

Capacity Planning: an Essential Tool for Managing Web Services

Virgílio Almeidaⁱ and Daniel Menascéⁱⁱ

Speed, around-the-clock availability, and security are the most common indicators of quality of service on the Internet. Management faces a twofold challenge. On the one hand, it has to meet customer expectations in terms of quality of service. On the other hand, companies have to keep IT costs under control to stay competitive. Therefore, capacity, reliability, availability, scalability, and security are key issues to Web service managers. Electronic business sites are complex computer-system architectures, with multiple interconnected layers composed of many software and hardware components, such as networks, caching proxies, routers, load balancers, high speed links, and mainframes with large databases. The nature of electronic business workload is also complex due to its transactional nature, security and authentication requirements, payment protocols, and the unpredictable characteristics of service requests over the Internet. Planning the capacity of electronic business services requires more than just adding extra hardware. It requires more than intuition, ad-hoc procedures, or rules of thumb. Many possible alternative architectures can be used to implement a Web service; one has to be able to determine the most cost-effective architecture and system. This is where the quantitative approach and capacity planning techniques come into play. *Capacity planning* techniques offer much more than just performance prediction. In these times of ubiquitous Internet services and businesses, capacity planning should be actually viewed as a powerful management technique.

Performance problems on the Internet are exacerbated by the unpredictable nature of service requesting and information retrieval over the Web. It is not uncommon for Web sites to experience, without warning, a manifold increase in traffic volume. This type of load spike, also known as “flash crowds,” creates terrible performance problems and slow page download times. Web delays frustrate customers and cost over four billion dollars each year to online businesses according to an Intelliquest report (www.intelliquest.com/press/). The causes of delays are various. Overloaded networks and servers are the most common ones. The viability of electronic business depends on the ability of the IT infrastructure to offer timely and reliable services. For companies, whose business depends on the behavior of their online services, long waiting times and unavailability can be disastrous. Electronic businesses must continuously guarantee quality of service to avoid losing sales and customers. Security, performance, and availability are key issues for any service on the Web. This article introduces capacity planning as an essential tool for managing quality of service on the Web and presents a methodology, where the main steps are: understanding the environment, characterizing the workload, modeling the workload, validating and calibrating the models, predicting the performance, analyzing the cost-performance plans, and suggesting actions.

Managing Web Services

The term “Web service” describes specific business functionality exposed by a company, usually through an internet connection, for the purpose of providing a way for another company or software program to use the service. The Web is evolving into a network of service providers. Web-based services are offered to potentially tens of millions of users by hundreds of thousands of servers (i.e., content providers and service providers). Users and customers count on being able to access any service at anytime. Customers increasing reliance on information-based services poses three requirements on the services provided by online businesses: availability, scalability, and cost-efficiency. Availability means that users and customers can count on being able to access any web service from anywhere, anytime, regardless of the load at the Web site and of the load at the network. Availability also means that services are provided with quality, i.e., short and predictable response time. Scalability means that providers of Web services should be able to serve a fast growing and unknown number of customers with minimal performance degradation. Cost-effectiveness means that quality of Web services, represented by availability and fast response times, should be achieved with minimal expenditures in IT infrastructure and personnel. Managing Web services involves being able to answer the following typical questions:

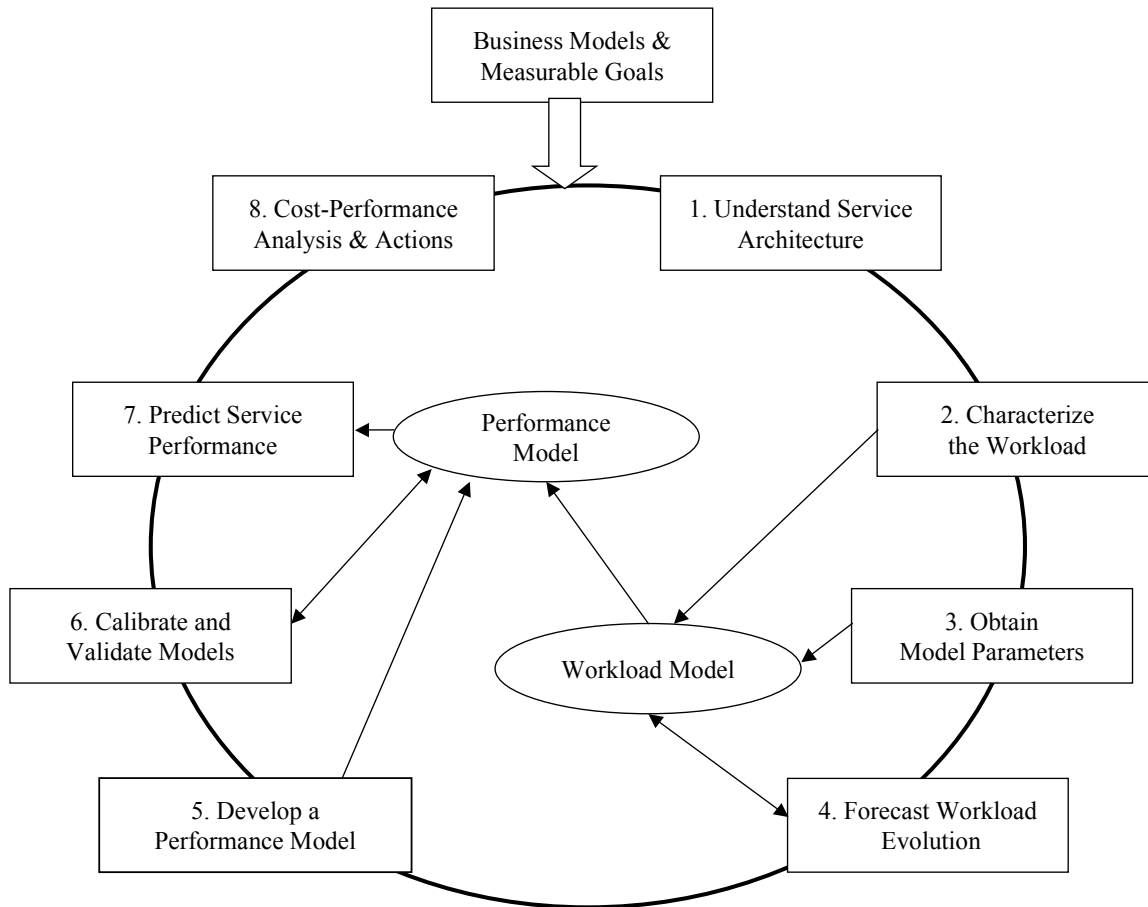
- ❑ Is the online trading site prepared to accommodate the surge in volume that may increase the number of trades per day by up to 75%?
- ❑ Is the number of servers enough to handle a peak of customers ten times greater than the monthly average?
- ❑ How many servers should be used to build the new site of the company that is expected to have a 99.99% of availability during business peak hours?
- ❑ How can we guarantee the quality of electronic customer service for the different scenarios of traffic growth? In a business-to-business environment, sending and receiving sensitive data, conducting financial transactions, and exchanging credit and production data depend on the secure and fast transmission of information.
- ❑ Various analyses can be made regarding cost-performance tradeoffs. Typical scenarios are: Should we use CDN services to serve images? Should we use Web hosting services? Should we mirror the site to balance the load, cut down on network traffic and improve global performance?
- ❑ E-Business sites may become popular very quickly. How fast can the site architecture be scaled up? What components of the site should be upgraded? Database servers? Web servers? Application servers? Network link bandwidth?

Capacity Planning for Web Services

Planning the capacity of Web services requires that a series of steps be followed in a systematic way. Figure 1 gives an overview of the main steps of the quantitative approach to analyze Web services. The starting point of the process is the business model and its measurable objectives, which are used to establish service level goals and to find out the applications that are central to

the goals. Once the business model and its quantitative objectives have been understood, one is able to go through the quantitative analysis cycle.

