Proceeding of the IEEE
International Conference on Information and Automation
Hailar, China, July 2014

Research of Performance Test Technology for Big Data Applications

Zhenyu Liu

Shanghai Key Laboratory of Computer Software Testing and Evaluation
Shanghai 201112, CHINA

lzy@ssc.stn.sh.cn

Abstract - **This paper studies the performance test in big data application. The existing performance testing technology is not suitable for the big data application. The paper proposed test technology for performance testing. The technology provided test goal analysis, test design, load design for big data application. The character for different application could be supported to consider specific multiple test data design method under this framework. The performance technology is used to test some applications and proved effective.**

Index Terms – Performance Test, Software Test, Big Data.

I. INTRODUCTION

Big data has become increasingly significant, any hope of success in extracting value from big data companies, are facing a revolution. The classical techniques cannot meet the performance test needs to test a bid data application, adapted to "big data applications" feature will facilitate performance testing to improve and optimize system performance.

Performance testing is a test method, which belongs to a non-functional testing. In performance testing, the system will to be tested by simulating a variety of normal and abnormal peak load conditions, in order to reduce operating, upgrade or patch deployment risks through performance tests, such as information systems to get the load on the user's response time, the system is an important means to ensure the successful operation of the application system.

The user uncertainty and application characteristics, the user may use application at various times. After performance tests, the application system could crash down at any time. As for big data application systems, performance evaluation is important. On the other hand, there is a huge correlation between the number of sample size distribution characteristics of performance and system data in big data application. The data play an important role in performance tests, analyze test data models.

A large amount of data is very large-scale data processing. Therefore, performance testing is very necessary for big data applications, big data applications through performance testing, we can achieve the following objectives:

1) obtain the actual performance of big data applications, such as response time, maximum online user data capacity size, and a certain maximum processing capacity;

2) access performance limits and found that the conditions can cause performance problems, such as testing under load is applied to some problems can occur after a long run in big data application;

3) achieve performance status and resource status in big data application, and to optimize the performance parameters in big data applications (eg, hardware configuration, parameter configuration and application-level code).

The big data application will handle a large number of structured and unstructured data. The data processing will involve more than one data node and completed in a shorter period of time. Due to the low quality and poor system design code, application performance as data volume growth will decline, even when the amount of data reaches a certain size, the application crashes and cannot provide mission services. If the performance of the application does not meet the service level agreements (Service-Level Agreement, SLA), will lose the goal of building big data systems. Therefore, due to data capacity size and complexity of systems in big application, performance testing has played a very important role to achieve the actual performance ability.

This paper describes the performance test methods and strategies of big data applications, focusing on applications analysis supporting test data design, testing model, to ensure high efficiency operation of big data-based applications.

The main structure of this paper is as follows: the second part of performance testing technology is given. Section 3 introduces the test framework. The next part gives test design and test development. The fifth section illustrated a detailed experiment. Finally, there are conclusions and future works.

II. PERFORMANCE TESTING

The main characteristics in the performance test consist of response time, resource use and stability.

The purposes of performance testing are not only acknowledging application performance levels to, but aim to improve the performance of the big data application. Before performance testing, test engineers should fully consider their testing requirement. And then test engineers design a complete test scenario to consider the test program with the actual situation of the user operation. Through the test execution and results analysis, performance bottlenecks can be found and analysis the reason further. In the performance test, test engineers need to collect the resource use information during performance test execution. Related to response time, the collecting resources use information, the more obtained performance information analysis, and the more analysis of system performance bottleneck.

Not only for big data application infrastructure, data processing capabilities, network transmission capacity in-depth testing, but also from the basic characteristics of big data to analyse the factors affecting the performance of big

978-1-4799-4100-1/14/$31.00 ©2014 IEEE

data applications. In big data applications, the rapid growth of mobile computing and network users, mobile devices, changing only the type of data occurs, and the data generated is very rapid with increase of real-time data transactions.

Large amounts of data within the enterprise or outside the enterprise to create through the network and computers, such as traditional computer or mobile terminals. Enterprise data are of concern to grow exponentially every year. So these data were obtained from multiple different applications and need to handle and analyse further. The challenge is to verify whether the data is processed correctly. Manually verify that all data is an extremely boring and repetitive activities. Therefore, test scripts in automated tools are used for data validation. For storage in HDFS, test engineers can compare difference between extract files and scripts written by tools [2]. In some extreme cases, it takes a lot of time to make comparison 100% of document.

