Disaster Tolerant Unix: Removing the Last Single Point of Failure

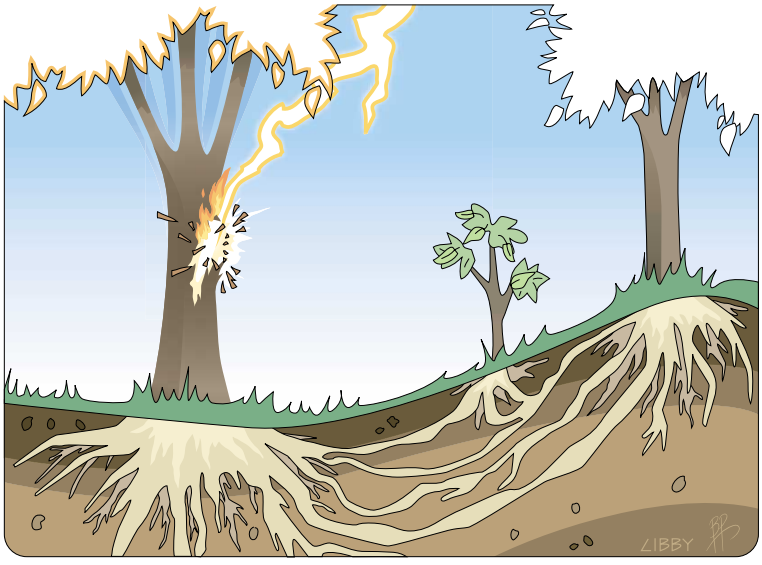
Research Note

David Freund
9 August 2002

Nearly every Unix server vendor touts its High Availability (HA) features, often focusing on how they "eliminate single points of failure." High availability is achieved by providing redundant components; if one fails, another part is still available to do the job. But a "solution" to an IT problem frequently just moves the problem elsewhere.[1] Even if all of a server's parts have been given backups, the server itself remains a single point of failure in the user's application infrastructure.

Clustering can eliminate *that* failure point by providing redundant servers. But, even with all the creative use of HA technology in a datacenter, from Uninterruptible Power Supplies to redundant storage arrays and clusters, the single point of failure is merely shifted again—to the datacenter itself. And for businesses that simply cannot survive without some or all of its IT services, that failure point must also be eliminated. The IT function must survive a disaster that damages or destroys the entire datacenter.

Disaster Tolerance (DT) is the ability of an IT organization to maintain ongoing productive operations even in the face of catastrophe. DT may include Disaster Recovery and Business Continuity products and services, but differs as a concept from them. Business Continuity is a broad term that encompasses many categories of IT disaster planning, both systemic and organizational; Disaster Recovery focuses on bringing IT back online after a disaster. DT's main goal is to build enough redundancy into IT systems to eliminate the datacenter as a single point of failure. Should one datacenter go offline, DT rapidly shifts operations to a redundant datacenter so that applications can "just keep running." Such protection is complex and expensive, but when a service or application is truly mission-critical, DT is often the way to go. This report compares the DT options available from the top Unix server vendors. But first, we must precisely define what Disaster Tolerance is—and what it isn't.

