



# Planning for Growth

**Authors:** High Volume Web Site Team  
([ibm.com/websphere/developer/zones/hvws](http://ibm.com/websphere/developer/zones/hvws))

**Contact:** Willy Chiu e-mail: [wchiu@us.ibm.com](mailto:wchiu@us.ibm.com)

**Date:** November 17, 2000

**Status:** Version 1.0e

***Abstract:*** This paper discusses a methodology for modeling your high-volume Web site so that proposed changes can be analyzed to determine how performance objectives can best be met.

<b>Executive summary</b>	3
<b>Introducing a methodology for capacity planning</b>	4
<b>Step 1. Identify your workload pattern</b>	5
<b>Step 2. Measure performance of current site</b>	6
<b>Understand workload metrics</b>	7
<b>Obtain site measurements</b>	11
<b>Step 3. Analyze trends and set performance objectives</b>	14
<b>Step 4. Model your infrastructure alternatives</b>	17
<b>Summary</b>	20
<b>References</b>	20
<b>IBM product resources</b>	21
<b>Appendix A.</b>	22
<b>Workload patterns and Web site classifications</b>	22

**Contributors:** The High-Volume Web Site team is grateful to the major contributors to this article: Jerry Cuomo, Alan Emery, Larry Hsiung, Mike Ignatowski, Zhen Liu, Mark Squillante, Noshir Wadia, Cathy Xia, and Li Zhang.

### **Notice**

The following are trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM®, MQSeries®, WebSphere®

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

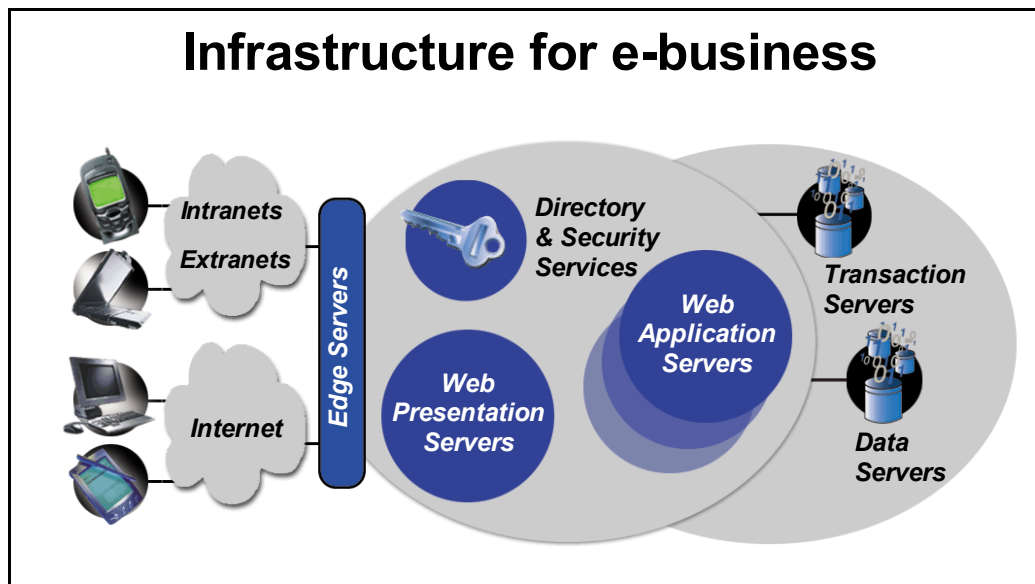
### **Special Notice**

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

## Executive summary

As e-business and its related requirements grow at “Web speed”, a critical issue is whether the IT infrastructure supporting the Web sites has what it needs to provide available, scalable, fast, and efficient access to the company’s information, products, and services. More than ever, CIOs and their teams struggle with the challenges to minimize downtime and network bottlenecks and maximize the use of the hardware and software that comprises their e-business infrastructure.

The IT infrastructure supporting most high-volume Web sites (HVWSs) typically has multiple layers of machines, frequently called *tiers*, and each tier handles a particular set of functions, such as serving content (Web presentation servers), providing integration business logic (Web application servers), or processing database transactions (transaction and database servers). Figure 1 shows an e-business infrastructure comprised of several tiers. Each tier consists of multiple machines, from two to hundreds, to provide capacity and availability for the functions running at that tier. The IBM WebSphere software platform for e-business includes edge servers, Web application servers, development and deployment tools, and Web applications.



**Figure 1. Multi-tier infrastructure for e-business**

Even with this growing complexity, typical IT infrastructures can be analyzed and related models developed to assist in predicting and planning how to meet future requirements. This paper presents a methodology for IT professionals to use to determine whether their Web site can satisfy future demands and to evaluate potential workload and infrastructure changes. It also introduces the concept of configuring a Web site based on an analysis of how different components combine to best meet the performance objectives of your particular workload pattern, potentially reducing the costs of prototyping and stress testing.

IBM’s IT experts have been working with many IBM customers to analyze large Web sites and help customers implement scalable Web sites [1]. Some of these customers are already working with IBM to exploit and further the technologies for capacity planning being developed. Our methodology is based on this on-going research [4,5,6,7,8,9], as well as IBM’s patterns for e-business.

# Introducing a methodology for capacity planning

Our methodology for capacity planning is based on our analysis of many large Web sites, including IBM's, and continuing engagements with large customers seeking to improve site performance, accurately project workloads, and make infrastructure changes that will satisfy future requirements. The methodology consists of four steps:

1. Identify your workload pattern
2. Measure performance of current site
3. Analyze trends and set performance objectives
4. Model your infrastructure alternatives

The steps are introduced below, then described in more detail throughout the rest of the paper. These steps are useful whether you are considering changes to a current site or planning a new site. The paper focuses on technologies for capacity planning. Implementing such technologies will affect your IT organizations, processes, and people; the related implications are not addressed in this paper.

**1. Identify your workload pattern.** Your workload is assumed to be high-volume and growing, serving dynamic data, and processing transactions. Beyond that, you must consider other characteristics, such as transaction complexity, data volatility, and security. After your analysis, it becomes clear that your workload pattern fits into one of five classifications: *publish/subscribe*, *online shopping*, *customer self-service*, *trading*, or *business-to-business*. Correctly identifying your workload pattern assures the best results from the remaining steps and maximizes your site's chances for satisfying future requirements.

**2. Measure performance of current site.** As always, you must understand the present before planning the future. You need to measure these site characteristics: volumes (hits, page views, transactions, searches), arrival rates, response times by class, user session time, number of concurrent users, and processor and disk utilization. If you're planning a new site, you'll need to estimate these metrics; IBM's high-volume Web site team can provide typical site profiles.

**3. Analyze trends and set performance objectives.** Your workload is growing and your current metrics, no matter how good they are, must improve, along with the capacities of the hardware and software that comprise your infrastructure. In this step, you analyze trends to determine future peak volumes, then set objectives for each metric identified in Step 2, along with any new metric that applies to your future requirements.

**4. Model your infrastructure alternatives.** At this point, you are ready to determine the components needed to construct your site's infrastructure. IBM can help you match components to the particular requirements and objectives of your workload pattern.

Our methodology for capacity planning complements IBM's patterns for e-business (see [Patterns for e-business](#)), in that each high volume Web sites (HVWS) workload pattern can be mapped to one of the e-business patterns for site design. Regardless of the methodology you used to design your site, our capacity planning methodology can complement that effort and establish a foundation for managing your capacity requirements.

