

Paper #4100
HP High Availability: MC/ServiceGuard, SwitchOver, and MirrorDisk/UX
Copyright 1997 Hewlett Packard Company, Inc.

Dennis McClure
Hewlett Packard Response Center
20 Perimeter Summit Boulevard
Atlanta, Georgia 30319-1417
(404) 648-2742

This paper discusses three Hewlett Packard High Availability (HA) products: MC/ServiceGuard, SwitchOver, and MirrorDisk/UX. It will describe how they work, what they can do for you, and what they cannot do.

SwitchOver was HP's first HA product that provided for redundant computer systems. In times of trouble, a standby computer reboots from a primary computer's disk drives and runs as though it were the primary. SwitchOver was sold in the days of HP-UX versions 8 and 9, and is still supported in HP-UX 10.

MC/ServiceGuard is HP's replacement product for SwitchOver. It was introduced with HP-UX version 10. Like SwitchOver, it provides for redundant computer systems, but it works very differently and has many advantages.

Whichever of these products you use for computer system redundancy, HP recommends that you also use an HA mechanism for disk subsystem redundancy. MirrorDisk/UX is an extra-cost add-on feature of Logical Volume Manager (LVM) which offers redundancy of disk drives, cables, and I/O interface cards. There are other HA disk products implemented in hardware, such as RAID drives (redundant arrays of independent disks). This paper is limited to HA software, so RAID drives are not discussed here, although they may be very valuable in your HA configuration.

As you can see, redundancy is the name of the game for High Availability. The intention is to eliminate single points of failure in order to achieve a higher degree of processing availability for the applications. Target availability is in the range of 99.9%. Although some failures are eliminated completely, others may still occur with results that are not so disastrous, and with more rapid recovery. The cost is somewhat higher than standard computer systems, but far less than the cost of Fault Tolerant systems.

In Fault Tolerance, the target availability is greater than 99.99%, and that extra fraction of a percent can be very expensive. There is usually totally redundant hardware in a single system, with multiple components functioning concurrently, duplicating computation and I/O. A Fault Tolerant system can cost 10 times more than a two-node High Availability cluster.

Whatever your investment level, any interruption of availability can be a frustration for you and the users. With that in mind, let's explore each of these three HA products.

MIRRORDISK/UX

The obvious advantage of MirrorDisk/UX is that it provides one or two additional copies of your data. We all wish it would also provide an immunity to the trouble caused by disk hardware failures. To some degree, it will.

First, even without mirroring, Logical Volume Manager has a feature called Bad Block Relocation. It can be turned on for logical volumes other than the boot lvol or dump lvols (this is because bootstrap and dump routines have to access these lvols when LVM software is not running). When a write fails cleanly, with a clear signal from the controller, LVM can relocate this block into a pool of spare blocks at the end of the lvol, and link the spare block into the desired sequence. This is done without the application's ever knowing that an I/O error took place.

Conceptually, MirrorDisk/UX is a relatively simple addition to LVM. LVM segments disk space into units called "extents". Each logical volume is segmented into "logical extents", which map to underlying "physical extents" on the disks. For each logical volume, there is an extent map kept in the LVM header area of the disk. Each logical extent of the lvol is mapped to the physical place the extent is located (the physical volume it is on, and the physical extent number). A mirror copy is implemented by having each logical extent point to an additional physical extent, presumably on a different physical volume for the best protection. Since there may be an original copy and two mirror copies of an lvol, each logical extent can point to a maximum of three physical extents.

The second (or third) copy of data provides a tremendous value for High Availability. In addition to being insurance against loss of data, it greatly speeds the process of recovery, as one copy can be updated from the other. Mirroring can also avoid some problems with I/O failures. If a read from one copy of an lvol fails cleanly, LVM will redirect the read to a mirror copy. Assuming the mirror read is successful, LVM will use the good read to repair the bad one, using bad block relocation. Likewise, if a write on one copy of an lvol fails, the others will probably succeed. The extent with the failure is marked "stale", and the application continues to run.

Some users, after investing time and money in disk mirroring, think they should never suffer another problem with disk drives. If problems happen, they want to know why mirroring did not prevent the problem.

