JENNIFER
JENNIFERSOFT, Inc.

Nov. 2011

Performance Analysis Model

JENNIFERSOFT, Inc.

Table of Contents

Executive Summary

Executive Summary

Challenge

Keeping the IT infrastructure up-to-date with latest information systems technology is a constant necessity for survival of enterprises in today's highly competitive market place. However, despite the efforts to comply with the advancement in technology, still many companies are frequently exposed to significant performance degradation which impact their business directly.

Opportunity

By understanding the performance theory/model of information systems, we can understand the fundamental principles which affect an information system and use this knowledge to establish proper plans for application lifecycle and application performance management, to be activities such as benchmarking, performance testing and capacity planning.

Benefits

This white paper provides introduction to the performance theory/model for a web based system along with factual example and scenario on how they may be applied in performance analysis and management.

Enterprises are migrating to next-generation information system in order to stay ahead in very competitive business environment. These high tech information systems provide competitive edge to companies by providing ability to process large amount of data in timely manner and respond requested information to customers within a reasonable time.

"Our marketing system must be able to handle 100,000 short-messages per hour to our customers during the campaign period.

"Our bank transaction system has to process 7000 transactions per second at the end of every fiscal month with maximum response time of 3 seconds."

In order to meet the proposed business requirements, companies perform variety of tasks. Hardware and software optimized for proposed business requirement determined by using "Benchmarking" during the early phase of project. Once project enters in to development phase, quality assurance management is planned through variety of testing and performance tuning in order to build system which meets the business requirement.

However, despite the enormous effort and invested capital to assure quality of the system, it is inevitable that companies will be faced with number of performance problems from the day system is opened.

"So what is causing this phenomenon?" No enough project budget? Not enough testing from lack of project time? Disparity of direction between experts collaborating on the project? Poor communication between project team and/or just simply common human errors?

Section 1

Performance Model

When observing the fundamental principle describing different phenomenon occurring in the real world, establishing and applying a mathematical model is commonly trying to understand it scientifically. By finding variables which can best represent a certain phenomenon's characteristics and mathematically reorganize the relative relationship of between these characteristics, one can create a virtual environment to perform experiments and use logical reasoning to draw meaningful conclusions or sometimes even come to understand the phenomenon. The result from such experiment can then be applied to the real-world scenario, simplifying what seemed like a complex problem before and even allowing prediction the cause-and-effect when one factor is changed.

Same principle may apply to the field of IT. To understand the fundamental principles of an information system, performance analysis activities, benchmarking, performance tests, capacity plans and etc..., performance model based on mathematical principles must be established. From IBM Mainframe, once popular in the '80's, to the CS system/TP-MONITOR in the '90's, the

"Queuing Theory" has been the most popular mathematical model applied. This theory take account a complex information system with queuing network, single queue or combination of multiple queues, and mathematically calculates the correlations among various performance variables in order to predict the outcome of system performance.

However, the Queuing Theory has the following limitations which prevent it from being applied to a web based system, which its use became popularized with explosive growth in use of internet.

Firstly, recent web based systems are made of very complex structures based on web application servers. The Mainframes or TP-MONITOR in the past were composed of simple architectures such as the middleware and the DBMS only, but recent web based systems are comprised of complex architectures resulting from distributed transaction processing, service oriented architecture and so on. Thus there is greater difficulty in establishing a performance model.

Secondly, due to the characteristics of HTTP protocol used by web based systems, it is very difficult to measure performance model variables, especially, number of concurrent users. In case of the main frame or TP-MONITOR in the past, the number of concurrent users was calculated by measuring the number of TCP connections between the client and the server. However, in a web based system, since the existing TCP connection is cut off after processing a business request, the same method cannot be used.

Thirdly, a web based system typically provides services to a large number of unspecific users, so it is very difficult to predict the inputs values for variables. Even if input variables were set based on large sample of data from collected over extended duration of time, these values are greatly influenced by external factors such as marketing activities, thus the reliability of data is not very high.

