

Research On DB2 Performance Testing Automation

Zhuang Lei¹ Gao Zhen² Wu Hao³ Yang ChunXin⁴ Zheng Miao⁵

^{1,2}School of Software Engineering, Tongji University, Shanghai, China

^{3,4}IBM CDL, Beijing China

corresponding author: Gao Zhen

¹onlyzhuanglei@gmail.com, ²gaozhen@tongji.edu.cn, ³haowu@cn.ibm.com

Abstract—Software testing play a significant role in modern software development and maintenance process, which is also an important means to ensure software reliability and improve software quality. With the continuous improvement of quality requirements of the software products and software engineering technology become more sophisticated, software testing has been participating into every phase of software lift cycle, become more and more important in software development and maintenance. DB2 Performance testing consists of four parts, which are environment setup, workload run, data measurement and environment clean up. Before all the operations are done manually and need about two hours' continuous attention. What's worse, even three times a day. This mechanical and complicated procedure is clearly unacceptable. This paper put forward a reusable automated testing framework based on IBM automated testing tools RFT to achieve the whole testing procedure automation. It reduces the count of human-computer interaction and greatly improves the efficiency of DB2 performance testing.

Keywords-DB2 performance testing , RFT, Automated Testing, Testing Framework

I. INTRODUCTION

With the development of the software industry, software quality control and management has become the core of the survival and development of enterprises. Almost every IT companies' software products requires a lot of performance control, testing and documents before release. Software testing market has become a unique market for the software industry. At the same time, the automated testing has also become a well known topic. Software engineers are realized the only way to mature and standardized of software testing is testing automation. In the near future, automated testing will be more and more widely used.

Automated testing is to automated execute the testing script using existing or develop some automated testing tools through recording, enhancement and playback script. All the operations are done by machine rather than human. The tester will completed more in the shortest time.

DB2 is the most well-known and most successful product in IBM data management product line. Now, not only in the bank industry, DB2 has been widely used to manage product information in major areas such as finance, manufacturing, transportation and so on. DB2 help the information technology process of various industry leap into a new stage. However, the widely used of DB2 bring some challenges to the performance testing. Before the new function realease, the performance testing needs to cover all the customers' environments. And the configuration of DB2 is complicated and time consuming. If all

the operations are done manually, the testing procedure is inefficiency.

This paper focus on DB2 performance testing procedure and put forward a reusable automated testing framework based on RFT to achieve the whole testing procedure automation. The paper describe the design and implementation of DB2 performance automated testing framework.

II. OVERVIEW OF DB2 PERFORMANCE TESTING

A. Steps Of DB2 Performance Testing

In general, DB2 performance testing can be divided into four steps, they are environment setup, workload run, performance data measurement and environment clean up. Every step has many detail operations. During this procedure, three kinds of applications included, which are java web application, batch application and TSO operation application. IPL related operations are done on HMC(Hardware Management Console) and HMC is belong to java web application. CMD and PUTTY are belong to batch application. TSO operation application means the operations on mainframe using TSO command.

1) Testing Environment Setup

- a) Putty used to apply configuration changes On Z Box
- b) CMD used to verify IPL done by TCPIP Connection
- c) HMC session to adjust CPUs Of Z Box and IPL LPAR

2) Workload Run

- a) TSO operations to start and configure DB2, CICS and other subsystems
- b) TSO operations to start workload run
- c) TSO operations to monitor CPU usage

3) Data Measurement

- a) TSO operations to measure performance data

4) Environment Clean Up

- a) HMC session to deactivate LPAR
- b) TSO operations to shut down machine

B. Challenges Of DB2 Performance Testing Automation

- 1) Operations on HMC are belong to java web application. When doing this application automation, GUI object recognition is complicated.
- 2) In terms of operations on TSO, how to connect to host and how to get terminal object are key technical
- 3) Getting window object of PUTTY and CMD is the key to batch application automation.

- 4) During the procedure, how to series the three kinds of applications
- 5) The environment setup of DB2 for z is really complicated. Accident handling is very important.
- 6) After IPL LPAR done, how to verify TCPIP ready

III. RFT INTRODUCTION

A. Features Of RFT

Rational Functional Tester software is an automated tool which provides testers with automated testing capabilities for functional testing, regression testing, GUI testing and data-driven testing.

