Data Management Institute

BUILDING A HIGHLY AVAILABLE DATA INFRASTRUCTURE

By

Jon Toigo

Chairman, Data Management Institute

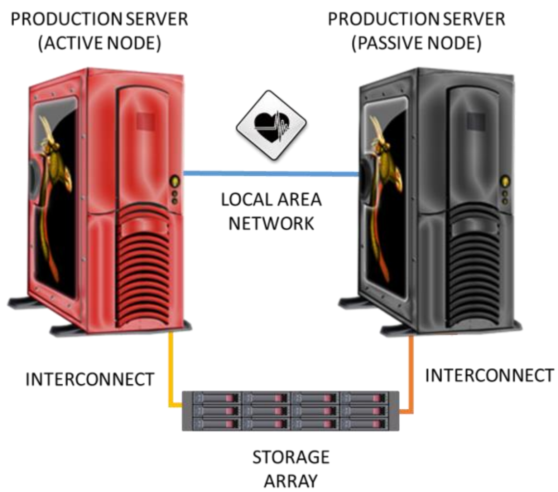
jtoigo@toigopartners.com

INTRODUCTION

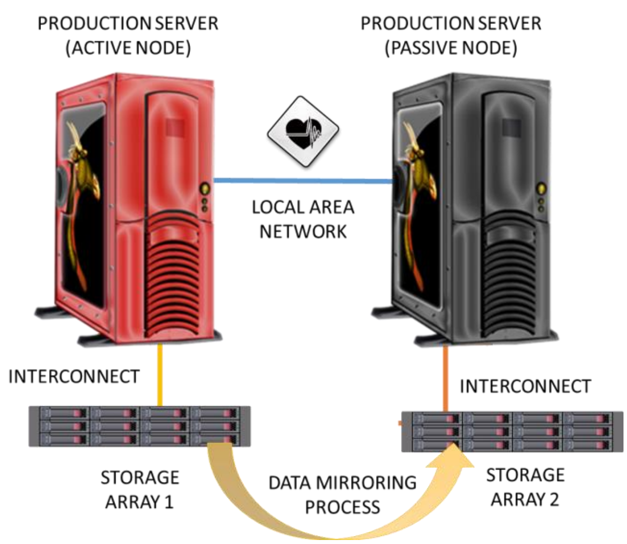
In survey after survey of business IT planners, a leading reason cited for adopting server virtualization technology is to improve application availability. Hypervisor-based computing vendors promise to improve the availability of applications by enabling failover between servers connected into clusters. For such failover clustering to work, servers typically use a "heartbeat" channel – a signal exchange – to confirm that both systems are operational. If the primary system ceases to send or respond to these signals, the issue is reported to an administrator so that a failover can be initiated and workloads shift to the alternative servers in the cluster.

In the view of many administrators, this is the meaning of high availability: active-passive server nodes with heartbeats and failover. Such technology is not only useful in providing operational continuity in the face of a user, hardware or software fault (unscheduled downtime) that impairs a production server, it also provides a fairly elegant means to move workloads from the active to the passive server when scheduled maintenance requires taking the active server offline. In that way, server HA clustering mitigates application downtime for scheduled maintenance.

That view of clustering and availability is incomplete, however. In addition to server equipment, for a system or application to be highly available, its data infrastructure must also be highly available. Without access to data, server failover alone will not ensure the continuity of business application processing.



As illustrated above, the simple clustering of two servers, facilitated by a health monitoring heartbeat process over a LAN, with both servers connected to a common storage platform, does not provide sufficient redundancy to ensure data availability. The single point of failure is the data storage infrastructure, which would make both servers and their hosted applications incapable of processing workload.



The logical alternative is to provide storage on each server independently, and to establish an on-going data mirroring process between the storage arrays. The above diagram illustrates such a configuration using a direct attached storage topology in which two identical arrays are tethered directly to each server via a host bus adapter, cabling, and ports on each array. Array to array replication can be implemented via server-based software or software-loaded on each array controller. Using on-array mirroring typically requires that identical equipment be used on both sides of the mirror operation.

