Hierarchical Agent System for Web-based Sensor Network

Tokihiro Fukatsu, Takuji Kiura, Kei Tanaka, Masayuki Hirafuji
National Agricultural Research Center,
3-1-1 Kannondai, Tsukuba, Ibaraki, 305-8666 Japan
fserver@zoushoku.narc.affrc.go.jp

Abstract

To achieve a smart sensor network with Web-based sensor nodes such as Field Servers, we propose a hierarchical agent system that manages sensor nodes autonomously and flexibly, and acts as an intelligent module of the sensor network. Our agent can treat Web applications in addition to sensor nodes and is a Web application itself; thus, it can control other agents hierarchically. This hierarchical structure provides the Web-based sensor network with high scalability and robustness. The leaf node of our agent system, called the Field Server Agent Box, is equipped with an agent system, a Web interface, a storage function, and data viewers. It can be used with Web-based sensor nodes at sites where an Internet connection is not available. In this study, we also examine the leaf node's scalability.

1. Introduction

A Field Server (FS) developed at the National Agricultural Research Center is one of the Web-based sensor nodes equipped with a sensing processor and a high-speed wireless LAN [1, 2]. We can use it not only for monitoring basic data as a conventional sensor node but also for handling a variety of data, controlling peripheral devices, and using large blocks of image data by accessing Web applications through it. Web-based sensor nodes generally do not function by themselves to provide autonomous monitoring or control action, but they can all be treated in the same way and universally accepted on the Internet [3, 4]. By using many Web-based sensor nodes and an agent system that acts autonomously according to the situation and manages enormous sensor nodes with small effort, a smart sensor network can be constructed with high scalability and a robust system. In this paper, we describe the agent system architecture used to realize a Web-based sensor network, and evaluate its effectiveness with small computers embedded in the agent function.

2. Agent System

To realize this sensor network effectively, we propose a hierarchical agent architecture in which a single agent system manages some nodes by performing an intelligent module of a Web-based sensor network. The Single agent is designed to treat itself as just one of the nodes (meta-nodes) at the same time. We describe in detail the functioning of the single agent system and the standard structure of the multi-agent system and the hierarchical agent system with a FS Agent Box. These structures are designed to achieve high scalability and a robust system.

2.1. Single Agent System

The agent system is constructed by an agent program written in Java, program configuration files describing the parameters of request operation, and a Web interface for easy management of the agent system (Fig. 1). The agent program can access all types of pages on the Web and can emulate user operations on a Web browser from anywhere at any time. It is designed so that it can operate automatically based on configuration files, which are selected from a configuration file list according to the situation. In the configuration files, there are agent program instructions combined with some tiny action commands which have a basic function to operate Web pages and a support function for the agent [5]. By simply changing the configuration files, users can easily manage the agent operation flexibly with general versatility. Using the action commands, it is easy to cooperate with sensor nodes and Web applications that execute various functions such as image analysis and signal processing. More versatile and easily expansible functions can be realized without changing or rebooting the agent program. This makes it possible to distribute calculation tasks by means of Web applications. To adapt to various situations using intelligent control, the agent system has IF-THEN rules and an Operator function, which are derived whether or not the agent action executed from a monitoring environment.

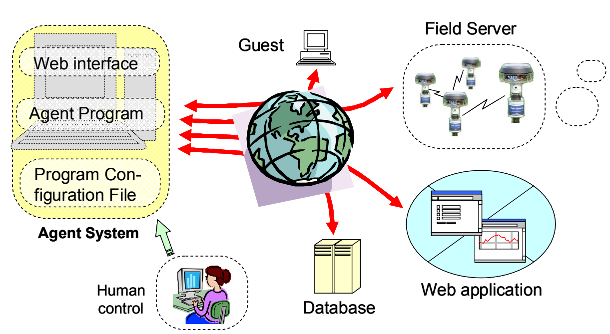

Figure 1. Agent system architecture.

We can establish simple conditional judgments and complicated rule-based AI algorithms by providing these specifications in the configuration files with these functions [6] (Fig. 2).