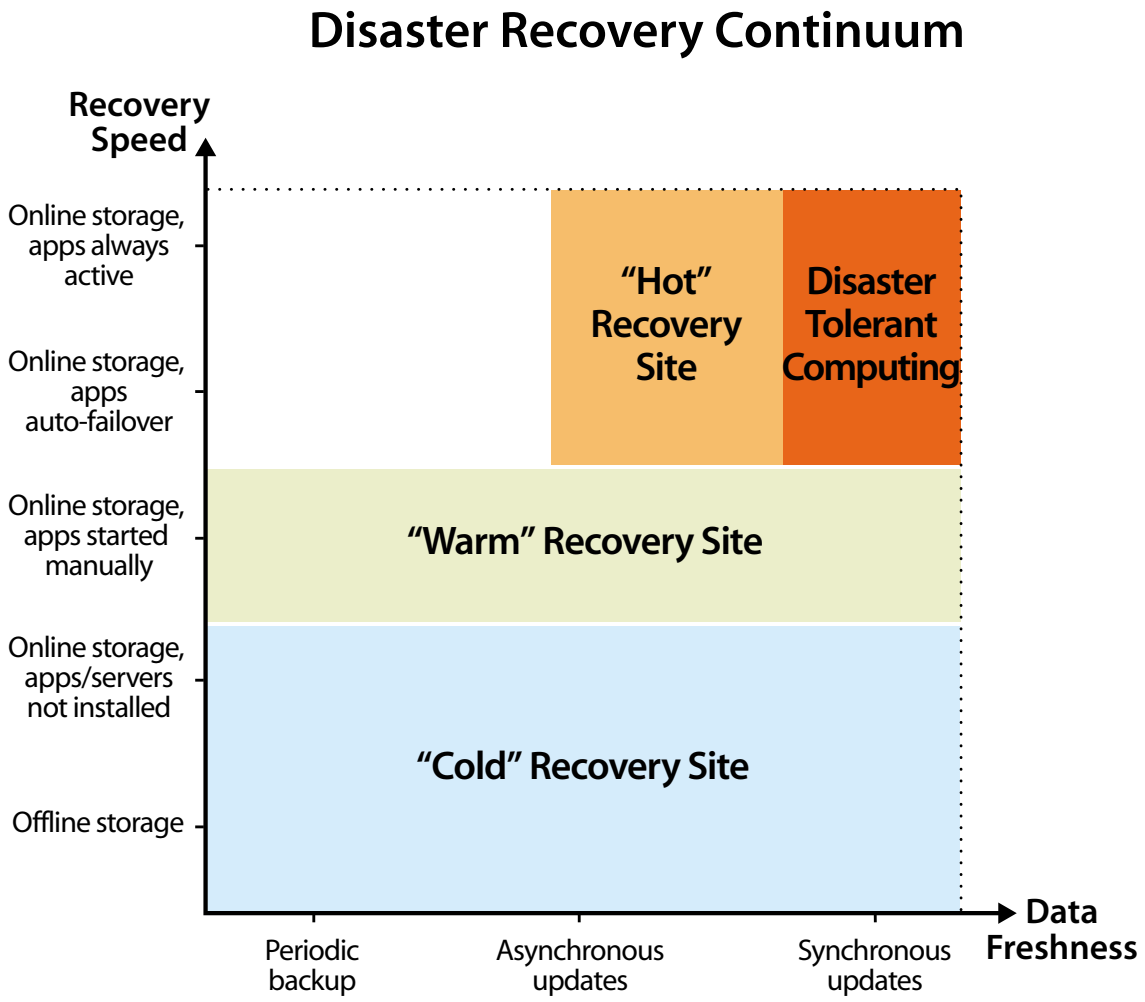


1. Just ask anyone who has spent time chasing performance bottlenecks!

Degrees of Survival

There are many terms used to describe the concept of keeping a business operating despite the occurrence of a catastrophic event. The ones used most often, and sometimes misleadingly, are:

- **Business Continuity,** the broad term that encompasses all of the plans, resources, and actions used to ensure the continued operation of a business in the event of a site-wide outage, whether planned or unplanned.
- **Disaster Recovery,** the ability, plan, or techniques used to return IT to an acceptable level of operation after a site-wide outage. This *may* include the use of servers, software, storage, networking, and staff duplicated at a remote site. Though also a broad term, Disaster Recovery is a subset of Business Continuity. Disaster Tolerance is a subset of both.
- **Disaster Tolerance,** a specialized form of High Availability and Disaster Recovery. The ability of an organization's IT facilities to rapidly *continue* after a site-wide outage by *automatically* transferring IT operations to duplicated resources at a remote site. A Disaster Tolerant system must be able to detect the failure of a primary system, notify the humans in charge, and (when authorized) proceed with a failover to redundant systems and sites without further manual intervention. DT systems should be able to make that change on their own, though many companies prefer to have a human make the final decision to declare a site-wide outage.



Disaster Recovery techniques include two dimensions of disaster recovery—the recovery time (how quickly the IT service becomes active again) and the recovery point (how fresh the data used by the IT service is). Disaster Tolerance denotes the top end of the range, with automatic failover of components to the surviving site using up-to-the-second data.

Building Disaster Tolerance

A Disaster Tolerant configuration must provide near-immediate[2] resumption of application services using up-to-date data. As with any form of High Availability, this is done using redundant components. The obvious starting point is a second site with servers, networking, and storage capable of taking over the work of the primary site. Naturally, applications and data must be copied there as well. But the crucial element for Disaster Tolerance is time—or rather, the elimination of its effects. If the primary site fails, the backup must be running within minutes, without having lost any of the most recent business transactions or other updates at the primary site.

Clustering and disk mirroring, two methods used to keep data available within a datacenter, can also be used to reduce the risk that a computing site will become an IT operation's a single point of failure.[3] But building a DT cluster means being prepared for emergency conditions and situations that are extremely unlikely in a normal datacenter. Hundred-year floods are by definition rare—but that is *exactly* the kind of disaster for which DT installations are designed, and for which they must be prepared.

