High Availability with Postgres Plus Advanced Server

An EnterpriseDB White Paper

For DBAs, Database Architects & IT Directors
December 2013

EDB
ENTERPRISEDB

Table of Contents

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

Introduction

High Availability (HA) refers to an overall system design for achieving a pre-arranged Service Level Agreement (SLA) and is defined by some number of 9's referring to a percentage of uptime. For example, a system designed to be available to its users 99 percent of a calendar year has two 9's of availability. A system with five 9's of High Availability means the system is down, due to disruptions or service updates, no more than 5.26 minutes per year so that it's available 99.999 percent of a calendar year.

When referring to High Availability for any database server, it is important to understand that this refers to the uptime of the overall system, and not just the database server. The database server, for example, cannot be more available than the hardware it runs on or the datacenter where the system resides.

However, the database server can be deployed in such a way that it can failover should its supporting hardware fail. The time it would take for such a failover would cut into the annual SLA downtime budget. So, in the course of a year, if the database server fails due to hardware failure and it takes 20 seconds to recover and there are no other interruptions in service during the same year, the system would exceed five 9's of availability. If the database hardware were to fail every other day and it takes 20 seconds to recover each time, the system would fall short of achieving an availability SLA of five 9's.

So the proper question is not, "What level of 9's can Postgres Plus Advanced Server achieve?" The real question is, "What level of 9's can be achieved when it is part of an HA solution?"

The answer is that Postgres Plus Advanced Server in a properly configured HA system, with most common workloads and proper upgrade planning, can achieve five 9's of availability. We have, in fact,

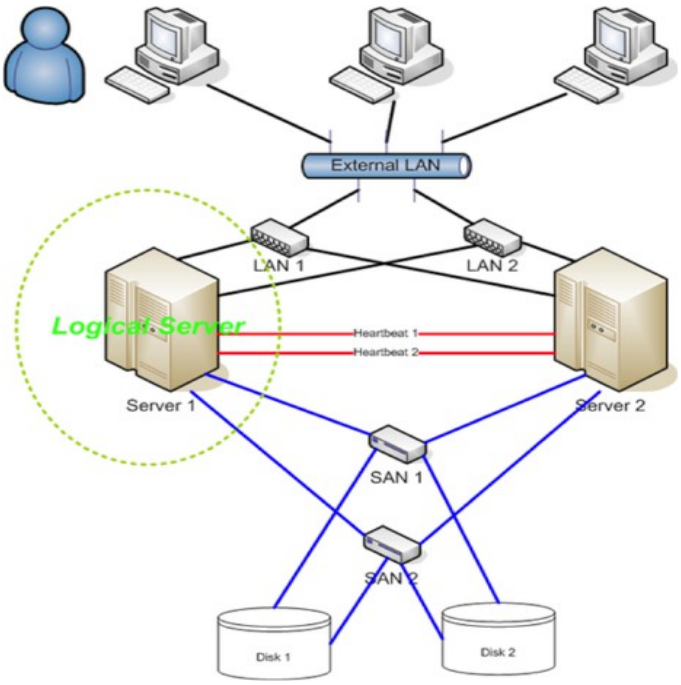
© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

had Postgres Plus Advanced Server customers run for years without any service interruption. What's important is the database is deployed as part of a system design that allows failover to a secondary server when common hardware failures occur. The following is an examination of several approaches to HA design and a look at the new EDB Failover Manager.

Active/Passive Clustering

One way to achieve High Availability is to deploy Postgres Plus Advanced Server in an active/passive cluster solution that allows the database to failover to a secondary system in the event of a hardware failure. Active/passive clustering solutions can be used to meet the requirements of a five 9's system, or 99.999% availability.

Figure 1 – Active/passive cluster system



© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

The basic components of an active/passive configuration as shown in the above diagram are two servers, shared storage, an interconnect and clustering software. The servers can be two physically separate machines or slices of a single machine (such as a VMware image or logical partition). One of these servers is considered the active node and the other one is the passive node. The shared storage is connected in some way to both servers so that each server can, at the appropriate time, mount any of the file systems on the shared storage.

