*Capacity planning addresses the unpredictable workload of an e-business to produce a competitive and cost-effective architecture and system.*

**Virgílio A.F. Almeida and Daniel A. Menascé**

# Capacity Planning: An Essential Tool for Managing Web Services

**S**peed, 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 must meet customer expectations in terms of quality of service. On the other hand, companies have to control IT costs to stay competitive. Therefore, capacity, reliability, availability, scalability, and security are key issues to Web services managers. E-business sites are complex 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 large-database mainframes. The e-business workload—composed of transactions and requests submitted to e-business services—is also complex because of its bursty and highly skewed load characteristics. Security and authentication requirements, payment protocols, and the unpredictable characteristics of Internet service requests add to the complexity.

For example, it is common for Web sites to experience, without warning, a manifold increase in traffic volume. This type of load spike, also known as a *flash crowd*, creates terrible performance problems and slow download times. Such Web delays frustrate customers and cost online business over $4 billion each year, according to a report from IntelliQuest, a market research firm (http://www.intelliquest. com).

That's why planning e-business service capacity requires more than just adding extra hardware based on intuition, ad hoc procedures, or rules of thumb. Many possible alternative architectures can implement a Web service; you must determine the most cost-effective architecture and system. This is where a quantitative approach and capacity planning techniques come into play. Capacity planning offers much more than just performance prediction—it is actually a powerful technique for managing Web services.
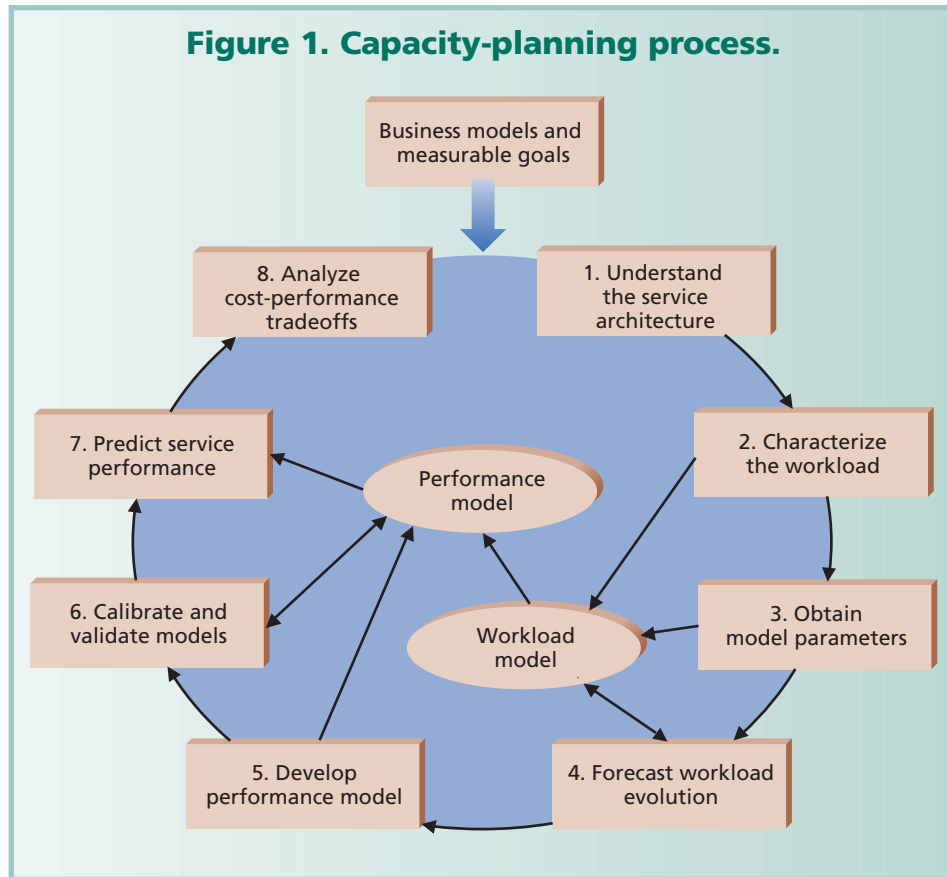
## MANAGING WEB SERVICES

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. As the Web evolves into a network of service providers, companies offer Web-based services to potentially tens of millions of users via hundreds of thousands of servers. Users and customers count on the ability to access any service at any time. Customers' increasing reliance on information-based services imposes three requirements—availability, scalability, and cost efficiency—on the services provided by online businesses.