2. Typically in the range of some number of minutes. What is an acceptable recovery time varies by customer and application. But anything approaching an hour or more would not be considered Disaster Tolerant.
3. While other forms of disk redundancy such as RAID 5 are frequently used to protect data within a datacenter, only mirroring (also known as RAID 1) can be used across multiple sites with acceptable performance.

Before comparing the available Unix DT options, let's examine a highly successful non-Unix model: the Disaster Tolerant OpenVMS Cluster. Still considered the "gold standard" in commercial clustering, it's not unusual for OpenVMS Cluster uptimes to be measured in years.[4]

Goals include:

Keeping the applications running. Using a built-in distributed lock manager and cluster file system, OpenVMS provides a "shared everything" cluster system. Applications running on as many as 96 cooperating servers can write simultaneously to the same files on shared disk volumes. Disk volumes are either directly attached to all nodes or, alternatively, "served" by one or more systems to the rest of the cluster. Should one or more of these servers go down, the data remains safe and sound; the applications can keep on running, or be launched onto, the remaining servers.

Keeping a fresh data copy. OpenVMS also includes "Volume Shadowing," a host-based disk-mirroring product[5] that shows applications a set of virtual devices that mimic disk drives. As data is added or modified, identical copies of the data are written to multiple physical disk drives, collectively called a "shadow set." Write operations are *synchronous;* the operation is acknowledged as complete to the application when all shadow-set members have been updated. This operation is transparent to the applications and other servers.

Because Volume Shadowing is implemented as a part of the cluster-conscious base OS, shadow-set members do not need to be attached to the same server—they can be served remotely by other cluster nodes.[6] Tight OS integration also enables

4. An example was posted to the "comp.os.vms" newsgroup on 15 Feb, 2002: "This cluster has been operational since 31-JAN-1993 04:12:54.36"
5. This is now known as software-based RAID 1. But Volume Shadowing first shipped in 1985—two years before the term RAID (Redundant Array of Inexpensive Disks) was even coined.
6. Reads always get their data from a locally-attached disk if possible, while writes unconditionally go to all shadow-set drives.

Volume Shadowing to provide *symmetric* updating—applications running on any cluster node can write to the volume with identical performance and be guaranteed that all nodes will have an identical view of the data.

However, disk mirroring does not by itself protect against application failures. If an application updates only a portion of a given set of data before failing, it will access that same partially-updated set wherever it is restarted in the cluster. While the OS can ensure file system integrity, guaranteeing that the data is consistent at the application level is the responsibility of middleware or the application. Consistency checking and recovery techniques such as journaling are standard features of today's commercial databases, for example.

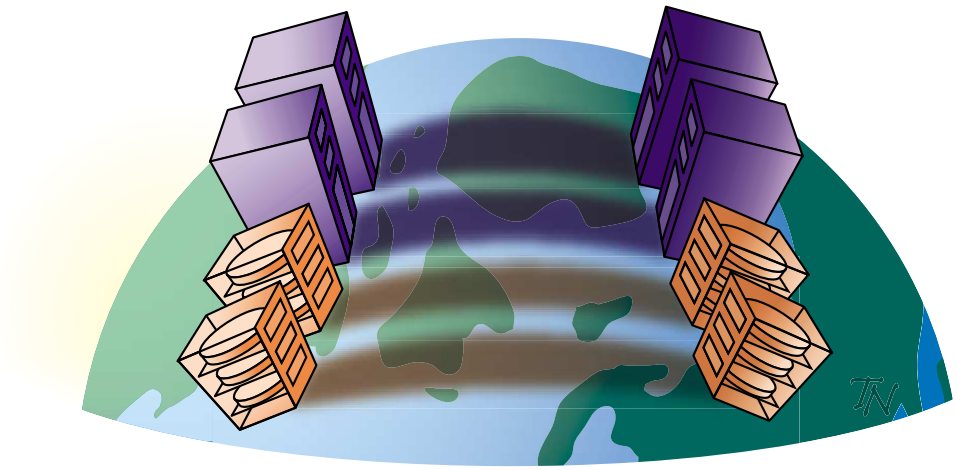
Delivering both across multiple sites. Two geographically-separated data centers can be combined into a single OpenVMS cluster. The ability to ensure that applications and data will be available across the entire cluster, even though it is distributed among several sites, has been the foundation for Disaster Tolerant OpenVMS. Applications can run simultaneously in one or both data centers, with data mirrored symmetrically and *synchronously* across the two locations. If either site completely fails, the cluster can continue operating with both a short recovery time and the freshest-possible data—which, together, deliver the current practical implementation of Disaster Tolerance.

