METHODOLOGY OF CAPACITY PLANNING FOR WEB SYSTEM

Alexandru COREŢCHI

Compudava S.R.L.
Str. 31 August 1989
Chişinău, Moldova
acoretchi@compudava.com

Abstract. In an ideal world, performance would be engineered into software starting early in the development process. The reality, however, is that budget and schedule constraints often lead developers, particularly devotees of the "agile" methodologies, to adopt a "make it run, make it run right, make it run fast" strategy. The result is that, somewhere near the end of the project, performance problems appear.

Nearly everyone runs into performance problems at one time or another. Today's software development organizations are being asked to do more with less. In many cases, this means upgrading legacy applications to accommodate a new infrastructure (e.g., a Web front-end), improve response time or throughput, or both.

This article presents a quantitative approach to performance planning that helps prevent problems, identify potential solutions, and prioritize efforts to achieve the greatest application performance with the least effort.

Keywords: performance, capacity, workload, model

INTRODUCTION

Planning performance for Web-based system requires that a series of steps be followed in systematic way. It is a mistake to begin planning performance characteristic for a future system before having the data needed to isolate problems and select appropriate solutions. Precise, quantitative, measurable performance objectives must be defined [1]. This information can express performance objectives in several ways, including response time, throughput, or constraints on resource usage. Some examples are: "The response time for a transaction should be one second or less with up to 1,000 users." or "CPU utilization should be less than 65% for a peak load of 2,000 events per second". When defining performance objectives, don't forget that customer needs may change over the product's lifetime. For example, current performance objective may be to process 10,000 events per second. However, in two years, customer may need to be able to process 30,000

events per second. It is a good idea to consider future uses of the software, so that performance objectives can anticipate these changes and build in the necessary scalability. Those objectives can be achieved using methodology of Capacity Planning. In Figure 1 the main steps of methodology for Capacity Planning on Web-based systems are presented. Capacity planning is the process of planning for growth and forecasting peak usage period in order to meet system and application capacity requirements. It involves extensive performance testing to establish the application's resource utilization and transaction throughput under load [1]. The main steps of methodology are: understanding the environment, workload characterization, workload model validation and calibration, performance model development, performance model validation and calibration, workload forecasting, performance prediction, cost model development, cost prediction and cost/performance analysis.

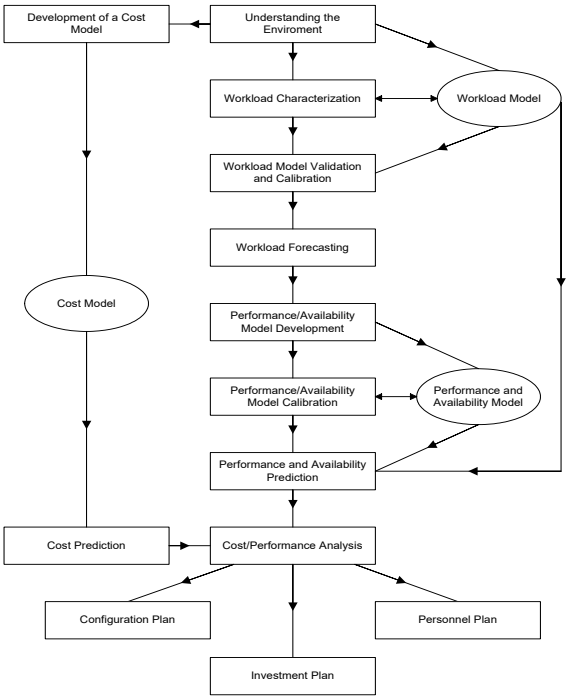

Figure 1. A methodology for capacity planning

UNDERSTANDING THE ENVIRONMENT

The initial phase of the methodology consists of learning what kind of hardware (clients and servers), software (operating system, middleware and applications), and network connectivity and network protocols are presented in the environment [2]. It also involves the identification of peak usage period and service-level agreements. In Table 1 the main element that must be cataloged and understood before the remain steps of methodology can be taken are presented.

Table 1. Environment elements

Elements	**Description**
Client Platform	Quantity and type
Server Platform	Quantity, type, configuration and function
Middleware	Type
DBMS	Type
Services/applications	Main Web services and applications supported
Network connectivity	Network connectivity diagram showing all LANs, WANs, network technology, routers, servers, load balancers, firewalls etc.
Networks protocols	List of all protocols used
Usage patterns	Peak periods (hour of days, day of week, week of month, month of year)
Service-level agreements (SLA)	Existing SLAs per Web Services. When formal SLAs are absent, industry standards can be used.
LAN management and support	LAN management support structure, size, expertise and responsiveness to users.

WORKLOAD CHARACTERIZATION

This is a process of precisely describing the system's global workload in terms of its main components. Each workload component is further discomposed into basic components. The basic components are then characterized by workload intensity (transaction arrival rate, for example) and by service demand parameters at each resource. In Table 2 examples of basic component parameters and types are presented.

