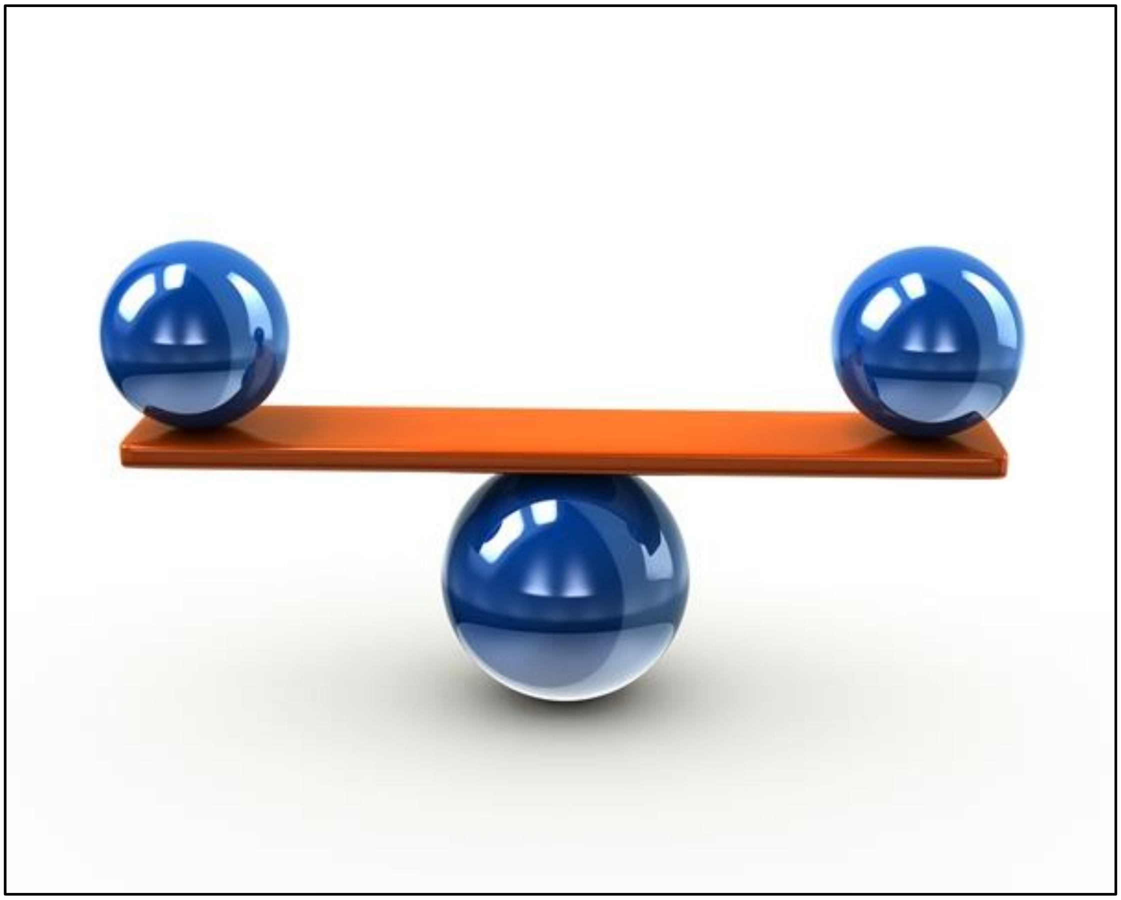
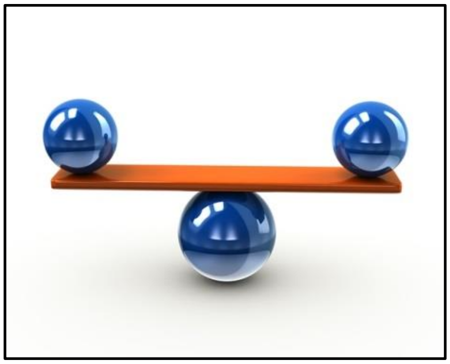
HPE Shadowbase Solutions and HPE Pathway Domains – Perfect Together!

A Gravic, Inc. White Paper



Executive Summary



In today's business world, access to real-time online transactional data is a competitive advantage. To realize this advantage, data must be available at any time, from anywhere, and it must be current. If this advantage is not realized, then the inability to access or update this current data carries a significant business cost, possibly measured in many thousands of dollars per second or even failure of the business itself.

These requirements necessitate an IT infrastructure that is continuously available, and an adequate business continuity plan in place to assure business service availability under any circumstances. These circumstances comprise both **unplanned outage events** (software or hardware errors, power outages, etc.), and **planned outage events** (software or hardware upgrades, datacenter relocations, platform refreshes, etc.).

The HPE Shadowbase suite of data replication software products provide solutions for the continuity of *data availability* in the event of planned or unplanned outages. These solutions enable:

- Continuous Availability – Scale applications and databases across multiple physical nodes to create one logical system that can scale linearly and survive the failure of a node or even an entire datacenter. Deploy highly available active/passive disaster recovery architectures, active/"almost active" higher availability architectures, also called *Sizzling-Hot-Takeover* (SZT), or continuously available active/active architectures, to eliminate unplanned downtime.
- Data Streaming – Perform homogeneous or heterogeneous data streaming using change data capture (CDC) with sub-second latency across two or more homogeneous or heterogeneous platforms. Use uni-directional or bi-directional replication depending on the needs of your applications and projects.
- Disaster Recovery – Load and maintain hot-site backups of production databases for seamless recovery in the event of a primary system failure (eliminate unplanned downtime). Gravic's patented bi-directional replication technology provides continuous availability, enabling two or more simultaneously active systems within the architecture.
- Application Upgrade, or Database, System, or Site Migrations – Create and maintain new or modified databases from legacy databases. Seamlessly upgrade the operating system or migrate to new systems or sites with minimal or zero application downtime (eliminate planned application downtime). Keep the new databases synchronized with the old until all applications have been successfully migrated to the new systems. Remove the risk of *big bang* all-at-once cutovers from your system or application upgrades by using Shadowbase Zero Downtime Migration (ZDM) solutions.

Shadowbase software can provide all the capabilities necessary to ensure ***data*** remains available in the event of any outage circumstance, planned or unplanned. But what about the *business* ***applications*** themselves? Clearly, as part of the business continuity implementation, mechanisms must be in place to ensure continued availability of both the data and the applications which operate on that data if business services are to be maintained – this is the realm of HPE Pathway Domains (PD) for HPE NonStop server systems.

Just as Shadowbase software creates copies of your production data across HPE NonStop server nodes, the PD feature provides a means to allow applications to be distributed across HPE NonStop server nodes, with user requests being load-balanced between them. If one of the nodes in the domain undergoes a planned or even unplanned outage, the user requests are automatically routed to the remaining active nodes in the domain, without need for operator intervention or special applications programming. Once the downed node is recovered, it is reintegrated into the domain and made available for user request processing. This feature therefore provides significant scalability and availability benefits for Pathway applications.

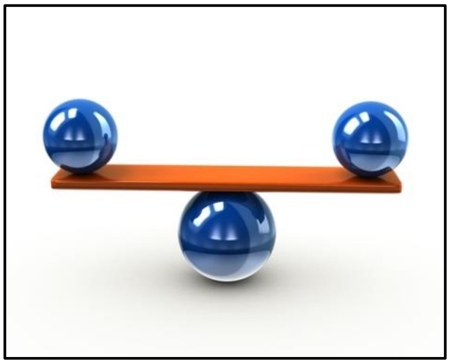
HPE Shadowbase software keeps the data available, and a PD keeps the applications available. This white paper describes how Shadowbase solutions working together with Pathway can provide seamless continuous availability of business services in the event of either a planned or an unplanned outage. In summary, PD and data replication are perfect for each other – together they provide scalability and protection for your applications ***and*** data from any single failure.

Table of Contents

Table of Figures

HPE Shadowbase Solutions and HPE Pathway Domains – Perfect Together!

HPE Pathway Domains



HPE Pathway with NonStop TS/MP software product:[1]

- is an application process container with a requester-server paradigm.
- provides high levels of application scalability by the distribution of workload (user requests) across dynamic pools of application server processes (serverclasses), spread across multiple HPE NonStop server CPUs.
- provides high levels of application availability by monitoring and automatically restarting server processes in the event of failure.
- is widely used as a programming and execution infrastructure for the implementation of OLTP applications on the HPE NonStop server platform.
- supports condition-based logic to dynamically route workloads across and scale serverclasses.