Availability means that users and customers can count on accessing any Web service from anywhere, anytime, regardless of the Web site or network loads. Availability also means that the site provides services meeting some measure of quality, such as a short and predictable response time.

## Figure 1. Capacity-planning process.



Figure 1. Capacity-planning process.

Scalability means that Web service providers should be able to serve a fast-growing and unknown number of customers with minimal performance degradation. Cost effectiveness means that the quality of Web services, represented by availability and fast response times, should come with adequate expenditures in IT infrastructure and personnel. Managing Web services involves answering the following typical questions:

- Is the online trading site prepared to accommodate a surge in volume that could increase the trades per day by up to 75 percent?
- Is the number of servers enough to handle a customer access peak 10 times greater than the monthly average?
- How many servers are necessary to build the company's new site, which management expects to have a 99.99 percent availability during business peak hours?
- How can we guarantee the quality of electronic customer service for various traffic growth scenarios? 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.
- A variety of analyses concern cost-performance trade-offs. Typical questions about these scenarios include:

Would CDN (content delivery network) services be an appropriate choice to serve images? Would Web hosting

services be a net benefit? Would the cost of establishing a mirror site add enough advantages in terms of balancing the load, reducing network traffic, and improving global performance?

- E-business sites can become popular very quickly. How fast can the site architecture scale up? What site components would need upgrading? Will expanding the site require additional database servers, Web servers, or application servers; or network link bandwidth?

### Capacity planning for Web services

Planning Web service capacity requires systematically following a series of steps. Figure 1 gives an overview of the main steps in the quantitative approach to analyzing Web services. The process' starting point is the business model and its measurable objectives, which establish service-level goals and identify the applications central to these goals. Quantitative analysis is possible only after defining the business model and its quantitative objectives.

*Understand the service architecture.* The first step of quantitative analysis entails obtaining an in-depth understanding of the service architecture. This means answering questions such as: What are the business model's system requirements? What is the site configuration in terms of servers and internal connectivity? How many internal layers does the site have? What types of servers (HTTP, database, authentication, or streaming media) does the site run? What type of software (operating system, server software, transaction monitor, or database management system) does each server use? How reliable and scalable is the architecture? By answering these questions, this step should yield a systematic description of the Web environment, its components, and services.

*Characterize the workload.* The second step characterizes the site's workload. Sessions, the building blocks of e-business workloads, are sequences of requests to execute e-business functions. A single customer makes the requests

during a single visit to a site.

For example, an online shopper might request e-business functions such as browsing the catalog, searching for products or services based on keywords, selecting products to obtain more detailed information, adding items to a shopping cart, registering for accounting and fast checkout services, and checking out. A customer at an online brokerage site would request different functions, such as entering a stock order, researching a mutual fund history, obtaining real-time quotes, retrieving company profiles, and computing earnings estimates.

Each service request exercises the site's resources differently. Some services can use large amounts of processing time by an application server; others can 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 qualify as repeat buyers, so do not use the site to obtain information. Other occasional buyers would spend most of their time browsing and searching the site. Understanding customer behavior is critical for achieving business objectives and for sizing the site's resources. Graph-based models of customer behavior can be quite useful. In addition to characterizing navigational patterns within sessions, you must also characterize the rate at which different types of sessions start. This data indicates workload intensity.

***Obtain model parameters.*** The third step consists of obtaining parameters for the workload models. This step also involves monitoring and measuring Web service performance, a key to guaranteeing quality of service and preventing problems.

Performance measurements should come from different reference points. The company management should carefully choose performance measurements to observe and monitor the environment under study. For example, transaction and server access logs are main sources of information. Further information, such as page download times from different points in the network, can help track the service level from the customer's viewpoint. The collected information should help answer questions such as how many customers visit the site per day? What are the average and peak traffic to the site? What characterizes those who buy a particular set of products? What demands do the main requests make on the resources (such as processors, disks, and networks) of the IT infrastructure? Steps 2 and 3 generate the workload model, a synthetic and compact representation of a Web service's workload.