In big data applications, diversity of data type is reflected not only in the growing and diverse data sources, including equipment, sensors, social networks, other applications, and so on. Moreover, the data have many unstructured data compared to traditional structured data, like graphics, images, sounds, documents, etc. Between and structured data, there are semi-structured data. These data generally are heterogeneous, lack of integration, the traditional testing process has been unable to adapt to the data processing needs diversity. Performance test data also need to focus on the impact of diversity on performance test.

Data flow is created faster and faster in the enterprise, making the response speed of data when becoming increasingly important. The real-time response on data flow, test engineers are facing a greater competitive advantage. Testing process should focus on how to implement the test data generated in real time, while the rapid response can meet to provide information available to IT organizations, and even bring the management of IT data used in order for the enterprise competitiveness.

III. Test Framework

Big data own capacity, diverse, fast and other features. The performance tests should take into account these characteristics of big data, especially for the efficiency of the application, resource use, stability and reliability.

A. *Test Type*

Performance testing can be divided into concurrent testing, load testing, stress testing, capacity testing.

1) Concurrent test: when testing multiple users to simultaneously access the same block of the same application in the software system if there are performance issues.

2) Load testing: test application performance at a certain load level, which determine the performance of the system in order to ensure that the application can work in a range of needs. With gradually increasing the load, test results should be observed change performance of each application and check whether the system is stable. Load testing is concerned satisfaction of user requests .

3) Stress test: examine the system performance under extreme conditions, that extreme conditions can be overloaded under transaction volume and the number of concurrent users. The extreme conditions are not necessarily the performance requirements of users, even much higher than the performance needs of the user. Compared to load testing, stress tests concern the application itself, such as peak capacity. Also stress test examines the application of the operation of the limit load and found weaknesses in the system.

4) Capacity Test: determine the maximum resources available to support the application or concurrent users. To ensure that the application in its ultimate state without any software failure, limits condition of the application are reflecting characteristics, such as the number of requests processed simultaneously, the maximum number of concurrent users , database records , etc.

B. *Test Goal*

In big data applications, the irrational distribution architecture or data manipulation will result in performance imbalance. For example, MapReduce applications, the input segmentation, redundant move, sort of operation, whether the operation should be considered in the appropriate step polymerization process conducted operations, from Map to Reduce step [5]. Through good architecture design in system, performance tests can identify performance bottlenecks, eliminating these performance issues. Therefore, the performance evaluation of big data applications, the test goal should be considered big data processing features and characteristics of the data of the application itself. The performance was evaluated from the perspective of the user:

1) When considering the data size increases, the growth of big data application response time is linear or random. If the response time changes with the amount of data grew exponentially, the test data set reaches a certain size. The application will quickly reach the performance bottlenecks and can not work properly.

2) consider increasing the size of the data, to analyze the performance test resource consumption with growth curve model when application system change is as linear or exponential.

3) examine the data size increases, the data application system based on whether the large-scale and stable operation for a long time.

4) complexity of the application itself on the performance. In different analysis and processing applications with the same scale or similar scale, the complexity (eg fine, accuracy) to deal with data processing is changes with performance.

C. *Test Indicator*

Big Data application performance test should provide application performance and monitoring indicators. The supply of these different indicators used to focus on the performance of different roles, through in-depth analysis of these indicators of system performance can be further improved or optimized performance.

Performance consists of the front-end and back-end performance properties. Generally, the performance is more concerned with the back-end performance testing while the front-end performance for the user experience also has a very important influence, no matter what kind of product end-user through the front perform the operation. Network as application running indispensable infrastructure will have an impact on system performance. Therefore, the test application is to be concerned about network performance while those particularly affecting the efficiency of the data transmission network. In general, performance testing needs to focus on three aspects of time: the performance of presentation time for the end user, the performance of the data transmission time in network, and the processing time on the rear end of the system performance.

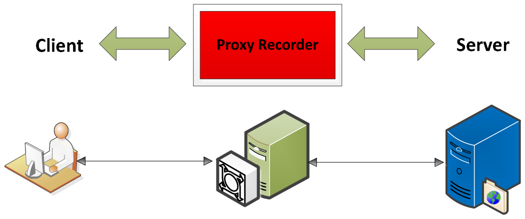

Fig. 1 the time distribution in application

Performance testing of big data applications need to establish a large size and scale of production close to the environment. Use performance monitoring tools can be used to capture performance metrics to identify performance issues, including performance metrics such as response time and throughput.

1) Response Time: the time needed to respond to the request, including the server processing time, network latency and client display time. For users or customers, when clicking a button, link, or issue a directive to submit a form began to apply the results in the form required by the user to show up so far, this process consumes time users of the application performance Characterization of this process is the time required for said response time.