The Pathway model provides excellent application availability for a single HPE NonStop server, providing local fault tolerance. But this model still necessitates application outages for certain planned system maintenance activities, and offers no protection against unplanned system outages. Application scalability is also limited to the capacity of a single HPE NonStop server.[2]

To address some of these issues, in the HPE NonStop TS/MP version 2.3 product, the notion of a *Pathway domain* was introduced. The Pathway Domains (PD) feature enables the configuration of multiple Pathway environments (PATHMONs)[3] that can behave as a single, large, integrated Pathway application environment. By replicating an application environment across the members of a PD, any of the environments within a domain may be taken out of service (for an application software upgrade, for example), while the remaining environments within the domain continue processing work with no service interruption.

- Pathway serverclasses (application processes) are replicated across the environments in the domain and share access to a common database.
- Serverclass requests are transparently load-balanced across the domain by Application Cluster Services (ACS), an internal component of the HPE NonStop TS/MP product (client requestors do not know or care which environment in the domain is servicing their request).
- If a Pathway environment becomes unavailable, ACS automatically directs client requests to the remaining environments within the domain.
- None of these steps require any special application programming or operator intervention.

Once the application upgrade is complete, that Pathway environment can be restarted and reintegrated into the domain, and one-at-a-time the upgrade can be performed across each of the other environments (this is called a *rolling upgrade*), all with no application outage visible to the users.

[1]For more information about Pathway, please visit the HPE website.
[2]Pathway has certain configuration attributes which are constrained by the limits of a single HPE NonStop server system (for example, the maximum number of server processes).
[3]A PATHMON is an individual Pathway environment, comprising multiple serverclasses distributed across NonStop CPUs. A PATHMON is the unit of configuration and management of a Pathway environment. Multiple PATHMON environments can share processing in a Pathway domain.

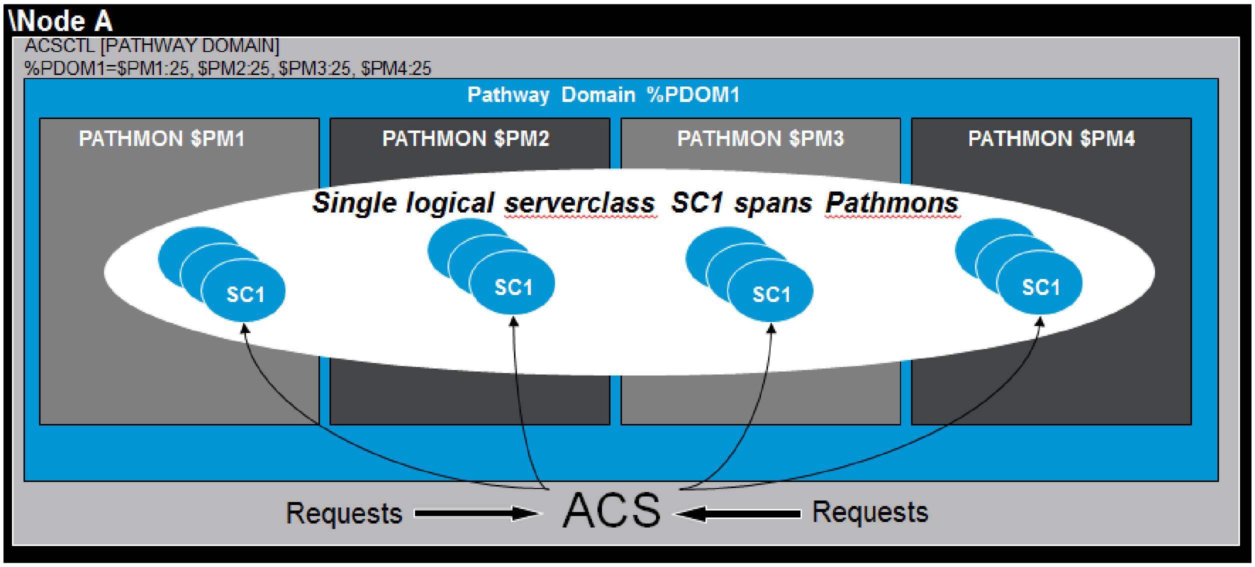


Figure 1 – Example Pathway Domain Comprising Four PATHMON Environments on a Single HPE NonStop Server Node

In the example shown in Figure 1, four PATHMONs are configured in a domain on a single HPE NonStop server node. The same serverclass (SC1) is running under each PATHMON, so any PATHMON in the domain can service any request for that serverclass. Under normal conditions, requests are routed by ACS to any PATHMON in the domain, based on load and weighting factors (more details on these weighting factors are provided in the "drill-down" section below).

To perform maintenance, PATHMON $PM1 is shut down. ACS then automatically routes requests only to the three remaining PATHMONs in the domain. Once the updates to $PM1 are complete, it is restarted and ACS automatically reintegrates it into the domain and begins sending requests to it (as demanded by load). PATHMON $PM2 may then be shut down to perform maintenance, and so on. Thus, the maintenance is repeated across each of the Pathway environments with no loss of service during the process.

PD also have another significant benefit. By enabling a single logical Pathway application to span multiple Pathway environments, the limits associated with a single Pathway environment are effectively removed. A domain can comprise up to four Pathway environments, hence the limits of a single Pathway environment are increased by up to four times.

Initially, PD were limited to a single HPE NonStop server node (all the member PATHMONs in the domain had to be running on the same system). This limitation meant that in the event of a node failure, or the need to take a whole system down for hardware or software maintenance, the Pathway application became unavailable and would have to be manually failed over to another Pathway instance running on another system. Such a failover takes time (during which the application is unavailable), and in some cases may not succeed at all (for example due to configuration or network routing errors), resulting in extended application service unavailability. Also, the remaining Pathway limits associated with a single HPE NonStop server were not resolved.

To alleviate these issues, in HPE NonStop TS/MP version 2.5, HPE enhanced the PD feature to enable a remote Pathway environment to be configured as part of a domain. This feature is known as "PD across nodes," meaning that a logical Pathway application can now span HPE NonStop server nodes. To an application, it neither knows nor cares where a particular server is located; ACS automatically routes the request to the "best" server instance that is available, whether that server be on the same or another system.[4] With this capability, an entire HPE NonStop node within the domain can be taken down for planned maintenance (for example for an O/S upgrade), or suffer an unplanned outage, and ACS will continue to automatically route requests to Pathway environments running on the other still-available nodes in the domain, resulting in end users avoiding an application outage.

[4]There are Pathway configuration options to enable control over the workload distribution, for example, to favor the local node. In general, ACS will route requests to the server process with the shortest response times.

In the example shown in Figure 2, four PATHMONs are configured in a domain spanning two HPE NonStop server nodes (\NodeA and \NodeB). The same serverclass (SC1) is running under each PATHMON, so any PATHMON in the domain can service any request for that serverclass. Under normal conditions, requests destined for this serverclass are routed by ACS to any available PATHMON in the domain where SC1 is defined, on either node.

If PATHMON $PM1 is taken down for an upgrade, then ACS will route further requests only to PATHMONs $PM2 through $PM4 until $PM1 is returned to service. Likewise, if \NODEA fails or is taken down to perform system maintenance, ACS then automatically routes requests only to the two remaining PATHMONs in the domain on \NODEB. Once \NODEA is updated/recovered, the system is restarted and ACS automatically reintegrates $PM1 and $PM2 into the domain and begins sending requests to them (as demanded by load). \NODEB may then be shut down to perform maintenance, etc.

In addition to increasing application availability, PD across nodes also increases application scalability since Pathway is no longer constrained by the limits of a single HPE NonStop server node (a logical serverclass can scale across multiple HPE NonStop server nodes).

HPE Pathway Domains Across Nodes – Drill Down

To use PD, the Pathway system administrator creates a PD configuration in the ACS control (ACSCTL) file for each node, which defines the environments in the domain, specifically, the PATHMON names qualified by node name if the PATHMON is remote.

