Greenplum Database Initialization

Pivotal® **Greenplum Database**

Pivotal

Agenda

- **Initializing the Greenplum System**
- Greenplum Array Configurations
- Mirroring Options

Pivotal

Greenplum Database System Initialization

To initialize the Greenplum Database:

1. Create a host list file with all segment host names
2. Create the system configuration file, in this example, `gp_init_config`
3. Set the correct locale for the database on the master server
4. Run `gpinitsystem` on the master host
   Example: `gpinitsystem -c gp_init_config`

Pivotal

Greenplum Database Configuration File

```
ARRAY_NAME="Greenplum"
MACHINE_LIST_FILE=/home/gpadmin/gpconfigs/hostfile_gpinitsystem
SEG_PREFIX=gpseg
PORT_BASE=50000
declare -a DATA_DIRECTORY=(/data1/primary /data1/primary /data1/primary
/data2/primary /data2/primary /data2/primary)
MASTER_HOSTNAME=mdw
MASTER_DIRECTORY=/data/master
MASTER_PORT=5432
TRUSTED SHELL=ssh
CHECK_POINT_SEGMENT=8
ENCODING=UNICODE
#Option Entries for segment mirrors
MIRROR_PORT_BASE=50000
REPLICATION_PORT_BASE=41000
MIRROR_REPLICATION_PORT_BASE=51000
declare -a MIRROR_DATA_DIRECTORY=(/data1/mirror
/data1/mirror /data1/mirror /data2/mirror /data2/mirror /data2/mirror)
```