Thus, in order correctly understand the performance phenomenon occurring in a web based system, understanding of the performance theory/models highlighted for the web application server env

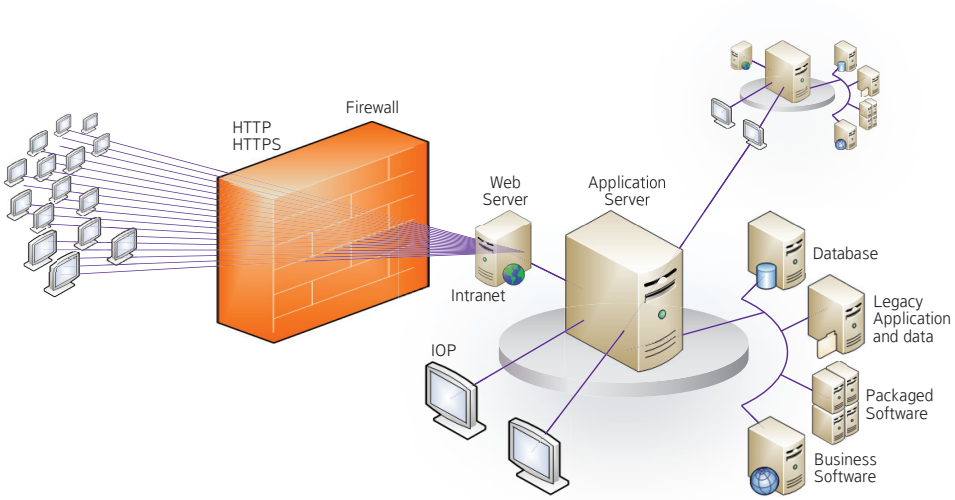
1.1 What is performance

Information system performance is a term very familiar to a person working in the IT industry. However, there are not many people who understand the meaning of it correctly. "Performance" is a very frequently used term but it is often used without correct scientific definitions, which may lead to lot of confusions.

To understand the correct definition of "performance", let take a simple example of something that we see everyday.

"When you go to a supermarket during Christmas holidays, you may spend lots of time standing in line waiting to be serviced. Since there are more customers than during normal days, the number of employees servicing the customer is relatively less and assuming that customer purchase more items on holidays, it take much longer time to service each customer"

Measuring performance of Information system is very similar to this example in terms of characteristics. As the performance of employees is measured by how quickly they service each customer, the information system performance depends on how quickly its own services can be provided to its users.

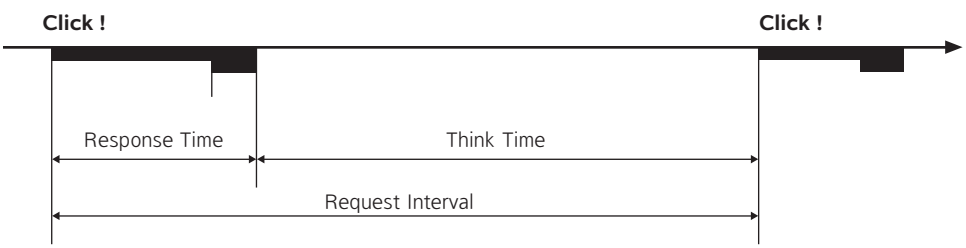


Thus, the information system performance can be viewed in the following three perspectives.

- How many Users?: How many users can be provided with the services?
- Resonable response time: Does it provide reliable service within the adaquote response time?
- Cost-effective: If possible, the cost-effectiveness should be high.

1.2 Performance Related Terminologies

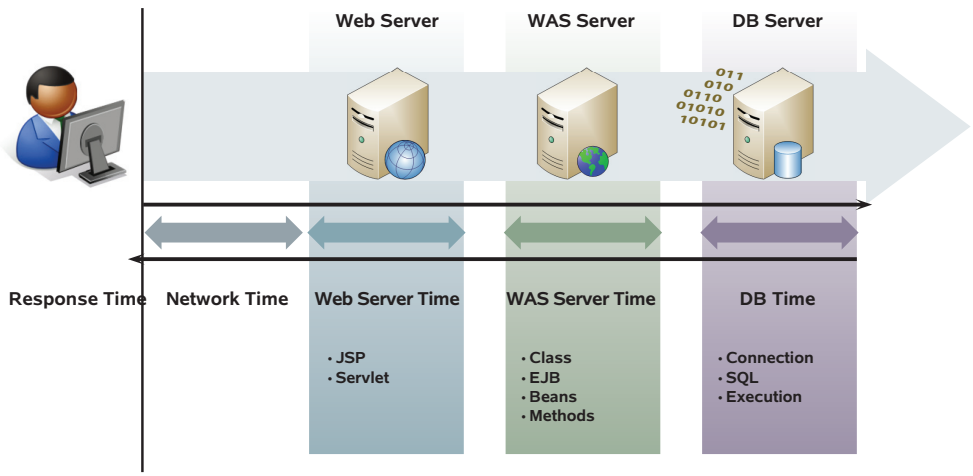
Before introducing you to the main parts of performance theory/modeling, let's learn various terms used in performance analysis.