The cluster software can be any of the popular ones such as Red Hat Cluster, Veritas Cluster, Linux-HA, etc. The cluster software is responsible for monitoring the health of the cluster and taking action when a failure is detected. The cluster software utilizes the interconnect to send a heartbeat signal between the servers. The successful receipt and acknowledgement of this heartbeat means that the cluster is healthy and no action is needed. If the heartbeat signal is lost, that means something has happened and a failover needs to take place. The failover process is to shutdown the failed node, mount the file systems on the passive node and restart the services on the passive node, thus making it the new active node.

Postgres Plus Advanced Server in this scenario would run on the active server. The cluster-ware would be configured to start up the Postgres Plus services on the passive node if the active node fails. Upon completion of this failover, the passive server becomes the active server and Postgres Plus will be ready to continue processing your business needs on this new active server.

This type of clustering architecture is proven, reliable, relatively inexpensive and ensures fast failover times. The entire failover process can take as little as 8-10 seconds, thus ensuring your ability to continue doing business with minimal interruption. However, this approach is complex and requires significant skill. Further, the cluster-ware solution may be a separately licensed product and therefore, could add to the overall cost and maintenance of the system.

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

Standby Database

Standby database is a configuration where the production database ships its transactions, either at the transaction level or at the log level, to a remote site. These transactions are applied to a standby database in order to remain in sync with the production database.

For this type of configuration, the database must be running in archive log mode. Then either the transactions are shipped to the standby server as they happen or when the transaction logs (WAL logs) are copied to the standby server. Once transactions reach their destination, they can be either automatically or manually applied to a database that is in recovery mode. Recovery mode simply means that the database is applying the received transactions/logs.

As mentioned, there are two ways to ship transactions to the remote site. The first is called log shipping, which means the database is configured to copy the transaction log to the remote site once the transaction log is filled to a certain size. The disadvantage of this configuration is that you have to wait for the transaction log to fill before it's shipped and thus your standby database is out of sync with the production database for longer periods of time. Also, in the event of a failure of the production database system, you could lose all of the transactions in the log file that hasn't yet been shipped as well as the log file in transit to the remote site.

The second way to ship transactions to the remote site is to use the Streaming Replication feature of Postgres Plus Advanced Server. This feature continuously ships transactions to the remote site as they occur. This shipping of transactions can be done either asynchronously or synchronously. The process means simply that when a transaction is performed on the production database, that transaction will be written to the production database's transaction log and written, asynchronously or synchronously, to a file on the remote site. Once the write to the remote site is complete, that transaction will be applied to the standby database. This obviously keeps the standby database more synchronized with the production database and cuts down on

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

possible data loss and recovery time.

In the event of a site failure, the standby database can be brought up in full production mode and your applications can access the 'new' production database with minimal disruption in business operations.

In a streaming replication setup, one needs to make a choice between asynchronous or synchronous replication. Asynchronous replication offers the best performance but you have the possibility of data loss because it's not guaranteed that a transaction committed on the primary database is written to the remote node. Synchronous replication provides that guarantee but because you are performing a two-phase commit (i.e. the write to the primary database and the remote node must both complete or they both fail) any network latency affects the performance of the write and thus could have an overall throughput impact on your application. Applications that require zero data loss protections that synchronous replication provides need to ensure they have a very low latency and reliable network connection between the primary and remote server.

Replication

Replication solutions such as Slony or xDB Replication Server can also be used to set up a highly available system. Slony and xDB SMR also provide read scalability since the replicas are open for read only access to the data that is being replicated. xDB also provides 'write availability' in its multi-master replication mode where any data can be updated on any node. These systems provide HA since there is a duplicate copy of the data present on a remote node that can be promoted to a master, i.e. in a master-replica setup, or is already a master, i.e. in a multi-master setup, in the event of a failure.

