



Introduction

E-commerce applications are putting greater pressure than ever on information technology organizations to accurately estimate system needs, especially as networks proliferate and take over more of the critical data processing.

Consider the recent boom in online retailing. According to a recent Forrester report, e-commerce sales are likely to reach \$41 billion by 2003. The future of today's e-commerce companies now depends on the efficiency of its infrastructure. Add to this statistic the fact that customers buy products online because of convenience and speed of delivery. If e-commerce companies cannot make e-business transactions efficient and easy, customers will purchase elsewhere or complain. For example, online stock traders offer an example of the need for adequate capacity planning. The recent spikes in online traffic and its impact on service quality resulted in end-user complaints to the Securities and Exchange Commission rising 330 percent from September 1997 to September 1998. As a result, online trading firms are busy ramping up capacity.

E-commerce applications have a tendency to saturate server and network resources quickly and could mire companies in an endless cycle of upgrades and migrations if proper capacity planning is not undertaken. Capacity planning provides users a way to predict the resources needed to ensure that e-commerce applications run optimally and estimate growth needs. Capacity planning is a means of preventing e-commerce interruptions.

Service Level Agreements

A critical aspect of any e-commerce application is its ability to guarantee quality-of-service to its users, and to provide the same guarantees on internal data delivery as it expects from outside providers of telephone, power and other essential services. A service-level agreement is a contract between IT groups and certain business units in which IT promises to deliver a specified level of performance across a network.

Availability is one metric used to guarantee network and application uptime, but according to a study conducted by Cahners, most business executives rank response time as the second most important service factor after availability. Sluggish response time is a major problem facing e-commerce. Potential customers who cannot get through to a Web site quickly get disgusted and bored, and may even give up on using the Web for commerce.

Load testing is another important metric guaranteed in service-level agreements. Good capacity planning includes tools that allow user loads to be scaled to determine if a system can handle expected loads. A precept of successful e-commerce is the ability to be open to all potential consumers when they come to browse and buy products and being able to handle all service requests and demands. E-commerce sites unable to handle unexpected user loads are analogous to a brick-and-mortar store having only one checkout line open during the holiday rush. It creates chaos and disorder, and eventually potential users who cannot gain entry either go to another store or give up on the idea of purchasing anything.

Addressing the Performance Conundrum

E-commerce companies used to think that the most urgent priority they had was to get "up



and running.” However, that is not so today. Steve Johnson of Andersen Consulting states, “It is quite clear that there’s much more attention now on performance, and so I think benchmarking is going to become a fairly useful and important tool in helping companies that are going to survive the dot.com shakeout to really focus on the things that make a difference to their business”. The tables below list responses from a telephone interview of 73 corporations conducted by Computerworld when asked about corporation e-commerce initiatives.

Which of the following thresholds or metrics does your organization use to measure your e-commerce site’s performance?

Threshold/Metric	Percentage
Number of transactions at a given period	71
Server load	59
Network load	45
Response time	3
EDI	1
External monitor (service metrics)	1
Measure the bandwidth	1
Our ISP does that for us	1
Page hits	1
Repeat hits	1
Responsiveness	1
Revenue	1
Sales dollars	1
Time spent on site	1
Unique users	1
Up-time	1
None/NA	4
Don’t know	8

Which of the following are the top three biggest performance bottlenecks on your electronic commerce site?

Bottleneck	Percentage
Bandwidth availability, especially during peak traffic periods	41
Trouble delivering detailed images and multimedia	23
Database connections	21
Slow authentication and/or security practices	18
Uneven server loads	18
Problems with site design	16
Sluggish credit card authentication	15



Delivery of dynamic content	14
Failure of 3 rd party site elements, such as ad banners	10
Firewall	1
ISP reliability	1
Log-on time	1
Low storage capacity	1
Overall sluggishness of the internet	1
Understanding the complexity of the technology	1
Uptime reliability	1
User base knowledge at our customers' desktops	1
None/NA	20
Don't know	3

Predicting Performance

Some experts say there are generally four ways to predict e-commerce system performance.

1. Statistical forecasting is collecting e-commerce performance data, such as CPU, memory, disk and network use and workload. Using this historical data, future trends may be predicted. This method is similar to weather prediction - the base predictions have a reasonable level of accuracy, but unexpected variations are not uncommon. In our opinion, statistical forecasting is a good performance reality check, but should not be used as the only method of forecasting performance.
2. User simulation is constructing e-commerce system models and applying various user loads against the system models to determine how a system responds. Constructing accurate system models is often a complex and time-consuming process. An accurate model gives a reliable estimate of performance at various user loads, but failing to create a useful model provides little useful information.
3. Profile is using standard benchmark routines developed by several companies and universities to test e-commerce system components. Results from these benchmarks, yield performance expectations of e-commerce hardware, but do not provide testing of the actual e-commerce application.
4. Synthetic Workload is a hybrid of User Simulation and Profile, consisting of simulated user scenarios exercising the e-commerce application at a variety of user loads. Synthetic workload tests the actual e-commerce application and hardware and closely simulates actual user activities. The bulk of this paper discusses using this testing methodology.