## Step 1. Identify your workload pattern

Your workload is assumed to be high-volume and growing, serving dynamic data, and processing transactions. Beyond that, you must consider other characteristics, such as transaction complexity, data volatility, security, and others. After your analysis, it becomes clear that your workload pattern fits into one of five classifications: *publish/subscribe*, *online shopping*, *customer self-service*, *trading*, or *business-to-business*. Correctly identifying your workload pattern assures the best results from the remaining steps and maximizes your site's chances for satisfying future requirements.

Review the workload pattern descriptions below to identify your workload pattern. You may also want to refer to Table 1 in Appendix A.

**Publish/subscribe Web sites provide users with information.** Sample publish/subscribe sites include search engines, media sites, such as newspapers and magazines, and event sites, such as those for the Olympics and for the championships at Wimbledon. Site content changes frequently, driving changes to page layouts. While search traffic is low in volume, the number of unique items sought is high resulting in the largest number of page views of all site types. As an example, the Sydney Olympics site successfully handled a peak volume of 1.2 million hits per minute using IBM's WebSphere Application Server, WebSphere Commerce Suite, and MQSeries. Security considerations are minor compared to other site types. Data volatility is low. This site type processes the fewest transactions and has little or no connection to any legacy systems.

**Online shopping sites let users browse and buy.** Sample sites include typical retail sites where users buy books, clothes, and even cars. Site content can be relatively static, such as a parts catalog, or dynamic where items are frequently added and deleted as, for example, promotions and special discounts come and go. Search traffic is heavier than the publish/subscribe site, though the number of unique items sought is not as large. Data volatility is low. Transaction traffic is moderate to high, and almost always grows. The typical daily volumes for many large retail customers, running on IBM's WebSphere Commerce Suite, range from less than one million hits per day to over 13 million hits per day, and with a range from 100,000 transactions per day to three million transactions per day in the top range; of the total transactions, typically between 1% and 5% are buy transactions. When users buy, security requirements become significant and include privacy, nonrepudiation, integrity, authentication, and regulations. Shopping sites have more connections to legacy systems, such as fulfillment systems, than the publish/subscribe sites, but generally less than the other site types.

**Customer self-service sites let users help themselves.** Sample sites include banking from home, tracking packages, and making travel arrangements. Data comes largely from legacy applications and often comes from multiple sources, thereby exposing data consistency. Security considerations are significant for home banking and purchasing travel services, less so for other uses. Search traffic is low volume; transaction traffic is low to moderate, but growing.

**Trading sites let users buy and sell.** Of all site types, trading sites have the most volatile content, the highest transaction volumes (with significant swing), the most complex transactions, and are extremely time sensitive. Products like IBM's WebSphere's Application Server play a key role at these sites. Trading sites are tightly connected to the legacy systems, for example, using IBM's MQSeries for connectivity. Nearly all transactions interact with the back end servers. Security considerations are high, equivalent to online shopping, with an even larger number of secure pages. Search traffic is low volume.

**Business-to-business sites let businesses buy from and sell to each other.** Data comes largely from legacy applications and often comes from multiple sources, thereby exposing data consistency. Security

requirements are equivalent to online shopping. Transaction volume is moderate, but growing; transactions are typically complex, connecting multiple suppliers and distributors. There are two styles of this pattern:

1. *Business-to-business integration*: This style includes programmatic links between arms-length businesses (where a trading partner agreement might be appropriate). Example: supply chain management.
2. *eMarketplace or B2M2B*: The M represents the eMarketplace, which supports multiple buyers and suppliers. The buying function can be performed online or programmatically. Example: e-Marketplace.

## **Step 2. Measure performance of current site**

As always, you must understand the present before planning the future. You need to measure these site characteristics: volumes (hits, page views, transactions, searches), arrival rates, response times by class, user session time, number of concurrent users, and processor and disk utilization. If you're planning a new site, IBM has site profiles you can use to estimate these metrics.

Our analysis of the performance of e-business infrastructures under various workload patterns demonstrates that workload pattern complexities (for example, bursty arrival patterns) can significantly affect resource demands, throughput, and the latency encountered by user requests, in terms of higher average response times and higher response time variance. Without adaptive, optimal management and control of resources, service level agreements (SLAs) based on response time are impossible. The capacity requirements on the site are increased while its ability to provide acceptable levels of performance and availability diminishes.

When analyzing your current site, do not overlook the design of your Web pages. IBM's work to-date suggests there are many practices that when followed reduce the time it takes to download a Web page. Web pages have common components and characteristics, such as page size and number of items, that can and should be managed with an eye toward minimizing download time. Doing the "right" thing will not always be possible, and some components or characteristics may be outside of the control of the page designer. Still, everyone with an interest in the site's performance should understand these factors and their related tradeoffs. Figure 2 summarizes page design metrics from 15 different Web sites; the metrics vary but strongly suggest that page design is an important performance component that when managed well can improve a site's capacity. Metrics considered good or excellent are shaded green; less favorable metrics are shaded with yellow, and unacceptable metrics are shaded with red. For more information, see [Design Pages for Performance](#) [3]. The IBM WebSphere Studio Page Detailer component is a tool that can help identify the types of information shown in Figure 2, but also graphically illustrates how these components can affect page responsiveness.

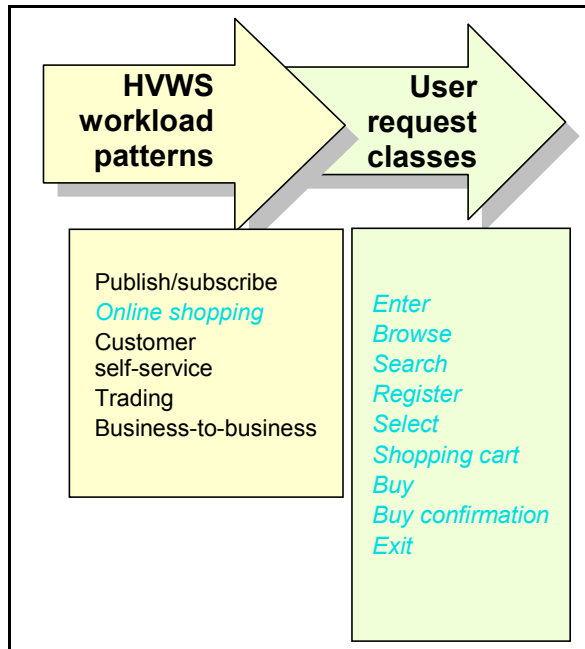
Example of Web page metrics						
Web page	Page load time (sec)	Page size (bytes)	Number of items	Number of connections	Number of servers	Failed connections
1	32.33	179,968	51	17	2	0
2	30.5	140,842	80	7	2	0
3	31.78	136,943	25	6	1	0
4	26.26	122,146	53	7	1	0
5	78.26	121,664	56	21	3	0
6	41.648	111,281	37	5	2	0
7	34.45	105,433	35	21	2	0
8	22.18	93,580	29	6	1	0
9	22.52	84,240	46	46	1	0
10	27.03	72,411	36	36	4	0
11	19.951	64,347	30	19	1	0
12	29.741	61,073	40	11	1	0
13	15.14	56,430	25	5	1	0
14	15.69	43,891	23	23	1	0
15	8.77	39,189	12	5	2	0

**Figure 2. Examples of Web page metrics**

### Understand workload metrics

Correctly identifying your workload pattern prepares you for measuring and understanding the complexities of your site. Each workload pattern has an associated class of user requests. Figure 3 shows an example of classes of user requests associated with the online shopping workload pattern.