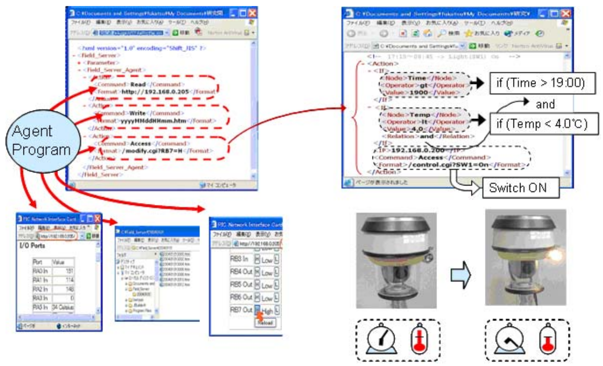

Figure 2. Example of Rule-Based Operation.

2.2. Multi-Agent System

The agent system has a Web interface that facilitates access to parameter files, the agent information, and monitoring data. This function allows the elements of the agent system to operate via an http protocol, so the agent system itself can be treated as just one of the nodes in our system. By using an agent program called a meta-agent with the same program used to manage other agent programs via the Web interfaces, we can construct a distributed agent system that can treat many nodes with high scalability (Fig. 3). The meta-agent determines how many node tasks should be allocated to each agent program considering the capacity and the situation of each [7]. When the number of nodes increases, the system can adapt with high scalability to being implemented in a layered meta-agent architecture in which a meta-agent handles other meta-agents according to their scale. It can carry out its management function effectively and automatically so as to decrease management efforts.

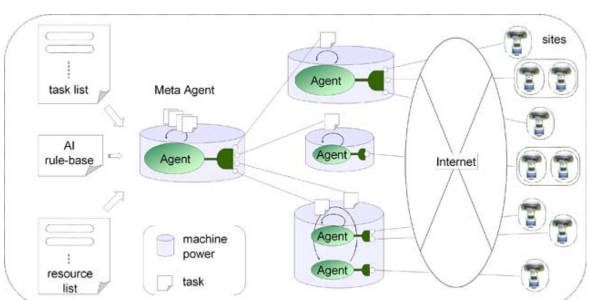

Figure 3. Multi-Agent System.

To establish a robust system, a nested structure of the agent system in which several agent programs handle several nodes simultaneously can be also realized to provide strongly reliable performance with respect to various factors. The system is constructed so that a secondary agent program does not access the target nodes while the primary agent program works normally. The secondary agent system periodically accesses the primary agent system in order to check its condition, and it operates on the target nodes only when the primary agent program is down. By cooperating with the nested agent system, it performs as an effective data backup system, thus making the system robust [8] (Fig. 4).

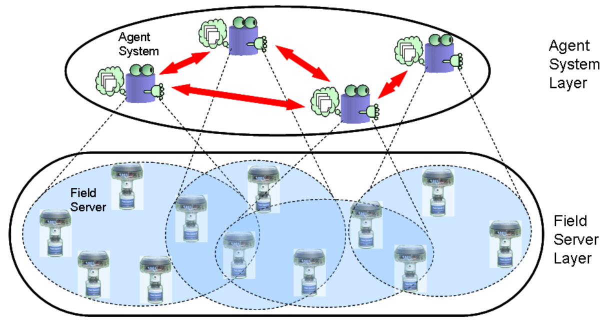

Figure 4. Nested agent system.

2.3. Field Server Agent Box

This framework works well when the Internet connection is stable, but in some locations the Internet connection is frequently interrupted. There have been requests to introduce this system in regions where no broadband Internet service is provided. To alleviate this problem we added a small Linux box called the “FS Agent Box (FSAB)” to the local site.