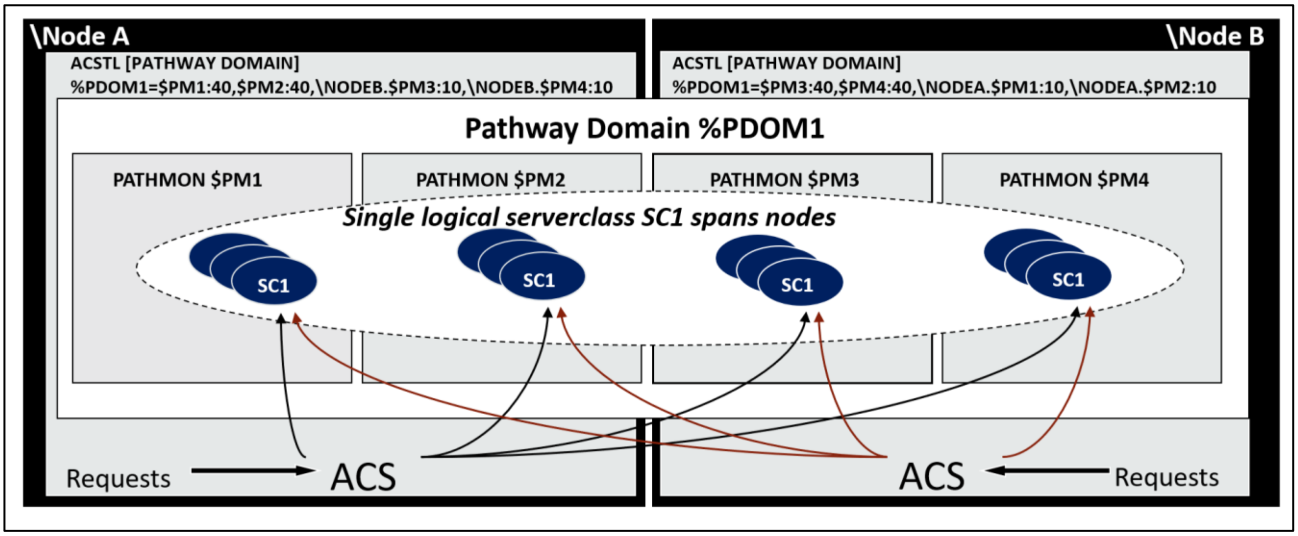


Figure 2 – Example Pathway Domain Comprising Four PATHMON Environments Spanning Two HPE NonStop Server Nodes

To facilitate application scalability and availability, it makes sense for each node in the domain to have the same PD configuration defined locally (each PATHMON comprises the same set of serverclasses, such that requests for these serverclasses can be satisfied by any PATHMON in the domain).[5] There can be up to four PATHMONs defined in a domain. Multiple domains can be defined with different members if required (each with a unique local name). In addition to the domain name and list of PATHMONs in the domain, the domain entry in the ACSCTL file specifies a weighting factor for each PATHMON in the domain, which ACS uses to distribute serverclass links between the PATHMONs in the domain. For example, %PDOM1 = $PM1:40, $PM2:40, \NodeB.$PM3:10, \NodeB,$PM4:10

In this example, on \NODEA, %PDOM1 is the name of the PD, $PM1 and $PM2 are PATHMONs executing on the local node (the one on which this PD (%PDOM1) is defined), and each are weighted at 40%, so each gets approximately 40% of the serverclass links created for server requests for this domain. $PM3 and $PM4 are remote PATHMONs (on \NODEB), and each gets approximately 10% of the links created. Note the total weighting factor must equal 100%. The same domain, %PDOM1, is also defined on \NODEB, but with the local and remote PATHMON weightings reversed. In this way, any PATHSEND request for serverclass SC1

[5]Each PATHMON in the domain may also be configured with a subset or superset of the serverclasses.

specifying the domain name %PDOM1, or a PATHMON configured in that domain, can be satisfied by any PATHMON configured in the domain (not necessarily the one to which it was addressed). Also, by specifying a higher weighting factor in the %PDOM1 ACSCTL configuration for local PATHMONs on each node, this helps ensure that requests will generally tend to be satisfied by locally resident servers, rather than incur cross-network overhead.[6]

It is also important to configure the number of servers for each PATHMON in a domain (the `MAXSERVERS` parameter) commensurate with the weighting factors; otherwise ACS will not be able to achieve the desired link distribution and a load imbalance and compromised response times will likely result. For example, a PATHMON with a high weighting factor (when viewed across all domain configurations which include that PATHMON), will not be able to satisfy demand if it is also configured with `MAXSERVERS` set to a low value.

Note that these domain weighting factors are not absolute, ACS observes them, but in some cases (especially in low-load situations), the links may not be distributed as defined. The reason is because the ACS load-balancing algorithm uses many factors to determine the "best" link to use (the one which is most likely to give the shortest response time). Other settings, such as `MAXSERVERS` as discussed above, can also impact the link distribution. Hence, while the weighting factors in these examples means it is likely the request will be sent to a local PATHMON about 80% of the time, it does not guarantee it. (If for some reason a link to a PATHMON on the remote node is getting faster response times, and the link is currently free, ACS will use it rather than create a new link just to satisfy the weighting parameter.)

From an application programming point of view, a PD appears as a single logical Pathway environment. No programming changes are required to use PDs. If the requester's PATHSEND request specifies a PATHMON name that is defined as a member of a domain, ACS routes that request to any PATHMON in the domain, according to the load-balancing algorithm.[7] If multiple domains are configured on that node, ACS scans the domain list for the specified PATHMON, and it routes the request to the last domain defined in the ACSCTL file containing the requested PATHMON name. A PATHSEND can also specify a domain name rather than a PATHMON name, in which case the request is routed to any PATHMON in the specified domain. If a PATHMON in a domain becomes unavailable, ACS automatically detects this unavailability and will stop routing requests to that PATHMON.

A new command-line interface was introduced to enable run-time management of PDs, known as PDMCOM, which is a superset of the familiar PATHCOM operator interface. PDMCOM enables status information to be gathered and operations to be performed on the members of a domain, or on the entire domain.[8]

But What About the Data?

All mission-critical business continuity architectures involve multiple (at least two) geographically dispersed processing nodes, each being capable of hosting the application database so that one node can continue to provide business services in the event of an outage of the other(s). This redundancy provides the basis of continuous availability. There are several requirements which must be met in order to accomplish this goal, but two of the most fundamental are:

1. *The data which the business services require must be available and be consistent/correct,* and
2. *The business services themselves (applications)*, *must remain available in the event of a failure*.

This goal is typically achieved via some form of application duplication across the systems, and data replication between them, so that if one node fails the applications and data are readily available on the other system and users can be quickly switched (via re-routing) to it. We have discussed how PDs maintain *application availability*; next we discuss how Shadowbase software maintains *data availability*.

Basic Active/Passive Uni-directional Replication

As an application makes changes (inserts, updates, and deletes) to its local database (the source database), those changes are immediately sent over a communication channel to the target system, where they are applied to the target database (Figure 3). The target database typically resides on another independent node

[6]Note that these configurations are included for example purposes only, to help explain the PD feature. They do not prescribe how any particular customer application should be configured, which will depend upon the specific requirements for that application.
[7]If the request must be sent only to the specified PATHMON, the application must set a flag on the PATHSEND call.
[8]For more information about PD configuration and management, see the current HPE NonStop TS/MP manuals.

that may be hundreds or thousands of miles away. The facility that gathers database changes made to the source database and applies them to the remote target database is known as a (Shadowbase) *data replication engine*.

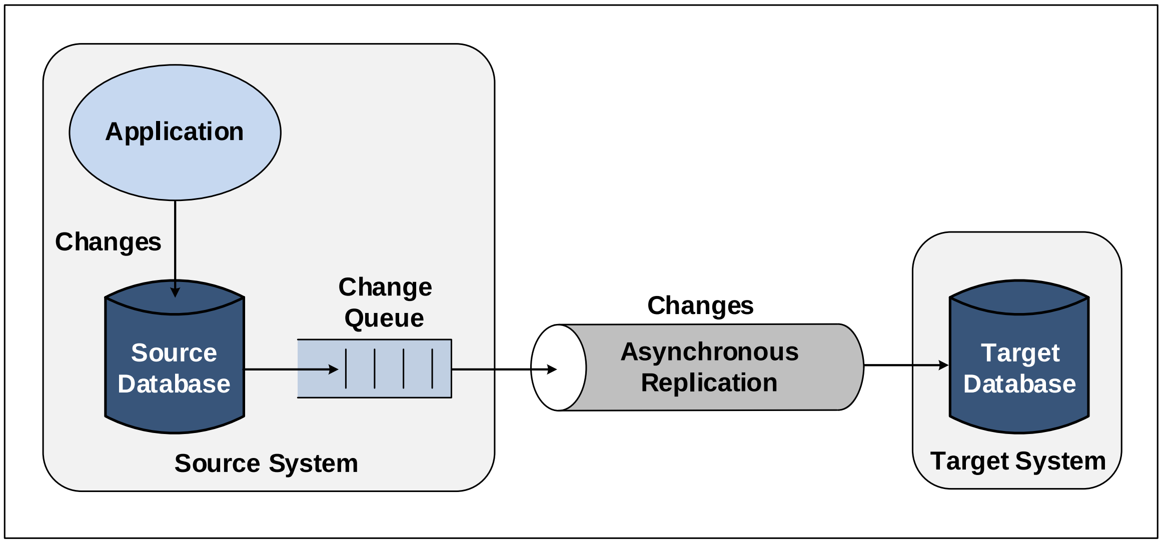


Figure 3 – Basic Active/Passive Uni-directional Replication

HPE Shadowbase Data Replication and the Business Continuity Continuum

There is a continuum of system architectures for business continuity which provides different levels of availability, as measured by the amount of data loss resulting from a failure, known as the Recovery Point Objective (RPO), and the amount of time taken to restore service after the failure, known as the Recovery Time Objective (RTO). These architectures can implement *high availability* (also referred to as "four nines of availability"), which means application downtime of about an hour or two per year, to continuous availability, which means application downtime of a few seconds per year (also referred to as "five to six nines") with rapid, automated recovery. Each of these modes is supported by Shadowbase replication and is discussed briefly below.