For browser-based (Browser) applications, the response time is the browser to submit a request to receive the interval among the response time to the Web server. When browser download all the elements (including embedded objects, JavaScript files, Cascading Style File CSS, images) on time, the spend time could be achieved for download time to the end-user in initialize the page elements.

2) Throughput: refers to the number of requests processed per unit time system users. For interactive applications, the throughput of the server index reflects the pressure on the load capacity of the reaction system. As for web applications: the application throughput parameter means the HTTP server page successfully processed per unit of time or the number of HTTP requests.

In different applications and different perspective, throughput is not exactly the same denominations, including: the number of requests/sec, transactions/sec, number of pages/sec, the number/day, the number of processing operations/hour. For web-based applications, application on the network for transmission, the number of bytes received per second can be calculated.

D. *Monitor Indicator*

Monitoring indicators known as performance counters also refers to the performance testing process, for example the system resources usage, used to measure resource utilization situation. The main functions of monitoring plays a analysis role in the performance test, analysis application scalability, optimize application performance and locate performance bottlenecks further.

Monitoring indicators have two expression methods. The one is proportion of indicators, such as resource utilization, also known as the occupancy rate, which is defined as the resources actually used total amount of resources available, the proportion of indicators using a percentage. Under CPU, memory, storage, indicators in test consists of the hardware which running the application. For these indicators, the general concern utilization requirements are usually no more than 75-80% is appropriate. The other one is the numerical indicators: describe the data server or operating system performance, these indicators often used numerical representation, such as the use of memory, the process time.

1) user monitoring (peak concurrent users operating system supports users, concurrent users);

2) period of time in the number of online monitoring (custom time period);

3) number of page views and visits within a certain period of time;

4)server parameters, database parameters, JVM parameters.

IV. Test Design

Big data needed to fully analyse application performance testing application characteristics and corresponding testing resources. The supporting data for a test should be design during the phase of test development. Test data creation is also closely related to performance issues. Therefore, test engineers need to focus on the performance test data creation rate. Performing data rate should be create corresponding to test aspects of the recognition performance bottleneck analysis. The system is able to handle high-speed data stream play an important role.

A. *Data Design Principal*

The 4V characteristics of big data are critical impact on application performance. The test engineers must analysed application of the characteristics of the data. All test data will lead to a uniform distribution of random data, not subjectively believe that all randomly generated can simulate the entire real environment. In some scenes all random is unrealistic.

Before the implementation of a big data application performance testing, stress testing resources should be take into account the test cycle. The test case has been most focused on some normal test data. The data application should be considered economists widely used in real Pareto "80/20" distributed in the design of big data, in compliance with the given software Boehm 80/20 distribution: 20% of the consumption of the module 80% of the resources, 20% of the module takes up 80% of the execution time. Therefore, the test designer needs focus on the design of big data representation, data width and data distribution density.

1)Typical

Test data should be representative, not only reasonable in the range of values of the test data. The test data should include unreasonable, unlawful, in cross-border value interval boundaries and test data, and even the limits of the data. In the test design, the test data should be considered deviates from the data, such as the value type of vacancy data, noise data, inconsistent data and duplicate data into four categories:

Vacancy value data: This type data must have for the missing fields of information.

Noise data: The data value generated from the raw data, is not correlated with the original data. Due to noise deviation, uncertainty uncertainty of the actual data is deviated.

Inconsistent data: data is generated by the business system is not perfect. There is no data constraints or simple constraints after you enter the no logic written directly reason, for example, enter 850 (the desired results 85.0). The date format is incorrect when the date crossing the boundary.

Repeat data: data tables are linked in the process, resulting in the merger process data.

The above data not only cannot be deviated from the correct result. In some extent, the data design will affect the performance of big data applications.

2)Universal

Requires extensive test data should contain as many desirable values. Extensive test data should be able to refer generally find an error, rather than repeated testing for a certain error.

Extensive requirements as much as possible to design some test data, can be tested for different content by converting part of its content. Test data should be concerned about every detail mentioned and test every possible case, these conditions must be combined.

In the big data applications, test data for the application characteristics should cover all possible combinations. However, this does not mean endless combinations of data. The combination of test data will lead to the unlimited expansion of test data.

3)Data Distribute

Distribution-intensive data can be understood. The table in the database and the large amount of data, value of a field can decide whether to establish a uniform value of the index cluster in this field. The number of clusters in SQL statement may not create an index on this field performed. Index provides a variety of types of databases, data generated for the index type may also verify whether the established efficient.

