S/390 Division

The Dinosaur and the Penguins: S/390 Qualities of Service for LINUX

June 2000

To capitalize on the opportunities promised by e-business, you need Web servers that are cost effective; provide rapid application development and deployment; and deliver the high reliability, availability, and scalability that your Web workloads require. Different platforms have met these requirements to different degrees but no platform has provided a clearly superior solution to meet all.

Enter LINUX®. Linux is leveling the playing field for all servers in terms of rapid application development and deployment. Linux® for System/390® provides new options for server consolidation on S/390®, as well as new options for deploying new Web workloads on S/390. Linux inherits S/390's superior self-configuring and self-healing attributes. This inheritance means that you can improve the reliability, availability, and scalability of a Linux application by running it on System/390, while achieving the rapid application deployment promised by Linux and the server consolidation enabled by S/390.

The summer intern who created your first Web page is now a dot-com CEO. The server he put together is obsolete and has been junked. Its replacement has grown to a shelf full of servers to handle a workload that is growing in both size and importance. To ensure the ability to handle peaks, you're running each of those servers at about 25% utilization. And you can see that shelf full of servers becoming a room full of servers as the workload and its importance continue to grow.

How many servers did you start with? How many do you have now? How many do you really want to have?

The preceding scenario probably sounds familiar, through either experience or hearsay. This scenario might be encouraging at the start of a company's entry into e-business but it isn't sufficient for effective exploitation of e-business into the future. It's worthwhile to examine why this scenario is pervasive and what has changed that can improve it.

The requirements a Web server must meet fall primarily into three categories: cost; time-to-market; and reliability and availability.

- Cost. That first Web server is, more often than not, selected on the basis of price—initial cost. This approach generally favors "white box" Intel® servers (running either Linux or Windows NT) followed by branded UNIX® servers. As the number of servers grows to accommodate growing workload, however, the attendant cost of server proliferation, from management nightmares to floor space, also grows. Server consolidation has become an important goal of many companies who find themselves too familiar with the preceding scenario.

- Time-to-market: rapid application development and deployment. As Web workloads grow in terms of users, they also grow in terms of applications. Even if you are just starting in e-business, you must deliver more than a few Web pages that users, whether consumers or business partners, can access and read. Your competitors are providing full-function Web sites. You have no choice but to do the same and to quickly deploy new applications to provide ever more sophisticated interactive functions.

  This criteria has usually given the edge to UNIX systems. UNIX applications require some degree of porting because of the many flavors of UNIX, but they can normally be deployed across servers more quickly than applications written for other platforms. UNIX has, in the past, seemed to offer more freedom of choice in terms of application development and deployment. Although Java™ offers the same promise, UNIX skills have been more prevalent.

- Availability, Reliability and Scalability. Web workloads that once were considered appropriate for "good enough" servers now require the high availability and scalability traditionally associated with OLTP workloads. Customers, suppliers, and business partners demand around-the-clock availability and rapid response time. Outages at highly visible Internet sites are embarrassing as well as expensive: they generate bad press, lost revenue, even market devaluation. All e-business servers — whether the workloads are Web or OLTP—need to deliver continuous uptime, regardless of workload, and good response time, regardless of demand.

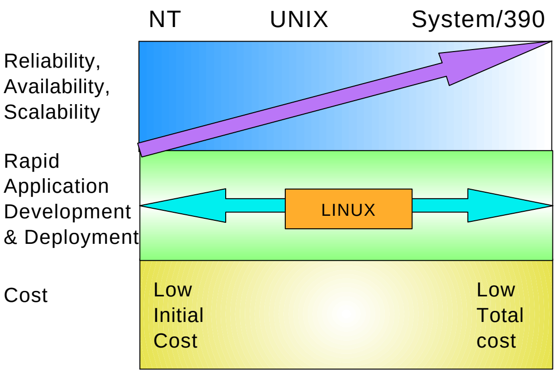
  Mainframes have always been acknowledged as the leaders in availability, reliability, and scalability—as endorsed by vendor claims for mainframe robustness:

- Sun’s E10000 “provides mainframe-style features.”[1]
- Compaq’s E2000 platform architecture has “high availability previously offered only with mainframe systems.”[2]
- HP is developing “mainframe-like resilience.”[3]
- Unisys’s 32-way delivers “the reliability, availability and serviceability of a mainframe."[4]
- Tandem and Silicon Graphics are emphasizing “traditional mainframe values.”[1]

Although the claims are similar in acknowledging the mainframe as the standard for robust qualities of service, the degree to which mainframe attributes are implemented varies widely (except, of course, on the mainframe). Because of initial cost and rapid application and deployment, UNIX and Windows NT servers have often been considered “good enough.”

To put all of these requirements together, you need a (1) cost-effective solution that supports (2) rapid application development and deployment while delivering (3) high reliability, availability, and scalability. It has seemed that different platforms have provided different ways, to different degrees, to meet each requirement; and no platform has provided a clearly superior solution to meet all. Mainframes are the acknowledged leader in reliability and availability. UNIX servers have led in rapid application development and deployment. For initial price, Intel servers boast low entry prices while mainframe capabilities for server consolidation can lower total cost of ownership.

Enter Linux into this scenario. Adding platform-neutral Linux to the picture levels the playing field for all servers in terms of rapid application development and deployment. Gartner Group gives Linux an ‘A’ for Web ISV enthusiasm, works well with others, and “coolness.”[5] Oracle and SAP have each named Linux its UNIX reference platform. Leading edge ISVs for net-gen, Internet service providers, and application service providers overwhelmingly prefer Linux as a development platform. Because Linux can run natively on a variety of platforms, including IBM’s S/390 mainframe, Linux is becoming the choice for deployment.


NT
UNIX
System/390
Reliability, Availability, Scalability
Rapid Application Development & Deployment
LINUX
Cost
Low Initial Cost
Low Total cost

Not long ago, the idea of using “big iron” for infrastructure applications would have been dismissed as preposterous, except where applications demanded the highest levels of reliability, availability, and security. Linux for S/390 is turning a spotlight on System/390 for applications not traditionally associated with S/390.

With the addition of Linux to S/390’s family of operating systems, S/390 becomes the only server to support all of the following scenarios—and all of these scenarios simultaneously on one machine:

- Web workloads that access data on OS/390®, VSE/ESA®, or VM/ESA®, supported by running them on Linux on S/390, enabling you to collapse the traditional three-tier structure to two tiers and gain faster access to the data
- Web workloads that are independent of any other S/390 operating system, supported by running them on Linux for S/390.
- Web workloads that demand the highest levels of reliability, availability, and security, supported by running them on OS/390 on S/390

The first two scenarios provide new options for server consolidation on S/390, as well as new options for deploying new Web workloads on S/390—options that deliver S/390’s reliability, availability, scalability, and flexibility to Linux.

Linux’s Inheritance from S/390

Linux inherits from S/390 the same qualities of service that OS/390, VSE/ESA, or VM, have inherited from S/390: S/390's **self-configuring** attributes, which provide scalable performance to workloads with unpredictable demands, and its **self-healing** attributes, which offer the continuous availability that e-business requires.

