Article

Research and Application of a SCADA System for a Microgrid

Shuangshuang Li, Baochen Jiang *, Xiaoli Wang and Lubei Dong

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, China; lishuangshuang_sdu@163.com (S.L.); wxl@sdu.edu.cn (X.W.); donglubei@mail.sdu.edu.cn (L.D.)

* Correspondence: jbc@sdu.edu.cn; Tel.: +86-186-6935-5366

Abstract: An effective Supervisory Control and Data Acquisition (SCADA) system can improve the reliability, safety and economic benefits of a microgrid operation. In this research, the lower central controller and upper WEB (World Wide Web) monitoring system are connected by the SCADA system, which is the hub of a microgrid intelligent monitoring platform. This system contains a set of specific functions programmed by Java as a middleware and can provide communication and control functions between the central controller and the upper monitoring system. For the sake of security and stability of the microgrid, the SCADA system realizes business processing on real-time data acquisition and storage, load balancing and resource recovery, concurrent security processing, and control instruction parsing and transmission. All those functions were tested and verified in actual operation.

Keywords: microgrid; SCADA; Java; middleware

1. Introduction

With energy and environment issues becoming increasingly prominent, the exploitation of clean renewable energy resources has become an economic and socially important strategy for sustainable development worldwide. In order to discover the advantages of distributed generation in the areas of economy, energy and environment, microgrid technology, which uses a large amount of renewable energy has gained in popularity and is rapidly developing. A microgrid can be defined as a power supply system which contains distributed generation (DG), energy storage and control systems that work together to supply the local loads [1,2].

In a microgrid, a Supervisory Control and Data Acquisition (SCADA) [3] system is used for data acquisition, monitoring and procedure control for spot devices, and is a computer based production procedure control and dispatching automation system [4,5]. It can fulfill data acquisition, equipment control, measurement, parameters adjustment and all kinds of signal alarms, which play an important role in improving the reliability, safety and economic benefits of power grid operation. Furthermore, it can reduce the burden of the dispatchers, comprehend the electric power dispatching automation and modernization, and improve the efficiency and the level of dispatch [6].

The SCADA system is usually divided into hardware and software parts, and it has been designed and applied in many fields. The study in Reference [7] introduced the design and implementation of hardware components in the microgrid SCADA system. Reference [8] presented the communication and PC interface of the microgrid SCADA, and Reference [9] presented a SCADA system using the IEEE 802.22 standard to overcome all existing limitations. Additionally, Reference [10] used a SCADA system to provide real-time control of power switching relays, and obtain information about their status and perform three-phase measurement, whereas the primary aim of Reference [11] was to propose a suitable private hybrid cloud based SCADA architecture that satisfied various necessities in the framework of interoperability of microgrid platforms while maintaining security restriction conditions. In contrast, our research designed a SCADA system programmed by Java to be the middleware in an innovative microgrid intelligent monitoring

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2017 doi:10.20944/preprints201703.0068.v2

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012

platform. The lower central controller and upper WEB monitoring system were connected by the SCADA system, which was the hub of the microgrid intelligent monitoring platform.

The remainder of the paper is organized in four sections. In Section 2, the architecture and main processing tasks of the microgrid SCADA system are presented. Section 3 describes the design of SCADA system by looking at three of the most important aspects in detail and Section 4 presents the implementation and operation test. Our conclusions are drawn in Section 5.

2. The Architecture of a SCADA System in the Microgrid

In this actual system, the microgrid is mainly composed of a microgrid hardware system and a microgrid intelligent monitoring platform. This paper introduces the SCADA system, which is one type of middleware used in intelligent monitoring platforms. The architecture of the SCADA system in the microgrid is shown in Figure 1. The module with the UART (Universal Asynchronous Receiver/Transmitter) Ethernet receives the lower central controller's real-time data from the serial port before the data is encapsulated and finally sent to the SCADA system through the Ethernet. The SCADA system is mainly utilized for parsing the microgrid packaged data, before the parsed data is stored in the MySQL database. Furthermore, the SCADA system is used for communicating with the upper WEB monitoring. After receiving the upper client control instructions, the system will parse the instructions and send them to the lower central controllers by the UART Ethernet. The main processing tasks of the SCADA are: system startup processing, client connection processing, business processing, load balancing processing, resource recovery processing and concurrent security processing. Section 3 highlights three of the most important parts to describe the design of the SCADA system in detail.

