國立交通大學

多媒體工程研究所

碩 士 論 文

資料中心機櫃自動化管理系統

Automatic Rack Cabinet Management System for Data Center

研 究 生：陳孟傑

指導教授：袁賢銘 教授

中 華 民 國 一 百 零 一 年 七 月

資料中心機櫃自動化管理系統

Automatic Rack Cabinet Management System for Data Center

研 究 生：陳孟傑　　　　Student：Meng-Jie Chen

指導教授：袁賢銘　　　　Advisor：Shyan-Ming Yuan

國 立 交 通 大 學

多媒體工程研究所

碩 士 論 文

A Thesis
Submitted to Multimedia Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
In

Computer Science

July 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年七月

資料中心機櫃自動化管理系統

研究生: 陳孟傑　　指導教授: 袁賢銘 博士

國立交通大學

資訊工程與科學研究所

新竹市大學路 1001 號

碩士論文

摘要

隨著時代的演進，各種雲端的概念正在不停地蓬勃發展，而在雲端中其中一個重要的精神就是利用大量電腦的計算與儲存能力來取代單一台超級電腦所能做的事情。然而隨著雲端服務的擴增所需要電腦的數量也開始日與遽增，此時如何管理這大量的電腦及其外部設備，使其可以達到自動化控制並且管理者易於操作，並且使設備在安全且穩定的情況下使用就是一個重要的議題。在現今市面上開始陸陸續續出現了一些機架式機櫃管理的外部設備，這類的設備著重於透過網路協定來控制這些外部設備，使得即使這些大量的機架式機櫃設備分散在許多地點，管理者在遠距離也能夠透過網路集中的管理這些設備。然而，在這類外部設備當中自動化議題至今還沒有被人們普遍重視，包含如何自動化處理常見的設備狀況，如何控制電腦啟動的數量等等。此外，管理者通常會希望能夠以語意的方式來思考如何控制這些設備，例如：當電腦溫度過高時，將該電腦對應的風扇加速。這些在管理者腦海中常出現的狀況卻仍依賴著管理者的控制來得到解決，當電腦與設備的數量相當龐大時，管理者將疲於這些常出現的狀況處理。本論文提出了一個可接受語意式命令並且允許使用者自行制訂流程的機架式機櫃外部裝置自動化管理系統，並參照智慧型家庭管理的語言：SHPL (Semantic Home

Process Language)，來描述語意化的流程，透過本論文自動化管理系統可使得大量外部的設備狀況達到自動化的管理，並達到節省電量的效果。

關鍵字:自動化, 管理系統, 節能

Automatic Rack Cabinet Management System for Data Center

Student: Meng-Jie Chen　　Advisor: Shyan-Ming Yuan

Institute of Computer Science and Engineering

National Chiao Tung University

1001 University Road, Hsinchu, Taiwan 300, ROC

Abstract

With the evolution of computer science technology the concept of clouds are kept booming. An important concept in the clouds is the use of computing and storage capacity of the large number of computers to replace the single supercomputer computer. Therefore the more and more computers are needed as the number of cloud services are dramatic increased. The management of a large number of computer and peripheral equipment so that it can achieve automatic controlling and easy operated under safe and stable and lower electricity consumption becomes an important issue. In the market today, it began to land one after another peripheral manage devices of rack server enclosure, and such devices focus on controlling peripheral devices through the network protocol. With such devices, data center managers could centrally control equipment which scattered in many long-distance locations through the network. However, the issue of user-configurable peripheral device automation has largely been neglected recently. The automation issue include automatically handling the common equipment status, and controlling the number of running computer. In addition, managers will usually want a semantic way to think about how to control these devices, for example: When the computer temperature is too high, the corresponding fan speed to accelerate. These often appear situation is still dependent on resolving by the managers. When the number of computers and equipment is substantial, managers will be tired of these common conditions to deal with. This thesis presents a user-configurable semantic control system which allows users define their own automation control processes. The language of the intelligent home management: SHPL (Semantic Home Process Language) is used to describe the semantics of the process. The system can make a lot of peripheral equipment status under automated management and reach a power-saving effect.

Keywords: Automation, Management system, Power saving

致謝

首先我要感謝我的指導教授—袁賢銘 博士，在研究上給予耐心的指導，並提供精闢的見解引導我們對問題作更進一步的思考，使我們對於研究有了更深入的認識與想法，並且袁老師在研究上追求卓越以及實事求是的態度，使我們在研究上有了正確的態度。袁老師細心與嚴謹的指導，使我研究生涯受益良多，在此謹向我的老師致上無限的敬意。

其次，我要感謝高永威學長在我的研究上擔任了啟蒙的指導，讓我對分散式專業領域上有了進一步的認識，使我在的碩士生涯，不再舉步維艱。謹此致上由衷的謝意。

很榮幸進入分散式系統實驗室，這裡提供良好的學習環境，並且有熱心與親切的實驗室夥伴能夠互相切磋討論，使我在知識與人生態度獲得許多的成長。我要感謝實驗室的學長姐們—林家峰 博士、羅國亨 博士、與江川彥 博士，一步步帶領我進入這個專業的領域，並給予我適時的幫助使我能夠在碩士生涯有了良好的開始；感謝我的好同學們黃聖凱、彭珮瑜、黃俊凱、郭紘維與黃冠穎，不論是研究上或是生活上，他們總是給予我最直接的協助以及苦樂分享，並提供我良好的意見供我參考；感謝我的學弟妹金柏志、楊先博、蕭丞訓與徐振庭，在論文撰寫與修改方面上幫了我很大的忙；在研究設備及架構上感謝友視達科技陳宏道經理在設備上及技術上給予許多的支持；在研究思維上我最要感謝的是高永威博士，給了我許多寶貴意見，使我了解我研究上優缺點以及應該改進的方向，並給予我信心使我能夠做出更好的研究。

最後，我要感謝我的父母—陳榮華 先生、高梅香 女士的栽培，在爭取碩士學位的路上，給予精神上的支持和溫暖，能夠專心的在研究領域上打拼。感謝我的姊姊—陳婷枚，給予我加油和打氣。感謝你們一路的陪伴打氣與支持，在此僅將這篇論文獻給各位，謝謝你們！

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Section 1-1 Preface

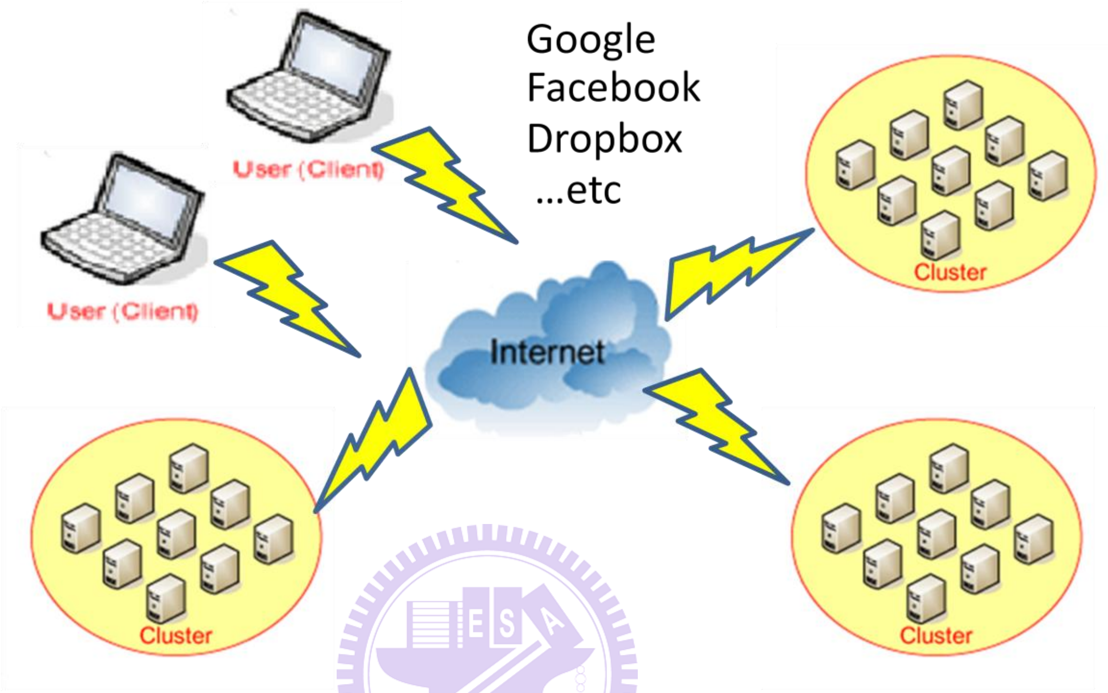

Figure 1.1 The concept of cloud service

The concept of cloud is arising in recent years, and more and more cloud services appeared in the market. For example, Google, Facebook, Dropbox ... etc. As shown in Figure 1.1, users connected to the interface which is provided by cloud service providers, and there are many distributed servers provide computing or storage capabilities behind the internet. With the rise of the number of cloud service and cloud concept, a large cloud service provider and many small companies are constantly increasing the number of cloud data center. The cloud data center management has become a very important issue. More than servers, there are many surrounding devices which are not directly provide the cloud service but play an important role in the stability and security of the server in the data center. The common devices include the rack server enclosure, PDU (Power Distributed Unit) and fan.

In the past, except the server the control of peripheral devices are dependent on staff's management. For example, when the data center staff was informed that the server does not reflect correctly, the data center staff will go to the server's location to confirm whether the server is broken by itself or damaged by peripherals device's broken. When the servers are distributed in different locations, the staff will be tired of these round-trip time. The reason of server's crash is that peripheral device have been broken always causing data center staff feel bad. Why such of this problem cannot be early detected but leading to damage the servers? Nowadays, there is some of these peripheral devices present in the market. These devices integrate network ability, so that it can communicate with manager over network. These IP-based peripheral devices can be controlled by the managers in a remote and centralized way. The status of these peripheral device could be monitored in advance through a simple user interface. However, sometime the number of server and its peripheral device and extremely large, and the management of these server and its peripheral device is very tired even there is a simple user interface. Why we cannot monitor these devices by an automatic way?

Section 1-2 Motivation

From our observations, numerous rack server enclosure peripheral devices are easy to be operated regardless of whether the equipment is old-alone or new type which operating through network protocols. Operators can simply push the button or use the graphical user interface which provided by vendor to achieve the control purpose even the operator have never been used these devices. For example, push the increase / decrease speed button to accelerate / decelerate the fan. The operation of a single rack server enclosure is very simple and easy to handle. However, when the number of rack server enclosure and its peripheral devices become very large, this problem is become very terrible. In old type data center , the staff may need to go around to check the server and devices are running correctly. In new type data center, staff still sit in front of the monitor interface, and do some control to these devices. If nobody keep monitoring these devices and try to set the device into the same setting. There is always hard to find a general device setting to correspond to all the equipment. For example, the temperature in different locations of a data center are not always the same. If all the fan speed are set to the minimum, the server at hotter location in the data center may be overheating. If all the fan speed are adjusted for the maximum, the server at cooler location in the data center may waste a lot of unnecessary electric energy. For another example, the temperature is quite high at noon time, you may need to set all fans speed to max level. However the temperature at night is cooler than noon time, and the fan speed is still the highest fan speed that waste a lot of energy consumption. In such a situation, how to dynamically appropriate control of these large numbers of peripheral equipment is a very important issue.