1.2.1 Response Time

Response time measures reasonable time spent for completing a service. It is defined as the total time taken for user's service request to be processed and result is provided. In case of web based systems, there response time can often be broken down to client time, network time, web server time, DB time and others as shown in belo

w diagram. The detailed analysis of response time can be very useful analyzing for cause of bottleneck which is causing performance problems.



1.2.2 Think Time

Think time is defined as time spent by user for preparing next service request. It consists of time spent to view the results for previous request and/or time spent to enter a value on the PC screen to send the next request. Here, the think time can vary depending on each user's unique service needs, speediness in reading and entering in value and so on, so a strategic plan for measure the value of user think time is recommended for each system. If it is impossible to make measurements in the actual production system, you can use the values used in the previous projects for each business domain as suggested in the below table.

Moreover, the solution vendors such as SAP and Oracle provide the recommended think time that can be applied to benchmarking and performance tests for their solutions.

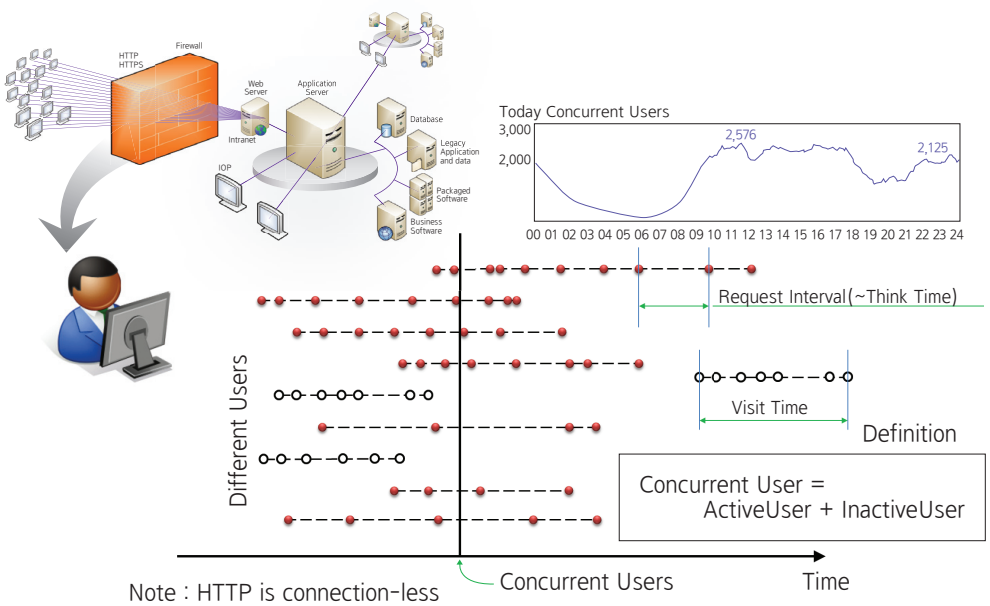
Classification	Think time	Remarks
TM(Telemarketing) system	10 ~ 15 sec	
MIS/intranet system	15 ~ 20 sec	
Internet banking system	30 ~ 35 sec	
Online shopping mall system	30 ~ 40 sec	
Portal system	40 sec or higher	

1.2.3 Request Interval

The request interval is defined as the total time spent by the user who sends a service request. It is the sum of response time and think time. Request interval is one of performance model variables directly related to the system capacity, and in automated performance test solutions, it plays an important rule as base of real-life scenario.