- Self-configuring attributes enable more work to be processed within a single server without over-configuring for complementary peaks. For example, with S/390's fine-grained resource sharing, processor resource will be delivered to the workload that requires it, in real time, without manual intervention. This is a significant improvement over the usual approach used by most UNIX and Windows NT vendors, to assign physical processors (or system "boards" containing processors, I/O, and memory) to a specific image.
- Self-healing attributes minimize application downtime because of hardware failure. S/390 hardware detects and corrects errors without, in most cases, affecting the application. All of this is done at machine speed, by hardware, with no involvement or interruption of software—or people— at any level.

The following table identifies key attributes of S/390, described in the next section of this paper, that enable the scalability and availability that make S/390 such a compelling choice for server consolidation. All S/390 hardware attributes are inherited by Linux transparently. The last column in the table identifies whether Sun’s E10000 provides an equivalent attribute. The comparisons between S/390 and the E10000 apply to a standalone SMP (symmetric multiprocessor), without clustering.

Although Sun claims mainframe class qualities, none of the attributes are matched by Sun's premier offering, the E10000. There is a significant difference between being *like* a mainframe and *being* a mainframe.

S/390 Self-configuring Attributes	**OS/390 on S/390**	**LINUX for S/390**	**Solaris 8 on E10000**
Fine-grained resource sharing: Multiple operating system images can concurrently share the same CP (processor or I/O path.	yes	yes	no
On-demand resource delivery: The system delivers resources to the operating system images as the resources are required, in real time, according to weightings that reflect business objectives	yes	yes	no
Simultaneous connectivity to data: Multiple operating system images can connect to data simultaneously.	yes	yes	no
Unbounded operating system support: Hundreds of operating system images are supported for server consolidation.	yes	yes	no
Operating system spawning: New images of operating systems can be started without affecting ongoing work.	yes	yes	no
S/390 Self-healing Attributes	**OS/390 on S/390**	**LINUX for S/390**	**Solaris 8 on E10000**
Computational integrity: Extensive error checking of arithmetic and logical functions to ensure computational integrity as well as data integrity.	yes	yes	no
Fault tolerant cache hierarchy: All data in the cache hierarchy protected by data redundancy, in addition to normal error detection in caches and ECC in memory	yes	yes	no
Transient error recovery: Extensive retry mechanisms to prevent downtime as the result of the transient errors that increasingly occur in semiconductor technology	yes	yes	no
Memory Chip Sparing[1]: Nondisruptive substitution of a new memory chip by hardware when an error threshold is exceeded for a chip.	yes	yes	no
CP sparing[1]: Nondisruptive substitution of a new processing chip when retry is unsuccessful.	yes	yes	no
Zero outage hardware repair: No outage at failure time or at repair time for most hardware failures.	yes	yes	no

Most of Sun's claims for mainframe-like qualities of service are based on their "Dynamic Domains." An E10000 is an SMP made from 4-16 4-way SMP building blocks called system boards, each board containing up to 4 CPUs, 4 gigabytes of memory, and 4 I/O ports. This 4-way SMP is the smallest boundary for everything: operating systems instances, reconfiguration, concurrent repair. When one 4-way crashes, the domain crashes. The domain can be rebooted without the failed 4-way and its subsequent repair can be done without another reboot (this is Sun's 'hot plug'), but not crashing in the first place is what defines self-healing. If you want to change the size of one Solaris from, say, 8-way to 12-way (it's got to be a 4-way boundary) there are manual procedures or you can script time-of-day based procedures (what's called 'dynamic attach'). It can take a long time if the 4-way being added to the domain needs to stop participating in another domain ('dynamic detach'). There is really no self-configuring; none of this occurs in real-time or in response to changing workload demands or without intervention.

Self-Configuring Servers

The importance of self-configuring attributes is most often associated with dynamic reconfiguration for maximum availability: the ability to switch to alternate or redundant resources in a failure situation. (See the side bar "How dynamically can your server reconfigure?") What is less recognized—probably because it's less common and, on UNIX systems, even unexpected—is the importance of self-configuring attributes for scalability *and* for capacity.

Today, scalability and capacity are mutually exclusive on many systems. To deliver scalable performance, many sites adhere to a policy of overcompensation and underconsumption. Dell claims to run at 33% capacity, E*Trade at 25%. Often, peak to average ratios for e-business applications are much worse, approaching 10:1, and actual installed capacity may be even greater. The result is not only widespread underutilization but also server proliferation and systems management nightmares.

How dynamically can *your* server reconfigure?

On S/390, dynamic reconfiguration means you can exploit redundant resources during normal processing and switch to a redundant resource after isolating a failing resource without stopping applications or the system on which they run. On more traditional UNIX systems, dynamic reconfiguration more often means the ability to reboot a system after switching to and activating an alternate resource—which sat idly during normal processing.

Ask your server if its flavor of "dynamic" reconfiguration supports the following capabilities that S/390 considers standard:

- Redundant I/O paths can be used for normal processing. (On Sun Solaris, this is true only for the RSM 3000, which is a more costly dual-pathing solution than Solaris's Alternate Pathing. RSM 3000 cannot be used with Alternate Pathing.)
- A different I/O path can be used to request I/O and to transfer the data. (On Sun Solaris, each read or write to the file system must use the same controller.)
- Multiple reconfigurations can be done simultaneously. (On Sun Solaris, only one dynamic reconfiguration of a system board can be active on the machine at one time. Multiple dynamic reconfigurations, even against different domains, are not supported.)
- I/O, processor resource, and memory can be dynamically reconfigured independently. (On Sun Solaris, dynamic reconfiguration is on the level of a system board, with up to four CPU processors, four gigabytes of memory, and four I/O controllers on each board.)
- The system can initiate dynamic reconfiguration for load balancing or error situations, without human intervention. (On Sun Solaris, limited scripting is the only alternative to manual intervention.)

When self-configuring is defined solely in terms of *switching* physical resources, it is limited in its ability to deliver the resources to meet real-time unpredictable demand for the resource. To address scalability and capacity, self-configuring must encompass *sharing* a resource. The granularity at which resources can be shared defines the system's ability to deliver resource to meet demand.

S/390 achieves highly granular sharing with its logical resource sharing. With logical sharing, each user of a resource, from its point of view, has the entire resource. Logical sharing is critical to the ability to quickly and effectively respond to unpredictable demands for resources. The extent to which physical switches and physical divisions are unnecessary is the most basic parameter in determining the speed and ease of self-configuration.

Simplified server consolidation scenarios from yesterday and today dramatize the kind of self-configuration required in today's environment.

Yesterday's server consolidation

Three business workloads run on three different machines in three different business units of a company—one business unit in Asia, one in Europe, and one in the Americas. Although each workload needs to be available 24 hours a day, the peak periods for each workload occur in three different 8-hour shifts. It is clear that the three workloads could be consolidated on a single computer with resources switched to each workload at the start of its 8-hour shift.

