Right-Sizing a Web Site

Let's say you've decided to create a Web site for your successful garden equipment company. You've been in business for five years, have established a regional brand for high-quality gardening tools, and have about 12,000 retail customers and 21 wholesale dealers who purchase from you. Based on a marketing report you commissioned, you expect that in the first year your Web site will have about 1,400 visitors a day. The average visitor will look at eight pages, producing about 4 million page views a year. About 10% will purchase something, and the rest will browse to explore prices and products.

However, in peak times (during the months of April, May, June, and December), you expect peak loads of 3,000 customers a day, concentrated during the hours of 9 A.M. to 5 P.M., producing about 375 visitors per hour or 6 per second. During this time, your Web site will have to serve up about 40 screens per second, with most of the content being read from a database of product and price information. Pages must be served up within 2 seconds of a customer click during peak times or customers may lose patience and go elsewhere.

Before you can proceed, there are some questions you will need answered. How many Web servers will your site require? How many CPUs should each server have? How powerful does the site's database server need to be? What kind of connection speed do you need to the Internet? Until recently, finding the answers to questions such as these was often done on a trial-and-error basis. However, hardware and software vendors such as IBM, Microsoft, and Hewlett-Packard have recently developed a number of simulation tools that can help you find the right answers.

IBM's simulator is called the On Demand Performance Advisor (OPERA) (formerly known as the High Volume Web Sites Simulator). OPERA enables
users to estimate the performance and capacity of a Web server based on workload patterns, performance objectives, and specific hardware and software. OPERA has a very easy-to-use interface that includes pre-built workload patterns for various e-commerce applications, such as shopping, banking, brokerage, auction, portfolio, B2B, and reservation systems, that can be modified as necessary based on the user’s own data assumptions. It can provide what-if analyses for various performance parameters such as throughput, response time, resource utilization, number of concurrent users, and page view rate. It also provides special algorithms to address increases in Web traffic during peak usage periods. The simulator includes built-in performance characteristics for various types of hardware (such as IBM, Sun, and HP servers), software, and infrastructure models. OPERA uses an analytic model to generate reports that allow users to assess the adequacy of proposed hardware and software configurations, forecast performance, and graphically identify bottlenecks that might develop.

Users of the simulator have included Charles Schwab, Aetna, Fidelity, Visa, Bank of America, Walmart.com, and eBay, among others. eBay first turned to the simulator in 2002, when it was attempting to cope with dramatic increases in customer demand. In its early years, eBay needed to serve up only about 1 million pages per hour, but as its customer base grew, and the number of page views per hour significantly increased, its original Web site hardware and software became insufficient, creating customer resistance. After running a simulation of its current and likely future workload, eBay decided to rebuild its auction system around IBM’s WebSphere application that integrates a variety of software tools into an integrated Web site design. In the third quarter of 2003, eBay returned to OPERA to examine the performance of its Sell Your Item application on its new three-tier Web server architecture. The simulation enabled eBay to determine both the number of servers (36) and the optimal CPU (the IBM x335, which provided a 30% performance increase due to a higher-speed CPU, increased amount of RAM, and faster bus speed) required to meet current workloads and future growth targets.
OUTSOURCING MAKES SENSE WHEN DIY IS NO BARGAIN

If you're a small or medium-sized retail business and you want to start selling your wares on the Internet, how do you keep up with the Amazons and Lands Ends of the Internet world while not spending a fortune in the process? The answer in the past has typically been DIY (Do It Yourself). Sometimes this works if all you really want to do is take your existing catalog of products and put it on the Web. But what if you want to not only put the catalog online, but also conduct transactions, accept credit cards, and most importantly, integrate Web site sales with your existing inventory and order entry systems? And what if you want customers to be able to return products purchased in the online channel to your physical store channel? Did you know that 34% of all consumers in the U.S. are multi-channel shoppers, who use the telephone, Internet, and retail stores? And did you know that multi-channel shoppers spend over three times as much as single-channel shoppers?