No Free Lunch

But the safety of a distant site[7] doesn't come for free. Because with every application update to disk, data must be written in both locations—network links between the sites must be of high quality, high bandwidth, and low latency. Each mile of ATM or DS3 cable distance adds approximately 0.01 milliseconds of latency.[8] Since a remote disk-write operation requires two round trips over the interconnect, the average write I/O over a 100-mile link takes at least 4 milliseconds longer than the same I/O over a short, local network. So customers implementing a multi-site cluster—regardless of OS or hardware platform—must consider the performance implications when choosing the second site.

In addition, as site-to-site distance increases, available bandwidth decreases. When two sites are first connected, the data to be mirrored must be copied in full from one to the other. A terabyte can be copied over a local Fibre Channel[9] in about 3 hours. Over a 155 megabit/sec OC3-ATM link, that same copy takes approximately 26 hours. Using 44

7. Basic OpenVMS supports links up to 150 miles; the limit extends to 500 miles with the optional "Disaster Tolerant Cluster Services for OpenVMS" product/services package.
8. The modest latency added at each end by a WAN-to-LAN bridge is generally insignificant compared to the total delay of inter-site links.
9. Using a full-duplex 1 gigabit connection.



megabit/sec T3-ATM links increases the time to about 91 hours, or almost four days. If the two sites become disconnected from one another, either deliberately or due to equipment or line problems, large amounts of data will need to be re-copied when the sites are re-joined. Various techniques can be used to avoid performing a full copy every time this happens, but as more time passes between the breaking and restoration of inter-site links, more data will need to traverse those links to resynchronize the mirrored volumes.

Using redundant links from multiple carriers can reduce the probability of an inter-site communications failure, but can't prevent it. Breaking the link between the two should not bring either site down, but can lead to a "split-brain problem," in which each site uses its full copy of the applications and data to continue operating, creating data that is inconsistent because neither site can update the other.

To prevent the split-brain issue,[10] OpenVMS uses a "quorum" strategy. Every cluster node gets some number of "votes." No group of cluster members can continue to operate if the total number of "voters" present and accounted for is less than a threshold.[11] This rule ensures that only one instance of the cluster can ever be running at any one time. If each site is assigned an equal number of votes and all inter-site links fail, both halves would suspend operation until quorum is regained—either by the reconnection of the two sites, or by operator intervention.

And that poses a dilemma. To truly *tolerate* a disaster, the cluster must be able to automatically sort things out and keep running. By giving one site more votes than the other, one half of the cluster could be designated as the one that continues operating by default in the event of such a communications failure. Choosing the backup site would seem the intuitive choice; after all, we would want the backup to automatically take over if the primary site is lost. However, a total inter-site connectivity failure would result in IT operations halting at the primary site and failing over to the backup—a potentially gratuitous disruption, however brief, of normal business operations. Configuring the primary site to continue by default would prevent that, but manual intervention would be required to get the backup site to take over for a lost primary. Avoiding the need for manual intervention, however, is a key feature of DT computing.

But designating *either* site as a "dominant" or default survivor renders the IT infrastructure vulnerable to a more sinister problem rooted in a common myth—that disasters are sudden and total. Many disasters come on gradually, but can often cause more problems than those that are sudden and calamitous. For example, if a fire were to break out in the site designated as the "survivor" or dominant site, it would eventually cut off communications; the designated "non-survivor site" would go offline and become dormant, even though it was undamaged. The survivor site, on the other hand, would continue to operate until it was completely destroyed, losing every record of all the transactions processed during the fire in the process. This failure scenario, known as a "creeping doom," could affect either the primary or secondary site, whichever is designated the survivor. Neither designation is safe.

The solution? A technique used to solve a similar problem for clusters with only two nodes: Use a physically separated third-party to break any tie and determine which site should continue operating. Two-node clusters often use a "quorum disk," a disk drive that is directly connected to both nodes and is given a "vote" for quorum-counting purposes. A "quorum site" for DT clusters can contain just enough nodes in a third geographic location to act as the "tie breaker" in the event of a connectivity break—or a true site failure in either location.

10. In VMS lingo, a "partitioned cluster." Not to be confused with the partitioning of a single server, which is commonly used in Big Iron servers today.
11. The threshold is "one vote more than half of the total possible votes."