1.2.4 Concurrent User

Term "Concurrent Users" refers to users who are currently accessing the target system by sending service request or preparing to send the next service request. It is one of the performance model variables directly related to the system capacity. As mentioned earlier, in web based systems, it is very difficult to measure the number of concurrent users, due to the nature of HTTP protocol, thus using alternate approach is recommended as seen below.



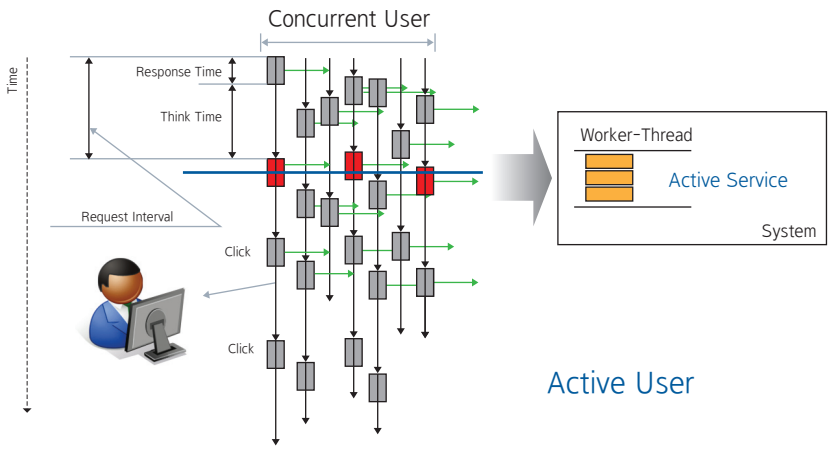
In above figure, service request made by an individual user at a certain point of time can be expressed as a dot on the graph. Each user makes service requests during the request intervals. Even if the TCP connection between the client and the server is cut off between each service request time, requests are made consistently, so you can assume that a user is currently using the system.

Up until now, there has not been a standardized definition of concurrent user in a web based environment. Some people would interpret concurrent users as named users and some as active users, which lead to confusion. Thus, in this white paper, definition of concurrent users is users sitting in front of a PC accessing the services provided by a web system.

1.2.5 Active User

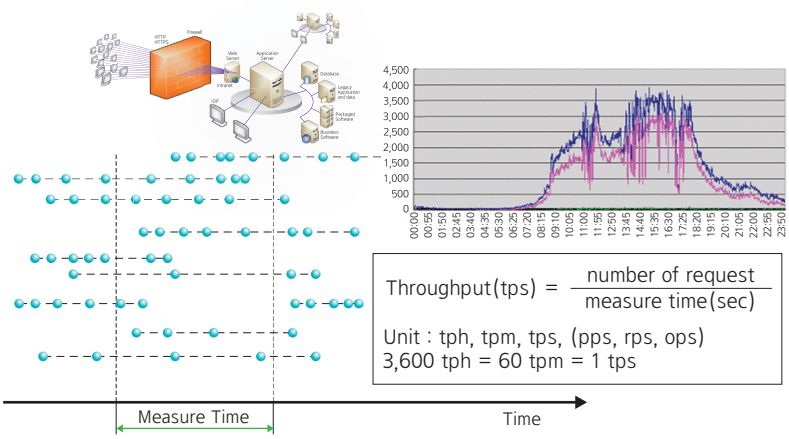
Each of concurrent users has his/her own response time and think time while sending service requests. If a vertical line is drawn on any specific timeline, then some users would not have received a response yet thus have no response time, and even those whose requests are being executed at that time. In the diagram below, there are three users who had sent requests and are waiting for their response. As shown here, an Active User is defined as a user who is waiting for the response after sending a service request.

If there are no network problems present, an active user's request is processed at the time when it reaches the server. Here, the number of services executed by the server at given time is defined as the number of "Active Service(s)". Whereas active user is a term used from the perspectives of clients, active service is a term used from the perspectives of servers.