Welcome to the world of multi-channel commerce, the fastest growing form of retail commerce in the United States. You'll read more about multi-channel commerce in Chapter 7 and Chapter 10. In multi-channel commerce, customers are completely free to conduct multi-channel transactions including via phone, e-mail, Web site, and physical store. Multi-channel shoppers buy products at a store and return them via UPS by obtaining a return authorization by e-mail; or buy on the Web, obtain a return number on the phone, and ship to the closest physical store.

When things get this complex, the DIY approach typically fails. It fails for a number of reasons. Small firms typically do not have the expertise or experience in developing backend database and transaction systems required to support customers in this manner. If they attempt these kinds of systems, development times and budgets often soar out of control, and they may end up late to market. Finally, a DIY effort often results in a home-grown patchwork of programs that is difficult to change, poorly documented, and lacks the functionality required. In the meantime, senior management will have spent oceans of time supervising its construction, ignoring the real business of delivering exciting products to consumers.

At this point outsourcing makes a lot of sense. The idea behind outsourcing of multi-channel commerce applications is the same idea that propels the sale of Microsoft Office and other packaged software: Why build it yourself when a world-class group of system designers have already built the programs and you can purchase them at a pittance?

This is what Big Al's, a Toronto-based retailer of aquatic products that operates 20 stores in Canada but does most of its sales online, decided. Since the company opened shop on the Web in 1999, traffic has grown from about 100,000 visitors to about 3.5 million visitors monthly. Unfortunately, its home-built Web transaction system was not up to the task and frequently crashed during busy holiday buying seasons. At that point, Big Al's management decided to outsource to Novator Systems, also based in Toronto.

In the old system, which could barely handle 300 transactions per day, the retailer would print out Web orders, and then re-enter the information into its warehouse management system, which in turn printed a pick ticket for warehouse workers. Once packed, the orders were handed to UPS

(continued)
drivers waiting on site who then re-entered the information into the UPS system. Under the new system, Web orders are automatically shared with the order entry system, the warehouse system, and the shipping system (including automatic entry into UPS's systems). With the new system, Big Al's easily handles 800 to 1,000 orders a day. Management says they could have developed this system in-house, but if they had, they would have never made it to market for the holiday shopping season in 2004.

How much does all this cost? The cost of outsourcing is all over the map and depends on the scope of the operations being outsourced, the complexity of integrating the outsourcer's software with your own, and the volume of transactions hitting the host servers. Some vendors will charge a single upfront fee of anywhere from $50,000 to $1 million, and then a monthly service fee. They will install the software on your servers, or host the software on their servers. Others, such as VCommerce, one of the largest U.S. vendors for multi-channel commerce, emphasize a low entrance fee of $50,000 and then a monthly payment-as-you-go, depending on the volume of transactions and the number of functions being performed. VCommerce is an application service provider providing Internet services. Customers simply connect over the Internet into their existing systems rather than installing any software on their own servers. This translates into additional savings and for small to medium-sized retailers, this makes the most sense.

Are there risks to outsourcing? Yes, for sure. One risk that has kept the DIY model in business for so many years is retailers' fears that they will lose control over their operations, potentially damage their brands if outsider systems fail, and pay far too much for a service they could perform themselves. However, retailers that do their homework (such as by performing a systems analysis as described in Table 4.1) and then carefully choose an outsourcing firm with a good track record can overcome these risks and take advantage of the benefits that outsourcing can bring.


---

graphics, and other data (see Figure 4.5 on page 201). We will look more closely at the variety of e-commerce software available in Section 4.2.

The decision to build a Web site on your own has a number of risks. Given the complexity of features such as shopping carts, credit card authentication and processing, inventory management, and order processing, the costs involved are high as are the risks of doing a poor job. You will be reinventing what other specialized firms have already built, and your staff may face a long, difficult learning curve delaying your entry to market. Your efforts could fail (Albrecht and Gaffney, 1985). On the positive side, you may be better able to build a site that does exactly what you want, and more important, develop the in-house knowledge to allow you to change the site rapidly if necessary due to a changing business environment.