As an automated testing tool, RFT has several features below:

- 1) Provide robust testing support for Java, Web 2.0, SAP, Siebel, terminal-based and Microsoft Visual Studio .NET Windows Forms applications
- 2) Perform story board testing to combine natural language test narrative with visual editing through application screenshots
- 3) Use keywords to bridge the gap between manual and automated testing
- 4) Manage validation of dynamic data with multiple verification points and support for regular expression pattern matching
- 5) Reduce rework, minimize the rerecording of scripts, and reduce script maintenance

The figure1 below shows the work flow of RFT

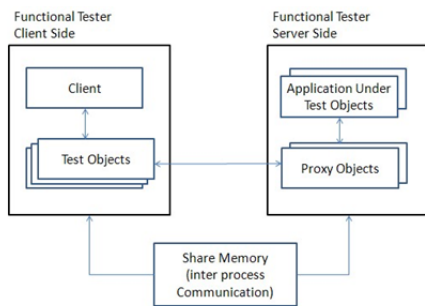


Figure1. Work of RFT

IV. AUTOMATED TESTING FRAMEWORK

The automated testing framework that this paper put forward learns from IBM automation framework using the automated testing tool RFT. The framework is divided into 3 layers from top to bottom, which are test cases layer, tasks layer and objects layer.

A. Layer Architecture Of Framework

- 1) Case layer
Testing script is implemented for special business and function points. The main responsibility of this layer including: reading the parameters from configuration file, parsing and inputting the parameters to each branch test case, calling the reusable scripts and object map scripts, Verifying the consistency of the results

with the expected. Finally recording the script testing log, which based on certain business processes and specific operational processes.

- 2) Task layer
This layer is mainly responsible for logical jump, click order and verification point screenshots. The script in the layer will call the object's functions in object layer. The task layer implement some commonly used testing operations, which including the entry of text content, menu selecting, the choice of the testing box contents, button click, output verification and so on. This layer correspond to object and every object should have its on set of events. For common performance testing in system Z platform, a reusable script can be designed and associated operations(click,enter,select and etc) can be used as the input parameters of the script. Similar method can be used to verification for special object. While the difference is that the input parameters are the properties and the result. Because the most performance testing on system Z platform is the combination of this basic operations,the reusable script is the same. The difference between them is the configuration parameters which are read from configuration file. These scripts can be shared in different tests, write once and multiple shared.

- 3) Object layer
The object layer provides the atomic function of object, such as button click, get text and set text of textfield and so on. In a shot, this layer constituted by the objects and object properties. And this layer mainly responsibility for how to find an object, the number of object in a page, types of object and how to deal with the object after finding out.

The figure2 below is the layer architecture of framework.

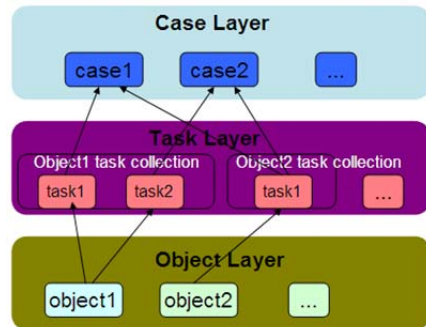


Figure2. Layer architecture of framework

B. Implementation of Framework

In order to more clearly describe the automated testing framework, this paper take the IBM project zMason as an example, which adopt this framework to achieve the auto mated testing. zMason is used to test the DB2 performance as the number of CPU change. In general, this project consists of four steps, they are environment setup, workload run, data measurement and environment clean up. During this procedure, there are three kinds of applications, they are java

web application, batch application and TSO operation application. In terms of these applications, difference methods are used in the object layer. The figure3 is the workflow chart of DB2 performance testing Automation.

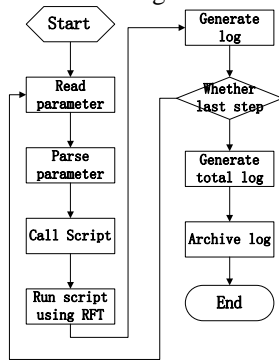


Figure3. Workflow of DB2 performance testing automation

1) Case layer

The script in the case layer is the executable script and in the uppermost of the framework, which also has a navigation effect. The function of the script including:

- a) Parsing the test cases related environmental parameters form the property file.
- b) In accordance with test procedure, calling the scripts in task layer. Each step will return the statue of previous step. After verify the result is true, and then go next.
- c) Recording the log information of the procedure, such as execution time, exception information and so on.

```

public void testMain(Object[] args)
{
    //Get the property
    Input_Properties pro = new Input_Properties();
    String DataSharingFlag = pro.getDataSharing().trim();
    String DB2s[] = pro.getDB2s().trim().split(",");
    int RNum = Integer.valueOf(pro.getClosregions().trim());
    String Mode = pro.getMode().trim();
    String WLType = pro.getWorkloadtype().trim();
    int TlimitNum = Integer.valueOf(pro.getClosthreadlimit().trim());
    int UTINum = Integer.valueOf(pro.getUti().trim());
    int WarmupTime = Integer.valueOf(pro.getWarmupduration().trim());
    int MachineCFU = Integer.valueOf(pro.getMachinecpu().trim());

    //Main Procedure
    Step1_PUTTY step1_putty = new Step1_PUTTY();
    step1_putty.testMain(args, MachineCFU);
    this.sleep(5);
    Step1_HMC_Adjust_CP step1_hmc_adjust_cps = new Step1_HMC_Adjust_CP();
    step1_hmc_adjust_cps.testMain(args, DataSharingFlag);
    this.sleep(5);
    Step2_HMC_IPL_P58 step2_hmc_ip1_p58 = new Step2_HMC_IPL_P58();
    step2_hmc_ip1_p58.testMain(args, DataSharingFlag);
    this.sleep(5);
    RTW rtw = new RTW();
    rtw.testMain(args, DB2s, Mode, WLType, TlimitNum);
    this.sleep(5);
    Measurement measurement = new Measurement();
    measurement.testMain(args, UTINum, WarmupTime);
    this.sleep(5);
    Clean_Up clean_up = new Clean_Up();
    clean_up.testMain(args);
}
  
```

Figure4. The code of case layer

The demo in the figure4 is case layer code. First of all, parsing the environmental parameters, such as DB2 numbers, CICS thread limit and so on. Then calling the script in the task layer. Step1_PUTTY, Step1_HMC_Adjust_CPs and HMC_IPL_P58 are used to setup environment. RTW is used to workload run. Measurement is used to collect performance data. Clean is used to clean up environment at the end of the procedure. Sleep is to slow the procedure.

2) Task layer

The task layer script is primarily used for playback procedure. This layer control the operation sequence, operation content, accident handling, verification checkpointing and so on. It is a relatively simple part and depends on test content. If test content is more complex, the code is more.

```

//If DataSharing
if(DataSharingFlag.equals("YES")){
    image_6().click(RIGHT);
    html_itemU57().hover();
    html_itemU62().hover();
    html_mi487().click();
    sleep(2);
    button_customizeProfilesubmit2().click();
    sleep(4);
    link_v73d_treeSel().click();
    sleep(2);
    text_v19b46().click();
    browser_htmlBrowser(document_hmC15CustomizeDeleteA(), DEFAULT_FLAGS).inputKeys
    sleep(2);
    button_savesubmit().click();
    sleep(2);
    button_oksubmit5().click();
    sleep(4);
    button_closesubmit().click();
    sleep(2);
}
  
```

Figure5. The code of task layer used to judge DB2 model

The demo in the figure5 is the branch code. Based on the environment parameter, the script will setup DB2 data sharing model or stand alone mode.

3) Object layer

The object layer is in the bottom of the whole framework and provides function to task layer. This layer is responsible for finding the object and operating object, which is object recognition. And object recognition is the core of object layer. zMason project contains two kinds of objects, which are GUI object and TSO session object. GUI object contain button, text box, text field, drop-down box and so on, which exist in java web application and batch application. TSO session object contains TSO terminal, screen object and so on, which exist in TSO operation application. The figure6 below is the object model class chart.

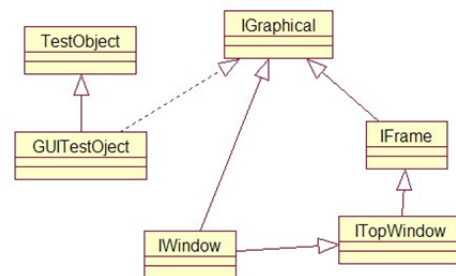


Figure6. Object model class

a) GUI object recognition

All GUI elements are mapped to object whatever GUI library based on. RFT provides an object model framework for management of a variety of GUI elements. The object recognition used by RFT can be divided into two types, which are static recognition and dynamic recognition. Static recognition adopt mapping model. The mapping file will automatically generated by RFT after

recording the operations. Dynamic recognition is achieved through defining find method of RationalTestScript or TestObject.

```
// Whether a Disconnected Session
TestObject[] NewSession_button =
browser_htmlBrowser().find(atDescendant("class","Html.INPUT.submit","value","New Se
if(NewSession_button.length==1){
timerStart("R148ChooseDisconnectedSessionhtt_11");
new GuiTestObject(NewSession_button[0]).click();
timerStop("R148ChooseDisconnectedSessionhtt_11");
}else{
System.out.println("new session button not exist!");
}
}
```

