.14.12.15
object_ip2	MS	*@172.14.12.0/24

The message forwarding scheme was implemented by extending the Access Control List (ACL) used to set the MS for each host. Table 1 shows an example of the ACL. ACL is composed of three columns. The first column, "object name," is the target object name for access control. Using this column, objects with forward-matched names can be specified. The second column, "management method," defines the method for message forwarding. The management method is set to "MS" if the object is managed using the normal method. When a message is forwarded to host1, we define the management method as "MS: host1" even if the relevant object is not registered in the relevant MS.

We can also define the management method as "MS: host1, host2." In this case, the message is forwarded to host2 when message forwarding to host1 fails. Thus, we can set a priority for message forwarding.

The third column, "account@hostname," defines an account name and a hostname in access control. The details are described in the next section

Figure 3 shows an example of multi-host message routing using our proposed extension in MADOCA II. A GUI sends a message, "get/objectA/start," and an EM responds by finding an intermediate host. The MSs have connections between host-b and host-m, and host-m and host-a. To implement multi-host message routing, we define the message forwarding rule by setting the ACL to host-b. The management method of the ACL is set to "MS: host-m" for "objectA." We can then forward the message from host-b to host-m. Following this, the message is routed to an EM because the intermediate host-m is connected to host-b through an MS, where the EM is located, and information regarding "objectA" has already been transferred from an MS on host-a to one on host-m.

To return the message from the EM to the GUI, however, only the subject name (S) in the message is not sufficient because we also need to know the message route. Therefore, we attach the message route (msg_route) while forwarding the message. When the message is forwarded from host-b to host-m, we set msg_route as "host-b: host-m." The msg_route is then updated to "host-b: host-m: host-a" when the message is forwarded from host-m to host-a. With msg_route, we identify the message route and can use it to return the message from an EM to a GUI.
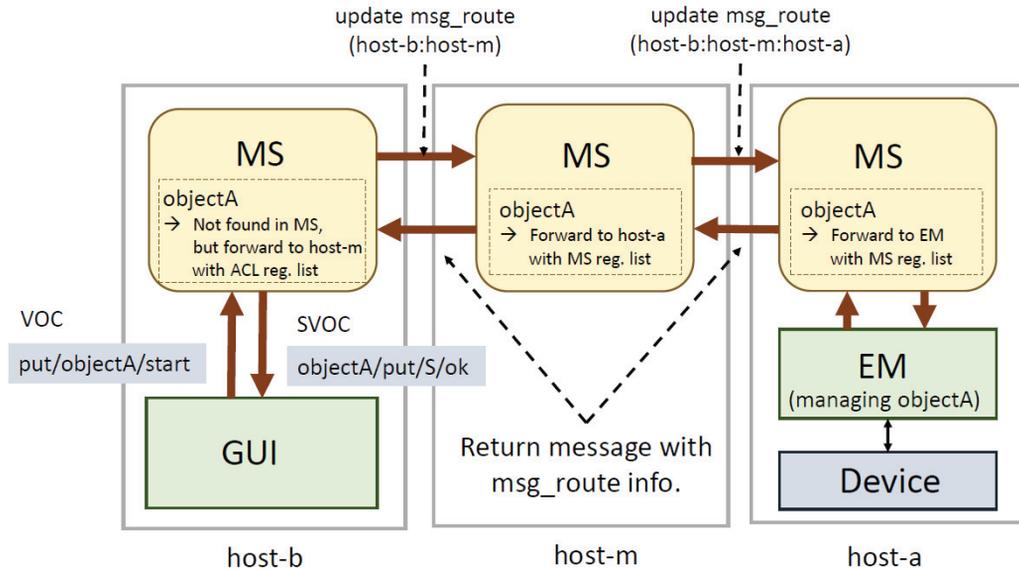
Figure 3: An example of messaging with multi-host routing in MADOCA II. A message is forwarded from host-b to host-m by setting the forwarding rule in the ACL. The message route is used to return the message to a GUI.

Figure 3 shows multi-host message routing through three hosts. In principle, the messaging scheme can be applied to multi-host message routing through more hosts.

Access Control

Access control in MADOCA II is implemented by referring to column "account@hostname" in ACL. Here, we define the account name and the hostname to allow control access. In Table 1, the object "sr_ms_manage" is allowed access with a control account to a local host only because the object is used for management purposes. However, there is no restriction on access to object "sr_ms_serve."

In the SACLA DAQ system, we need to apply control access at specific network segments because some end user PCs are connected through the Dynamic Host Configuration Protocol (DHCP) and we cannot identify these hostnames. Therefore, the ACL is extended to manage access control using IP addresses with subnet masks, as shown in Table 1. If the requirements of access control in the ACL are not satisfied, a message with un-authorized error is returned.

Preventing the Messaging Loop

In multi-host message routing, messaging loops should be avoided because they increase network traffic use and cause control problems. To prevent message loops, we use msg_route attached to the message while forwarding the message. The messaging loop can be detected by checking for the duplication of the hostname in msg_route. If a message loop is detected, an error message is returned.

RTT

In Multi-host message routing, we expect longer RTTs between a GUI and an EM because the messages are mediated by hosts.

We measured the RTT with varying configurations on the hosts, as shown in Table 2. When we performed message routing through three hosts, RTT of approximately 3 ms were observed. RTT delays of approximately 1 ms were observed in comparison with the case with message routing involving two hosts.