When the applications runs in the background to make distributions out of the field operations, a value will be calculated according to a formula based on the value and then distributed to the back of a computing server operation. If a value in too concentrated phase pressure will lead to a distributed too large in computing server, which led to anomalies. For this scenario can be seen in the test data can also affect application performance.

B. Data Generate

Automated testing cost per test performed relatively low, so a test execution is formed by adding a new type of test data to test and a lower cost to extend the existing test data test cases.

Due to the relatively large size of the data, the data needs to design test the specific binding to each other and the data generation method in big data applications. According to the characteristics of big data applications, a variety of methods are used to test data design. Each method has its own data generation design features needed to effectively consolidate using various methods designed to ensure the accuracy of data representation. Therefore a typical density improves the performance test better.

Test design and test development, test data design will involve two aspects, one is the basis of data in large scale data applications, there is no data to support the basis of a certain size, you cannot get real performance indicators. The other is the testing process enter the required test data.

Design of test data including manual, rules, workflow:

1) Manual

Manual design is in accordance with the needs of the general instructions equivalence partitioning class technology and boundary value methods.

Since the need for exhaustive testing data is huge, test engineers may require a large amount of data from the central part of a selected test , the basic assumption is that each part of the data error is found in the same capacity. Equivalence partitioning, including effective equivalence class equivalence class and invalid equivalence classes are based on application requirements specification to design the test data. Valid equivalence class is reasonable, meaningful test data used to verify the functionality and performance specifications related to the regulations. Invalid equivalence class consists of invalid input used to validate the application invalid input filtering and processing capabilities. Equivalence partitioning class method have been classified under ascertain equivalence classes of elements in different ways in the program processing. The equivalence classes can be further divided into smaller equivalence classes.

Designed for a variety of boundary conditions for testing data is boundary value analysis methods. Because of the large number of errors occur in the input or output boundary of the range, but not within the input range. Boundary values analysis method is supplemental test data design techniques. In boundary values analysis, test engineers select the test data equivalence class boundaries. Practice has proved that the design of the test data near the border could get good test results.

2) Rules

In big data applications, the data in a way have certain rules. The existing data dictionary method used to generate the rules is too simple, so the results are not satisfactory.

A regular expression is a regular way of describing syntax, widely used in pattern matching. If the regular expression can be applied to data generation, it will definitely improve the success rate of the test. Regular expressions can use tools, such as Microsoft Visual Studio Team Edition for Database Professionals, to generate meaningful test data. [5]

2) Workflow

Now almost all software used to control the flow of event triggers, event triggers the scene when they formed a scene. The different trigger event sequence will lead to results on the formation of event streams. This idea in software design can also be incorporated into software testing. Test results can be compare vividly depicts the scene, help test designers design test cases when the event is triggered. Test cases make it easier to understand and implement.

C. Load Strategy Design

Load strategy is taken during the test mode to increase the pressure in the process system under test. The big data applications has different model, such as application load model, data sample model.

Test strategy is a collection of performance test execution time taken to execute the request, which requires the testing requirements, data requirements converted into quantifiable, measurable, achievable goals to load performance testing. The load targets were based on different scenarios to choose. During test execution, the load strategy is depending on and be calculated or specify a variety of direct and indirect target, generally be considered the load from the server and multi-cluster network, the client three areas. Automated testing is the basic method of performance testing. In performance testing, automated test provide effective method through automated performance testing tool will simulate huge concurrency under certain scenarios. Therefore, performance testing could achieve real results.

Requirements analysis is the process of testing needs refinement. According to preliminary needs analysis, system performance analysis is determined to select the application load model and select performance indicators to the resources needed to monitor.

Robert B miller described report of the three-level response time in the 1968 [1]. Generally, the response time of operation are classified into three levels: 2,5,8. In three levels, 2 seconds consider excellent, 5 seconds are good and 8 seconds means only accepted. As for other special operations, such as uploading, downloading can be based on the user experience. Peter Bickford found in the investigation of the user response: Even after 27 consecutive feedback, the first 28 operations, the computer allows the user to wait two minutes. The results of the first half in about 8.5 seconds are go away or start to reload. Use the mouse pointer turns into a funnel interface will prompt the user to extend the waiting time to about 20 seconds when using a animated mouse pointer will prompt the user interface of the waiting time more than one minute. In the progress bar, you can allow users to wait until the operation end. The findings are widely used to analyse web response time performance requirements of the definition of a software system. Third-party research shows the page is loaded gradually, the best sequence is first appeared banners, then text appears and last image appears. Under these conditions, the user will tolerate longer wait time, the user will within 39 seconds of delay is identified as good, more than 56 seconds was considered to be poor[6].