Regardless of whether you use a direct attached storage array, or a network-attached storage (NAS) appliances, or a storage area network (SAN) to host your data, if this data infrastructure is not designed for high availability, then the data it stores is not highly available by extension, application availability is at risk – regardless of server clustering.

The purpose of this paper is to outline best practices for improving overall business application availability by building a highly available data infrastructure.

DATA AVAILABILITY STRATEGY

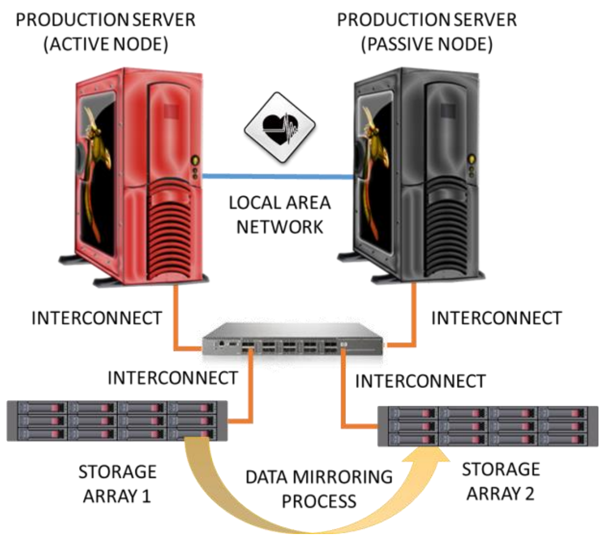
A long time dictum of continuity planning is that one of two strategies are required to ensure availability of data. One is a strategy of replacement: essentially, you do nothing prior to a data corruption or deletion event, then as soon as possible once the cause of the corruption or deletion event has been resolved, re-key all of the lost data from original source materials. That is a strategy of replacement, and if it sounds somewhat far-fetched in these days of exponential data growth rates, it is!

For data, there is only one viable strategy for ensuring availability: a strategy of redundancy. To make data highly available, you need a copy of the data.

By extension, to create a highly available data infrastructure, you need to eliminate single points of failure. You need to ensure that both the storage devices that host the data, and also the interconnect elements -- including switches, host bus adapters and/or network interface cards, and the wiring and cabling connecting servers to storage infrastructure -- are also made redundant.

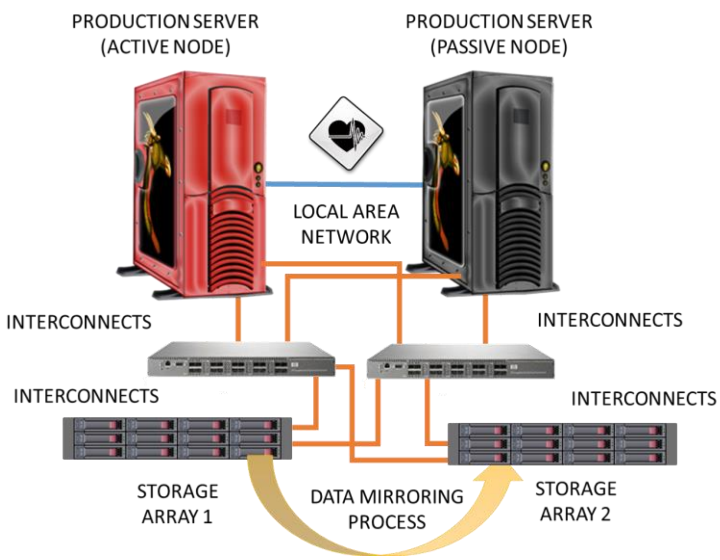
As stated in the introduction, simple clustering of two servers, with both servers connected to a common storage platform, remains vulnerable to an interruption of access to the shared array. Adding a second array and mirroring data between direct-attached storage systems compensated for the vulnerability, but in a manner that could quickly become unwieldy as infrastructure grows.