It turns out that, at the time of the attack that destroyed the World Trade Center towers in New York, several businesses with their primary data-centers in the Trade Center towers had OpenVMS Disaster Tolerant clusters with backup sites located out of the area. Every one of them had their operations running just moments after the catastrophe.

DT Functions: Variations on a Theme

It's the combination of clustering and data mirroring that forms the basis of a Disaster Tolerant computing platform. Of course, there's room for variation. For example, clustering need not use a "shared everything" approach. But there are implications of these different designs that must be considered when evaluating options for protecting Unix-based services.

Database Replication is a software-based technique to keep a duplicate copy of a database up-to-date. Replication ranges from manually applying redo logs to automatically repeating transactions made on the primary database. For example, Oracle9i Data Guard can be configured to guarantee that transactions made on the primary instance are not acknowledged as complete until they are confirmed on a backup instance. But this protects only the database contents; the failover of database services to the backup instance is beyond the scope of such products. Furthermore, it provides no protection for applications, nor data not stored within the database. Database replication has an important place within a Disaster Recovery strategy, but it is generally not the key to Disaster Tolerance.

Storage Mirroring uses a duplicate storage configuration: typically a RAID array controller and attached disks, which communicate over a private communications link. Writes made on a local array are replicated at the block level onto the remote array. The fact that there is another array at the far end of an inter-site link is hidden from the locally attached server. Examples include EMC's Symmetrix Remote Data Facility (SRDF), HP's Data Replication Manager (DRM), and Hitachi's TrueCopy.[12]

Most products of this type provide *asymmetric* updates—changes are copied in only one direction. A volume on one end of the inter-array link is designated as the "source" and a volume on the other end as the "target." The source volume is mounted by the locally attached server and used like any other disk volume. The target volume is only used to store a duplicate copy; it is not mounted by any server. All updates on the target are made as a result of changes that occur on the source. During site failover, scripts on the backup server reconfigure the local array to sever any link to the "source" array and allow local applications to use the "target" volume. Hardware-based products generally perform better than their software-based counterparts if the inter-site storage link is fast enough. A physically separate link can even increase this performance advantage, but often comes at the expense of increased management complexity.[13]

These products usually offer both *synchronous* and *asynchronous* updating. Synchronous updates require both the source and target be written to before either acknowledges to the application that the write process has completed. Asynchronous updates only need to complete the write locally, updating the target later on. For complete disaster tolerance, *only synchronous updating should be used* to ensure complete and consistent data at the backup site.

The threat of "split brain" corruption using this class of disk-mirroring products is no different compared to host-based mirroring. Even if the array controller has features to prevent a host from mounting a target volume without first disabling the remote copy and/or changing the volume's role, the fact that a host server can force such a state

12. More information on data replication can be found in the following Illuminata reports: *Keeping the Family Jewels: Remote Copy DR* (Sept. 2001), *PPRC Goes the Distance* (Jan. 2002), *Storage as Risk Management* (Feb. 2002), and *Distribute the People, Not Just the Data* (April 2002).

13. Any solution that uses physically separate cluster and storage interconnects between sites must be able to handle cases where only one of the links fails.

change without human intervention makes it vulnerable to this type of error. So it's just as critical that the clustering software prevent a split brain situation from occurring.

Like multi-site clustering, **Stretched SAN** configurations use disk-bus extenders or special-purpose Fibre Channel routers to provide a direct connection to disk devices in two or more sites separated by a relatively short distance—up to about six miles.[14] In this type of SAN, any server can access a disk device directly in either site. Host-based mirroring can provide data redundancy across the two sites.

As with array-based mirroring, the threat of split brain corruption remains when using a stretched SAN. But in fact the threat is increased, because there are no inherent mechanisms to prevent split-brain servers at one site from writing to volumes at either site.

Disaster Tolerant Unix

While still years away from the OpenVMS "gold standard," Unix clustering has been coming of age in recent years. All of the major Unix implementations have a failover capability, some have developed a Cluster File System (CFS), and two of them—Tru64 UNIX and Solaris—have made clustering an integral part of the OS kernel. While none has built-in remote mirroring, all can use some combination of the hardware or software mirroring techniques described above. But integrators or end users must spend additional time to integrate these components to create a really Disaster Tolerant Unix cluster with application and storage failover across multiple sites.

Each of the top three Unix server vendors (HP, IBM, and Sun) have an array of options for providing a Disaster Tolerant cluster for their Unix server platforms. All three prefer to use their professional services divisions to lead engagements that custom-tailor a solution to fit a customer's request. The high degree of professional services and customization involved complicate the comparison of each vendor offering with its competitors. Yet the truly Disaster Tolerant offerings from all three tend to use the same basic techniques.

