PREDICTION OF PERFORMANCE FOR WEB APPLICATION BY PERFORMANCE MODELING WITH JMETER

Sheetal S.Patil, Prof. S.D.Joshi, S.S.Dhotre

Abstract— Performance analysis and modeling of web application [1] allows users to record many different parameters relevant to web application's performance. Our goal is the identification of the most influential parameters which determine the factors improving performance of web-application with the help of performance modeling. We create performance model with respect to stages of SDLC as Performance problems are frequently introduced early in the design. We first identify application scenarios and performance objectives. Performance objectives can be measured in terms of response time, throughput (how much work in how much time), and resource utilization (CPU, memory, disk I/O, and network I/O).We apply performance scenarios into steps and assign performance budgets. These budgets are used to set constraints across performance objectives and this will help to predict the performance of web application.

Index Terms— Performance modeling, Performance objectives, Performance budgets, Webhelix..

———————————— ◆ ————————————

1 INTRODUCTION

As the importance of the World Wide Web has become ubiquitous, and it continues to grow unabated at exponential rate. Web-based systems and applications now deliver a complex array of varied content and functionality to a large number of heterogeneous users. The interaction between a Web system and its backend information systems has also become more tight and complex. As we now increasingly depend on Web-based systems and applications, their performance, reliability and quality have become great importance, and the expectations of and demands placed on Web applications have increased significantly over the years. As a result, the design, development, deployment and maintenance of Web-based systems have become more complex and difficult to manage. Many of the web applications are of following types: Informational Interactive, Transactional Workflow oriented, and Collaborative work Environments, Online communities' marketplaces. The scope and complexity of Web applications vary widely: from small scale, short-lived (a few weeks) applications to large-scale enterprise applications distributed across the Internet, as well as via corporate intranets and extranets. Web applications now offer vastly varied functionality and have different characteristics and requirements. Some of today's Web application development processes have been derived from well-known Web content tools. Most of today's Web application development processes are extensions of standard software engineering processes. The waterfall model process was perfect for developing a file maintenance program for mainframes, but far too restrictive a process for building a Web application. Similar to software development, Web application development is also an iterative process [5] and one such iterative process is the spiral [6] approach; but the exact steps at each cycle of the spiral are debated, as is the metric to be used to determine the completion of a cycle, while the iterated waterfall model, another iterative process, is considered too rigid an approach for developing Web applications [7].

WebHelix uses a modified spiral approach to systems development. Some parts of the design process, like the business analysis and planning, are done once at the beginning of the process. Some parts of the process, like deployment and maintenance, are done once at the end of the process. The major creation of the Web application (coding) is done by using an identical set of steps repeatedly, producing a set of more and more complete prototypes, with the final prototype being the completed Web application. A detailed description of the WebHelix process is:

- PreHelix 1 Business Success Evaluation
- PreHelix 2: Planning
- Helix 1: Analysis
- Helix 2: Design
- Helix 3: Coding and Integration
- Helix 4: Testing
- Helix 5: Evaluation
- PostHelix 1: Deployment and Training
- PostHelix 2: Maintenance and Future Updates

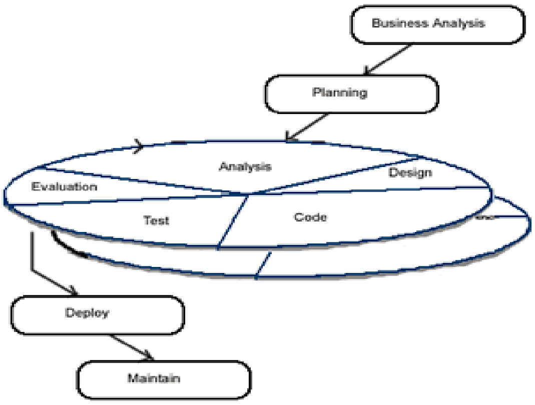

Fig.1 WebHelix as a modified spiral process

All the software projects as well as web applications follow Software Development Life Cycle (SDLC) throughout their development .As users expect web applications to be quick and responsive, attention to system's performance is paid from an early software development stage and followed throughout the SDLC by means of a well defined systematic process: Software Performance Engineering (SPE) [9]. As part of the SPE process, measurements and performance modeling can be used to study a system. To directly assess if an application will meet the required performance objectives based on available resources, for the purposes of capacity planning [10], a measurement-based approach is adopted. The behavior of the system under given customer workload can provide results which can help identify performance bottlenecks [11]. Performance modeling, which uses performance models, also finds use in capacity planning by means of predicting system's performance and by pinpointing system bottlenecks. Other uses of modeling include capacity provisioning i.e. allocating and preparing resources to handle the demands and finding application configuration parameters that meet the desired objectives.

