Jurnal Teknologi

Full Paper

DS-DASHBOARD: DISTRIBUTED SERVER'S RESOURCE MANAGEMENT, CASE STUDY INSTITUT TEKNOLOGI DEL

Marojahan Mula Timbul Sigiro*

Institut Teknologi Del, Sitoluama, Laguboti, Sumatera Utara Indonesia

Article history
Received
29 June 2015
Received in revised form
28 September 2015
Accepted
27 December 2015

*Corresponding author
marojahan@del.ac.id

Abstract

Most of organizations or companies nowadays heavily rely on the Information Technology to manage their activity. They deploy many servers and application which each of them handle or keep specific tasks. File server, email server, user account, application server and many others. These resources are stored and manage by servers that may have different configuration and different platform. To manage these resources is a complicated task because of those different attribute and platform. DS-Dashboard is a system that provides transparency to manage user's resources that distributed on different servers. A different approach is taken where the resources seen as attribute or part of main resources which is user.

Keywords: distributed resource, dashboard, resources connectivity

1.0 INTRODUCTION

Del Institute of Technology (IT Del) is an institution that has many digital resources which deployed on may servers. Resources in this case are digital resources managed by different serves such as user account, emails, database records and file. Each servers responsible for specific tasks and may have different platforms, Linux or Windows.

Managing resources that distributed on different servers and different platform is a complicated task because each servers and platform has different attributes and characteristics. Managing file on Linux server is different from managing it on windows file server.

If we see from different perspective, those resources are tied to one entity, user. Based on that pattern, a system is developed with a different approach in managing server resources. This system provides transparency in managing servers. In this system, all the resources are belong to user and we see those resources as users attribute. System admin do not need to manage the resources directly but through the application, called ds-dashboard.

2.0 BASIC CONCEPTS

2.1 Basic Concept of Dashboard

Dashboard is a visual appearance of important information that is needed to achieve one or some purposes with consolidating and organizing information in a single screen, in order to make the organization's works are able to monitor at a glance [1].

Dashboard that was known before, as Executive Information System (EIS) is a computer based system that is able to serve the needs of information for executives, access the rapid and most up-to-date information, and access management report directly.

2.2 Server

Server is a computer system that provides services on a computer network. Server runs administrative software that control access for network, resources inside such as file or printer, and clients connected to the network [2].

Based on the function, servers are categorized as Account server, File server, Database server and Application Server.
Based on the platform (operating system), servers can be installed on some kinds of platform, such as Windows Server and Linux.

2.3 Resources

Servers are also capable of managing other kinds of resource, such as user account, file, memory, bandwidth, storage, and device.

2.4 Programming Language

Programming language is one of the keys in this system. In order to provide transparency, the system must be able to run specific function on different servers. For example, the system admin must be able to grant write permission of a file to specific user without knowing that the file stored in windows file server or Linux file server. To handle this case, we have to provide an abstraction to platform specific task and command by writing a native application on each platform.

The programming languages used in the building of the application are Batch Script that will be ran on Windows Shell, Java to build the application server and the agent, and Shell Script to provide abstraction to Linux system. The main consideration to choose java is the portability, which can be executed in different platform and it also widely adopted that will make it easier to extend the application.

2.5 Client-Server Communication

DS-Dasboard is a client server application with agent-based model. The application has coordinator (server) that manages all the resources on different servers and platforms through agents (client) installed on each server.

3.0 DESIGN

In detail, architecture design of the system is visible on Figure 3.

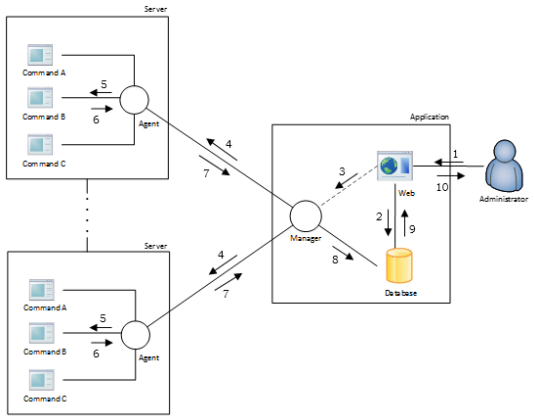

Figure 1 Architecture design of the system

The figure's explanation based on the flow of the application is in the following way:

1. The DS-Dashboard has a web application as its frontend. Therefore, the administrator organizes the resources by entering commands that is sent to the server through the web.
2. The application interacts with the database to retrieve data. The data is command which was sent to the server depends on the server's type.
3. The application has an Application Coordinator to retrieve command that was sent to the Server Agent on each destination server.
4. The Application Coordinator sends the message to the Server Agent on each server that handle user's resource.
5. The Server Agent runs the command based on the message it retrieved from the Application Coordinator. The command can be selected. If the server runs on Linux, the Server Agent runs the Linux command (shell) and if the server runs on Windows, the Server Agent runs the Windows command (batch).
6. After the commands in each server had run, the Server Agent receives message respond that will show whether the command worked, failed, or replied other appropriate information.
7. The respond from the command is sent back to the Application Coordinator.
8. The Application Coordinator saves the respond of the command to the database.
9. The Application Coordinator interacts with the database to retrieve the remote server responds.
10. The application shows the information that contains server's respond for the command so the administrator is able to ensure whether the command worked, failed or replied other important information.

4.0 TESTING

4.1 First Case Testing

In this first case, we assume that we want to add employ data to the servers. The data will be added to the file-server and mail-server. The employee also can be given specific roles in the system, as shown in figure 2.

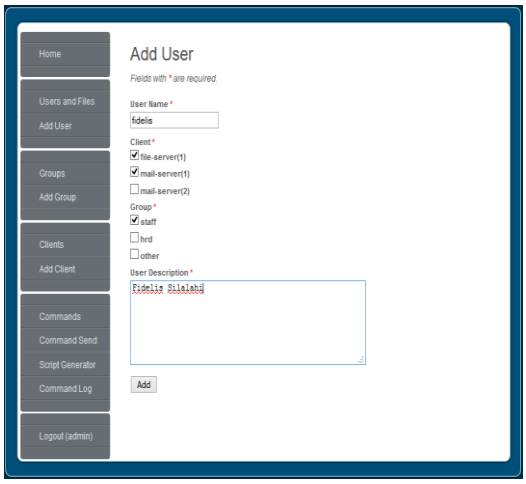

Figure 2 First case, stage 1

As shown in figure 3, on the mail-server, the employee is also given a database account for mysql server.

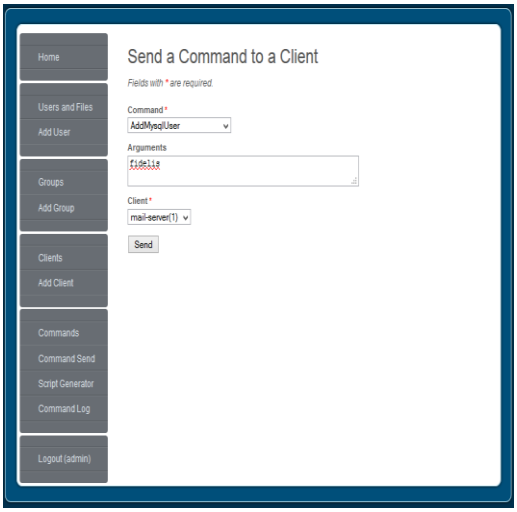

Figure 3 First case, stage 2

The employee's status is active at IT Del and the employee can use the email from mail-server with the account.

4.2 Second Case Testing

The second case is how the employee who has a project and has too shares the project's folder to other employees. The folder is located on file-server in the employee's home directory.

In this case, the employee is not on site, so the sharing thing has to be done by the administrator. The project owned by the staff group. Therefore, the administrator changes the project's folder's group to staff.

Figure 4 Change group

The system provides interface as seen in Figure 4 to change the remote folder permissions. The permission of the folder itself is changed to bel full-access for the user on the staff group as seen in figure 5.