sdw1-1
sdw1-2
sdw1-3
sdw1-4
sdw2-1
sdw2-2
sdw2-3
sdw2-4

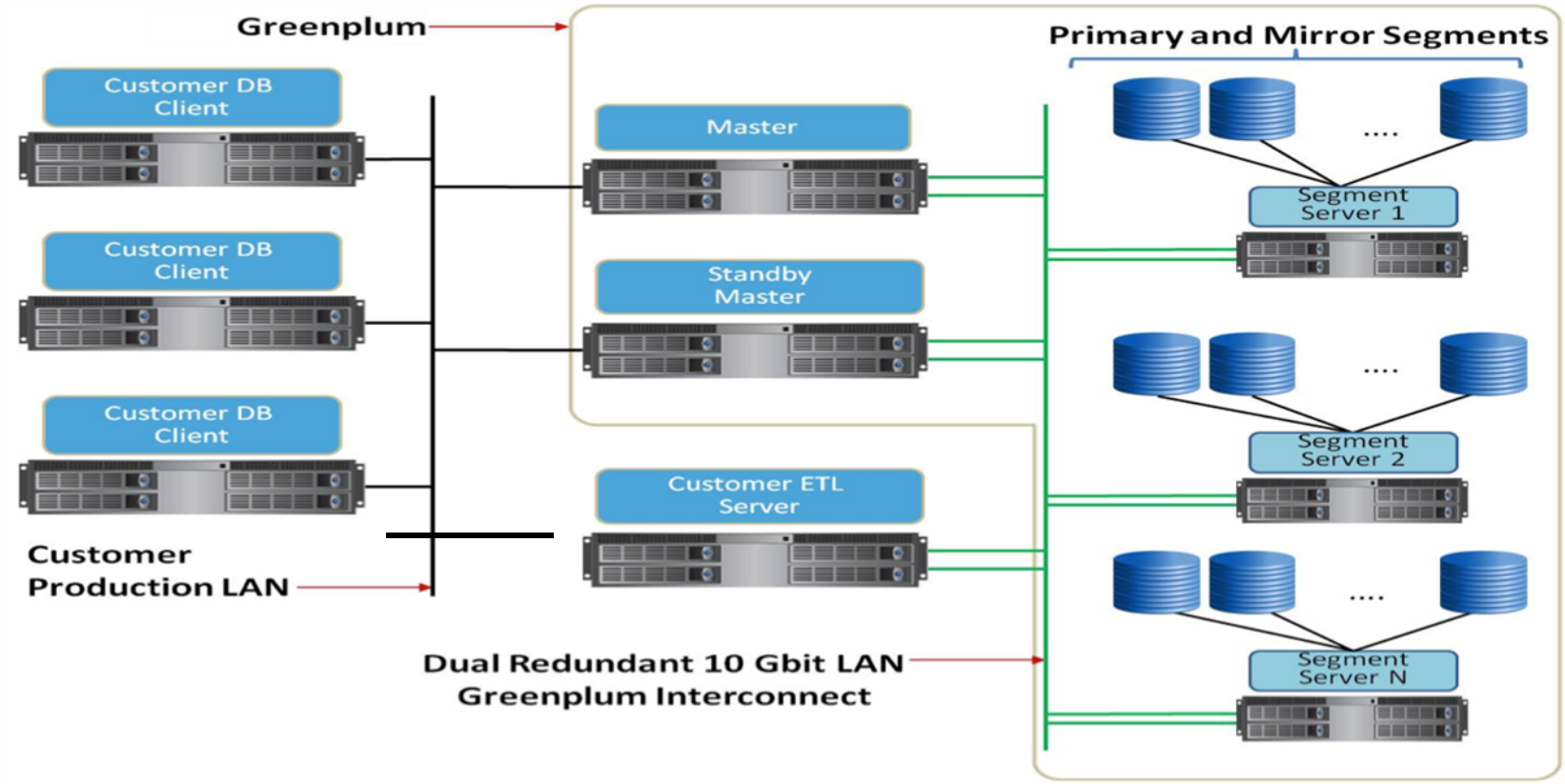
Pivotal

Agenda

- Initializing the Greenplum System
- **Greenplum Array Configurations**
- Mirroring Options

Pivotal

Production Greenplum Cluster



Pivotal

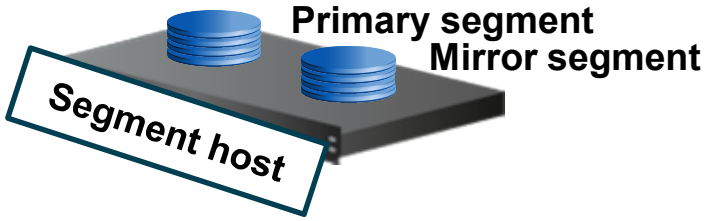
Greenplum Segment Host Allocation

Primary Segments are:

- Can be either Master or segment instances

Mirror segments are:

- A replica of a given primary segment
- Used for data redundancy



Mirroring can be enabled:

- At array initialization time
- On an active Greenplum Database system:

Greenplum DB init sequence

To see the state of a Greenplum system (after it has been initialized) you can run a simple SELECT command

SELECT * from gp_segment_configuration;

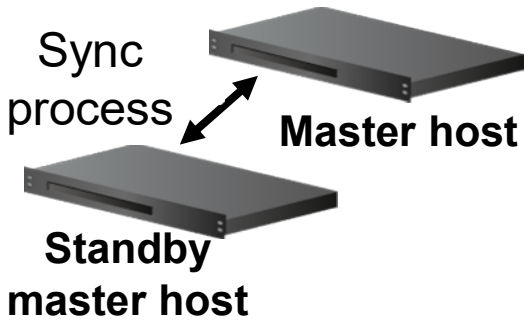
This shows a lot of information, but the **mode** column is valuable:

- S means it is in sync – no need to do anything out of the ordinary
- C mean change tracking mode – this means that the system has noticed that there is a Primary without a Mirror
- R means resync mode – means that the system is attempting to resync the mirror by playing back the change log

Creating Mirrors for High Availability and Redundancy

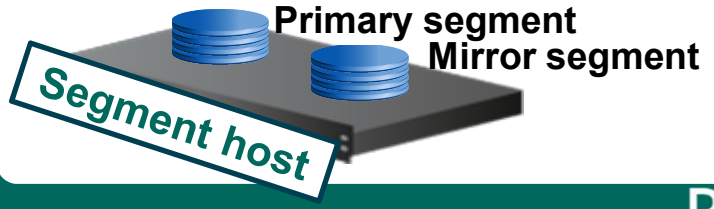
Master mirroring:

- Lets you create a warm standby master
- Starts the synchronization process between hosts



Segment mirroring:

- Creates a mirror segment for a primary segment
- Requires enough nodes to spread mirroring
- Can be configured on same array of hosts

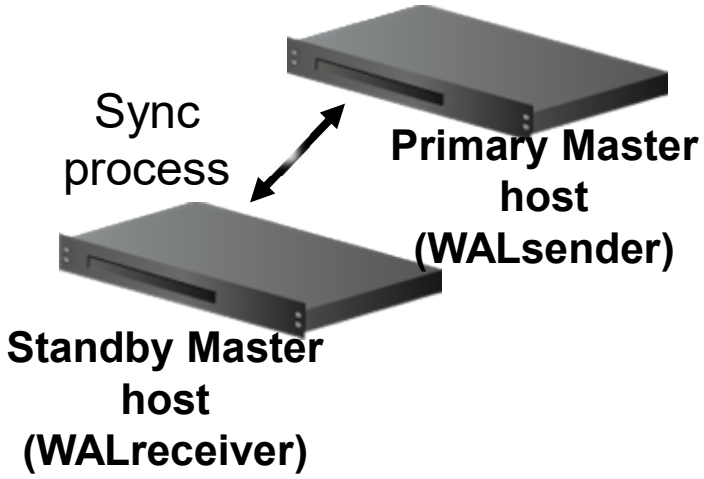


Pivotal

Warm Standby Master

A warm standby master is:

- A replica of the Greenplum master instance (system catalogs)
- Used to remove single point of failure
- Kept up to date by the *WALsender* (Primary Master) and *WALreceiver* (Standby Master) replication processes
- System Catalogs and Transaction Logs are replicated.