2. PERFORMANCE MODELLING:

A performance model provides a path to discover what we do not know. The benefits of performance modeling include the following:

- We can achieve performance in every step of development process.
- We evaluate our tradeoffs earlier in the life cycle based on measurements.
- Test cases show us whether we are trending toward or away from the performance objectives throughout application life cycle.

Modeling allows us to evaluate design before investing time and resources to implement a flawed design. Having the processing steps for our performance scenarios laid out enables us to understand the nature of our application's work. By knowing the nature of this work and

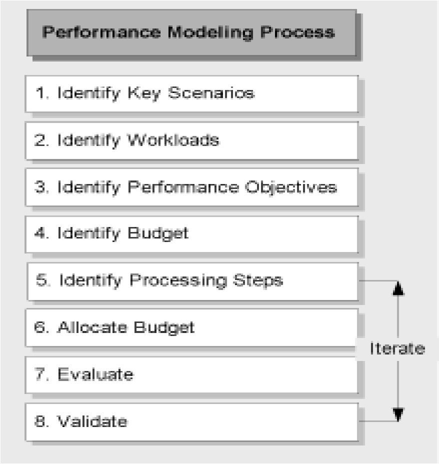
Performance Modeling Process
1. Identify Key Scenarios
2. Identify Workloads
3. Identify Performance Objectives
4. Identify Budget
5. Identify Processing Steps
6. Allocate Budget
7. Evaluate
8. Validate
Iterate

Fig2.Eight step performance model[4]

The constraints affecting that work, we can make more informed decisions. A number of inputs are required for the performance modeling process. These include initial (maybe even tentative) information about the following:

- Scenarios and design documentation about critical and significant use cases.
- Application design and target infrastructure and any constraints imposed by the infrastructure.
- QoS requirements and infrastructure constraints, including service level agreements (SLAs).
- Workload requirements derived from marketing data on prospective customers.

  The output from performance modeling is the following:

- **A performance model document**.
- **Test cases with goals**.

The performance model document may contain the following:

- Performance objectives.
- Budgets.
- Workloads.
- Itemized scenarios with goals.
- Test cases with goals.

An itemized scenario is a scenario that you have broken down into processing steps. For example, an order scenario might include authentication, order input validation, business rules validation, and orders being committed to the database. The itemized scenarios include assigned budgets and performance objectives for each step in the scenario.

Test Cases with Goals: We use test cases to generate performance metrics. They validate our application against performance objectives. Test cases help us determine whether we are trending toward or away from our performance objectives**.**

3 PERFORMANCE TESTING AND PERFORMANCE METRICS:

Performance testing is all about numbers and metrics. Jmeter is free open source, performance measurement tool written in java [12]. Thread group is basic element of a jmeter test plan. Each thread represents user. Samplers do the actual work and interact with the server loading. There are a number Samplers in JMeter by default. For web testing, we use 'HTTP Request'. The information produced by Samplers is consumed by Listeners Popular listeners are: Graph Results,View Results Tree,Simple Data[3] Writer. Logic Controllers determine the order that Samplers are processed More complex scripts will make use of Loops, Interleave, Random, If. A test isn't a test unless it checks something. Assertions are what do the checking for web applications, these are the useful ones Response, Duration, HTML Assertion. Configuration Elements set default values for other parts of the Test Plan as well as configure variables:

- CSV Data Set Config allows you do to DDT within JMeter
- HTTP Cookie Manager will automatically intercept and sent cookies with requests
- HTTP Request Defaults saves you time when doing lots of HTTP Samplers

We will get different values based on the jmeter test tool like Throughput (responses/minute), Average (ms), and Median (ms). JMeter is a flexible tool that not only allows you to test the HTTP servers but also to load-test Web services. Web service and SOAP samplers are new features of JMeter; hopefully they will evolve as Web services get a wider acceptance in industry and among developers. Using the existing functionality of JMeter and its provided user interface we were able to simulate a load of 5 concurrent threads hitting the server with 10 and 5 ms delays.