1.2.6 Throughput

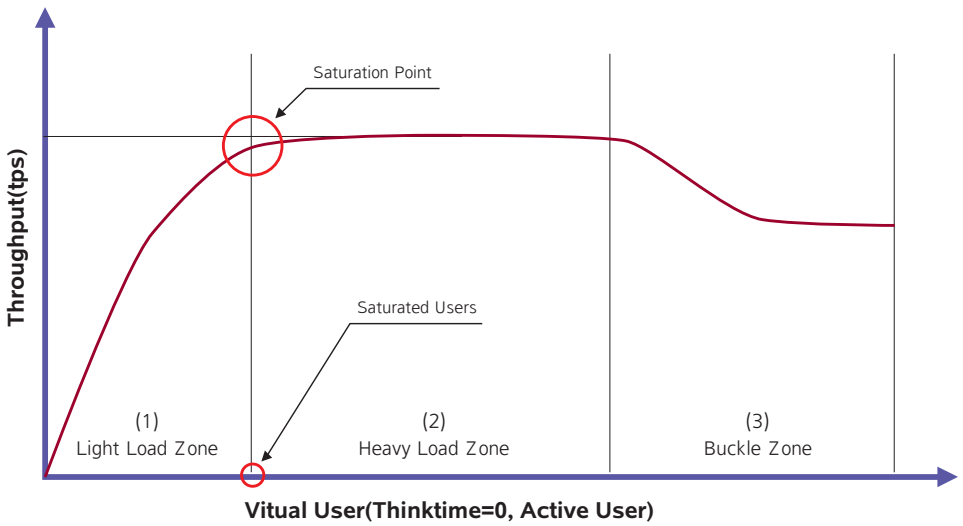
Throughput corresponds to User, which is one of the three performance evaluation perspective which was discussed earlier. It is defined as the number of user requests processed per unit time. It is differently expressed depending on the type resource or environment. In case of network resources, it is expressed in bits/sec or bytes/sec. For CPU resources, it is expressed in MIPS (millions of instructions per second) and for OLTP (On-Line Transaction Processing), it is expressed in transactions per sec or TPS. Since in this white paper is mostly about web systems and processed services, throughput will be measured in TPS for contents discussed in this article.



The throughput is calculated by dividing the number of processed user requests by time spent processing the requests, typically in seconds. As long as there is no delay of service caused by service queuing, the throughput should be equal to Request rate.

The characteristic of throughput may be generally characterized depending on the increase in user and load (request rate) compared to the capacity of the web system as shown in the below graph; the graph identifies the number of users in 3 groups: Light Load, Heavy Load, and Buckle zone.

In Light Load zone, the throughput is linearly proportional to the increase in the number of concurrent users. In here, there is no bottleneck so the throughput is increased proportionally according to the increase in user and user requests. Next, in the Heavy Load zone, when the request rate is increased to maximum the one or more constraint of the system can handle, the throughput cannot increase further bottleneck effect appears, but it still maintains a steady flow of throughput. Last, in the Buckle Zone, the throughput actually starts to decrease. In addition to the first bottleneck area, there will be other constraint bottleneck overlapping one another, which result in further degrading of system performance.

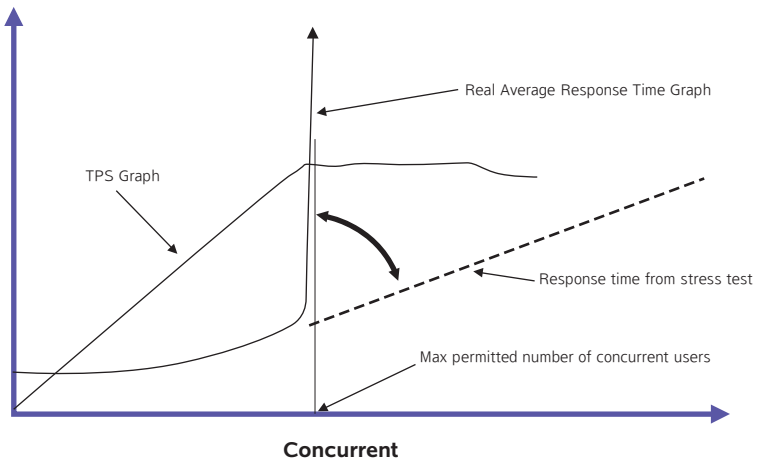


1.2.7 Throughput vs. Response Time

An information system performance analysis must include the correlation analysis between the Throughput and the Response time. In every benchmarking and performance tests reports, this item is always mentioned because it is that much important.