The domains provided by Sun E10000 Solaris systems can meet the kind of demand represented by this scenario. Each workload can run in a separate physical partition (called a domain by Sun). System boards, each containing four processors, four I/O adapters, and four Gb of memory, can be detached from one partition and reattached to another partition at the switch of each shift. Note that the resources on a system board can be *switched* between partitions, but they cannot be *shared* by partitions

Today's server consolidation

Three Web workloads run on three different Web servers, one Web server supporting Asia, one supporting Europe, and one supporting the Americas. In this scenario, the peaks of each workload are not predictable—the World Cup playing in Europe, or the Olympics in Australia, will cause peaks that have no respect for time of day or for scheduled resource switching.

To consolidate these workloads on one machine, resources must be capable of being exploited by the server that requires the resources whenever the demand occurs. The key differences between today's scenario and yesterday's scenario are the unpredictability of the demand and the granularity of the demand: minute to minute changes in demand can require anything from the total resources of a system to a percentage of a processor or I/O path.

System/390 enables this scenario, and its even more realistic variations that include hundreds of servers and multiple operating systems.

To make the second scenario practical, S/390 supports:

- Hundreds of operating system images—the same or a mix of operating systems—on one physical machine
- The ability to spawn additional images as required, *without* affecting existing images.
- Access to data by multiple operating system images *without* replicating the data.
- Sharing a single physical resource, such as an I/O path or a processor, between multiple system images *without* physically switching the resource
- Changing the delivery of resources to each system as demand requires, from minute-to-minute (in reality, millisecond to millisecond or less), *without* human intervention.

These attributes achieve maximum scalability and capacity, to the point where server farms, with system images in the hundreds, can be consolidated on a single System/390 SMP. You can consolidate multitudes of servers on a single machine with simplified systems management and minimal environmental requirements (imagine how many fewer electrical outlets you need).

This ability is especially significant to Linux and the Linux culture of one application per server. Many Linux application providers certify their applications on Linux only if the application is the only application running on Linux. With S/390, you can literally run hundreds of Linux servers while each believes it is S/390's only guest.

How Does S/390 Do That?

The key to S/390's unmatched self-configuration attributes is granularity of resource sharing that is invisible to the operating systems running on S/390. S/390 allows its guests to share the same real estate while each believes it is sole owner.

Sharing Processor (CP) Resource

The basis of S/390's processor sharing is S/390's logical partitions (LPARs, to those familiar with S/390). Instead of physically dividing the machine, you define logical partitions, each running one—or more—operating system images. You do not assign physical processors to each partition; you *control delivery of processor resource* to each partition by assigning CP shares and specifying weights for each partition.

Logical CPs define the maximum amount of processor resource that can be delivered to a partition, when that capacity is available; weights define the pecking order among partitions, when the partitions compete for capacity. Both logical CPs and weights can be changed dynamically, in real-time, without affecting the workloads running in the partitions. Neither logical CPs nor weights refer to physical processors:

- A processing request from a partition can be satisfied by any of the shared physical processors; the next processing request from that same partition can be satisfied by a different physical processor.
- When one partition loads a wait state, the system will deliver the processor resource it was using to another partition—the sharing is done at the level of milliseconds.

<ins>The Hard Case for Sun's "Dynamic" Partitions</ins>

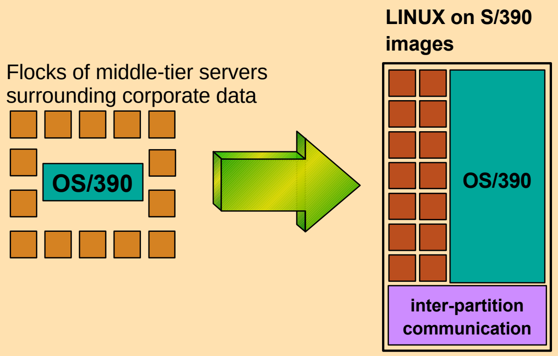
Although Sun names its partitions Dynamic System Domains, they can only "be reconfigured though software scripts (but not dynamically) as the movement of partition resources takes place under operator control (and not applications control),"[6] as described in an article on the significance of server partitioning by Giga Information Group. The article calls this type of partitioning "reconfigurable hard partitioning"[6] as contrasted with "mainframe (logical) partitioning"[6] where "Partitions have greater flexibility in that the domains can be built, resized, merged, and/or deleted under application control."[6]

Giga states that logical partitioning, as exemplified by LPARs on IBM S/390, is "considered best in class for optimal use of partition technology"[6] while Illuminata states that "IBM S/390 was the originator of much of today's partitioning technique, and its Logical Partitions (LPARs) remain the gold standard."[7]

Consolidation of middle-tier servers that access data on OS/390, VSE, or VM

An S/390 system running OS/390, VSE, or VM (or a combination) is frequently surrounded by middle-tier servers that access data on the S/390 operating system via network or database protocols and then deliver the data to clients. Linux running with OS/390 provides an opportunity to consolidate the middle-tier servers on S/390 *and* get faster access to the data.

Using S/390's **logical partitions**, you can run Linux on the same machine as OS/390, VM, or VSE. Linux can be used to quickly deploy the data already resident on S/390, accessing the data via high speed, low latency inter-partition communication. To run Linux in quantity, you can also use S/390's **virtual machines** within a logical partition to run hundreds of Linux images that access data on OS/390, VSE, or VM.


LINUX on S/390 images
Flocks of middle-tier servers surrounding corporate data
OS/390
OS/390
inter-partition communication

Flocks of logical "middle-tier" servers in the same physical system as corporate data accessed via high-speed, low-latency interpartition communication

- You can assign as many logical CPs as you have physical CPs to any partition that should be allowed to exploit all the processor capacity, if that capacity is available. If, for example, you have a 12-way, you can assign all 12 logical CPs to *each* partition you define.

It is this implementation of logical sharing on a granular level that enables S/390 to deliver resource to each partition to meet its peak demand when that peak occurs.

The total number of logical CPs that customers assign to all logical partitions usually exceeds the number of physical CPs. The number of logical CPs they assign reflects the value they feel they are receiving. When a customer with a 10-way defines four partitions and assigns 4 shared CPs to each, the customer is, in essence, stating that the 10 physical CPs are delivering 16 CP's worth of work. It is common on S/390 systems for customers to assign twice as many logical CPs as physical CPs: it is common that customers perceive twice as much CP processing power as compared to the number of physical CPs. On systems limited to physical resource sharing, the need to configure for peaks means that customers are seeing less total CP processing power compared to the number of physical CPs—often significantly less.

Sharing I/O Resources

S/390's I/O connectivity might already seem dramatic enough to support the server consolidation scenarios of today and tomorrow: 256 paths; 8 paths to a device (and those paths can be from the same or different partitions); 65,535 devices. The numbers are certainly impressive enough to indicate that S/390 can sustain a great deal of work. (See the table to compare these numbers with Sun Solaris on an E10000.)

Again, however, the key is the ability to share those resources among many instances of work, whether that be multiple workloads in one operating system image or many operating system images.

