The E-Business Store
Is Always Open

Continuous Availability with Progress

PROGRESS
SOFTWARE

Contents

Introduction

> *"…as the Internet's ability to aggregate and deliver information, facilitate low-cost, asynchronous 24x7 communication, and foster collaboration continue their inevitable drive in transforming business, expect the 'e' in e-business (i.e., the Internet) to become the single most important factor in setting enterprise IT standards."*
>
> – **Daryl Plummer, Gartner Group, "Internet Scenario: The Infrastructure for E-Business," October 2000**

Globalization has created more e-business initiatives with requirements for continuous availability than many organizations can handle. Simply installing failover software and acquiring redundant hardware does not attain the goal of continuous availability. Systems should be built so that they never have to failover, but when they do it is imperative to have the right recovery plans in place. Building a resilient system and putting protective processes in place are difficult enough achievements. In the e-business world where speed to market and responsiveness are critical, an organization must meet its availability goals while still being concerned with rapid deployment of solutions and limited staff resources for development and management. Solutions built on the Progress OpenEdge™ e-business platform, which includes the Progress RDBMS, the Progress AppServer, and WebSpeed, provide continuous availability and quick deployment options throughout an e-business infrastructure, without increasing daily management costs.

Defining Continuous Availability

We can define availability as the portion of time that an application is available to a customer (internal or external customer) for productive work. Overall availability determines the resiliency of a system. When viewed in the context of the whole environment—software, hardware, and personnel—it becomes clear that addressing the many factors involved in providing continuous availability can result in significant costs. The first step in designing your availability strategy is asking what level of availability does an application require? For example, Technical Support applications for your company might require 24-hour access and sub-second response times, while other applications, such as reports or synchronizations, might only have to be available or highly responsive at specific times in the business cycle. The more available an application has to be, the higher the associated costs. Table 1 illustrates how little time is afforded for downtime as availability requirements move closer to 100% uptime.

Table 1: Uptime Requirements

Percentage Uptime	Downtime per Year	Downtime per Week
98%	7.3 Days	3 Hours, 22 Minutes
99%	3.65 Days	1 Hour, 41 Minutes
99.9%	8 Hours, 45 Minutes	10 Minutes, 5 Seconds
99.99%	52.5 Minutes	1 Minute
99.999% (5 Nines)	5.25 Minutes	6 Seconds

Assessing How Downtime Impacts Your Business

One of the most important steps in providing continuous-availability is balancing downtime with costs. The equation is simple—the more you spend on availability, the higher you move up the availability curve. However, incremental costs increase exponentially as you move from one point of the curve to the next. The cost of moving from 99.9% to 99.99% is not nearly as great as the cost of subsequently moving to 99.999% uptime.

It's significant to note that the cost of software (database, application/web servers) becomes a smaller percentage of incremental costs as availability increases due to the substantial hardware associated with approaching zero downtime. Figure 1 shows how hardware costs rise at substantially increasing rates as availability approaches 100%.

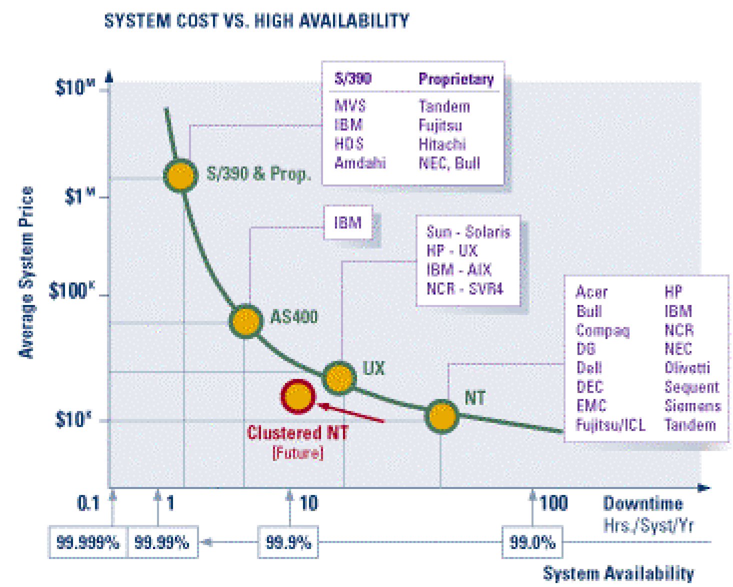

Figure 1: Hardware Costs of Providing Continuous Availability

The result is that organizations must first price out their downtime in order to determine how much they can afford to spend to protect against service outages. The real costs of downtime affect so much of a company's business that IT departments have now become an actual part of the business process.