Each class is characterized by how the requests arrive at the Web site and the resources required to satisfy the request. The major factors affecting arrivals include standard (marginal) distribution, dependence structure and seasonality. In general, IBM's analysis demonstrates complex behaviors that include light-tailed and heavy-tailed distributions, short-range and long-range dependencies, strong seasonality and periodicity, and geographic effects. The typical assumption of independent exponential interarrivals of Web requests does not hold true under these conditions and one has to solve the problem with nontraditional assumptions that require complex mathematical algorithms. IBM's mathematical research of these characteristics has enabled the development of improved models to understand and predict the impacts of these relationships and behaviors.

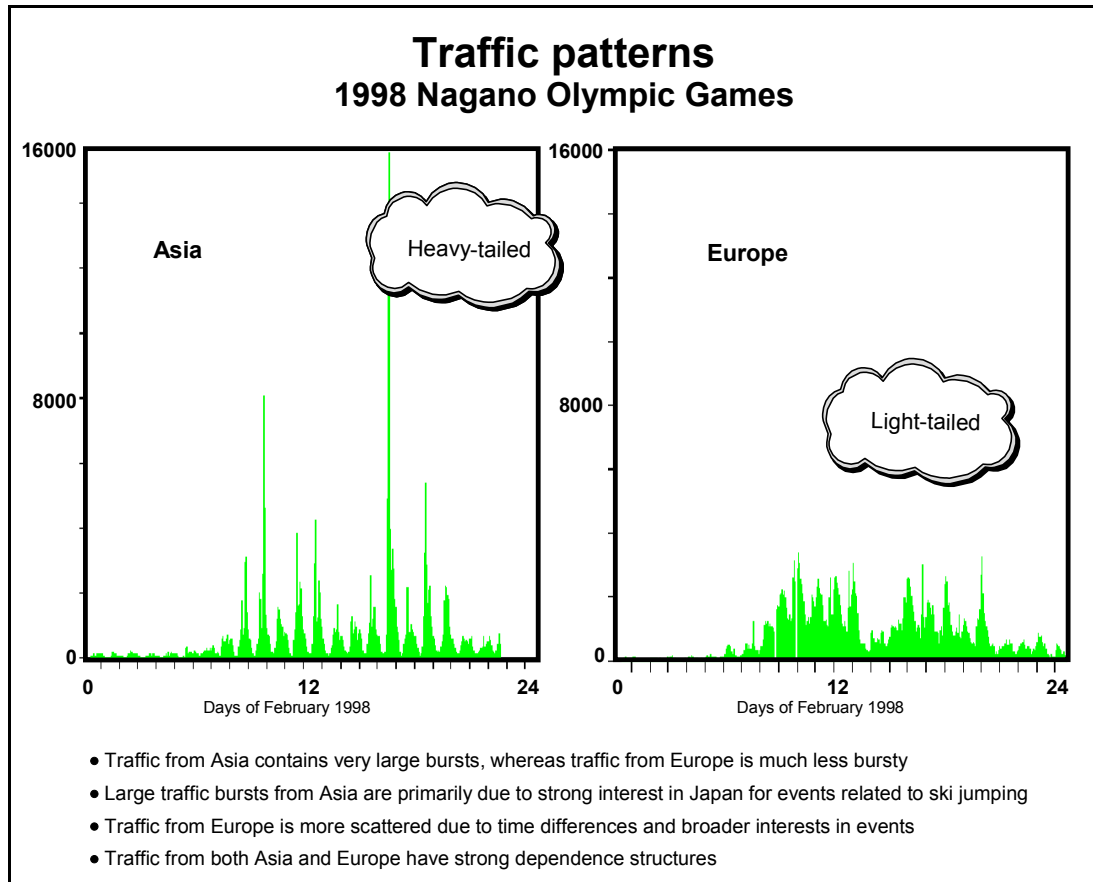


**Figure 3. Each workload pattern has an associated class of user requests**

**Distribution and dependence.** Web traffic exhibits bursty, heavy-tailed, and correlated arrival patterns. Bursts refer to the random arrival of requests, with peak rates far exceeding the average rates. These bursts are caused by unpredictable events such as major stock market swings or special events such as Christmas or Valentine's day. Such events yield dependencies among requests (for example, larger bursts tend to occur in close proximity), heavy-tail distributions (for example, very high variability in the sizes of the bursts), and the combination of dependencies and heavy-tail distributions. A heavy-tailed distribution for a random variable is one where the tail of the distribution decreases subexponentially. For these distributions, the probability that a large value occurs is nonnegligible. The batch arrival process exhibits such heavy-tailed behavior and the batch request sizes tend to be strongly correlated. A practical consequence of burstiness and heavy-tailed and correlated arrivals is difficulty in capacity planning.

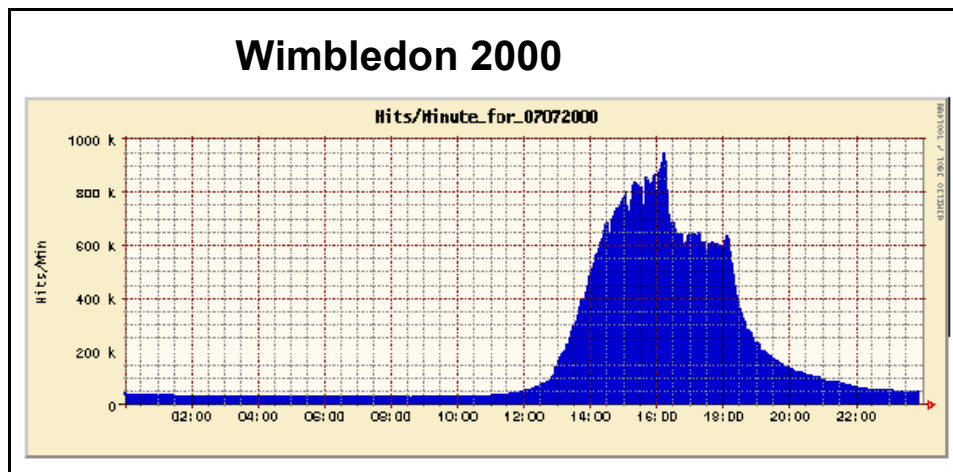
Burstiness and wide-ranging hit rates are among the most obvious workload pattern complexities that affect Web site performance and availability. In traditional models, requests are independent and the variance in the burst sizes are relatively small. These distributions belong to the class of light-tailed distributions. The burstiness of a HVWS yields heavy-tailed distributions and a strong dependence structure. This is illustrated by the traffic patterns from the 1998 Nagano Olympic games, as shown in Figure 4. Such bursts of requests are triggered by some special event, for example, in this case when Japan won the Gold medal for ski jumping in Nagano. Contrast the heavy-tailed distribution from Asia with the light-tailed distribution on the same day from Europe. Further, note the dependence structure from day to day, as well as within each day, at both locations.





