International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:2, 2009

Performance Evaluation of an Online Text-Based Strategy Game

Nazleeni S. Haron, Mohd K. Zaime , Izzatdin A. Aziz and Mohd H. Hasan

***Abstract*—**Text-based game is supposed to be a low resource consumption application that delivers good performances when compared to graphical-intensive type of games. But, nowadays, some of the online text-based games are not offering performances that are acceptable to the users. Therefore, an online text-based game called Star_Quest has been developed in order to analyze its behavior under different performance measurements. Performance metrics such as throughput, scalability, response time and page loading time are captured to yield the performance of the game. The techniques in performing the load testing are also disclosed to exhibit the viability of our work. The comparative assessment between the results obtained and the accepted level of performances are conducted as to determine the performance level of the game. The study reveals that the developed game managed to meet all the performance objectives set forth.

***Keywords*—**Online text-based games, performance evaluation

I. Introduction

NOWADAYS, there are many online text-based games in the Internet. Most of these games are using the HTTP/HTML web approach. In these games, the performance is vital as good performance contributes to the motivation of the players to keep on playing with the game.

Even though these games are supposed to be lightweight in terms of performance since the interface did not involve heavy usage of graphic, but some of text-based games did not offer performances that are acceptable to the players. Most of the time, players will have to wait for a page that they requested. They will have to wait for a blank screen to be filled with data loaded from the server before they can interact with the page. This may not provide a good user experience as the players might feel bored waiting and thus demotivates them in pursuing the game.

One of the significant issues normally exist in an online-based game is due to the scalability problem. Some of the games cannot accommodate the increasing number of players and resulting in deterioration of the games' performances as the number of players increased. This problem should be alleviated as these games should be able to cater to at least a moderate amount of simultaneous players.

The response time that some of these games take to process and send back a reply to the user is a pertinent parameter that should be considered in designing any online game.

N. S. Haron, M. K. Zaime, I. A. Aziz and M. H. Hasan are with the Department of Computer and Information Sciences, Universiti Teknologi Petronas, 31750 Perak, Malaysia (phone: 605-3687486; fax: 605-3656180; e-mail: nazleeni@petronas.com.my).

The main reason why all these issues need to be taken seriously although it is only meant for a text-based game is because some of the players are using low bandwidth connection in playing the game. By fulfilling the accepted level of performance in designing and implementing the game, this type of player will enjoy an experience similar to the users with high bandwidth connection.

Based on the aforementioned points, therefore this paper presents the implementation details of our online text-based strategy game called Star_Quest. We present the result of the experiment on Star_Quest using four parameters such as throughput, scalability, response time and page loading time. A comparative assessment between the performance of Star_Quest and the standard of acceptable performance are conveyed in order to yield the performance level of Star_Quest.

II. Background of Study

A. Online Games

Online games are games that are integrated into a web browser and the user needs to load the web page in order to play the game. They are distinct from normal video and computer games in that they do not require any client side software to be installed. Online games are normally hosted on a web server that can handle a lot of traffic since usually many users will login at the same time. The database used to create online games must be able to interact quickly with server to provide users with good response time when users request something. Some examples of online web-based games are Utopia, Online Football Manager and Neopets.

B. Text-based Games

Text-based games are games that use text characters instead of bitmapped or vector graphics. Text games are typically easier to write and require less processing power than graphical games, and thus were more common from 1970 to 1990.

An online text-based game is a game that is played online using solely text-based interface. Users still use their mouse to navigate and select elements of the game that they want to interact with, but all the input and display will mostly be in text with the exception of light use of static graphic. Examples of online text-based games are Utopia, ezRPG and Earth: 2025.

C. Text Strategy Games

Text Strategy games are games that focus on game play requiring careful and skillful thinking and planning in order to achieve victory. In most strategy games, the player is given a full view of the game world, indirectly controlling the units under his command.

The origin of strategy games is related to their close counterpart, board games. Strategy games instantiated on computers generally take one of four archetypal forms, depending on whether the game is turn-based or real-time and whether the game's focus is upon military strategy or tactics. Examples of strategy games are Sid Meier's Civilization series, Gunbound, Command & Conquer series and Warhammer: Dark Omen.

III. Game's Description

The online text-based game that has been developed in this study as the test bed is called Star_Quest. Star_Quest is developed using PHP and HTML as its programming language with JavaScript and Cascading Style Sheet (CSS) inserted in the development stages. Star_Quest uses a database to store all of the games information and variables. This database is managed using MySQL and PHPMyAdmin.