WAL = Write Ahead Log

Pivotal

Adding a Standby Master

Add a standby master to an existing Greenplum system:

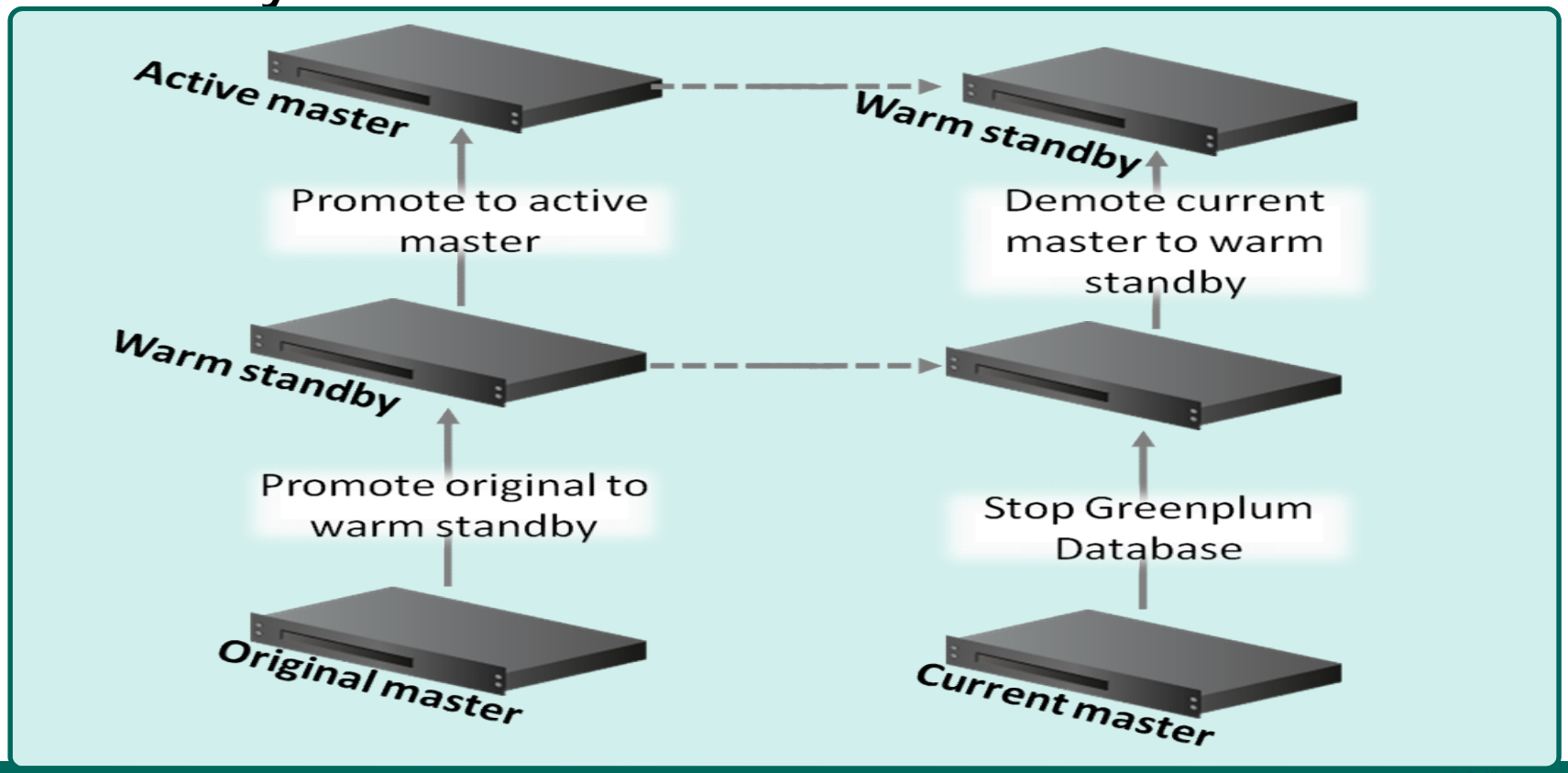
1. Verify Greenplum binaries were installed
2. Source `/usr/local/greenplum-db/greenplum_path.sh`
3. Exchange keys
4. Create data directories
5. Initialize the database with the following:
   `gpinitstandby -s` *`standby_hostname`*

Add a standby master during initialization:

1. Verify Greenplum binaries were installed
2. Source `/usr/local/greenplum-db/greenplum_path.sh`
3. Create data directories
4. Initialize Greenplum and specify the warm standby master with the following:
   `gpinitsystem -c gp_init_config -s` *`standby_hostname`*