In the case of a Progress application, the costs for ensuring higher levels of availability are associated almost exclusively with acquiring hardware and third-party software to manage hardware failover and fault-tolerance. The Enterprise-class products that you purchase assure you continuous availability out of the box. For example, once you've determined that the Enterprise RDBMS (as opposed to the Workgroup database) offers you the performance and availability your application requires, the standard license affords you the maximum availability. You don't pay a premium or have to purchase add-ons to get the reliability you need.

The total cost of a service outage can be difficult to tally, because the resulting lost business or customer dissatisfaction can't always be quantified. In many casesTable 2 lists some well-publicized service outages that provide examples of the wide range of associated costs when a business (not just an online business) experiences an interruption in availability.

Table 2: Losses Due to Service Outages

Company	Downtime	Cost	Cause
eBay June 1999	22 hours	$3-5 Million 26% decline in stock price	Operating-system failure
AT&T April 1998	6-26 hours	$40 Million Forced to file SLAs with FCC	Software Upgrade
MCI August 1999	10 days	20 days free service to 3,000 enterprise customers	Software Upgrade
Charles Schwab Feb-April 1999	4 outages of at least 4 hours each	Unknown – announced a $70 million infrastructure investment	Operator errors and upgrades
Hershey Foods Sept 1999		12% decline in 3Q99 sales; 19% drop in net income from 3Q98	System failures & application rollout

– **John Phelps, "Data Center E-volution," Gartner Group, 2000**

> *"Online retailers that enjoyed rapid success found out that they risk losing business and damaging their brands if their sites don't stand up to strong demand. Excellent site performance requires excellence in functionality in three areas: last-mile performance (i.e.,. connection to the Internet), server performance, and the middle-mile (i.e., Internet) performance."*
>
> – **Jupiter Communications, "The Internet Business Enterprise: Moving Beyond the Sell Side," 2000**

When customers or users experience service outages, they associate that outage with the application provider or Web site host, regardless of whether the failure was on the Internet service provider's side of the wire or within the Internet's backbone. Since your e-business application is vulnerable to failures in the Internet and communications infrastructure, it is all the more critical to minimize the risk of failure within the part of the system that *is* under your control.

The first step in addressing a problem is to isolate its causes. All hardware has mechanical issues and most machines come with stated *mean time between failures* (MTBF). Software systems contain your network, desktop/client, and any server software, such as database and applications. Staffing issues must be accounted for and include missed days, operator errors, and lack of skilled workers. Another large part of continuous availability is the factor of scheduled downtime such as system upgrades. Figure 2 illustrates the percentage breakdowns for each of these factors in availability.

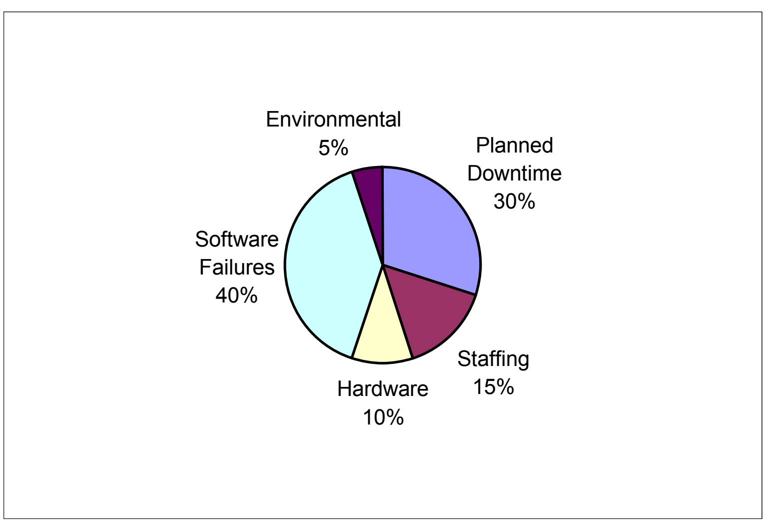

Figure 2: Factors That Influence Continuous Availability (IEEE Computer, April 1995)

Hardware systems have advanced significantly so that the two largest areas affecting availability are planned downtime for system maintenance and software failures. Software failures fall into three more categories: server software, client software, and network software. These issues result in the requirement to have technology and plans in place to affect software failures (40%) and planned downtime (30%) to ensure the availability you require (IEEE Computer, April 1995).