***Forecast workload evolution.*** The fourth step forecasts the Web service's expected workload intensity. This step

should provide answers to questions such as how will the number of online-auction users vary during the next six months? How many simultaneous users will access streaming-media services six months from now?

***Develop performance model.*** In the fifth step, we use quantitative techniques and analytical models based on queuing theory to develop performance models. Such models can predict performance after changes in workload or the site's architecture.

***Calibrate and validate models.*** The sixth step aims at validating the models that represent performance and workload. A performance model is valid if the performance metrics calculated by the model (response time, resource utilization, and throughput) match the actual system measurements within a certain acceptable margin of error. Accuracies from 10 to 30 percent are acceptable in capacity planning.

***Predict service performance.*** Prediction is key to capacity planning because you must determine how a Web service will react when changes in load levels and customer behavior occur or when new business models develop. This determination requires predictive models and not experimentation. So, in the seventh step, the performance models predict Web service performance under many different scenarios.

***Analyze cost-performance tradeoffs.*** The eighth step calls for analyzing many possible candidate architectures to determine the most cost-effective one. Future scenarios should consider the expected workload, site cost, and customer-perceived quality of service. Finally, this step should indicate to management what actions will guarantee that IT services will meet future business goals.

## Resources

➤ *Capacity Planning for Web Services: Metrics, Models, and Methods,* **Daniel A. Menascé and Virgilio A.F. Almeida, Prentice Hall, Upper Saddle River, N.J., 2002.**

➤ *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software,* **Connie U. Smith and Lloyd Williams, Addison Wesley, New York, 2001.**

➤ *Scaling for E-business: Technologies, Models, Performance, and Capacity Planning,* **Daniel A. Menascé and Virgilio A.F. Almeida, Prentice Hall, Upper Saddle River, N.J., 2000.**

➤ *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling,* **Raj Jain, John Wiley & Sons, New York, 1991.**

## Table 1. Companies offering products and services for Web and e-commerce capacity planning.

| Company and URL | Description |
|---|---|
| Accrue Software, http://www.accrue.com | Analyzes Internet marketing campaigns, Web-based content effectiveness, and e-commerce merchandising. |
| Appliant, http://www.appliant.com | Analyzes customer experience. |
| BMC Software, http://www.bmc.com | Measures end-to-end response time and site management; predicts performance through analytic models. |
| Cyrano, http://www.cyrano.com | Monitors performance and tests loads. |
| Exodus Communications, http://www.exodus.com | Provides monitoring and management services. |
| HP Openview, http://www.openview.hp.com | Provides performance monitoring and diagnostic tools. |
| Keynote Systems, http://www.keynote.com | Measures end-to-end response time and availability; provides benchmark indices for Web and e-commerce performance; provides load-testing services. |
| Mercury Interactive, http://www.mercuryinteractive.com | Provides application performance management tools. |
| NetIQ, http://www.netiq.com | Provides tools for managing, monitoring, and analyzing performance and availability of applications and servers. |
| Peakstone, http://www.peakstone.com | Provides capacity management and technology to let enterprises quantify and dynamically allocate site capacity. Controls dynamic service quality. |
| Performant, http://www.performant.com | Provides performance management and workload characterization tools. |
| RSW Software, http://www.rswsoftware.com | Provides functional/regression testing, load/scalability testing, and Web application quality monitoring. |
| Segue, http://www.segue.com | Provides load and performance testing. |
| Tivoli, http://www.tivoli.com | Provides resources for managing performance and availability. |

### Tools and services

Several companies offer tools and services to aid in capacity planning. We list some of them in Table 1.