Pivotal

Primary Master Failure and Restoration

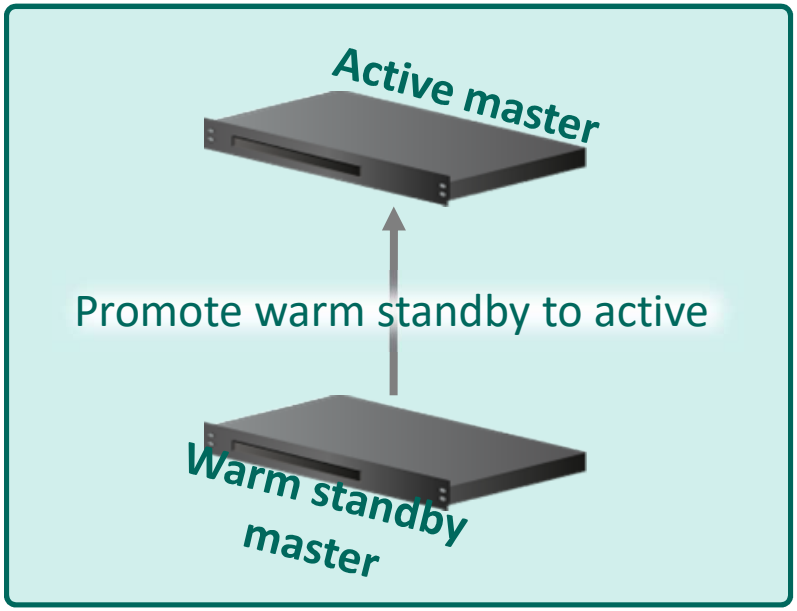


Pivotal

Promoting a Warm Standby to an Active Primary Master

Promote a standby master to primary:

1. On the standby master, run:
   `gpactivatestandby -d $MASTER_DATA_DIRECTORY`
2. Optionally, configure a new standby that has already been configured:
   `gpactivatestandby -d $MASTER_DATA_DIRECTORY -c` *`new_standby_hostname`*
3. Verify the active master is `Active` and the standby is `Passive`, if configured:
   `gpstate -f`

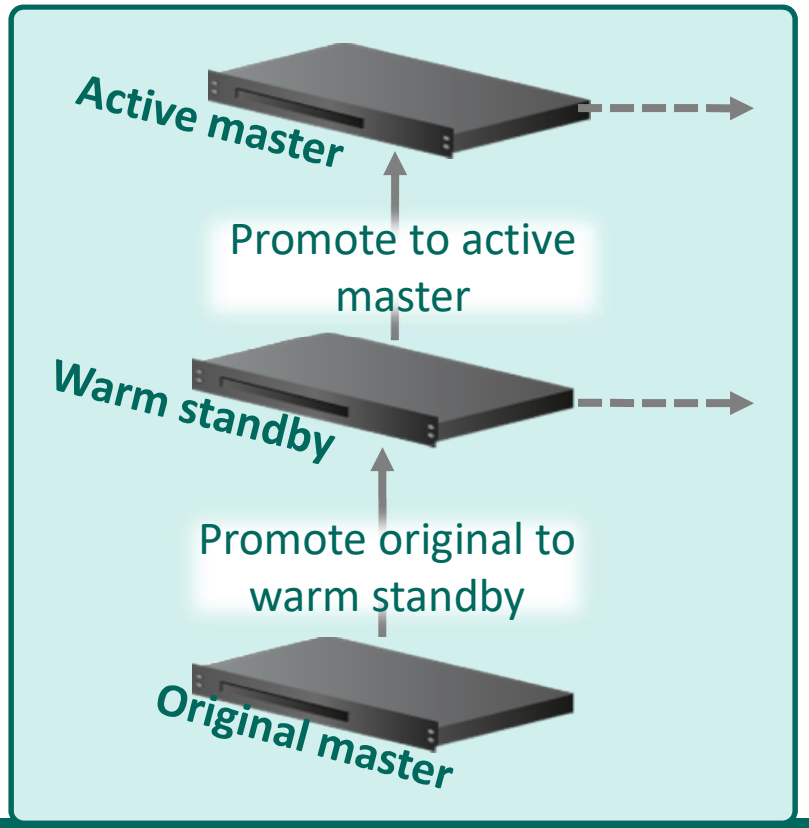


Pivotal

Promoting the Original Master to the Active Master

Promote the original failed master back to a master:

1. Fix problems on the original standby and verify the Greenplum Database processes have not started.
2. Initialize the original master as a standby master:
   `gpinitstandby -s` *`original_master_hostname`*
3. Stop the Greenplum Database on the current master:
   `gpstop -m`