Continuous Access to Application Data

> *"One major advantage that Progress brings to the table is reliability. Our product, ChemPax, has been in environments where customers have used other databases, and they are just amazed by the reliability of the Progress database. It never breaks down. They're also amazed that they don't need a database administrator to run it. The tightness of the technology and the products' ease of use bring tremendous cost savings to our customers. They really appreciate that in the long run."*
>
> – **Sean O'Donnell, President, DataCor**

At the center of most any application system is the database. The database handles a variety of tasks, such as reading and writing to the disk, responding to queries, and managing placement of data. Additionally, the database may use sophisticated processes to help handle certain operations (such as disk I/O and block cache). The database must maintain the resiliency of the data while still providing fast access and the ability to recover gracefully in the event of a system failure. A database system that serves as the backbone of an e-business implementation must also minimize the amount of planned downtime required for routine administrative and maintenance tasks. The Progress RDBMS meets all these requirements.

The Enterprise RDBMS is recommended for high-volume, distributed, enterprise-level applications with the following characteristics:

– A moderate (30) to high (5,000) number of concurrent users
– A moderate (2 Gigabytes) to large (256 250+ Gigabytes) database size, using multiple disk drives
– Real-time distributed updates performed across more than one database or one computer system
– Executes in a symmetric multi-processor (SMP) environment that includes two or more CPUs

Table 3 shows the primary features that distinguish the Enterprise RDBMS from the Workgroup RDBMS. Many of the features that ensure continuous availability, especially from the administrative perspective, are unique to the Enterprise RDBMS.

Table 3: Enterprise and Workgroup RDBMS Comparison

Feature	Enterprise RDBMS	Workgroup RDBMS
Asynchronous Page Writers	X	
Background I/O	X	
Buffer Pools	X	
Concurrent Commit Lock Protocol	X	
Cluster Support	X	
Database Quiet Point	X	
Fire Wall Support	X	X
Multiple Brokers Per Protocol	X	X
NT Backup Integration	X	X

Feature	Enterprise RDBMS	Workgroup RDBMS
On-Line Index Fix	X	X
On-line Table and Index Reorganization	X	
Raw Device Support	X	
Roll Forward Retry	X	X
Schema Change Enhancements	X	X
SMP Support	X	
Updated Two Phase Commit	X	X
Variable Block Sizes	X	
Variable Database Blocksize	X	
Virtual System Tables	X	X
Zero Impact Backup Support	X	

Platform, Cluster, and Failover Support

As more and more hardware and software solutions are presented in the goal to obtain continuous-availability, it is important to ensure that your software technology will support your ultimate choice. No value is gained from adopting the latest clustering solution and the fastest server machines if your applications and database can't support them. The Progress RDBMS provides a wide range of operating system support, as well as cluster and storage solutions, to enable you to deploy the solution your business requires.

Clusters can be used for increased performance capabilities by providing concurrent access to two or more servers to resources such as applications and databases. Clusters provide continuous-availability by guaranteeing that if a server in the cluster fails, the application or database won't fail since it's running on another server, and clients can be routed to available servers. There are clusters that are marketed as MPP (massive parallel processors) from vendors such as IBM. One example of this type of cluster ran the Progress NxTrend 5,000 user benchmark (IBM RS/6000 SP cluster). For detailed information on the specifics of this benchmark, read the report at www.progress.com/benchmark/nxtrend.htm.

Support for symmetric multiprocessing (SMP) is a common requirement for any critical business and Progress has had support since Version 6.3. In fact, the Progress RDBMS was the first database product to run on many SMP platforms including IBM RS/6000. For many Windows NT platforms, Progress RDBMS has proven to be one of the best scaling multi-processor databases. Read about the ATM benchmark test at www.progress.com/benchmark/v91.htm. This commitment to support the latest technology translates into the freedom to design best-of-breed solutions for your continuous-availability needs.