In the SACLA DAQ system, we found that the delay in RTT was not a problem because we used the messaging for slow control.

Table 2: RTT of a messaging between a GUI and an EM with several configurations. We used an EM, which simply returns a clock value. The test was performed using PCs with Intel Core i5-425U processors and a 1 GbE network switch under a local network. RTTs were averaged over 10,000 samples.

Case	RTT
A GUI and an EM in the same host	0.40 ms
A GUI and an EM in different hosts	1.89 ms
A GUI and an EM in different hosts with an intermediate host.	2.75 ms

APPLICATION OF MULTI-HOST MESSAGE ROUTING TO SACLA DAQ SYSTEM

In the SACLA DAQ system, shot-by-shot information synchronized with an X-FEL (X-ray Free Electron Laser) beam was stored at 60 Hz at the maximum repetition rate [5]. The challenge for multi-beamline operation was handling massive image data from experiments in each

beamline. To enhance the data handling capacity, the data stream and the network layer were upgraded to process separately [6].

We also needed reliable control in the face of complexity in multi-beamline operation. To satisfy these requirements, we redesigned the control architecture as shown in Figure 4.
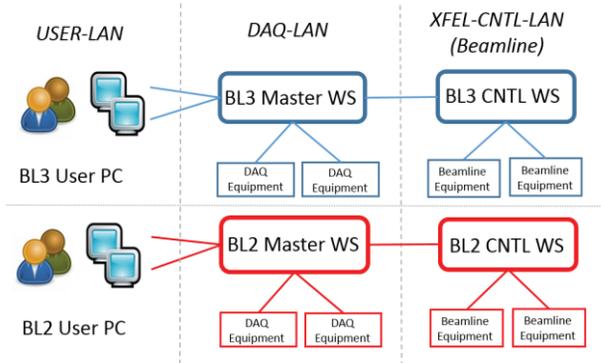

Figure 4: Control architecture of multi-beamline operation in the SACLA DAQ system. Access control was centralized through BL-master WSs and secured with settings in VLAN.

We split the network into three layers: DAQ-LAN was the network that controlled the DAQ equipment, XFEL-CNTL-LAN controlled beamline equipment, and USER-LAN controlled experimental end users as well as the equipment for DAQ and the beamlines. Direct access to XFEL-CNTL-LAN from USER-LAN was forbidden. We implemented multi-host message routing for the controls between USR-LAN and XFEL-CNTL-LAN.

For messaging control, a BL-master workstation (WS) was introduced to mediate all commands between user PCs and applications for equipment management. The user-PC could control all equipment in DAQ and the beamline simply by connecting to an MS in the BL-master WS. By passing BL-master WS in the messaging, we could monitor the message log for user PCs thorough the BL-master WS. We were also able to centralize the access control settings at the BL-master WS, hence avoiding the influence of missed configurations by experimental end users. As an example, it was possible to limit the use of some motor axes to facility usage. In USER-LAN, PCs were available for the experiments. We also forbade access to beamline equipment for user PCs through the BL-master WS. Such access control can be dynamically updated by modifying the ACL setting of the BL-master WS according to the experimental conditions.

Therefore, access controls can be more flexibly adjusted than settings in the firewalls of networks.

We implemented multi-host message routing on multi-beamline operations in the SACLA DAQ in September 2014. In general, there were two user operator PCs in each experimental hatch, for a total of six. The operation thus far has been limited to slow controls, and no major problem has been reported during the one-year multi-beamline operation.

In 2015, an additional beamline (BL1) was added to SACLA. The adoption was effected by simply scaling up using the same control architecture.

SUMMARY

In this paper, we proposed multi-host message routing for MADOCA II and its application to the SACLA DAQ system for sophisticated access control under multi-beamline operation. The multi-host message routing was implemented by forwarding specific messages to other hosts by setting the ACL. With multi-host messaging, we can achieve flexible access control by centralizing access controls in the BL-master WS. The control scheme was successfully implemented to the SACLA DAQ system and has operated stably for a year. The same control scheme was applied to a new beamline (BL1) in SACLA. We also plan to adapt multi-host message routing to the SPring-8 beamline, where access control over the firewall network layer is required.

REFERENCES

[1] T. Matsumoto et al., "Next-generation MADOCA for SPring-8 control framework," Proceedings of ICALEPCS 2013, San Francisco, USA, (2013) p. 944.

[2] http://zeromq.org/

[3] A. Kiyomichi et al., "Development of MicroTCA-based Image Processing System at SPring-8," Proceedings of ICALEPCS 2013, San Francisco, California, USA, (2013) p. 78.

[4] Y. Furukawa et al., "MADOCA II Interface for LabVIEW," Proceedings of ICALEPCS 2013, San Francisco, California, USA, (2013) p. 410.

[5] M. Yamaga et al., "Control and data acquisition system for X-Ray free-electron laser experiments at SACLA," IEEE Nuclear Science Symposium and Medical Imaging Conference 2012 Record (NSS/MIC), Anaheim, USA. (2012).

[6] K. Okada et al., "Upgrade of SACLA DAQ system adapts to multi-beamline operation," PCaPAC 2014, Karlsruhe, Germany, (2014) p. 22.