Power saving is always an important issue for environmental protection and data center owner. Data center owner always spends a lot of money for the electric energy. If the electric energy consumption in the data center could be reduced, the cost of cloud service will be lower. It is not only help the data center owner, but lots of user can get benefit from the cloud service provider. In our observation, the server in the data center is not always highly used, but there are many servers just keeping the standby state and waiting for the user to use its resources. These standby computers provide limited functionalities, but they still non-stopping consume the electricity. If there are many such standby servers, it will cause excessive energy waste. It is not only an unnecessary waste of the data center owner ,but also a serious destruction to the earth environment.

The rack server enclosure door access control system is a very important part of data center management. Nowadays, the rack server enclosure door access system is mostly using RFID for the recognition of identity. The sensor will read out the RFID card number when a user's RFID card touch the RFID sensor on the rack server enclosure door. After, the sensor check the RFID card number is on the valid card number list. If the card number is valid, the rack server enclosure will unlock the door. If the card is not valid, the door will keep locked. Sometimes, we will want these RFID cards are not always valid, if the more meticulous security management is considered. For example, you would want the RFID card of a staff is valid during his working hours, but his RFID card will be set to invalid when he off-duty. That will avoid misconduct data center staff using his position in the non-duty hours to do something misbehavior. For this purpose, managers need to add RFID card number of a staff into the rack server enclosure door's valid access list when the staff is on-duty, and remove the RFID card number from the valid list after his working time. If there is a lot of rack server enclosure which have their own valid RFID card number list, the manager must repeatedly perform add and remove action to each rack server enclosure. It is very tired when there are many staff and rack server enclosures.

The importance of each rack server enclosure may not be the same. Some of the server may manage the entire company financial system, and some may simply be a personal cloud storage server. The data importance in the server may be definitely different, but the server enclosure doors have the same access permit. We will always hope only certain the privileges of the people are able to open the important rack server enclosure door, instead of every staff can access every rack server enclosure door. The rack server enclosure door authorized privilege level is a quite important functions in the management.

For situations have mentioned above, this thesis have implemented a cross-device automatic rack server enclosure management system of data center. This system present a semantics input interface to establish the automatic control flow. The managers can take advantage of the semantic approach to control common peripheral device situation, and get better controlling to the dynamical peripheral devices situation in real-time. This system also integrates the RFID card management. The administrator can set the effective time and authorized level for each RFID card number. Therefore, servers can be protected from misconduct staff after their work hours, and the high importance of the servers could be protected safer due to lesser staff can access it.

Section 1-3 Outline of Thesis

In the chapter 2, we will introduce the background and related work of our research. The state of the art IP-base rack server enclosure peripheral device will be introduced in chapter 3. In the chapter 4 , we will introduce the whole design and detail implement of our system. The demonstration and interface will be introduced in chapter 5. Some of experiments and simulation results will be discussed in chapter 6. The conclusion and future work will be present in chapter 7.

Chapter 2 Background and Related Work

Section 2-1 Background

With the emergence of cloud service, the development of data center infrastructure is seen worldwide nowadays[1]. A data center or computer centre is a facility used to house computer systems and associated components, such as telecommunications and storage systems. It generally includes power supplies, data communications connections, environmental controls (e.g., air conditioning, fire suppression) and security devices. The data center plays an such important role, but there is so many detail we need to concern for better management. Therefore, more and more people pay attention to the management of data center gradually [2][3]. The management of data center includes a lot of factors, including infrastructure, security, server resource distribute, heat dissipation, the power control and automation.

In the issue of heat dissipation, some people began to analysis the material of the rack server enclosure and heat flow direction inside a sealed rack server enclosure and the correlation between running server and heat resource [4] [5]. These analysis try to predict the heat situation inside the rack server enclosure and providing possible solution of the heat dissipation. Some of study focus on the cooling system of the rack server enclosure to create more stable rack server enclosure environment[6]. Such studies tend to focus on the model generation and accuracy of the prediction, but lacking of true temperature in real-time management. It will cause some differences between reality and prediction, and the reliability sometime will be doubted in the actual data center management.

The power management issues is also widely discussed in the data center management. In the past, the design and the use of newer server power supply achieved lot of energy saving in single server [7]. These products are also widely used in current servers and personal computers. For remote monitoring the power information, some people started using JSP technology [13] on the web-site to monitor the power situation in the remote site [11] [12]. Some studies of the power monitoring and control system use SNMP[8][9] combined with the UPS device and Internet archive the power control [14]. These studies combine power management with network monitoring, but the extension of data center power control ware not considered. In the other hand, some of studies have focused on the whole data center design and management structure. These studies consider the demand of workload for electricity to design the structure and power supply in the data center [15]. The server consolidation based on virtualization is an important technique in cloud infrastructures[10]. There are many studies discuss the power efficiency and resource utilization of server consolidation based on virtualization. Server consolidation through the virtualization to control the power and effectiveness, and there are some studies base on the control theory or queue theory to predict the utility of the servers to guarantee QoS [16] [17]. However, these studies focus on the prediction and the real situation is regardless.

The automation is also a important factor in the data center management. The data center automation concept is discussed since 1990's [18]. Although the equipment in that time is far away from nowadays. The importance of the concept is never decreased. The policy-based concept of data center management is discussed recently[19], and it give conceptual hint of the data center management. However, most of works are focus on individual part of data center management. The cross-devices data center management is seldom discussed. The concept of cross-devices management most appear in the smart home automation. We have also referenced some smart home automation to inspire our inspiration. The Yung-Wei Kao have proposed a system, which provide a semantic and user configurable interface to achieve home automation[20]. In the system, a SHPL (Semantic Home Process Language) was defined for describing semantic processes. In this thesis, we have inspired by the SHPL, and modified the SHPL to fit our management situation.

Section 2-2　Related work

In this section, we will introduce a work related to our system. The InfraPower software is the product implement by Austin Hughes Electronics Ltd, which is helping to control their PDU. We will introduce the software functionality , and the advantage and disadvantage are also discussed.

Section 2-2-1.　The InfraPower Software

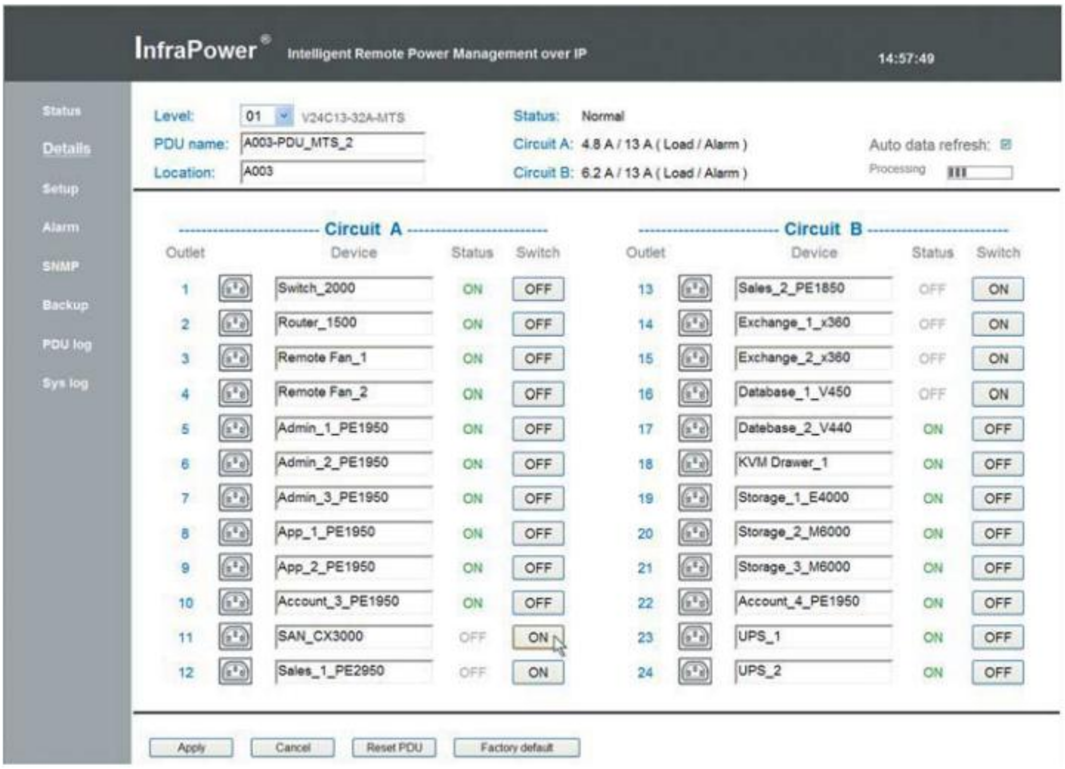

Figure 2.1 The system interface screenshot of InfraPower software

Because of the IP-based PDU is controlled by the controlling protocol, this software is providing a GUI to control the PDU. The software is implemented on Windows XP 32-bits version, and the Windows 7 version is not support currently. The system are encapsulated into a setup file, which will install the whole software files and PHP5 and Apache and PostgreSQL. A complete web-server are install into the system, and all of the program are executed by PHP5.

The interface provide the setting function, such that user can assign the IP address of the PDU . As shown in figure 2.1, the user can turn on or turn off each outlet by the interface. The current and voltage status can be seen in the interface also. In order to let user notify which outlet are over some voltage or current, there is also a alarm page provided. If alarm is occurred, there is a E-mail will be sent to the set mail address. The SNMP set and event log functionality are also provided.

In the conclusion, the software implements a easy using graphical user interface, which provides basic function of the IP-based PDU. The alarm function can notify user which outlet status is weird. The event log can record the weird situation of the PDU. However in the rack server enclosure, there are peripheral devices in the same rack. The system only support the function of PDU. Some events are correlated between peripheral devices, and it seems not enough for the management usage. The system only provide alarm function, but the alarm still need the staff to solve it.

Chapter 3 IP-Based Rack Server Enclosure Peripheral Devices

Figure 3.1 The picture of a IP-based lock and rack server enclosure

For this thesis, we adopt product and environment which provided by Austin Hughes Electronics Ltd. There is three types of device are adopt in the thesis, including IP-based rack server enclosure lock, IP-based fan and IP-based PDU.As shown in figure 3.1, a single rack server enclosure include some servers and fans and some power outlets providing the electric energy. Every single fan and outlet will corresponding to a computer server for cooling and electricity supplying. In our simulation, we support there are 5 computer servers in a rack server enclosure and 5 fans are used for cooling every server and 5 outlets are used provide server electric power.

(a) (b)

Figure 3.2 The picture of IP-based fan(a) and PDU(b)

The main functionality of the IP-based rack server enclosure lock(as shown in figure 3.1) includes that remote lock / unlock the electromagnetic lock, remote polling the status of rack server enclosure's door, remote add / remove a RFID card numbers. The main functionality of the IP-based PDU (as shown in figure 3.2 (b)) includes that turn on / off outlet, polling current reading. The main functionality of the IP-based fan (as shown in figure 3.2 (a)) includes that turn on / off fan, polling temperature reading, set fan speed. These device use the RS-485 as connection I/O interface, and can be linked by daisy chain connection. As shown in figure 3.3, the connection between device use RS-485 as the interface, and there is a device called IP-dongle which receive network protocol from RJ-45. The daisy chain between device can gather the same type device into a group, so that the number of IP-dongle can be reduced. After IP-dongle receive the network protocol from internet, it will bypass the protocol to the device. According to the protocol type, the device will do corresponding action.