I/O Capacity	**IBM System/390**	**Sun E10000**
Physical paths to the same device from a single domain	8	2
Physical paths to the same device from different domains		0 (not supported)
Physical devices	65,535	10,000 (64 Sbus or 32 PCI I/O ports)
Virtual paths to the same device from multiple domains, via same physical path	15	0 (not supported)
Number of paths	256	32

With System/390, you can:

- Connect a device to the same partition via different physical paths. These paths are not restricted to a role of redundant or alternate paths, for use only in failover situations, as they are on most servers. S/390 will use all paths to a device (as many as 8 can be defined) during normal processing. In fact, the same path need not be used to satisfy even a single request: the request to read data can be sent on one path and the data itself retrieved via another path.

- Connect a device to more than one partition via different physical paths. For example, given a maximum of 8 paths to a single device and allowing for 2 paths to a device from a partition, you can *physically* connect a device to 4 partitions.

- Exceed physical connectivity limits by defining logical paths: a single physical path can be connected to multiple partitions by defining logical paths that resolve to the same physical path. This characteristic greatly expands the number of partitions that can access a path, far beyond any physical connection limitations.

The I/O connectivity S/390 supports enables consolidation of server farms on a single S/390 SMP *without* data replication. Either by physically connecting a single device to multiple partitions, or by defining virtual paths from many partitions, many servers can access the same I/O device. The physical limitations of the system do not force data replication.

The uniqueness and sophistication of these capabilities become obvious when compared to Sun Solaris I/O capabilities: Sun Solaris has no functions comparable to the preceding. With Solaris, you can attach a device to an alternate path on a different board in the same domain, but you cannot use that path except in a failure situation. You cannot attach a device to different domains and access it from more than one domain at the same time. To switch a device from one domain to another, you need to use a system board as a “swing board,” physically moving the board—its processors, its memory, *and* its I/O—from one domain to another to switch access to a device from one domain to another.

Sharing the Machine: Virtual Machine Technology

As seen with both CP and I/O sharing, S/390 provides impressive physical capacity and connectivity; and then enables you to seemingly exceed physical limits by logically sharing the resources. Virtual machine technology, pioneered by the S/390 VM operating system, takes the logical concept even further, beyond individual resources, to servers as a whole.

Virtual machine technology enables you to define hundreds of virtual machines on an S/390—within logical partitions or on an unpartitioned system. You can use virtual machines to run Linux production servers, maintaining Linux culture of one application per Linux virtual machine; to test new Linux environments or applications without duplicating hardware or disrupting production workloads; to provide standby systems for immediate backup of failing applications; to provide faster communication between Linux guests or between Linux guests and other operating system guests

You can also use VM’s accounting facilities to determine charge-back for clients. Using accounting records optionally produced by VM, ISPs and ASPs, for example, can establish charge-back based on actual resources used.

In addition to consolidation of multitudes of servers for multiple purposes—production, test, backup, high-speed communication—you also gain significant performance and productivity advantages through Linux’s inheritance of operational attributes from VM.

Performance:

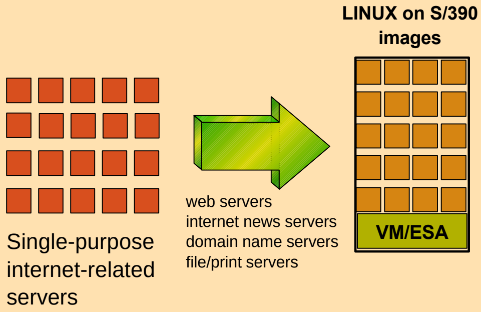
- VM’s high-performance networking supports virtual network speeds up to 340 MB per second. This can eliminate network time when Linux accesses data on, for example, OS/390.
- Data-in-memory support, provided by VM Virtual Disks in Storage and Minidisk caching, provides transparent, high-speed data access for Linux guests.

Productivity:

- The Linux minidisk driver uses VM facilities to access all devices that VM supports in a device-independent manner with complete error recovery.
- Temporary disks can be used to meet interim needs for additional Linux disk space.
- VM offers a functionally rich debug environment that is particularly valuable for diagnosing problems in the Linux kernel and device drivers.
- Extensive performance measurement, reporting, and control facilities in VM can be used to manage Linux guests.
- Facilities for virtual machine scheduling and automation can be extended to perform these functions for Linux guests.
- Because no real resources need to be dedicated to a Linux virtual machine, creating and deleting them is quick and easy.
- VM simplifies the ability to provide standby systems for immediate backup of failing applications.

Consolidation of single-purpose servers such as file/print servers, news servers, or domain name servers.

Using S/390's **virtual machines** (within a logical partition or on an unpartitioned system), you can run LINUX in quantity: hundreds of LINUX images that enable you to consolidate servers on one physical machine while maintaining the LINUX culture of one application per server. S/390's ability to connect I/O to multiple images also enables server consolidation without replicating the data for each server. And VM (the virtual machine operating system) delivers its own software qualities of service to LINUX running on a virtual machine.



Anyone investigating Linux for S/390 has likely heard of the user who started, as a proof-of-concept exercise, thousands of instances of Linux in thousands of virtual machines in one partition of an S/390 G5. What is often overlooked is the reliability that VM demonstrated in this exercise: when VM ran out of resources to allocate, it did not crash. Virtual machines, in combination with S/390 hardware reliability, provide increased reliability compared to conventional server farms, while also offering increased functionality and reduced cost.

Self-healing Servers

Self-healing is the ability of a system to recover from failures without application downtime. Downtime is becoming increasingly unacceptable for workloads that used to be considered appropriate for "good enough" servers.

In a survey done by DataQuest[8], the majority of respondents that measured the cost of downtime estimated that mission-critical system downtime cost between $50,000 and $99,999 per hour. In the past, mission-critical transactional workloads and Web workloads were different workloads. Today they are becoming one and the same as the Web becomes the new interface to yesterday's transactions. New and future workloads also will depend on the Internet: a survey of IT executives reveals that 70% of future mission-critical applications will run on the Internet[9].

S/390 once was considered "overkill" for Web workloads, which were more often relegated to servers selected strictly on the basis of price. The philosophy was that unplanned downtime was regrettable, but not worth the investment needed to guard against it. Today, unplanned downtime is unacceptable; even planned downtime must be minimized.

Eliminating Unplanned Downtime

How many mainframe-like features must a server exhibit before it delivers, not just claims, mainframe-like hardware reliability and availability? S/390 engineers are flattered that mainframes are the yardstick for high availability and reliability, but dismayed at the claims for mainframe-like robustness. UNIX and Windows NT vendors most loudly claiming mainframe quality lack not only key self-healing attributes but also lack the mainframe's philosophical approach to RAS. For UNIX and Windows NT vendors, a reliable system is a system that can be rebooted around the failed components. For S/390, a reliable system is a system that (1) rarely fails; and (2) recovers *without* the need to reboot in the event of a failure.

Sun, for example, states that "a fully redundant system will always recover from a system crash."[10] S/390 defines a fully redundant system as one that *never* crashes.

Sun also warns against the metric "mean time between failures"[10] (MTBF) because there is "no industry adopted standard for measuring MTBF."[10] S/390 is very clear about what MTBF means: the average time before a system crash requiring reboot (reIPL) or repair. For example, Sun's Enterprise 10000 detects data errors in Level 1 and Level 2 cache and, as a result, eliminates the number one potential hardware cause of data integrity exposures. But, once detected, there is no recovery; the system crashes. S/390 recovery from similar errors is virtually 100% and the mean time to system crash of an S/390 server from ***all*** hardware causes is more than 40 years, as indicated by first-quarter 2000 full-field data.

