the Availability Digest

www.availabilitydigest.com
@availabilitydig

Swapping Replication Engines with Zero Downtime

Dr. Bruce Holenstein, President and CEO, Gravic, Inc.
Paul J. Holenstein, Executive Vice President, Gravic, Inc.
Dr. Bill Highleyman, Managing Editor, Availability Digest
December 2017

A mission-critical application often runs in redundant systems to ensure that it is always available to its users. Such a system may be configured as an active/passive pair, in which one system runs the production workload while the other system is standing by, ready to take over application processing in the event that the production system fails. Alternatively, the system can be configured as an active/active architecture, in which both systems are processing transactions. A data replication engine is used to keep the databases of the two systems synchronized.[1]

Sometimes, companies may decide to change data replication engines or to upgrade to a new version of the existing data replication engine. With mission-critical applications, it is necessary to do so without taking the applications down – a zero downtime migration (ZDM). Furthermore, it is imperative that a backup copy of the database is always available, ready to take over if the production database fails. The backup database must be kept synchronized with the production database while the data replication engine is being migrated.

In this article, we describe how a data replication engine can be changed without taking down either applications or the backup database.

Redundant Systems

As shown in Figure 1, changes to the active database in an active/passive configuration must be replicated to the database of the passive system so that the two systems are always in synchronism. These changes are replicated with a unidirectional data replication engine.

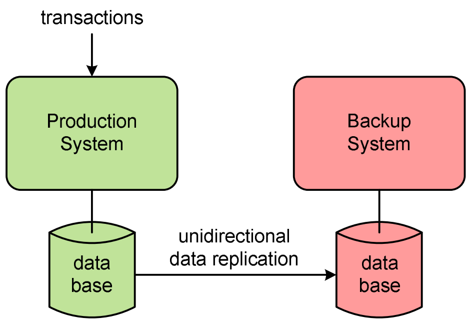


Figure 1: An Active/Passive Redundant System

[1] This article was first published in the November/December 2017 issue of The Connection. It is republished here with the kind permission of The Connection.

In an active/active configuration, changes made to either database are replicated to the other database via bidirectional replication, as shown in Figure 2.

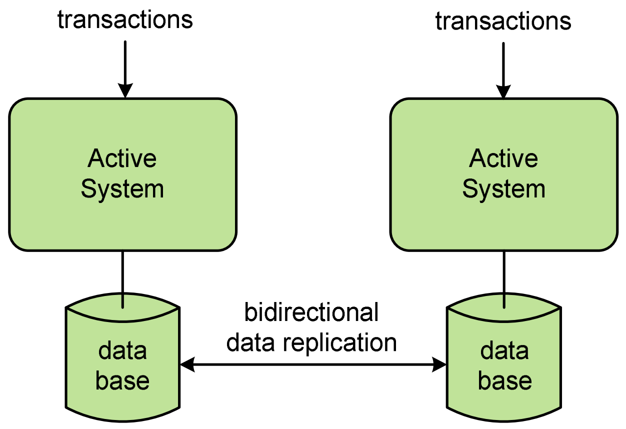


Figure 2: An Active/Active Redundant System

Changing the Replication Engine

There are cases where the replication engine may need to be changed or upgraded. For instance, the need may arise to migrate from RDF supporting an active/passive system to the HPE Shadowbase product suite to upgrade to an active/active system. Or it may be necessary to upgrade from one version of the current replication engine to a more recent version.

If the application is critical to the users of the system, it is important to migrate from one data replication engine to another with no application downtime – a zero downtime migration.

Interoperating Versions

In NonStop systems, changes to the source database are appended to an Audit Trail set of files by the NonStop Transaction Management Facility (TMF) for audited databases. Figure 3 shows version 6100 (v6100) of a data replication engine reading the changes from the source database's Audit Trail and sending them to the target system to be applied to the target database. In this way, the target database is kept synchronized with the source database.