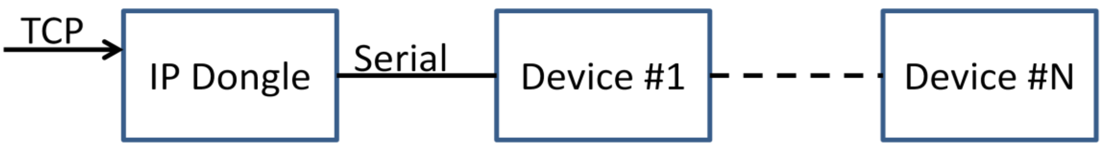

Figure 3.3 The connection of IP-based devices.

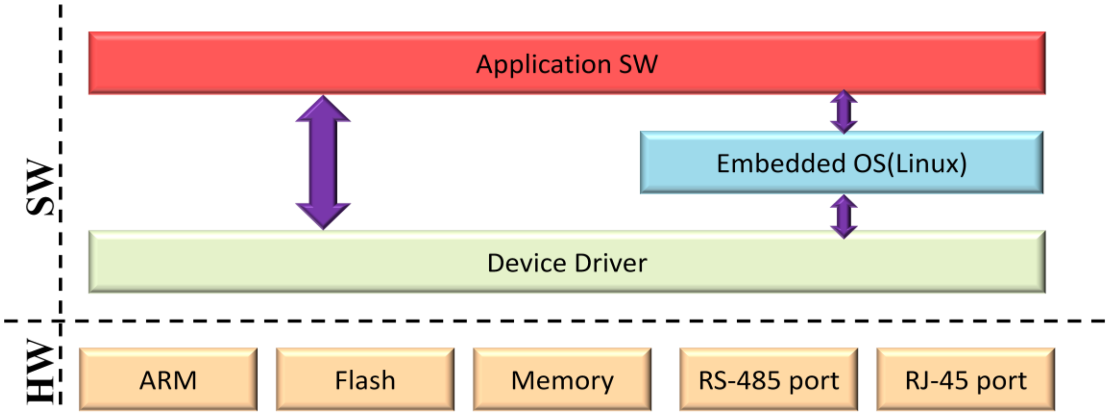

Figure 3.4 Hardware architecture of the embedded Lan-module

Obviously, the connection framework needs three IP-dongles to link lock and fan and PDU in a rack server enclosure. The three IP-dongles which have individual physical or virtual IP address will cause the waste of IP address and cost of IP-dongle when build data center environment. Therefore, we have helped to develop a new type of IP-dongle which integrate sensors and fans and PDU into a single embedded board. As shown in figure 3.4, the new type of IP-dongle which called Lan-module have some RS-485 ports to connect to fan and PDU. The RJ-45 port is designed for receive network protocols. The micro linux is ported on Lan-module for developing protocol parsing applications. The software architecture in embedded site is shown in figure 3.5. The main function of the Lan-module is that keep polling the linked device status and listening the network port for immediate command. There are sensors and outputs connected to Lan-module. The sensors include smoke sensor, water sensor which response weather there are fire or flood situation in the data center. The outputs include LED, Buzzer which can notify staff on the spot that there is some problem on the devices. For our simulation we have adopt this Lan-module as hardware network interface.

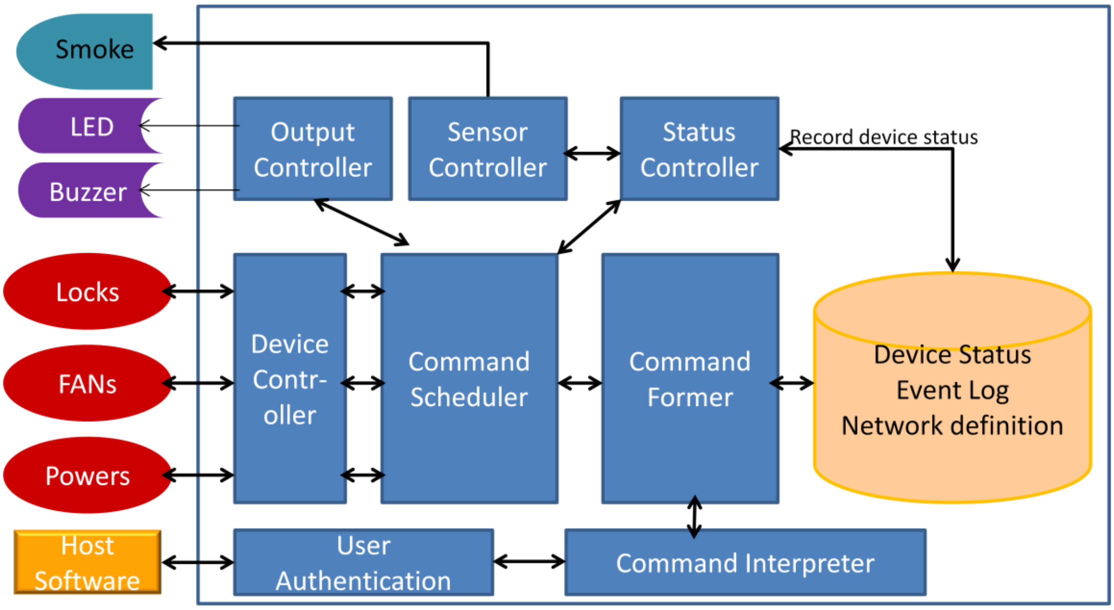

Figure 3.5 software architecture of the embedded Lan-module

Chapter 4 System Design and Implementation

In this chapter, the overview of this system which include the system hardware / software functionality and architecture is presented first. Each software component include detail implement and functionality will be introduce in section 4.2 to section 4.4.

Section 4-1 Overview

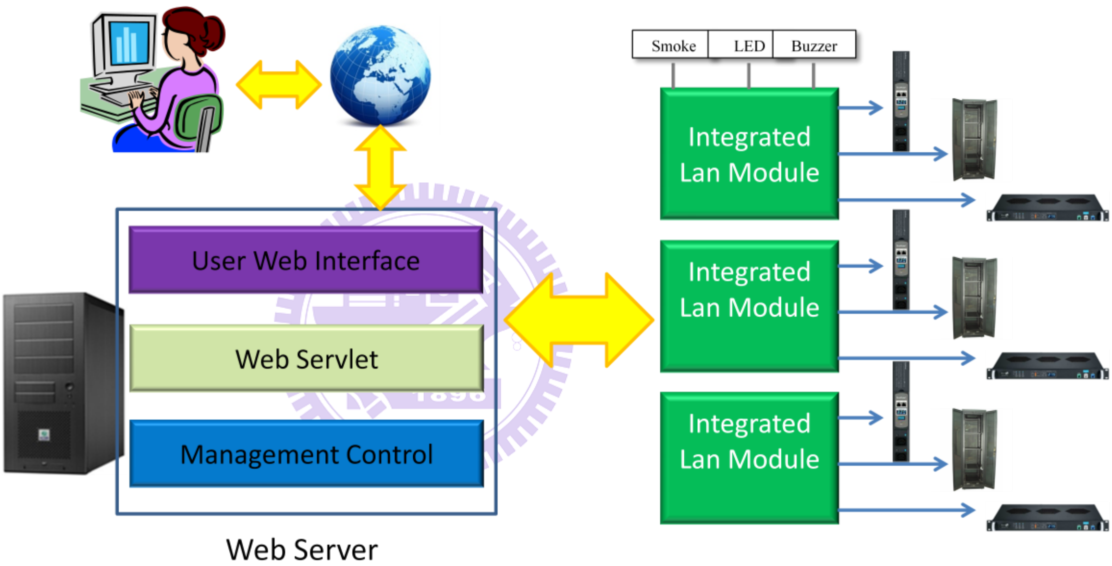

Figure 4.1 The overview of the purposed system

As shown in figure 4.1, the left hand site is the user interface and the right hand site is the peripheral devices. In the right hand site, the Lan-module integrate network and lock and fans and PDU and outputs and sensors. Each Lan-module link to a rack server enclosure, and there are several fans and PDU outlets and sensors and outputs connected to Lan-module also. In the left hand site, there is a web server response to main control functionality. The manager connect to the manage web server which have a web-based management interface implemented by JSP. Web interface using Java Script send HTTP request to servlet on the web server, and the servlet will run the control thread according to the reques message. As shown in figure 4.2 , there are four part control thread in the servlet. The first part is energy part which will control the overall server on / off status to satisfy some energy criteria. The second part is key part which monitor and control every RFID card number. The third part is automatic control part which monitor server and peripheral status according to the semantic settings. If the status and condition are satisfied, the predefined action will be performed. The final part is polling part which simply polling the Lan-module and record the device status into web server database.

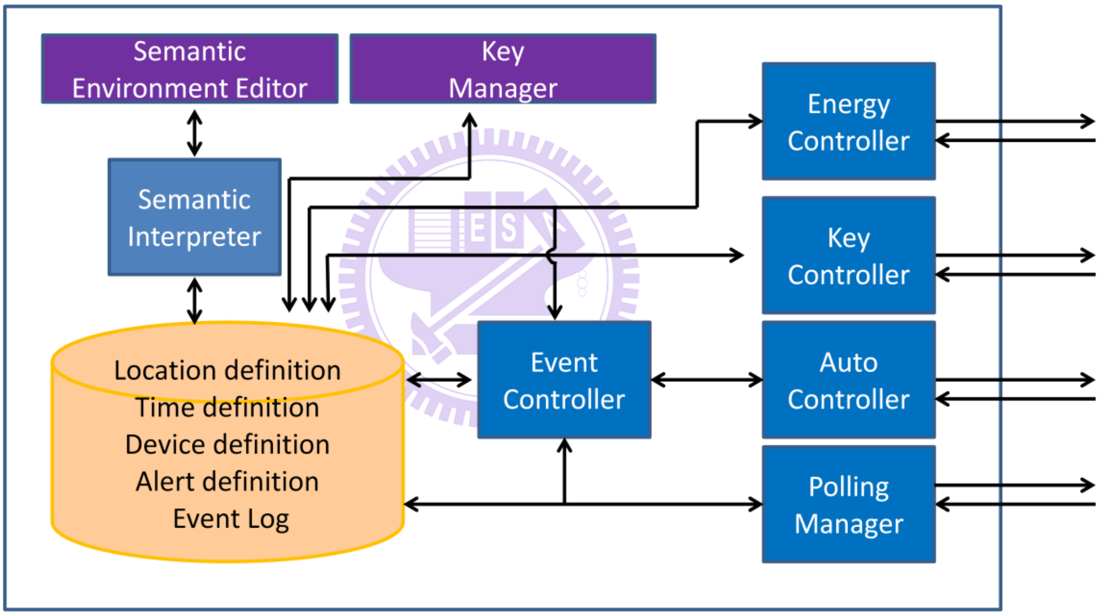

Figure 4.2 The control server software architecture

Section 4-2 Semantic Automatic Control

```
public class AutoModel {
    public AutoModel(){
        ID = -1;
        TargetDevice = "";
        TargetLocation = "";
        TargetStatus = "";
        SourceDevice = "";
        SourceLocation = "";
        Time = "";
        SourceStatus = "";
        SourceCondiction = "";
        Action = "";
    }
    public int ID;
    public String TargetDevice;
    public String TargetLocation;
    public String TargetStatus;
    public String SourceDevice;
    public String SourceLocation;
    public String Time;
    public String SourceStatus;
    public String SourceCondiction;
    public String Action;
}
```

Figure 4.3 The model of the automatic command in control server

As shown in figure 4.3, there are nine major components in the semantic automatic control thread. The TargetDevice component define this control command will affect which device. With TargetDevice component, we can define which device we will do action instead of doing action to the specified device. The TargetLocation component define the target device at where, and we can easy control device at different locations. The third component is TargetStatus which define the device under what status will do the user defined action. The SourceDevice ,SourceLocation and SourceStatus are define what device at where under what status will trigger the automatic control thread to do action. The time states when this automatic control action will be checked. Sometime it is hard to use words to describe the situation of a device, but the number is much easy to understand. The SourceCondiction is used to define the variable in devices, ex temperature, current...etc. With SourceCondiction, we can more easy to define the device's weather need to be controlled.