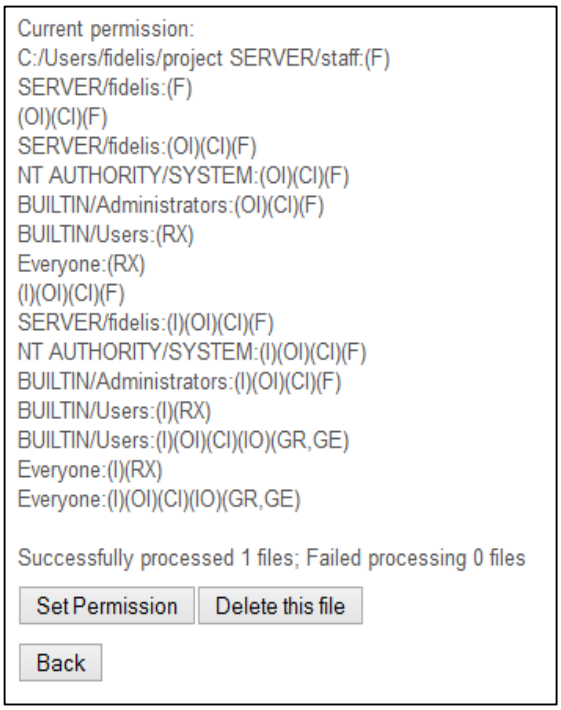

Figure 5 Command result

4.3 Third Case Testing

The third case is for the old employee who is no more works for IT Del. The account of the old employee is on file-server, mail-server. As shown in figure 6, we can list all the resources that belong to the employee. The resources are shown along with its location and configuration. The resources now can be deleted or archived through the application

without knowing where or how the resources are stored.

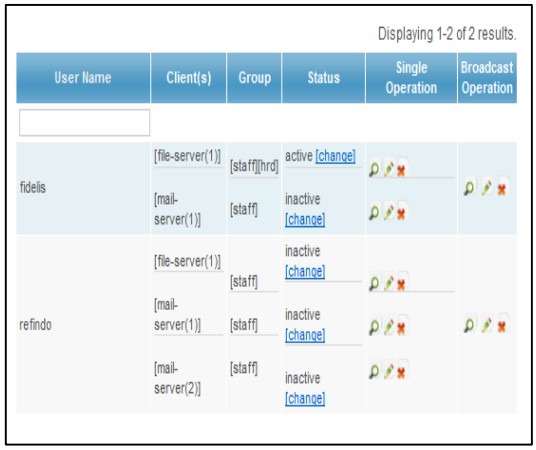

Figure 6 Resources List

4.4 Testing Constraints

Based on the implementation, the system has some constraints:

- The management is done for user and file. User and file was managed to see the result from adding user, adding group for user, removing group from user, access on a file, and file property.
- Security matter is not covered yet in the system.
- The project is about making dashboard and building client-server connection.
- The management is done on two operating systems. The system runs on the system are Windows Server 2008 and CentOS 6.x.

+

5.0 CONCLUSION

Based on the implementation and the testing that have been done, the server's resource management will be easier by using the DS-Dashboard. Lots of other functionality on the agent has to be added in order to provide full transparency in managing server resources. The system can also be extended to support other kind of platform, such as embedded system. A specific agent and platform specific command to provide abstraction need to be implemented to support different kind of servers or resources.

References

[1] Takuya Araki, 2015. Autonomic WWW Server Management with Distributed Resources, *NEC Internet System Laboratories*.

[2] Katia Sycara, Keith Decker, Anandeep Pannu, Mike Williamson, Dajun Zeng, 2014. Distributed Intelligent Agents, *Carnegie Mellon University*.

[3] Stefen Few, 2013. Dashboard Design for Real –Time Situation Awarnes,https://www.perceptualedge.com/articles/Whitepapers/Dashboard_Design.pdf

[4] Kusuma, Yogie. 2013. Aplikasi EIS Untuk Monitoring Transaksi Penjualan Dan Pembelian. Tasikmalaya: *TI Universitas Siliwangi.*

[5] Sinambela, Eka. 2013. Dashboard untuk Pemantauan Operasional Jaringan. Bandung: *Institut Teknologi Bandung.*

[6] Abhijit A. Sawant, Dr. B. B. Meshram, 2013. Network Programming in Java Using Socket, *IJERA*.