### DEFINING CUSTOMER BEHAVIOR

The workload model describes the workload of an e-business service in terms of workload intensity (such as transaction arrival rates) and service demands on the various resources (such as processors, I/O subsystems, and networks) that make up the service. The workload model can be derived from the *customer model*, which captures elements of user behavior in terms of navigational patterns, e-business functions used, frequency of access to the various e-business functions and times between access to the various functions provided by the service. It takes a bit of planning to translate customer behavior into workload models that drive system decisions. We begin by first defining and quantifying the customer behavior that affects system performance. A customer model helps navigational and workload prediction. Models can answer what-if questions about the effects of site layout changes or content redesign on user behavior. They can potentially predict future user movements and prefetch objects to improve performance.
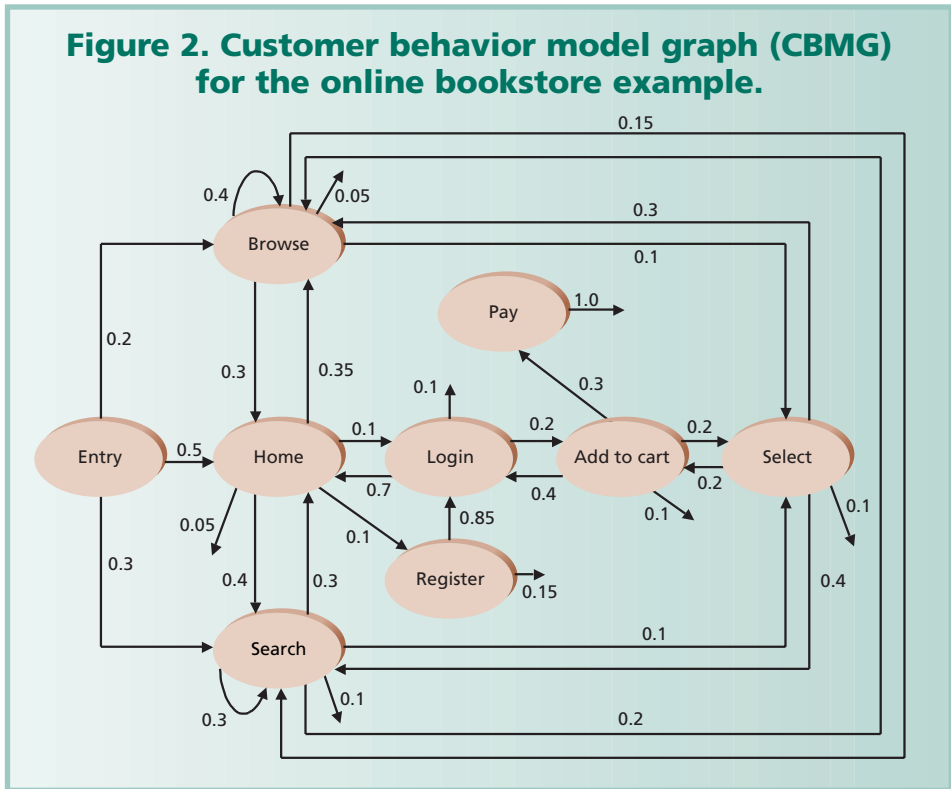
### Bookstore example

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 bestsellers and promotions of the week per book category.
- Search for titles according to various criteria including key-

words, author name, and ISBN.

- Select one book 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 lets the user provide a user name and password, payment information (such as a credit card number), mailing address, and e-mail address for order status and books-of-interest notifications.
- Login with a user name and password.
- Add items to the shopping cart.
- Pay for the items in the shopping cart.



**Figure 2. Customer behavior model graph (CBMG) for the online bookstore example.**

During a session with the online bookstore, a customer issues several requests that will execute these functions. For example, customers can execute a search by submitting a URL specifying the name of an application that runs at the server as well as the keywords the search will use. Examples of technologies that execute server-side applications include CGI (common gateway interface) scripts and ASPs (Active Server Pages). 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. The site can also classify a customer in different states, according to the function the customer requests. For example, the customer can browse, search, and register (as a new customer), log in, add books to a shopping cart, select the result of a search, or check out. The possible transitions between states depend on the site layout. For example, one customer might go from the home page to search, from search to select, from select to add to cart, and from add to cart to pay. Another customer might go from the home page to browse before performing a search and then leave the online bookstore without buying anything.

## Customer behavior model graph

To capture the possible transitions between a customer's states, we proposed a model that reflects a user's navigational pattern during a visit. This model is in the form of a graph, the customer behavior model graph (CBMG); Figure 2 shows an example. CBMG nodes depict a customer's states during a visit. Arrows connecting states indicate possible transitions.

*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. Because customers can leave the site from any state, each state except Entry has a transition to the Exit state. We do not fully show these transitions in Figure 2. Instead, the CBMG depicts them as dangling arrows leaving a state.