Active/Passive Systems

Shadowbase basic uni-directional (one-way) replication is often used to keep a passive backup (target) system synchronized with an active (source) production system. These systems are known as *active/passive systems*. All online users are connected to an active node which processes all transactions and replicates the database changes made to a geographically remote standby database, keeping the two databases in synchronized[9].

Applications may be up and running in read-only mode on the passive (backup) node so that the backup database may be actively used for query and reporting purposes (known as a *hot-standby*). If the active node fails, the applications at the backup node can remount the database for read/write access and take over the role of the original active node. Active/passive systems always have much longer recovery times than active/active systems because of the time needed to bring the passive system into a fully operational state and switch over all users. There could be other considerations, such as the time needed for the management decision and approval to perform the takeover in the first place, and also failover faults. Recovery times for active/passive systems can last hours, or even longer.[10]

With Shadowbase asynchronous replication, only the data in the replication pipeline (due to replication latency) at the time of the source node failure is lost, thus supporting minimal (but not zero) data loss. A new Shadowbase release, Shadowbase ZDL, is able to support synchronous replication, where no data is lost following an active node failure. Figure 3 is an example of a uni-directional active/passive system.

---

[9]This scenario assumes a "synchronous replication" architecture, otherwise, the databases are synchronized in "near real-time," meaning the potential for data loss is constrained to the "replication latency" of time taken to replicate the data to the standby database.
[10]For more information on business continuity architectures and their corresponding recovery times, please see *Choosing a Business Continuity Solution to Match Your Business Availability Requirements*.

Sizzling-Hot-Takeover Systems (SZT)

A sizzling-hot-takeover (SZT) system is similar to an active/passive node architecture using uni-directional replication, as described above, except that it is immediately ready to start processing transactions if a failover occurs (Figure 4). This configuration has its applications running and the local copy of the application database already open for read/write access. It is in all respects an active/active system, with the exception that all user transactions are directed to one node (thereby avoiding data collision issues).

While using Shadowbase data replication, the SZT system can take over processing very quickly because its local database is synchronized with the active database and is completely consistent and accurate; the applications are already up and running with the database open for read/write access.

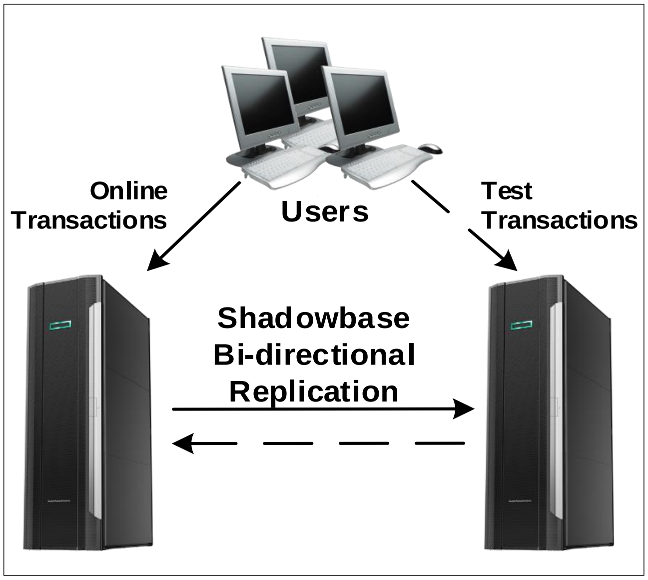


Figure 4 – Sizzling-Hot-Takeover (SZT)

If the active node fails, all that is required for failover is to switch the users or their transactions to the standby node. The switch can be done in seconds to sub-seconds, leading to very small RTOs. To operate in this mode, it is essential that the replication engine that is used allows the application processes to also open the target database for read/write access.

The SZT configuration has another big advantage over active/passive systems and is more commonly used over active/passive architectures, which is the absence of *failover faults*. In active/passive systems, the standby system is not actively involved in the application. Consequently, you do not know whether it is really operational, and whether a failover to the new system will even work. In an SZT system you know that the backup node is working because it can easily be exercised during normal operations without requiring an outage of the primary system, by periodically submitting test or verification transactions to the application to ensure proper operation. Consequently, failover can be automated, which is a requirement if very short RTOs are to be satisfied (seconds or sub-seconds).

Finally, the SZT system can be optionally configured with Shadowbase bi-directional replication already up and running so that after a takeover, there is a backup system available as soon as the former primary node is recovered. With bi-directional replication enabled, the takeover node queues the changes that it is making to its copy of the database so that the failed node can be quickly resynchronized upon recovery.

In summary, SZT systems solve the potentially long RTO times and potential failover risks associated with active/passive systems, but without some of the complexities of fully active/active systems.

Active/Active Systems

An active/active configuration takes the SZT arrangement one step further, and provides continuous availability. Rather than routing all transactions to one node as in the SZT case, all nodes in an active/active network may be simultaneously processing transactions for the same application using their local copies of the application database (Figure 5). Shadowbase bi-directional data replication is configured between each node pair so that any change that an application makes to its local copy of the database is immediately replicated to the other nodes in the application network (each database is acting simultaneously both as a source and target database). Thus, all nodes (and thus, applications) have the same view of the application database and all can process online transactions.

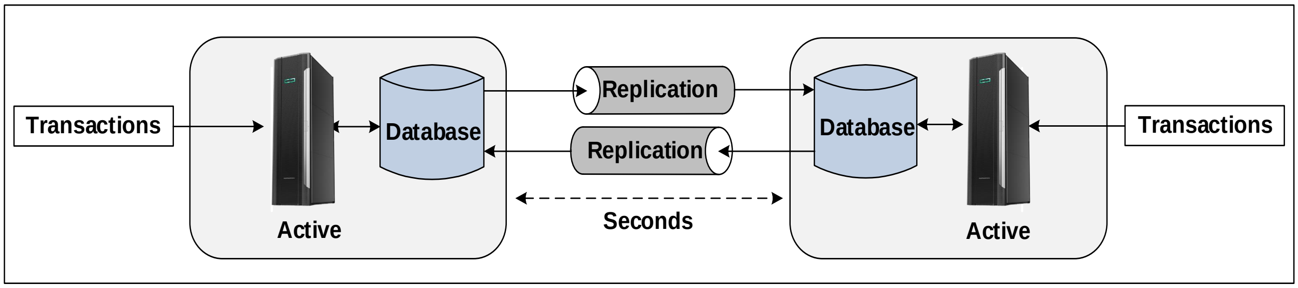


Figure 5 – Active/Active Bi-directional Replication

Active/active systems provide several advantages compared to active/passive systems:

- There are fewer users affected by a failure.
  - In an active/passive architecture, all users are immediately down when the active node fails.
  - In an active/active architecture, only the users connected to the failed node are slightly inconvenienced while they wait for their request to be rerouted to a surviving node.
- Failover of the subset of users affected is very rapid, supporting RTOs measured in sub-seconds to seconds.
  - In an active/active architecture, transactions only need to reroute or reconnect users to a surviving node to recover from a node failure.
- Failover is easily, periodically, and safely tested since all nodes are known to be operational because they are actively processing transactions.
  - Such testing is very difficult to accomplish for active/passive systems, resulting in failover faults which delay the restoration of service.[11]
  - When a failure does occur in an active/active architecture, it is to a known-working system, providing peace of mind for management.
- Planned downtime is eliminated by taking down one node at a time, performing upgrade or maintenance activities on it, and then returning it to service.
- An application uses all available processing capacity. There is no idle standby system.

Data Collisions

A consideration with active/active systems is the possibility that the same data item might be changed in each copy of the application database within the replication-latency interval. If this event happens, the change is replicated to the other database, and the replicated change overwrites the original change in each database. Now both databases are different, diverge, and are both wrong; this situation is called a *data collision*.

Some active/active applications can avoid data collisions, or in some cases they may not matter even if they do occur. One frequently used method to avoid data collisions is to partition the database between the nodes. For example, if the database can be partitioned by customer range so that only one node updates any given partition of customer data, data collisions cannot occur. In this case, the application must route all updates to the proper node that owns the customer data partition being updated.

The preferable method is for data collisions to be avoided, since this avoidance allows transactional requests to be performed by any system in the network, thereby enabling optimal load-balancing between the systems (this is known as the *route anywhere* model). However, if data collisions are possible, they must be detected and resolved. Shadowbase technology provides several mechanisms for automated detection and resolution of data collisions; in a future release, it will support bi-directional *synchronous replication*, which eliminates the possibility of data collisions altogether.