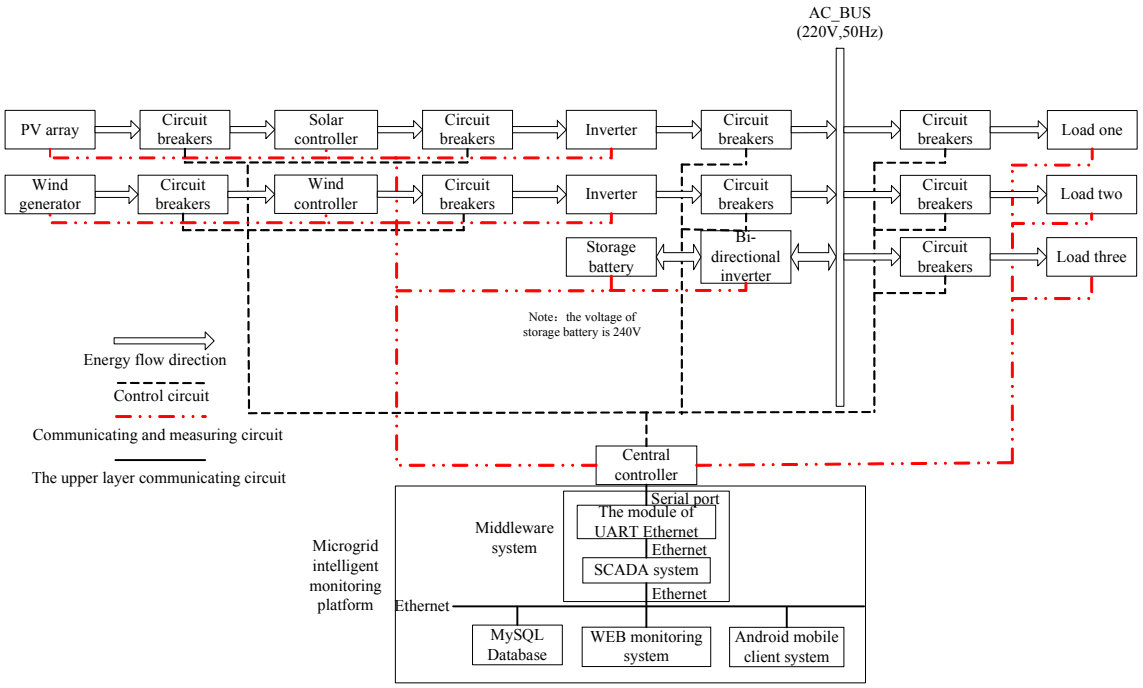

Figure 1. The architecture of the SCADA (Supervisory Control and Data Acquisition) system in the microgrid.

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012

3. Design of the Microgrid SCADA System

3.1. Business Processing in the SCADA System

The SCADA system invokes the "start" method of Thread class to start the thread. The "start" method invocation will invoke asynchronously the "run" method of this Thread class, and the business processing is in the "run" method. The system receives the Modbus/TCP protocol data frames by the IO stream in the "run" method. After the data frames have been parsed and verified, they are classified according to the control code field in the protocol data frames. The data are processed in five different ways according to the results of the classification: Electrical message business processing, alarm message business processing, message business processing of clients requesting the system to check the time, message business processing of the upper monitoring system sending control instructions, and message business processing of upper monitoring system giving the power generation plan instructions. Finally the processed data are inserted into the corresponding MySQL database tables, which are displayed in the web pages in the WEB monitoring system, so that the administrator can handle the event. The business processing flow of the SCADA system is shown in Figure 2.