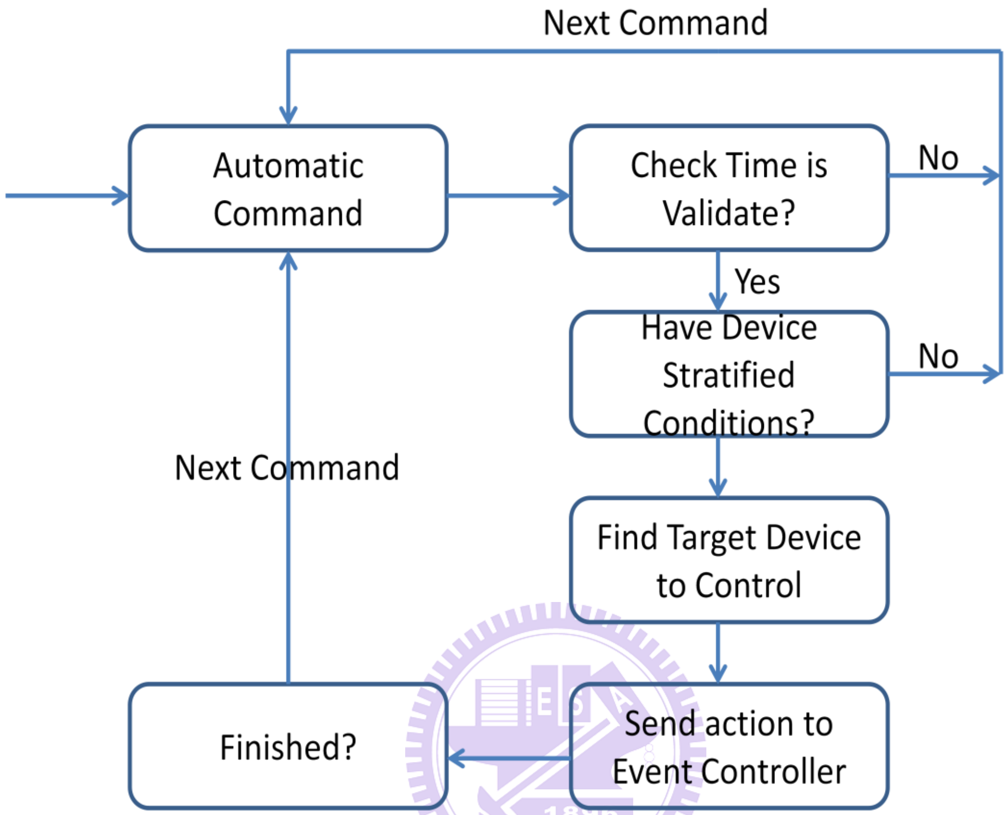

Figure 4.4 Execution diagram of the automatic command

As shown in figure 4.4, an automatic command will pass four main check process. First the time string will be check the time string is a period or a specific point in time. If the time string is a period, the system time will be checked whether in the period or not. If the time string is a specific point in time, the system time will be compared whether the same as time string. If the time is valid, the database will be searched according to the source device type, source device location, source device status and source device condition. If there is any device satisfied all the condition, the device which need to be control will be searched from database according to target device type and target location. Finally, the event controller will do corresponding action to target devices. On the path, if there is any condition which is not satisfied, it will back to the start point and check the next automatic command.

Section 4-3 Energy control

There are two parts in the energy control. The first part is the server maintain rotation part which will keep a group of server under maintaining. The second part is the standby ratio part which will control the online standby server percentage.

```
package xjxj.model;

public class MaintainModel {
    public MaintainModel(){
        ID = -1;
        Period = -1;
        RestGroup = -1;
        StartTime = "";
        EndTime = "";
    }
    public int ID;
    public int Period;
    public int RestGroup;
    public String StartTime;
    public String EndTime;
}
```

Figure 4.5 The model of the maintaining functionality in control server

As shown in figure 4.5, there is 4 objects in the maintaining part. First, the period variable record that how long a group can stay in maintaining. Second, the rest group variable record which group is under maintaining. Then start time variable record the beginning time of current maintaining group. Final, the end time variable record the ending time of current maintaining group. With these 4 objects, we will compare the ending time with system time. If the system time equals to the ending time, we will turn on the group servers and fans. After turning on, we will switch the rest group to the next, and turn off the server and fans in next group. Finally, the starting time will be recorded, and the ending will be starting time add the period time. The ending time will be checked recursively, so the server in each group will take a rest alternately. Using this maintain function, the staff in the data center can use the group maintaining time to do some regular hardware checking.

```java
package xjxj.model;

public class StandbyModel {
    public StandbyModel(){
        ID = -1;
        UpperRate = -1;
        lowerRate = -1;
        Criteria = -1;
        Time = "";
        Location = "";
    }
    public int ID;
    public int UpperRate;
    public int lowerRate;
    public int Criteria;
    public String Time;
    public String Location;
}
```

Figure 4.6 The model of the standby ratio functionality in control server

As shown in figure 4.6, there are 5 objects in the standby ratio part. First, the UpperRate variable record the upper bound of the standby server percentage in all server number. Second, the LowerRate variable record the lower bound of the standby server percentage in all server number. The Criteria variable decide the server closing order. The time variable record the effective time for this setting. Final, the location variable record the effective time for this setting.

$$ratio_{\text{standby}} = \frac{n_{\text{standby}}}{n_{all}} \quad (4.1)$$

First, we will compute the $ratio_{\text{standby}}$ value in the (3.1). The n_{standby} and n_{all} in the (3.1) stand for the number of server which status is standby and the number of all servers respectively. After the $ratio_{\text{standby}}$ is calculated, the value will be compared to the UpperRate and LowerRate variable. If the value of $ratio_{\text{standby}}$ is larger than UpperRate, we will count how many servers need to closed such that the value can lower than UpperRate. We will also search a list of standby server, then we arrange the list according to the criteria. If the criteria is for long-term, the server which have longest turn-on time will be listed in the top of the list, and the second longest turn-on time will be listed in the second place of the list and so on. If the criteria is for concentrate, the server which have lowest turn-on time will be listed in the top of the list, and the second lowest turn-on time will be listed in the second place of the list and so on. After the arrangement of the standby server list, the number of servers which we have calculate before will be closed from top of the list. In the other hand, if the value of $ratio_{standby}$ is smaller than LowerRate, we will count how many servers need to turn on such that the value can larger than LowerRate. We will also search a list of server which is off-line and it is not in the rest group . After the searching the list, the number of servers which we have calculate before will be turned on from top of the list. Using this standby ratio function, we can keep the number of server which turn on and wait for request resources. The electric energy will be saved, and the ability of data center is almost the same.

Section 4-4 Key control

```
package xjxj.model;

public class KeyModel {
    public KeyModel(){
        ID = -1;
        RFIDNumber = -1;
        AuthLevel = -1;
        effectiveLocation = "";
        effectiveTime = "";
        Added = -1;
    }
    public int ID;
    public int RFIDNumber;
    public int AuthLevel;
    public String effectiveLocation;
    public String effectiveTime;
    public int Added;
}
```

Figure 4.7 The model of the key management functionality in control server

As shown in figure 4.7, there is 5 objects in the key control part. First, the RFIDNumber variable record each RFID number in the database. Second, the "AuthLevel" variable record the access level of RFID number . Then effective location variable record the lock location the RFID number can access. The effective time variable record valid period of the RFID number. Finally, the added variable is used to record the RFID number is added into proper locks.

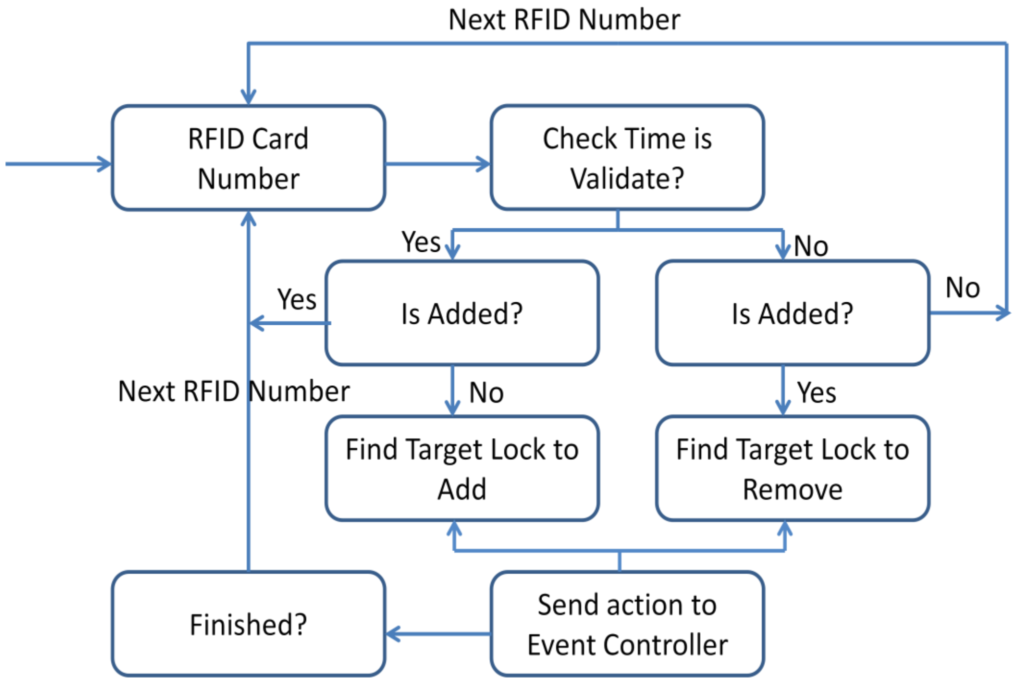

Figure 4.8 The key checking process execution diagram

With these 5 objects, we will control all the RFID card number add into proper locks and remove from locks when the time is invalid. As shown in figure 4.8, we will check every RFID card number in the database. At the beginning, we will compare the RFID time with system time. If the time is in the valid period, we will check this card number is added into the locks or not. If the card is not added into locks, we will find the proper locks according to lock's location and access level. Then the RFID card number will be added into the locks. If the card is added already, the next RFID card number will be checked. In the other hand, if the RFID card number is not in the valid time, we will also check this card is added into locks or not by the "Added" variable in the database. All of the locks location and access level will be compared to the "AuthLevel" variable and effective location variable in the database. Then the locks which this RFID card number is added will be listed. After, the RFID card number will be remove from these locks using event controller. The RFID card number which is not in valid time period and not added into locks also will be neglected.

Chapter 5 System Demonstration

This chapter will demonstrate the system how to work, and the system screen shot is also provided for easy understanding. The section 5-1 will talk about the basis of semantic language. The section 5-2 will talk about how to use the web interface to control the whole rack servers and its peripheral devices, using the semantic command structure talked in section 5-1. The large scale rack server enclosure simulator is present in section 5-3.

Section 5-1 Semantic Control Process Language

Section 5-1-1. Structure of semantic control process language

Since the semantic control is not existed in current system, we have referenced the SHPL[20] to support semantic process definition. Some of details of the SHPL are modified for fitting our system. An entire string input command is also provide for more flexible command format, and some keywords are preserved for easy interpreting the command to machine understandable language.