A "clean" failure is one where the disk controller provides a prompt and intelligible reply that the I/O failed. Then it is possible for software to make smart choices about the redundant copies of data that are available. If the result is not prompt, as when there are long timeouts and repeated retries, users will notice what looks like a system hang. When results are not intelligible, as when a device is transmitting garbage or signaling badly on the SCSI bus, software may never get the chance to take alternative choices, even when mirror copies are available.

In the worst of cases, malfunctions in disk hardware may lead to UNIX panics, whether or not disk mirroring is used. Some panics are an intentional response to the detection of trouble, to protect user data from damage that might result if processing continued. Other panics are not intentional, but indicative of the kernel's inability to deal with conditions that were never expected, and that never would happen if hardware did not fail in some "noisy" or unpredictable way. For the best possible kernel resilience, particularly with newer disk hardware, it is good to stay up to date with patches.

SWITCHOVER

SwitchOver is a software product that monitors the "heartbeat" of one or more primary systems, and directs a standby system to boot and run in place of a failed primary. Most configurations have one primary and one standby, but there may be up to four primaries using SCSI disks, or up to seven primaries using FiberLink disks. The value is high availability due to redundancy of computer hardware. This provides protection against failure of a central component, such as a CPU, central bus, memory, or cache memory. Instead of crashing and being down pending a repair, processing can resume quickly on the standby while the problem is diagnosed and fixed.

The standby literally "becomes" the primary, because it forces the failed primary to halt, and the standby boots from the primary disks. Every attribute of the primary that is configured on disk will then appear on the standby. The standby also impersonates the primary by complying with SwitchOver's strict rules of hardware configuration.

The primary and standby must be in the same CPU class, which means they are similar in processing power and, more importantly, similar in their I/O card cage slots. It is necessary for all the primary's disk drives and LANs to be shared with the standby, and it is necessary for them to be attached at exactly the same hardware addresses. Each system has a file, `/etc/ioconfig`, which describes all the peripheral devices and what hardware paths they are on. When the standby boots from the primary root disk, the `ioconfig` file has to work properly. That is, all essential devices have to be found at the addresses shown in `ioconfig`.

SwitchOver uses shared disks, and this can be tricky. SCSI interfaces come with a default address of 7, the highest priority. The root disk is usually address 6. When a SCSI bus is connected on the far end to a second computer, the second SCSI interface should be set to address 6, and the disk addresses have to be lowered. If address conflicts occur, corruption of disk data is possible. Also, if a disk drive is already in use in LVM, its device file name is in the volume group configuration. Changing the address of the disk will affect its device file name, so it no longer works in LVM unless you adjust it with something like a `vgexport` and a `vgimport`. These changes should be made by experienced people, either yours or HP consultants, so that all important considerations are given.

There is one place in a standby system where the impersonation of the primary is incomplete: the LAN interfaces' hardware MAC addresses (also called link level addresses). For the outside world, a standby's interface will answer traffic for the correct IP address, which comes from the network configuration file on the primary disk. For the HP9000 internally, it works because the interface is in the same slot number as the primary's interface. But for routers or anything else that looks at MAC addresses, this LAN interface will obviously be the wrong interface. SwitchOver solves this by using relocatable MAC addresses.

As part of the product, HP provides you with two soft MAC addresses for each system in the SwitchOver configuration. These addresses are in the number range reserved for HP manufacturing, but they are designated for your SwitchOver usage, and are not imbedded on any hardware LAN interface. You configure them in SwitchOver. At boot time, the primary and the standby each download their soft MAC addresses to the LAN interfaces, and these are the MAC addresses that are learned by routers. If the standby boots up as the primary, it downloads the primary's MAC addresses, and the standby's interfaces look just like the primary's.

The details of the switch from standby to primary are fairly simple. A heartbeat process runs on the primary and uses one or two LANs configured for the purpose. (They are not dedicated to the heartbeat, so they can be used for normal network activity also.) The heartbeat process sends a special LAN packet indicating that the primary system is still running. A "readpulse" process runs on the standby with a configured timeout period, usually 15 to 30 seconds. If readpulse fails to receive a heartbeat over either LAN during the timeout period, readpulse triggers the switchover. The standby directs the primary to halt, the standby changes its own boot path to the primary system's boot disk, and the standby reboots. The bootup almost always includes running "fsck" (file system check) on every file system. Depending on the particular circumstances, it can take as little as three minutes or longer than 30. Once the boot is finished, the applications can be restarted.

