A Multi-Threaded Test Automation Framework for Testing Data-Centric Applications using Data Processing Algorithms

Chandra J
Dept. of Computer Science
Christ University, Bangalore

Kaushik R
Dept. of Computer Science
Christ University, Bangalore

Vishal D Souza
Dept. of Computer Science
Christ University, Bangalore

Joy Paulose
Dept. of Computer Science Christ University, Bangalore

ABSTRACT
Testing data-centric applications is always a challenging and a time consuming task. The goal of this paper is to present a test automation framework for testing data warehouse and business intelligence applications using an open source test automation tool. An Implementation methodology for multithreaded test execution is be explained that brings down the test execution time significantly. It is a rigorous task to test the applications with high dimensionality of data as the number of test inputs is huge. Over and above it requires a huge amount of time to test the data-centric application as the test cases needs to be executed with various input combinations to bring out the application failures. An effective way for selecting quality test data using boundary value analysis, equivalence portioning and orthogonal array technique is presented in this paper. A methodology for preparing a quality data set for testing using various pre-processing and data mining techniques are experimented. A Strategy data selection and reduction for effective testing is described A mechanism for reducing the test inputs to perform a risk oriented testing is also presented. Overall, a step by step approach for testing high dimensional data-centric web applications compressively, using selenium, weka and R tools is presented.

General Terms
Test Automation Framework, Software Testing

Keywords
Test Automation Framework, Orthogonal Array testing, Test Data preparation, Selenium web testing, Risk Oriented Testing, Data Driven Testing, Boundary Value Analysis, Equivalence Partitioning, Rule based data extraction engine (RBDEE).

1. INTRODUCTION
Testing is an inevitable part of the software development life cycle and usually constitutes to more than 35% of overall development effort [1]. Testing data centric applications is a big challenge, since it would require a lot of time to test various combinations of data. Testing Data warehouse applications are very critical because the number of users for any application increases based on the quality of data and the interestingness of the information gained form the applications. Testing the data warehouse applications manually is a monotonous task for the testers as the testers have to repeat the same test with huge data set [2]. So automating the test cases can bring in a lot of sanity into the testing activities. Test Automation also reduces the turnaround time for testing teams, which reduces the overall defect distribution across modules.

Automating the test cases can be usually expensive because the automation testing tools come with a huge price tag [3]. A test automation tool can cost anywhere between $1200 and $ 8500 per seat per annum; but this tool cost can be avoided by using an open source test automation tool called selenium. Unlike other open source tools, selenium is regularly upgraded to match the current web technology. Selenium is properly supported by several online forums with a huge number of active users.

Selenium is one of the best test automation tool for automating web based applications. Selenium also supports cross browser testing, i.e., the applications can be tested on various browsers. Selenium also supports testing web 2.0 and HTML 5 applications and this makes selenium a perfect choice for testing data-centric applications like e-commerce sites. Selenium test scripts can be written in different languages like Java, Ruby, C# and PHP, thus reducing the learning curve for test engineers [4]. Selenium Server allows parallel test execution. Executing the test cases in parallel reduces the test execution time thus making it a perfect tool for testing data-centric applications.

Virtualization and cloud-based technologies can be used to reduce the hardware cost and increase the efficiency of the testing process. These technologies can come in handy to improve the test automation efficiency [5]. Virtualization reduces the need to invest in the physical hardware needed to set up the test environment. Using Virtualization techniques web applications can be tested in parallel with different browser and operating system combinations. Selenium server has an unmatched support for testing web applications in virtually, executing test cases in parallel, reducing costs, and increasing speed of test execution and code coverage.

Selenium server can be used as a multi-threaded test execution component. The MTE component can seamlessly distribute the test scripts across multiple machines [6]. These can be either physical or virtual machines. This ensures that the test cases are executed in parallel and thus bringing down the time required for test execution. So the proposed methodology drastically fosters the testing efforts, giving quick and accurate feedback to the developers. With the MTE component, the existing hardware infrastructure can be leveraged. Multiple test cases are run in parallel by using the different data sets, on multiple hosts in a heterogeneous environment with different OS and browser combinations. MTE component can be used to run multiple instances of selenium server in parallel. MTE component brings down the time required to run the Selenium test cases to a fraction of the time that a single instance of test execution component would take to run.