Progress also supports non-uniform memory architecture machines (NUMA). In fact, Progress has one of the most powerful NUMA systems in the world in our Bedford, MA development labs. Progress actively works closely with vendors to run as best as possible on NUMA and many other systems.

More important for continuous availability is the role of clusters as "failover" support. Failover is defined as the moving of resources from a primary server to a secondary server when the primary server either system failures or there is manual intervention to force resource reallocation from one node to another. Failover provides a way of keeping resources available to users in the event of a system failure. The primary server and secondary server are independent under normal conditions, but have access to each other's disks or a shared disk system. In the event of failure in the primary server, the secondary server takes over requests. The secondary server may stand idle until required, or may run non-critical applications as a "hot failover." In the event of a failure, it will still have to carry on supporting its own applications and users in addition to those of the primary server.

Progress works with failover cluster systems on Windows NT and UNIX.The Progress RDBMS allows the recovery time of the database to be tuned to limit the time it will take to failover. In practice, failing-over a Progress database is a small part in both management actions and time required on a failover cluster.

There are also a large number of disk subsystems that can be employed in a continuous-availability environment. Most disk subsystems present themselves to the operating system as standard disk devices. This means that if a system is using some special device like a RAID (Redundant Array of Inexpensive Disks) controller, Semiconductor Storage (solid-state memory that looks like a disk), or anything else, Progress will not need to know that the device is not a standard disk drive and Progress will automatically work with such devices. We encourage that you test all disk systems thoroughly to ensure that they meet requirements Progress has for I/O operations.

Background Writers

On shared-memory systems, the Progress RDBMS uses background writers to improve performance by continually performing overhead functions in the background. There are three types of writers: asynchronous page writers (APWs), before-image writers (BIWs), and after-image writers (AIWs). APWs help reduce checkpoint overhead, buffer replacement, and help supply empty buffers. BIWs continually write filled before-image buffers to disk so that client and server processes don't have to wait for a filled buffer to be written, improving I/O performance. AIWs write after-image buffers to disk when filled and also prevent client and server processes from waiting for a modified buffer to be written to disk.

Online Backup

Online backup provides a mechanism to back up the database while it is still in use. This single feature greatly reduces the amount of planned downtime for your e-business solution. You don't have to close your doors to perform regular maintenance and protect your data's integrity. When you perform an on-line backup, Progress automatically switches over to the next after-image file so that you can continue operations. Additionally, the database is placed in a quiet state for backup while users are still connected.

Based on your backup plan, you may choose to do fewer full backups and more incremental backups (both methods are supported by the Progress RDBMS). An incremental backup backs up only the data that has changed since the last full or incremental backup. Incremental backups take less time and return the database to full availability for complete transaction processing faster.

Site Replication

Another consideration for assessing whether a database can meet the challenges of 24x7 demand is the ability to support site replication. Progress *log-based site replication* allows the same data used to support roll forward recovery to support an on-site or off-site complete replication of a database. The replicated database (on another remote site) can be maintained to reflect the primary database's state within seconds. In the event of a complete site loss (fire, earthquake, etc), a remote *hot standby* site could be quickly restarted to provide continuing service.

Recovery

In the event of any system failure, it is vitally important to have a database (as well as other components) capable of smoothly and quickly recovering in order to reduce the amount of unscheduled downtime. The critical concepts are *reliable database state* and *crash recovery*.

A database must serve two goals that can sometimes be in conflict. First, users' requests must be serviced within acceptable response times. Some requests are complicated, so the database sometimes changes many parts of the physical storage. Second, and sometimes in conflict with the first due to associated overhead, changes to the database must be committed so that in the event of a system failure, recovery to a reliable database state can take place. A reliable database state (maintaining consistent state) is provided through three techniques: write ahead logging, fuzzy checkpoints, and database startup/shutdown.

Write ahead logging (WAL) allows multiple updates to a database buffer. The database buffer is not written to disk immediately and it also allows for updates from different transactions. Changes to the database buffer are recorded using the before-image log. This before-image log is written first using reliable I/O and it is written when convenient.