[11]This benefit is also true for SZT configurations.

HPE Shadowbase Solutions and HPE Pathway Domains – Perfect Together

The combination of Shadowbase data replication and PD across nodes offers a powerful business continuity architecture to meet the requirements of continuous application and continuous data availability. PD and Shadowbase replication together provide geographically dispersed application services and database services. The data is always made available on a backup node by Shadowbase data replication, and the application is always made available on a backup node by PD. When a node outage occurs (for whatever reason), ACS automatically detects it and switches all user request traffic to the backup node(s), which can continue processing because the data on those nodes has been synchronized with the failed node by Shadowbase technology. Thus, all the necessary components (applications and data) are in place to maintain business service with minimal downtime (if any).

Therefore, it makes sense to take a deeper look at the combination of Shadowbase replication and PD across nodes, for each of the main business continuity architectures discussed above.

Active/Passive Systems

Without using PD in this case, it is possible to have duplicate PATHMON environments up and running on both the active and passive nodes with transaction requests routed to both nodes. But the Pathway applications on the passive node must open the database on the *active* node for updates, in order to avoid data corruption (i.e., only one copy of the database, the production copy, can be updated at a time). No updates are directly made to the backup database; therefore, the Source node is updated first, and then data replication transports and applies the data to the Backup node, which is kept synchronized in near real-time. The passive node incurs the overhead of network TMF transactions, but that is not a big problem for co-located systems.[12] Upon an active node failure and a failover to the passive node, it is necessary to switch the users connected to the active (now defunct) node to the passive (surviving) node, and redirect the applications on the passive node to open and use the backup database. However, these manual actions and the time taken to perform them can lead to extended application outages. The use of PD and the other business continuity architectures discussed below are superior (from an availability perspective) and safer (from a risk of failover perspective).

Pathway Automatic Weights Reconfiguration (AWR)

Using PD, it is possible to specify a weighting factor of zero for a PATHMON, so the Pathway application (servers) can be up and running on both the active and passive nodes (with the *local* database open in each case), with a PD configured across both nodes. The PATHMONs on the passive node are given an initial weighting factor of 0. Transactions can be routed to either node, but when the Pathway requesters on the passive node receive these transactions and issue PATHSEND requests to servers, those PATHSEND requests are all routed by ACS to the PATHMONs on the active node (because the servers on the passive node are weighted at zero), thereby ensuring no updates are performed on the passive database.

When PATHMONs Become Unavailable

If the PATHMONs on the active node become unavailable, ACS automatically detects this and begins routing PATHSEND requests to the passive node PATHMONs, with no operator intervention required - this feature is called Automatic Weights Reconfiguration (AWR). The servers on the formerly passive (now active node) can now update the local database which has been kept synchronized with the active database by Shadowbase uni-directional replication, thereby maintaining application service availability.

In a nutshell, AWR automates the process by which a passive (zero weighted) PATHMON becomes an active PATHMON, in response to the unavailability of one or more active PATHMONs in a domain (i.e., it supports *automatic Pathway server outage failover*).

With AWR, ACS periodically monitors the state of each PATHMON in the domain, and when it detects that all PATHMONS in the domain with an original non-zero weight (i.e. the active PATHMONs) are not available, it automatically rebalances the client request load to the original zero-weighted PATHMONs (by changing the zero weight to a non-zero weight), moving them from passive to active state, thereby maintaining application availability. ACS will automatically detect when the original zero-weighted PATHMONs are recovered, and reset the PATHMON weights back to their original settings.

[12] However, co-location does not lead to high levels of availability, since localized or even regional issues such as power outages, fires, or floods can affect all systems and literally destroy the data!

AWR introduces new configuration options for the PD ACSCTL file:

- It can be enabled or disabled;
- the frequency that ACS checks on availability of other PATHMONS in the domain can be specified;
- how long ACS should wait after detecting a PATHMON outage before reconfiguring weights can be specified;
- how long ACS should wait before resetting the PATHMON weights to their original settings after failed PATHMONs are recovered can be specified; etc.
- Additionally, there are also operator commands that were added to PDMCOM to manually perform the rebalancing/fallback operations, etc.

So, what benefit does this configuration provide? One of the most time-consuming issues when performing a failover in an active/passive environment is switching all of the users from the active to the passive node (updating and deploying new network routing tables, starting servers, opening the database, etc.).

First, some of the users would already be connected to the standby (not passive) system with applications running and would not see any outage at all.

Second, all that would be necessary is to switch the remaining users to the already up and running standby (now active) node, thereby reducing failover times. The standby node is a known-working state,[13] thereby eliminating failover faults. Last, but by no means least, the capacity of the backup system would be better utilized for application processing (since client requests can be executed on the passive system).

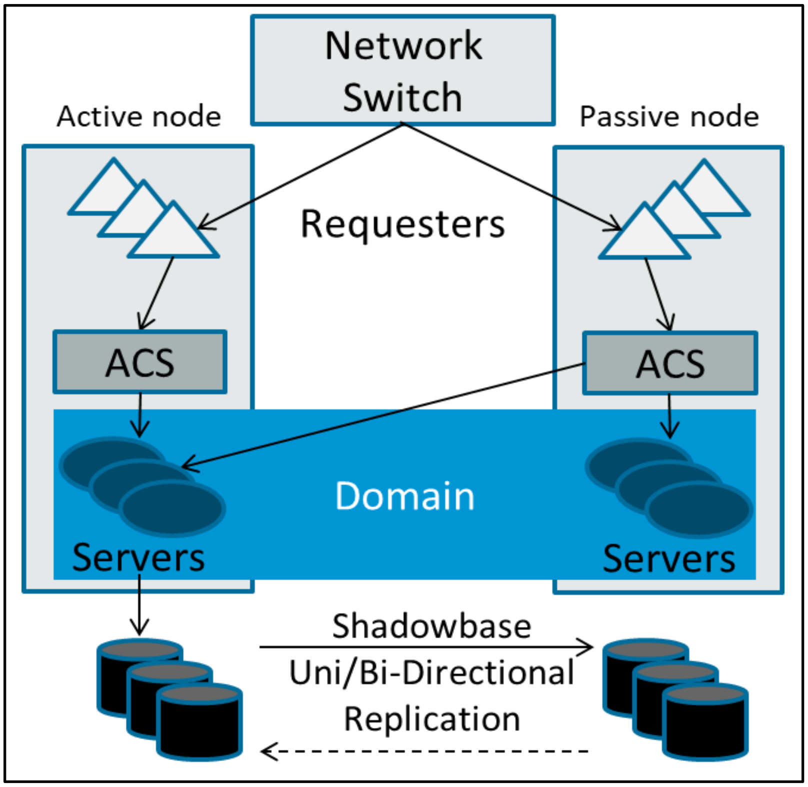


Figure 6 – Pathway Domains in Active/Passive and Sizzling-Hot-Takeover Models

In this architecture (Figure 6), one significant factor to note is that Shadowbase technology allows applications on the passive (or standby) node to have the local (passive) database open for read/write access, even while replication to it is taking place, which is *not* the case for all NonStop data replication software. In the PD case described above, this feature allows the Pathway servers on the passive node to be fully up and running with the local database open for read/write access (there is no possibility of data corruption if the passive database is updated; this cannot happen as ACS will not route any requests to PATHMONS configured with a weight of zero). Without the need for the servers to be either started or switched from read-only to read/write access at takeover, this feature further reduces takeover times and further reduces the potential for failover faults (since the servers are already running, there will not be any server startup issues).

This combination of Shadowbase data replication and PD with its PATHMON, ACS, and AWR features enables the automated failover of Pathway servers to a standby node. After a failure event of the primary (active) node, the passive application does not need to be started, thereby minimizing takeover times, reducing complexity, and the possibility of errors arising from manual intervention. It is a very powerful and popular business continuity architecture, without the complexities of an active/active configuration.

[13] "Standby" servers are already up and running with the local database open. Test transactions can periodically be sent to these servers to verify end-to-end functionality.

Sizzling-Hot-Takeover

This architecture (Figure 8) is generally identical to the active/passive case described above (Figure 6) even though all updates are only being made to the database on the active node. This architecture is an improvement over the basic active/passive architecture because when a takeover by the passive node occurs, it is already set up to replicate those changes back to the downed node once it is recovered, eliminating reconfiguration of the replication engine and speeding the failover time, and the time required to put the recovered node back into service (thereby re-establishing a backup system). And, with this clever testing approach, the reverse replication path can be verified while the active node remains running, by updating test data in test accounts in the standby database.