2. MOTIVATION

Testing the data warehouse applications end to end is not practically possible because of the size of the data set [7]. It is also important that most of the data combinations are tested so that all the errors are surfaced out [8]. In most of the cases the testers select a random data set to test the application, which does not guarantee the quality of application. So it is important to select the right data set for testing so that the most of the defects are found during the test execution. Selecting the right data set for testing is a big challenge. Following are the major complications associated with testing data warehouse applications.

1. The source data comes from a variety of data repositories and are usually very big.
2. The quality of the data set for testing is not guaranteed as it is not pre-processed and cleaned.
3. The data might consist of some inconsistent values and there could be redundant data that won't add any value during testing [9].
4. It is also possible that some of the data could be missed out during the sourcing process and these data might be really important for testing.
5. There are no readymade methodologies that can extract the required test data based on boundary value analysis [10] and the equivalence partitioning [11]. Using random data set for testing leaves the applications untested for various scenarios.
6. Most of the times, the testing team depends on the business logic written by the developers to extract the test data set and this approach might seed defects into the application as the business logic written by the developers can be erroneous.
7. Although test automation tools are available for conducting data driven tests, executing them sequentially is time consuming.
8. Regression testing for data warehouse applications is a challenge as the business logic and the data are updated very often.
9. Most of the time, testers get very little time for providing the feedback.
10. Though some tools have the capacity to extract quality test data, sometimes the data set becomes too huge and testing the applications for a quick turnaround time is not possible.
11. There are some proprietary test automation tools for testing, but these are costly and small and medium sized organizations cannot afford it.

A study of existing proprietary testing tools was conducted and the **a**pproximate cost of these tools is listed in table 1.

Table 1. Approximate Tool Cost

Tool	Term	Cost	No. of Users
QuerySurge	Annual	$5,100	2
Ab Initio	Perpetual	$5,00,000	50
RapidRep Test Suite	Annual	$699	1
TestRail	Annual	$239	1

3. THE PROPOSED SYSTEM

The objective of the proposed system is to perform test automation of data-centric applications comprehensively and quickly in an effective way using various data processing algorithms. A step by step approach for testing the data warehouse applications is presented. The following are steps are followed to test the application with high dimensional data.

1. The data is sourced from various heterogeneous sources and is be stored in a centralized repository called Data Pool.
2. As the data in data pool is sourced from various sources, the data needs to be cleaned to fill the missing values, smooth noisy data, remove the outliers and resolve inconsistencies.
3. The data obtained after the cleaning processes is stored in a separate container called 'data repository'.
4. FP-Growth Algorithm is applied to the 'data repository' to eliminate the frequently appearing data sets [12]. Using redundant data for testing won't yield any new defects. The resultant data from this technique is called as 'Total Test Data Mart'
5. As it's not practically feasible to test the application with 'Total Test Data Mart', declarative business logics are applied to extract test data based on limit values, equivalence classes, negative values, exception values. The data obtained from this process is called 'Test Data Mart'.
6. The 'Test Data Mart' is further reduced using orthogonal array technique [13]. This is called 'Reduced Test Data Mart' and is be used for risk oriented testing.
7. Test script is created using an automation tool as per the application flow. The test script is parameterized with the 'Test Data Mart'.
8. The test automation script created using the tool is then distributed to several machines. The test scripts are then run parallel.
9. In case if there is limited time the tests are run using 'Reduced Test Data Mart'.
10. Multithreaded test execution component also logs the results of each and every execution which is analyzed to find the defects in the applications.
11. The proposed test automation framework also contains a suite file that supports selective execution.

The proposed test automation framework is implemented using Java as programming language as the selenium server supports scripting in Java language. The overall approach is divided into the stages (a) Planning (b) Preparation and (c) Execution [14]. The planning stage involves identifying the test cases to be automated and creating the test script for the test cases. The preparation stage involves cleaning the data, reducing the data and parameterizing the data with the test script. The execution stage involves distributing the test across several machines and executing the test cases in parallel. The overall test automation framework is shown in the Figure 1.