From our observation, six vital factors are usually included in rack server enclosure peripheral device automation processes: trigger device, target device, action, status, condition, execution time, and location. For example, given a command "if the server temperature is higher than 60 ℃ during 6:00 ~ 18:00, turn the fan to max", the server is a trigger device; "temperature is higher than 60 ℃" is a condition; 6:00 ~ 18:00 is an execution time; the fan is the target device, and "turn to max" is an action.

The command string will be separated into three parts by three keywords, that is "If", "Target" and " when time". The sentence from the keywords "If" to another two keywords, "Target" and " when time", or end of command will be consider as the description of the trigger device. In the other hand, the sentence from the keywords "target" to another two keywords or end of command, will be consider as the description of the target device. The sentence from the keywords " when time " to another two keywords, "Target" and "If", or end of command will be consider as the execution time. For example, the valid command " If FANS at 2F status LOW satisfied TEMP>40 do HIGH to target sourcedevice when time 00:00:00~18:00:00", the " If FANS at 2F status LOW satisfied TEMP>40 do HIGH" will be considered as trigger device part; the "target sourcedevice" will be considered as target device part; and the "when time 00:00:00~18:00:00" will be considered as execution time part. Once the whole semantic command is separated into three part, that is description of the trigger device part, description of the target device part and execution time part, we will further parse these three parts independently. The execution time part is mostly easy to understand, because there is only a time right after the keywords "when time". There are some keywords further describing the trigger device and target device. The word after the keyword "at" will be regarded as the location of trigger device or target device. The word after the keyword "status" will be regarded as the status of trigger device or target device. The word after the keyword " satisfied" will be regarded as the condition of trigger device or target device. Because of the English grammar often said "If something do some action to something", we put the action variable into description of trigger device part. The word after the keyword " do" will be regarded as the action doing to the target device. In the whole semantic command, there are only two keywords that are necessary. The keyword can be appear in semantic, but which is not appeared will be regarded as don't care condition. In other words, we consider all the device are satisfied the condition if the condition is unassigned. For flexibility, the case of every word in the command sentence will be transform to upper case, so the case of word will not affect the result of parsing. Besides, in our parsing flow the order of keywords in the same part will not affect the parsing result. In other words, the command " If FANS at 2F status LOW satisfied TEMP>40 do HIGH to target FANS at 2F status LOW when time 00:00:00~18:00:00 " and " If FANS status LOW at 2F satisfied TEMP>40 do HIGH to target FANS at 2F status LOW when time 00:00:00~18:00:00 " will get the same parsing result. Moreover, the order of each part is also not affect parsing result. In other words, the command "when time 00:00:00~18:00:00 target FANS at 2F status LOW if FANS at 2F status LOW satisfied TEMP>40 do HIGH" and " If FANS status LOW at 2F satisfied TEMP>40 do HIGH to target FANS at 2F status LOW when time 00:00:00~18:00:00 " will get the same parsing result.

Token	Is keyword?	Is necessary?
If	Yes	Yes
FANS	No	Yes
at	Yes	No
2F	No	No
status	Yes	No
LOW	No	No
satisfied	Yes	No
TEMP>20	No	No
do	Yes	No
HIGH	No	No
target	Yes	Yes
FANS	No	Yes
at	Yes	No
status	No	No
LOW	Yes	No

when time	Yes	No
00:00:00~08:00:00	No	No

Table 5.1 Analysis of the automatic command by tokens

So far, we have talked how the execution time and description of trigger / target device are parsed from the semantic command. For easier understanding the structure of the semantic command, another valid semantic command " If FANS at 2F status LOW satisfied TEMP>40 do HIGH to target FANS at 2F status LOW when time 00:00:00~18:00:00 " is provided. The meaning table of each token of the example is also provided in table 5.1. As shown in table 5.1, the first column in the table stand for each token separated by blanks. The second column in the table indicate the each token is a keyword or not. If it is a keyword, there must be a variable which describe the keyword condition following the keyword. The third column in the table stand for the token is necessary or not. If the token is not necessary, the semantic command without the token can be also parsed correctly and considering the keyword as don't care.

Section 5-1-2. Keyword definition of semantic control process language

Up to present, the structure and parsing of semantic command are discussed. The description format of each keyword will be talked in the following.

Device Type	Acceptable Name	Description
Fan	Fans, Fan	The fans in the rack server enclosure
Lock	Locks, Lock, Door, Doors	The lock outside the rack server enclosure

PDU	PDU, PDUs, Power, Powers, Outlets, Outlet	The outlets in the rack server enclosure
Server	Computer, Computers, Servers, Server	The servers in the rack server enclosure
Source Device	SourceDevice	indicate the source devices
Corresponding Computer	Corr_Computer	Indicate the correspondent computers of PDU or fan
Corresponding Fan	Corr_fan	Indicate the correspondent fan of server
Corresponding Outlet	Corr_pdu	Indicate the correspondent outlet of server

Table 5.2 Devices and valid name used in semantic automatic command

As shown in table 5.2, we have listed the name and its meaning which is use to define the trigger / target device in the semantic command. As shown in the second column of table 5.2, there are many string can map to the same device, so user can remember the device name easier. Sometime, we hope to do some action to the device which trigger the event, so we designed the "source device" device type for easy using. Moreover, if we want to do action to the server which corresponding to the fan and the outlet event, we designed the " Corresponding Computer " device type for easy handling the server.

Device Type	Status
Fan	Off , Max, High, Low, Disconnected
Lock	FrontOpened, RearOpened, BothOpened, BothClosed, FrontAlarmOpened, RearAlarmOpened, BothAlarmOpened, Disconnected
PDU	On, Off, Disconnected
Server	Off, High, Normal, OnStandby, Monitoring, WaitHandle

Table 5.3 Statuses of each device used in semantic automatic command

In the thesis, we have defined some status to record the situation of peripheral devices and servers. As shown in the table 5.3, the table shows the status we have defined for each peripheral devices. The fan has 5 statuses, Off , Max, High, Low, Disconnected. The off status stand for the power of fan is turning off; the max, high, low statuses are the three different speed of fan; when the connection of fan was undetected, the status of fan will be marked as disconnected. The lock has 8 statuses, that are FrontOpened, RearOpened, BothOpened, BothClosed, FrontAlarmOpened, RearAlarmOpened, BothAlarmOpened and Disconnected. The FrontOpened status stand for the front door is authorized opened and the rear door is closed; The RearOpened status stand for the rear door is authorized opened and the front door is closed; The BothOpened status stand for the front and rear door are also authorized opened; The BothClosed status stand for the front and rear door are also closed; The

FrontAlarmOpened and RearAlarmOpened and BothAlarmOpened status stand for the front door is unauthorized opened and the rear door is unauthorized opened and both door are unauthorized opened respectively; when the connection of lock was undetected, the status of lock will be marked as disconnected. The PDU has 3 statuses, that are On, Off, Disconnected. The on status stand for the outlet is turning on; The off status stand for the outlet is turning off; when the connection of PDU was undetected, the status of PDU will be marked as disconnected. Finally, there are 6 different statuses for server status, that is Off, High, Normal, OnStandby, Monitoring, WaitHandle. The off status stand for the power of server is turning off; the High, Normal, OnStandby statuses stand for the three different rate of utilization, which are highly used, normal used and almost not used respectively.

Device Type	Actions
Fan	On, Off, Max, High, Low, Reboot, BuzzerOn, BuzzerOff, LEDOn, LEDOff
Lock	OpenFront, OpenRear, CloseFront, CloseRear
PDU	On, Off, Reboot, BuzzerOn, BuzzerOff, LEDOn, LEDOff
Server	Off, Reboot, SetHigh, SetNormal, SetOnStandby

Table 5.4 Valid action of each device used in semantic automatic command

In our system, it is important to define what action will be done. The table 5.4 shows the valid action list for each device type. There are 10 actions for the fan device, that is On, Off, Max, High, Low, Reboot, BuzzerOn, BuzzerOff, LEDOn, LEDOff.

The on and off actions stand for the action of turning on / off the fan power; the max, high and low action stand for setting the fan speed to maxima or high or low respectively. There are 7 actions for the PDU device, that is On, Off, Reboot, BuzzerOn, BuzzerOff, LEDOn, LEDOff. The on and off actions stand for the action of turning on / off the outlet power; the reboot action stand for the action of turning off then turning on the outlet power. The BuzzerOn, BuzzerOff, LEDOn, LEDOff actions defined in fan and PDU are used to control the notification unit linked to Lan-module; the BuzzerOn, BuzzerOff actions stand for the on / off switch of the buzzer on the Lan-module; the LEDOn, LEDOff actions stand for the on / off switch of the LED on the Lan-module.

So far, the semantic structure, device type, status and action have introduced. Examples and its explanation for valid semantic control command are also provided. Then the next section will introduce how to use the semantic command in the web interface.

Section 5-2 Web Application

Figure 5.1 The web-based interface of the index page

In the web interface, there are two major block in the interface, that is function menu block and main function block. The function menu block shown in blue frame in figure 5.1 is used to select the control function that you want to do. The main function block shown in red frame in figure 5.1 is used to display the functional interface. For example, if you want control the RFID keys you will click on the "Key Manager", then the functional interface will show in the main function block. Each functional interface will be introduced in the following.

Section 5-2-1. Semantic Automatic Control Interface

Figure 5.2 The web-based interface of the semantic automatic comtrol page

The first part we are going to introduce is the semantic automatic control which is clicked on the "Controls" in the function menu block. As shown in figure 5.2, there is a text field for inputting the semantic command. The semantic command was composed from some objects and keywords. With these keywords, the command which is easy understanding can be parse into the database structure. In the previous section 5.1, we have introduced the structure and every component of the semantic automatic command. If the command format is correct, the server response row will show a "Added OK" message. Otherwise a "Add Fail. Incorrect format" message will be shown in server response row. Once the command is parsed and store into the database, the command will be run correctly as we have discuss in the section 3-2.