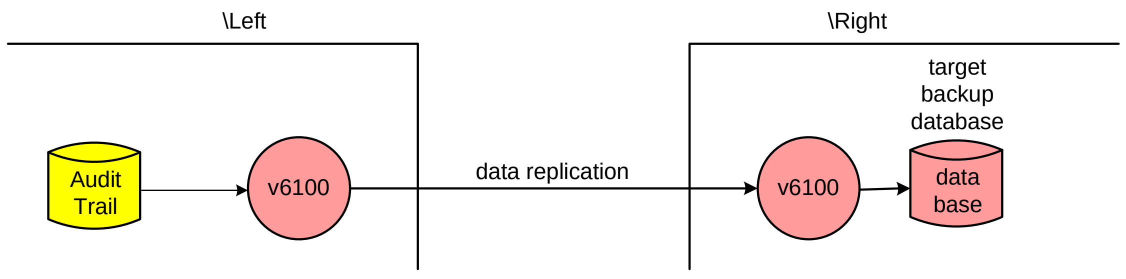


Figure 3: Replicating Data from a Source Database to a Target Database

The concept is the same for other transactional databases. For example, an Oracle database provides a REDO log that contains all changes to the source database and can be used as a source of database changes for a replication engine.

Figure 4 shows what may appear to be a straightforward solution to this migration. There are two versions of a data replication engine – version v6100 and version v6400. Version v6100 is the current version of the replication engine. The objective is to migrate to version v6400.

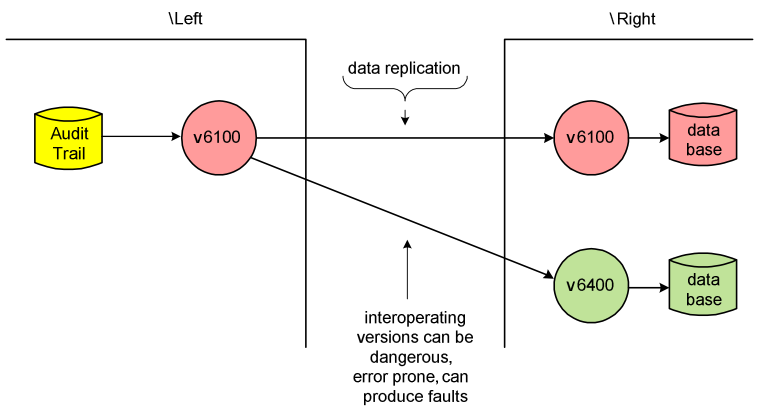

\Left
\Right
data replication
Audit Trail
v6100
v6100
data base
v6400
data base
interoperating versions can be dangerous, error prone, can produce faults

Figure 4: Data Replication Engine Migration by Intermixing Versions

The migration can be accomplished by installing version v6400 on the target system and beginning data replication from the original version v6100 on the source system (System \Left) to the new version v6400 on the target system (System \Right). As shown in Figure 5, when the database has been migrated from version v6100 to the version v6400 database on the target system, version v6400 can be installed on the source system. Version v6400 on the source system then can take over from version v6100 and continue replication. As shown in Figure 6, version v6100 of the data replication engine now can be retired. Replication has been migrated to version v6400 without having to take down the application.

