Global Journal of Computer Science and Technology
Cloud & Distributed
Volume 12 Issue 12 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Performance Test Automation with Distributed Database Systems

By Dr. R. Mahammad Shafi & Mungamuru Nirmala
Sree Vidyanikethan Engineering College, Tirupati, Chittoor (Dist), A.P, India

Abstract - Our previous research paper 'A Focus on Testing Issues in Distributed Database Systems' led us to a conclusion that Distributed Database Systems supports many good engineering practices but there is still place for refinements. A Distributed Database (DDB) is formed by a collection of multiple databases logically inter-related in a Computer Network. Apart from managing a plethora of complicated tasks, database management systems also need to be efficient in terms of concurrency, reliability, fault-tolerance and performance. As there has been a paradigm shift from centralized databases to Distributed databases, any testing process, when used in DDB correlates a series of stages for the construction of a DDB project right from the scratch and is employed in homogeneous systems. In this paper, an attempt is made to describe the establishment of Performance Testing with DDB systems. It focuses on the need for maintaining performance and some techniques to achieve performance in DDB systems. Three sample web based systems are tested by using TestMaker, one of the open source software, in order to highlight the helpful role of performance in the context of testing. The strengths and weaknesses of chosen performance testing tools viz., TestMaker, OpenSTA, and httperf are discussed.

Keywords : *distributed database system, testmaker, openSTA, httperf , performance testing, TPS.*

GJCST-B Classification: *C.1.4*

Strictly as per the compliance and regulations of:

Performance Test Automation with Distributed Database Systems

Dr. R. Mahammad Shafi [α] & Mungamuru Nirmala [σ]

Abstract - Our previous research paper 'A Focus on Testing Issues in Distributed Database Systems' led us to a conclusion that Distributed Database Systems supports many good engineering practices but there is still place for refinements. A Distributed Database (DDB) is formed by a collection of multiple databases logically inter-related in a Computer Network. Apart from managing a plethora of complicated tasks, database management systems also need to be efficient in terms of concurrency, reliability, fault-tolerance and performance. As there has been a paradigm shift from centralized databases to Distributed databases, any testing process, when used in DDB correlates a series of stages for the construction of a DDB project right from the scratch and is employed in homogeneous systems. In this paper, an attempt is made to describe the establishment of Performance Testing with DDB systems. It focuses on the need for maintaining performance and some techniques to achieve performance in DDB systems. Three sample web based systems are tested by using TestMaker, one of the open source software, in order to highlight the helpful role of performance in the context of testing. The strengths and weaknesses of chosen performance testing tools viz., TestMaker, OpenSTA, and httperf are discussed.

Keywords : distributed database system, testmaker, openSTA, httperf , performance testing, TPS.

I. Introduction

A DDB is formed by a collection of multiple databases logically inter-related in a computer network [4]. In a distributed database, the network must allow users to share the data as transparently as possible, yet must allow each node to operate autonomously, especially when network linkages are broken or specific nodes fail. The transparences provided by a DDBMS can be understood as the high level semantic separation of the details inherent to the physical implementation of a DDB. The focus is to provide data independency in a distributed environment.

This way, the user sees only one logically integrated image of the DDB as if it were not distributed. Performance evaluation of database systems is an important concern. However, easier said than done, performance evaluation of database system is a non-rivial activity, made more complicated by the existence of different flavors of database systems fine tuned for serving specific requirements. However performance analysts try to identify certain key aspects generally desired of all database systems and try to define benchmarks for them.

