Capacity Planning for testing web-sites

Ajit Sarangi and Rakesh Agarwal
Infosys Technologies Limited, India
{ajitsarangi}{rakesh_a}@infy.com

Abstract: In the recent times it has been found that many web-based projects are unable to meet the deadline and this is because of their lack of understanding of capacity planning and its implications on system architecture and design. Capacity planning does not only tells us about the space required by each user on the disk (in a server) but it provides us with the correct methodology for network design, reliability and testing considerations. This paper discusses and gives practical examples of how capacity planning is performed for Web site's server and network hardware.

1. Introduction

Maximizing output and minimizing input is the goal of Capacity Planning. *Capacity planning[1][2] is the process of measuring a Web site's ability to serve content to its visitors at an acceptable speed.* In other words capacity planning is the process of determining the future network resource requirements.
Testing a web-site which may in seconds encounter million of hits is a very difficult and tedious method. Capacity planning[3][2] helps in testing by measuring the number of hits to the site which in turn gives the computing resources (CPU, disk space, RAM, and network bandwidth).

Testing a web site driven by capacity planning is determined by three factors[2][4]:

1. Number of hits per unit time. The popularity of the site increases the number of accesses made to the site, which in turn leads to performance degradation.
2. Hardware and software configuration. The servers having a better hardware and software configuration will increase the site's capacity.
3. Amount of content. More the contents of a web site (text, picture, graphic, etc) will result in the servers having to do more work per user, thereby lowering the performance of the site.

Capacity planning is important so that web-based application give a better response to the user. Sites that do not have sufficient capacity to serve all of their users will frustrate them with slow response times, timeouts and errors, and broken links[1][4]. There are three factors that provide full potential to a site: Service quality, Content quality and the speed at which the content and service are accessed.

Capacity planning[3][4] allows us to make sure that the site delivers quality content to users at a rapid speed. If capacity planning is measured[5] incorrectly, or no action is taken based on the recommendations from the capacity planning reports, then users may choose to go elsewhere to find better service, quality, and speed[6][7].

The process capacity planning which will provide better testing results can be categorized into five major areas[2][3][6][7]:

- Understanding the existing network configuration
- Workload Characterization
- Data sets of different scenarios
- Required deliverables
- Statistical analysis and reporting

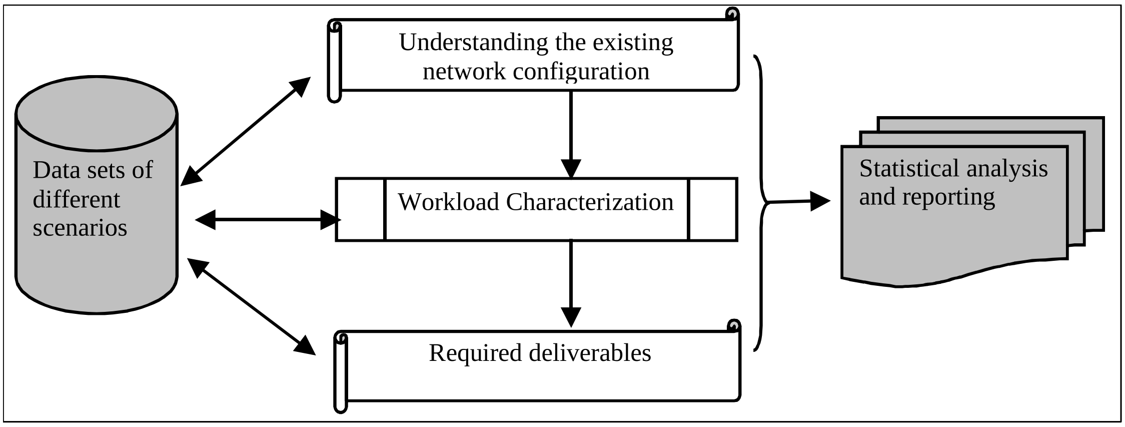
Understanding the existing network configuration
Data sets of different scenarios
Workload Characterization
Statistical analysis and reporting
Required deliverables

Figure 1: Stages in Capacity Planning

Figure 1 shows the different stages in capacity planning. **Understanding the existing network configuration** will provide information about the present network configuration and resource requirements. This analysis describes the current performance of the network. **Workload characterization** is the process by which the capacity planner will identify the resources that are affecting the required workload. It identifies what users do on the network on average. This will give an analysis of the computing resources required to do those jobs. The **Data sets of different scenarios** is the store of the collected measurement information. There should be a provision of test cases and the results delivered..

The **Required deliverables** are the service level agreements that need to be provided to the end user in context of accuracy, timeliness and reliability. These are often determined and driven by the business objectives identified. The **Statistical analysis and reporting** of capacity planning deals with the analysis of data to project future capacity requirements and management must have timely, accurate information. Various modeling and simulation tools may be employed to perform these tasks.