One advantage with xDB multi-master replication is that it allows writes to occur on any node. This comes in handy when your sites are geographically dispersed and thus your ability to write locally provides a performance benefit to the local users. Also, in the event of a failure of one of the masters, there is no startup of a new node or database

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

recovery required because the database is already open in complete read/write mode. However, transactions that were committed to the failed node but not yet replicated to the other nodes will need to be identified by the user/application and resubmitted. That is a manual process.

There are some challenges to consider with this approach. Both Slony and xDB are trigger-based replication solutions so they introduce overhead into the system. Further, there is no automatic failure detection and promotion of read replicas to masters in the event of a failure; users must implement their own detection mechanism and then run through the proper procedure to promote a replica to a master.

The Critical Link: EDB Failover Manager

With the exception of the active/passive clustering solution, the above solutions are missing a critical piece: automatic failover. This has long been a gap in Postgres High Availability solutions and has forced users to build their own or attempt to cobble together various other tools to solve this problem. To close this gap, EnterpriseDB developed and introduced EDB Failover Manager, a reliable, cost-effective solution for Postgres that eliminates the need for third-party tools to ensure High Availability. The solution is the newest in EDB's collection of integrated enterprise-class tools that target the needs and requirements of global enterprises increasingly adopting Postgres for more workloads to limit their use of more costly, traditional database vendors.

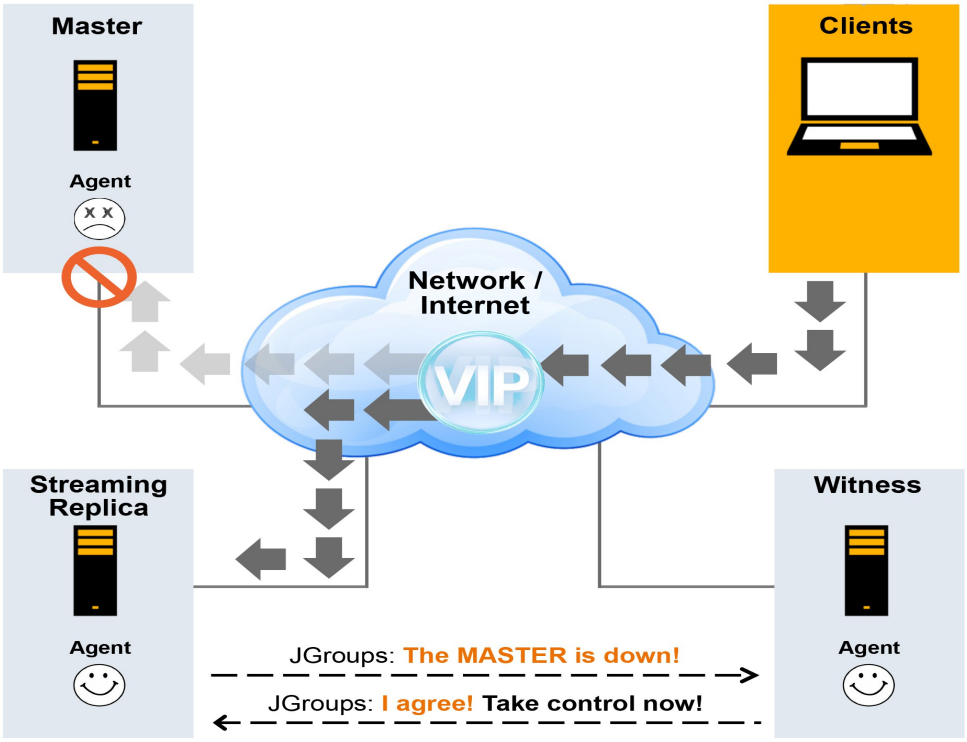
EDB Failover Manager complements the Postgres Streaming Replication feature for setting up standby databases by allowing the Master node to automatically failover to a standby replica node in the event of a software or hardware failure on the master.

Traditionally, the expression *cluster* refers to a single instance of Postgres Plus Advanced Server managing multiple databases. In this case, we'll use the term cluster to refer to a Failover Manager cluster. A

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

Failover Manager cluster consists of a Master agent, Standby agent and Witness agent that reside on separate servers in a cloud or on a traditional network and communicate using the JGroups toolkit.