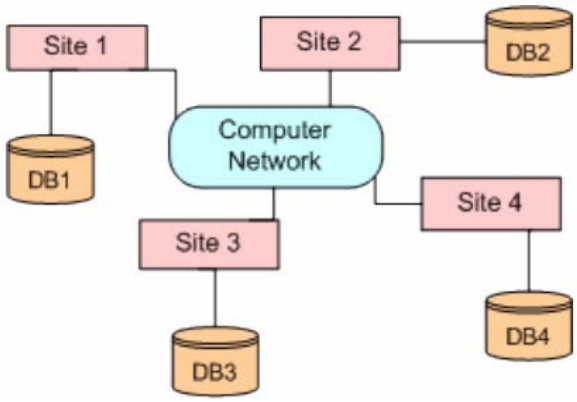


Fig. 1 : A Distributed Database on geographic dispersed Network

II. Overview of Performance Testing in DDB Systems

Distributed applications have traditionally been designed as systems whose data and processing capabilities reside on multiple platforms, each performing an assigned function within a known and controlled framework contained in the enterprise. Even if the testing tools were capable of debugging all types of software components, most do not provide a single monitoring view that can span multiple platforms. Therefore, developers must jump between several testing/monitoring sessions across the distributed platforms and interpret the cross–platform gap as best they can. Testing distributed applications is exponentially more difficult than testing standalone applications.

Neuman [2] categorizes three different dimensions used to measure the Performance of a system, viz., the size of the system, geographical performance, and administrative performance. These measurements ensure that although the users and resources may lie far apart, it is still easy to manage.

Performance testing provides the information regarding whether the system can still function in a fast

Author α : Professor & Head, Department of Master of Computer Applications, Sree Vidyanikethan Engineering College, Tirupati, Chittoor (Dist), A.P, India. E-mail : rmdshafi@gmail.com
Author σ : Research Scholar, Department of Computer Science and Engineering, Senate House,University of Allahabad, Allahabad, U.P.India. E-mail : nirmala.mungamuru@gmail.com

manner under heavy load. This information assists the IT manager to decide about the inclusion of additional servers. Tanenbaum and Steen [5] discussed a number of issues caused by performance. Centralized services, data, and algorithms can become problematic when the number of users increases. Although having a single server to serve a large number of users is not a good idea, adding additional servers to increase the system performance is also not practical in some situations. For example, the information on the server is highly confidential, such as information about bank accounts; adding additional servers create more chances for security attacks.

Another issue is geographical performance which also slows down the communication between clients and server. There are some techniques that can help to maintain and improve performance of distributed database systems. The most common technique is replication which adds more and faster processors as well as design components to be scalable [6]. Several copies of the same server are made available, the requests are sent to the servers based on their physical location, the loads of the server, or can be sent randomly. Consistency should be taken into consideration when this technique is used. It is to ensure that users can only view the up-todate information and the information appears the same to all users. This technique is not suitable when handling sensitive data, which was mentioned in the previous section.

Neuman proposed another technique which is known as distribution [5]. Distribution allows a component to be split into smaller parts which can be spread across the system. For instance, a name space is divided into different parts and a part of the naming database is assigned to different servers. Hence, it reduces the number of queries and updates to be processed; also each request is handled faster since the size of the database is smaller. Caching technique is also introduced by Neuman which allow the result to be remembered, thus, additional requests for the same information can be reduced.

According to Tanenbaum and Steen, hiding communication latencies technique can be used to enhance the performance of a system [5]. That is to say, after the client sends the request, the client continues to do other tasks rather than wait for the response from the server. When the response arrives, the application is interrupted to complete the previous task.

III. Testing the Performance of Sample Web Based Systems

Web databases serve the back-ends of Web-Servers. Web databases are the classic examples of high load, high performance database systems. The OLC Assets, DCRR and Employee Directory are web based systems. These three systems are chosen as samples for the performance test which is to test how each server reacts when there is an increase in number of users. These tests aim to highlight the role of Performance testing in analyzing system performance.