In the HP support model, the standby is idle except for readpulse. Some sites try to use the standby for lower priority processing that could be postponed in an emergency. To avoid trouble, the standby processing should also be interruptable by a sudden reboot command, both for the sake of the data being processed and for the sake of terminating all processes successfully and quickly. We often hear

of the primary being used for production and the standby for test and development. When an idle standby switches over, you can be happy that HA worked for you. If the standby was being used for something, you might not be so happy.

Once a switch takes place, the possibilities for the failed primary depend on how the disks were cabled. The preferred configuration is symmetrical, where the production disks are shared with the standby computer, and the standby disks are shared with the primary computer. In that case, the failed primary can boot as the standby, and the two systems might remain switched indefinitely until the next failure causes them to switch again. Most likely, the failed primary will save its memory dump to the standby disks, and the cause of the problem will be diagnosed. It might be desirable to shut down the failed primary and do some repair.

Unfortunately, many SwitchOver sites were installed with an asymmetrical configuration where the primary disks are shared, but the standby disks are not. This saves the cost of one SCSI interface, but means the failed primary is unusable until the operator can manually stop production processing on the standby, and reboot the primary on its own disks. A site is often reluctant to do this without knowing why the primary failed, but there is no other choice unless the configuration is symmetrical.

The restoration of switched-over systems to their original posture is a manual procedure. The standby, running from the primary disks, has to be shut down, and its boot path has to be changed back to the standby boot disk. This is the point at which you will discover if you have documented the configuration correctly, because there is no place on disk where the hardware path of the standby boot disk is still recorded. It is highly recommended that you map all the disks and buses on paper, and label the hardware paths, device file names as seen by each system, and hardware model numbers. Such a map is helpful in normal operation, as well as when down and working with HP support.

There are some failures of hardware for which a memory dump is not required, and a diagnosis may be made directly from the hardware. These are high priority machine checks (HPMC's). In these cases, the repair can be made while the primary is down, but there is still the inability to test the repair without having some way to boot. Also, before powering off the failed primary, you should be certain that any shared SCSI buses will remain terminated, as when using SCSI cables with inline terminators.

Once SwitchOver is properly set up with both hardware and software configured, and it is all documented, the hardest part is finished. The remaining challenge is to run the configuration properly. It is critical that everyone involved with the operation understands how it works, so they do not make mistakes.

False switches happen if something breaks the LAN connection, or stops the primary heartbeat process, or causes a delay of primary processing that is longer than the heartbeat timeout. To avoid these problems, it is necessary to start and stop the systems in the correct order. If the primary is shut down first, without disabling SwitchOver on the standby, the standby would detect loss of heartbeat, and switch over. If the standby is booted before the primary heartbeat is running, the standby would find no heartbeat, and switch over.

There are some failure conditions in which SwitchOver is not helpful. If the failure is due to a bug in software, on the primary disk, then the standby will suffer from the same bug when running from the primary disk. Likewise if the primary had trouble related to a bad condition on the shared LANs, disks, SCSI cables, or SCSI interfaces, then the standby will suffer the same problem. In those conditions where the failure prevents normal execution, both systems will run badly or be down until the bad component is identified and fixed or removed. Finally, the only thing SwitchOver monitors is the heartbeat. SwitchOver will not detect a problem that affects only an application.

SwitchOver is no longer being sold as of HP-UX 10.20. It is not supported on HP's newer hardware models, including the K-series and D-series, and will not be supported on HP-UX 11.x. Where it is currently in use on supported hardware and software, it may continue to be used for a long time.

MC/SERVICEGUARD

Multi-Computer ServiceGuard is HP's latest High Availability product for computer redundancy. It was introduced with HP-UX version 10.00, and was updated with each release since then, so it achieved a rapid maturity. It replaces SwitchOver in the marketplace, but it bears little resemblance to the way its predecessor worked. MC/ServiceGuard seems to have improved upon SwitchOver in just about every way.