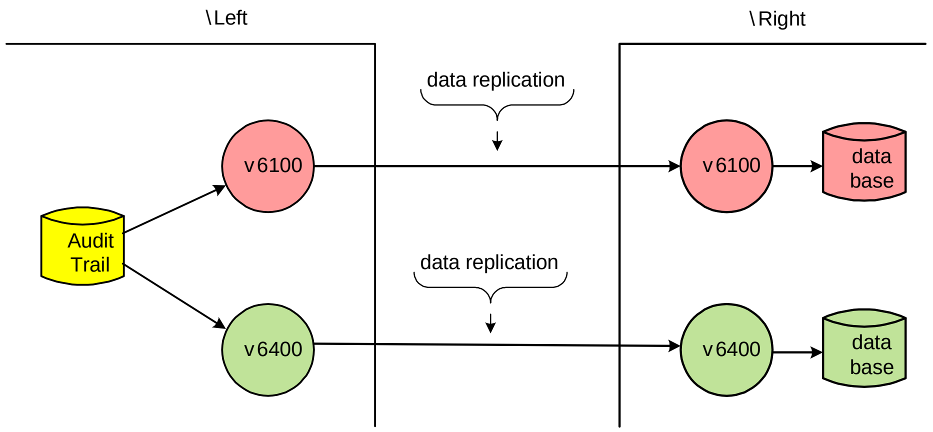


Figure 5: v6400 Takes Over Replication

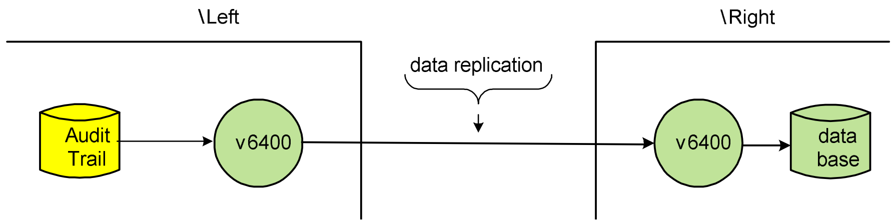


Figure 6: v6100 Is Retired

However, this type of migration requires either different data replication engines or different versions of the same replication engine to interoperate with each other, as shown in Figure 4. Note, as shown in Figure 4, this sequence is fraught with peril; as the various versions have to be installed, tested, activated, and then interoperated for various periods of time during the process.

For instance, with HPE Shadowbase software, there are dozens of older versions. It is impractical to certify each older version of HPE Shadowbase software interoperating with each newer version and vice versa. The coding changes required can be quite extensive to revise the message structures upward and downward as well as the effort to fully exercise and test each such combination. A similar situation exists if the data replication products are from different vendors. It is a risky business having them interoperate at all. Additionally, the relative usage of each combination would be for a very small population of the installed base.

Therefore, the interoperation of different data replication engines or their versions is a dangerous procedure. It is not known how well they will work together or if they even will. As such, it is likely that this configuration will be error-prone and will produce replication faults.

Rather, a method is needed to install and configure a new parallel replication environment alongside the previous/existing replication environment so that the new environment can be tested and certified while the existing environment continues to run. Once trusted, the new environment can slowly "take over" the replication load from the previous environment until all such traffic has been moved to the new environment. At that time, the previous environment can be shut down and decommissioned. This is the essence of the zero downtime migration process.

Version Independence

An alternative migration that does not require version interoperation is reflected in the following figures. This example assumes an application with 1,000 tables is initially being replicated by version v6100 of the data replication engine.

As shown in Figure 7, a test environment for version v6400 is established before migrating to version v6400. Of course, the initial checkout of the new replication engine (version v6400) should be undertaken ahead of time in a nonproduction test environment. The following sequence assumes that the initial checkout has already been completed and describes how to replace the previous version v6100 in the production environment.

A tentative target database (PROOF) is created, and a representative sample of 20 tables is loaded from the production database into PROOF. The tables can be loaded offline via a SQL tool or online via the HPE Shadowbase Online Loader (SOLV), which is an online loading utility for loading source files/table into target files/tables while the application continues to run. SOLV thereby allows both the source and the target databases to be online and active while the source data is copied into the target environment.