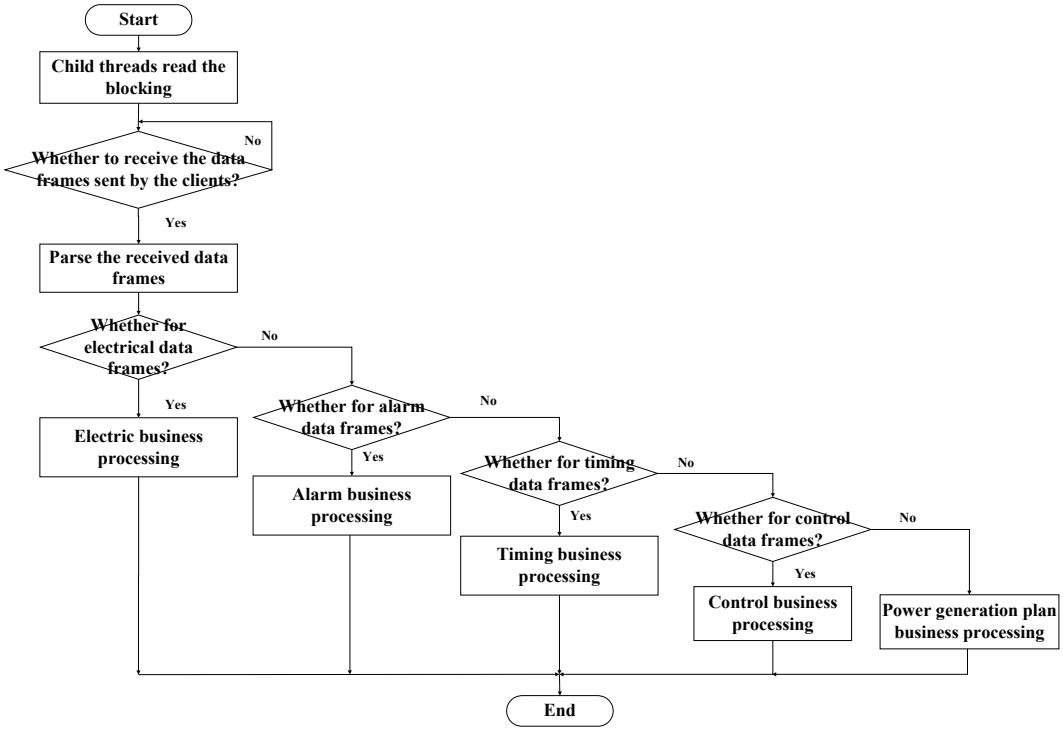

Figure 2. The Business Processing Flow in the microgrid SCADA system.

For information required by the microgrid regarding temperature, wind speed, and light intensity, the system starts the web crawler service program to crawl for weather information from the observatory close to the microgrid system. Next, the system parses the weather information to generate the required weather information field and stores it in the database.

The system inserts the aggregated electrical data into the microgrid electrical table. As the SCADA system gathers microgrid electrical data every five seconds, there will be 17,280 items a day, which means that over time, the data volume will become huge. The query access to the microgrid electrical data is still very slow even with the creation of an index used for the microgrid electrical data table. To solve the problem mentioned above, this study used a sub-table mechanism to dynamically generate a new table daily. When data was inserted into the microgrid electrical database table, the system judges whether there is a current date microgrid electrical table using the "Create Table If Not Exists" statement first. If the table exists, the system inserts the data into it; however, if the table does not exist, the system will create the table and insert the data into it. Every

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012

time these tables are accessed, even simple database connections may affect efficiency. This paper used cache mechanisms to solve that issue. Cache mechanism identifies the current date microgrid electrical table that exists or is nonexistent through the map cache of the current microgrid electrical table and combines it with the "Create Table If Not Exists" statement to solve the cache empty problem when the system restarts, which ensures that the microgrid electrical data acquisition is stable in the SCADA system.

3.2. Load Balancing Processing in the SCADA System

Load balancing processing is mainly aimed at solving the problem of unbalanced loads in the SCADA system. The SCADA system needs to use multithreading technology to deal with business requests that are sent from multiple central controllers. As reported in Section 3.1, after business processing, the processed data is inserted into the MySQL database. If the SCADA system supports more than one hundred central controllers sending data to it, issues of "hot spots" occur. This means that the MySQL database connections are exhausted when all central controllers request MySQL connections to the SCADA system. As a result, unsuccessful abnormal links appear and with the increase of the number of central controllers, the probability of this occurrence becomes even higher. Although the value of the "max_connections" in the mysql.ini file can be modified to increase the maximum number of MySQL connections to mitigate this problem, the value of the maximum number is different under the different operating systems. Furthermore, this number has certain limitations, which cannot fundamentally solve this problem.