Section 5-2-2. Key Management Interface

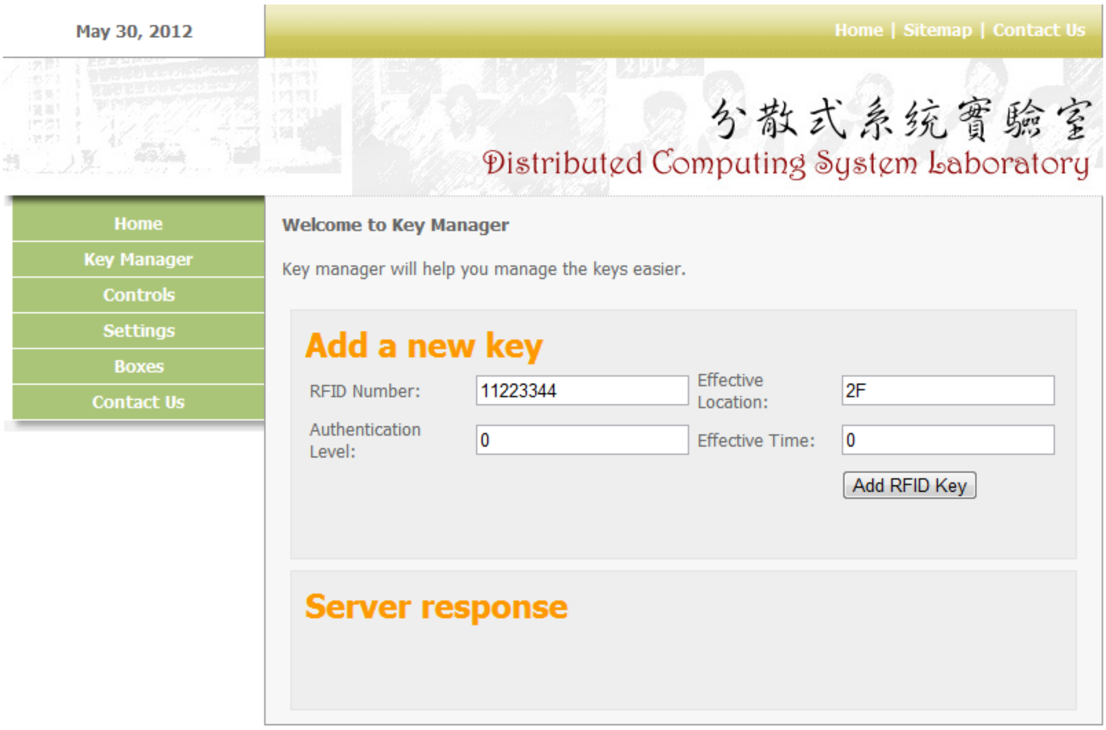

Figure 5.3 The web-based interface of the key management page

The second part we are going to introduce is the key management function which is clicked on the "Key Manager" in the function menu block. As shown in figure 5.3, there is four text field for inputting the key's data. The RFID Number text field is the number will be stored into database; the effective location text field is the location of lock that the RFID card number can access; the authentication level text field stand for the access level of the RFID card number. The lesser the authentication level is the higher access authority. The effective time field is the duration that the RFID card number is valid. The time string format "HH:MM:SS~hh:mm:ss" is acceptable by the interface. The HH,MM and SS stand for the starting hours, starting minutes and starting seconds. The hh,mm and ss stand for the ending hours, starting minutes and starting seconds. There is a '~' symbol between staring time and ending time. The special string "Any" is also valid which stand for we don't care the time. Once the RFID card number information is finished, we can click the "Add RFID Key" bottom to add the RFID key. Then the system will check the key format and store the key information into the database. If the key format is correct, the server response row will show a "Added OK" message. Otherwise a "Add Fail. Incorrect format" message will be shown in server response row. Once the key is stored into database, the key will be added / removed into proper locks as we have discuss in the section 3-4

Section 5-2-3. Energy Control Interface

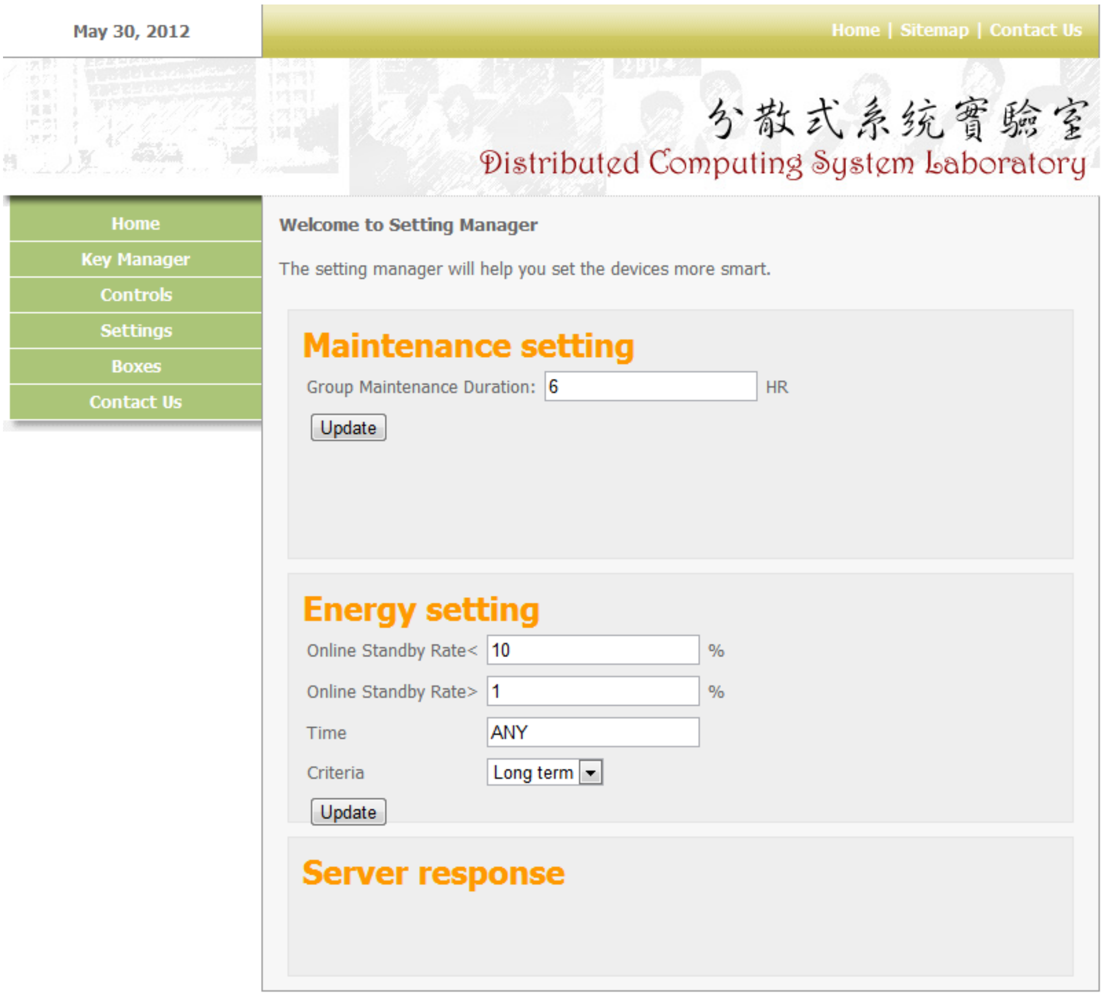

Figure 5.4 The web-based interface of the energy control page

The third part we are going to introduce is the energy function which is clicked on the "Settings" in the function menu block. As shown in figure 5.4, the energy function has two parts, maintaining part and standby ratio part.

We will introduce the maintaining part firstly. There is only one text field for inputting the maintaining data. The "Group Maintenance Duration" text field is the duration of hours that every group rack server enclosure will spend in regular hardware maintenance. Once the maintenance information is finished, we can click the "Update" bottom to update the new maintenance settings. Then the system will check the maintenance format and store the maintenance information into the database. If the maintenance format is correct, the server response row will show a "Added OK" message. Otherwise a "Add Fail. Incorrect format" message will be shown in server response row. The system will turn on /off proper server and its peripheral devices as we have discuss in the preceding part of section 3-6.

The second we will introduce the standby ratio part. There is three text field and one dropdown list for setting the standby ratio data. The first and second text field are the upper bound and lower bound of the standby ratio. The third text field are the effective time of this setting. That is only current time in the setting time duration, the standby server ratio is under controlled. The time string format "HH:MM:SS~hh:mm:ss" is acceptable by the interface. The HH,MM and SS stand for the starting hours, starting minutes and starting seconds. The hh,mm and ss stand for the ending hours, starting minutes and starting seconds. There is a '~' symbol between staring time and ending time. The special string "Any" is also valid which stand for we don't care the time. The "Criteria" dropdown list is used to select the closing computer criteria, and the more detail is shown in section 3-6. Once the standby ratio information is finished, we can click the "Update" bottom to update the new standby ratio settings. Then the system will check the standby ratio format and store the standby ratio information into the database. If the standby ratio format is correct, the server response row will show a " Update OK " message. Otherwise a " Update Fail. Incorrect format" message will be shown in server response row. The system will turn on /off proper server such that servers will between the standby upper bound and lower bound as we have discuss in the rest part of section 3-6.

Section 5-2-4. Lan-Module Display Interface

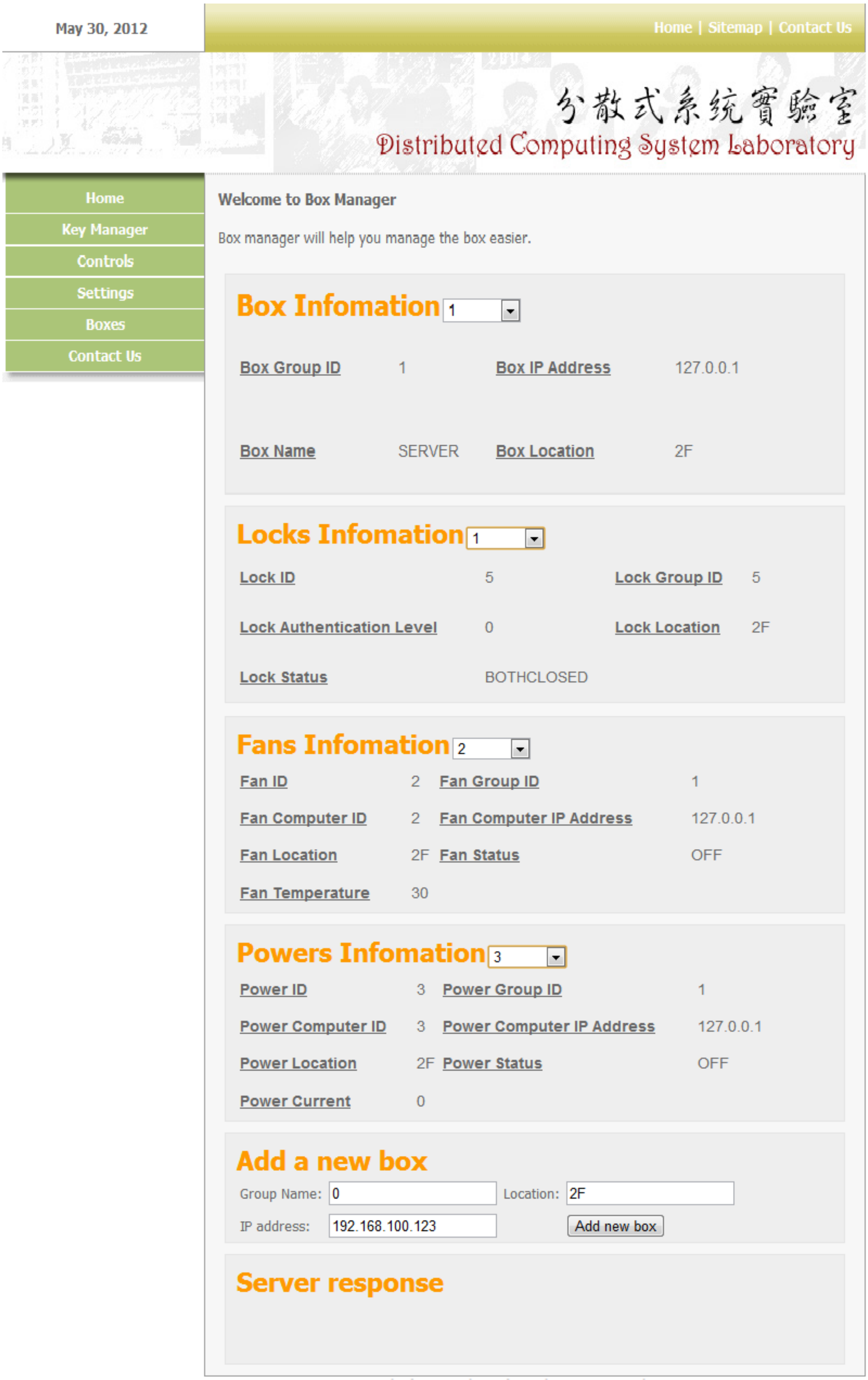

Figure 5.5 The web-based interface of the device information page

In the system, there is also a server information display interface which is clicked on the "Boxes" in the function menu block. As shown in figure 5.5, there are four columns in the interface. The box information column shows the information including the group id, the IP address, the name, the location of Lan-module. We can use the dropdown list near the column title to select which Lan-module information we want to see. After select the Lan-module, there are three dropdown list appeared near the locks and fans and powers information column title. The dropdown list can help us pick the devices linked with the selected Lan-module. With this display interface, we can clearly observe every device's information on the web interface.