The Progress RDBMS uses checkpoints to reconcile buffer pools via a periodic synchronization between the database and shared memory. This reconciliation occurs each time a cluster is filled and limits recovery time to changes since the last checkpoint completed. This brings up a problem with checkpoints that Progress *fuzzy checkpoints* resolve. Because a traditional checkpoint is filled and finishes reconciliation before another cluster is opened, a backlog of I/O can occur to flush buffers and the database cannot change during a checkpoint. This can result in erratic performance——affecting availability. Fuzzy checkpoints can span two clusters and spread out the required I/O. Since shared memory and database context never match, the result is a "fuzzy" or asynchronous checkpoint.

Requirements for recovery of the database are determined by the loss of media, or no loss of media. The Progress RDBMS state is contained in three parts: database, shared memory, and before-imaging. The recovery basically consists of reconstructing shared memory. In order to do this, Progress uses the before-image log to "redo" transactions, making committed transaction durable.

In the event of a loss of media during a system failure, *roll-forward recovery* is implemented to recover transactions lost. Online backups are integrated completely with our after-image logging subsystem to allow roll-forward recovery of transactions that completed after an online backup was started. The initial database is restored from backup and then after-image logs read forward and "redo" each note. When crash recovery is completed, the database is in a consistent state. The Progress RDBMS can recover from a system failure during a recovery and also resume a cancelled roll-forward operation to reduce recovery time.

Remember, to protect against loss of media, you can enable after-imaging for the RDBMS. Before-imaging, which occurs automatically, protects you from system failure, but not from loss of corruption of your database or primary recovery area and its extents. Roll-forward recovery uses after-image files to restore your database to the condition it was in before you lost your database. After-image extents, together with your most recent database backup, contain the same information as your database files. After-imaging, combined with a proper recovery plan, provides tremendous coverage to limit unscheduled downtime after a system (or media) failure.

Progress Application Servers Guarantee Continuous Availability

A critical component of a continuous-availability solution is the application server that handles the requests from users. These servers act as brokers for access to the database. The Progress AppServer or the WebSpeed Transaction Server are the application servers at the core of many of the e-business solutions built with OpenEdge™. Figure 3 details the architecture of the Progress AppServer, which powers distributed Progress applications.