Fig 1: Capacity Planning Process



We now cover the various steps of the quantitative analysis cycle:

1. The first step entails obtaining an in-depth understanding of the service architecture. This means answering questions such as: What are the system requirements of the business model? What is the configuration of the site in terms of servers and internal connectivity? How many internal layers are there in the site? What types of servers (i.e., HTTP, database, authentication, streaming media) is the site running? What type of software (i.e., operating system, HTTP server software, transaction monitor, DBMS) is used in each server machine? How reliable and scalable is the architecture? This step should yield a systematic description of the Web environment, its components, and services.
2. The second step characterizes the workload of an e-business site. E-business workloads are composed of sessions. A session is a sequence of requests to execute e-business

functions made by a single customer during a single visit to a site. Examples of e-business functions requested by an online shopper include browse the catalog, search for products or services based on keywords, select products to obtain more detailed information, add to the shopping cart, user registration, and checkout. A customer of an online brokerage site would request different functions, such as enter a stock order, research a mutual fund history, obtain real-time quotes, retrieve company profiles, and compute earning estimates. Each service request may exercise the site's resources in different manners. Some services may use large amount of processing time from the application server while others may concentrate on the database server. Different customers exhibit different navigational patterns and, as a consequence, invoke services in different ways with different frequencies. For instance, in an e-commerce service, some customers may be considered as heavy buyers while others, considered occasional buyers, would spend most of their time browsing and searching the site. Understanding the customer behavior is critical for achieving the business objectives as well as to an adequate sizing of the site's resources. Graph-based models of customer behavior can be quite useful as indicated later. In addition to characterizing navigational patterns within sessions, one needs to characterize the rate at which sessions of different types are started. This gives us an indication of the workload intensity.

3. The third step consists of obtaining values for the parameters of the workload models. This step also involves monitoring and measuring the performance of a Web service. It is a key step in the process of guaranteeing quality of service and preventing problems. Performance measurements should be collected from different reference points, carefully chosen to observe and monitor the environment under study. For example, logs of transactions and accesses to servers are the main source of information. Further information, such as page download times from different points in the network may help to track the service level perceived by customers. The information collected should help us answer questions such as: What is the number of customer visits per day? What is the site revenue for a specific period of time? What is the average and peak traffic to the site? What characterizes the shoppers of a particular set of products? What are the demands generated by the main requests on the resources (e.g., processors, disks, and networks) of the IT infrastructure? Steps 2 and 3 generate the *workload model*, which is a synthetic and compact representation of the workload seen by a Web service.
4. The fourth step forecasts the expected workload intensity for a Web service. The techniques and strategies for forecasting predictability in the electronic marketplace should provide answers to questions such as: How will the number of users of an online auction vary during the next six months? What will be the number of simultaneous users for the streaming media services six months from now?
5. In the fifth step, quantitative techniques and analytical models based on queuing network theory are used to develop performance models of Web services. Performance models can be used to *predict performance* when any aspect of the workload or the site architecture is changed.

6. The sixth step aims at validating the models used to represent performance and workload. A performance model is said to be validated if the performance metrics (e.g., response time, resource utilizations, and throughputs) calculated by the model match the measurements of the actual system within a certain acceptable margin of error. Accuracies from 10% to 30% are acceptable in capacity planning.
7. Prediction is key to capacity planning because one needs to be able to determine how a Web service will react when changes in load levels and customer behavior occur or when new business models are developed. This determination requires predictive models and not experimentation. So, in the seventh step, one uses the performance models to predict the performance of Web services under many different scenarios.
8. In the eighth step of the cycle, many possible candidate architectures are analyzed in order to determine the most cost-effective one. Future scenarios should take into consideration the expected workload, the site cost, and the quality of service perceived by customers. Finally, this step should indicate to management what actions should be taken to guarantee that the IT services will meet the business goals set for the future.

The next two sections discuss customer behavior models—an important component of the workload model—and performance models, in more detail.