Figure 2 - A failover manager scenario employing a Virtual IP address.



A closer look at the agents involved in EDB Failover Manager reveal the processes that reside on the following hosts on a network:

- Master - The Master node is the primary database server that is servicing database clients.
- Standby - The Standby node is a streaming replication server being synchronized with the Master.
- Witness - The Witness node confirms assertions of either the Master or the Standby in a failover scenario.

JGroups provides technology that allows EDB Failover Manager to

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

create clusters whose member nodes can communicate with each other and detect node failures. For more information about JGroups, visit the official project site at http://www.jgroups.org

Figure 2 illustrates a Failover Manager cluster that employs a virtual IP address. You can use a load balancer in place of a virtual IP address if you provide your own fencing script to reconfigure the load balancer in the event of a failure.

The benefit of adding the automated failover capability of EDB Failover Manager is the ability to achieve fast failover times that meet the most stringent high 9's High Availability Service Level Agreements with a fully integrated tool supported by EnterpriseDB.

Choosing an HA System Design

Now that you have some background on the available HA options, it is time to consider which solution is right for you. To do this, you need to balance your needs and wants vs. cost and complexity. It is also important to note that one would use a combination of solutions in order to provide the most protection against any type of failure.

An easy to set up and robust solution would combine a standby database design using streaming replication with EDB Failover Manager providing the failover and cluster health monitoring. With the sync and async modes of streaming replication and the configurable health check timing of EDB Failover Manager, one could achieve the highest levels of High Availability at a reasonable cost if the application requires it.

An alternative approach would be to utilize active/passive clustering for the main database. This provides application High Availability in that data center and fast failover times but it doesn't provide any data redundancy or site disaster protection. Therefore, one would configure the active/passive cluster in the main data center and then use streaming replication to ship transactions to a remote site so there's a

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

copy of the database for use in the event the primary data center goes down. Adding in EDB Failover Manager to this configuration beefs up the solution for fast automatic failover to the remote data center.

One could take this a step further. Set up one of the streaming replication replicas in the same data center (but not on the same main power support or server rack) with EDB Failover Manager providing the cluster monitoring and performing automatic failover if the main system were to go down.

Conclusion

Your database needs to be online and available around the clock to serve your business, your customers and your partners. Hardware failures, network glitches, or a server crash can cost you money and opportunities. Systems need to remain accessible during planned maintenance as well. A cornerstone for ensuring ongoing data access during scheduled downtime or unexpected failures is a High Availability design. Clearly, there are multiple combinations of designs and solutions for achieving High Availability, each with their own benefits, challenges and costs. As a result, end users must consider individual needs and tolerances for outages or data loss as well as sensitivities to cost or complexity. The good news is that with EDB Failover Manager, Postgres users now have a fully integrated and configurable HA tool based on industry proven and stable technology that is fully supported by EnterpriseDB.

Get Started Today. Let EnterpriseDB help you build and execute your game plan. Contact us at +1-877-377-4352 or +1-781-357-3390, or send an email to info@enterprisedb.com to get started today on your path to database independence.

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com

About EnterpriseDB

EnterpriseDB is the leading worldwide provider of Postgres software and services that enable enterprises to reduce their reliance on costly proprietary solutions and slash their database spend by 80 percent or more. With powerful performance and security enhancements for PostgreSQL, sophisticated management tools for global deployments and Oracle compatibility, EnterpriseDB software supports both mission and non-mission critical enterprise applications. More than 2,000 enterprises, governments and other organizations worldwide use EnterpriseDB software, support, training and professional services to integrate open source software into their existing data infrastructures. Based in Bedford, MA, EnterpriseDB is backed by top-tier venture capitalists and strategic investors like Red Hat and IBM. For more information, please visit http://www.enterprisedb.com/.

20131220

© 2013 EnterpriseDB Corporation. All rights reserved. EnterpriseDB and Postgres Plus are trademarks of EnterpriseDB Corporation. Other names may be trademarks of their respective owners.
http://www.enterprisedb.com