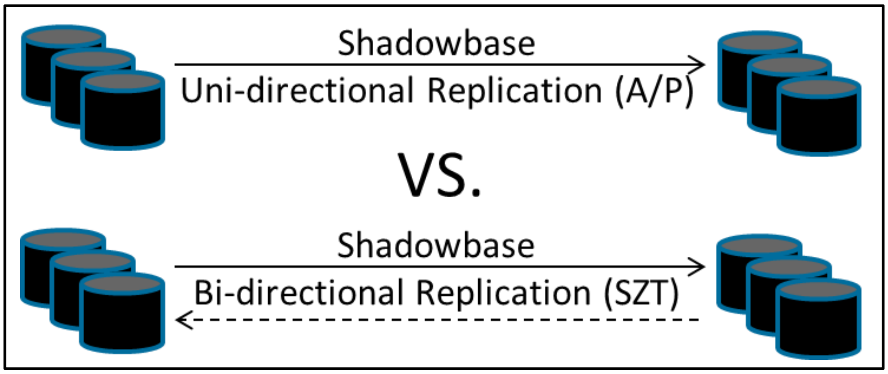


Figure 7 – Shadowbase Uni-directional (A/P) vs. Bi-directional (SZT) Replication Architectures

Active/Active Systems

This type of architecture (Figure 8) is where Shadowbase software and PD capabilities fully complement each other to provide significant benefits, with PDs providing active/active application processing and Shadowbase replication providing active/active database replication.

Each PATHMON in an active/active configuration is included in a PD configuration on each node, with a subset/superset of the same applications (serverclasses) running in each of them. All PATHMONs in the domain are configured with a weighting factor > 0, so workload can be distributed across all of them. Transactions can land on any active node, and ACS handles the distribution of workload between the PATHMONs configured in the domain. Shadowbase technology is providing bi-directional replication to keep all active database copies synchronized, and enabling read-write database access to applications even while replication is in process.

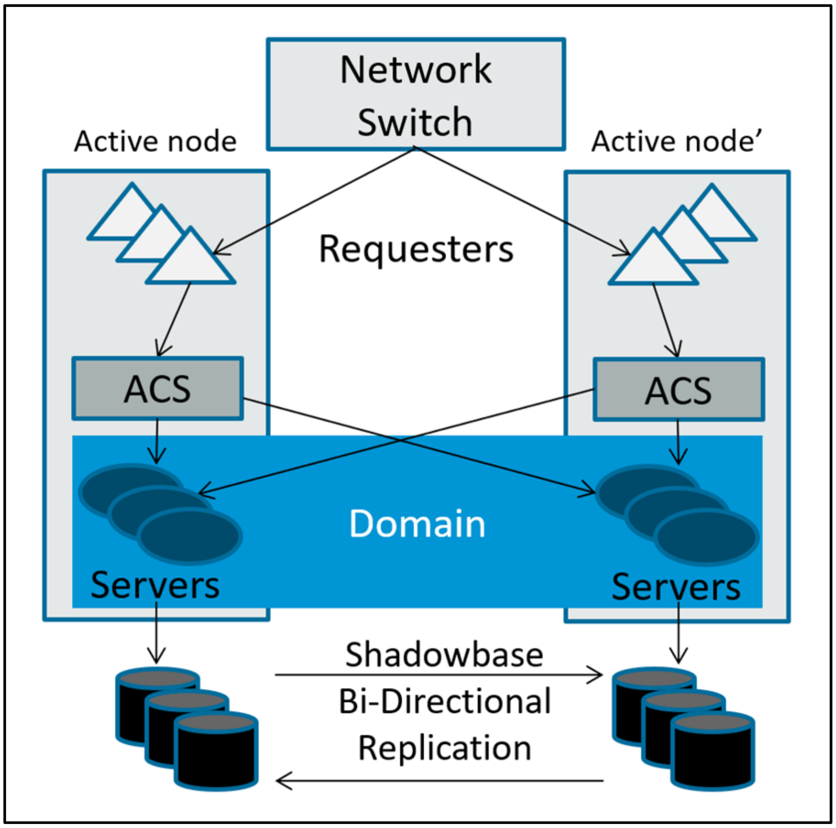


Figure 8 – Pathway Domains in Active/Active Mode

When a node is taken down for maintenance, or suffers an unplanned outage, ACS detects that the PATHMON(s) running on that node are no longer available, and stops sending requests to them. The workload is thus automatically routed by ACS to the remaining node(s). Since each node also has a current copy of the online database (due to Shadowbase bi-directional replication), business transactions can proceed with no noticeable outage.

Once the down node is recovered, ACS automatically detects that the PATHMON(s) configured on that node as part of the PD are once again available, and begins routing requests to them in accordance with the load balancing algorithm.
Similarly, Shadowbase replication detects the system is once again available and begins replicating to it all the queued updates (those updates which occurred while the downed system was offline), as well as ongoing transactions.

Likewise, updates now occurring on the newly recovered system are replicated to the other system to ensure business continuity. All of these steps happen with no outage, and no operator action except to recover the failed system. The failover is fast, transparent, and automatic.

As discussed above, with any active/active environment, there is a possibility of data collisions. There are ways to partition data and/or applications to avoid this problem (for example, to partition the database such that each handles a separate subset of user accounts). This partitioning can be done by network connection routing (assuming the end user is remotely connected to the HPE NonStop server systems).[14]

Even better is the active/active *route anywhere* model, where ACS load-balancing and partitioning issues are avoided completely by allowing any request to be processed by any node. This model applies naturally to some applications (e.g., ATM transactions where it is highly unlikely the same card is being used for two different transactions simultaneously). For others, it can be achieved by using the built-in capabilities of Shadowbase technology for automatic data collision detection and resolution. But, for some applications, data collisions are simply not acceptable, period. Additionally, if some form of partitioning is not feasible, an active/active architecture simply cannot be used. For these applications, an in-plan future release, called *Shadowbase ZDL+*, enables active/active configurations with no possibility of data collisions. Shadowbase ZDL+ uses unique patented synchronous replication technology to achieve this goal.[15]

Tiered Application Architecture

The combination of PD and Shadowbase capabilities maps into the Indestructible Scalable Computing (ISC) paradigm for HPE NonStop servers. ISC is an HPE NonStop initiative to enable continuously available and scalable, multi-node system architectures. In this model, the application may be tiered, split into two parts, a front-end (FE) component handling more compute intensive functions such as presentation services, and a back-end (BE) component handling database operations.[16] Each component is also replicated across multiple systems to ensure continued availability in the event of an outage. With PD across nodes, the front-end and back-end applications can be configured within a domain, such that should a back-end node in this configuration become unavailable, ACS on the FE nodes automatically route requests to the remaining BE node(s). If an FE node fails, then standard network switching routes requests to the remaining FE node(s). The only missing piece is to ensure the data remains accessible should a back-end node become unavailable, which is where Shadowbase bi-directional data replication comes in.

[14]The partitioning could also be done by the user writing a Pathway server, which acts as the distributor for all requests and forwards the requests to the appropriate system for processing. But obviously, this step somewhat defeats ACS load-balancing and the benefits of PDs. It would be a nice enhancement however, if ACS had the ability to include in the routing algorithm the notion of *data dependent routing*, where users could configure key fields in the PATHSEND request, which ACS uses to distribute requests to certain specific nodes. This enhancement could save the user from special applications programming to achieve partitioning and the avoidance of data collisions.

[15]Contact Gravic for further information about the availability of Shadowbase ZDL+.

[16]Please see Application Capacity Expansion (ACE) for more information on this model.

In the example shown in Figure 9, the same domain configuration is defined on each of two front-end nodes, the domain comprising two PATHMONs, one on each of two back-end HPE NonStop server nodes. The back-end nodes are in an active/active configuration, with Shadowbase bi-directional replication in place between them, to keep the databases synchronized. The same serverclass (SC1) is running under each back-end PATHMON, so any PATHMON in the domain can service any request for that serverclass. Network routing is employed to distribute remote users between the two front-end nodes.

Under normal conditions, PATHSEND requests initiated on either front-end node are routed by ACS to either back-end PATHMON in the domain. Changes made to either database are replicated to the other by Shadowbase replication. If one of the front-end nodes were to fail or be taken down for maintenance, network routing would shift all remote users to the remaining front-end node, and service would continue uninterrupted.