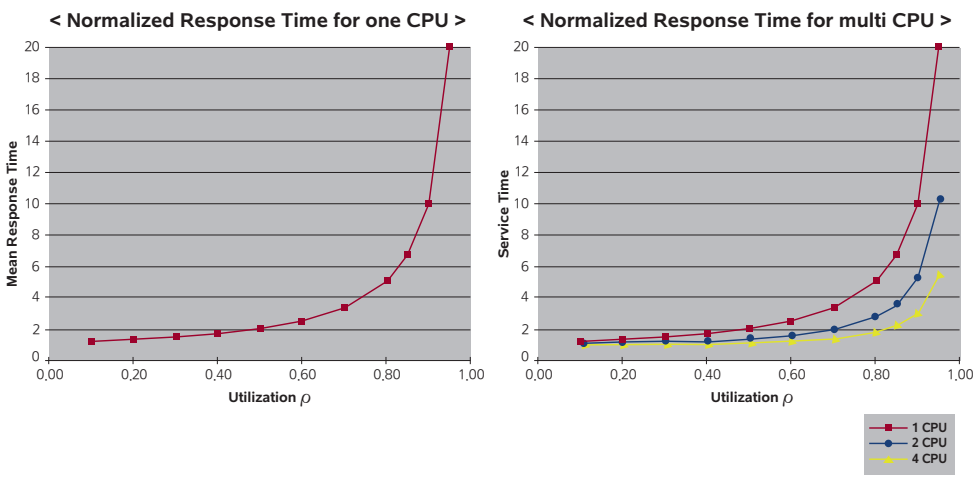
In principle, at the Saturation point (where one or more constraint of the system is used up to capacity), the throughput and the response time shows opposite behaviors with each other as the load is continually increased. The throughput is linearly increased up until the saturation point and after the saturation point, it is maintained steady throughput or throughput may even decrease. However, the response time is maintained at a steady rate until the saturation point reached, but after, the increasing queuing time due to the bottleneck effect rapidly increases response time for every additional increase in the load system processes.

Measuring the maximum capacity of system's performance, throughput graph needs to be set for the highest value attained before the saturation point and response time needs to be average value for all intervals prior to saturation point or 90 percentile value of interval prior to saturation point.



1.2.8 CPU Utilization vs. Response Time

Compare the changes in the response time against the CPU utilization in single/multiple CPU system and an interesting correlation can be found.



In the left graph (single CPU environment), as the CPU utilization is increased by more than 50%, the service queuing lengths is gradually increased and response time is increased significantly also. In other words, in the single CPU environment, when the CPU utilization is increase by more than 50%, the server queue length and the service response time are more rapidly increased proportionally compared to the CPU utilization. On the other hand, in the right graph (multiple CPU environment), as the number of CPU is increased, the response time is gradually decreased. When the number of CPU is 4 or more, CPU utilization is maintained a steady increase until at around 80%, and it rapidly increases afterwards.

Thus when planning for information system capacity, 70 to 80% of saturated CPU utilization rate is recommended because statistical data long observed based on performance models..

1.3 Web Based Performance Model

Performance analysis using a web based performance model is executed in following three phases as described below.

First, in the model construction phase, the objective of performance analysis is defined and the model type is determined. Performance analysis can be used for various purposes such as benchmarking, performance tests and capacity planning, etc... The objective needs to be clearly defined in advance. Since there are two types of performance model, practical verification method based on automated performance test solution and simulation verification method based on analytical models, which method to use should be decided in advance. In general, practical verification method is recommended for benchmarking and performance tests and simulation verification method is recommended for capacity planning.

Second, in the model parameterization phase, the model parameter that determines the performance model is defined. For a web based system performance model, the two types of model parameters are typically observed: Service Throughput which determines the workload and the response/think time which describes the demand. The model parameter values may be collected manually from the platform or systematically collected by using an APM (Application Performance Monitoring) solution.

Finally, in the model validation phase, the actual parameter values are applied to the predefined performance model for the verification purposes. In case of benchmarking or performance tests, the user requests are simulated for system performance verification. In case of interpretation models, a performance model based on the queuing theory is used to predict the future trends of performance.

In a web based system, the service subject to analysis includes all dynamic contents (JSP / Servlet / Web Service) executed by the web application servers, thus all request rate and the response time for all the services must be known. In order to systematically measure of each variable for large number of services, the gathering and analysis must be well planned out.