Ecommerce Architecture

Firewall

Firewalls and switches must check out requests before releasing them to Web servers. Firewalls are a major concern for e-commerce security and performance. A firewall must



provide adequate protection and throughput. If a bottleneck occurs at a firewall, performance becomes a moot point.

Load Balancer

Load balancers (content jugglers) use a rules-based engine to determine which Web servers handle each user. In many cases, Web server clusters are used instead of, or in addition to, a load balancer. Clustering groups use independent servers to work as a single system and to improve overall site performance.

Web Server

Web servers process Web pages in response to requests from browsers. Much of the e-commerce application resides on the Web server.

Cluster Solutions

Clustering lets users run a single application over multiple servers, which gives systems more channels through which to send data. The main intent of cluster servers is to provide improved performance and reliability by integrating the processor, memory storage systems and network connections of multiple network servers into a cooperative whole and distributing it across various servers.

Database Server

Database servers have become an integral part of e-commerce, and are therefore a metric to monitor. Most of the e-commerce application data resides on the database servers. Database makers strategize availability and dynamic content into its database e-business design. One feature of *Patrol Perform* is its ability to test the availability and dynamic content of database servers directly, or through the ecommerce applications.

Transaction Server

Transaction servers manage e-commerce business transactions. Transaction servers are responsible for maintaining high performance, availability and data integrity.

Cache Server

Cache servers store, or cache, frequently accessed Web pages locally and, therefore, improve response time, and therefore, Web traffic. E-commerce sites routinely experience thousands of users looking for the same information at the same time and place.

Transaction Flow

The transaction flow process begins with a customer (end user) initiating a page request from a browser to an e-commerce application. Initiating a request can be as simple as clicking on a link in an email message. The request is sent to an IP address that is assumed to be a Web server. The IP address resides behind a firewall.

The firewall decides whether or not to allow the request to pass to the Web server. If a cache server is present, it determines if it can handle the request. If the cache server can handle the request, it immediately sends the response back through the firewall to the customer. If it cannot handle the request, (the requested information is not cached) it forwards the request to the load balancer.

Load balancers vary in complexity. In the simplest case, a load balancer is a Web server that alternates transaction redirection to one of the other Web servers. More advanced load balancers determine Web server availability before initiating a transaction redirection. In some cases, a cluster of Web servers is used instead of a load balancer/Web server combination.



Once a Web server receives a request from the load balancer, it determines if any database activity is required. Assuming no database activity is required, the Web server responds directly to the customer.

If, however, the request from the load balancer does require database interaction, the Web server interacts directly with the database server (server cluster) or uses a transaction server to interact with the database. The transaction server provides an optimized means for interacting with the database.

The database cluster accepts a transaction from the Web server or transaction server and executes it on one of the database servers. The cluster appears as a single database server to the Web server or transaction server. Multiple servers are used generally with a common storage area to provide a highly reliable, high performance database system. Once a database server completes a transaction, it returns the results to the Web server through the transaction server, if the transaction server was used.

Once the Web server receives the results from the database server, it renders the information, if necessary, and responds directly to the customer. Future interactions between the customer and the e-commerce application typically are done to the same Web server.

Capacity Planning Process

Creating representative test scenarios for various e-commerce transactions is paramount to capacity planning. The following should be considered when capacity planning:

1. Most accessed pages -which pages are most frequently visited by users. Testing the most frequented pages is important when trying to determine the capacity of a Web site. By testing the most accessed pages, it is possible to simplify the test procedure while still accurately representing system users. In cases where frequently accessed pages are static, caching servers can be used to reduce the huge amount of Web traffic.
2. Transaction time -the period it takes a page to load. Response time must be in tune with users' needs and expectations. Response time issues are essentially productivity issues. Users may not care about spending money on an e-commerce site, but they do not like having to wait excessive periods to spend their money. Many IT professionals use the "eight-second rule" as the threshold for the maximum period users lose patience with a download and abort the download. According to Zona, about one-third of online shoppers frustrated with download times simply give up.
3. Latency -the period a user spends on each page. To adequately determine system capacity it is important to determine how much time is spent on each page. If simulated users spend too little time on a page, you will underestimate system capacity. If simulated users spend too much time on a page, you will over estimate system capacity. Providing some randomness for the period users spend on each page is necessary to accurately estimate capacity.
4. User load -the number of users expected to simultaneously access a site. It is critical to run an e-commerce application under a variety of user loads for sustained periods to determine how many users can simultaneously use a site before, and until, the system reaches its capacity. One of the many strengths of *Patrol Perform* is its ability to simulate large user loads on a minimal amount of client hardware.



5. User sessions -common paths through a Web site. Simulating how users navigate an e-commerce site is very important to capacity planning. User sessions simulate or recreate system events, and therefore provide an accurate representation of how a user might traverse an e-commerce Web site. *Patrol Perform* provides the user session function to conveniently and easily simulates user movements on a Web site.

Baseline Measuring

Baseline measurements are considered “best case” or ideal measurements. Baseline measurements are taken under a no-load state, and serve as a basis or as a known starting point from which to compare other measurements. It is important to perform baseline measurement. tests on common paths (user sessions), static content and dynamic content.