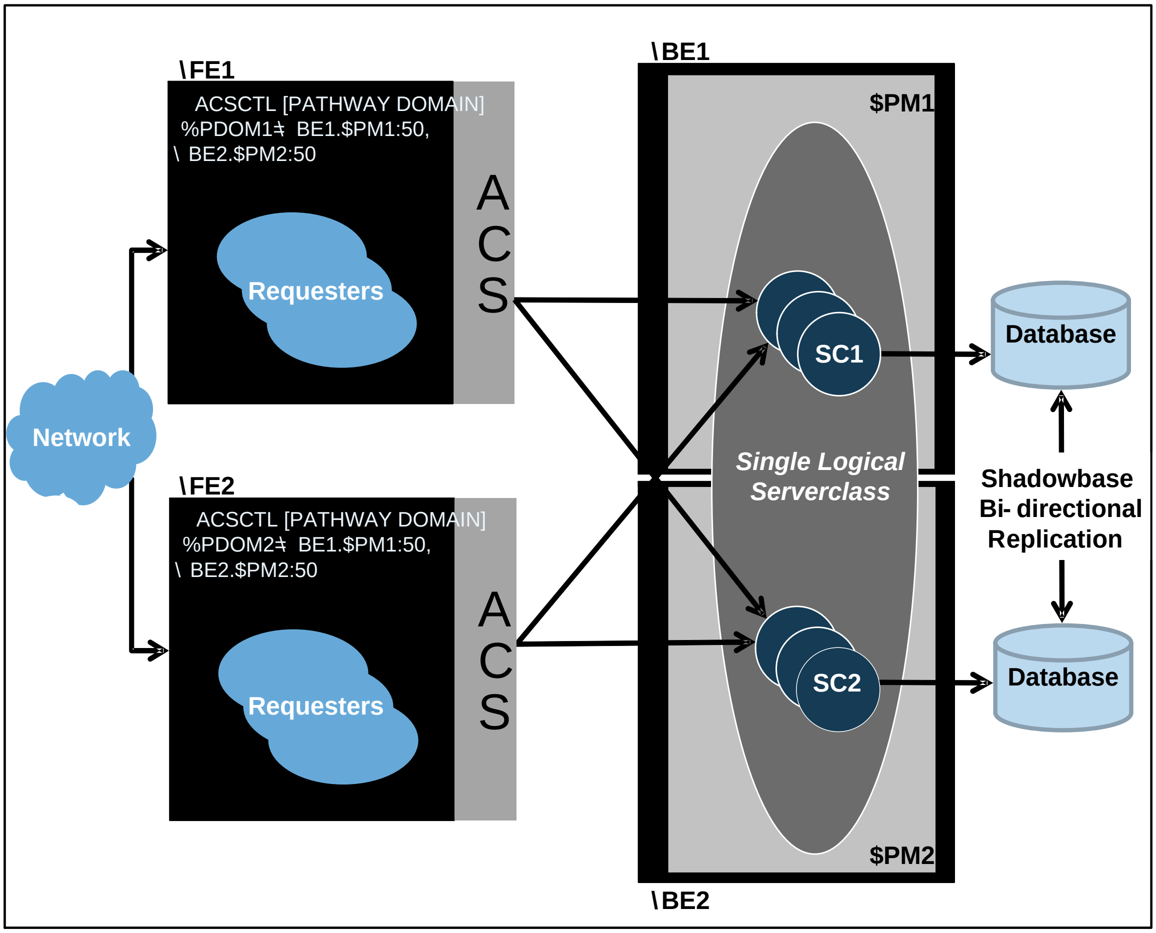


Figure 9 – Pathway Domains Application Architecture Split into Front-end (Presentation) and Back-end (Database) Components, in an Active/Active Configuration

If one of the back-end nodes were to fail, or be taken down to perform system maintenance, this failure would be detected by ACS on each front-end node which then automatically route requests only to the remaining back-end node. Once the down system is restarted, Shadowbase architecture replays all the queued updates made while the system was down, bringing its database back into synchronization. The PATHMON on the recovered back-end node can then be started. ACS automatically detects and reintegrates the PATHMON into the domain and begins sending requests to it again (as demanded by load).

Zero Downtime Migrations (ZDM)

It is worth making a point about the use of PD and Shadowbase architecture in the context of eliminating planned downtime for system upgrades and migrations. The use of the two products to avoid planned outages is as applicable as it is to avoiding unplanned outages.

In the case of system maintenance or upgrades, the online workload is routed by ACS to another system in the domain to maintain business services while the maintenance is carried out. Once the maintenance is completed, the system is brought back online and begins processing transactions. Then, if required, another system can be taken down for maintenance while the original system takes care of the online processing, and so on.

This technique is called a *rolling upgrade*, where each system is taken out of service in turn, and all the while business services seamlessly remain available to end users as ACS automatically redistributes the workload to other available systems in the domain. Shadowbase technology brings the database of the upgraded node back into synchronization once it is returned to service. The technique can also be used to migrate to a new system without any downtime, called a *zero-downtime migration* (ZDM).

Big-bang upgrades requiring production application outages are no longer required, even when performing very disruptive changes to your environments. The Shadowbase ZDM approach dramatically reduces the inherent risk with the classic big-bang approach.

Planned Downtime and Disaster Recovery

A common requirement while performing system maintenance is that disaster recovery capabilities are not compromised in the meantime. In a two-node architecture, while one system is offline for maintenance, a failure of the other system brings business services to a halt, which is unacceptable for many applications. Companies are increasingly deploying multi-node configurations to meet this need. In such configurations, multiple systems are capable of taking over the business services, located in different datacenters. In a PD, there can be up to four such datacenters.

A typical configuration is to have two local systems in each of two datacenters with the local systems operating in either active/passive or SZT modes, as described below (Figure 10).

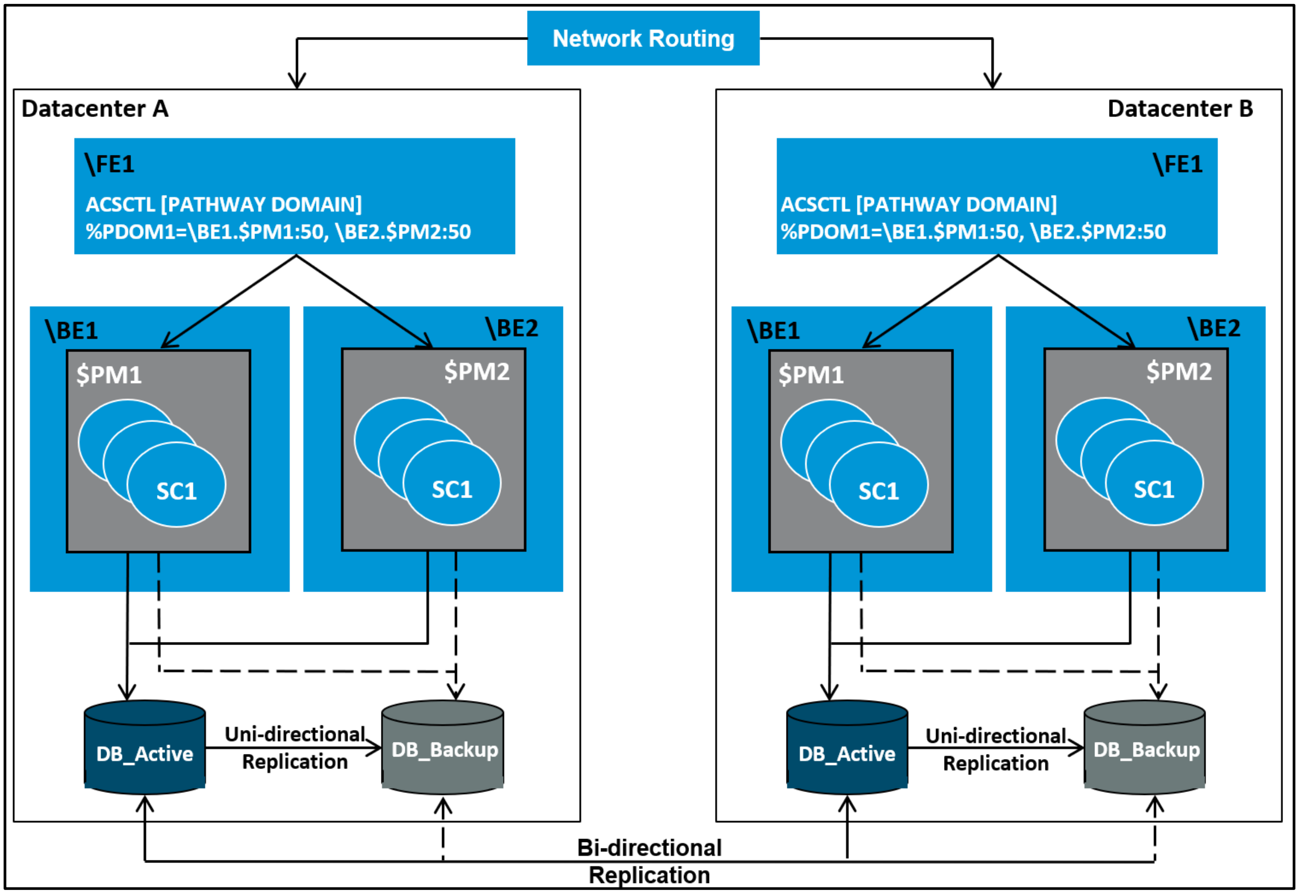


Figure 10 – Pathway Domain Configuration Supporting Planned Downtime While Maintaining Disaster Recovery Protection

In such a configuration, while one system is down for maintenance, the other local system can take over active processing. If either datacenter should then experience a complete outage, there will remain at least one other system available to continue active processing.