**Figure 4. Traffic patterns from Nagano Olympic Games**

IBM's Wimbledon 2000 Web site also exhibited extreme bursts on its busiest day, 7 July 2000. Figure 5 graphs the record-breaking site traffic on that day when peak hits per minute reached 963,948 and peak hits per day totalled 281,605,872).



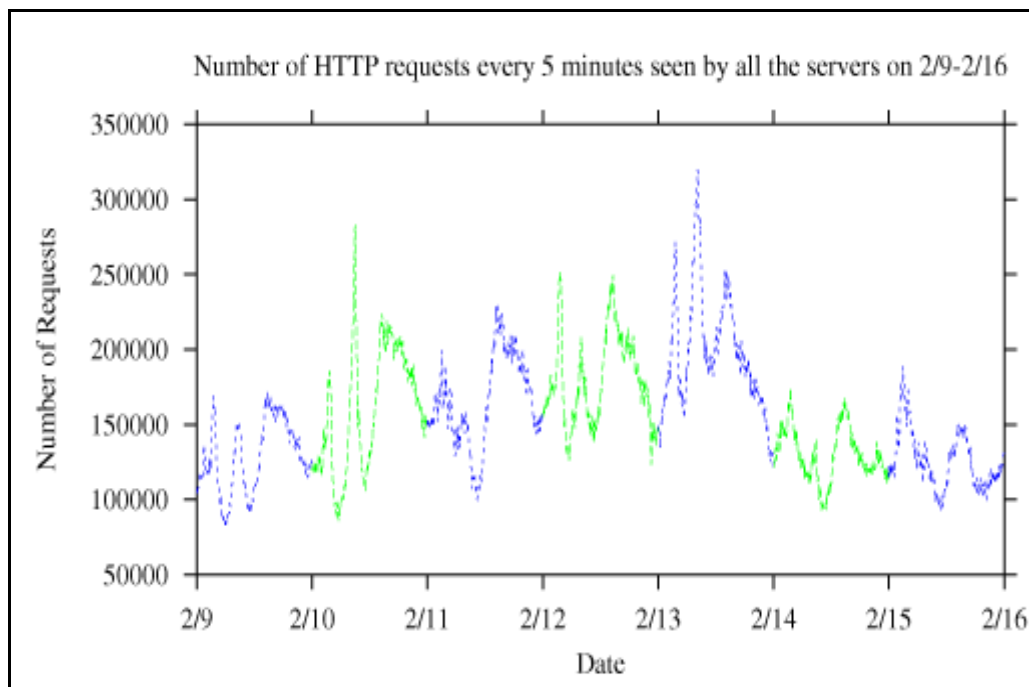
**Figure 5. IBM's Wimbledon Web site on its record-breaking day**

The foregoing non-traditional request traffic stresses the Web server. Bursty traffic with heavy-tailed distributions degrades the performance by several orders of magnitude over light-tailed distributions. For heavy-tailed distributions, the extremely large bursts occur relatively more frequently than the light-tailed model. Moreover, the dependence structure causes these bursts to occur in close proximity to each other. With such input traffic characteristics, the performance measures, in particular, the response time, have similar characteristics as the input traffic. This helps to explain why some sports and e-business Web sites are more difficult to maintain than relatively simple Web sites (for example, a university Web site serving only static content).

With respect to SLAs, a more powerful set of servers is needed to achieve the same level of service for heavy-tailed distributions in comparison with the case of independent light-tailed request traffic. To guarantee good performance, we need to focus on peak traffic duration because it is the huge bursts of requests that most degrade performance. That is why some busy sites require more “head room” (spare capacity) to handle the volumes; for example, a high-volume online trading site reserves spare capacity with a ratio of three to one.

**Seasonality.** Seasonality refers to the periodicity of the request patterns. Seasonal traffic is most often represented by the regular daily activities of the users of a Web site. For example, traffic to some e-trading Web sites has consistent peaks and valleys each day when the market opens and closes. Seasonal traffic is also observed in monthly intervals, for example, when users pay bills at the end of the month, and during designated periods, for example, the holiday season.

Figure 6 shows an example of seasonal traffic from the Nagano Olympic Web site. The figure plots the number of requests received every five minutes by all servers from Monday 9 February through Sunday 16 February. While each daily cycle varies considerably, note that each day has three peaks and that overall traffic intensity increases each weekday then decreases on the weekend. These patterns repeated each week, demonstrating seasonal variations that correspond to weekly cycles.



**Figure 6. Example of seasonality demonstrated by one week from Nagano**

Seasonal requests can degrade the performance of the Web server because, for the peak duration, large batches of requests occur around the same time. The central questions are how high is the peak and how long is the peak duration. The answers to these two questions can have a significant impact on how powerful the Web server should be in order to handle a specific SLA. To satisfactorily handle request traffic, the capacity of the Web server should be close to peak request level, with some “head room” to allow for unexpected growth.

Other factors that define the workload pattern include the volume of page views and transactions, the volume and type of searches, the complexity of transactions, the volatility of the data, and security considerations.

The rest of this section introduces techniques available to obtain the measurements you need to complete your capacity plan.

### Obtain site measurements

Each workload pattern requires specific measurements. Figure 7 is an example of some current measurements of an online shopping site.

Measurement	Today
Concurrent users	40,000
Hits/second	3,480
Response time in seconds	28
Pages/second	346
Pages/visit	10.6
Visits/second	32.6
Minutes/visit/user	20
Ratio of user visit type	92% browse only 6% browse/search 2% buy

**Figure 7. Example of some current measurements of an online shopping site**

By analyzing typical user visits, it’s possible to create probabilities about future user visits. Online shoppers, for example, typically browse, may search, and occasionally buy. You can develop various scripts to describe user visits. Figure 8 contains samples of scripts for online shopping, online banking, and online trading.

Online shopping script	
<b>Browse</b>	Home page Choose department (static HTML) Choose category Choose subcategory Choose product 1 Choose product 2 Choose department (dynamic category display) Choose category Choose subcategory Choose product 1 Choose product 2
<b>Search</b>	Home page Select product search Submit keyword Select new search Submit keyword
<b>Buy</b>	Home page Select "AtHome" department Select "Candles" category Select "Scented" subcategory Select "tripod candle" Select "Add to shopping bag" Select "Checkout" Select "Complete order online" Select "Charge it"

**Figure 8. (Part 1 of 3) Example of a script for online shopping**

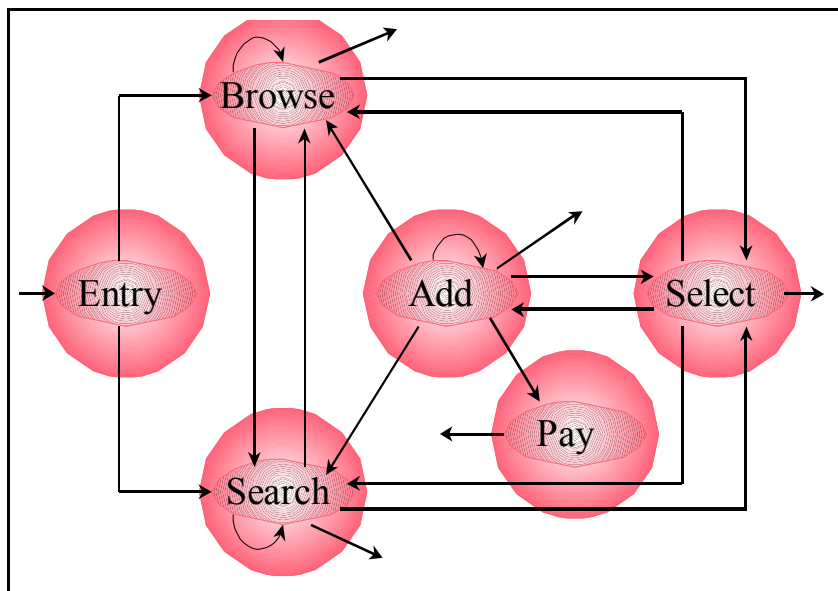
Online banking script	
	Login Force PIN change Main menu Add a payee Schedule 6 bill multi-payment Edit a payment Customize Financial summary Account details Request a check copy Verify check copy request Sign-off