4. DATA SOURCING AND CLEANING

At the outset the data from various heterogeneous sources is collected into a data pool. This sourced data is then fed to a data cleaning system. Data cleansing is an essential step towards achieving quality data repository. Performing cleansing early in the testing life cycle ensures the success of testing and helps eliminate the surprises. In this stage of testing the data is cleaned by removing noisy data, outliers and inconsistent data. For cleaning the data, Weka 3.7.10 APIs are used [15]. Weka is an open source ETL framework. Weka classes provide the methods needed to perform basic data cleansing and processing functions against the incoming data from the data pool. Weka classes are used for finding potential duplicate values. It is also the used for adding and updating records. There are several classes to support the cleaning process.

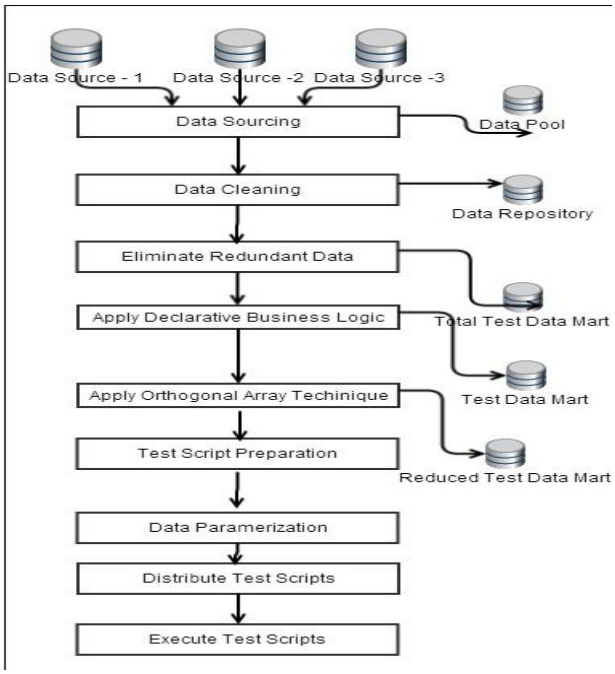

Figure1. Proposed Framework for Automation Testing

Weka APIs can be integrated with eclipse, which is also be used for building selenium test scripts and suites. The missing values in the in the data pool are replaced using techniques like global values, attribute mean values, attribute mean for all samples belonging to the same class or by using a probability value. The techniques differ from data set to data set.

So when the missing values are replaced, caution has to be exercised based on the data set. The eclipse is integrated with Package Amelia II [16]. Amelia II is a package developed by R [17]. Amelia II is an efficient program to generate the missing values. This program can be combined with any statistical method based on the data set. A pictorial representation of the cleaning process is shown in figure 2. The entire process is implemented using R package bundle and java development kit.

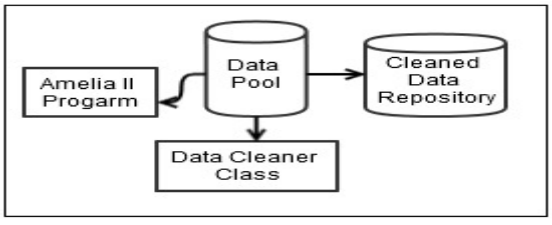

Figure 2: Process of Data Cleaning Process using Amelia II

5. CLEANING REDUNDANT DATA SET

Eliminating redundant data from the data pool is an important aspect to increase the efficiency of test data mart produced. There is no point in using redundant data during testing as it won't be able to find any new defects. The redundant data can be eliminated by finding the frequently appearing data items. This can be done by using a data mining algorithm called FP-Growth algorithm. The Algorithm is applied on the cleaned data using the packages available with R. The R tools are integrated into the eclipse, which is used for generating the selenium scripts. Using the R tool a Java class is created to identify the frequently occurring data sets. Once the frequent item sets are discovered. After identifying the frequent items are found, all the redundant data are removed. The outcome of this operation is a data set without redundant data items. The reason for choosing FP-Growth algorithm is that it's the most cost effective algorithms in terms of CPU and memory.