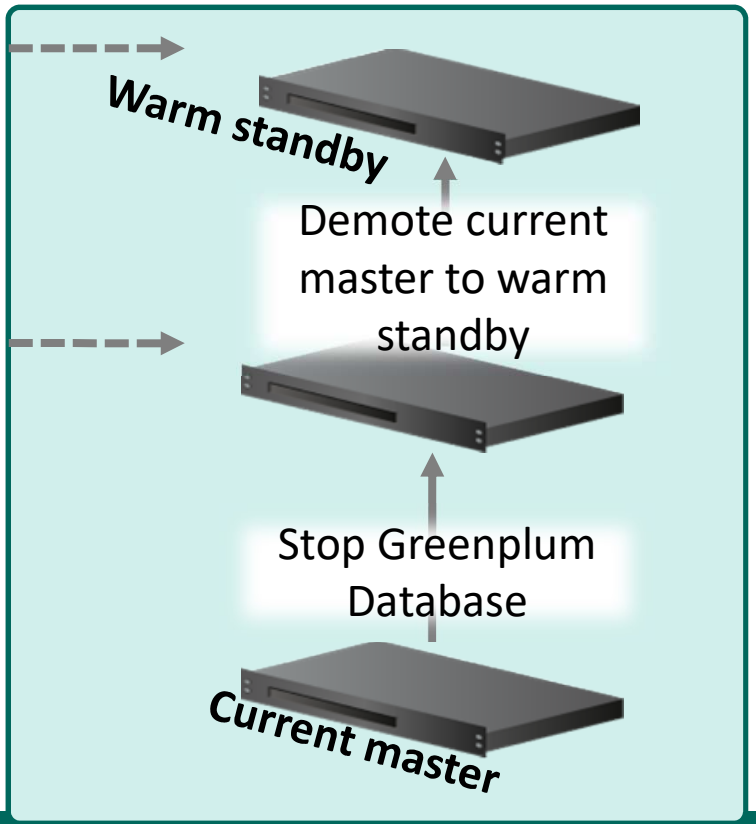


Pivotal

Promoting the Original Master to the Active Master (Cont)

Promote the original failed master back to a master:

4. Promote the original master from a standby to the master:
   ```
   gpactivatestandby -d $MASTER_DATA_DIRECTORY
   ```
5. Reinitialize the original standby master to be the standby master again:
   ```
   gpinitstandby -s original_standby_master_hostname
   ```
6. Check the state of the master and standby master:
   ```
   gpstate -f
   ```



Pivotal

Agenda

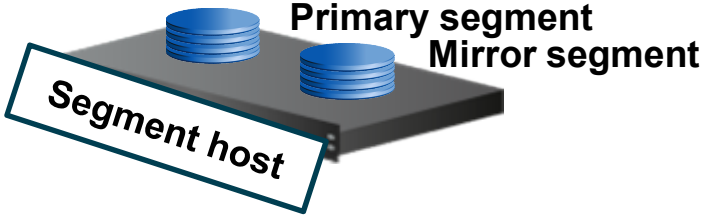
- Initializing the Greenplum System
- Greenplum Array Configurations
- **Mirroring Options**

Pivotal

Mirror Segments

Mirror segments are:

- A replica of a given primary segment
- Used for data redundancy



Mirroring can be enabled:

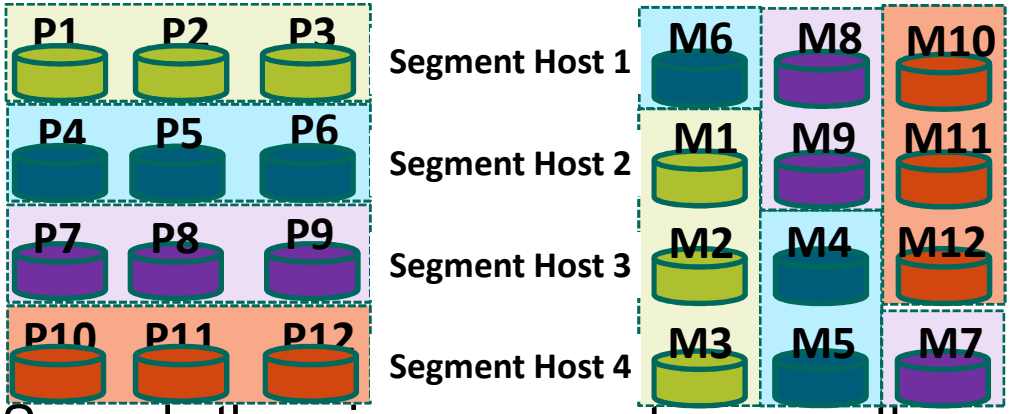
- At array initialization time
  - (set parameters in `gp_init_config file`)
- On an active Greenplum Database system:
  - `gpaddmirrors command`

Mirror segments can be deployed:

- On the same hosts as your primary segments (Not recommeneded)
- On a different set of host that your primary segments (Spread Mirroring)

Pivotal

Mirroring in Greenplum – Spread Mirror Distribution



4 Segment Hosts
3 Primary Segment Instances

- Spreads the mirror segments across the available hosts
- Mirror spreading will place each mirror on a different host within the Greenplum Database array
- Spreading is only allowed if there is a sufficient number of hosts in the array
- Number of hosts is greater than the number of segment instances