HP's Tru64 UNIX

HP's Tru64 UNIX has an integrated clustering capability called TruCluster Server that provides application-failover, a Cluster File System (CFS) with shared-write access to volumes from multiple nodes, and support for both Ethernet and low-latency Memory Channel as cluster interconnects. Using the same techniques as OpenVMS, Tru64 UNIX includes a quorum-based algorithm to try to avoid split-brain situations.

It also offers volume-level hardware mirroring using HP's Data Replication Manager (DRM) product. DRM uses an asymmetric, peer-to-peer remote copy function in the StorageWORKS HSG80 array controller, together with CLI-based tools that run on the server, to control array functions over Fibre Channel. Both synchronous and asynchronous updates are available (though only synchronous updates are applicable to Disaster Tolerant TruClusters). While host-based mirroring is available from HP for Tru64 UNIX, it's only supported for use within a single datacenter.

Disaster Tolerant TruClusters can span "campus" distances of up to 6 kilometers (about 3.7 miles). The distance constraint comes from the cluster interconnect; the maximum distance is the limit of Memory Channel II.[15] Since TruCluster systems can use a proxy (or "served") mechanism for granting file-structured shared-disk access to the rest of the cluster, applications running in either site can access a given DRM disk unit over the cluster interconnect.

HP provides a full suite of scripts that run on the Tru64 UNIX server to automate several specific DRM failover and fallback procedures. The

14. Very long distance GBICs exist with distance ranges up to 100 kilometers, but for disk storage they are limited to use by array-based remote mirroring.

15. An Ethernet-based cluster would be limited to just 200 meters.

company recommends, however, that a human decision be required to initiate such significant actions such as datacenter failovers. Once decided, the steps involved to sever the peer-to-peer data copy relationships, change volume roles, or re-synchronize data during site re-integration can be performed automatically. Integrated with the TruCluster failover scripts, file system and application takeover can proceed in lockstep with the DRM configuration changes.

By adding the company's product/services package, Disaster Tolerant Cluster Services (DTCS), the cluster administrator can control all of the DT cluster components from a single management console, using tools developed by HP and management-software partner Heroix. The management-software suite gathers and distributes DT-specific events; lets operators remotely control systems, air conditioners, power controllers, and other devices with remote interfaces; and gives the cluster admin a consolidated view of the entire configuration within a GUI framework. The console can also be accessed remotely using a Web browser.

A built-in rules engine determines when a significant event occurs—such as a series of warnings from remote storage controllers that they are losing power, followed by a loss of connection to those units. It then provides the administrator with step-by-step, semi-automated procedures to failover the site. Also included is software to help re-integrate the sites when the failed location is ready to return to service.

HP plans to extend the distance and capability of Disaster Tolerant Tru64 UNIX configurations by the end of 2002.[16] Instead of a single, multi-site cluster the new offering will add another layer of failover protection over multiple clusters using IP-based inter-site links.

16. HP is currently migrating TruCluster capabilities into future releases of HP-UX, with the intent of melding the best of both cluster products. You can bet they will pay particular attention to Disaster Tolerance as they do so.

HP's HP-UX

HP-UX options are largely based on its clustering product, MC/ServiceGuard, which provides automated application failover in a "shared-nothing" style cluster—which allows only one server to perform I/O on a file-structured disk at a time. Since MC/ServiceGuard has no cluster file system or distributed lock-manager traffic between nodes, there is no need for a special cluster-optimized interconnect. Distance limits are largely determined by the disk-interconnect and replication systems installed. HP breaks down its multi-site failover products into three classes:

- **Extended Clusters** are designed to support two datacenters located up to 100 kilometers[17] apart by combining MC/ServiceGuard with a stretched SAN and disk-mirroring software such as MirrorDisk/UX.[18] Each site must have an equal number of cluster members. Split-brain problems are a real threat, because any "lock disks" (somewhat similar in concept to the OpenVMS quorum disk) must be in both sites. This can be mitigated by the use of an "arbitrator site" that acts as a tie-breaker similar to the OpenVMS and Tru64 UNIX quorum site.
- **Metropolitan Clusters,** or MetroClusters, are also based on MC/ServiceGuard, but use synchronous, asymmetric data mirroring that is strictly hardware-based (either HP's Continuous Access XP on its XP Series disk arrays or EMC's SRDF). This setup increases inter-site distance to 50 kilometers using HP's Continuous Access XP, and 100 kilometers using SRDF.
- A **Continental Cluster** is not a single MC/ServiceGuard cluster. It is a combination of array-based mirroring and special tools that