The correlations among the variables for this performance model are defined by the following equations. Since details of the following equations are beyond the scope of this white paper; please refer to the references attached at the end of this document.

Little’ s Law

This Little's law governs the relationship between the throughput and the response time and is used for measurement of active users or services in a web based system. When executing a performance test, if think time is set to 0 sec, and then the number of virtual users is proportional to the number of active users.

Active users(#) = throughput(tps) x response time(sec)

Response Time’ s Law

This law governs the relationship between the throughput and the number of concurrent users and is used to calculate of the number of virtual users/request intervals necessary for benchmarking and performance tests and prediction of the number of saturated users based on scalability tests.

Concurrent users(#) = throughput(tps) x (response time(sec) + think time(sec))

Active User’ s Law

This law determines the relationship between the number of concurrent users and the number of active users. It is used for measuring of the number of concurrent users when it is very difficult to measure the throughput.

Active users(#) = concurrent users(#) x response time(sec) / (response time(sec) + think time(sec))

Section 2

Performance Analysis in the Real World

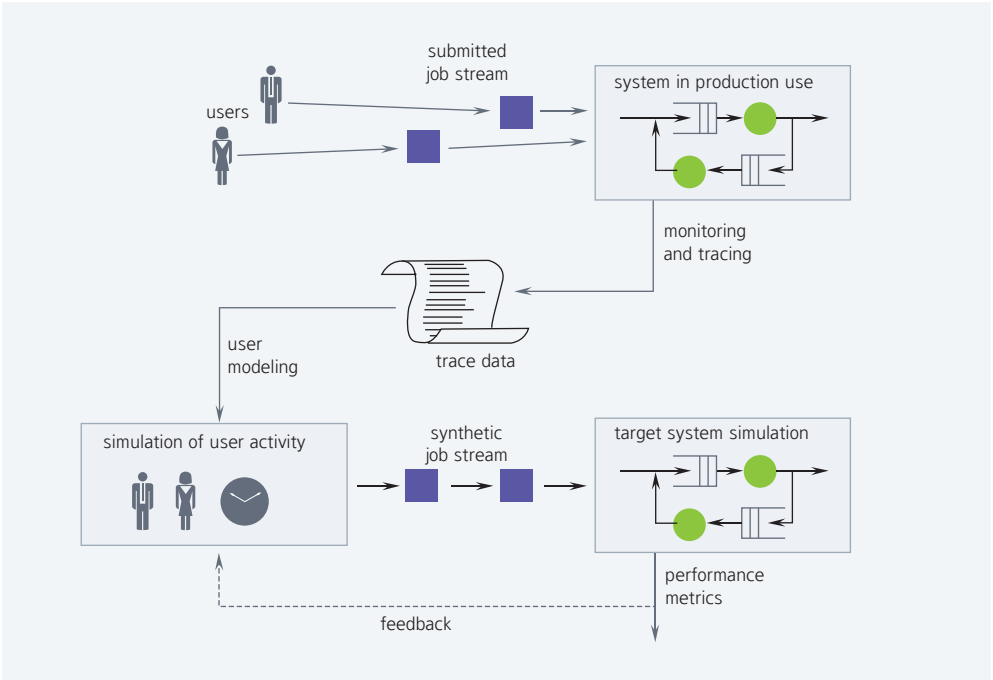
To perform analysis based on the performance models introduced earlier, establish a plan that can measure the input values for each performance model parameters in a production environment. There are three types of such plans depending on the type of performance model.

Firstly, in selecting Hardware, system administrator must be able to determine the target performance indicators such as throughput and the response time by measuring relevant values from existing system then convert them into the target performance indicators for the new systems.

Second, developed application must be measured for performance in test environment to check if it satisfies the response time and the throughput set for target performance indicators. These should also be determined by measure the relevant items in the existing system and convert them into the target performance values.