**Figure 8. (Part 2 of 3) Example of a script for online banking**

Online trading script	
	Login
	Query position
	Get quote 1
	Get quote 2
	Get quote 3
	Get quote 4
	Get quote 5
	Trade - buy
	Check status
	Get quote 6
	Get quote 7
	Get quote 8
	Trade - Sell
	Check status
	Logoff

**Figure 8. (Part 3 of 3) Example of a script for online trading**

Using the scripts and the data from your measurements, you can create what is called a *transition matrix*. Figure 9 is an example of a transition matrix for an online shopping visit. Viewing the sample transition matrix as it relates to the sample script above, you can easily see the browse and search requests; the buy request occurs when the user decides to *add* (to shopping bag) and *pay*.



**Figure 9. Example of a transition matrix for an online shopping visit**

### Step 3. Analyze trends and set performance objectives

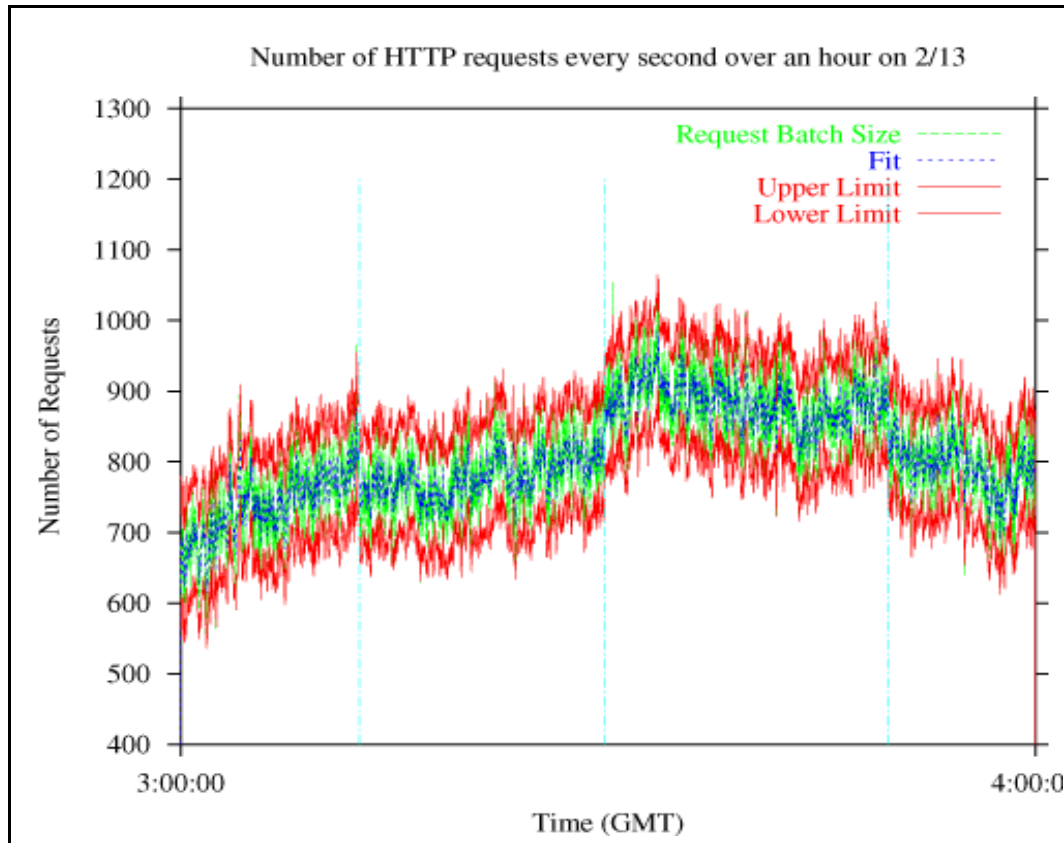
Your workload is growing and your current metrics, no matter how good they are, must improve, along with the capacities of the hardware and software that comprise your infrastructure. In this step, you analyze trends to determine the characteristics of future peak volumes, then set objectives for each metric identified in Step 2, along with any new metric that applies to your future requirements. Figure 10 is an example of current and projected measurements for an online shopping site. Performance objectives are usually driven by business objectives, for example, to improve response time to preferred customers.

	Today	Projections
Concurrent users	40,000	100,000
Hits/second	3,480	8,700
Response time in seconds	28	<10
Pages/second	346	865
Pages/visit	10.6	10.6
Visits/second	32.6	81.6
Minutes/visit/user	20	20
Ratio of user visit type	92% browse only 6% browse/search 2% buy	92% browse only 6% browse/search 2% buy

**Figure 10. Example of current and projected measurements for an online shopping site**

The ability to accurately forecast request patterns is an important requirement of capacity planning. Our forecasting methodology is based in part on constructing a set of mathematical methods and models that isolate and characterize the trends, interdependencies, seasonal effects, randomness, and other key behavior in the Web site's request patterns [4,6,7,8]. This includes, for example, the use of piece wise autoregressive moving average (ARIMA) processes combined with heavy-tailed distributions. Figure 11 is an example of a request pattern we would consider for our models; it shows the number of requests received per second over the period of one hour at the Nagano Olympic site. The curve fitted to these measurements is shown in the middle of the data points.

Since each of the key traffic characteristics can scale differently, we use a general set of mathematical methods to estimate the intensity of each characteristic in future request patterns and to scale each characteristic to its forecasted intensity [4,7,8]. We then combine these scaled mathematical models to characterize and predict request patterns. By using our mathematical methods to isolate, characterize, and forecast the trends, interdependencies, seasonal effects, randomness, and other key behavior in the request patterns, we have developed a general methodology that provides a more accurate and effective approach for predicting request patterns than other approaches being used today.



**Figure 11. Requests per second over one hour during Nagano Olympics**

Our methodology has proven effective in practice, having been used to predict request patterns of actual sites over both short-term and long-term time frames. In particular, we used our methodology to predict the peak hour request volumes for a recent sporting event Web site hosted by IBM based on request patterns from the Web site in three previous years, as shown in Figure 12, together with 95% confidence intervals. The arrows illustrate one of our approaches, namely a first-order rate-of-change method, for estimating the traffic intensity scaling factor from year to year. Note the exponential growth seen as you go from 1997 to 2000. We then used the forecasted scaling factor for 2000 and our methodology to scale the peak hour traffic model from 1999 to obtain our forecasted peak hour traffic model for 2000. Our forecasts were found to be in excellent agreement with the site's actual peak volumes. Moreover, these methods have been applied with equal success to estimate request patterns for upcoming seasonal events, such as the Christmas online shopping rush. We also used our methodology to forecast request patterns over short-term time frames (days, weeks); again, our forecasts were found to be in excellent agreement with the site's actual request patterns.

