Realtime publishers

The Essentials Series: The Business Benefits of Rapid Application Failover

The Critical Role of Failback in Application Disaster Recovery

sponsored by

ARCserve® More than Backup

by Greg Shields

Copyright Statement

The Critical Role of Failback in Application Disaster Recovery

So the worst has happened.

Without warning, a tornado has fallen out of the sky and taken with it the building that used to house your data center. Or, its air conditioning unit failed, resulting in an overheat scenario that melted down key server hardware. Maybe the power company lost a main breaker and doesn't anticipate a fix for 2 to 3 more days. Perhaps you weren't truly protected from that Internet-based malware or zero-day attack, and every server is now full of corrupted data.

These are all bad situations. Although not all may be considered "disasters" in the classic sense, the fallout of any means a massive loss in your data center's ability to serve your business and your customers. To the unprepared business, any of these situations will require a string of long nights to fix, and could potentially impact the very health of the business itself.

But not you. *You were prepared.*

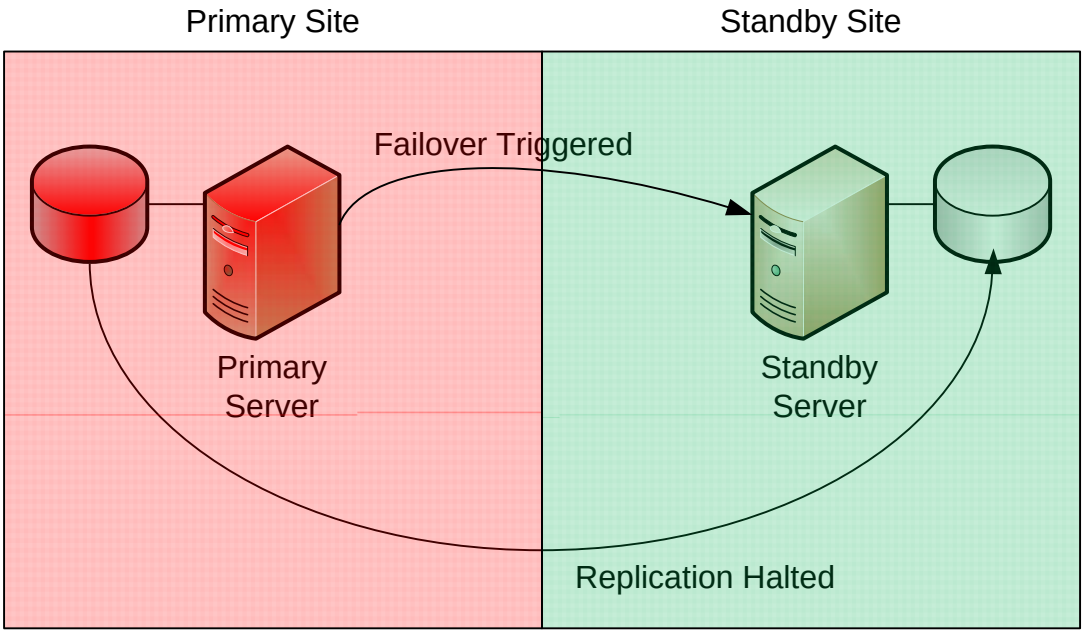

Figure 1: Once the disaster strikes, applications automatically fail over to the standby site.

You recognized before the disaster struck that an emergency plan for business services requires both data and applications to be complete. You've successfully implemented failover technologies to protect your applications coupled with a real-time replication solution for ensuring the data is also at the standby location. As a result, by the time the emergency plan was invoked, your data and applications were already failed over to standby servers in another location (see Figure 1). In the process, users saw little or even no change in their ability to work with their needed applications.

Recovering from Recovery

Now, with the disaster over, the next task is to return your operations back to normal. Most organizations with critical application availability requirements have a disaster recovery plan in place. If the loss of IT services is a large impact on your business, you've probably got the right processes and technologies in place to ensure their availability.

Yet it is the failback process that is often forgotten when disaster recovery plans are written. Making this omission even more insidious is the fact that the failback process is actually quite a bit more difficult from a technological standpoint than the failover process. Consider the steps that are required to invoke a failback of an application:

- **The primary site must be restored to operations.** Depending on the scope of the disaster, it may be possible that the entire primary site is unavailable. If a tornado removes the data center's building from its foundation, restoring servers will require a bit more effort than with a simple power outage.
- **Standby and primary servers must be reconnected.** Once the facility is ready for occupation, replacement primary servers must be networked and made available for use. This means procuring equipment that can operate your applications once again in primary operations. Once a server is built to specifications, the failover solution's software must be installed to that server, exposing the replacement primary server to your failover infrastructure.
- **Application state and user data must be reverse-replicated.** You should expect your failback process to be as non-affective to users as the failover process. This means that the failback process must first reverse replication flow to populate the replacement primary server with the data it needs (see Figure 2). Large-scale disasters can involve the loss of large amounts of equipment—and their applications—so this process can take an extended period of time. The time required here will largely depend on the size of your network bandwidth and the amount of data that must be copied over. To speed this process, some solutions enable the integration of traditional backup solutions to handle the bulk restoration, with the individual deltas being updated through reverse replication.