The FSAB has an agent system for controlling the FSs or collecting their data. The FSAB stores data to its local hard disk drives and provides data to the local Web-based sensor network at the observation site. We have already developed the FS Gateway (FSG) equipped with our virtual private network (VPN) to maintain stable connection [9]. By installing a FS agent system to one of our FSGs as a prototype FSAB, we tested this prototype with shell scripts that checked unexpected halt and restarted the system. After finishing the test of the prototype FSAB, we developed two types of FSAB. Type A was based on a small PC named the MicroPC manufactured by Hi-Tech Systems (Fig. 5). Type B was based on an NAS kit named

- Hightech System EES-3611
  - CPU: C3 (1GHz)
  - RAM: (up to 1GB)
  - HDD: 2.5inch (80GB)
  - LAN: 10/100
  - VGA, PS2, USB2x4, etc.
  - OS: Vine Linux 3.2
  - JRE: Sun or IBM
  - 68×223×153(mm)

    Over 9 months at Florida

    (< US$ 600)

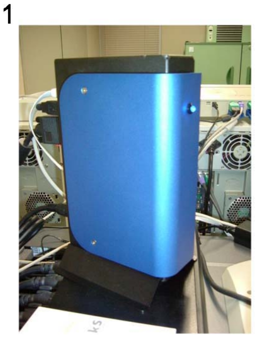

Figure 5. Specs and photo of TYPE A FSAB.

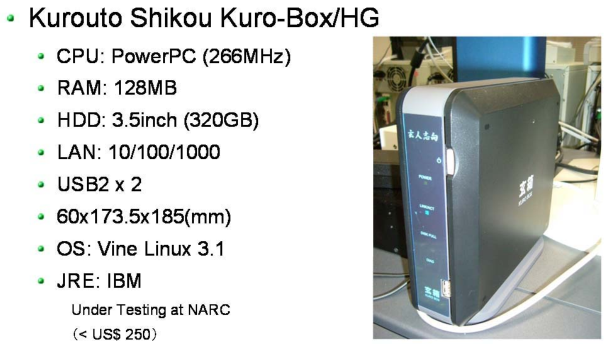

Figure 6. Specs and photo of TYPE B FSAB.

KURO-BOX HG, manufactured by Kurouto Shikou (Fig. 6). The default Linux kernel of the KURO-BOX (Type B FSAB) is 2.4.17, and the Vine Linux 3.1 requires a kernel of 2.4.20 or later. We downloaded the kernel 2.4.20 source code from the LinkStationWiki [10] and compiled it in order to use /dev/hda3 as a root device. We also used loader.o [11] to load the compiled 2.4.20 kernel and installed the Vine Linux 3.1 disk image to /dev/hda3. To provide data, we installed an Apache http server, a ProFTP server, and a samba server on the FSAB. We also installed applet version FS data viewers [12]. Figure 7 shows screen shots of the image viewer and data viewer.

Figure 7. Screen shots of image and data viewers.

3. Trial

3.1. Trial Management

To evaluate the sensor network with the agent system, we installed dozens of sensor nodes in different locations all over the world and managed them with this system via the Internet. The agent programs were executed on the PC clusters and were managed at a remote site with a VPN. This system has performed reliably since 2002, more than 4 years, and the data regarding its nodes that have been collected over the long term show its capabilities and its reliability at stable connection sites [13].

3.2. Scalability Test

To check how many FSs one FSAB can support, we created a virtual test environment using 8 PC servers equipped with a VMware Workstation [14]. Using VMware Workstation 5.5 [15], we created 24 virtual hosts in each PC server, 192 virtual hosts in total. An Apache http server was installed in each virtual host to provide sample FS data. We considered those virtual hosts to be virtual FSs. As a standard FS costs US$ 2,000, 192 FSs cost US$ 384,000. Our virtual test environment cost US$ 50,000, thus saving a great deal of money and time.

In this test, FSAB collected data from the virtual FSs every 2 minutes. Each virtual FS was connected to the same private network 192.168.0.0/24 and obtained an IP address from a DHCP server. So each virtual FS had the same network status. During this test, we found that Type A FSAB took a significant amount of time to parse XML data for data browsing. This problem was caused by the XML parser using Document Object Model (DOM), so it will be solved to change from DOM to Simple API for XML (SAX). In this test, we commented out the XML file creation part for data browsing from agent rule files.