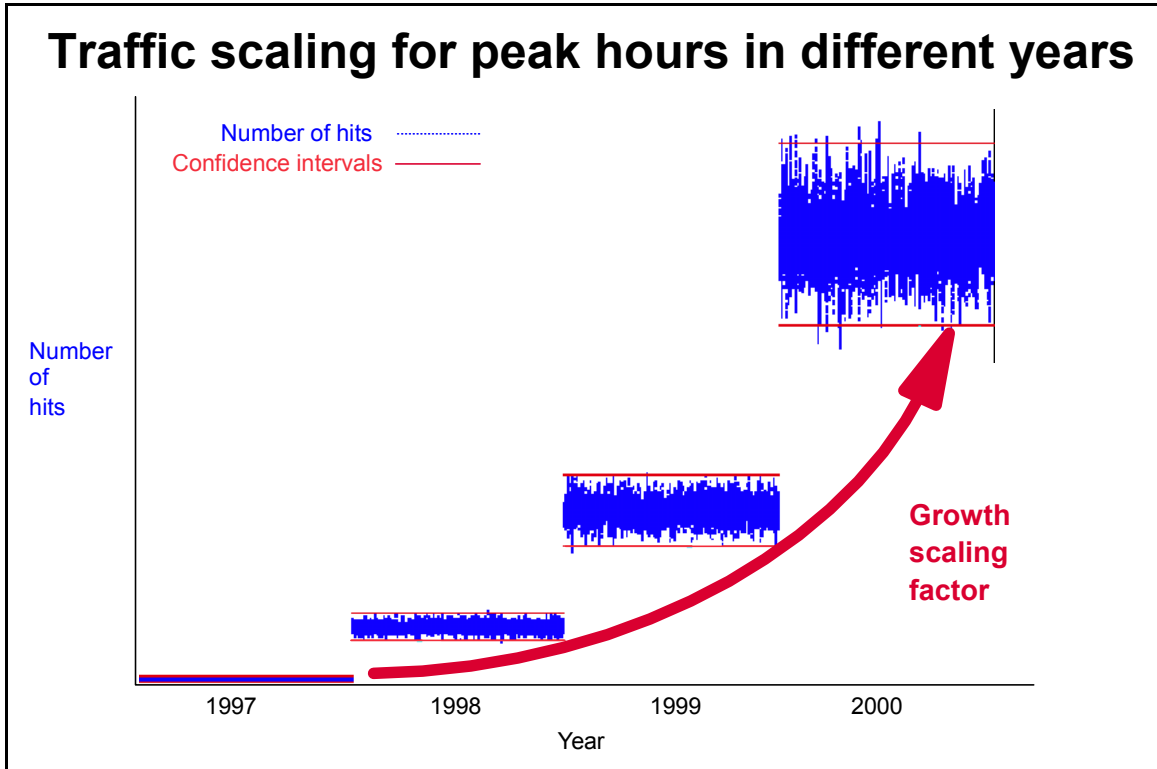


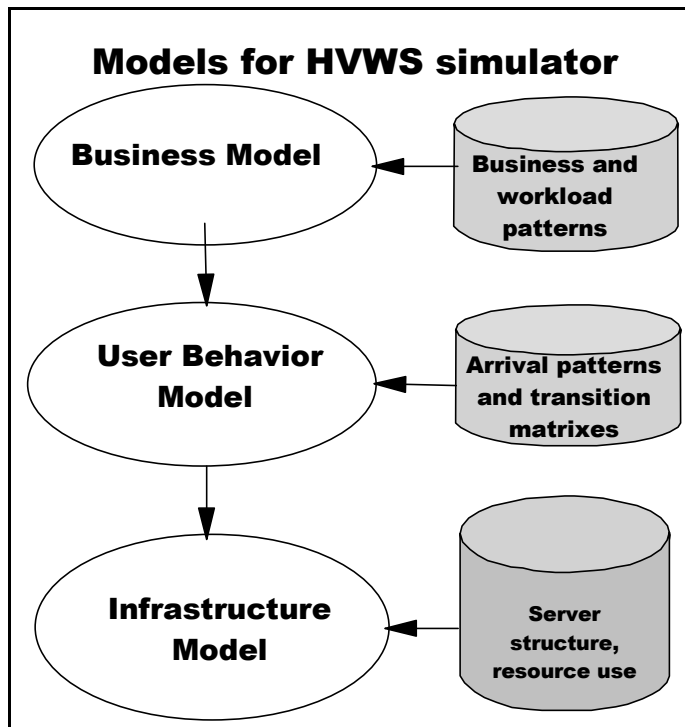
Figure 12. Traffic scaling for peak hours in different years



## Step 4. Model your infrastructure alternatives

At this point, you are ready to determine the components needed to construct your site's infrastructure. IBM can help you match components to the particular requirements and objectives of your workload pattern.

The technologies IBM is developing for HVWS capacity planning rely on the three models depicted in Figure 13.

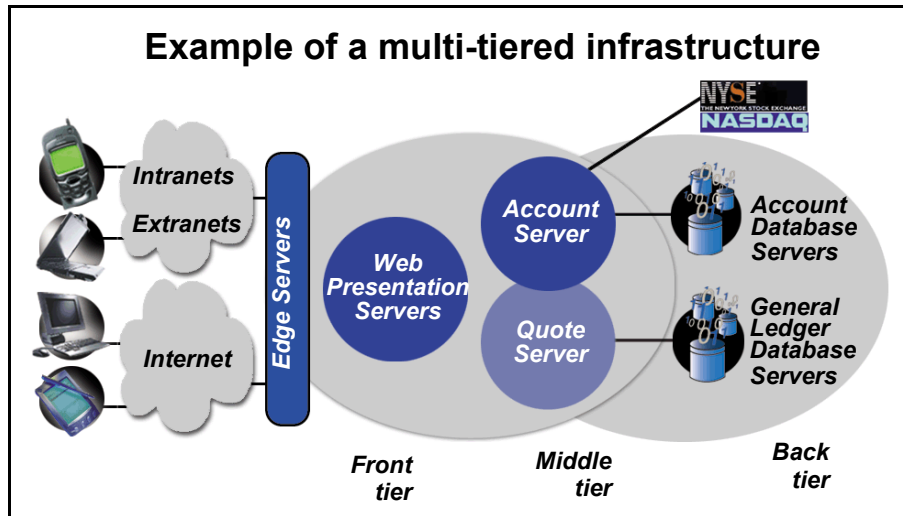


**Figure 13. Models for HVWS capacity planning**

The *business model*, or business usage model, defines the e-business pattern and workload pattern. There is a *user behavior model* for each workload pattern. Each workload pattern consists of several classes of user requests. The arrival patterns and routes (transition matrices) site visitors follow for each class comprise the user behavior model. The hardware and software resources, and the amount of each required to satisfy each class of user requests, comprise the *infrastructure model*.