How Does S/390 do That?

Some knowledge of hardware errors is useful for understanding how S/390 accomplishes its level of hardware RAS—and for understanding to what extent UNIX and Windows NT vendors don't meet that level.

Hardware failures can be transient or permanent (or intermittent):

- A transient error (also called a soft error) occurs randomly when environmental conditions, noise, or cosmic particles cause an incorrect result but the circuit itself functions correctly. Errors in CMOS technology are predominantly environmental and, therefore, transient. A transient error can be recovered by retrying the operation. The primary difference in different systems is the ability of the system to (1) detect the error; and (2) recover dynamically and transparently from the error.

- A permanent error (also called a hard error) is an error in a circuit: the circuit no longer gives the correct output, given the same input. A permanent error requires repair or replacement of the circuit. The primary difference in different systems is the ability of the system to (1) detect the error; and (2) repair the circuit without application downtime.
- Intermittent errors sometimes produce an incorrect result, sometimes not. They can be handled as transient errors if recoverable; or as permanent errors, if the error recurs beyond a threshold and requires repair.

Distinctions in hardware RAS of different vendors, therefore, are based on their ability to:

- Detect errors at their source
- Recover transient errors
- Repair permanent errors

and to do the preceding without incurring performance penalties.

Detecting Errors at Their Source

Software does not always fail quickly when it encounters hardware errors, resulting in corrupted data. It is necessary for the hardware itself to detect errors in the hardware. Modern microprocessors usually include error checking for cache data and for data paths where data is generally moved without being altered. They do not usually include error checking in all functional elements. For example, checking of control, arithmetic, and logical functions is generally considered difficult and time-consuming.

Therefore, to understand how adequately a system addresses hardware reliability, you have to understand what it is not doing as well as what it is doing. If the logic that generates an address is not checked and produces an incorrect address, it doesn't matter if the system ensures that the address doesn't change when it is moved. S/390 ensures *computational* integrity as well as *data* integrity via extensive error checking in all functional elements. Other servers do not.

Recovering Transient Errors

The distinction between whether a server detects errors (for example, parity checks) or detects and corrects errors (for example, error correcting code, known as ECC) determines the extent to which the system can recover from transient errors. S/390 design is sufficiently robust to recover from transient logic errors; other microprocessors are not. For example, Sun's E10000 has implemented error correcting code for memory. For Level 1 and Level 2 cache, it does parity checking only: cache errors are detected but not corrected. On S/390, all data in the cache hierarchy is protected by data redundancy, provided by means of write-through cache design and ECC: errors are detected and corrected.

Error recovery implies retry, which can impact performance. The challenge is to achieve retry without performance impact. S/390, for example, ensures that any completed CPU instruction is error-free at the successful completion of the instruction execution, with *no* impact on performance. The recovery is completely controlled by the hardware. This independence from any operating system running on the hardware ensures that the benefits are delivered to Linux as well as to more traditional operating systems such as OS/390.

Repairing permanent errors

An error that cannot be successfully retried and that exceeds a certain threshold is considered a permanent error that requires repair rather than recovery. Obviously, if a system does not attempt to recover transient errors, it cannot distinguish between permanent and transient errors. All hardware errors, therefore, imply repair. On a system like Sun E10000, the failure to recover transient errors and, therefore, treat them as permanent errors, can lead to repair of healthy circuits, increasing reboots and warranty/service costs on behalf of circuits that don't require replacement.

S/390 recovers transient errors; a transient error that exceeds a threshold is treated as a permanent error. Only these permanent errors require repair and S/390 achieves that repair dynamically with dynamic CPU chip sparing. [1]

No other server provides dynamic CPU chip sparing. Some microprocessors are designated as "spares." If a running CPU chip fails and instruction retry is unsuccessful, the spare CPU chip begins executing at precisely the instruction where the other CPU chip failed. The hardware is effectively doing CPU retry across CPU boundaries. Activation by the spare is done completely by hardware, with no operating system awareness, enabling the system to be restored to full capacity in less than one second as opposed to hours. Therefore, again, Linux, as well as operating systems such as OS/390, benefit.

In addition, S/390 provides memory chip sparing [1] : an error threshold is maintained for each chip and, when exceeded, a new chip is nondisruptively substituted by hardware.

Similarly, S/390 provides cache line sparing. [1] When an error threshold is exceeded, the defective cache line can be nondisruptively removed and later substituted by hardware.

In contrast, Sun provides a function it calls “dynamic attach” on the E10000. Previously, Sun had hot-swap of logic components and a repair scenario went as follows: crash, reboot, insert new card, take system down to bring new card online. With “dynamic” attach, the scenario is crash, reboot, insert new card and bring online without an additional reboot. The improvement is one outage instead of two.

S/390 has *no* outage for over 75% of all hardware repairs and a crash for only 5%. Virtually all repairs on Sun are preceded by a crash.

S/390 effectively achieves zero outage for unplanned hardware repair. Most hardware failures requiring repair will not cause an outage at time of failure or at time of repair. “Dynamic” for S/390 means no application downtime; “dynamic” for UNIX and Windows NT vendors means the ability to reboot quickly.

Minimizing Planned Downtime

Big, complex IT installations are constantly upgrading and modifying server hardware and software. E*Trade notes over 100 planned updates per month. Yet Internet-connected systems must appear to be available 24x7. Planned updates with unforeseen problems have resulted in highly publicized downtime at E*Trade and eBay, among others.

S/390’s concurrent hardware maintenance enables repair or upgrade of hardware and microcode elements while the system remains in operation. Hardware elements such as processors, channels and power supplies can fail and then be repaired while the system keeps running. Microcode patches can also be applied nondisruptively.

Where “Good Enough” can be Better: Linux for S/390

Linux for S/390 inherits from S/390 qualities of reliability, availability, scalability, and serviceability not found on other platforms. Self-configuring and self-healing attributes of S/390 give you the mainframe hardware’s qualities of service without modifications to your Linux applications, no matter how you choose to run Linux: on a single-image S/390; using S/390’s logical partitions; in a virtual machine environment. To integrate your OS/390 and e-business workloads, you can run both environments in logical partitions on the same machine, enabling efficient communication between them while allowing consolidation of your UNIX-type workloads. When compared with the leading UNIX server, Sun’s E10000, S/390 offers more opportunity to support your e-business workloads with qualities of service enjoyed by your mission-critical application.

All of the hardware attributes described above are inherited by Linux, just as they have always been inherited by the more traditional mainframe operating systems OS/390, VSE, and VM. Linux has hardware mainframe attributes on S/390 because S/390 *is* a mainframe.

Because Linux is open-source, it is platform-neutral and, therefore, promises the best time to market for application development and deployment in today's heterogeneous environments. Linux will not incorporate platform-specific features. OS/390, developed over the past three decades specifically for the S/390, does work closely with the S/390 hardware to deliver unsurpassed qualities of service in an operating system. Some of those qualities --such as dynamic I/O reconfiguration and disaster recovery-- will benefit Linux when both Linux and OS/390 run in logical partitions on the same machine.