That was one reason for the introduction of storage area networks in the 1990s. SANs enabled a switched infrastructure for storage devices to be created and shared as a pooled resource to servers. A simple SAN is illustrated in the following diagram.

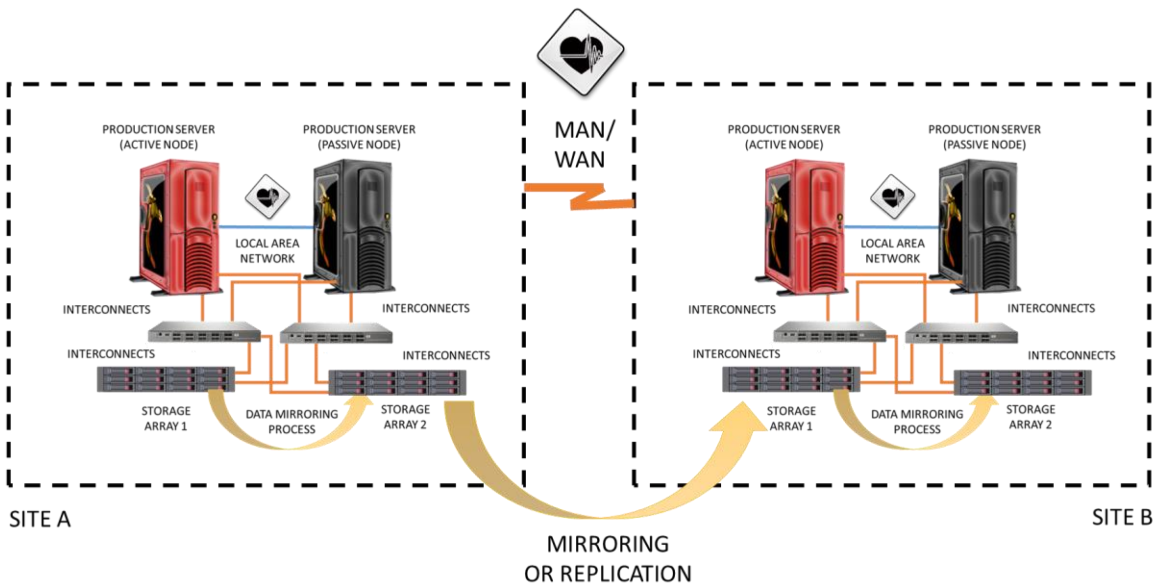


Notice that the single switch in the data path provides access to either data repository, so if storage array 1 becomes unavailable, I/O requests can be shunted to storage array 2, where a mirrored copy of data should be found. However, the switch itself is a single point of failure in terms of access to stored data. If the switch fails, neither clustered host has access to its data.

The logical resolution of this potential problem for data availability is to add a redundant storage area network switch in order to provide alternative paths to the same data storage. The benefits of the strategy are that it provides multiple paths to data from each host. The potential downside is that expenses for the configuration have multiplied. Each server must be equipped with multiple host bus adapters, additional cabling is required both from each server to both switches and from each switch to both storage arrays. Moreover, this configuration grows more complex with the addition of more servers and more storage arrays.



There is also another factor to consider: shared utility services. To ensure that this HA topology is truly effective, different switches and storage need to be on different electrical sources so that a single power outage doesn't take out the entire infrastructure. In some cases, it may be desirable to place different SANs into different locations (different buildings in a campus environment, different branch office facilities, etc.). For this to work, heartbeat functionality, plus data mirroring/replication services, must be extended across a Metropolitan Area or Wide Area Network interconnect.