The objective of the game is for players to manage a fleet of space ships that wanders within the game's universe and collecting resources while building up their army and going against other players.

To start playing, a player must first create an account that will require their basic profile. Once done, the player can login into the game using the created account.

After a player has login into the game, a page where all the available game listings for the player to choose will be loaded up and the player must select a game listed there to start playing.

Then, player will need to select their race or faction that they want to use in the game which is either the United Nations or the Blackstar faction. Players need also to specify the name that they will be used in the game.

After choosing a race or faction, player will come to the home page of the game. This page contains almost the entire summary of the player's condition including the number of available turns, amount of money available, list of the player's fleet status, list of enemy's fleet in the current Star System, resources available for mining in the current Star System and games related news.

From this page, players can take a quick look on the situation and condition of the current situation and then select the best course of action by going to other pages to continue their action to either start mining the resources, attack the enemy, warps to another Star System and other options.

IV. Performance Evaluation

Load testing is the process of testing the performance of the online game by putting the game under different load levels and measures its performance under different load scenarios. These load tests were conducted to mimic and emulate the players' behaviors and measure the performance of the game. The steps taken in performing the load testing are as follows [1]:

1) Identify the performance-critical scenarios.
2) Identify the workload profile for distributing the entire load among the key scenarios.
3) Identify the parameters that to be collected in order to verify them against the performance objectives.
4) Design tests to simulate the load.
5) Use tools to implement the load according to the designed tests, and capture the parameters.
6) Analyze the parameter data captured during the tests.

A. Performance-Critical Scenario

Battle scenarios that involve enemy attacking and choosing resources have been chosen to be evaluated because they have been identified as the resource-exhaustive activities in this game.

B. Parameter and Performance Objective

Four parameters which are throughput, scalability, response time and page loading time are chosen because they are the key performance parameters that are used when assessing any web application's performance. Performance objective for each parameter is determined based on literature survey and available standard for accepted performance level.

1) Throughput

Throughput is the measurement of how much requests per minute that a web application can handle [2]. This parameter shows whether the web application can handle heavy load in which multiple virtual users will make requests to the web application and the ability of the web application to handle the requests. Performance objective for this parameter is 50 requests per second.

2) Scalability

Scalability is a desirable property of a system, which indicates its ability to process more operations in a graceful manner as the number of users or requests increased [3]. In this study, scalability refers to the ability of the web application to handle a maximum number of users before the web application starts to generate errors to the clients. The performance objective for this parameter is Star_Quest can handle maximum number of 300 users simultaneously.

3) Response Time

Response time is the time a system or functional unit takes to react to a given input. The response time of a task or thread is defined as the time elapsed between the dispatch (time when task is ready to execute) to the time when it finishes its job (one dispatch)[2]. The same standards from forty years ago are still used to determine accepted level of response time [4]. According to R.B Miller standard [5], the accepted level of response time for any web application must be below than one second. Once second is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay [6]. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.

4) Page Loading Time

Page load time is the measurement of how long does the particular web page takes to be fully loaded into the browser. This is important as text-based games should give a quick response time due to the minimum use of graphic in the interface. This parameter is tested by recording the time taken to fully load a page of the web application into a web browser. This parameter's accepted level of performance is based on a study done by Microsoft [7] that stated that a web page should not take longer than three seconds to fully load into a browser as the longer time it takes, the more chances that users will get frustrated.

C. Test Design and Tools

The load test and the associated tools for measuring the performance of the four parameters are described as follows:

1) Throughput

Throughput is measured by simulating 100 virtual users accessing the Star_Quest site simultaneously without looping, with 0 seconds interval between users. This means that each initial user requests will be started simultaneously. Throughput is determined by calculating how many requests Star_Quest can handle in one second. This test is going to be performed three times with different size of loads. The tool used for measuring throughput is JMeter [7], an open source software specifically built for load testing.

2) Scalability

The scalability test will be started by simulating 300 of simultaneous users accessing Star_Quest. The number of users will be gradually increased from time to time. The test will be put to stop once the web server has reached its limit. The limit is identified through a metric called error rate which will be produced by the web server once it is no longer can afford to handle the requests.. JMeter [8] is also used as the tool to measure the scalability parameter of Star_Quest.