These tests also explain how performance testing can assist testers in making important decisions such as whether the current server(s) are able to serve an increase in the number of concurrent users, or whether additional servers should be implemented in order to handle a high volume of users.

a) OLC Assets

Otago Language Centre receives a number of new items every day to facilitate students in studying English [3]. These items can be library books, hardware, software, TVs, desks, chairs, etc., whose details are stored in the Center's database for future maintenance. OLC Assets System is a web based system that grants user's access to enter details of each item into the database by completing a simple form. The performance of the system is tested under the situation where there are multiple users who have entered a number of items into the database. Figure 2 illustrates the scalability index and XSTest performance index in the context of virtual users and TPS.

b) DCRR

A web based system that provides access to the users to search for a particular restaurant, comment about a restaurant via an electronic evaluation form, and to add a new restaurant into the database. Figure 3 shows two main interfaces of the DCRR system. A number of virtual users accessing the system to add a new restaurant are taken as the scenario for this performance testing.

c) Employee Directory

This web based system allows users to search for a particular employee and view the details of that employee. Two main interfaces of the system are shown in Figure 4. Performance testing is conducted in the situation where there are an increasing number of users searching for a particular employee.

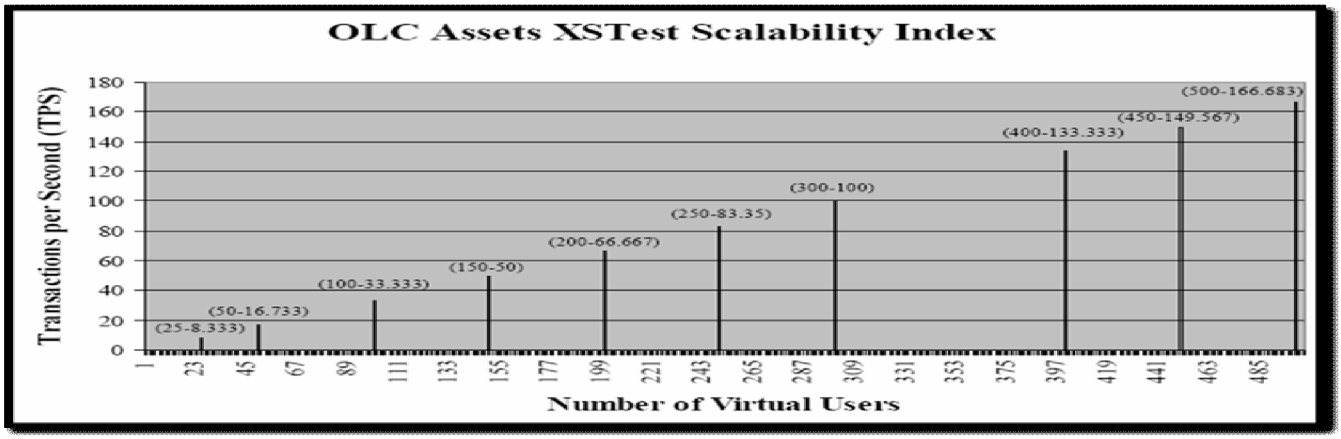
IV. Test Results and Discussion

The three systems are tested using a range of concurrent virtual users (2 - 500) in one minute. Because of the limited resources, the maximum number of virtual users was set to 500. The output graphs (Figure 2, 3, and 4) which starts from 25 virtual users are redrawn based on the original graphs because as they do not show the ratio between number of virtual users appropriately. The original graphs are generated by TestMaker, one of the open source software that supports performance testing.

Each graph describes the number of successful transactions per second (TPS) versus the number of