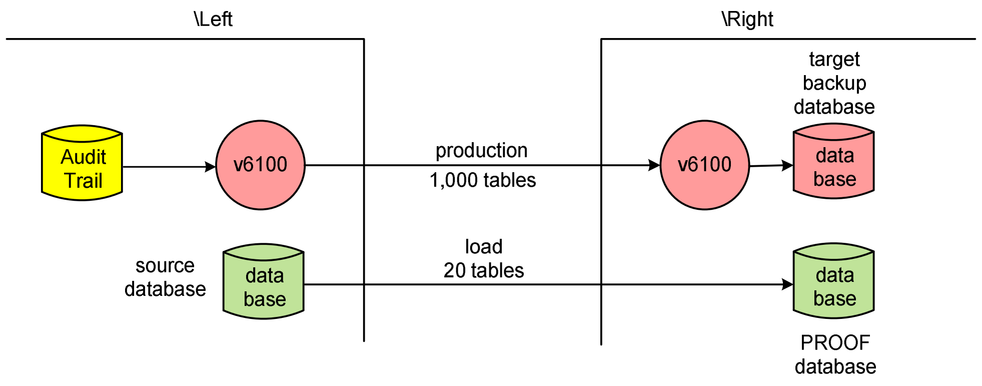


Figure 7: A PROOF Temporary Target Database with 20 Tables

As shown in Figure 8, data is replicated by the new data replication engine (v6400) to the 20 tables in PROOF; and the updated data is compared to the target database. A database comparison tool such as the HPE Shadowbase Compare product can be used to verify the databases; it examines and compares the databases online while they are being updated.

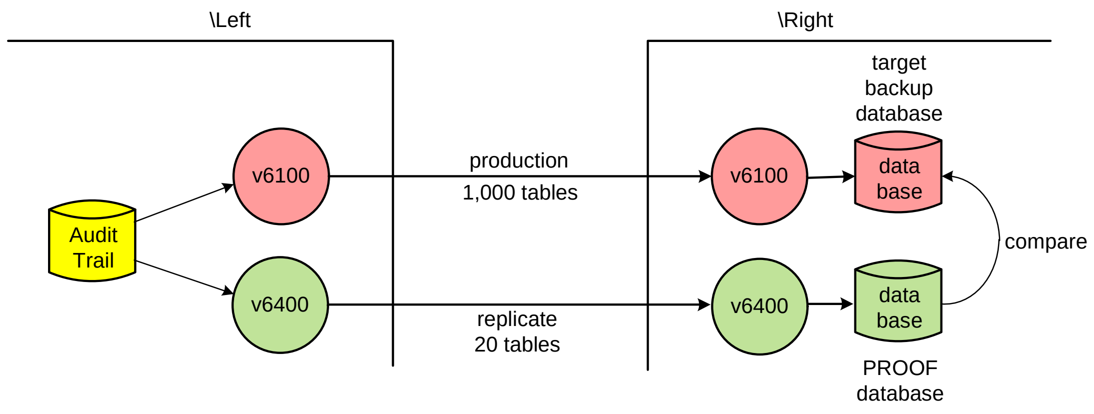
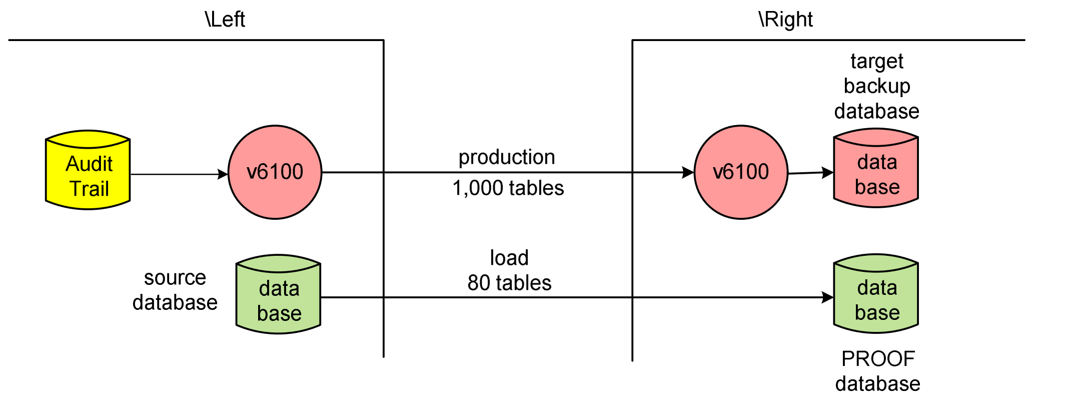


Figure 8: Test the Replicated Temporary Target Database

The PROOF replication is run for a period of time to ensure that the new version v6400 of the replication engine is operating properly in this quasi-production environment. If the results do not match, the discrepancies must be resolved and the test rerun. If the replication test is successful, it can be expanded if desired to more tables, for example 80, as shown in Figure 9.