This paper gives an outline of how to do capacity planning of a particular system to achieve better results in testing. A method is given which is based on a real life study for a project.

2. Determinants of performance

The essence of performance diagnosis is to determine which factor is creating bottleneck in the system. The corrective solution depends on proper diagnosis

Following are the major key factors that need to be considered before the application's capacity and performance is evaluated. These factors also play a key role in evaluating the scalability of the application.

1) Hardware (Includes the network)
2) Number of transaction
3) Number of concurrent user
4) Number of graphics Image
5) Complexity of the Architecture
6) System Up time

After considering the above factors the goal of the application can be fixed and various methods, as described in the following sections, can be used to achieve the set goal. Thus, the sequences of steps followed are:

- Determine and set the objective.
- Adopt any one of the methods described below to evaluate the architecture decided.
- Fine tune and optimize the architecture to bench mark the results.
- Extrapolate the benchmarked results to arrive at the final capacity of the application.

In general the factors that affect the scalability of the server and long-term strategy are tabulated below. These factors also affect the cost per transaction.

Determinant	Current Factor	Increase in 3 Yrs.
Number of Transaction	X	30%
Per cost transaction	Y	10%
Number of concurrent user	z	10%

From the metric shown in
Figure 2, we can infer the following:

1) With low cost, and without optimization, the performance will be low.
2) With high cost and unplanned implementation, the performance still remains low.
3) With high cost and planned implementation of architecture, performance can be increased easily.
4) With proper planning and capacity optimization, desirable high performance can be achieved at much lower cost.

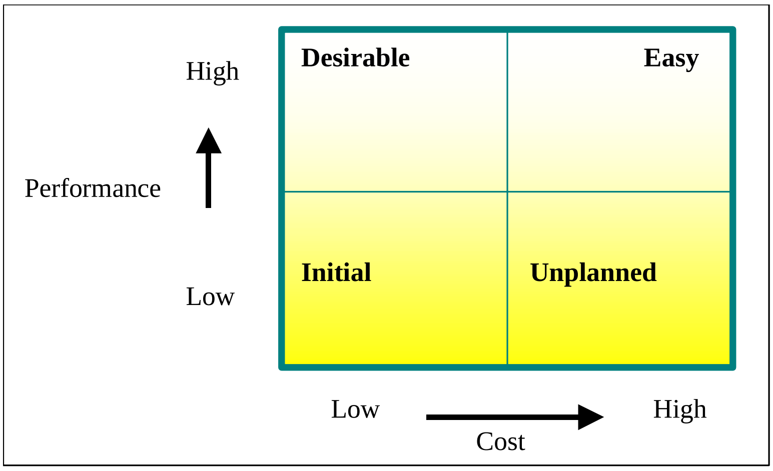

Figure 2: Cost performance matrix

3. Performance diagnosis in practice

Based on the above metric parameter a sophisticated method is needed in actual practice. We will improve performance diagnostic system by making it a more quantifiable technique.

Theory of effective Capacity planning – cost optimization

The Total cost can be defined as

$T_C = \sum_i C_i + F$ $T_{C =}$ Total Cost

F = Fixed cost of Installation

C_i = Cost of configuration for a specific services

i = Number of services in the web server (varies from 1 to number of services)

$$\mathbf{C_{i\,=}\ \sum f\ [j]}$$

where f [j] is the function of j, represents the optimized cost at point marked as X in the graph for each component

j = number of resources like CPU, RAM, Network, Cache

4. Methods

Capacity planning methods can be categorized as follows:

- Primitive Approach
- MBO approach
- Cost Approach

4.1. Primitive Approach

The Primitive method is outlined in Figure 3.
Since this method is followed at a very early stage, based on previous experiences, the hardware is chosen. After an initial selection of the hardware and development of

software, the web site is subjected to the load test using a web load-testing tool. The results obtained by load testing are analyzed and the bottlenecks in the code are identified and resolved.

After fine tuning the code for bottlenecks, the web load test is performed again and the results obtained are benchmarked for the hardware configuration and the number of concurrent users. Based on these benchmark results, the results can be extrapolated for changed hardware configuration and users and the capacity and performance of the web site analyzed.

This can be done at system testing stage of the Development Life cycle.