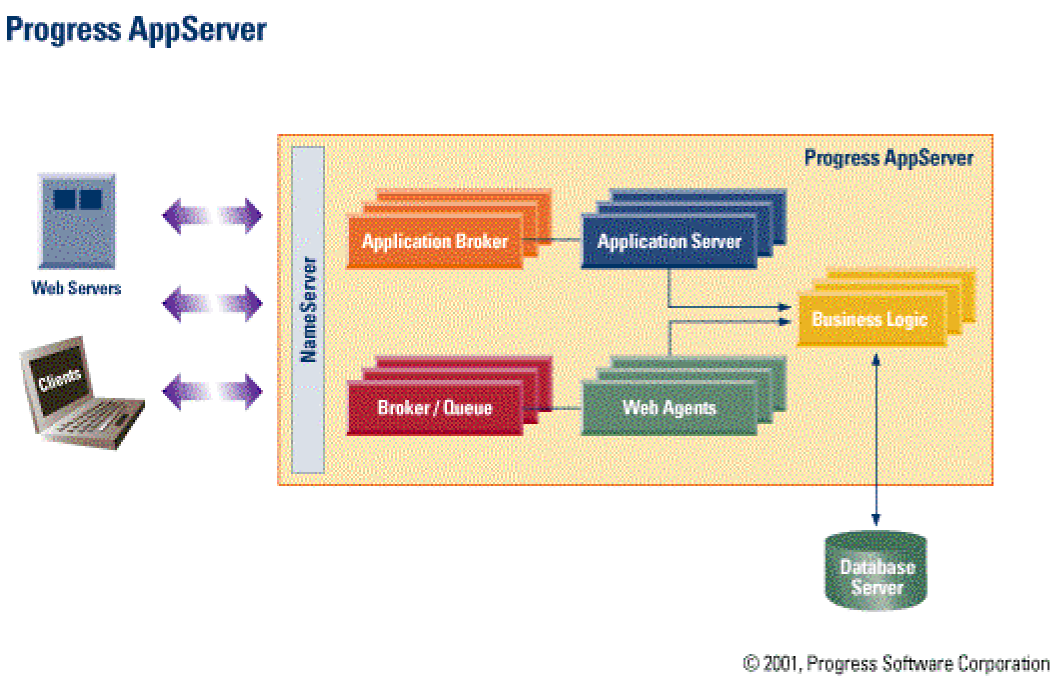

Figure 3: Progress AppServer Architecture

The other application server option for Progress e-business infrastructures is the WebSpeed Transaction Server, which provides an optimized transaction-processing environment to help achieve high transaction volumes and rapid responses for Web applications. With high throughput, sub-second transaction times, and dynamic load balancing, you get unsurpassed scalability to handle thousands of simultaneous users and provide continuous-availability. Figure 4 illustrates the WebSpeed Transaction Server architecture.

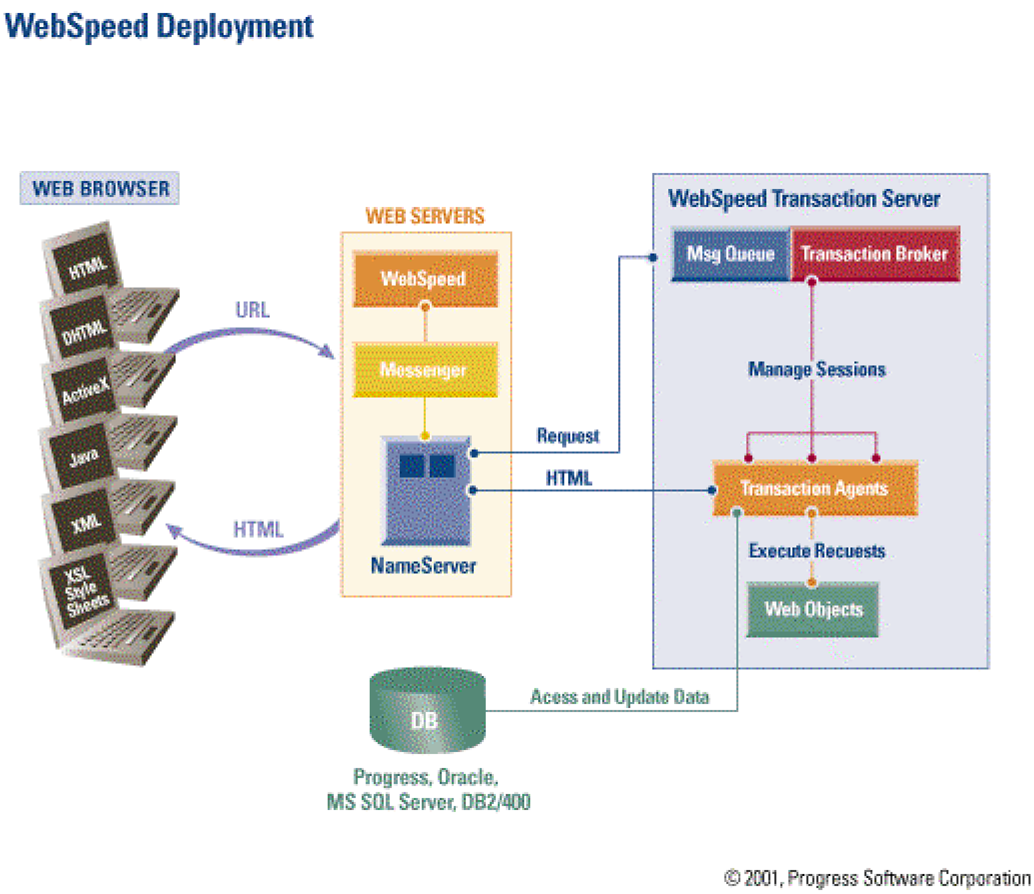

Figure 4: WebSpeed Architecture

The WebSpeed Transaction Server consists of a messenger, brokers, and agents. Message queuing increases agent availability by simply adding the messages to a queue when all the agents are busy with other requests. As agents become available, requests waiting in the queue are answered in the order in which they were received.
Record locking, transaction rollback, and two-phase commit allow WebSpeed to safeguard application and data integrity—even if transactions are interrupted—and guarantee the integrity of transactions that span multiple databases.