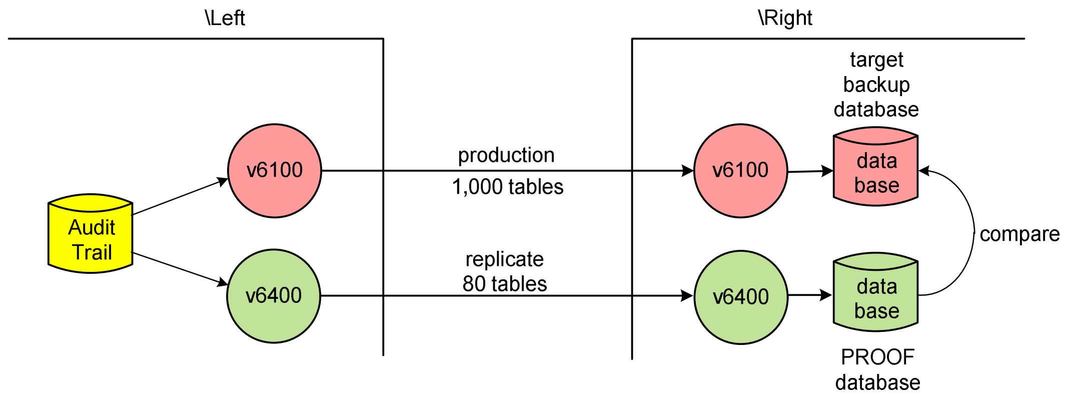


Figure 9: Replicate 80 Tables to PROOF and Compare

If the 80-table replication test works properly for a period of time, then the new version v6400 of the data replication engine is now validated to be working properly – it is a trusted and *known-working* environment. If disk space is at a premium, the successfully compared tables in the PROOF database can replace the original target database tables; and they can be removed from the version v6100 configuration. This step allows the data to be moved over to the new environment a few tables at a time, which can reduce the risk of a *big-bang* migration. The HPE Shadowbase product allows such a migration of the database objects from one replication configuration to another over time.

As shown in Figure 10, the test can then be expanded to replicate all 1,000 tables. If it is successful, the PROOF database can replace the production database, the data replication engine version v6400 can replace version v6100, and version v6100 can be retired, as shown in Figure 11. The 1,000 tables now are being replicated to the target system with the new data replication engine version v6400.