(*WI=workload intensity; SD=service demand*)

Table 2. Examples of basic component parameters and types

Basic Components Parameters	**Parameter Type**
Sale Transaction	
Number of transaction submitted per client	WI
Number of clients	WI
Total number of I/Os to Sale DB	SD
Average message size sent/received by DB server	SD
Web-based training	
Average number of training session/day	WI
Average size of video file per session	SD
Average size of HTTP documents retrieved	SD
Average number of image files retrieved/session	SD
Average number of document retrieved/session	SD
Average CPU utilization of Web server	SD

This allows understanding the system performance characteristics such as CPU utilization, I/O rates and average service time, network utilization, message size, and so on.

Also, understanding the environment allows to identify the bottleneck(s)—the device(s) with the highest utilization. When workload intensity is high, large collection of workload measurements can be obtained. Dealing with such collection is seldom practical, especially if workload characterization results are to be used for performance prediction through analytic models. One should substitute the collection of measured values of all basic components by a more compact representation – one per basic component. This representation is called a ***workload model***.

Workload models can depict the end-to-end processing steps for performance scenarios. The models will help to quantify the effects of more complex solutions, particularly those that involve contention for resources. In building any model, abstraction of reality being modeled are made for simplicity, ease of data collection use and computation simplicity. The abstraction compromise the accuracy of the model, so the model must be validated within an acceptable margin of error, a process called ***model validation***. If a model is deemed invalid, it must be calibrated to render it valid. This is called ***model calibration***. Validating workload model entails running a synthetic workload composed of workload model results and comparing the performance measures thus obtained with those obtained running the actual workload. If the results match within 10-30 % margin of error, the workload model is considered to be valid.

WORKLOAD FORECASTING

Workload forecasting is a process of predicting how the system workload will vary in the feature. This process involves evaluating workload trends if historical data are available and/or analyzing the business or strategic plans of the organization and then mapping these plans to changes in business process (for example, staff increases and paperwork reduction initiatives will yield 50% more e-mail and Internet usage and 80% more request on the corporate Web sever). During this process, basic workload components are associated to business process so that changes in the workload intensity of these components can be derived from the business process and strategic plans.

PERFORMANCE MODELS

Capacity Planning involves predicting whether a system will deliver performance metrics (e.g. response time, throughput and availability) that meet desired or acceptable levels. Performance prediction requires the use of models. Two types of models may be used: simulation models and analytical models. Both types of models have to consider contention for resources and the queues that arise at each system resource – CPUs, discs, routers etc. Queues also arise for software resource

– threads, database locks protocols ports. Analytic models, based on set of formulas and/or computational algorithms, are quite appropriate for the performance prediction of any capacity planning process. A performance model is said to be valid if the performance metrics calculated by the model match the measurements of the actual system within a certain acceptable margin of error.

AVAILABILITY MODELS

Availability models provide a way of predicting the availability of Web services based on the configuration of the infrastructure used to support the services as well as on the intrinsic reliability of the different components used. Using this model we can answer such a question as "How many and what type of servers should be used to build a site that will have a 99.99% availability?"

COST MODEL

A capacity planning methodology requires the identification of major sources of cost as well as the determination of how costs vary with system size and architecture.

Costs are categorized into startup and operating costs. Startup costs are those in setting up the system, while operating costs are the annual expenses incurred to maintain the system and provide upgrades in hardware and software to avoid performance degradation and security vulnerabilities.

COST/PERFORMANCE ANALYSIS

Once the performance model is built and solved and a cost model developed, various analyses can be made regarding cost-performance tradeoffs. The performance model and cost model can be used to access various scenarios and configurations. For each scenario, we can predict what the performance of each basic component of the global workload will be and what the costs for the scenario are [2]. We will also need to quantify the effort to make changes. A list of options that compares the performance goals versus the costs will determine the best candidates. Note that we may not always choose the option with the best performance because other factors may make it less desirable. For example, the option with the best performance characteristics may be risky because it uses an unfamiliar new technology. Or, a change with a high performance payoff may negatively impact another important quality such as maintainability or reliability.

CONCLUSIONS

The main goal of this article was to present a methodology that leads the system performance planner, in a step-by-step way, through the process of determining the most cost-effective system configuration and network topology. The methodology presented here requires the use of three models:

- workload model;
- performance model;
- cost model.

The workload model captures the resource demands and workload intensity characteristics of the load brought to the system by different types of transactions and requests.

The performance model is used to predict response times, utilizations and throughput as a function of the system description and workload parameters.

The cost model accounts for software, hardware and personnel expenditures.

REFERENCE

[1] Daniel A. Menascé, Virgilio A. F. Almeida "*Capacity Planning for Web Services. Metrics, Models and Methods*". Prentice Hall 2002.
[2] Daniel A. Menascé, Virgilio A. F. Almeida "*Performance by Design: computer capacity planning by example*". Prentice Hall 2004.