Figure7. Dynamic object recognition

The demo in the figure7 is the part of accident handling code, which is used to judge whether a disconnected session. Based on the properties (.class .value), finding the new session button on the html page. If the button exists, click it and then start a new session to logon the HMC. If not skip this step and reconnect the previous session to logon the HMC. Timer is just for statistics.

b) TSO Session object recognition

RFT provide good support to TSO operation application. FTE is a plug-in RFT provide which is used by terminal-based application to connect the host. TerminalTesterOps is the API provided by TSO library. TSO session object will be got with the help of API. Then through the functions of this object, the TSO operations automation will be achieved.

```
public void Logon(String IP){
ops = null;
if(IP == "P58"){
ops = new TerminalTesterOps("P58.conn");
}
else if(IP == "P93"){
ops = new TerminalTesterOps("P93.conn");
}

RationalTestScript.sleep(4);
terminal = ops.getTerminalScreen();
terminal.activate();
terminal.setText("1 userID");
RationalTestScript.sleep(2);
terminal.enter();
RationalTestScript.sleep(2);
terminal.setText("password");
RationalTestScript.sleep(2);
terminal.enter();
RationalTestScript.sleep(2);
terminal.enter();
RationalTestScript.sleep(2);
terminal.enter(2);
RationalTestScript.sleep(2);
}
```

Figure8. Tso session object recognition

The demo in the figure8 is used to logon TSO. Ops is TerminalTesterOps object and terminal is TerminalScreen object, both of which are API provided by the library. Firstly, initializing the ops with the connect file as parameter. Then, using the function of ops to get the terminalscreen, after activating the terminal, the FTE will pop-up automatic to connect the host. Finally, using the

function of terminal to input the ID and password to complete the logon.

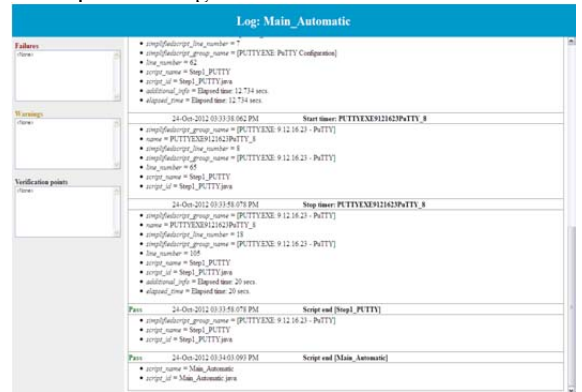


Figure9. Run log of zMason automation

The figure9 are the run log of zMason Automation, which use the automated testing framework put forward in this paper to achieved three kinds of application automation.

V. CONCLUSION

This paper firstly introduces the main work of DB2 performance testing. The whole process takes two hours of continual attention and all operations are manual operation. This complicated and mechanical procedure is unacceptable. For improve the efficiency, an automated testing framework was put forward and IBM zMason project was taken an example to introduce the implementation details of the framework. This framework is reusable rather than simple script recording, enhancement and playback. Any testing on Z platform project can adopt the proposed framework to achieve automation and only work is to modify the function code.

ACKNOWLEDGEMENT

This paper was funded by IBM CDL (China Development Laboratory)

REFERENCES

- [1] DB2 Rapid Deployment of z/OS and DB2 User 's Guide.IBM RedBook. (Nov,11, 2011).
- [2] DB2 10 For z/OS Technical Overview. IBM Redbook. (June 12, 2011)
- [3] Testing Object-Oriented Systems: Models, Patterns, and Tools by Robert V. Binder (2011)
- [4] Using IBM Rational Functional Tester to test financial services applications by Michael Kelly (2009)
- [5] Improving the Maintainability of Automated Test Suites by Cem Kaner (2007)

Totally Data-Driven Automated Testing by Keith Zambelich(2008)