6. CREATING TEST DATA MART

It has been already discussed that the testing an application with entire data set is not possible. In order to test any application comprehensively it is important to test the application with five different classes of values and they are (a) Positive, (b) Negative, (c) Boundary, (d) Equivalence, and (e) Exception classes. In this stage of testing phase the total test data mart is be processed to obtain data items matching these five kinds of values. This can be achieved by embedding the RBDEE into the total test data set. This RBDEE is a background daemon process. The test values depend on the business domain that change from business to business, which poses a change to have a generic rule for extracting the values. In order to overcome this challenge, the rules are separated from the RBDEE. The rules are generated from the training data using prism algorithm [18]. Prism is a classification algorithm and is used to separate the test data mart from the total test data set. The prism algorithm is implemented as a Java class which first reads the training data and establish the rules. The success of the algorithm depends on the quality of the training data. Concept based learning and Lazy learning techniques are used.

In concept based learning algorithm is used to extract the rules from the training data. The training data consists of pre-classified data. Then the rules are evaluated on the data to extract the test data mart [18]. In lazy learning, the data mart is generated from the training data without creating any explicit models [19]. This is implement using a popular approach called Majority Predictor using ZeroR in Weka.

The RBDEE is implemented as a Java class. The RBDEE call extends the PRISM class and this class evaluates the total test data set to generate the test data mart. This class runs as multi-threaded process to speed up the extraction process. To extract the data quickly the whole data set is be divided into several parts and each thread works on one of the parts. The output of this stage is a test data mart that can be used by the selenium script to execute the script.

7. CREATING A REDUCED TEST DATA MART

During an emergency release, it is not possible to conduct a test with all the present data in the test data mart due to time constraint. In such cases a risk oriented approach is used. The test execution time can be reduced is by reducing the size of the test data mart; but this reduction can lead to a defect leakage. Intelligent data selection techniques produce quality test data mart. To generate a quality reduced test data mart, orthogonal array transformation technique is used. Orthogonal array testing

strategy is a statistical and systematic pair-wise testing technique. Orthogonal Arrays find more defects with fewer inputs without compromising on coverage. Orthogonal arrays can bring our most of the defects using minimum number of experiments. The test data mart can be transformed into an orthogonal array by using accelerated line search algorithm for simultaneous orthogonal transformation package called GARCH (GO-GARCH) models available with R-Packages [20]. The implementation is be done though a Java class which takes the test data mart as input and generate reduced test data mart. An example of orthogonal transformation is shown in Table 2.

Table2: Orthogonal Array Transformations

Possible Values for Each Data Item		
Nationality	Gender	Category
Indian	Male	Child
Foreigner	Female	Adult
Test Data Set before applying Orthogonal Transformation		
Nationality	Gender	Category
Indian	Male	Child
Indian	Male	Adult
Indian	Female	Child
Indian	Female	Adult
Foreigner	Male	Child
Foreigner	Male	Adult
Foreigner	Female	Child
Foreigner	Female	Adult
Test Data Set After applying Orthogonal Transformation		
Nationality	Gender	Category
Indian	Male	Child
Indian	Female	Adult
Foreigner	Female	Child

The reduced test data mart formed by orthogonal transformation has the potential to uncover most of the defects. This approach has to be used only during emergency releases, where there is very limited time for testing.

8. TEST SCRIPT GENERATION.

Once the Data mart is prepared, the test scripts are generated. The test scripts are generated using selenium IDE. The selensese scripts are then transformed into Java code. The scripts are executed using selenium server and testNG framework [21].

The test script are composed of four mandatory methods. The SetUp method takes care of initializing the test and TearDown method takes care of terminating the test. The actual test flow is present in Main Test Method and the DataProvider Method reads the data form Test Data mart or reduced test data mart and execute the test cases for all data items. All the test scripts are set to extend SeleneseTestBase class. This class provides all the primitives for executing the selenium test scripts. Figure 3 shows the overall solution that is be built in eclipse. This figure shows only the signatures and not the actual code. The Entire scripting and execution is done using Java and Eclipse.