Section 5-3 Rack Server Enclosure Simulator

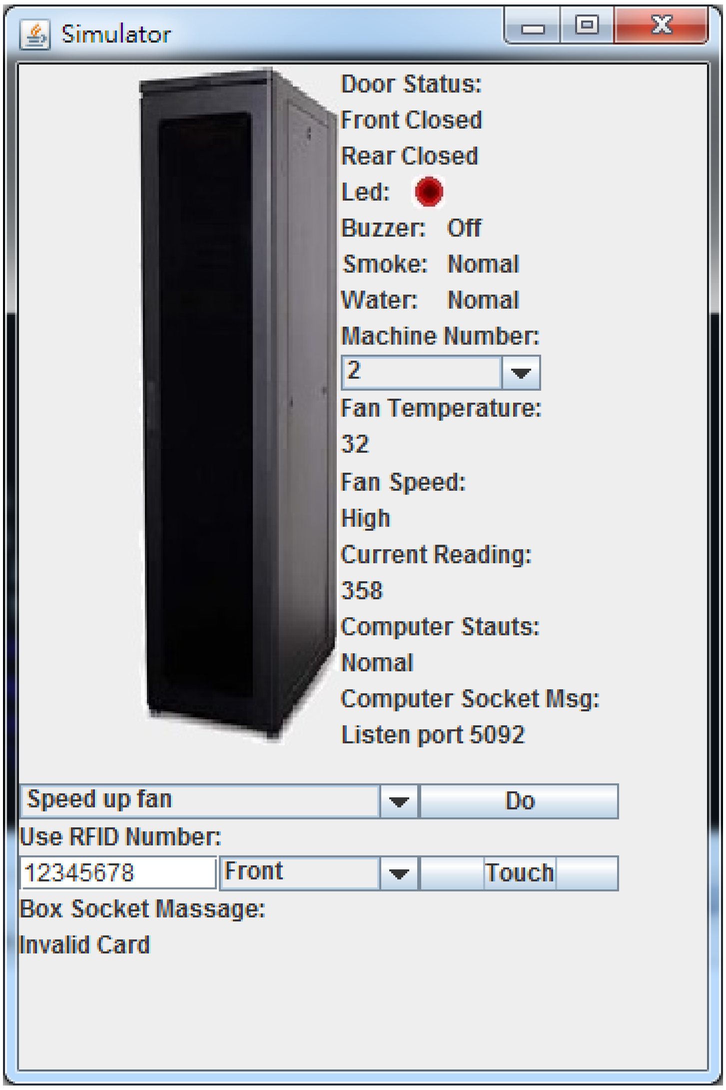

Figure 5.6 The normal status screenshot of the simulation

Since it is hard to find so many rack server enclosures and peripheral devices for testing. We have implemented a simulator in order to check the functionality and do experiments. The screenshot of the simulator is shown in figure 5.6. We can see a rack server enclosure on the left-top of the simulator, which will display the door situation of the rack server enclosure. There are many information displayed at the right site of the simulator. The door statuses are display at the second and third row at the right site and the output and sensor statuses are also provided. A set of device include a fan, a server and an outlet. The machine number dropdown list can select which set of device you want to observe. Once the machine number are selected, all information of the set of device is displayed below the dropdown list. In order to more like there is a rack server enclosure just in front of you, a action dropdown list just below the picture of rack server enclosure. We can use the action dropdown list to achieve some real action, such that you just stand in front of a rack server enclosure. The RFID card touching functionality is also provided. You can simulate the situation of opening the door use the valid card.

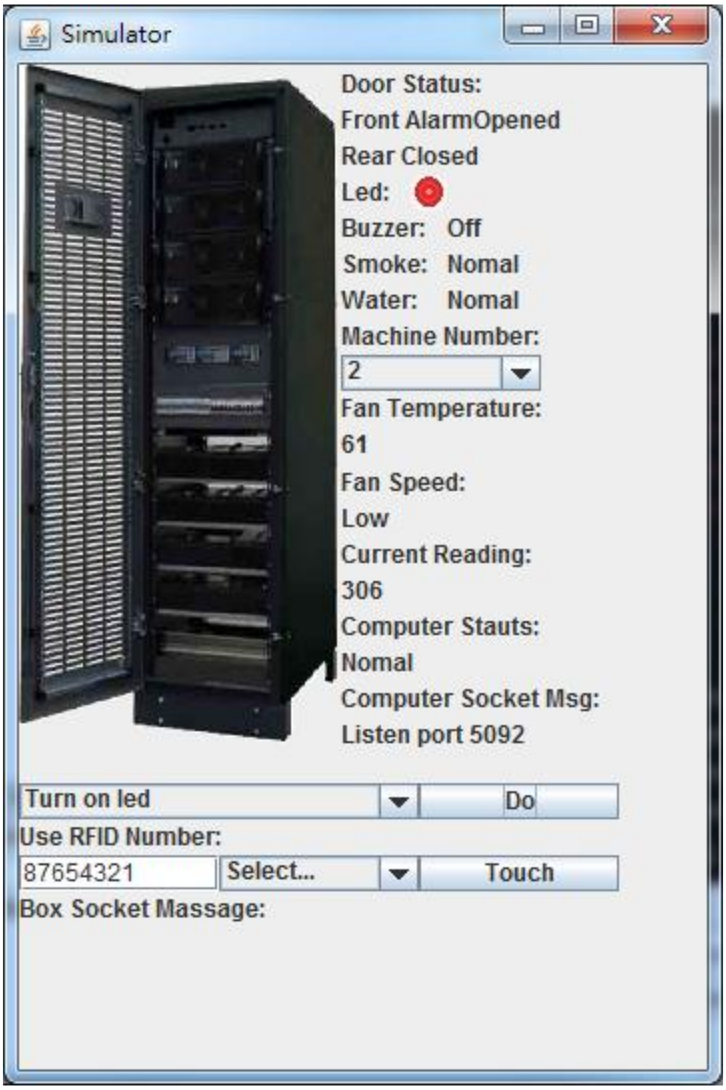

Figure 5.7 The open status screenshot of the simulation

As shown in figure 5.7 ,in order to more close to the reality, the picture of rack server enclosure and LED will be replaced according the server situation.

Chapter 6 Experiments

In this chapter, the test environment is given, and the power consuming observation and experiment are provided. We can see there is about 39% power saving in the purposed system.

Section 6-1 Test Environment

The power consuming testing server environment is in the following:

- OS: Win 7 32-bits version.
- CPU: Intel Core 3 Dual 6300 1.86 dual core
- HDD: Hitach HDS721616PCA380 ATA
- Memory: 2G DDR2 800

Section 6-2 Power Consuming Experiment

In this section, we have tested the electric power consuming under the environment described in section 6.1.

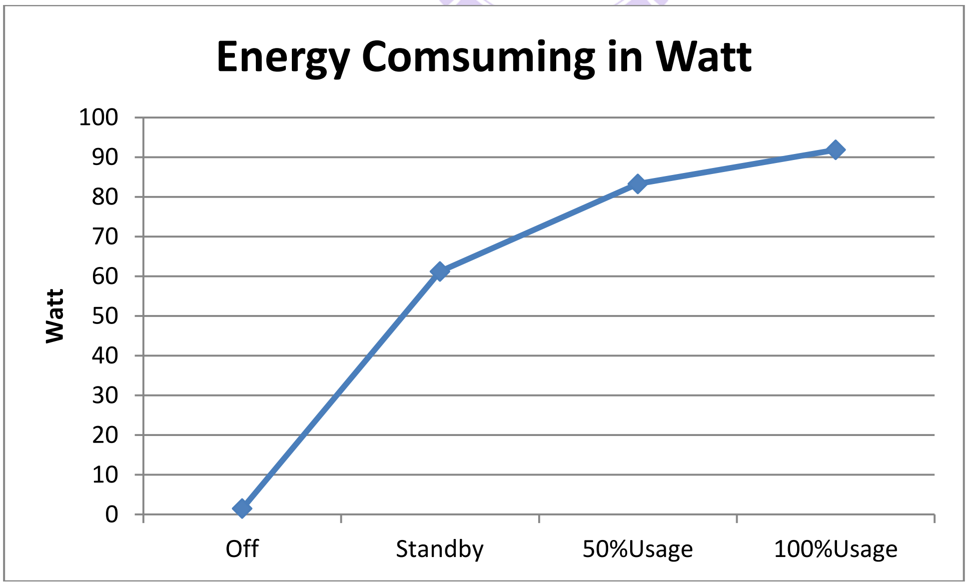

Figure 6.1 The watt value under different server utilities

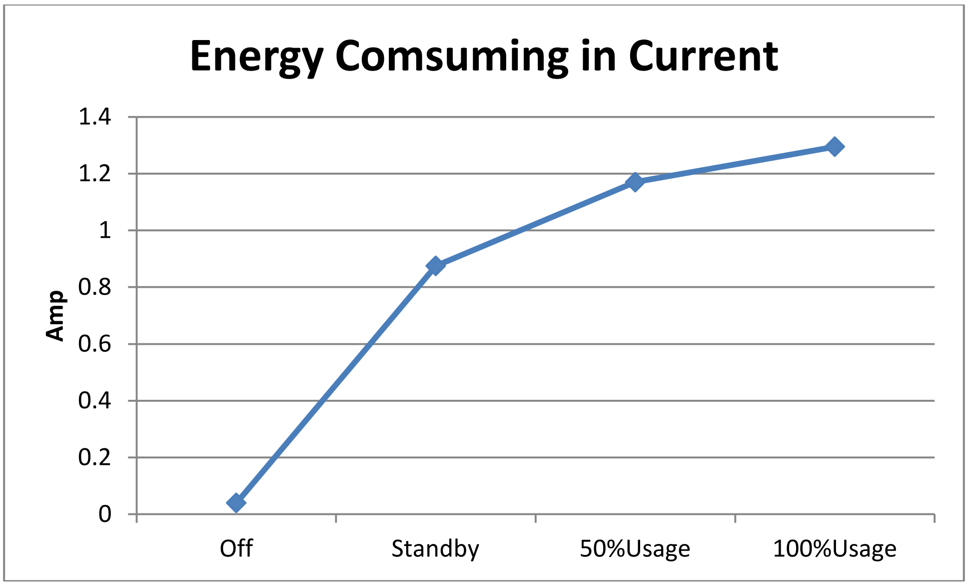

Figure 6.2 The current value under different server utilities

In our thesis, we have define the computer status in the different power consuming situation. The watt is a derived unit of power in the International System of Units (SI), and it's the most used unit of electric energy consuming. According to the Ohm's Law in the electric physics, we know the correlation between watt and current in the function(6.1). The W denotes watt and the J is the joule, and the S is the second. The watt defined as joules per second, we can also get it by the product of voltage and current.

$$W = \frac{J}{S} = V \bullet A \qquad \text{(6.1)}$$

In our assumption, there is correlation between server utility and the current value. The figure 6.2 shows that the higher server utility the higher current value. Compare to the figure 6.1 and figure 6.2, the curve is almost the same. We can reprove the relation between watt and current in the function (6.1)

Section 6-3 Simulation of the rack server enclosure

In this section, we will use the simulation introduced in section 5-3 to experiment the 100 servers' electric power consuming in different standby ratio.