The infrastructure model processes browse, search, and buy transactions. The model assumes:

- The Web site has multiple layers of machines, or tiers, each handling a particular set of functions, such as the site depicted in Figure 14 (figure does not include firewall layers).
- A load-balancer, such as the network dispatcher, routes requests to multiple Web front-end servers using an algorithm to distribute requests evenly among the servers.
- The front-end Web servers handle requests for static or dynamic pages.
- The Web application server processes business logic for the transactions initiated by the request. In Figure 14, the account and quote servers are the application servers.
- The back-end database server handles requests for dynamic pages that involve obtaining or updating information; such requests do not return through the load balancer.

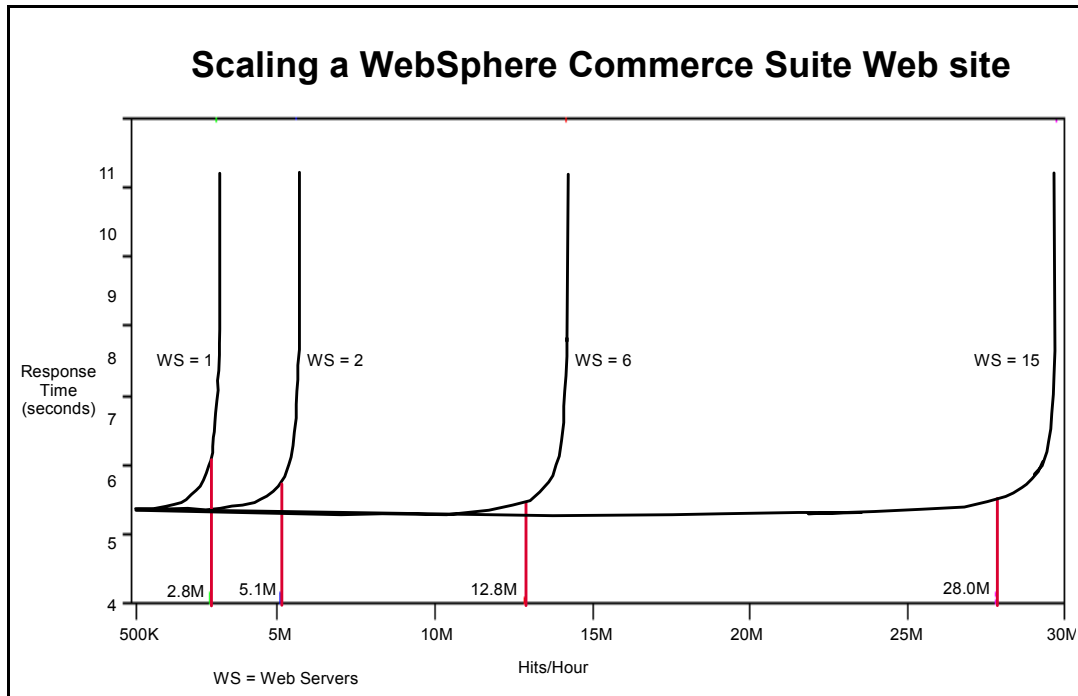


**Figure 14. A Web site with multiple tiers**

We formulate a class of queuing networks to model multi-tier architectures in order to analyze performance at different levels. We further derive a variety of solutions to these models under different input traffic patterns and at different time-scales. This family of mathematical models and solutions are general enough to abstract the underlying hardware and software details, but detailed enough to produce meaningful performance results. We consider the queuing system, where the resources of each tier is modeled by multiserver queues that have specific relationships to one another. These relationships are determined through measured or estimated workload characteristics. We then solve the performance/capacity problem against a set of user requirements, such as the number of concurrent users, response time, or throughput rate. We have also developed unique formulas to allow us to estimate the behavior of the system where peak demands are significantly higher than average demand, and there is a nonnegligible probability of accessing large amounts of data by the user. Our method is flexible enough to model horizontal and vertical scaling, or a combination, depending on user requirements and workload characteristics. For example, we can increase the number of Web servers by adding another server or by adding another processor on a given server. Given the appropriate Web site and workload data, we are then able to obtain performance estimates from our performance models and analysis.

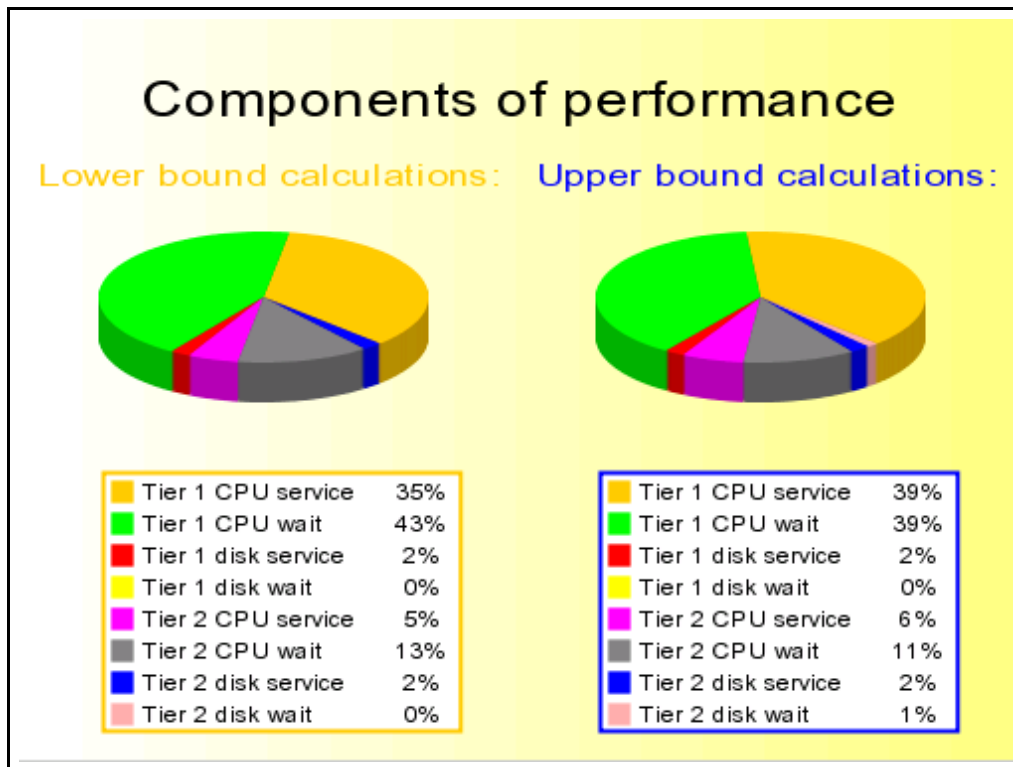
We have developed capacity plans for a number of IBM-hosted Web sites. Figure 15 depicts one such site and reflects the process of calibrating our model using current data from the site, then developing projections based on current data, trends, and objectives. In Figure 15, the first three response time curves reflect the current data used to calibrate the model as discussed in Step 2. By analyzing current metrics and component information, we are able to project the fourth curve.

Referring to Figure 15, the results show that when the request traffic is light, one front-end server is enough to handle the load. As the traffic increases, the response time curve remains flat until the front-end CPU reaches a utilization of 90% (2.8 million hits/hour). At this point, a minor increase in the load can rapidly plunge the system into a situation resembling deadlock, where the front-end server attempts to serve more and more files at slower and slower speeds, such that few are experiencing satisfactory response times. This means that the front-end server has become the bottleneck. We therefore need to add a front-end server, and upon doing so, the front-end CPU utilization drops as desired. The response time curve becomes flat until the front-end CPU again reaches a utilization of 90% (5.1 million hits/hour), when we need to consider adding another front-end server. The back-end server becomes the bottleneck only after about 15 front-end servers are handling around 28 million hits/hour.