There could be just a single active system in one datacenter, with uni-directional replication between it, and the other datacenter to keep the databases synchronized. But, to maximize the utility of the four systems, it makes more sense to use an active/active configuration between the two datacenters, with systems configured as part of PD and workload distributed between them (one system active in each datacenter, the other used as a local backup).[17]

Shadowbase bi-directional replication is used to keep the pair of active databases synchronized, with uni-directional replication between the active and backup systems in each datacenter. The system architects decide whether or not the front-end routing is performed on a separate system in a tiered architecture with a database back-end tier (consisting of two nodes at each datacenter); PD's do not require this approach, but do support it.

Availability Nirvana

Taking this architecture to completion brings us to "Availability Nirvana". Rather than having just two nodes running active/active, while the other two nodes run passive, all four nodes can be run in an active/active configuration, with Shadowbase bi-directional replication keeping all four copies of the database synchronized (Figure 11).

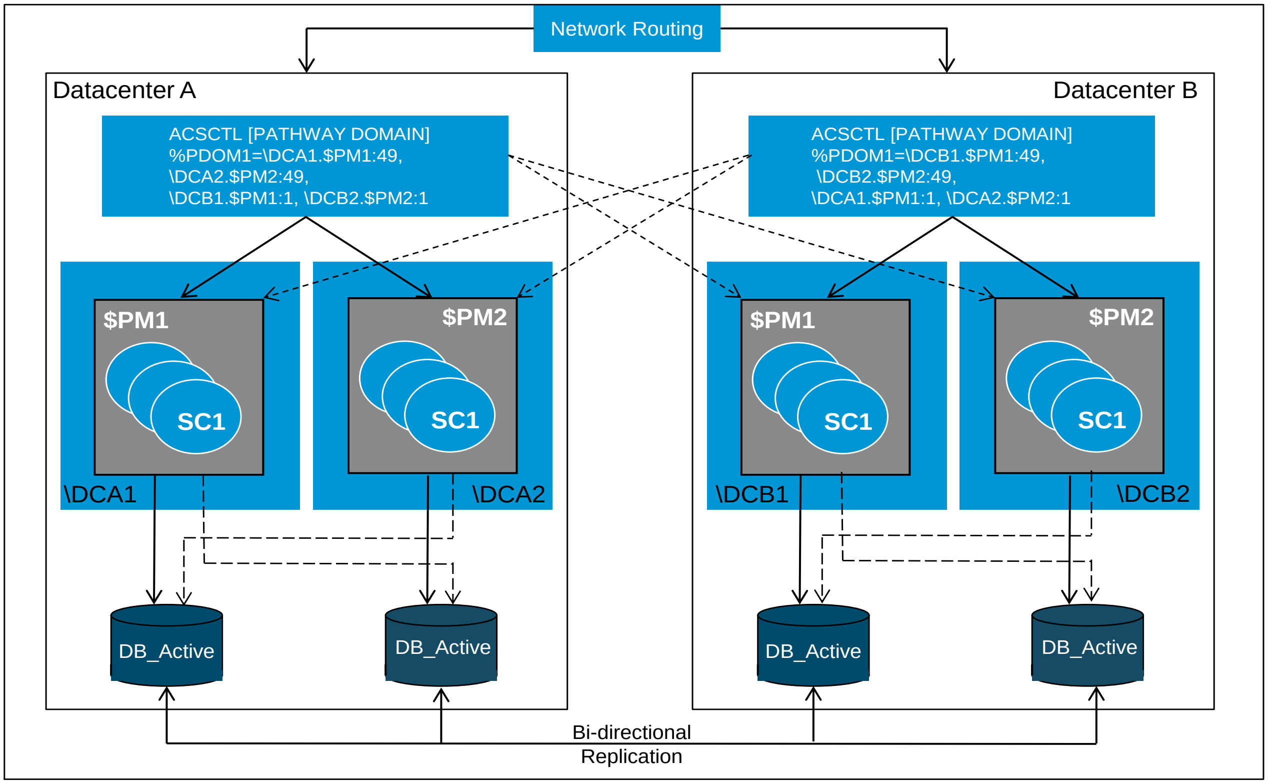


Figure 11 – Planned Downtime While Maintaining Business Continuity Protection, Fully Active/Active Architecture

PD can be configured on each node to span the PATHMONs across all four nodes. With this architecture, it does not matter what node a request lands on, or what nodes are unavailable. With the database consistent across all four nodes, ACS can route requests to any PATHMON on any node which is available. During the outage of a single local node, or a whole datacenter, whether planned or unplanned, ACS will route requests to remaining nodes, maintaining service availability, with no reconfiguration or operator action.

[17]This architecture also benefits from the ACS enhancement to allow zero-weighting for certain PATHMONs, unless other PATHMONs become unavailable, to better control request distribution in favor of local nodes (when available).

It is also possible to take one of the databases within a datacenter offline for maintenance, and have the applications on both nodes remain active, both accessing the other local database. This architecture is truly the definition of continuous availability, and can now be implemented using PD together with Shadowbase data replication software.

If an active/active architecture is impossible for a particular application (for example, because data collisions cannot be avoided nor tolerated), this multi-node setup can be run in a quasi-active/active configuration. Applications will be active on multiple nodes and ACS will distribute requests between them. However, all database updates are to a single online database copy attached to one system, with Shadowbase uni-directional replication keeping the other database copies synchronized.

This configuration does incur network TMF transaction overhead, but also utilizes multiple systems to run the online business applications. The applications on the backup system are known to be up and running correctly (in practice, this setup is just like the SZT architecture mentioned above in Figure 6). Therefore, if any outage of the active system occurs, the backup system can be quickly brought online, with little likelihood of failover faults. For such applications where full active/active is not possible due to data collisions, the forthcoming Shadowbase ZDL+ feature as described above would allow these applications to run in a fully active/active environment without the overhead of network TMF transactions.

Another possible configuration (to meet the need to avoid a single point-of-failure during maintenance) is to have three systems located across three geographically distributed datacenters. Again, if one system is down for maintenance, and a simultaneous event occurs which takes out another datacenter, there is still one system available to continue providing business services. Whether this configuration is better than four systems located across two datacenters really depends on the costs of an additional datacenter vs the costs of an additional system, and upon whatever existing IT infrastructure is available.

Summary

It takes two to tango! It is of no use having data available on alternate/takeover systems if the applications required to provide the business services are not also available when needed. HPE Shadowbase data replication solutions provide all the capabilities necessary to keep databases replicated and synchronized across multiple systems, whether it is a uni-directional (active/passive), bi-directional (SZT, active/active), heterogeneous, or homogeneous environment. The introduction of Pathway Domains (PD) across nodes and the AWR feature for HPE NonStop servers also helps to ensure that applications remain available, even when catastrophic failures take out a system, an entire datacenter, or a geographic region.

By configuring applications within a PD across multiple HPE NonStop nodes, in the event of planned or unplanned outages, user requests are automatically routed to the remaining systems, thereby preserving application availability. In addition, application scalability is increased by automatic workload distribution across those systems, and the removal of Pathway single system limits. Since Shadowbase technology is making sure the data is available on those remaining systems, the combination of Shadowbase software and PD provides a firm basis to achieve increased levels of availability for your business services.

In short, Pathway Domains and HPE Shadowbase data replication solutions are perfect together!

International Partner Information

Global

Hewlett Packard Enterprise
6280 America Center Drive
San Jose, CA 95002
USA
Tel: +1.800.607.3567
www.hpe.com

Japan

High Availability Systems Co. Ltd
MS Shibaura Bldg.
4-13-23 Shibaura
Minato-ku, Tokyo 108-0023
Japan
Tel: +81 3 5730 8870
Fax: +81 3 5730 8629
www.ha-sys.co.jp

Gravic, Inc. Contact Information

17 General Warren Blvd.
Malvern, PA 19355-1245
USA
Tel: +1.610.647.6250
Fax: +1.610.647.7958
www.shadowbasesoftware.com
Email Sales: shadowbase@gravic.com
Email Support: sbsupport@gravic.com

Hewlett Packard Enterprise Business Partner Information

Hewlett Packard Enterprise directly sells and supports Shadowbase Solutions under the name ***HPE Shadowbase***. For more information, please contact your local HPE account team or visit our website.

Copyright and Trademark Information