Figure 8 shows the graphs produced by Multi Router Traffic Grapher (MRTG) showing the traffic of the network interface (upper) and the load average (lower) of the Type A FSAB that collected data from 100 virtual FSs. The green bars of the upper graph represent 30-minute averages of the incoming traffic, and the blue lines represent 30-minute averages of the outgoing traffic in bytes per second. Figure 9 shows the corresponding results

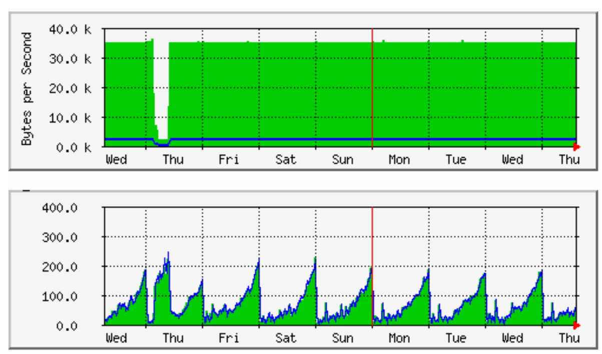

Figure 8. Traffic and load averages of Type A.

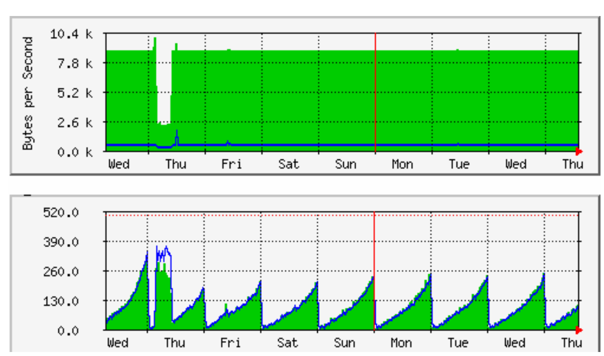

Figure 9. Traffic and load averages of Type B.

for a Type B FSAB that collected data from 25 virtual FSs. The network traffic of each FSAB is small, and the Wi-Fi network provided by the FS has enough band width to transfer data from 100 standard FSs. The load averages were also not very high, and the FSAB can accept data requests from the network. Low traffic and high load averages were observed on Tuesday. At that time, the DHCP server was stopped due to an accident all of the virtual FSs lost their IP addresses, and the agent could not reach any virtual FS. After the DHCP server was restarted, agents could collect data form 100 or 25 virtual FSs. In this condition, our FSABs ran well without any problem. From those result, we concluded that Type A FSAB can support 100 FSs and Type B FSAB can support 25 FSs.

4. Discussion and Conclusion

We examined the agent system of managing Web-based sensor nodes effectively and evaluated the validity of the proposed system for the Web-based sensor network. This system is based on the assumption that all objects can be accessed on the Web and treated as Web-based sensor nodes for scalability and robustness. By using this system in the network environment, it is easy to utilize various modules and to expand their function.

On the other hand, there have been requests to use this system in the unstable network environments. To apply this system in different environments, we developed FSAB as a unit of the agent system to manage Web-based sensor nodes. Our FSAB is proposed to function as a local data collector and provider for a Web-based sensor network. Moreover, it is possible for local users to utilize various functions by installing assistance applications on the FSAB. We are planning to install a reduced version of MetBroker [16] on our FSAB that can use data from the FSAB. Using this version of MetBroker on the FSAB, the user of the FSAB can utilize many MetBroker client programs without an Internet connection.

In stable network environments, the FSAB is also available as a leaf-node of our agent system in a local area and can cooperate with other agent systems at a remote site, creating a hierarchical agent system (Fig. 10). It works as a secondary agent system to manage nodes on its site in order to construct the nested structure. We propose to use our hierarchical agent system with other Web-based sensor networks. We are planning to test this hierarchical agent system using our virtual test environment for FSs to prove the scalability of our Web-based sensor network.