In the case represented by Figure 2, customers can enter the virtual bookstore at only three states: Home, Browse, and Search. From Home, they can visit 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, a customer may not visit all states. Also, the frequency with which the same customer visits states will probably change from visit to visit. So to provide a complete characterization of customer behavior, you must also capture the transition frequency, as Figure 2 shows. By processing Web logs, you 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 make five transitions from Search to Select out of 20 transitions that leave the Search state. Using clustering techniques, the set of all

these session CBMGs can group into CBMGs that represent similar types of sessions (see Daniel Menascé and Virgílio Almeida, *Scaling for E-Business*, chapter 11, Prentice Hall, 2000, for a method to obtain clusters of CBMGs out of Web logs). This clustering analysis can identify interesting customer patterns, such as heavy buyers or window shoppers. Companies could improve site revenue giving higher priority (for example, better quality of service) to customers who are more likely to make a purchase.

The set of all these CBMGs and the session arrival rates for each type of session constitute the workload model.

## SIMPLE PERFORMANCE MODEL

In capacity planning, performance models are an important class of models that are most useful in answering questions about the behavior of a system and can determine a system's scalability. First, consider what 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 scaling the system shouldn't require any changes to the system or software architecture. Examples of straightforward changes are adding more servers to a system that already employs multiple servers, adding more CPUs to a multiprocessor, or 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. Scaling vertically or scaling up means replacing existing servers with faster ones. Scalability is a key issue for Web services. Mission-critical business sites require careful planning and design to ensure that the application delivers reliable and scalable services. You must analyze the entire end-to-end system to understand and document the characteristics and performance of applications, servers, networks, load balancers, and firewalls. However, in many cases, scalability is not achievable because of bottlenecks, such as hardware or software resources that limit a system's overall performance.

### Performance analysis

Performance analysis is a key technique to understanding scalability problems in e-business. Because estimating traffic is difficult, an e-business designer must know beforehand the system's limits, keeping scalability in mind. For instance, a designer must know the maximum number of transactions per second the system can process or the maximum response time that the business site can tolerate. Performance-bounding techniques allow you to calculate optimistic and pessimistic bounds. Throughput upper bounds and response time lower bounds are optimistic bounds.

### Scalability analysis

Scalability analysis refers to techniques that find a single bottleneck that cannot be sped up. Irremovable bottlenecks make the system nonscalable in terms of performance. Managers must know their systems' capacity limitations in advance. With their unpredictable traffic spikes, business sites bring new challenges to performance modeling. Detailed and costly modeling analysis might not be worthwhile when a capacity-planning analyst faces many possible future scenarios. Quick bounding studies might be the right solution for these cases.

### Example

Consider an online business that is preparing for a surge of customers because of a special event, such as soccer's World Cup, or because of an advertising campaign. Management does not know how many customers the site will attract during the World Cup games. Some market analysts estimate that the number of visitors varies from game to game, depending on which teams are playing. However, they expect about 30 to 40 million visitors during the two-hour period right before the final game.

Developing a detailed model to calculate that the proposed system can support 5,555 visitors per second may be overkill. Simply knowing that the site can serve approximately 1,000 visitors/s for one alternative or 8,000 visitors/s for another alternative is the right level of information to select one option over another. Consider the following example of bounding analysis. The e-business search function requires 0.005 s of disk I/O on average, and disk I/O at the database server is the bottleneck for this type of transaction. Then, according to the bounding analysis models, the maximum e-business service throughput is 1/(service demand at the bottleneck resource) = 1/0.005 s = 200 transactions/s. Suppose that 2 percent of the search requests generate a sale and that each sale generates an average of $25. Thus, the upper bound on "revenue throughput" is $100/s. Managers might find this type of metric more meaningful, and it gives them an indication on how the IT infrastructure can limit business revenue.

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

*Virgilio A.F. Almeida is a professor of computer science at the Federal University of Minas Gerais (UFMG), Brazil. Contact him at virgilio@dcc.ufmg.br.*

*Daniel A. Menascé is a professor of computer science at George Mason University, director of the MS in e-commerce program, and codirector of its E-Center for E-Business. Contact him at menasce@cs.gmu.edu.*