4. PROPOSED SYSTEM:

In the proposed system the performance metrics are calculated using Performance Testing Tool Jmeter.[12] Web application built using php as well as dot net technology is used for performance testing. We implement performance model for web application which consider performance in each step of web process model (web helix) .Here we try to integrate performance model with process model for web application. As Performance is generally ignored until there is a problem. This will help application developer to consider performance in early phases of web application design and continues throughout the application life cycle. With performance testing using case tool Jmeter we will calculate throughput, Rate of incoming visitors, No of users at particular peak time which will help in design phase of SDLC.This performance model helps developer of web application for consideration of performance in each phase of SDLC.

CONCLUSION:

This paper, we present the overview of a performance model applied for web application throughout it's SDLC. We have considered here WebHelix process integrated with performance model. Again we have considered the challenging task of load testing Web sites in systematic and repeatable manner with Jmeter. Scientific load testing allows finding out the metrics related to throughput, response time ,Concurrent users,error rates.

This gives us realistic, useful and reliable data about a Web site's capacity and scalability. Poor performance of Web-based systems can adversely impact the profitability of enterprises that rely on them. As a result, effective performance modeling integrated with process modeling essential for understanding whether a Web-based system will meet its performance objectives when deployed in the real world.

References:

[1] Kredel, H. Kruse, H. ; Ott, I. **Performance Analysis and Performance Modeling of Web-Applications** P2P,Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2011 International Conference on
26-28 Oct. 2011.

[2] Bo Li ,Haiying Zhou ; Decheng Zuo ; Zhan Zhang ; Peng Zhou ; Long Jia , **Performance modeling and benchmarking of bank intermediary business on high-performance fault-tolerant computers,** Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on**27-30 June 2011.**

[3]] http://www.apache.org.in

[4]J.D. Meier, Srinath Vasireddy, Ashish Babbar, Rico Mariani, and Alex Mackman
"Improving .NET Application Performance and Scalability ", Microsoft Corporation May 2004.

[5]Schach, S., (2005). ***Object-oriented and classical software engineering*****. McGraw Hill.**

[6]Boehm, B., & Hansen, W. J. (2001). ***The spiral model as a tool for evolutionary acquisition.*** **Retrieved on September 27, 2007,**
http://www.stsc.hill.af.mil/crosstalk/2001/05/boehm.html

[7]Pressman, R. S. (2005). ***Software engineering:A practitioner's perspective*****. New York:McGraw-Hill.**

[8] ***George Whitson*** **, WEBHELIX: ANOTHER WEB ENGINEERING PROCESS JCSC 21, 5 (May 2006)**

[9]C.U. Smith and L.G. Williams, ***Performance Solutions: A Practical Guide to Creating Responsive,***
Scalable Software**, Addison-Wesley, 2002.**

[10]D. Peng, Y. Yuan, K. Yue, X. Wang and A. Zhou, "Capacity planning for composite web services using queueing network-based models," in ***Advances in Web-Age Information Management*****, Q. Li, G. Wang and L. Feng, Eds. Springer Berlin / Heidelberg, 2004, pp. 439-448.**

[11] Daniel A. Menasce, "Load Testing of Web Sites," ***IEEE Internet Computing*****, vol. 6, no. 4, pp. 70-74,**
July 2002.

[12] ***Sheetal S.Patil*** **,** ***Prof. S.D.Joshi*** **"Identification of Performance Improving Factors for Web Application by Performance Testing" in International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459, Volume 2, Issue 8, August 2012,pp 433-436**

[13] Naveenkumar, M.P.Bhor, Dr. S.D.JOSHI, SOFTWARE QUALITY AND DEFECT TRACKING SYSTEM, IJERIA