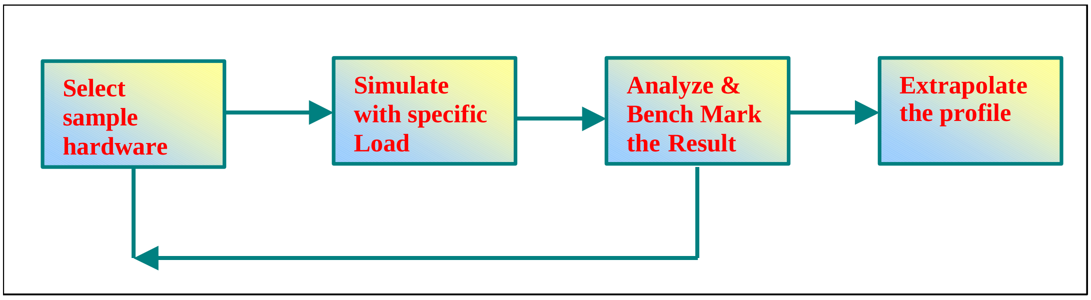

Figure 3: Primitive approach

4.2. MBO Approach: (Manage by Objective)

In the MBO approach the objective is fixed before hand and the tests are performed to determine the optimum hardware requirement. The target load is identified and by iterative methods, the target hardware is determined such that there is optimization of resources and cost.

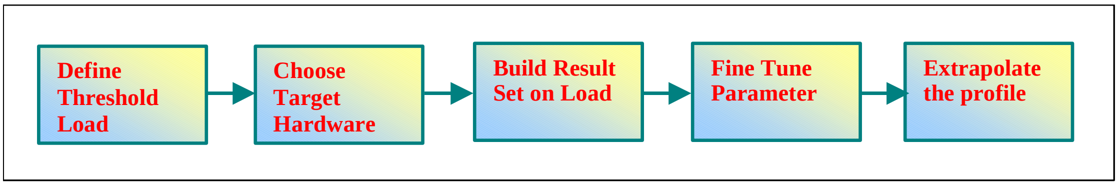

Figure 4: MBO approach

After determining the target hardware and fixing the load, web load testing tool is used to analyze the performance and cost of the system and to determine the bottlenecks, if any. The bottlenecks are resolved by fine tuning the code or altering the hardware configuration and the round of testing is done again to build a bench mark result set. This result can be extrapolated to obtain the profile for changed load and hardware configuration such that there is an optimization in the cost and performance of the system, which is developed.

This method can be used at beta testing level.

4.3. Cost Approach: (optimization technique)

In the Cost Approach method, the hardware is selected and this is subjected to load expected. The results obtained by this are analyzed and benchmarked. The graphs obtained from the results of load testing are analyzed and the optimum value for TPS and concurrent users is obtained from the graphs, as described above.

Using the equation, the cost of transaction can be analyzed and based on the cost, the capacity of the application obtained.

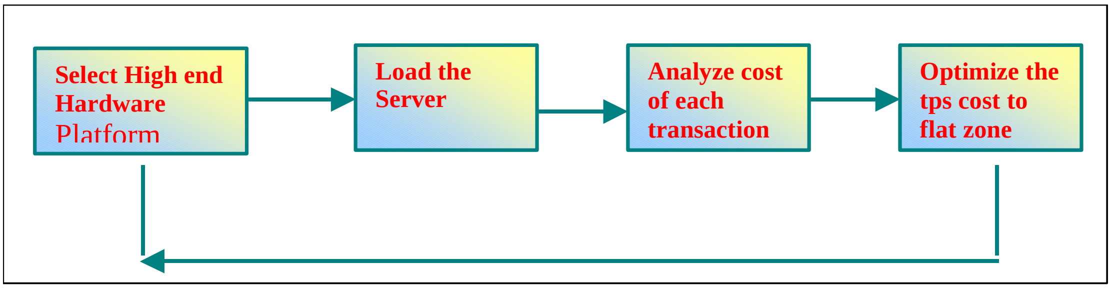
Select High end Hardware Platform
Load the Server
Analyze cost of each transaction
Optimize the tps cost to flat zone

Figure 5: Cost approach

The Cost Approach method enjoys the following merits:

1. Pre-testing of alternative plan
2. Short testing duration
3. Hybrid process of primitive to MBO
4. Defined rational throughput

5. Generic Components of a Testing tool

The tool monitors the response time of an Internet or Intranet Web server for the request of the client and stores with configured setting. Tools simulate the activity of a Web server and its many client Web browsers communicating across one or more networks. Primary subsystems for the tools:

- a server,
- a client,
- a controller and the network.