17. This configuration used to be called a "Campus Cluster," with a 10-kilometer limit due to Fibre Channel being stretched using long-wave switches. Using Dense Wave Division Multiplexing (DWDM), the distance has been lengthened to 100 kilometers.
18. Veritas VxVM mirroring can be used, but with a distance limit of 10 kilometers.

monitor multiple clusters running in different sites over TCP/IP-based inter-site links. Each cluster within a Continental Cluster can itself be a multi-site configuration. Unlike the Campus and Metropolitan clusters, there is no automated failover. Applications are installed on the different sites, in preparation for their possible use. Since there is no quorum or other product-based interlock, administrators must take care to avoid split-brain problems.

IBM's AIX

IBM's AIX systems have a wide—and sometime confusing—range of clustering options within a single datacenter. Its Disaster Tolerant configurations, however, are consistently built around the company's High Availability Cluster Multi-Processing (HACMP) product, which provides automated application failover in a "shared-nothing" cluster style.

HACMP can be set up in a multi-site cluster to connect two sites using Peer-to-Peer Remote Copy (PPRC)—a synchronous, asymmetric disk mirroring capability of the company's Enterprise Storage Server (ESS) storage arrays.[19] PPRC also provides AIX-based tools that can control functions of the ESS array, such as its relationships with volumes in the remote ESS array, which can be up to 45 kilometers away.

IBM's preferred solution, however, is High Availability Geocluster (HAGEO), a tight integration of HACMP with a host-based disk-mirroring product named GeoRM that performs synchronous "geographic mirroring" over unlimited distances in a manner similar to the OpenVMS host-based Volume Shadowing product. Using that setup, a write operation from an application server is sent to both the local volume and a remote copy via its attached server. Failover operations for the applications and their data—as well as the rejoining of failed nodes and their respective geo-mirror volumes into the cluster—are handled together as integrated subtasks.

Linking sites requires at least two private, primary TCP/IP links, which carry status updates and mirroring I/O traffic, and a third link as a backup communications link. HAGEO uses the backup link to determine if a remote site is truly down. Since this link could also be severed at the same time as the two primary links, HAGEO requires that one site be configured as the default sole survivor.[20] When this happens, the other site automatically shuts itself down, and the survivor takes over its applications and data. This type of predetermined site "dominance" leaves HAGEO vulnerable to creeping doom failures, however.

Sun's Solaris

Sun's Solaris disaster tolerance is provided by Sun Cluster 3.0, an application-failover function that is fully integrated with the OS, a cluster file system with shared-write access to volumes from multiple nodes, and support for both Ethernet and low-latency SCI as cluster interconnects. It also uses a quorum-based algorithm and disk to avoid split-brain corruption—preferably in a third site (which Sun calls a "three-room" configuration).

At the low end, Sun Cluster 3.0 supports a campus cluster" of systems based on a stretched Fibre Channel SAN that can reach up to 10 kilometers between sites. The higher-end Sun StorEdge 9900 series array—a version of the Hitachi Lightning 9900 line—can span up to 45 kilometers using Hitachi's hardware-based mirroring product, TrueCopy.

TrueCopy provides synchronous, asymmetric data updates along with server-based utilities that can control the array's volumes and their logical relationships with volumes on the remote array. So Sun Cluster failover scripts can be crafted to automate the array takeover process as part of application failover handling.

Sun offers a host-based mirroring software product named StorEdge Availability Suite 3.1 Remote Mirror Software[21] that provides synchronous (or asynchronous) mirroring via a remote host

19. One could likely substitute other similar arrays such as Symmetrix.

20. Or "dominant site" in HAGEO-speak.

over TCP/IP inter-site links, similar to IBM's GeoRM. But the product's support for Sun Cluster 3.0 is limited to making one end of the mirror set more available—the two ends of one mirror set cannot be in the same cluster.

Critically, however, a multi-site Sun Cluster supports only two nodes, forcing the customer to choose between high availability within a site and the ability to survive a site-wide failure.[22] This is not an acceptable choice. The whole point of going to the extreme of Disaster Tolerant computing is to remove as many single points of failure as possible. And each site in a multi-site Sun Cluster would have a single server as just such a failure point. While Sun can support more nodes under a special customer agreement, this is the current limit for the generally available product.

Conclusion

With an ever-increasing reliance on IT to do business continuously, more and more companies are discovering—sometimes the hard way—how much downtime can cost. When a high-profile outage occurs, a company not only loses business, it can also lose its reputation, translating into further losses due to fewer customers and a depressed stock price.