To solve this issue, a producer-consumer model was used. When SCADA system starts, it starts a custom "handler" thread to handle the business alone, and the "handler" thread inherits the Thread class and implements the "run" method. The "handler" thread can be started by invoking its "start" method. There is a double linked list in the handler thread, and elements in the double linked list are custom "DisposeBean" class objects, which have two parameters: the "sql" statement and the "paras" arrays. The "sql" and "paras" arrays are on behalf of the "sql" statement and parameters when we create the tables or insert data into tables, respectively. The double linked list is locked to guarantee the safety of the thread. Additionally, in order to prevent repeated switching of the system context, an optimistic locking mechanism is used to achieve the locking operation of the double linked list. The optimistic locking mechanism completes the locking operation by the CAS mechanism of a CPU that can lock the instructions stream. As the double linked list is used to realize a blocking queue, the "await" and "signal" operation can be used to set up an observer model on the queue where, if the producer finds that the queue capacity reaches the maximum value at one end of the queue, it will call the "await" method to block until the consumer calls "signal" method to tell the producer to stop blocking at the other end of the queue.

By using the above operation, the SCADA system can task the operation of obtaining a connection from the MySQL database to the "handler" thread to manage. Only generating a custom "DisposeBean" class object will insert into the blocking queue without the requirement of obtaining a connection from the MySQL database. Thus, this handling can resolve the issue of "hot spots," and make balance the system loads. Furthermore, it may support business requests from thousands of central controllers.

3.3. Concurrent Security Processing in the SCADA System

Concurrent technology is primarily used for solving problems of concurrent communication, on what kind of communication technology communication is based, and is essentially point-to-point communication technologies when the client communicates with the server. To support the communication between the server and the client by the way of many-to-many, concurrent technology has to be used.

Concurrent security processing in the SCADA system is mainly embodied in the thread resources competition. Threads are stored in the threadPools, and are marked in an unused state when they are initialized. As the SCADA system uses the BIO model—that is "one per one thread"—the SCADA system is used as a server. There are multiple clients to apply simultaneously

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2017 doi:10.20944/preprints201703.0068.v2

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012

available threads resources from threadPools, as well as multiple clients to select simultaneously from the same thread and to bind its own connection "socket" separately when the concurrent occurs, which may result in the loss of the connection "socket" to update. To solve these problems, this study used a thread synchronization mechanism, which creates a lock processing operation when clients apply simultaneously available thread resources from threadPools to ensure that only one client can hold the lock. The system releases the lock after the application process is completed and then allows other clients to continue to access the lock using a synchronized synchronous block.

4. System Test

This SCADA system has been tested and implemented at a real microgrid system. Tests mainly included a system startup test, a client connection test, a business processing and load balancing test, a resource recovery test, a concurrent security module test, and a control instruction parsing and transmission test.

4.1. System Startup Test

First, "net start mysql" was entered to start the MySQL database at the command line console before "java -jar server.jar" was entered to start the SCADA system. When the SCADA system starts, it will create a "ServerSocket" connection socket, which provides two parameters to bind the IP address and port number of the computer. Next, the system starts to read the configuration file, and will complete reading the configuration information and load the startup module in the configuration file by the static block. As known, the static block is executed during the initialization stage of the JVM loading and is executed only once. The system creates a threadPools when it first loads the startup module. The maximum capacity of the threadPools is obtained from the configuration file and creates a certain number of threads into the threadPools in advance according to 10% of the maximum capacity in the threadPools. Following this, the system creates two link tables: the central controller link table (mcu link table) that records the lower central controllers connection information, and the upper WEB monitoring system link table (web link table) that records the upper WEB monitoring system connection information. Finally, the system invokes the "accept" method of the "ServerSocket" connection socket to block and wait for the connection requests from clients. The results show these threads' states, including their serial number and a flag that marks if was used or not. Test results are shown in Figures 3 and 4.

Figure 3. The server of MySQL starting.