There are three methods in load strategy design: data amount, function points, business situation.

1) Data amount

For some time the application is already running, test engineers can refer to the size of the historical data in the database. Simultaneous analysis system used for several years, which every year the number of pieces of data. If we pushed back in three years, five years, we can calculate the size of the required data. For the project has not yet run, because there is no direct reference data growth model, business needs analysis, based on the growth of business growth model to estimate the future data.

2) Function point

In good overall frame design, database design perfect premise is get the user by monitoring usage, the number of page views, analysis of household scale. If access to commonly used functions point and then applied to the test scenario, the test is valid.

Test engineers can select specific features in function point selection. The first one is occurrence of very high frequency, such as a mailbox core business system login, e-mail and other services. The second is that very high use function, for example product managers think it is absolutely not a problem, such as login. The last one is resource utility is very high, such as a business need for access to dozens of data results to the table, or a query that retrieves a large number of data records when submitting the request.

3) Business situation

All performance tests are purpose to reproduce the actual business scenarios. Therefore, the business scene analysis should be fully analysed and selected according to the needs. Business scenarios should be considered from two perspectives time and space. Time means respectively a year, month, day variation of the angle of observation of the system in test, especially the existence of the business peak hours. According to the characteristics of application areas, space is used to observe the existence of the system under different scenarios and trading peak, for example the number of concurrent and general operating time.

V. Experiment

As for a network public opinion monitoring system, full-text retrieval application is installed on the same machine configuration. The stress test is record of each concurrent operation and the average response time for each query page keyword spent is used to observed as well as CPU usage test server. The test data is select 10 typical words (China, Economy, Nanjing, Fazhan, technology and support, technical or support, network or economy, China or the world or technology or market).

During the automation performance test, the distribution of 10 users, 20users and 30users were simulated separately. The scenario is concurrent, thinking time set to 0 and each concurrent operation time is 10 minutes.

TABLE I Test Result

Test data	Response Time (s)			
	1 User	10 Users	20 Users	30 Users
China	0.359	2.563	5.082	7.497
Economy	0.297	2.471	5.146	7.618
Nanjing	0.297	2.614	5.104	7.872
Fazhan	0.266	2.452	5.092	8.481
Technology AND support	0.437	2.537	5.005	8.319
Technology OR support	0.282	2.459	5.012	8.545
China OR Market OR Technology	0.375	2.503	5.095	8.396

V. Conclusions

This paper studied performance test method in big data application. The paper firstly introduces the software automated test background and development and subsequently carried out a study of performance testing framework for automated testing tools. Performance testing is to use a certain number of virtual users to simulate real user and in accordance with the specified load models for the system under a variety of software or equipment operation.

A well designed test technology and related test method could get practice test result in actual big data applications. But for different application, there are various features and test results. In the future, we will study the relation between test data and its test results and other test design automation.

ACKNOWLEDGMENT

The work is supported by Shanghai STCSM Program under Grant No. 12QB1402300, Shanghai STCSM Program under Grant No. 13DZ0500700 Shanghai STCSM Program under Grant No. 13XD1421800.

REFERENCES

[1] Robert B miller,resopnse time in man-computer conversational transactions, 1968

[2] Yang Genxing, Cai Lizhi, Chen Haopeng, Software Quality Assurance, Test and Evaluation, Tsinghua UniversityPress. 2007

[3] Cui Qiliang, Hu Yiming,. Internationalized Software Testing, Electric Industry Press, 2006

[4] Daniel J.Mosley, Brace A.Posey,. Just Enough Software Test Automation. New Jersey: Prentice Hall PTR, 2002

[5] http://msdn.microsoft.com/en-us/library/aa833253(v=vs.80).aspx

[6] Peter Bickford_ Worth the wait! View Source_ Human Interface Online http://devedge.netscape.com/viewsource/bickford_wait.htm

[7] Moris. Software Test Automation. Mechanical Industy Press, 2003

[8] Chen Nengji, QTP Automation Test Practice, Electric Industry Press 2008

[9] Dean J, Ghemawat S. MapReduce:Simplified Data Processing on Large Clusters[C] //Proceedings of the 6th Symposium on Operating Systems Design and Implementation, 2004: 137-149.

[10] Miller R B. Response Time in Man-Computer Conversational Transactions[C]//Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I. ACM, 1968: 267-277.