Dynamic Location Transparency, Load Balancing, and Failover Protection

The Progress application server architecture includes a NameServer, which directs client requests to available server processes. The NameServer provides applications with three benefits directly related to continuous-availability: location transparency, load balancing, and failover. Location transparency means that client applications don't have to know about all of the application servers that are on the middle tier of an e-business solution's infrastructure. The client simply makes a connection to a NameServer requesting to execute some business logic on an application server. The NameServer is responsible for keeping track of where the business logic exists and then re-connecting the client with that 'service.' As additional application servers are added to the middle tier or moved to other locations, the NameServer records these events. The client application doesn't have to be modified (unless new business logic was added that the client doesn't know how to access). This greatly reduces the amount of maintenance and update on client applications.

Load balancing is the ability to route requests across multiple application servers. For example in an AppServer configuration, a machine set up to process Inventory requests could become overloaded. On a separate machine, another AppServer broker could be created for Inventory that is delegated by the NameServer to take 60% of inventory requests while the remaining 40% remain on the first machine. This is all transparent to the client application, which greatly reduces maintenance and administration. The benefit is that the Progress application servers identify and resolve bottlenecks that could lead to denial of service to a client or a long response time.

Failover in the NameServer means that two brokers that provide exactly the same functionality can be created. If the first broker fails for whatever reason (hardware failure, etc.), the NameServer can automatically direct all subsequent requests to the second broker. You can even replicate the NameServer so that in case of a failure, connection requests can be routed to another NameServer.

Flexible Operating Modes Maximize Processing Resources

Another issue influential in continuous availability is flexible operating modes. The OpenEdge application servers support state-reset, state-aware, and stateless modes. The appropriateness of these operating modes depends on the goals of your application.

In the AppServer architecture, state-reset and state-aware are similar modes because a client application is directly connected to an application server process for the lifetime of the connection to the AppServer installation. The application server process that is connected to a client application is dedicated to that client application until the client application disconnects (one application server process per client application). Because the application server process is dedicated to the client application, all “state” information associated with that client application can be maintained via the 4GL executing within the application server process.

With the stateless operating mode, a client application’s request is dedicated to whichever application server process is available. Requests are first sent to an application broker and then the application broker controls directing the request to an available application server process. If all server processes are busy, the request is put into a message queue to be dealt to a server process when one becomes available. In the stateless operating mode, a few application server processes can service many client applications and provide continuous-availability benefits to more clients.

As with the AppServer, WebSpeed includes flexible state management. This state management provides full support for extended database queries and updates using stateless, state-aware, or state-persistent Web objects.

Security

Part of ensuring continuous availability is protecting against security attacks that can result in unscheduled downtime and compromises in critical data. The Secure AppServer is enabled to allow you to use HTTPS tunneling through the secure AppServer Internet Adapter (AIA/S). HTTPS is HTTP on a Secure Socket Layer (SSL) connection rather than the clear text TCP/IP connection. It is the standard for secure connections over the Internet and is used when privacy, integrity and server-side authentication are needed in addition to HTTP tunneling. In the case of WebSpeed, its open architecture gives you the freedom to integrate your choice of security solutions, including firewall, authentication, authorization, and encryption technologies.

Conclusion

The costs associated with continuous availability are investments in the continuing access of your systems, and the operation of your business. The level of availability that is required by your business is determined by weighing how much you can spend against how long you can afford to deny your customers or users access to your systems. After you have assessed your e-business application's availability requirements, you can count on the core components of the Progress OpenEdge e-business platform—the Progress RDBMS, Progress AppServer, and WebSpeed—to provide for continuous-availability solutions without adding prohibitive complexity and cost.

Corporate Headquarters
Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: 781 280 4000 Fax: 781 280 4095

Europe/Middle East/Africa Headquarters
Progress Software Europe B.V. Schorpioenstraat 67 3067 GG Rotterdam, The Netherlands Tel: 31 10 286 5700 Fax: 31 10 286 5777

Latin American Headquarters
Progress Software Corporation, 2255 Glades Road, One Boca Place, Suite 300 E, Boca Raton, FL 33431 USA Tel: 561 998 2244 Fax: 561 998 1573

Asia/Pacific Headquarters
Progress Software Pty. Ltd., 1911 Malvern Road, Malvern East, 3145, Australia Tel: 61 39 885 0544 Fax: 61 39 885 9473

PROGRESS SOFTWARE

www.progress.com