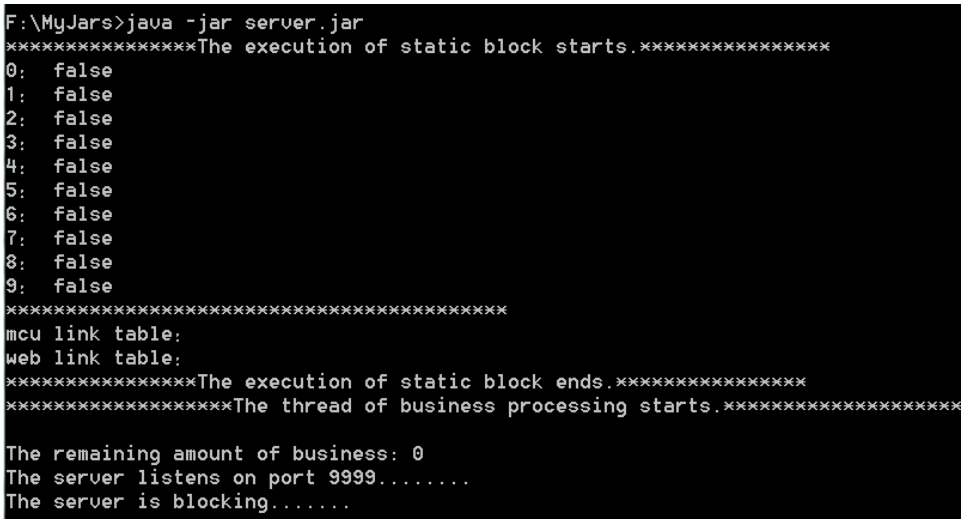

Figure 4. The SCADA system starting.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2017 doi:10.20944/preprints201703.0068.v2

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012

4.2. Client Connection Test

After the SCADA system startup is completed, the client can send a connection request to the SCADA server. If the client's IP address is in the legal permission scope, the system will receive the protocol data frame by the IO stream and parse it. If the parsed protocol data frame is legal according to the Modbus protocol specification and there is an available thread in threadPools, a connection will be established, if not, the connection is disconnected. Test results are shown in Figures 5 and 6.

```
××××××××××××××××The execution of static block ends.××××××××××××××××
××××××××××××××××××××The thread of business processing starts.××××××××××××××××××××

The remaining amount of business: 0
The server listens on port 9999........
The server is blocking.......
New Link is coming: /127.0.0.1:36463
Registration failed, there is a illegal link!
The server is blocking.......
```

Figure 5. The SCADA system receiving an illegal connection request.

```
New Link is coming: /127.0.0.1:19705
mcu link table:
1: Socket[addr=/127.0.0.1,port=19705,localport=9999]
web link table:
Send a frame.
0:  true
1:  false
2:  false
3:  false
4:  false
run method is executing: /127.0.0.1:19705
5:  false
Read the blocking......
6:  false
7:  false
8:  false
9:  false
The server is blocking.......
```

Figure 6. The SCADA system receiving a legal connection request.

4.3. Business Processing and Load Balancing Test

The lower central controller sends electrical data frames to the SCADA system by means of a UART Ethernet module. The SCADA system parses and processes the received protocol data frames, before finally inserting them into the MySQL microgrid electrical table. The test results are shown in Figures 7 and 8.

```
××××××××××××××××××××The thread of business processing starts.××××××××××××××××××××

The remaining amount of business: 0
The server listens on port 9999........
The server is blocking.......
New Link is coming: /127.0.0.1:47167
mcu link table:
1: Socket[addr=/127.0.0.1,port=47167,localport=9999]
web link table:
Send a frame.
0:  true
1:  false
2:  false
3:  false
4:  false
5:  false
6:  false
7:  false
run method is executing: /127.0.0.1:47167
8:  false
9:  false
The server is blocking.......
Read the blocking......
Read the blocking......
【 2016-9-24 17:22:04 Creating tables is completed this time.】
The remaining amount of business: 1
【 2016-9-24 17:22:04 Inserting data is completed this time.】
The remaining amount of business: 0
```

Figure 7. The business processing test and load balancing test.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2017 doi:10.20944/preprints201703.0068.v2