3) Response Time

For this parameter, the test consists of emulating 100 virtual users sending requests while playing Star_Quest and capturing the response time from the server. The OpenWebLoad [9] software will be used to monitor and capture the time taken for the web application to reply to the user. The time taken is considered as the response time from the server. A few trials will be conducted that will measure both average time and the maximum time it took for the web server to response. OpenWebLoad is chosen because it can provide near real-time performance measurements of the application under test.

4) Page Loading Time

This parameter is measured using the Load Time Analyzer [10]. The Load Time Analyzer is embedded into the browser and it will capture the total time taken to fully load the Star_Quest page into the browser. The test will be conducted three times and average of the time will be taken as final value.

V. Results and Analysis

This section presents the results obtained through the load testing conducted on the Star_Quest, our online text-based strategy game.

A. Throughput

Table 1 presents the results based on the throughput testing. The test was conducted three times with different load sizes. The average throughput of Star_Quest is 56.3 requests per seconds. The results indicated that the relationship between throughput and load is linear. The throughput is increased at the same rate of the load and this actually conforms to the theory of throughput and load relationship. This result also implies that no scalability or contention issues occur when 100 users accessing the Star_Quest site simultaneously.

TABLE I THROUGHPUT TEST RESULT

Throughput (100 Users)		
Test No	Load Size (bytes)	Request per seconds
1	53.98	55.4
2	56.1	57.6
3	54.37	55.8
Average	54.8	56.3

Fig. 1 illustrates the Star_Quest average throughput against the performance objective. In average Star_Quest performs better than the performance objective with 6.3 requests per second more.

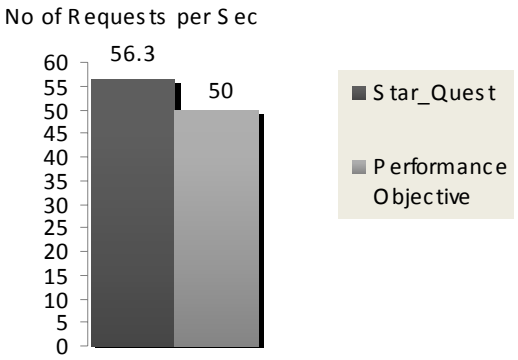

Fig. 1 Star_Quest average throughput

B. Scalability

Scalability is evaluated by gradually and constantly increasing the number of virtual users. The result of scalability test is shown in Table 2. From Table 2, it can be seen that Star_Quest web server only started to produce errors when the number of simultaneous users is 550 users. This means that Star_Quest is scalable enough to accommodate up to 550 users at the same time.

The performance objective for scalability has been determined to be 300 simultaneous users because for a text-based game it is a big number for any web application to handle.

Fig. 2 shows the comparison of the scalability test's result of Star_Quest against the acceptable performance defined earlier. It is clearly shown that Star_Quest manages to handle

almost twice the number of users as defined earlier as the acceptable level of performance.

TABLE II SCALABILITY TEST RESULT

Scalability	
Users	Error rate %
300	0
400	0
500	0
550	1.82
600	6.83
700	10.57

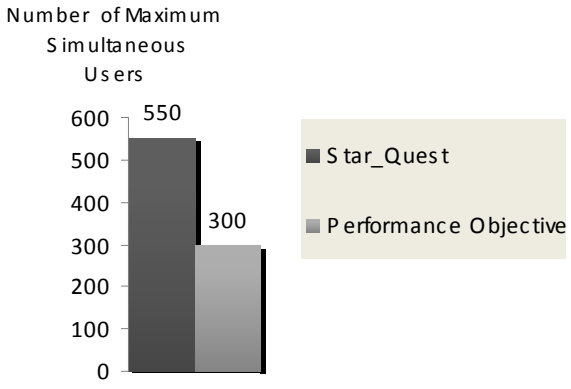

Fig. 2 Star_Quest Scalability Result

C. Response time

Table 3 presents the test result of the response time parameter. The test was conducted three times and the average response time produced by Star_Quest is 0.236 seconds. This result indicates that Star_Quest provides fairly immediate response. This is satisfactory for online text-based games because players cannot keep up with too fast response time.

TABLE III RESPONSE TIME TEST RESULT

Response Time (100 user)		
Test No	Average Time	Max Time
1	0.249	1.465
2	0.234	1.227
3	0.225	1.408
Average	0.236	1.367

Fig. 3 exhibits the comparison between Star_Quest average response time and the performance objective.

The performance objective was set based on [4]. It can be inferred that the Star_Quest has achieved the performance objective.