If you choose more expensive site-building packages, you will be purchasing state-of-the-art software that is well tested. You could get to market sooner. However,
INSIGHT

USING AJAX AND FLASH FOR FAST FORMS AND HIGH-SPEED INTERACTIVITY

Ever make a mistake on a Web site form, only to have to start over again after hitting the Submit button? Or use a map Web site that takes forever to deliver the next frame showing the area adjacent to the first map? You probably have. But, have you ever used a Web shopping cart that gathered all of the required information on a single Web page, and where, if you make a mistake such as entering a telephone number in the wrong format, you are asked to correct the entry immediately? Chances are, you have not. But that may soon change.

Most Web sites today still work on the standard Web model: a client computer asks for a Web page, and a Web server delivers the page. This occurs every time you press a Continue or Search button. Today's highly graphical Web pages often contain several hundred kilobytes of data. But even if you have entered only a few lines of information containing a few hundred bytes of information, in the traditional Web client/server model, all the processing is done on a remote server computer, and the client is not much more than a keyboard, screen, and interface. Under this traditional model, entire Web pages of information are transferred across the Web, creating delays for the user and hogging bandwidth on the Web.

But there is a different way to create Web pages. Dubbed "Ajax" (for Asynchronous JavaScript and XML) by one set of aficionados, and more simply "rich Internet applications" (RIA) by others, in this newer model, the client and server work in the background to transfer information immediately as the user enters it, and the server responds immediately, all without the user being aware of the transfer. The result is a smooth, seamless, seemingly continuous user experience. You can get a good feel for this type of application by going to Google Maps. Enter a city, then use the scroll arrows in the upper left corner of the screen. Go North one click, and nearly instantly, the application takes you North; there is no wait for an entirely new map to download. Go to an older map site and try the same action and you will find quite a delay as an entire new map gets downloaded. Other sites using these techniques include Flickr.com (a photo site), Amazon's search engine A9.com, TJMaxx.com, and HomeGoods.com.

How do rich Internet applications work? There are several ways of building rich Internet applications, but they all involve downloading a small program to the client. Ajax and RIA use existing tools to improve the user experience. One method is to download a small JavaScript program to a client computer that has a Java-enabled browser. JavaScript was one of the first client-side languages and technologies with the ability to run a computer program delivered over the Web. This small program carries on a background conversation with the server, retrieving only the information the user needs at whatever interval the application program requires. This method requires that the JavaScript program be capable of running on all target client computers, which may be using any one of several different browsers, and which may or may not be Java-enabled.

A different method is supported by Adobe, which now makes the Flash plug-in, which nearly 98% of U.S. client computers are currently using. In this method, a Flash program is downloaded to the client. This program runs within the Flash Player installed in most browsers. Flash provides a nearly universal client-side solution.

(continued)
Whatever method is chosen, the early results at consumer Web sites are encouraging. During the 2005 holiday shopping season, TJMaxx.com and HomeGoods.com both tested a single Web page shopping cart using RIA methods. About 50% more customers completed the one-page shopping cart (which combined checkout, billing, and shipping pages) compared to a multiple-page shopping cart used earlier. TravelClicks, an iHoteller unit that sells hotel reservation systems to other hotels, recently began licensing a reservation program built using Adobe’s Flex program (a program specifically designed by Adobe to create rich Internet applications). On the roughly 2,000 hotel sites using iHoteller, customers now can see instantly the impact of changing rooms, or changing dates, without loading new pages.

As RIA applications spread, the original vision of the Web as a highly interactive medium—as opposed to a slow page-turner—will become a reality. And hopefully, shopping carts will not become places where transactions go to die but instead avenues to a more pleasing customer shopping experience.


ColdFusion

ColdFusion is an integrated server-side environment for developing interactive Web applications. Originally developed by Macromedia, ColdFusion combines an intuitive tag-based scripting language and a tag-based server scripting language (CFML) that lowers the cost of creating interactive features. ColdFusion offers a powerful set of visual design, programming, debugging, and deployment tools.