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012

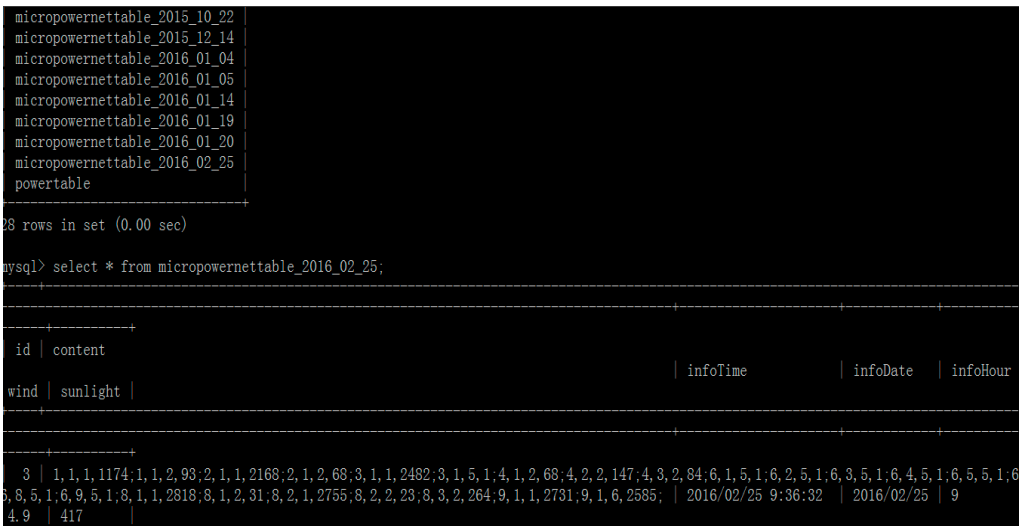

Figure 8. The processed data being inserted into the microgrid electrical table.

4.4. Resource Recycling Test

The resource recovery of the SCADA system is mainly to resolve the robustness problems of the system. The SCADA system is a hub, so other communications need to be completed by the SCADA system. The SCADA system can detect abnormal links and close them, after which the SCADA system must recycle resources to prevent the OOM (Out Of Memory) memory leak phenomenon, which may result in the downtime of the SCADA system. The test result is shown in Figure 9.

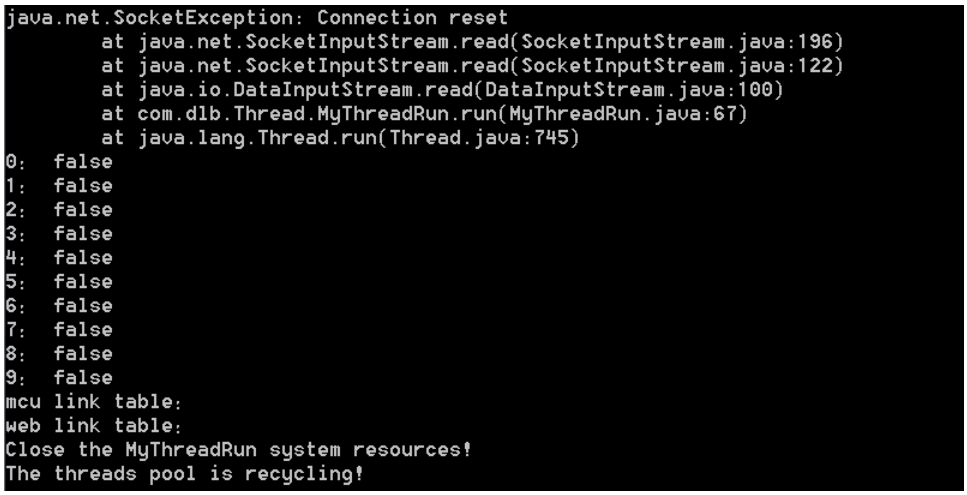

Figure 9. The SCADA system closing the abnormal link and recycling system resources.

4.5. Concurrent Security Module Test

When multiple clients connect to the SCADA system, the system can operate normally, so this system has good scalability. Furthermore, any abnormal client does not affect the normal communication between the client and the SCADA server. These test results are shown in Figures 10 and 11.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2017 doi:10.20944/preprints201703.0068.v2

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012


```
The server is blocking.......
run method is executing: /127.0.0.1:4539
Read the blocking......
New Link is coming: /127.0.0.1:4574
mcu link table:
1: Socket[addr=/127.0.0.1,port=4574,localport=9999]
web link table:
Send a frame.
0:  true
run method is executing: /127.0.0.1:4574
1:  true
Read the blocking......
2:  false
3:  false
4:  false
5:  false
6:  false
7:  false
8:  false
9:  false
The server is blocking.......
```