Other qualities remain distinctive to OS/390 itself and account for OS/390's unmatched availability, reliability, scalability, flexibility and integrity. Customers rate OS/390 as the most "battle-hardened" operating system, with a score of 8.2 out of a possible 10; variants of UNIX and Windows NT ranged from 6.4 to 6.9 [11]. OS/390, long trusted for database and transaction workloads, is also the best choice for Web applications that absolutely require the highest qualities of service. WebSphere on OS/390 provide the environments for running today's applications with qualities of service normally associated with bet-your-business database and transaction processing.

The following table lists OS/390 qualities of service described in the rest of this paper and identifies which are inherited by Linux when Linux and OS/390 are run on the same machine. The table also compares the OS/390 qualities to Solaris 8, Sun's most recent operating system, running on E10000 (not all Solaris functions are supported across the hardware line). UNIX and Windows NT vendors claim mainframe-class qualities for their operating systems as well as for their hardware. As with the hardware, mainframe-like is not the same as mainframe.

OS/390 Operating Systems Attributes	OS/390 on S/390	LINUX for S/390	Solaris 8 on E10000
Service Level Agreement management: The system can manage and track response time and transaction rates.	yes	no	no
Dynamic load balancing of network traffic: route around failures in IP stacks dynamically at runtime.	yes	no	no
Recovery of software errors: errors are isolated to minimize their impact; failing application is restarted.	yes	limited	limited
Concurrent SW maintenance: nondisruptive software updates.	yes (files also)	no	yes (files also)
Disaster Recovery: minimizing data loss and time to recover in event of disaster	yes	yes (inherited)	limited
Dynamic I/O configuration: dynamically modify the I/O configuration while the system is running.	yes	yes (inherited)	limited
Storage Management: automate and centralize storage management, according to policies set up to reflect an installation's business priorities.	yes	no	Via Veritas products
Scalable file system: File system can expand as necessary, within a single physical storage unit (volume) or across multiple units (volumes).	yes	no	Via Veritas products

Solaris's support for mainframe-like availability, from clustering support to resource management to volume and file management, are delivered in add-on products: the Solstice High Availability product, Solaris Resource Manager (bundled with the Solaris Network Bandwidth Manager), and products from Veritas. They are not integrated into the Solaris operating system. The philosophy of OS/390 is that mainframe-like qualities of service are built in from the bottom up. This approach allows IBM to deliver a single package with the features needed to drive all of your mission-critical workloads.

A Warranty for System Integrity

OS/390 has a total commitment to system integrity. Data and system functions are protected from unauthorized access, whether accidentally or deliberately with sinister intent. IBM is so confident of the system integrity of OS/390 that it provides an "integrity warranty" for OS/390 that is unique in the industry. IBM accepts, as a code defect, any means by which a program can access or modify data for which it does not have appropriate authority. Only OS/390 on S/390 warranties system integrity.

The enforcement of system integrity relies on the storage key protection mechanism provided within the S/390 hardware. Storage key protection prevents concurrently running programs from writing into storage areas that are not theirs. Hardware storage protect keys are assigned to 4K blocks of memory independently of whether the memory contains data or program code. Every program has a key assigned to it under which it executes. During execution, when a program accesses a memory location, the keys are compared to determine whether to allow or deny access. OS/390's granular program overlay-prevention is superior to UNIX (and Sun Solaris) page-level protection approaches, which typically distinguish only user from kernel content, and Read-Only vs. Read-Write access, for the specified storage pages.

Service Level Agreement (SLA) Management

OS/390's Workload Manager (WLM) provides real-time management of resources to meet your defined service level agreements. Workload management shifts the focus from tuning at a system resources level to defining performance expectations (performance goals) based on business needs. Once the goals are defined, Workload Manager continuously manages the available system resources to meet the demands of incoming work requests.

The distinction between *workload* management and *resource* management is significant. Resource management enables an administrator to control an application's consumption of a resource based on the priority of the applications. Whether that resource control actually delivers the resource required to meet service level agreements depends on the administrator's skill and ability to react quickly to system changes, the types of resources the administrator can control, as well as on numerous factors within the system and the workload. This uncertainty has led to the demotion of Service Level Agreements (SLAs) to Service Level Objectives (SLOs) on many systems.

OS/390's WLM adjusts resource usage automatically to meet performance requirements. It's faster than a systems administrator, provides better control of unpredictable and inconsistent workloads—two characteristics for which Web workloads are infamous— and, as a result, can drive a system harder.

How does WLM do that? The following are examples of the sophisticated capabilities that have evolved since WLM's precursor (system resources manager) was introduced decades ago:

- You do not have to calculate the appropriate number of server processes (address spaces) to handle the workload and you do not have to deal with fluctuations. Workload manager will start and stop server processes (address spaces) based on workload, with no manual intervention.
- You can associate a performance goal with a transaction irrespective of the various subsystems the transaction runs under. For example, a CICS® transaction that accesses DB2® data will have tasks or threads that run in the CICS and the DB2 address space. Those tasks can be associated with a performance goal for the transaction, instead of with performance goals associated with CICS and with DB2.
- You can manage resources that represent intangible qualities, such as a period of time, as well as physical entities, such as a database or device. Workload Manager, in turn, manages how some software handles queue lengths, server dispatching, and other areas sensitive to system processing.

Sun acknowledges the standard set by OS/390 when it states that its Solaris Resource Manager can control resources "using methods similar to mainframe-class systems."[12] But, like UNIX and Windows NT vendor claims for mainframe-class reliability, there is a gap between what is claimed and what is delivered. In addition to the significant difference between OS/390's workload management and Solaris's resource management, OS/390 is superior in terms of resources managed and scope of management:

- Resources Managed. OS/390's Workload Manager manages more types of system resources than Solaris's Resource Manager. Solaris Resource Manager allows CPU to be managed by allocating shares to groups of users and then allows hard limits to be set for virtual storage usage, the number of processes, and the time a user is connected to the system. (OS/390 has had similar hard limits for decades.)

OS/390 Workload Manager manages the following types of resources to achieve the goal-oriented policy an administrator sets:

- CPU
- Physical memory
- Number of server address spaces (server processes) for a a wide set of OS/390 subsystems, including DB2 stored procedures, HTTP Server, MQSeries® WorkFlow, WebSphere and Batch jobs
- I/O resources, including I/O priority and Shark Parallel Access Volumes.

- Scope of Management. Solaris Resource Manager primarily allows management of processes under the control of the basic operating system. OS/390's Workload Manager can cooperate with its subsystems to manage transactions within the subsystem even if these transactions do not have separate processes (address spaces). Workload Manager cooperates in this fashion with CICS, IMS, DB2, HTTP Server, WebSphere and MQ Series.

  Solaris Resource Manager also only manages within a single operating system image. The policy managed by OS/390 Workload Manager has a cluster-wide scope. Since its inception, Workload Manager has been involved in balancing incoming work requests across the sysplex, based on Workload Manager policy.

WLM provides an additional benefit: it supplies data that can help you plan for future capacity. Capacity planning has become another critical administration task with the exponential growth of Web workloads. WLM's data enables a clearer look into that particularly murky crystal ball.