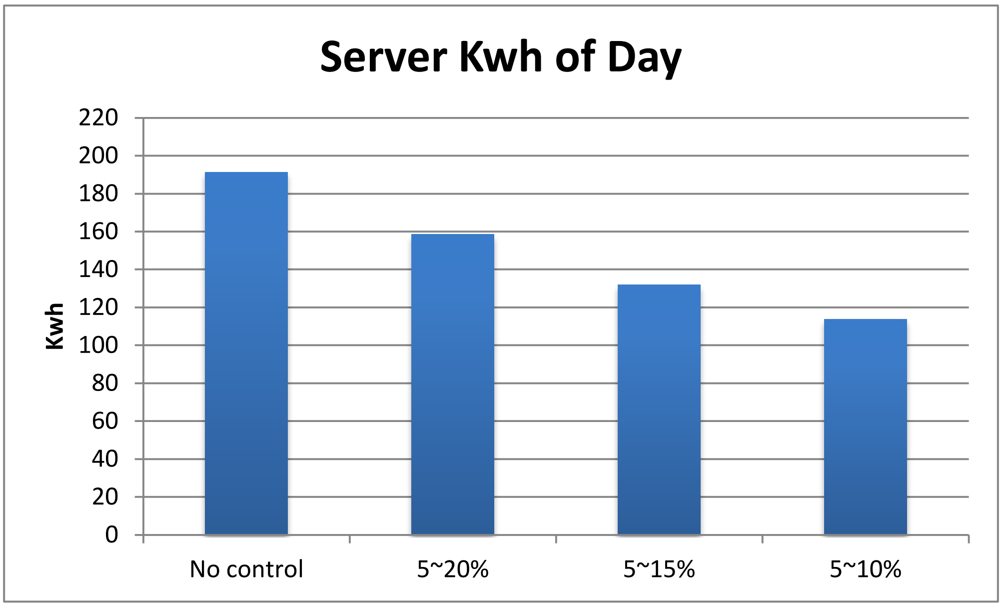

Figure 6.3 The kWh value of server under different standby ratio setting

In the experiment, without loss of generality we assume that the initial percentage of standby server, normally used server and highly used server are 30%, 50%, 20% respectively. Each server will follow the normal distribution with zero mean and standard deviation equals 3. Server will add or reduce the utility according the normal distribution and the correlations of utility of server are independent. We use the standby ratio functionality introduced in section 4-3, and record the kWh value of the different upper bound value. As shown in figure 6.3, if we do not use the standby ratio control, the kWh per day will be 191 in our simulation. If we set the standby ratio upper bound as 20%, 15% and 10%, the kWh per day will be 158, 131 and 113 respectively in our simulation. In the simulation, we can see that the electric power consuming will be reduced when the standby ratio functionality is used. There is about 40% electric power saving compared to without our system when the standby ratio is set as 10%. In the thesis, the simulation scale is much smaller than the real data center. If the number of servers increased as ten or hundred times, the saving in cost of data center owner will be significant.

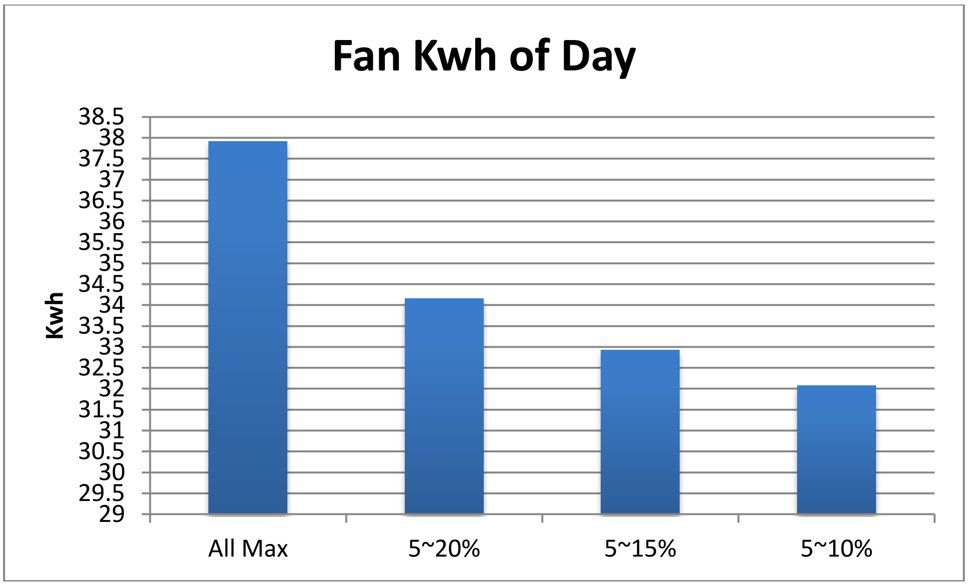

Figure 6.4 The kWh value of fan under different standby ratio setting

The second simulation is the fan KWh experiment. We use the semantic automatic control function introduced in section 4-2 to control the speed of fan. We set the fan speed to max if its corresponding computer status is high, and fan speed to high if its corresponding computer status is normal. We also set the fan speed to low if its corresponding computer status is standby. If the computer is off, the corresponding fan will be turn off also. As shown in figure 6.4, if we set all fan speed to max, the kWh of fans per day will be about 38 in our simulation. If we dynamic control the fan speed, the kWh per day will be about 34, 33 and 32 respectively in our simulation. In the simulation, we can see that the electric power consuming will be reduced when the dynamic controlling of the fan speed is used. In the thesis, the simulation scale is much smaller than the real data center. If the number of servers increased as ten or hundred times, the saving in cost of data center owner will be also significant.

Chapter 7 Conclusion and Future Work

Section 7-1 Conclusion and Discussion

In the thesis, we have implement a semantic automatic rack server enclosure management system. The system provides a user-configurable interface to control the rack server and its peripheral devices by semantic input in the web-site. The system can achieve cross-device management between lot of servers and peripheral devices in data center. The devices and server status and condition will be recorded in the web server. Server status is be predicted according to its electric power consuming, and corresponding action will be performed. Furthermore, The data center always use the air-condition system to cool the heat generated by servers. However if we use the fan more wisely, the usage of air-condition can be reduced. The dynamic control of fan speed can achieve this goal. In our system, the security issue of dynamic RFID key management is also provided. The manager can control the RFID key more easy and smartly. We also create a easy used energy control interface, which can reduce the redundant standby computer by user's setting. A simulator of the rack server enclosure and it's peripheral devices is implement to verify the reliability of system. In our system simulation, there is about 39% electric power saving under the 5%~10% standby ratio setting and dynamic fans speed control in semantic automatic controlling function.

Section 7-2 Future Work

In this thesis, there is some important issue need to be concerned. For the further extension, we shall settle the following problem in the nearest future:

1. There is some studies of the heat flow analysis. We could combine with the predict module and real temperature value as the criteria of the server schedule.

2. Actually, the definition of semantic automatic controlling command may be conflicted. In the future we can adopt several models, such as Petri Nets and Colored Petri Nets (CPN), as verification tools to solve this problem.
3. In our system, the environment real information is available. We could provide the suggestion server list by APIs to the server consolidation software, such as OpenStack, VMware, Virtual Box. That will enable these server consolidation software to achieve more wise server schedule by real server situation.
4. This system will control the whole data center device. The security is absolutely essential in the data center environment. The safer communication between devices and web server should be considered. The authentication and encryption, like DES-3 AES ... etc, mechanism need to be designed for real environment.
5. Sometime the device will be displaced for regular maintaining. The more easy configuration setting for the device replacement may be considered. The more friendly user interface should be provided for user easy set and used.

Reference

[1]. Burkey, Roxanne E.; Breakfield, Charles V. (2000). Designing a total data solution: technology, implementation and deployment. Auerbach Best Practices. CRC Press. p. 24. ISBN 0-8493-0893-3

[2]. R. Nathuji, et al., "Exploiting Platform Heterogeneity for Power Efficient Data Centers," in Proc. of the IEEE International Conference on Autonomic Computing Washington, pp. 5-15, DC, USA 2007.

[3]. Gueyoung Jung, et al., "Mistral: Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures," in Proc. of the IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 62-73, 2010.

[4]. Taliyan, S.S., et al., "Finite element based thermal analysis of sealed electronic rack and validation," in Proc. of the IEEE International Conference on Reliability, pp 443-447, Safety and Hazard, 2010.

[5]. M. J. Marongiu, “Some issues in experimental testing and methodologies in the thermal management of telecommunication components, systems and enclosures,” in Proc. 17th Int. Telecommun. Energy Conf., INTELEC ’95, The Hague, pp. 834–838, The Netherlands, Oct. 29–Nov. 1, 1995.

[6]. Chen Yu, et al., "Design and Simulation of the Evaporator for Cooling System of the Blade Server," in Proc. of the IEEE International Conference on Measuring Technology and Mechatronics Automation, pp 992-925, 2011.

[7]. Calwell, C., Mansoor, A., "AC-DC server power supplies: making the leap to higher efficiency" in Proc. of the IEEE Conference and Exposition on Applied Power Electronics, pp 155-158, Vol. 1, 2005.

[8]. Douglas R. Mauro & Kevin J. Schmidt, "Essential SNMP", (1st ed.),

Sebastopol, CA: O’Reilly & Associates, 2001.

[9]. D. Harrington, et al., "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks" RFC 3411, December 2002

[10]. Miller, Rich. "Complexity: Growing Data Center Challenge," Data Center Knowledge, May 16, 2007

[11]. HUANG S J, TSENG P S. "A www-based virtual instrument software for electric power measurement with java enhancement", in Proc. IEEE Power Eng. Soc. Summer Meeting, 3: 1160–1164,2002.

[12]. R. Bacher and T. Orfanogianni, “WWW Based Computation Services: Transfer of Power System Applications to the WWW”, IEEE Power Engineering Society Winter Meeting, Vol. 1, pp. 496-501, New York, USA, February 1999.

[13]. S. Ritchie, “System Programming in Java”, IEEE Micro, Vol. 17, No. 3, pp. 30-35, May/June 1997.

[14]. Pollo, L.F., et al., "A network-oriented power management architecture," in IFIP/IEEE Eighth International Symposium on Integrated Network Management, pp 693-706, 2003.

[15]. Landsman, E.M., "Scalable data center architecture for on-demand power infrastructure," in Proc. of the IEEE Conference and Exposition on Applied Power Electronics, vol.1 , pp 10-13, 2003.

[16]. Xiaorui Wang, et al., "SHIP: A Scalable Hierarchical Power Control Architecture for Large-Scale Data Centers," in IEEE Transactions on Parallel and Distributed Systems, pp. 168-176, 2012.

[17]. Ying Song, et al., "Utility analysis for Internet-oriented server consolidation in VM-based data centers" in IEEE International Conference on Cluster

Computing and Workshops, pp. 1-10, 2009.

[18]. Hendershot, L., "Automating the data center operation: `Do we have a choice?' `Can we manage it effectively?'," In IEEE Symposium on Mass Storage Systems, pp. 85-87, 1990.

[19]. Kaminsky, D., et al., "Policy-Based Automation in the Autonomic Data Center," in International Conference on Autonomic Computing, pp. 209-210, 2008.

[20]. Yung-Wei Kao, Shyan-Ming Yuan, "User-configurable semantic home automation," Computer Standards & Interfaces, 34(1): pp. 171-188, 2012.