Customer Behavior Model Graph (CBMG)

A *customer model* captures elements of user behavior in terms of navigational patterns, e-commerce functions used, frequency of access to the various e-commerce functions, and times between accesses to the various services offered by the site. A customer model can be useful for navigational and workload prediction. By building models, such as the Customer Behavior Model Graph (CBMG), one can answer what-if questions regarding the effects on user behavior due to site layout changes or content redesign. Such models could potentially be used to predict future moves of a user and pre-fetch objects in order to improve performance.

Consider an online bookstore in which customers can perform the following functions:

- Connect to the home page and browse the site by following links to bestseller books and promotions of the week per book category.
- Search for titles according to various criteria including keywords, author name, and ISBN.
- Select one of the books, which result from a search, and view additional information such as a brief description, price, shipping time, ranking, and reviews.
- Register as a new customer of the virtual bookstore. This allows the user to provide a user name and a password, payment information (e.g., credit card number), mailing address, and e-mail address for notification of order status and books of interest.
- Login with a user name and password.

- Add items to the shopping cart.
- Pay for the items in the shopping cart.

Thus, during a session with the online bookstore, a customer issues several requests that will cause the above functions to be executed. For example, a customer may cause a search to be executed by submitting a URL that specifies the name of an application to be run at the server through a server API (e.g., CGI and ASP) and the keywords to be used in the search. The application will then execute a search in the site's database and return an HTML page with all the books that match the search criteria.

During a session, a customer may be classified as being in different states according to the type of function requested. For example, the customer may be browsing, searching, registering as a new customer, logging in, adding books to the shopping cart, selecting the result of a search, or checking out. The possible transitions between states depend on the layout of the site. For example, one customer may go from the home page to search, from search to select, from select to add to cart, and from there to pay. Another customer may go from the home page to the browse state before doing a search and then leave the online bookstore without buying anything.

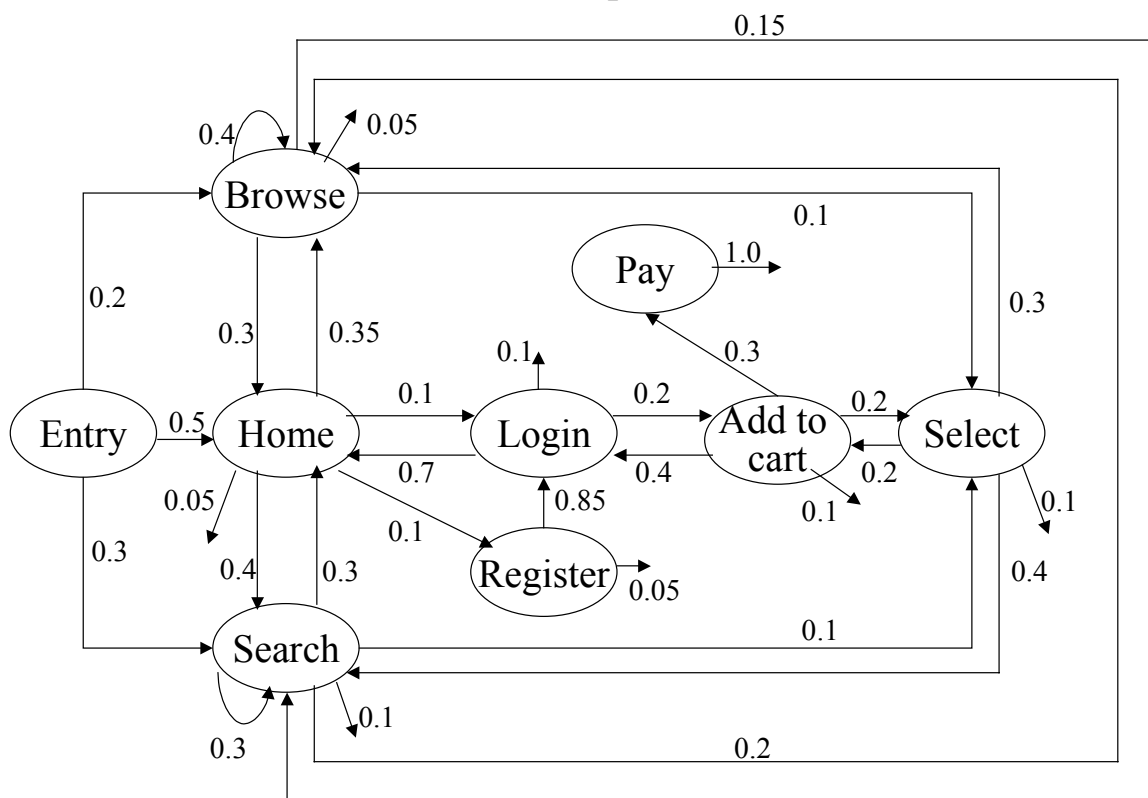
To capture the possible transitions between the states in which a customer may be found, we need a model that reflects the navigational pattern of a user during a visit to an e-commerce site. This model is in the form of a graph, as shown in Figure 2, and is called the Customer Behavior Model Graph (CBMG). The nodes of the CBMG depict the states a customer is in during a visit to the e-commerce site. Arrows connecting states indicate possible transitions between them. The states of this CBMG and their descriptions follow.