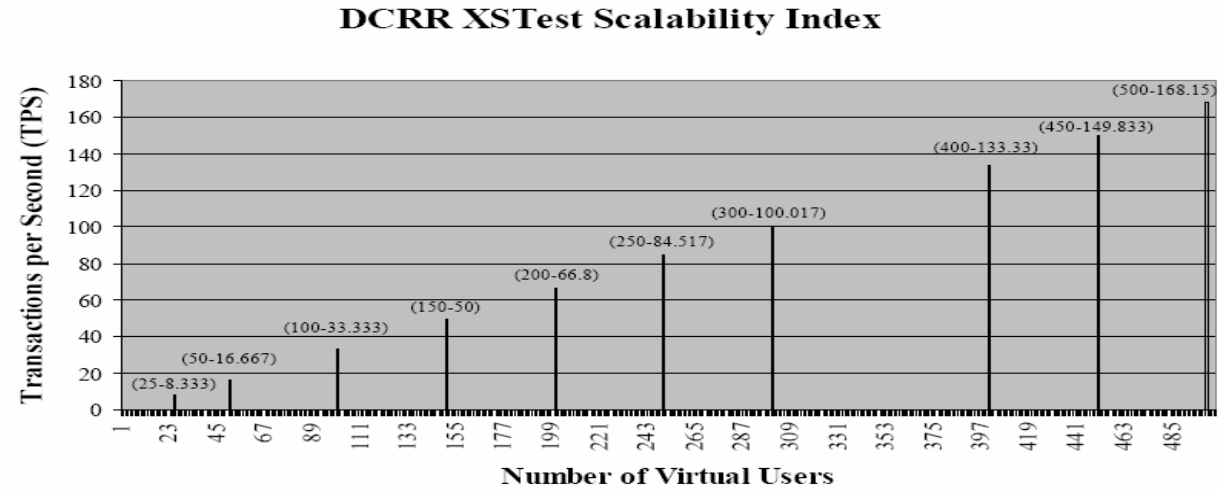
virtual users. The term "transactions per second" is defined as the number of pages per second a server can generate. It can be seen from the graph that as the number of users rises, the number of transactions per second gradually increases.

This indication shows that the server is still able to handle a large amount of concurrent requests. On the other hand, if the number of transactions per second drops as the number of users increases, which indicates that there is a concurrency problem and hence leading to performance problem. Although the results generated verify that the servers are still able to manage a large amount of concurrent requests; it is believed that when the number of users keeps growing, the server will fail to handle these requests at some point. Figure 2 illustrates the OLC Assets System – XSTest Performance Index, Figure 3 indicates the DCRR system performance index and Figure 4 shows the Employee Directory system performance
index.



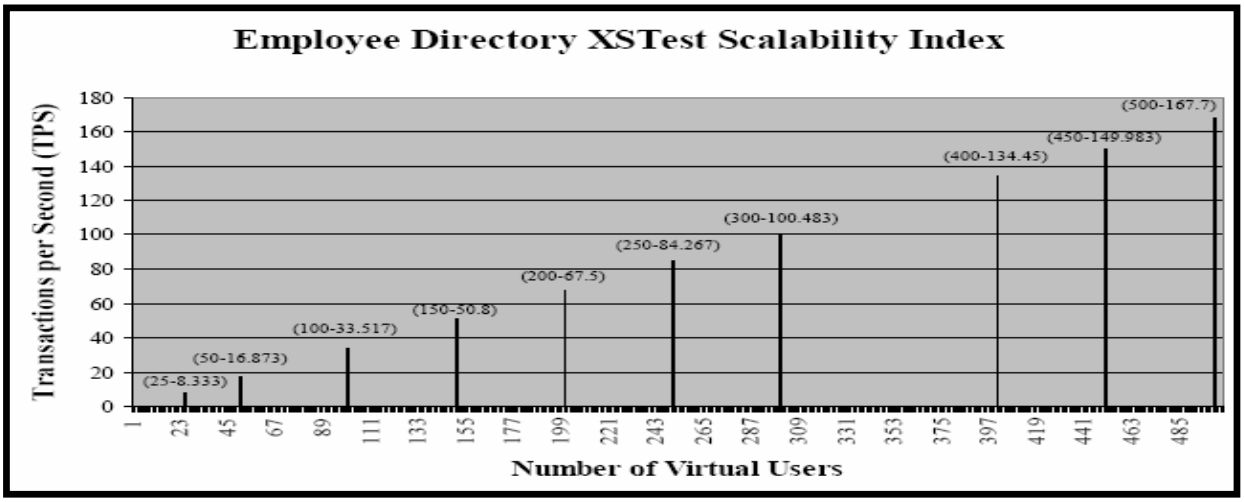
Number of Virtual Users	25	50	100	150	200	250	300	400	450	500
Transaction per Second (TPS)	8.333	16.733	33.333	50	66.667	83.35	100	133.333	149.567	166.683