Pivotal

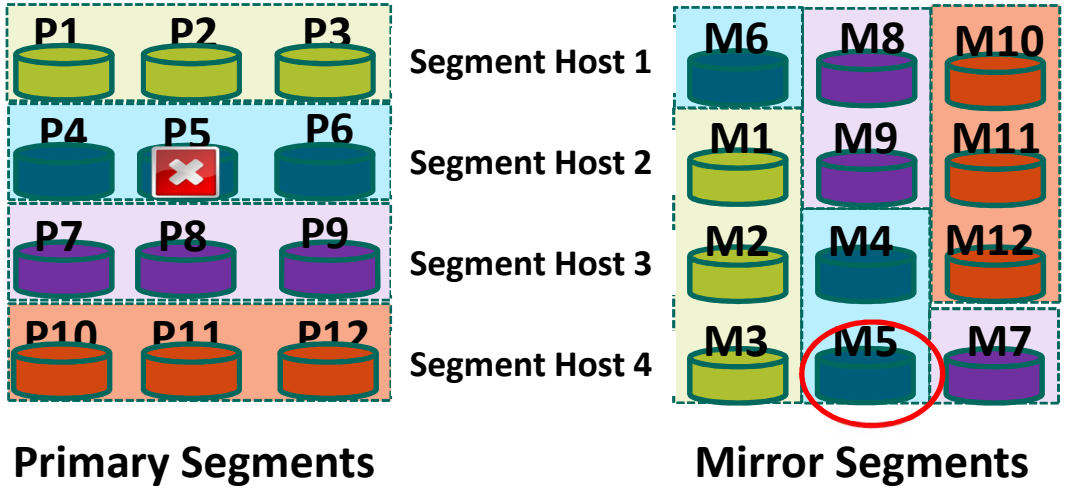
Fault Detection and Recovery

Fault detection:

- Is handled by `ftsprobe`
- Marks a segment as down when a connection fails or a response timeout is exceeded.
- Allows subsequent connection requests to switch to the mirror and succeed
- Requires Greenplum administrators to manually recover a segment marked invalid with `gprecoverseg`
- May also require that the Greenplum administrator has to manually rebalance the database cluster with a `gprecoverseg -r`
- Requires vigilance from Greenplum administrators

Pivotal

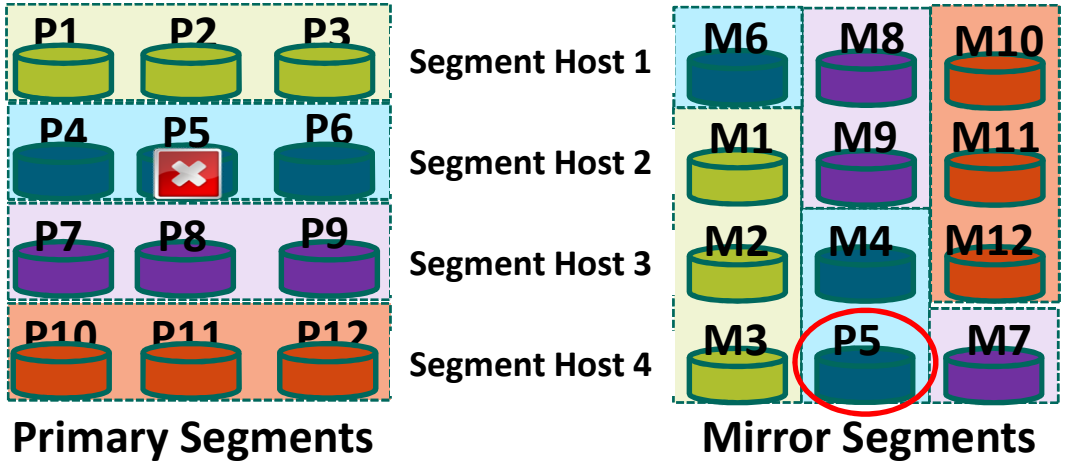
Mirroring in Greenplum – Failed Primary Segment Example



Primary Segment Failure:

1. The `ftsprobe` process on the master detects the segment down and marks it invalid.

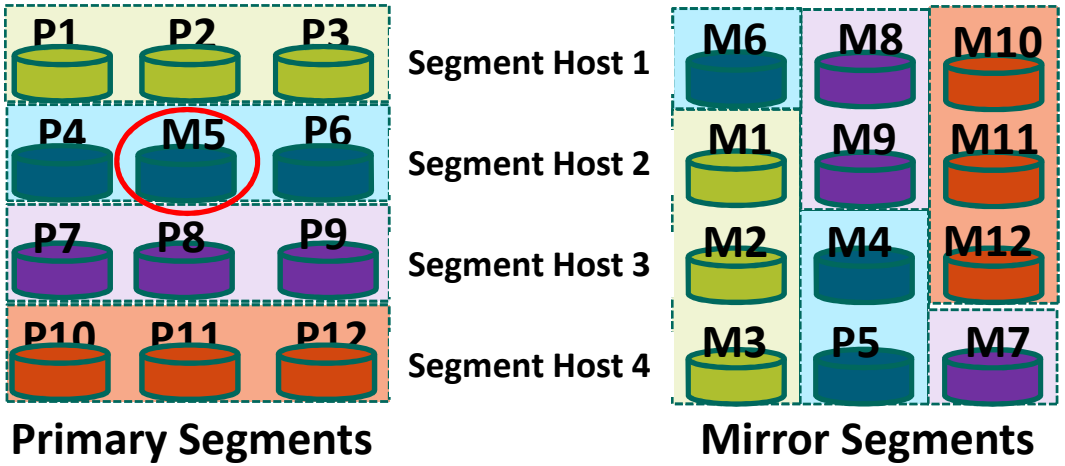
Mirroring in Greenplum – Failed Primary Segment Example