Dynamic I/O Reconfiguration

Dynamic I/O configuration is another area in which OS/390 has had a lot of practice. You can add, delete, or modify the definitions of paths and devices and activate the changed I/O configuration while the system is running. The process also provides immediate online validation of configuration data when you define the changes.

When Linux runs on the same machine as an OS/390 system, it inherits the benefits that OS/390's dynamic I/O reconfiguration provides.

Linux can use devices that are defined dynamically to OS/390 when both Linux and OS/390 are running on the same machine. Dependent on the type of device, Linux will recognize changes in the I/O configuration.

Disaster Recovery

For some workloads, mission critical includes the requirement to recover quickly in the event of a disaster, such as a major fire, earthquake, terrorist bombing, or other catastrophe. Three criteria are important in disaster recovery:

- How much data is lost.
- How quickly the data can be recovered.
- The effect of data backup on your system's availability.

You can use OS/390's disaster recovery to protect Linux data as well as OS/390 data when Linux runs on the same machine as an OS/390 system.

On most platforms, you have no choice but to run expensive backup jobs at planned intervals (data since the last backup is lost in event of a disaster); freeze the data while you're copying it (which affects your system's availability); and manually send the data to a remote location (which affects how quickly the data can be recovered).

OS/390 supports two levels of disaster recovery, both of which provide superior disaster recovery compared to almost all systems.

- The first level minimizes data loss and the effect on system availability but still requires sending physical backup copies to a remote location. Using Concurrent Copy, you can copy the data while it is being written to your database.

  Through close interaction with disk hardware, Concurrent Copy provides frequent full-image copies of data, which can be physically transported to the disaster recovery site. It allows the data to be copied while users and applications continue to have access to and modify the data.

- The second level minimizes data loss and any effect on system availability while maximizing how quickly the data can be recovered. OS/390 can copy important data automatically at specified frequencies to large disk drives located remotely, using either:

- An asynchronous model: little impact to performance; minimal data loss if failure occurs while the data is in transit between the two locations; unlimited distance between sites
- A synchronous model: data committed to backup before completing the primary write (some performance impact) no data loss; up to 103 Km between sites.

  Both functions write your data to a remote secondary disk volume in addition to your primary disk. If any failure occurs on your primary volume, you can recover up-to-the-minute data from the copy. And that copy can be many miles away, allowing you to have a remote site for disaster recovery purposes.

  If your OS/390 system is participating in a cluster (a Parallel Sysplex® environment), data can be copied every time it changes. This approach, called a Geographically Dispersed Parallel Sysplex™, can be set up to ensure no data loss, with full recovery in less than within 60 minutes.

Because disaster recovery approaches respond to changes on the physical I/O device, OS/390's disaster recovery can be used to protect Linux data, as well as OS/390 data.

Robust Recovery Routines

OS/390 has a sophisticated approach to error recovery by which each part of the operating system can intercept a failure, collect diagnostic data (without bringing the rest of the machine down), and terminate that process. The process can be restarted to reestablish the function that failed. Meanwhile, there is no impact to the remaining workload running on the system.

In general, Sun Solaris 8's failure recovery approach is immature compared to OS/390. Solaris 8's recovery strategy is to reboot. If Solaris 8 itself fails, it writes a log entry of the failure to disk. In some cases, Solaris may attempt to recover from the error, but the recovery only goes so far, and then the system "panics" —it may collect a memory dump—and automatically reboots.

TCP/IP load balancing

OS/390's TCP/IP support also adheres to OS/390's definition of "dynamic" as "at runtime" and again demonstrates the power of logical functions to achieve dynamic changes without disrupting applications. OS/390's TCP/IP load balancing routes around failures in IP stacks on a system image dynamically *at runtime*. Sun's load balancing after failures is done *at initialization*.

OS/390's Approach to Recovery

Recovery Termination Management (RTM), a component that made its debut with MVS™ (the grandfather of OS/390) in 1974, completely changed the paradigm for diagnosing errors. Before then (and even today on other platforms), an application that encountered an error failed and, frequently, brought other work and the system down with it. Then debuggers went to work, looking at the final state of the system for clues to the failure.

RTM changed all that. A customer, back in the 1970s, described the new MVS paradigm for error diagnosis with the following metaphor.

The ERROR: A truck traveling too fast on a wet road comes to a curve and skids. The truck crashes through a guard rail, plows through a field of corn, crashes into a tree, and flips over onto its roof, its wheels still spinning.

The RECOVERY: RTM comes along and rebuilds the guard rail, replants the corn, and prunes the broken branches from the tree and carts them away. All without disrupting other traffic on the road.

The DEBUGGER finds a dry road on a sunny day with traffic flowing smoothly. But in the distance, beyond a corn field, lies a truck on its roof next to a tree. Its wheels are no longer spinning.

To be more accurate, RTM intercepts detected failures and calls recovery routines to take the recovery actions. All OS/390 components and subsystems provide recovery routines. And the application itself can provide recovery routines that RTM will call. Those recovery routines can put the truck back on its wheels (cleanup) and back on the road (retry). Recovery routines also provide a picture of the truck crashing and even enable the debugger to control the camera angle, providing the debugger with the data needed to address the original problem.

And, with a Parallel Sysplex environment (OS/390's clustering technology), everything the truck was carrying will continue on its way on another truck, without disturbance or disruption.

In addition, OS/390 TCP/IP availability features are key to making stack and system failures invisible to applications. OS/390 supports virtual IP addressing, which allows the definition of a virtual address to represent multiple network interfaces in a single system image. VIPAs can provide recovery that is transparent to end users and applications for certain protocols. OS/390 also provides a dynamic VIPA takeover capability, which allows VIPA addresses to be moved from one TCP/IP stack to another (in a multi-stack environment), *without* knowledge of the application. With Dynamic VIPA Activation, the virtual address may be activated by the application binding to that address, and the VIPA is automatically deactivated when the application ends. This can be very handy when moving an application from one partition to another with consistent connectivity. The TCP/IP connections for a given application are quickly reestablished when a TCP/IP stack suffers a failure.

Sun also supports a form of virtual IP addressing, but it is physical, not logical: the VIPA represents addresses on different physical network interface cards that are part of a domain. Although Sun can also provide transparent recovery for certain routing protocols, it has no function that matches OS/390's dynamic VIPA takeover.

Concurrent Software Maintenance

The OS/390 operating system running on S/390 enables nondisruptive upgrades. You can install a new software version to a disk that resides on (or is connected to) a running system, and activate the new version on the next restart of the system. Recently, Sun Solaris 8 introduced new features called Live Upgrade and Live Update. Live Upgrade enables a new operating system version to replace a previous version when the system is rebooted. Live Update provides for updating the kernel operating system code while it is running. Both are more memorable names for functions that other platforms, including OS/390, have been doing for years.

OS/390, all of its middleware, and applications also go a few steps further. Certain file libraries (like big directories) can be refreshed on the fly any time during system operation. Rather than restarting an entire operating system, which could affect the availability of many users, parts of the operating system, middleware or entire applications can be restarted while the operating system is running, bringing in the new version of the software automatically.