Fig. 2 : OLC Assets System – XSTest Performance Index



Number of Virtual Users	25	50	100	150	200	250	300	400	450	500
Transaction per Second (TPS)	8.333	16.873	33.517	50.8	67.5	84.267	100.483	134.45	149.983	167.7

Fig. 3 : DCRR System – XSTest Performance Index



Number of Virtual Users	25	50	100	150	200	250	300	400	450	500
Transaction per Second (TPS)	8.333	16.667	33.333	50	66.8	84.517	100.017	133.33	149.833	168.15

Fig. 4 : Employee Directory System – XSTest Performance Index

V. Performance testing tools

The fact that performance testing helps in providing valuable information about the system's performance led to the development of number of performance testing tools. This section examines the strengths and weaknesses of three open source performance testing tools; viz., TestMaker, OpenSTA, and httperf.

a) TestMaker

It is a comprehensive testing framework that is used for performance, functionality, and load testing. A variety of test cases can be generated to perform thorough testing. TestMaker can support the Open Internet Protocols (HTTP, HTTPS, SOAP, XML-RPC, SMTP, POP3, IMAP, etc.). This software allows testers to create a number of virtual users and with the support from Test Network, testers can generate up to 10,000 or more concurrent test agents. TestMaker generates a clear output graph that represents the performance of the server based on the number of transactions per second versus the number of virtual users.

Apart from the strengths that are provided in the open source version, the commercial version offers some additional features such as the XSTest Pro that delivers performance index which helps to identify performance problems for capacity planning. Service Monitor System can send notifications via emails to notify, when a service fails or responds too slowly. Reports (charts and graphs) are generated to show the Performance index, result timing distribution, and performance timing by operation. This is an extremely useful feature since it is the quickest way to analyze the performance of the system. Another feature called Test Network, allows testers to generate a large amount of virtual users to test a specific web service.

On the contrary, TestMaker does have some weaknesses. Graphs are generated based on the number of transactions per second. In doing so, it assumes that all transactions are the same, even though some transactions may be long or short or complex. There is a possibility that a complex transaction may take shorter time to complete and this is not reflected on the graph. In addition, test cases should be performed more than one time to get the correct outputs.

b) OpenSTA

It is a completely free framework for testing the performance of Web Application Environments (WAEs). It is designed to create and run HTTP/S load tests in order to assess the performance of WAEs. OpenSTA provides a very simple GUI to create a new test, new collector, and a new script.

A new script is written in Script Control Language (SCL) and created by recording a list of activities that are carried out by the users. According to OpenSTA documentation [5], the process of creating collectors involves "deciding from which host computers or other devices performance data has to be collected and the type of data to collect".

After scripts and collectors are created, they are added into a test. Each test can be edited and controlled by using a number of features that are provided by the Graphical User Interface. These features include description about the test, start time, number of

iterations, host name, number of virtual users, duration of the test, delay between each iteration, etc. Test can be monitored during execution and the result is displayed in graph when the test completes. Graphs can be customized to improve the presentation of data.

Finally, Easy to follow documentation along with online help and commercial support are provided by the vendors. One of the weaknesses is that OpenSTA can only support HTTP and HTTPS protocols; it can only run on Windows OS. Another difficulty is that testers have to understand the SCL language.

c) httperf

It is a tool for measuring web server performance. A test is executed at the command line by specifying hostname, port, page address, rate, number of connections, and timeout. Timeout features help testers to define the number of seconds that each client is willing to wait for the response from the server. When a test completes, the result is generated in the form of plain text. Mosberger and Jin explained that the output consists of six groups [1]. These groups covers the overall results, results pertaining to the TCP connections; requests that were sent; results for the replies that were received; CPU and network utilization figures and a summary of the errors that occurred. A performance graph is generated by httperf to illustrate the server performance. httperf has some weaknesses such as it only supports the HTTP protocol and can only run on a Linux OS. In addition, Mosberger and Jin point out that the testers have to start httperf on each client machine collect and summarize each clientresult. They also suppose that a single command line that can control multiple clients could help in improving httperf.