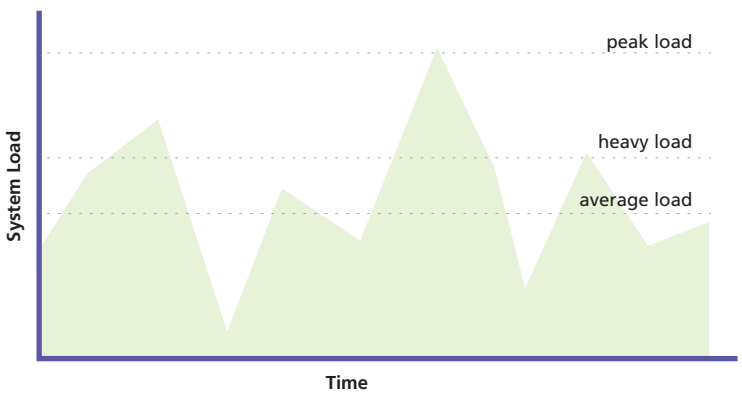
Third, to create and maintain a system which satisfies the throughput and response time set for target performance indicator, performance data must be sampled regularly and compared with established with baseline data, and then applied to the appropriate analytic model.

The below figure illustrated performance analysis activities based on the performance model and it indicates that actual measurement of the performance model parameters are executed by logging and monitoring of the currently operating system.



<Excerpts from the Workload Modeling for Computer System Performance Evaluation>

System Load VS Time



The monitoring period for measurement of the performance model parameters should be set around the time of peak loads. Moreover, since each peak load varies from one another, multiple measurements need to be made and an average should be obtained.

In a web based system, to collect the sample data, web log provided by a web server is used because it contains various values necessary for workload analysis. However, as the scale of system tends to be larger and the number of servers constantly increasing, it is becoming harder to use logs for the purpose of workload analysis due to huge amount of data which the administrator must sift through. Moreover, if analysis work is done as batch jobs, it is almost impossible to use only logs. Recently, as a great alternative for such problems, service monitoring features provided by APM solutions provide easy way to gather, organize, and analyze large amount of data quickly and easily.

2.2 APM Solution

The original purpose of APM solution was to improve the visibility performance indicators and ability to analyze performance problems in a system. Recently, in addition to the original purposes, some solutions provide service monitoring in the workload modeling features.

At present, APM solution JENNIFER is providing various types of service monitoring features in addition to the basic web application performance monitoring features.

- **Status of concurrent users**
- **Status of active service**
- **Status of business request rate**
- **Status of business processing(throughput)**
- **Status of mean response time**
- **Status of mean think time**
- **Status of mean CPU utilization rate**

Section 3

Conclusion

So far, we have discussed the performance analysis method in the real world and introduced the performance theory/model commonly used for performance analysis of an information system.

Today, companies need to establish high performing information system to handle drastically changing business requirements. Moreover, a well-established plan on verifying whether the implemented system can satisfy the performance requirements is a must.

I sincerely hope that the material introduced in this white paper can contribute to those who are engaged in implementing of new information systems and those who have interests in improving their systems.

<Reference>

1. "SAP AG" Theory and Practice of Sizing SAP Software
2. "Peter J. Denning" Queuing Network of Computes
3. "Andy Lee" "Mathematical Approach for Web-based System Performance"
4. "Dror G. Feitelson" "Workload Modeling for Computer Systems Performance Evaluation"

Section 4

About The Author

Bruce Park is currently working as a senior consultant in charge of application quality management at a BTO consulting company. For many years, Bruce has been engaged in various positions in area of application management such as performance benchmarking, performance testing, and quality management in enterprise environment settings.
Prior to joining BTO consulting company, he was employed as professional services consultant at HP Korea, Mercury Interactive Korea and other notable firms.

Bruce Park, Consultant, mail. bruce@jennifersoft.com

about JenniferSoft...

JenniferSoft, Inc. is the software vendor company with expertise in application performance monitoring and performance problem resolution, providing Application Performance Management (APM) solutions and services to enterprise companies around the world.
Technology is foremost of what's valued in JenniferSoft, as we strive to bring to the market the latest and best technology available.
We think that software solution should not be just a mesh of form and functions but it should be designed with its user in mind, each component formed with intent to elevate the experience of its user. JenniferSoft combines latest technology in APM with field-tested expertise and experience to bring to our customers a well-balanced solution that is both advanced in features and practical. With vision of combining technology and experience into one, JenniferSoft pledges to continue focuing on R&D to develope world class solution.

JENNIFER
JENNIFERSOFT, Inc.