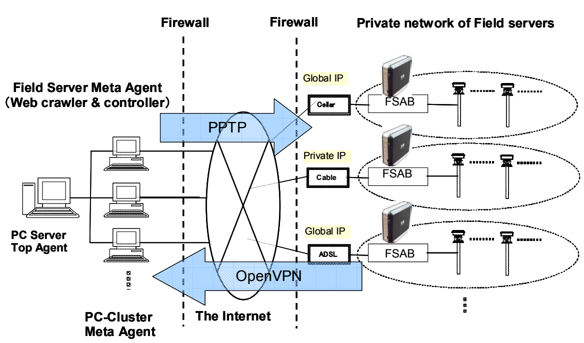

Figure 10. Hierarchical Agent System.

Acknowledgement

This study was supported in part by funds from the MEXT (Ministry of Education, Culture, Sports, Science and Technology) via the "Global Earth Observation Data Integration and Fusion Research Project".

References

[1] M. Hirafuji, T. Fukatsu, "Architecture of Field Monitoring Servers", *Proc. of the 3rd Asian Conference for Information Technology in Agriculture*, 405-409, 2002.
[2] M. Hirafuji *et al.*, "Field Server: Multi-functional Wireless Sensor Network Node for Earth Observation", *Proc. of the 3rd International Conference on Embedded Networked Sensor Systems*, 304, 2005.
[3] T. Fukatsu, M. Hirafuji, "Field Monitoring Using Sensor-Nodes with a Web Server", *Journal of Robotics and Mechatronics*, 17(2):164-172, 2005.
[4] T. Fukatsu, M. Hirafuji, "The Agent System for Field Monitoring Servers to Construct Smart Sensor -Network", *IFAC Proc. of Fifth International Workshop on Artificial Intelligence in Agriculture*, 1-5, 2004.
[5] T. Fukatsu, M. Hirafuji, T. Kiura, "Agent System for Providing Seamless Management to Web-based Monitoring System", *Proc. of the 5th Asian Conference for Information Technology in Agriculture*: (not published yet).
[6] T. Fukatsu, M. Hirafuji, T. Kiura, A. Imada, S. Ninomiya, "Long-Term Monitoring System using Field Monitoring Servers", *Proc. of AFITA/WCCA Joint Congress on IT in Agriculture*, 685-691, 2004.
[7] T. Fukatsu, M. Hirafuji, T. Kiura, "An Agent System for Operating Web-based Sensor Nodes via the Internet", *Journal of Robotics and Mechatronics*, 18(2):186-194, 2006
[8] T. Fukatsu, M. Hirafuji, T. Kiura, "A Distributed Agent System for Managing a Web-based Sensor Network with Field Servers", *Proc. of 4th World Congress on Computers in Agriculture*, 223-228, 2006.
[9] T. Kiura, T. Fukatsu, M. Hirafuji, "Field Server Gateway – Gateway box for Field Monitoring Server", *Proc. of 3rd Asian Conference for Information Technology in Agriculture*, 410-413, 2002.
[10] http://www.linkstationwiki.net/
[11] http://jan.csie.ntu.edu.tw/~jo/kuro/loader/
[12] K. Tanaka, T. Fukatsu, M. Hirafuji, "Data and Image Viewer Application for Field Server", *Proc. of SICE-ICASE International Joint Conference*, 4852-4855, 2006.
[13] http://model.job.affrc.go.jp/FieldServer/
[14] T. Kiura, T. Fukatsu, M. Hirafuji, "Developing a virtual test environment for scalability of Field Servers", *Proc. of the 5th Asian Conference for Information Technology in Agriculture*: (not published yet)
[15] http://www.vmware.com/
[16] M.R. Laurenson, T. Kiura, and S. Ninomiya, "Providing Agricultural Models with Mediated Access to Heterogeneous Weather Databases", *Applied Engineering in Agriculture*, 18(5), 617-625, 2002.