9. MULTITHREADED TEST EXECUTION

Once the scripts are created, the test execution starts. Executing the test cases in sequential takes a lot of time. In order to execute the scripts faster selenium server is used. Selenium Server allows us to execute the test in parallel from several machines [22]. For this either physical machines are virtual machines from cloud infrastructure are procured.

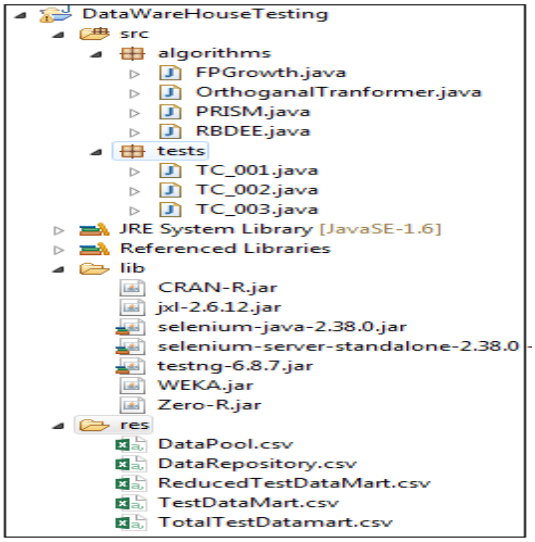

Figure 3: Implementation in Eclipse

Virtualization is a better option as the provisioning time is less and return of investment was found to be high. For executing the test cases using a selenium server, several physical and virtual machines are used depending on the need and availability. One of the machines is considered as server and the remaining machines are considered as remote hosts.

The server distributes the test cases across several machines remotely. Once the test cases are distributed across several machines the server sends commands to all the remote hosts to execute the test cases in parallel. The execution happens with test data mart or reduced test data mart depending on the time constraint. Once the execution starts the results are dumped to the server machines. The overall execution process is described in figure 4.

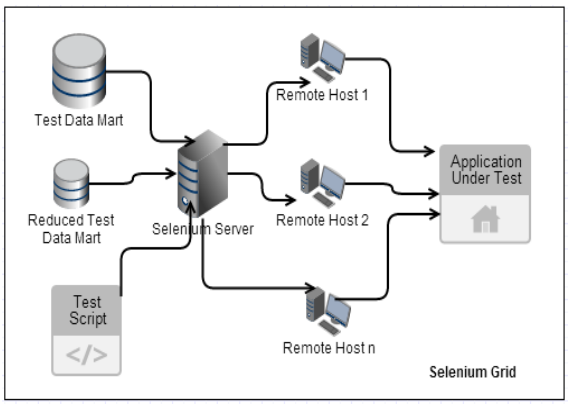

Figure 4: Multi-threaded Execution Process

10. CONCLUSION

This work presented an overall approach for implementing a data warehouse and business intelligence application testing framework which can execute test cases faster. Frameworks for extracting test data mart using various algorithms were presented. A way to reduce the test data mart using orthogonal array transformation was experimented. A mechanism for executing the test scripts in parallel which are developed using selenium and java is explained. This implemented provided is used for automating data-centric applications built on web technology. The proposed work can be extended to automate web services, business centric proprietary application and mobile applications using selendriod.

11. REFERENCES

[1] Burnstein, I.; Suwanassart, T.; Carlson, R., "Developing a Testing Maturity Model for software test process evaluation and improvement," Test Conference, 1996. Proceedings., International , vol., no., pp.581,589, 20-25 Oct 1996 doi: 10.1109/TEST.1996.557106

[2] M. Golfarelli and S. Rizzi, "A Comprehensive Approach to DataWarehouse Testing," in ACM 12th international workshop on Data warehousing and OLAP (DOLAP '09) Hong Kong, China, 2009.

[3] Mustafa, K.M.; Al-Qutaish, R.E.; Muhairat, M.I., "Classification of Software Testing Tools Based on the Software Testing Methods," Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on , vol.1, no., pp.229,233, 28-30 Dec. 2009 doi: 10.1109/ICCEE.2009.9.