Entry is a special state that immediately precedes a customer's entry to the online store. This state is part of the CBMG as a modeling convenience and does not correspond to any action started by the customer. Home is the state a customer is in after selecting the URL for the site's home page. Customers may leave the site from any state. Thus, there is a transition from all states, except the Entry state, to the Exit state, not shown explicitly in Figure 2. The transitions to the Exit state are shown as dangling arrows leaving a state.

In the case of Figure 2, customers can enter the virtual bookstore at only three states: Home, Browse, and Search. From the Home state, they can visit the Register, Login, Browse, and Search states as well as exit the site. Note that Figure 2 reflects all possible transitions between states. However, during a single visit to the e-store, a customer may not visit all states, and different visits by the same customer or by a different customer may differ in terms of the frequency by which states are visited. Thus, in order to provide a complete characterization of customer behavior during a visit to a site, one must also capture the frequency by which transitions occur, as illustrated in Figure 2. By processing Web logs, one can identify sessions and build a CBMG for each session. Instead of transition frequencies out of each state, the CBMG for each session will have a transition count indicating how many times during a session a customer went from one state to another. For example, a customer could have made five transitions from Search to Select out of twenty transitions that leave the Search state. The set of

all these session CBMGs can then be grouped, using clustering techniques, into CBMGs that represent “similar” types of sessions (see Menascé and Almeida, “Scaling for E-business,” chapter 11, Prentice Hall, 2000, for a method to obtain clusters of CBMGs out of Web logs). As a result of this clustering analysis, one may identify interesting customer patterns, such as “heavy-buyers” or “window-shoppers.” The site revenue can be improved by giving higher priority (i.e., better quality of service) to customers with higher probability of making a purchase. The set of all these CBMGs along with the session start rate for each type of session constitutes the workload model.

Figure 2. Customer Behavior Model Graph (CBMG) for the Online Bookstore Example



A Simple Performance Model for The Scalability Problem

What do people mean when they say that a system is scalable? We consider a system to be scalable if there is a “straightforward” way to upgrade the system to handle an increase in traffic while maintaining adequate performance. By straightforward we mean that no system or software architectural changes should be required to scale the system. Examples of straightforward changes are: adding more servers to a system that already employs multiple servers, adding more CPUs to a multiprocessor, replacing existing servers with faster servers that use the same architecture. One approach to upgrading capacity is *scaling horizontally* or *scaling*

out, which means adding more servers of the same type, while *scaling vertically* or *scaling up* means replacing the existing servers with faster ones. Scalability is a key issue for services on the Web. Mission critical business sites require careful planning and design in order to ensure that the application delivers reliable and scalable services. The entire end-to-end system must be analyzed and one must understand and document the characteristics and performance of applications, servers, networks, load balancers, and firewalls. However, in many cases, scalability cannot be achieved because of the existence of *bottlenecks*, i.e., hardware and/or software resources that limit the overall performance of a system.