From these illustrations it is easy to see the point that ensuring data availability requires more than simply clustering two servers together. From a data availability standpoint, planners need to consider

- Redundancies at the storage layer, including data replication, RAID or erasure coding on storage media itself
- Redundancies at the storage system layer, including redundant power supplies, fans, interconnecting controllers, ports, etc.
- Redundancies at the switch layer
- Redundancies in the cabling and interconnect layer
- Redundancies in host bus adapters or NIC cards in the server hosts

Those are just the hardware components. From a storage software perspective, there is a need for perfect data mirroring software -- software that works with all storage gear that is deployed and with the interconnect capabilities that are present in the topology. If storage equipment is heterogeneous (that is, is not of the same model type or from the same manufacturer), it is likely that third party mirroring and replication software will be needed to ensure that bits written to one array are faithfully copied to mirror targets.

Mirroring software provides synchronous replication of data over reasonably jitter free links of less than 70 kilometers. To mirror data beyond that distance, asynchronous replication software will likely be needed.

Additionally, the entire end-to-end storage I/O path needs to be continuously monitored, as does the integrity of mirrors themselves to ensure that data states in the source and target arrays do not move too far afield of each other. Hence, there is a "transparency" requirement that must be addressed in the data availability infrastructure that the planner designs.

NEXT GENERATION: FROM HARDWARE-DEFINED TO SOFTWARE-DEFINED

Over the past few years, substantial push-back has been coming from business IT executives regarding the cost and complexity of building resilient hardware storage topologies for data availability. Critics have argued that embedding storage services on the controllers of individual arrays has a tendency to obfuscate their ability to monitor the I/O path or to test and validate data mirror processes.

Software-defined storage (SDS) offers an optional model in which on-array storage services are abstracted away from the array controller and instead instantiated on the server as a layer of the server virtualization hypervisor software stack or as a standalone virtual machine. SDS, however, has no universal definition. Some vendors interpret it to mean moving caching functionality only off array controllers and into a server-based software layer. Other vendors include a broad range of storage services that may include thin provisioning, data mirroring, snapshots, de-duplication and other functionality. To optimize data availability, however, SDS must include capacity management as well as other services provided traditionally by array controllers.

By providing all functionality to an SDS "storage-controller-in-software" underlying physical storage infrastructure can be optimized and both storage capacity and services can be doled out to application workloads that need them in a more agile way.

Consider that, with traditional storage infrastructure, the movement of a virtualized application between servers (for example, vMotion) must be accompanied by a manual intervention within the application itself in order to provide a "new route" to the data store that is accessed and used by the application. The only alternative is to engage in expensive local mirroring of application data to storage locations on every conceivable hosting system, which can be an extremely expensive and complex proposition.

If data infrastructure itself is virtualized, if capacity is pooled into a shared resource and surfaced or exposed as virtual volumes, these virtual volumes can travel with the virtual machine as it moves from server host to server host. The result is that path re-routing between application and the physical location where its data is stored is managed transparently and automatically by the SDS controller.

Moreover, by abstracting capacity management (sometimes called virtualizing storage), the SDS layer can enable the assignment of data availability services to virtual storage volumes so that all data written to the target is immediately subjected to the right combination of protective services and replication processes that are appropriate given its importance to the business and criticality to operations.

Such a strategy insulates planners from the complexities brought about by scaling storage capacity, integrating different brands and models of hardware from different vendors, and the typical adds, moves and changes to the storage switch and cabling infrastructure. Plus, the contents of virtual volumes can be compared in real time to validate mirrors without disrupting the data copy process: a real boon.