Figure 10. Multiple clients connecting to the SCADA system.

```
The server is blocking.......
java.net.SocketException: Connection reset
        at java.net.SocketInputStream.read(SocketInputStream.java:196)
        at java.net.SocketInputStream.read(SocketInputStream.java:122)
        at java.io.DataInputStream.read(DataInputStream.java:100)
        at com.dlb.Thread.MyThreadRun.run(MyThreadRun.java:67)
        at java.lang.Thread.run(Thread.java:745)
0:  true
1:  false
2:  false
3:  false
4:  false
5:  false
6:  false
7:  false
8:  false
9:  false
mcu link table:
web link table:
Close the MyThreadRun system resources!
The threads pool is recycling!
```

Figure 11. One abnormal client that is not affecting others.

4.6. The Control Instruction Parsing and Transmission Test

In this upper WEB monitoring system, the administrator can control six parts of the microgrid hardware platform: Load 1, Load 2, Load 3, PV array, Wind generator and Storage battery. When the power generation is insufficient to supply all loads, the administrator can instruct the system to disconnect one load to the microgrid system. As seen in Figure 12, the administrator selects Load 1 to be disconnected from the microgrid system and clicks the "Confirm" button in the WEB monitoring system, and the background program will form a frame of control according to the instructions of the foreground interface. Then, the control frame will be sent to the SCADA system through the IO stream of socket. Next, the SCADA system sends the instruction to the central controller through the UART Ethernet module after parsing this instruction and inserts this operation into the "microEventTable" of the MySQL database as shown in Figures 13 and 14. Finally, the microgrid hardware platform carries out the instruction to cut off Load 1 (four light bulbs) to make the microgrid safe and stable (Figure 15).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 March 2017 doi:10.20944/preprints201703.0068.v2

Peer-reviewed version available at Technologies 2017, 5, 12; doi:10.3390/technologies5020012

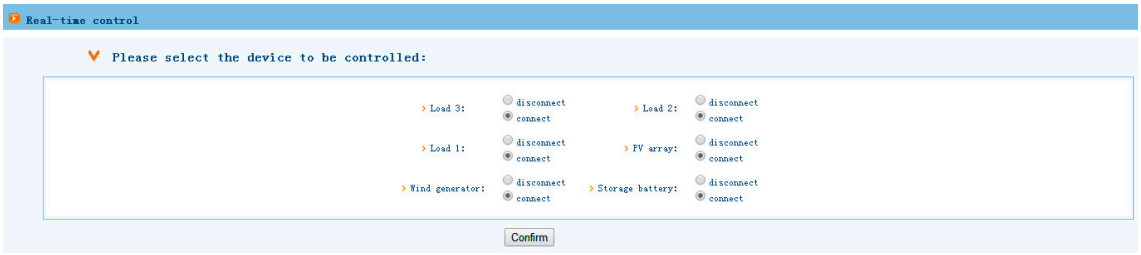

Figure 12. Real-time control user interface in the WEB monitoring system.

```
The server is blocking.......
New Link is coming: /127.0.0.1:43710
mcu link table:
1: Socket[addr=/127.0.0.1,port=43679,localport=9999]
web link table:
51: Socket[addr=/127.0.0.1,port=43710,localport=9999]
Send a frame.
Print threads' s states including their serial number and flag:
0:  true
1:  true
2:  false
3:  false
4:  false
5:  false
6:  false
7:  false
8:  false
run method is executing: /127.0.0.1:43710
9:  false
The server is blocking.......
Read the blocking......
Send a frame.
Scada sends control instruction to mcu!
Read the blocking......
【 2017-3-7 19:00:25 Inserting data is completed this time.】
The remaining amount of business: 0
```

Figure 13. The control instruction being transmitted to the UART (Universal Asynchronous Receiver/Transmitter) Ethernet module by the SCADA system.

```
Stable Library
=========================================
Native lib Version = RXTX-2.1-7
Java lib Version   = RXTX-2.1-7
Connect to the COM3 successfully!
Search port COM3 completely!
The mcu is receiving blocking!
The control frame is:
0 1 6 0 0 1 0 0 0
The module of UART Ethernet sends the control instruction to mcu!
The mcu is receiving blocking!
```

Figure 14. The control instruction being sent to the mcu (Microcontroller Unit) by the UART Ethernet module.