[4] Holmes, A.; Kellogg, M., "Automating functional tests using Selenium," Agile Conference, 2006 , vol., no., pp.6 pp.,275, 23-28 July 2006, doi: 10.1109/AGILE.2006.19.

[5] Wissink, T.; Amaro, C., "Successful Test Automation for Software Maintenance," Software Maintenance, 2006. ICSM '06. 22nd IEEE International Conference on , vol., no., pp.265,266, 24-27 Sept. 2006 doi: 10.1109/ICSM.2006.63.

[6] Zhen Li.;Yong Hu Sun.;, "Use Selenium Grid to enhance testing of web applications", "IBM Techinical Library",07 June 2011.

[7] Sneed, H.M., "Testing a Datawarehouse - An Industrial Challenge," Testing: Academic and Industrial Conference - Practice And Research Techniques, 2006. TAIC PART 2006. Proceedings , vol., no., pp.203,210, 29-31 Aug. 2006 doi: 10.1109/TAIC-PART.2006.27.

[8] Kuhn, D.R.; Reilly, M.J., "An investigation of the applicability of design of experiments to software testing," Software Engineering Workshop, 2002. Proceedings. 27th Annual NASA Goddard/IEEE , vol., no., pp.91,95, 5-6 Dec. 2002 doi: 10.1109/SEW.2002.1199454.

[9] Caniupan, M.; Placencia, A., "Data Warehouse Fixer: Fixing Inconsistencies in Data Warehouses," Computer Science Society (SCCC), 2011 30th International Conference of the Chilean , vol., no., pp.28,32, 9-11 Nov. 2011 doi: 10.1109/SCCC.2011.5.

[10] Ramachandran, M., "Testing software components using boundary value analysis," Euromicro Conference, 2003. Proceedings. 29th , vol., no., pp.94,98, 1-6 Sept. 2003 doi: 10.1109/EURMIC.2003.1231572.

[11] Reid, S.C., "An empirical analysis of equivalence partitioning, boundary value analysis and random testing," Software Metrics Symposium, 1997. Proceedings., Fourth International , vol., no., pp.64,73, 5-7 Nov 1997, doi: 10.1109/METRIC.1997.637166.

[12] Min Chen; Xuedong Gao; HuiFei Li, "An efficient parallel FP-Growth algorithm," Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009. CyberC '09. International Conference on , vol., no., pp.283,286, 10-11 Oct. 2009, doi: 10.1109/CYBERC.2009.5342148.

[13] Maity, S.; Nayak, A., "Improved test generation algorithms for pair-wise testing," Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium on , vol., no., pp.10 pp.,244, 1-1 Nov. 2005, doi: 10.1109/ISSRE.2005.23.

[14] Glicker, S.; Hosch, F., "A design approach for a distributed test automation system," Applied Computing, 1990., Proceedings of the 1990 Symposium on , vol., no., pp.9,11, 5-6 Apr 1990, doi: 10.1109/SOAC.1990.82132.

[15] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[16] AMELIA II: A Program for Missing Data. James Honaker, Gary King, and Matthew Blackwell. Version 1.7.2. June 8, 2013. (internet) http://cran.r-project.org/.

[17] Ying Wah Teh; Abu Bakar Zaitun; Lee, S.P., "Data mining using classification techniques in query processing strategies," Computer Systems and Applications, ACS/IEEE International Conference on. 2001 , vol., no., pp.200,202, 2001 doi: 10.1109/AICCSA.2001.933977.

[18] Feature Extraction, Construction and Selection: A Data Mining Perspective edited by Huan Liu, Hiroshi Motoda, Kluwer Academic Publishers, 2001.

[19] C Alexander. A primer on the orthogonal GARCH Model, 2000, URL http://www.icmacenter.rdg.ac.uk/pdf/orthogonal.pdf (internet) http://testng.org/.

[20] Unmesh Gundecha, "Distributed Testing with Selenium Grid" , Packt Publishing, November 2012.