Where SwitchOver was a "crash and boot" technology, MC/ServiceGuard is based on switching the application and all its resources to a different computer. Up to eight computers, or nodes, form an MC/ServiceGuard cluster. Each of the nodes in the cluster has its own non-shared root volume group, and runs independently from the other nodes, although in close coordination with the cluster as a whole. Disk resources for the application are located in a separate volume group (usually one, but possibly multiple volume groups). In times of trouble, the application volume group can be deactivated on its usual node and activated on an adoptive node. Users connect to the application via an IP address which MC/ServiceGuard also moves to the adoptive node prior to automatically restarting the application. No reboot is done. Failover time is measured in seconds - maybe 60 seconds or less - instead of minutes like in SwitchOver.

The MC/ServiceGuard software product consists of three parts: MC/ServiceGuard itself, shared LVM, and enhanced networking software. MC/ServiceGuard components coordinate the nodes of the cluster and supervise the execution of the application. Shared LVM makes it safe to manipulate a volume group that can be activated on any of the nodes. Modified networking makes it possible to attach an IP address to an application and handle traffic for it on various LAN interfaces, depending on which node is running the application.

Compared with SwitchOver, the hardware configuration for MC/ServiceGuard is much simpler. Each node boots and runs from its own root volume group, which is never shared, so each node's I/O configuration can be unique. The CPU models of the nodes do not have to be similar. This allows more independence, flexibility, and stability.

MC/ServiceGuard depends on some shared disk drives, so the precautions listed above for SwitchOver apply here too. However, the only shared disk drives are the volume groups that are created for "guarded" applications. The shared volume groups are on SCSI or FiberLink buses that are cabled to each possible adoptive node for this application.

It is typical for each node of the cluster to run one or more of its own applications, rather than sit idly as a standby system. It is assumed that each adoptive node has enough extra processing capacity to handle the additional workload of an adopted application, or at least enough to get by for an emergency period.

Shared LVM allows for the application volume group to be flagged as a "clusterized" volume group. The volume group is removed from each node's `/etc/lvmrc` file, so it is not automatically activated at boot time. Additionally, the file systems are not listed in `/etc/fstab`. Instead, MC/ServiceGuard handles the volume group activation and file system mounts after choosing which node will run the application. Shared LVM allows the volume group to be activated for updating on only one node at a time, which is critical to prevent data corruption. (MC/LockManager, another HA product not discussed here, provides concurrent updating in an Oracle-Parallel operation).

Enhancements to networking software allow an IP address to be assigned to an application. The application is also configured for a subnet, with a LAN interface on each possible adoptive node in the cluster. When the application starts on a node, the interface on that node is assigned to handle traffic for the application IP address in addition to its own IP address. If the application switches to another node, the application IP address goes with it; the address is deleted from the original LAN interface and added to the new LAN interface. MC/ServiceGuard avoids problems with MAC addresses by sending MAC-cache flushing requests, which force new requests to use IP addresses again until the new MAC addresses are cached.

Networking modifications also allow the use of standby LAN interfaces, adding to the hardware redundancy that MC/ServiceGuard provides. The standby interface is idle until another interface fails. Then at the driver level, the entire stack and I/O queue are redirected to the standby interface, so that it goes into use without loss of service.

The applications have to be organized into services and packages. A package may consist of many services, and each service is one process. If a package has many services, they should be logically related, as they will move as a whole to an adoptive node in times of trouble.

The application volume group is first configured completely on one node. Then it is `vgexport`'ed with the `-p` and `-m` options. The `-p` option is for preview, so the volume group is not really removed from the first node. The `-m` option creates a mapfile, which can be copied to additional adoptive nodes, where it is used for `vgimport`. In that way, the volume group configuration is propagated identically to each node that can run the package. You can also use `sam(1M)` to distribute the volume group to a list of nodes in the cluster.

Each package is configured for MC/ServiceGuard, with information about the nodes it can run on, the subnet it uses and the IP address it will be identified by, and the timeout and switching options. The package has to be completely controllable by start and stop scripts, which MC/ServiceGuard will run automatically as it manages the application.

The nodes of the cluster are very closely coordinated by MC/ServiceGuard manager processes. There is a heartbeat from each node that is monitored by other nodes, similar to `SwitchOver`, but usually with a shorter timeout. In case of heartbeat timeout, the node is considered to have failed, and the cluster reforms without that node. Any packages that were running on that node are started on the next adoptive node for that package. The failed node is prevented from running the package, to avoid possible data corruption from concurrent access. If the failed node does not communicate with the rest of the cluster, MC/ServiceGuard will typically force it to halt.