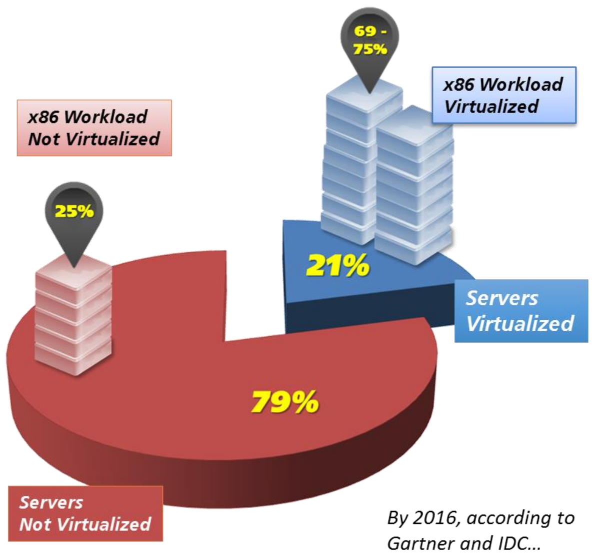
Additionally, this strategy enables a truly available data infrastructure to be created that can handle not only active-passive -- but also active-active -- clustering and failover. With active-active clustering, application workload is spread over multiple hosts which cooperate in the processing of data. In a true active-active cluster, if one server goes offline, the other active node simply takes on the entire workload burden until the failed server or a replacement can be brought online. When that happens, data is resynchronized and the active-active relationship is reestablished. Without capacity abstraction, this infrastructure is extremely difficult, if not impossible, to do build and operate – at least not with less expensive heterogeneous storage.

BUILDING IN HIGHLY AVAILABLE DATA INFRASTRUCTURE USING HYPER-CONVERGED STORAGE

Lately, software-defined storage has spawned pre-configured combinations of servers and storage devices joined together with SDS "middleware," and offered as "atomic units of computing." This model for deploying infrastructure, known as hyper-converged, is still undergoing significant development, but it is likely to become a fixture in highly virtualized server environments.

Hyper-converged infrastructure provides an opportunity to realize the goal of a high availability data infrastructure. By enabling top-to-bottom management of the infrastructure stack, and by standardizing the techniques for connecting and clustering hyper-converged appliances, one can see how horizontal and vertical scaling of capacity could be handled in an efficient way. Plus, the best products will enable clusters to stretch across metropolitan area network (MAN) connections with the same alacrity as local area networks. These stretch clusters provide another level of availability to infrastructure and data.

In some cases, hyper-converged infrastructure, server+SDS+storage "building blocks," will come to displace some "legacy" infrastructure consisting of servers connected to SANs and NAS appliances. However, the latest word from analysts suggests that the environment going forward will be a mixture of legacy and hyper-converged models.



Needed is a robust "hybrid" software-defined storage technology that can embrace both the requirements of data availability in hyper-converged infrastructure and also on legacy infrastructure like SANs. Finding such an SDS architecture will require a careful evaluation of available options. Here are some suggested questions for identifying the solution that best fits your needs.

First, planners need to ask whether the subject SDS solution works with their current infrastructure and with future storage architectures, like hyper-converged, that may be under consideration.

Second, planners need to know whether the subject SDS solution will support their current workloads – virtualized and not.

Third, planners need to discover whether the SDS solution under consideration offers capacity virtualization as well as storage services aggregation.

Fourth, planners need to assess their own on-staff skills and determine, based on this assessment, the importance of user friendliness features of the SDS solution. Does it provide simple configuration and wizard-based allocation of capacity and services?

Finally, since we are talking about data availability, planners need to look closely at what the SDS solution offers from the standpoint of data mirroring and replication services. More to the point:

1) Does the solution enable mirroring of data across unlike hardware?
2) Does the solution enable centralized management of diverse data replication processes for ease of management?
3) Does the SDS solution offer mirror transparency – visibility into mirroring processes for ease of testing and validation?
4) Does the solution automate the failover of data access between different media, arrays or SANs that are part of the SDS-administered infrastructure so that the re-routing of applications to their data requires no human intervention?
5) Does the solution provide the means to failback from a failover once the error condition that necessitated the failover is resolved?

A word of warning, most SDS solutions will not pass muster given these criteria. That is mainly because only a handful embrace the notion of a hybrid legacy/hyper-converged infrastructure or the concept of virtualizing storage capacity. Planners need to press for this functionality and to prefer SDS solutions that meet these critical requirements if they are to build an infrastructure that delivers data availability.