A typical configuration

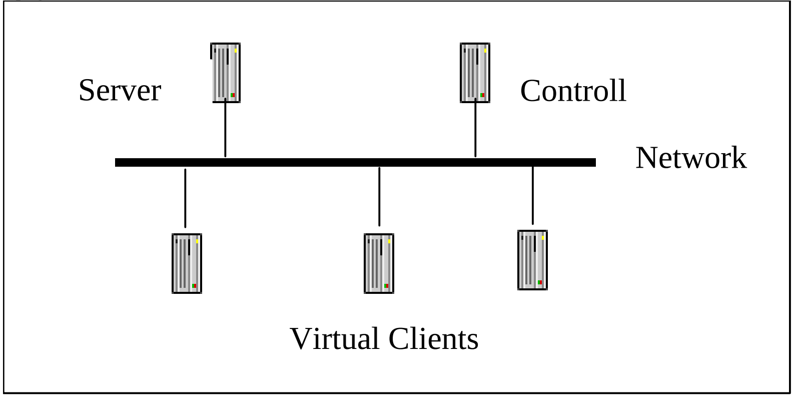

Figure 6: The hardware for a single-network load test

For the case study, we have taken a *server* configured with Internet Information Services in Windows NT 4.0, SQL-7 Server. The Controller is in-build in the tool (Webload).

For number of clients is simulated through the tool. The limit will reach for number of clients when it uses the physical memory. The number of concurrent user will be

Sequence of test in tool:

- Responds to requests for connections.
- Establishes, manages, and terminates connections.
- Receives requests for Web content.
- Processes client requests and sends responses.

6. Some of the load testing tools available

- **Webload 4**: A comprehensive testing and analysis solution to combine scalability, performance and integrity as a single process for testing of Web applications. More information about this tool can be found at http://www.radview.com/products/index.asp
- **LoadRunner**: A load testing tool that predicts system behavior and performance and exercises an entire enterprise infrastructure by emulating thousands of users to identify and isolate problems. It supports multiple environments.
- **Astra Load Test**: An interactive automated testing tool for transactional Web applications. It offers the easiest way to test the scalability and performance of Web applications. It emulates the traffic of hundreds or thousands of real users to identify and isolate bottlenecks, optimize performance and overall user experience.
- **WinRunner**: An integrated and functional testing tool for entire enterprise. It captures, verifies and replays user interactions automatically. This helps to identify defects and ensure that business processes, which span across multiple applications and databases. More information about this tool can be found at http://www-heva.mercuryinteractive.com/products/winrunner/
- **CYRANO WEBTESTER**: It is an end-to-end quality assurance provider to its customers, helping them maximize their IT investments and ensure uninterrupted e-business. The tool includes: functional testing, load testing and regression testing. More information about this tool can be found at www.cyrano.com/products/webtester.html
- **e-Load**: This gives the fastest and accurate way to test the scalability of Web applications. Companies use it regularly to accurately assess the load capacity of their applications. More information about this tool can be found at http://www.empirix.com/empirix/web+test+monitoring/products/

7. CONCLUSION

Capacity planning is an important aspect of developing good quality websites; proper planning ensures a healthy network that can grow to meet future needs[3]. Many companies depend on web-based applications to simplify business processes so that applications can run efficiently on the networks.

The Primitive Approach or the MBO approach or the Cost Approach helps in predicting scalability, reliability and performance issues of an e-business application. This is essential for understanding an application's limitations. Load testing tools tests system behavior under real-time conditions and converts this data into easy-to-use graphs and reports. With this information, one can assess the capacity of the application, maximize its performance, identify cost of transactions and arrive at an optimization between software, hardware architecture and web traffic.

8. Disclaimer

The authors of this report gratefully acknowledge Infosys for their encouragement in the development of this research. The information contained in this document represents the views of the author(s) and the company is not liable to any party for any direct/indirect consequential damages.

9. References

[1] MercuryInertactive.com

[2] Web Performance Tuning, Patrick Killelea, First Edition, October 1998, ISBN: 1-56592-379-0, 374 pages

[3] White paper on Effective Capacity Planning for the Enterprise Network http://www.concord.com

[4] Capacity Planning - Microsoft Enterprise Services White Paper E-Commerce Technical Readiness, Authors: Louis de Klerk (Inobits Consulting Pty., Ltd.), Jason Bender (MSNBC) www.microsoft.com/technet/ecommerce.

[5] G. Bruno and Rakesh Agarwal. Modeling the Enterprise Engineering Environment, in IEEE Transactions on Engineering Management, volume 44(1), pages 2-30, February 1997.

[6] M.E. Crovella and A. Bestavros, "Explaining World-Wide Web Self-Similarity," technical report, Dept. of Computer Science, Boston Univ., Oct. 1995.

[7] B.M. Duska, D. Marwood, and M.J. Feeley, "The Measured Access Characteristics of World-Wide Web Client Proxy Caches," Proc. Usenix Symp Internetworking Technologies and Systems, 1997.

[8] F. Douglis, A. Feldmann, and B. Krishnamurthy, "Rate of Change and Other Metrics: A Live Study of World Wide Web," Proc. Usenix Symp. Internetworking Technologies and Systems, 1997.