System-Managed Storage: Self-Configuring Attributes for Storage Management

As your enterprise grows, the need for storage media to hold applications and data increases, as does the cost of managing that storage. The cost of storage hardware is part of the equation, but the highest cost is people time needed to do storage management tasks (regardless of platform). Updates may not be possible during the working day; they may need to be done during time windows that exclude running transaction systems, which also drives up cost. The ability to eliminate many of these tasks extends the self-configuring features of OS/390.

OS/390, together with IBM hardware products, automates and centralizes storage management, according to policies set up to reflect an installation's business priorities. The policies specify classes of space management (e.g., formats, compression, migration), performance (e.g., minimize I/O delays), and availability (e.g., nondisruptive backup). Using naming conventions specified in the policy, classes are automatically assigned to data files throughout the life of the data - when the data is created, after it's been stored, during backup or migration of the data. The data is stored and cataloged automatically, whether stored on disk, optical or tape resources, so that it can be quickly identified and retrieved. Distributed data access allows authorized systems and users in a network to exploit the automated storage management provided by OS/390.

Scalable File System

Almost everyone working in a UNIX environment is familiar with the frustration when the file system runs out of room, requiring storage cleanup measures to locate space. Rather than putting the file system in a single storage partition, as other UNIX systems do, OS/390 bases its Hierarchical File System (HFS) on a data set (file) model, allowing the file system to grow as necessary when out of space. A "secondary extent" is defined as the amount by which the file system can be automatically expanded, and is specified as one of the normal attributes maintained for all operating system datasets. When a file system is managed using Systems Managed Storage, it inherits file scalability with even greater flexibility—secondary extents are not limited to the initial disk (volume) and the file system is allowed to expand to other disks, all managed automatically without human intervention.

Systems Managed Storage also supports the ability to migrate and recall a file system. When a file system remains unused for a specified period of time, the system can copy it to another location, allowing the physical disk to be used by another application. When the file system is needed, it is automatically recalled and set up for normal usage. This feature allows the physical storage for the file system to be managed efficiently, based on installation policy.

Beyond Single-Server Consolidation

Clustering eliminates single points of failure through redundancy in both hardware and software. OS/390's clustering support, called Parallel Sysplex, distinguishes itself for its ability to balance workload across the cluster, provide continuous operations, and deliver very high availability.

Workload balancing in OS/390's cluster environment is achieved through:

- Data Sharing: Multiple instances of an application running on multiple systems (physically separate or in partitions) can work on the same databases simultaneously.
- Workload Manager: OS/390's state-of-the-art Workload Manager balances work across a cluster as well as within a single system.

Workload balancing is as valuable in update scenarios and production workloads, as it is for component failures:

- Hardware and software changes can be accomplished without disruption by removing the system that needs to be changed from the cluster (while the applications continue to run on the remaining systems), making the change, and returning the system to the cluster. Workload balancing ensures that the work being done on the removed system is distributed across the remaining systems in the sysplex.

  Software upgrades can be propagated around the cluster while fully supporting all business workloads and associated data sharing on the remaining systems. In doing so, varying levels of software (operating systems, middleware, etc.) operate in the sysplex during the upgrade period, which may last hours, days or months, depending on installation requirements.
- Production work is balanced across systems based on system capacity, as well as availability of a system within a sysplex. Workload manager includes system capacity as a factor in many system and middleware queuing, routing, and scheduling decisions. It also assists in determining on which system to restart an application that was running on a failed system.

These features, along with a high speed automatic restart capability, are key in delivering high availability and continuous operations in OS/390. The resulting continuous availability that OS/390 users enjoy is driven through strong integration of software in the operating system, communications server, transaction managers and database managers, along with exploitation of the S/390 hardware platform.

Cluster in a Box

Instead of building a Parallel Sysplex environment with two or more interconnected mainframes, you can build a Parallel Sysplex environment using separate logical partitions within one mainframe—a cluster in a box. A single-mainframe sysplex, unlike a standalone server, is tolerant of software failures. An internal coupling channel uses processor microcode to communicate between the partitions, achieving performance that supports production workloads as well as test workloads. With the internal coupling channel, enterprise applications can take advantage of the hardware fault tolerance of a standalone server combined with the software fault tolerance of a distributed Parallel Sysplex.

When "Good Enough" is not Enough: OS/390 on S/390

Although other vendors compare themselves to mainframes, their ultimate claim is that they are good enough for certain workloads. None claims to match the qualities of service of OS/390. Some workloads demand more than good enough. OS/390 meets the demands of those workloads.

References:

1. Mainframe attraction: Large-capacity server vendors positioning for mainframe market, Telephony, January 26, 1998

2. Compaq to Advance Standards-based Computing to Highest Levels of the Enterprise, Business Wire, January 26, 1998

3. UNIX: the Next Generation, Information Week, May 4, 1998

4. Unisys Aims For The High End, Information Week, May 18, 1998

5. Choosing Server Platforms, GartnerGroup, 1999

6. Consolidation in the E-Business Enterprise (Part 2): Significance of Server Partitioning, Giga Information Group, December 28, 1999

7. Enterprise Servers on Parade, Illuminata, Inc., March 27, 2000

8. Mission-Critical Support: What End Users Really Want, Dataquest, November 30, 1999

9. International Demand Assessment and Requirements Tracking Study Spring '99, The Standish Group, 1999

10. Ultra™Enterprise ™ 10000 Server: SunTrust ™ Reliability, Availability, and Serviceability Technical White Paper, Sun Microsystems, Inc., April 1996.

11. Show Me the Benchmarks, Merrill Lynch analysis, June 23, 1998

12. Solaris Resource Manager™ 1.0 White Paper: Controlling System Resources Effectively, Sun Microsystems, Inc., 1998

Endnote:

[1] CPU (processor unit) chip sparing was first introduced in the S/390 9672 Parallel Enterprise Server, Generation 3, in 1996; and became transparent, occuring without operator intervention, in Generation 5 in 1998. Memory chip sparing became available in S/390 9672 Parallel Enterprise Server, Generation 3, in 1996. Deletion of a cache line was introduced in the S/390 9672 Parallel Enterprise Server, Generation 4, in 1997; deletion with relocate, in Generation 5 in 1998.

IBM Corporation
Marketing Communications, Enterprise Systems Group
Route 100
Somers, NY 10589

Printed in the United States of America, 07/00
All Rights Reserved

This publication was produced in the United States.
IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the products or services available in your area.

You can find additional information via IBM’s World Wide Web server at http://www.ibm.com.

IBM Hardware products are manufactured from new parts or new and serviceable used parts. Regardless, our warranty terms apply.

Actual performance and environmental costs will vary depending on individual customer configurations and conditions.

By “Year 2000 ready”, IBM means the capability of a product, when used in accordance with its associated documentation, to correctly process, provide and/or receive date data within and between the 20th and 21st centuries, provided that all products (for example, hardware, software and firmware) used with the product properly exchange date data with it. IBM’s statements concerning the Year 2000 readiness of its products are Year 2000 Readiness Disclosures under the Year 2000 Readiness Information and Disclosure Act of 1998.