In addition to the heartbeat, the health of each LAN interface is constantly monitored. In case of failure where there is a standby interface, I/O functions are moved to the standby interface. When there is no standby, MC/ServiceGuard will halt the package on this node and start it on an adoptive node that has a working interface on the designated subnet. As long as there is still network communication to the node with the failed interface, and the cluster knows that this node is no longer running the application, the node is allowed to stay up.

Another improvement over `SwitchOver` is that MC/ServiceGuard monitors the application itself to insure that its services are still executing. If a service terminates, MC/ServiceGuard will restart it on the same node as many times as are configured, and when that value is exceeded, MC/ServiceGuard will move the package to another node.

With such complexity of behavior, it is important for MC/ServiceGuard to log its activity. Two logs are used. All cluster management and package management is logged into the file `/var/adm/syslog/syslog.log`. There is also a package script log to record the processing steps and failures, if any, of the commands that are executed. If necessary, it is possible to activate more detailed diagnostic logging levels.

When a node reboots, HP-UX renames `syslog.log` to `OLDsyslog.log`, and the previous `OLDsyslog.log` is lost. Information about errors leading to a reboot is always found in `OLDsyslog.log`. If one additional reboot occurs, the pertinent log is lost. It would be prudent to implement a scheme of name changes so additional syslog files can be kept.

Because MC/ServiceGuard nodes can run independently and flexibly, it is possible to stop a package manually on a particular node and start it on a different node that can handle it for awhile, and perform maintenance on the free node. It is even possible to update HP-UX software on the free node, in a procedure called rolling updates. Nodes with new versions of software can run and rejoin the cluster, as long as no cluster reconfiguration is done. Then the packages can be moved again, including to the updated node, for the remaining nodes to be updated.

Disadvantages exist for MC/ServiceGuard too. First, as in `SwitchOver`, it has the liability that hardware failures on shared components, like LANs, disk interfaces, cables, and disk drives, may badly affect all systems that are connected. Also, although each node runs from its own root volume group, they usually all run the same versions of software, and so are subject to the same software bugs.

Second, MC/ServiceGuard uses short timeouts to minimize failover time. However, the short timeouts are sometimes a nuisance when processing anomalies cause a delay that exceeds the timeout. MC/ServiceGuard has become the inadvertent test standard of HP9000's. If anything prevents HP-UX from prompt multiprocessing, MC/ServiceGuard will likely be the product that focuses attention on the problem, due to cluster reformations. It is sometimes necessary to cure short delays caused by too much dynamic buffer cache or the size of the inode table. MC/ServiceGuard was the product that uncovered a bug in the FDDI driver, where it momentarily would answer that a healthy interface was down. This would cause a local LAN failover, if there was a standby LAN, or a package switch. The bug was fixed in an FDDI patch, of course.

Perhaps the most obvious disadvantage of MC/ServiceGuard is that it is even more complex than `SwitchOver`. It is more flexible about hardware configurations, but far more complex in its software configuration, process interrelationships, and system administration. HP recognizes how difficult it is to install and run MC/ServiceGuard. Our configuration guide says that HP consulting is required for the first installation of MC/ServiceGuard at a site. A week of consulting time is usually needed, and possibly more if the application has to be adapted to MC/ServiceGuard scripts, or if the situation is particularly complicated. There is a three-day customer class for MC/ServiceGuard that is recommended even if a consultant installs and configures MC/ServiceGuard. The class gives information and experience that is needed to run and administer MC/ServiceGuard, and to make adjustments to the configuration. It is called "High Availability: Hands on with MC/ServiceGuard", course number H6487S. In the US, call 1-800-HPCLASS. In Canada, call 1-800-563-5089.

CONCLUSION

Complexity is a common issue in High Availability products, including MC/ServiceGuard, `SwitchOver`, and `MirrorDisk/UX`. All the problems so candidly discussed in this paper relate to complexity. As you evaluate the costs of HA, you should count complexity as one of the costs. That cost is paid by applying expertise, and the value gained is increased availability. It can be your expertise or HP's or, in the best of situations, both in partnership. Expertise can make difficult projects succeed, and valuable products like High Availability worth their price.

#