Primary Segment Failure:

2. The mirror segment is validated to ensure that it was synchronized with its primary segment. Mirror segment becomes the primary.

Mirroring in Greenplum – Failed Primary Segment Example

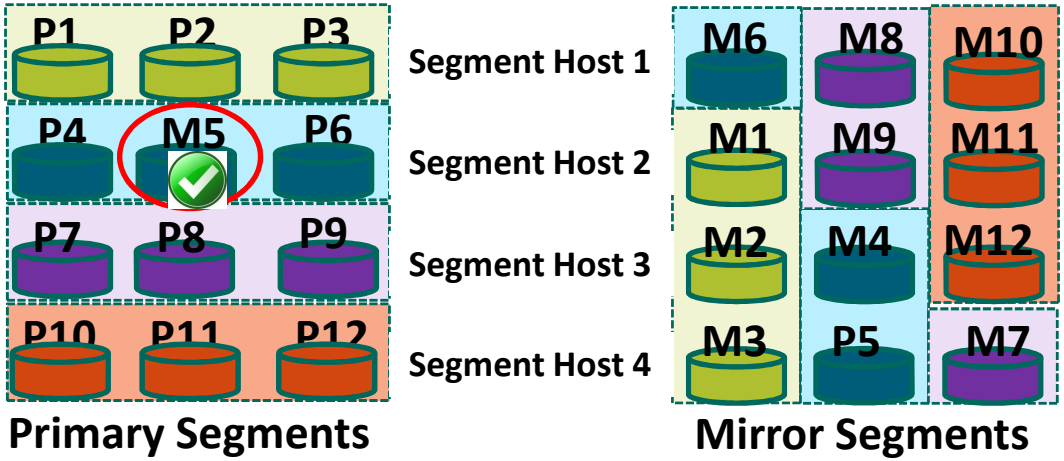


Primary Segment Failure:

3. Once it has been determined why the original primary P5 went down and is repaired, you can then bring that segment back online to become the mirror segment to protect the new primary P5.

Pivotal

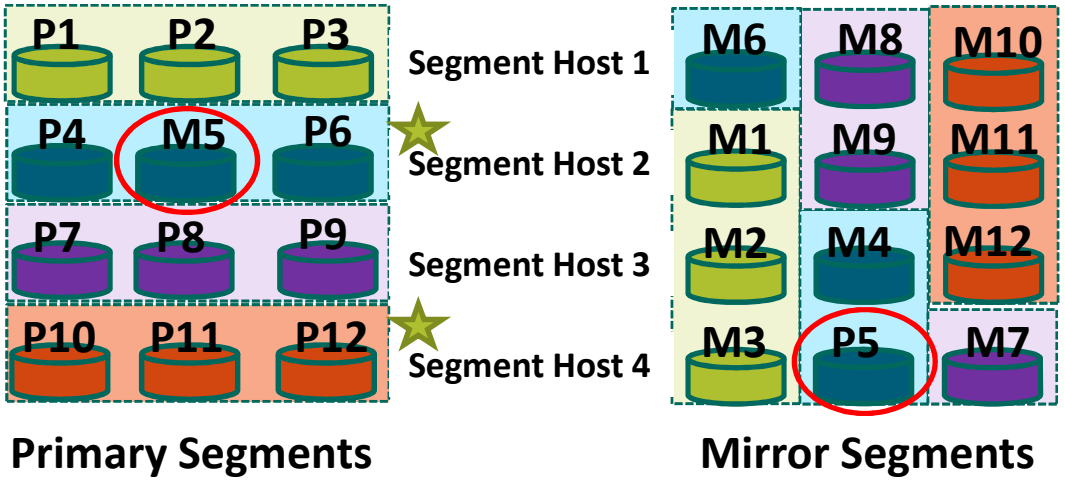
Mirroring in Greenplum – Failed Primary Segment Example



Primary Segment Failure:

4. Run the gprecoverseg command to bring back up the segment that went down. The old primary P5 becomes the M5 mirror.

Mirroring in Greenplum – Re-balancing Segments

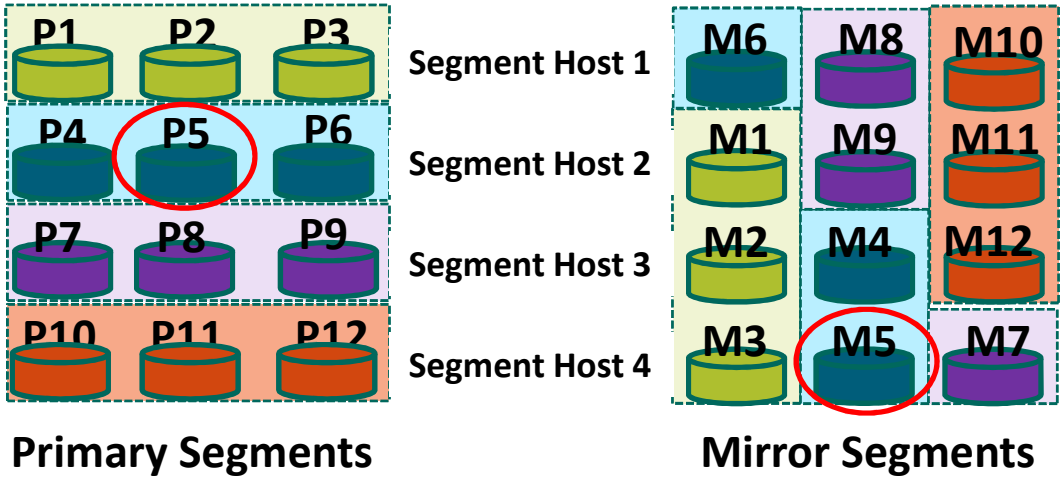


The database cluster is in an un-balanced condition

- Segment Host 2: Two Primaries, 4 Mirrors
- Segment Host 4: 4 Primaries, 2 Mirrors

Pivotal

Mirroring in Greenplum – Re-balancing Segments



Run the following command to re-balance the cluster

- `gprecoverseg -r`

The database cluster would then be back in a balanced condition

- Segment Host 2: 3 Primaries, 3 Mirrors
- Segment Host 4: 3 Primaries, 3 Mirrors

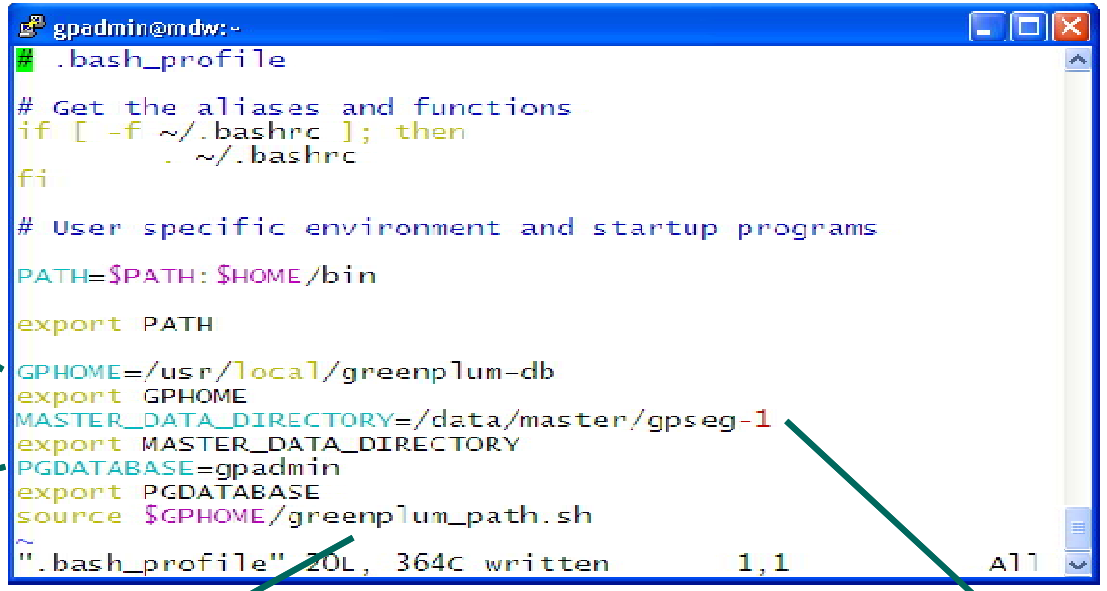
Pivotal

Setting Greenplum Environment Variables

GPHOME points to the base Greenplum directory (executables and libraries)

PGDATABASE sets up your default database to connect to

Always source the *greenplum_path.sh* file (sets up paths to executables and libraries)



MASTER_DATA_DIRECTORY is the location of the data directory on the master server

Pivotal

Wrap Up

During this lesson, the following topics were covered

- Initializing the database
- Array configurations
- Mirroring options

Pivotal


Pivotal
A NEW PLATFORM FOR A NEW ERA