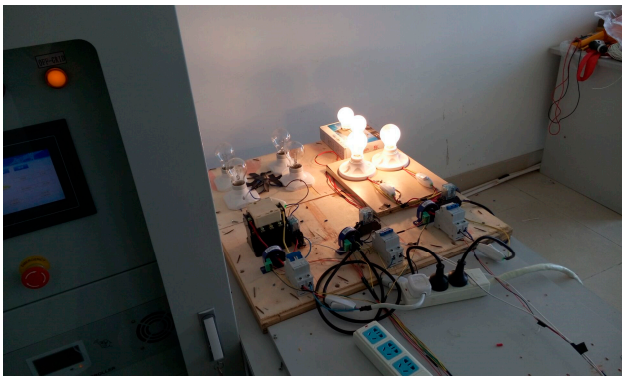

Figure 15. The control instruction to cut off Load 1 being carried out in microgrid hardware platform.

5. Conclusions

This paper first introduced the importance and the functions of a SCADA system in the microgrid. Second, it presented the architecture of the SCADA system as middleware in the microgrid and described in detail the design of the SCADA system in Java. The system can realize real-time data acquisition and storage, control command parsing and transmission, system security and stability, load balancing and resource recovery of the microgrid. Finally, the implementation and operation test of the SCADA system proved to be both practicable and feasible. Its implementation provided a common interface for other business systems to access the SCADA system data, and could be easily integrated into the microgrid monitoring systems. Thus, it has shown capacity for good reusability, stability and easy expansibility. However, to improve the storage performance and access speed of the system, the distributed file storage system HDFS and the parallel computing framework MapReduce can be used as a storage and computing platform for microgrid data in the future.

Author Contributions: Baochen Jiang conceived and designed the experiments; Shuangshuang Li and Lubei Dong performed the experiments; Shuangshuang Li, Lubei Dong, and Xiaoli Wang analyzed the data; Shuangshuang Li and Lubei Dong wrote the paper. Baochen Jiang and Xiaoli Wang supervised the research project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shen, Z.Q.; Deng, W.; Pei, W.; Mu, L.S.; Ouyang, H. Design and implementation of microgrid SCADA platform. *Adv. Mater. Res.* **2013**, *732–733*, 1358–1364.
2. Palma-Behnke, R.; Ortiz, D.; Reyes, L.; Jiménez-Estévez, G.; Garrido, N. A social SCADA approach for a renewable based microgrid—The Huatacondo project. In Proceedings of the IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24–28 July 2011; p. 7.
3. Mehta, B.R.; Reddy, Y.J. *Industrial Process Automation Systems: Design and Implementation*. Elsevier Publishers: Amsterdam, The Netherlands, 2015; Volume 7, pp. 237–300.
4. Pampashree; Ansari, M.F. Design and implementation of SCADA based induction motor control. *Int. J. Eng. Res. Appl.* **2014**, *4*, 5–18.
5. Cao, K. SCADA system research of the grid. *China New Technol. Prod.* **2013**, *5*, 23. (In Chinese)
6. Win, K.T.Z.; Tun, H.M. Design and implementation of SCADA system based power distribution for primary substation (control system). *Int. J. Electron. Comput. Sci. Eng.* **2014**, *3*, 254–261.
7. Chen, Y.N.; Pei, W. Design and implementation of SCADA system for Micro-grid. *Inf. Technol. J.* **2013**, *12*, 8049–8057.
8. Lázár, E.; Etz, R.; Petreuş, D.; Pătărău, T.; Ciocan, I. SCADA development for an islanded microgrid. In Proceedings of the 21st IEEE International Symposium for Design and Technology in Electronic Packaging, Brasov, Romania, 22–25 October 2015; pp. 147–150.
9. Mollah, M.B.; Islam, S.S. Towards IEEE 802.22 based SCADA system for future distributed system. In Proceedings of the 1st International Conference on Informatics, Electronics and Vision, Dhaka, Bangladesh, 18–19 May 2012; pp. 1075–1080.
10. Regula, M.; Otcenasova, A.; Roch, M.; Bodnar, R.; Repak, M. SCADA system with power quality monitoring in Smart Grid model. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016.
11. Nguyen, V.H.; Tran, Q.T.; Besanger, Y. Scada as a service approach for interoperability of micro-grid platforms. *Sustain. Energy Grids Netw.* **2016**, *8*, 26–36.