Performance analysis is a key technique to understanding scalability problems in electronic business. Because it is difficult to estimate traffic, e-business sites must be designed with scalability in mind. In other words, a designer of an online business must know a priori what are the limits of the system. For instance, a designer must know the maximum number of transactions per second the system is capable of processing (i.e., an upper bound on throughput) or the minimum response time that can be achieved by the business site (i.e., a lower bound on response time). Performance bounding techniques allow us to calculate optimistic and pessimistic bounds. Throughput upper bounds and response time lower bounds are optimistic bounds. Scalability analysis refers to techniques that find a single bottleneck that cannot be sped up. When a bottleneck cannot be removed, the system is considered non-scaleable in terms of performance. In short, managers must be aware in advance of the capacity limitations of their e-business systems.

Business sites with their unpredictable traffic spikes bring new challenges to performance modeling. Detailed and costly modeling analysis may not be worthwhile when a capacity planning analyst faces a large number of possible future scenarios. Quick bounding studies may be the right solution for these cases. Consider an online business that is preparing for a surge of customers due to a special event, such as the World Soccer Cup, or an ad campaign. Management does not know how many customers would be attracted to the site during the games of the World Soccer Cup. Some market analysts estimate that the number of visitors vary from game to game, depending on which teams are playing. However, they expect something in the order of 30 to 40 million visitors during the two-hour period of the final game. Developing a detailed model to calculate that the proposed system will be able to support 5,555 visitors per second may be an overkill. Simply knowing that the site is able to serve approximately 1,000 visitors per second for one alternative or 8,000 for another alternative is the right level of information to select one option over another. Consider the following example of bounding analysis. The search e-business function requires 0.005 seconds of disk I/O on average and consider that disk I/O at the database server is the bottleneck for this type of transaction. Then, according to the bounding analysis models, the maximum throughput is the inverse of the total time spent at the bottleneck resource. So,

$$\text{MaximumThroughput} = \frac{1}{\text{TotalBottleneckTime}} = \frac{1}{0.005} = 200 \text{ tps.}$$

In other words, the site cannot process more than 200 search transactions per second. Suppose that 2% of the search requests generate a sale and that each sale generates \$25 on average. Thus, the upper bound on the “revenue throughput” is \$150 per second. This type of metric may be more meaningful to the management and gives them an indication on how the IT infrastructure may limit the business revenue.

The bottom line in managing Web services is guaranteeing performance, availability, and ROI. This can only be achieved if the IT infrastructure is ready to provide customers with service of high quality. The IT infrastructure of Web services is complex enough to preclude any guesswork when it comes to capacity planning. When planning the site capacity, it is very important to make sure that the site can handle the peak and not just the average load.

ⁱ **Virgilio A. F. Almeida** is a Professor of Computer Science at the Federal University of Minas Gerais (UFMG), Brazil. He holds a PhD in Computer Science from Vanderbilt University. He has published extensively in the area of distributed systems and Internet performance. Almeida held visiting faculty and research positions at Boston University, HP Research Laboratory, XEROX PARC. (virgilio@dcc.ufmg.br)

ⁱⁱ **Daniel Menascé** is a Professor of Computer Science at George Mason University and the co-director of its E-Center for E-Business. He holds a Ph.D in Computer Science from the University of California at Los Angeles (UCLA). He is a Fellow of the ACM and has published extensively in the areas of Web and e-commerce performance, capacity planning, and software performance engineering. Menascé has consulted extensively for government and private organizations in the area of capacity planning and performance modeling. (menasce@cs.gmu.edu)

Menascé and Almeida are the co-authors of “Capacity Planning for Web Services,” “Scaling for E-Business,” and “Capacity Planning and Performance Modeling,” published by Prentice Hall in 2002, 2000 and 1994, respectively.