The good news is that Disaster Tolerance has genuinely arrived in the Unix world. Though Unix DT is still in its relatively early stages, DT is no longer the exclusive purview of virtuoso proprietary platforms. HP (with Tru64 UNIX) and IBM (with AIX), in particular, have taken the imperatives of disaster tolerance to heart. Both have produced compelling multi-site cluster portfolios that can establish the near-immediate return of application service using up-to-the-second data. HP-UX and Solaris have some DT capabilities, but they trail far behind the leaders.

Deciding what needs to be protected and how, and choosing from among the many options available, involves a daunting number of trade-offs. All of the Unix server vendors—and countless services firms—are more than willing to help wade through it all, but may not always provide an easy way to compare one vendors' solution to another.

Disaster Tolerance—with its specialized software and extra equipment, buildings, and people (with specialized training, to boot!)—is expensive. It's never easy to put such a big-ticket item in the corporate budget. A company that would not lose a significant amount of money for every hour of downtime could justifiably view Disaster Tolerant computing as an overly expensive insurance policy; less aggressive and less expensive Disaster Recovery options may be more appropriate. But a company that stands to lose a really significant amount of money, time, or customers during systems outages should look at that same insurance policy as a mandatory investment. Nearly half the companies that lose their data through disaster never re-open, and 90% are out of business within two years.[23]

If your business truly depends on applications running on your servers, Disaster Tolerant Unix is certainly worth a closer look.

21. Previously known as StorEdge Network Data Replicator (SNDR).
22. Since Sun Cluster 3.0 uses a quorum method to avoid split brain problems, it also forces the use of a quorum disk—either at one of the two sites or a third one.
23. Source: University of Texas Research Center on Information Systems.

OS	OpenVMS	Tru64 UNIX	HP-UX			AIX	Solaris
Cluster Style	All	Campus	Extended	Metropolitan	Continental	Geographic	Campus
Cluster Software	OpenVMS Cluster	TruCluster	MC/ServiceGuard	MC/ServiceGuard	Manual utilities	HAGEO (HACMP)	Sun Cluster 3.0
DT Focus	Excellent	Very good	Average	Average	Average	Very good	Fair
Max nodes	96+	8	4[a]	16[b]	Not clustered	8	2
Cluster File System	Yes	Yes	No	No	No	No	Yes
Automatic continuation after site failure	Yes	Yes (but not recommended)	Yes	Yes	No	Yes	Yes
Split-brain threat	Low (quorum, optional quorum site)	Low (quorum, optional quorum site)	Medium (Lock disks or quorum site)	Medium (quorum site)	High	Low (dominant-site designation)	Low (quorum disk, optionally at third site)
Server Inter-Site Links	ATM, DS3, FDDI	Memory Channel	FDDI, Ethernet, DWDM	FDDI, Ethernet, DWDM	TCP/IP WAN	TCP/IP WAN	TCP/IP WAN
Inter-Site Distance Limit	800km	6km	100km (10km with VxVM mirroring)	100km with Symmetrix (50km with XP arrays)	None	None (trade I/O performance for distance)	45km with StorEdge 9900 (10km with stretched SAN)
Remote Serial Console Management	Yes	Yes	No	No	No	No	No
Mirroring Product	OpenVMS Host-Based Volume Shadowing (HBVS)	Data Replication Manager (DRM)	MirrorDisk/UX, VxVM mirroring	Continuous Access XP or Symmetrix Remote Data Facility (SRDF)	Continuous Access XP or Symmetrix Remote Data Facility (SRDF)	HAGEO	TrueCopy
Mirroring Model	Host-based, peer-to-peer	Hardware-based, peer-to-peer	Host-based, direct	Hardware-based, peer-to-peer	Hardware-based, peer-to-peer	Host-based, peer-to-peer	Hardware-based, peer-to-peer
Storage Inter-Site Links	Cluster interconnect	Extended Fibre Channel	Extended Fibre Channel	Extended Fibre Channel, ESCON	ESCON	Cluster interconnect	Extended Fibre Channel, ESCON
Symmetric writing	Yes	No	No	No	No	Yes	No
Updates (Sync/Async)	Yes/No	Yes/Yes	Yes/Yes	Yes/Yes	Yes/Yes	Yes/Yes	Yes/Yes

a. There must be an equal number of nodes in each site. Adding a quorum site raises the limit to 16 (1-7 nodes per site, plus two for the quorum site).

b. 1-7 nodes equally per site, plus two for the quorum site.