Table 1 : Comparative Analysis of performance Testing Tools

FEATURES	TOOLS		
	TestMaker	OpenSTA	httperf
Unix OS	✓		✓
MacOS X	✓		
Windows OS	✓	✓	
Support Junit Test	✓		
Commercial Version	✓		
Open Source Version	✓	✓	
Extensible Library of Protocols (HTTP, HTTPS, SOAP, XML-RPC, SMTP, POP3, IMAP)	✓	Only HTTP & HTTPS	Only HTTP
Friendly and easy to use GUI	✓	✓	
Support J2EE and .NET	✓		
Support for Test Result Analysis (i.e. graphs and charts)	✓	✓	
Object Oriented	✓		✓
Maintainability and Support	✓	✓	
Agent Recorder	✓	✓	
Sample Test Agent	✓		
Run Test from Command Line	✓		✓
Support for Virtual Users	✓	✓	✓
P2P Support			

GPL Open Source License	✓	✓	
Script Control Language		✓	
Transaction per Second	✓		
Calculate the rate of data transfer (bytes/sec)		✓	
Timeout Management		✓	✓

VI. Conclusion

A distributed database is not stored in its entirety at a single physical location. Instead, it is spread across a network of computers that are geographically dispersed and connected via communications links. A key objective for a distributed system is that it looks like a centralized system to the user. The user should not need to know where a piece of data is stored physically. Testing the performance of such an environment is really a typical task. Existing web-application performance-testing tools offer a broad variety of functionality. However, none of them combines all the functionality we expected to use in our projects. In an ideal world, everything works perfectly when the test is run. In reality, first runs often show problems in server configuration or in application itself. An effort has been made to test the performance of DDB systems like OLC Assets, DCRR and Employee Directory using some open source performance testing tools like TestMaker, OpenSTA and httperf.

References Références Referencias

1. Mosberger, D. & Jin, T., httperf – A Tool for Measuring Web Server Performance.
2. Neuman, B. C., Scale in Distributed Systems. Readings in Distributed Computing Systems. IEEE Computer Society Press.
3. OLC Assets System – University of Otago Language Centre. (2005).
4. R. Mahammad Shafi, Dr. B. Kavitha – A Framework for Designing and Testing a Distributed Database System, Proceedings of AMCM-2011.
5. Tanenbaum, A. S., & Steen, M. V., Distributed Systems – Principles and Paradigms. America: Prentice-Hall.
6. Emmerich, W. (1997). Distributed System Principles. UK: University College London.
7. Yuanling Zhu and Kevin Lu, Performance analysis of Web Database systems, LNCS, Volume1873/2000, pp 805-814, 2000
8. Codd, E.F., "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377-387, 1970
9. Stonebraker, Michael with Moore, Dorothy. Object-Relational DBMSs: The Next Great Wave. Morgan Kaufmann Publishers, 1996.
10. Yao-S Bing, Alan R. Hevner, A Guide to Performance Evaluation of Database Systems, Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402,1984
11. Haran Boral, David J DeWitt, A methodology for database system performance evaluation, ACM SIGMOD Record, Volume 14, Issue 2, June 1984
12. Peter G. Harrison and Catalina M. Llado, Performance Evaluation of a Distributed Enterprise Data Mining System, Vol 1786/2000,pp 117-131, 2000
13. C. J. Date: "What is a Distributed Database System?" in Relational Database writings 1985-1989, reading, Mass:Addision-Wesley (1990).
14. Stefano Ceri and Giuseppe Pelagatti: Distributed Databases: Principles and Systems, New York, McGrawHill (1984).