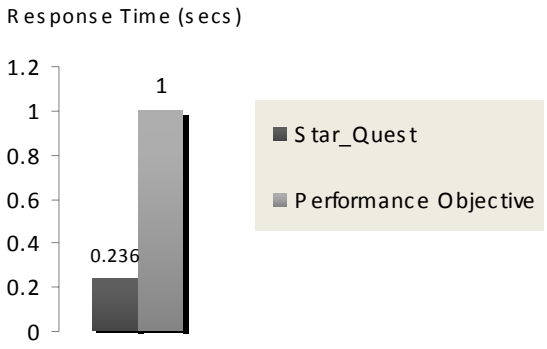

Fig. 3 Star_Quest Average Response Time

D. Page loading time

Table 4 shows the total time taken for a browser to fully load a page of Star_Quest. The average time taken to fully load a page is measured as 1154.3 milliseconds.

TABLE IV PAGE LOADING TIME TEST RESULT

Page Loading Time	
Test No	Time (ms)
1	1123
2	1170
3	1170
Average	1154.3

This parameter's accepted level of performance is based on a study done by Microsoft [7] that stated that a web page should not take longer than three seconds to fully load into a browser as the longer time it takes, the more chances that users will get frustrated. Fig.4 shows the comparison between the page loading time test's result and the accepted performance as defined by a research done by Microsoft [7]. Star_Quest's total page loading time is 1154.3 milliseconds, which is much lower than the 3000 milliseconds standards defined by [2]. This shows that Star_Quest's performance as an online text based game is good and as expected.

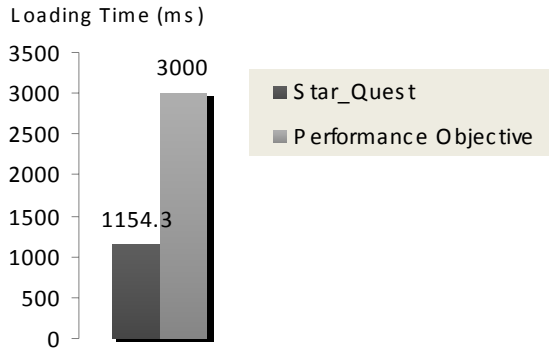

Fig. 4: Star_Quest Average Page Loading Time

VI. Conclusion

In this paper an experimental study on the performance of an online text-based game called Star_Quest has been presented. The performance of the game has been evaluated using four pertinent performance parameters such as throughput, scalability, response time and page loading time and using load testing techniques. The performance objectives for each parameter have been set forth based on literature survey and available performance standard. Based on the simulation analysis, it can be inferred that the game developed, Star_Quest has met and exceeded all performance objectives. This implies that Star_Quest can provide a satisfying user experience to the players.

However, more performance parameters can be analyzed further to obtain better performance review. Furthermore, the testing phase was mainly done using virtual users generated by the load testing tools. The result of this test could differ if the test subjects used are real users because we did not take into account such scenarios as the users abandoning sessions or constantly sending requests. It is predicted that if it were to be tested by real users, the performance results are more accurate and slightly lower.

References

[1] Load Testing Web Applications; Available : http://msdn2.microsoft.com/en-us/library/bb924372.aspx
[2] D. A. Menasce, "Load testing of web sites," *IEEE Trans. Internet Computing*, vol. 6(4), pp. 70-74, 2002.
[3] S. Ran, "A model for web services dsicoverry with QoS," *ACM SIGecom Ecxh.*,vol. 4(1), pp. 1-10, 2003.
[4] J. Nielsen, *Usability Engineering.*San Francisco: Morgan Kaufman, 1994, ch. 5.
[5] R. B. Miller, "Response time in man-computer conversational transactions" in *Proc. AFIPS Fall Joint Computer Conference*, Vol. 33, pp. 267-277, 1968.
[6] S. K Card, G. G. Robertson and J. D. Mackinlay, "The information visualizer: An information workspace". in *Proc. ACM CHI'91 Conference*, New Orleans, LA, 28 April-2 May,pp. 181-188, 1991.
[7] Microsoft, "Plan for Software Boundaries", Available: http://technet2.microsoft.com/WindowsServer/WSS/en/library/, Microsoft.
[8] JMeter; Available : http://jakarta.apache.org/jmeter/index.html.
[9] OpenWebLoad;Available : http://openwebload.sourceforge.net/.
[10] Load Time Analyzer; Available: https://addons.mozilla.org/en-US/firefox/addon/3371.