PERSONALIZATION TOOLS

You will definitely want to know how to treat each customer on an individual basis and emulate a traditional face-to-face marketplace. Personalization (the ability to target people based on their personal qualities and prior history with your site) and customization (the ability to change the product to better fit the needs of the customer) are two key elements of e-commerce that potentially can make it nearly as powerful as a traditional marketplace, and perhaps even more powerful than di
DESIGN YOUR WEB SITE FOR ACCESSIBILITY

In 1998, Congress amended the Rehabilitation Act to require U.S. agencies, government contractors, and others receiving federal money to make electronic and information technology services accessible to people with disabilities. Known as Section 508, this legislation requires Web sites of federally funded organizations to be accessible to users who are blind, deaf, blind and deaf, or unable to use a mouse. However, the legislation applies only to U.S. agencies, government contractors, and others receiving federal money, not to the broader e-commerce environment.

In 2001, Access Now Inc., an advocacy group for the disabled, sued Southwest Airlines on behalf of more than 50 million disabled Americans for operating a Web site that was inaccessible to the disabled, on the grounds that this violated the 1990 Americans with Disability Act (ADA). In November 2002, a Federal District Court in Florida, in one of the first court decisions on the applicability of the ADA to Web sites, ruled that ADA applies only to physical spaces, not virtual spaces. However, the judge noted in a footnote that she was surprised that a customer-oriented firm like Southwest Airlines did not “employ all available technologies to expand accessibility to its Web site for visually impaired customers who would be an added source of revenue.” But this court ruling has not stopped the efforts of some government officials to make private Web sites more accessible. For instance, in August 2004, New York State’s Attorney General Eliot Spitzer announced a settlement with Ramada.com and Priceline.com that requires them to implement a range of accessibility standards (such as screen reader software, which converts Web page text into speech, and clearly labeling graphics and images) set forth by the Web Accessibility Initiative of the World Wide Web Consortium to make the sites more accessible to the blind and visually impaired. Announcing the settlement, Spitzer also opined that the American with Disabilities Act did apply to private Web sites.

If designing a Web site to be accessible to the physically challenged is easy to do, as advocates for the disabled argue, why aren’t more Web sites built with accessibility in mind? Part of the problem is that retrofitting an existing site is very difficult, according to online merchants. Building an accessible site from scratch is far easier and far less expensive than the cost of retrofitting an old site. For instance, Hewlett-Packard Co. has been working to make its public Web site, HP.com, more accessible for over five years, and it believes there’s still more work to do.

Today’s graphically rich, multimedia-enable sites make it even more difficult for the approximately 54 million disabled Americans to access Web sites. Blind users, for example, rely on a screen reader that uses software and a speech synthesizer to describe what is on a Web page. But sites that fail to use descriptive tags to explain links or images make it impossible for the screen reader to provide complete information. In the end, only 10% to 20% of all Web sites are accessible to the close to 10 million blind Americans.

Other site design elements make it difficult for the disabled to even navigate through a site. For instance, sites with small or crowded links make it more difficult for people with impaired motor skills to click on a particular link. Sites without closed-captioning cause the hearing impaired to miss built-in audio messages altogether.
There are several simple strategies Web designers can use to improve accessibility. Embedding text descriptions behind images is one example that allows screen readers to announce those descriptions. So instead of saying “Image,” when a screen reader passes over an image, the visually impaired user can hear “Photo of a cruise ship sitting in a harbor.” Allowing users to set the color and font schemes can also make a difference for the visually impaired. Adding screen magnification tools and sound labels where hyperlinks appear are two additional ways to increase accessibility.

These are examples of “equivalent alternatives” to visual content that disability advocates suggest should be required, both for visual and auditory content, to ensure that individuals with disabilities have equal access to information that appears on-screen. Other guidelines for creating accessible Web sites include ensuring that text and graphics are understandable when viewed without color, using features that enable activation of page elements via a variety of input devices (such as keyboard, head wand, or Braille reader), and providing clear navigation mechanisms (such as navigation bars or a site map) to aid users.

Making page design simpler and easier to follow sounds like good advice for sites trying to appeal to the disabled and well-bodied alike. But should Web sites be required by law to meet everyone’s needs?