End-to-End Testing

End-to-end testing assesses a complete site. It is used to determine the overall system performance from an end user’s perspective and the overall system capacity of an e-commerce application.

End-to-End Process

The following describes the route a transaction follows during a typical end-to-end process:

1. A transaction travels from the load testing tool to the firewall. Firewalls authenticate transactions.
2. Transactions travel from the firewall to the load balancer. Load balancers balance Web server loads boost performance and prevent a web server from being overtaxed.
3. Transactions travel from the load balancer to the Web server. Web servers accept and process transactions
4. Transactions travel from the Web server to the cluster solution. Cluster solutions decide which database servers respond to a transaction.
5. Transactions travel from the cluster solution to the database server. Database servers gather information requested by a transaction.
6. Transactions return from the database server to the Web server.
7. Transactions return from the Web server to the firewall.
8. Transactions return from the fire wall to the load testing tool.

Patrol Perform simulates a specified number of users “firing” transactions at an ecommerce application. Some customers may access a single page, while other customers may browse and purchase items -the number and types of users are configurable.

The metrics collected by *Patrol Perform* for end-to-end testing include transaction information, the period it takes to load each page, and throughput information or the bytes per second or pages per second.

While performing the end-to-end load testing, other monitoring tools may be used to help isolate bottlenecks. Be aware of the affect monitoring and management tools have on measurements. According to the *Heisenberg Principle*, one cannot measure something without also affecting it in some way. For this reason, after tuning a system, ensure all monitors or tuning tools are turned off before conducting final measurements. Failing to turn off monitors may cause the system capacity to be underestimated. The exception to this principle is that any monitoring or management tools that run 24x7 should remain turned on during the capacity planning process.

Isolated System Testing

Isolated system testing assesses individual components of a site. It is used to determine the



individual system performance from an application perspective and is necessary to determine the individual system component capacity. Tuning can be performed while running isolated system tests to address bottlenecks identified during the end-to-end system testing process.

Web Front-End Process

The following describes the path a transaction follows during a typical Web front-end process:

1. A transaction travels from the load testing tool to the firewall. Firewalls authenticate transactions.
2. Transaction travels from the firewall to the load balancer. Load balancers balance Web server loads, boost performance and prevent a Web server from being overtaxed.
3. Transaction travels from the load balancer to the Web server. Web servers accept and process transactions.
4. Transaction returns from the Web server to the firewall.
5. Transaction returns from the firewall to the load testing tool.

It is possible to isolate a system further, to determine the performance of a Web server, or to determine the firewall or load balancer overhead. Again, monitoring or tuning tools may also be used during this process.

Database Back-End Process

The following describes the path a transaction follows during a typical database back-end! process:

1. A transaction travels from the load testing tool to the cluster solution. Cluster solutions decide which database servers respond to a transaction.
2. Transaction travels from the cluster solution to the database server. Database servers gather information requested by a transaction.
3. Transaction returns from the database server to the load testing tool.

In some cases, fail-over tests are performed on the cluster solution to determine the performance when one server in the cluster fails, and to determine the fail-over time.

Running, Monitoring, Tuning and Repeating Test!

The load testing process often is an iterative one. Frequently, the load testing process begins by running an end-to-end load test. The load test runs at the appropriate user loads to determine how an e-commerce site handles different loads. Monitor the system, application and real-time statistics while the user load test is running.

Monitored statistics provide vital information about an e-commerce site. System statistics include CPU and memory usage. Application statistics include caching performance faults and Web or database errors. *Patrol Perform* monitors real-time statistics, actual transaction times, and throughput.

While the system is under load, perform a reality check using a browser. This provides a sense of the period it takes to load pages and how an end user might respond if the system is under a specific load.

Once the initial load test completes, tune the system if possible. Tuning may be as simple as adjusting database indexes, or as performing network architecture changes (NIC size, number of NICs used.) In some cases, reverse tuning may be performed using less expensive hardware where a system was not taxed during the load testing process.

During the tuning process, isolated system tests can be performed. This allows stressing a specific system to optimize its performance.



If a system requires tuning, rerun the end-to-end procedure to determine the overall capacity and to ensure the tuning produced the expected results.

Analyzing Results

Once the load test is complete, analysis may be performed. Analysis begins by determining where the transaction times exceed acceptable levels.

Depending on the business rules, maximum time or 90th Percentile time may be selected to determine the capacity of an e-commerce application.

In addition to determining the system capacity, it is important to note where peak throughput occurs. Peak throughput can be considered to be the level at which the system is performing

Best.

Use monitoring tools to find the bottlenecks. Common places to look for bottlenecks include:

CPU

- Web servers
- Database server
- Transaction servers

Network

- Web servers
- Database servers
- Transaction servers
- Firewall

Memory

- Web servers
- Database servers
- Transaction servers

Conclusion

Use the best tools available when capacity planning e-commerce sites. An effective e-commerce site must scale with the number of users, respond reliably and quickly, and optimize its use of bandwidth. *Patrol Perform* provides all of the essential functions necessary to test an e-commerce application.