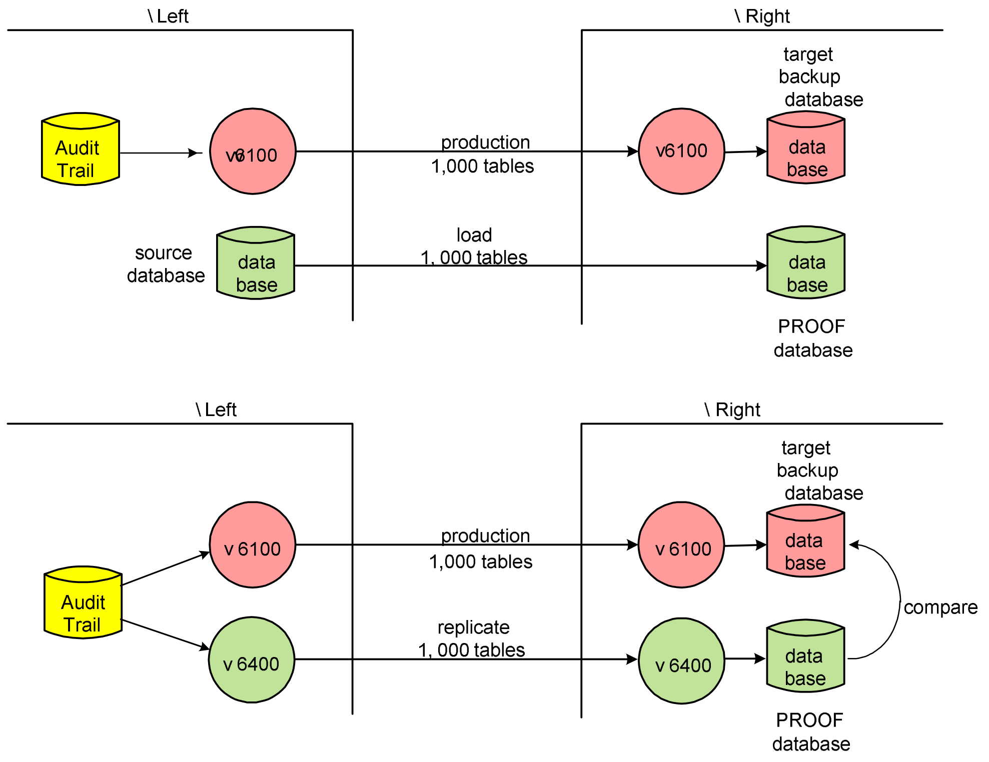


Figure 10: Replicate 1,000 Tables to PROOF and Compare

During the entire migration, the target database was available as a backup to the production database. At any point, if an error with the new environment was identified, the PROOF testing could be stopped without affecting the production replication environment. The replication engine was migrated from version v6100 to version v6400 with no application downtime and with continuous availability of the target database – a zero downtime migration.

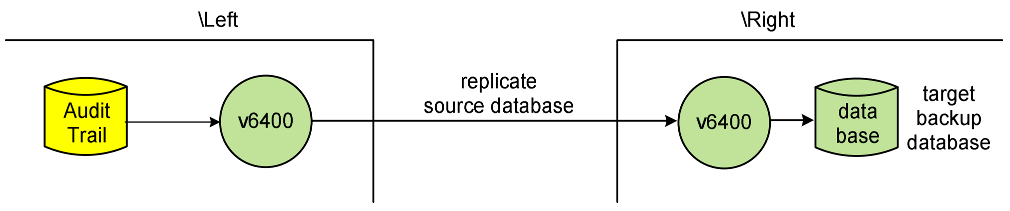


Figure 11: Configure New Version of Replication Engine

If disk space is limited, or if the environment is running as an active/active architecture, then as each PROOF test completes, the tables that were just validated should be removed from the old (v6100) replication environment and added to the new (v6400) replication environment using the production version of the target database. This step will allow for a phased cutover from the old environment to the new one over a period of time, dictated by the speed in which the system operations team can complete the PROOF tests rather than via a classic all-at-once big-bang scenario.

Summary

Sometimes it is necessary to change or update a data replication engine. Properly undertaken, such a migration will impose no downtime on either applications or users. We call this a zero downtime migration.

The results are greatly reduced risks for error as well as staff stress levels during the migration process. There is no big-bang cutover. The migration can take place at normal working times rather than late at night or on weekends, and it can even occur over an extended period of time. Furthermore, the backup database is always available and is fully synchronized with the production database the entire time. Thus, application availability is ensured during the migration process.

This migration technique is similar to the Shadowbase Zero Downtime Migration (ZDM) technique that customers have been using for decades to upgrade their applications, database schema formats, file and table locations (or indices), operating systems, or perform a hardware refresh. For additional information, please see the white paper: http://shadowbasesoftware.com/white-papers/2015/07/using-shadowbase-to-eliminate-planned-downtime-via-zero-downtime-migrations-white-paper/.