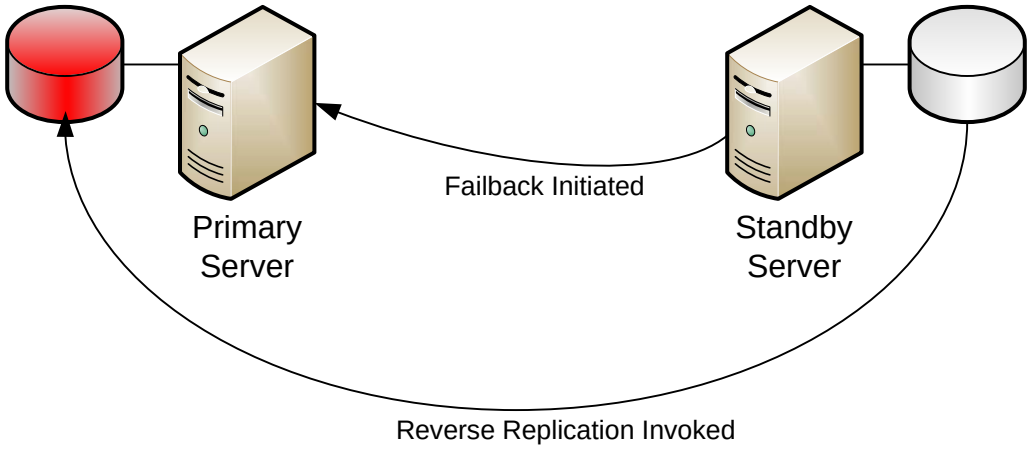

Figure 2: An application that has failed over must eventually be failed back. That process requires the reverse replication of changed data back to the primary server.

- **The failback process must be invoked without downtime.** Once the two servers are again synchronized, the final step is to actually invoke the application failback. That process, along with its network and client orchestrations, must occur with the same level of seamlessness as the failover process.

Your failover infrastructure platform must include the functionality necessary to ensure zero-impact failbacks. This includes the ability to replicate data in both directions, with the right level of transaction monitoring to identify which hosts need what data. It must also include the management flexibility to invoke a failback to different hardware in the case of a primary site failure.

Above all, ensuring these capabilities in your platform of choice provides a mechanism for the live testing of your disaster recovery planning. Although many organizations have a disaster recovery plan in place, technical limitations mean that few can actually test those plans. By implementing a solution that effortlessly enables both failover and failback, you gain the flexibility to test your planning at regular intervals without impacting users.

Incremental Rewind

A final capability that protects the IT environment against the dangers of data corruptions involves the capability to roll a server's configuration backwards to a particular point in time. This capability becomes useful in the case where a service interruption involves more than just a server outage. Consider the painful situation where an application and/or its data have become corrupted. That corruption can occur due to a misconfiguration of the application or server. It can relate to an installed patch or update. It can even be related to the introduction of malware into your production network.

In any of these situations, the introduction of corruption into an application's data cannot be directly resolved through failover alone. Such is the case because any application in a failover infrastructure will automatically replicate changes to its standby server in real time. Needed to protect against this outage situation is the ability to roll the replication of changed data backwards to a previous point in time at the backup site. Due to the snapshot-based nature of many replication solutions, this can be easily accomplished by instructing the server to roll back to a previous transaction that occurred at some point in the past.

As a result, that application enjoys the protections gained through a kind of "time travel." By creating a log of data transactions between a primary and standby server, an administrator can revert a server to a state immediately prior to the corruption, or to any point in its past. This single feature can be dramatically powerful in quickly resolving what would otherwise be a painful restoration incident involving long periods of downtime.

Failback Is as Critical as Failover

Many organizations create disaster recovery plans without recognizing the need for seamless failback. Lacking the right tools, that failback process can be exceptionally difficult or even impossible. Smart organizations recognize that failover infrastructure solutions that support the same functionality with failback as with failover enable an environment of complete flexibility. Disaster recovery plans can be tested without impact to users. Applications can be failed over with the complete assurance that they can be returned back to operations. Even incidents involving application or data corruption can be minimized through transactional-based replication. Consider each of these capabilities as important when architecting your rapid application failover infrastructure.