**Figure 15. Scaling a WebSphere Commerce Web site**

Figure 16 is a sample of a graph we produce when analyzing performance objectives against specific hardware and software components.



**Figure 16. Sample graph showing components of performance.**

## Summary

The challenge of effective capacity planning for high-volume Web sites is an awesome one, but not insurmountable. The methodology suggested by this paper offers a road map toward understanding your workload pattern and current metrics, analyzing trends and setting objectives for the future, and, finally, selecting the IT infrastructure components needed to meet your performance objectives. The ability to analyze your site's requirements in the context of your particular workload pattern can contribute greatly to making the right selections. IBM's HVWS team is available to assist you in the application of these steps and the development of appropriate models for your environment. And IBM's WebSphere and MQSeries products, as evidenced by their extraordinary performance at the Sydney Olympics, are significant parts of IBM's solution for rapidly growing e-business infrastructures.

Of course, the subject of capacity planning is an ongoing study. Increasingly valuable information is available [10], as well as exciting new offerings from IBM, such as the Capacity Advantage tool and "capacity on demand" options that will greatly enhance the ability to respond to traffic growth. With an eye toward discovering and documenting modern design practices that allow ever greater capacities and scalability, IBM's HVWS group is refining its methodology, developing tools that embody that methodology, fine tuning its mathematical methods, and looking at the additional challenges presented by such areas as network caching and the fast-growing business-to-business workloads. The promise remains great, however, of meeting future needs while planning effectively and reducing costs.

## References

1. IBM High-Volume Web Sites, [Design for Scalability](http://ibm.com/websphere/developer/zones/hvws/library), December 1999 (ibm.com/websphere/developer/zones/hvws/library)
2. IBM High-Volume Web Sites, [Web Site Personalization](http://ibm.com/websphere/developer/zones/hvws/library), February 2000 (ibm.com/websphere/developer/zones/hvws/library)
3. IBM High-Volume Web Sites, [Design Pages for Performance](http://ibm.com/websphere/developer/zones/hvws/library/), May 2000 (ibm.com/websphere/developer/zones/hvws/library/)
4. A.K. Iyengar, M.S. Squillante, L. Zhang. *Analysis and characterization of large-scale web server access patterns and performance*. World Wide Web, Vol. 2, 1999.
5. W.N. Mills, M.S. Squillante, C.H. Xia, Li. Zhang. *Web workload service requirement analysis: A queueing network approach*. IBM Research Report, 2000.
6. Z. Liu, N. Niclausse, C. Jalpa-Villanueva. Web traffic modeling and performance comparison between HTTP1.0 and HTTP1.1. *System Performance Evaluation: Methodologies and implications*, pp. 177-189, 2000.
7. M.S. Squillante, D.D. Yao, L. Zhang. *Internet traffic: Periodicity, tail behavior and performance implications*. *System Performance Evaluation: Methodologies and Applications*, pp. 23-37, 2000.
8. M.S. Squillante, D.D. Yao, L. Zhang. *Web traffic modeling and web server performance analysis*. *IEEE Conference on Decision and Control*, December 1999.
9. D. A. Menasce and V. A. F. Almeida. *Capacity Planning for Web Performance: Metrics, Models, and Methods*, Prentice Hall, 1998.
10. D.A. Menace and V.A.F. Almeida. *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*, Prentice Hall, 2000.

## IBM product resources

- Download a demo version of [PageDetailer](http://www-4.ibm.com/software/webrowsers/studio/download.html), the tool in WebSphere that measures in detail every element in a page download to assist in performance analysis and optimization. See <http://www-4.ibm.com/software/webrowsers/studio/download.html>
- The IBM WebSphere software platform for e-business includes edge servers, Web application servers, development and deployment tools, and Web applications. Find out more at the [WebSphere Developer Domain](http://www7b.boulder.ibm.com/wsdd/) (<http://www7b.boulder.ibm.com/wsdd/>)
- Find out about the IBM software that ran the Sydney Olympics site: [IBM's WebSphere Application Server](http://www-4.ibm.com/software/webrowsers/appserver/) (<http://www-4.ibm.com/software/webrowsers/appserver/>), [WebSphere Commerce Suite](http://www-4.ibm.com/software/webrowsers/commerce/) (<http://www-4.ibm.com/software/webrowsers/commerce/>) and [MQSeries](http://www-4.ibm.com/ts/mqseries/messaging) (<http://www-4.ibm.com/ts/mqseries/messaging>)
- Find out about the [IBM WebSphere Commerce Suite](http://www-4.ibm.com/software/webrowsers/commerce/) (<http://www-4.ibm.com/software/webrowsers/commerce/>), used by customers who run large-scale online shopping sites that we have studied.
- Find out more about the software used by trading sites we've studied: [WebSphere's Application Server](http://www-4.ibm.com/software/webrowsers/appserver/) (<http://www-4.ibm.com/software/webrowsers/appserver/>), and [MQSeries](http://www-4.ibm.com/ts/mqseries/messaging) (<http://www-4.ibm.com/ts/mqseries/messaging>)
- Find out about more [IBM's Capacity Advantage](http://www-1.ibm.com/servers/eserver/introducing/capacity/html) tool (<http://www-1.ibm.com/servers/eserver/introducing/capacity/html>)

## Appendix A.

### Workload patterns and Web site classifications

Site Type → Pattern ↓	Publish / Subscribe	Online Shopping	Customer Self-Service	Trading	Business-to- Business
Categories / Examples	Search engines Media Events	Exact inventory Inexact inventory	Home banking Package tracking Travel arrangements	Online stock trading Auctions	eProcurement
Content	Dynamic change of the layout of a page, based on changes in content, or need  Many page authors and page layout changes frequently  High volume, non user specific access  Fairly static information sources	Catalog either flat (parts catalog) or dynamic (items change frequently, near real time)  Few page authors and page layout changes less frequently  User specific information: user profiles with data mining	Data is in legacy applications  Multiple data sources, requirement for consistency	Extremely time sensitive  High volatility  Multiple suppliers, multiple consumers  Transactions are complex and interact with back end	Data is in legacy applications  Multiple data sources, requirement for consistency  Transactions are complex
Security	Low	Privacy, nonrepudiation, integrity, authentication, regulations	Privacy, nonrepudiation, integrity, authentication, regulations (Banking); Low for others	Privacy, nonrepudiation, integrity, authentication, regulations	Privacy, nonrepudiation, integrity, authentication, regulations
Percent Secure Pages	Low	Medium	Medium	High	Medium
Cross- session Info	No	High	Yes	Yes	Yes
Searches	Structured by category Totally dynamic Low volume	Structured by category Totally dynamic High volume	Structured by category Low volume	Structured by category Low volume	Structured by category Low to moderate volume
Unique Items	High	Low to Medium	Low	Low to Medium	Moderate
Data Volatility	Low	Low	Low	High	Moderate
Volume of transactions	Low	Moderate to High	Moderate to Low	High to Very High (Very large swings in volume)	Moderate to Low
Legacy Integration/ Complexity	Low	Medium	High	High	High
Page Views	High to Very High	Moderate to High	Moderate to Low	Moderate to High	Moderate