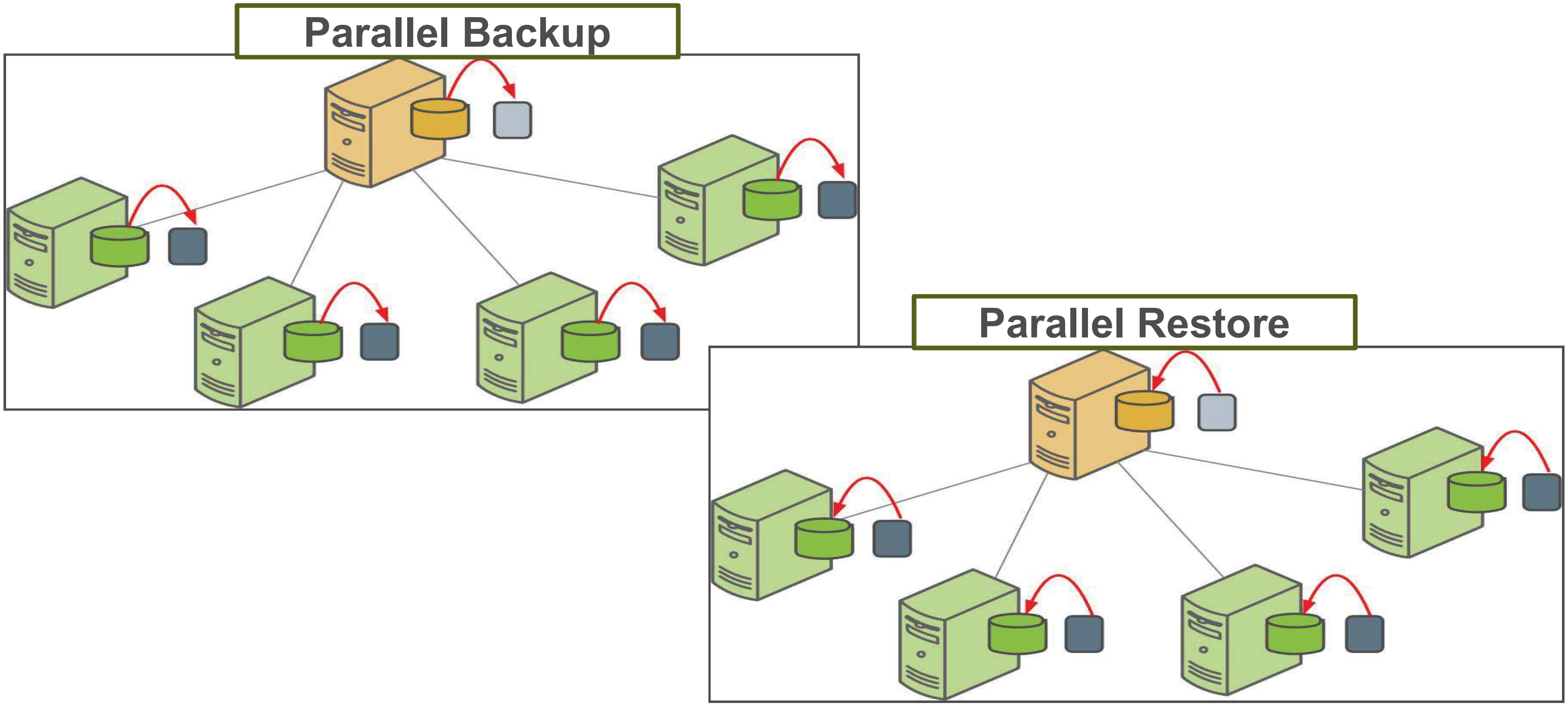
# Performing Backup and Restore

Pivotal® Greenplum Database

# Agenda

- **Performing Backups and Restores**
- Incremental backups
- Defining the strategy for backups

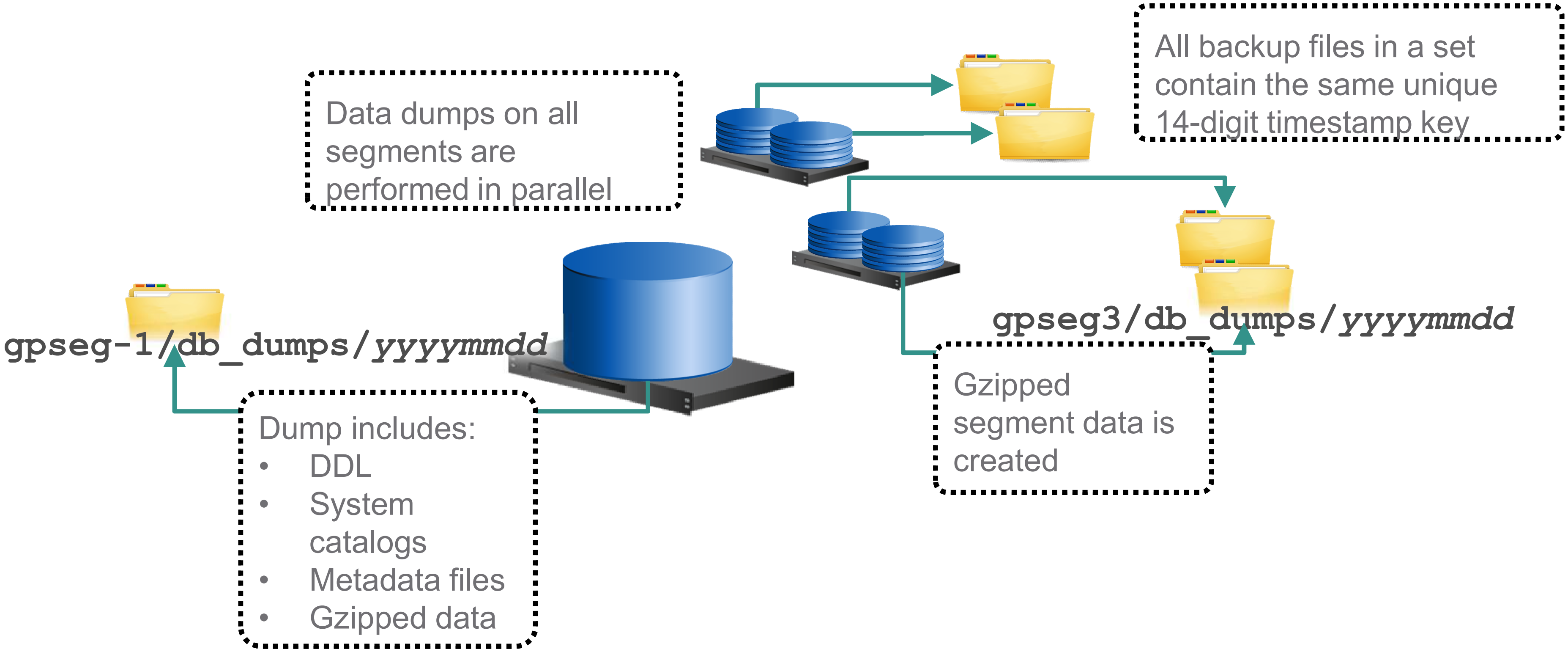
# About Parallel Backups and Restores



Parallel Backup

Parallel Restore

# Creating Parallel Backups



Data dumps on all segments are performed in parallel

All backup files in a set contain the same unique 14-digit timestamp key

`gpseg-1/db_dumps/`*`yyyymmdd`*

`gpseg3/db_dumps/`*`yyyymmdd`*

Dump includes:

- DDL
- System catalogs
- Metadata files
- Gzipped data

Gzipped segment data is created

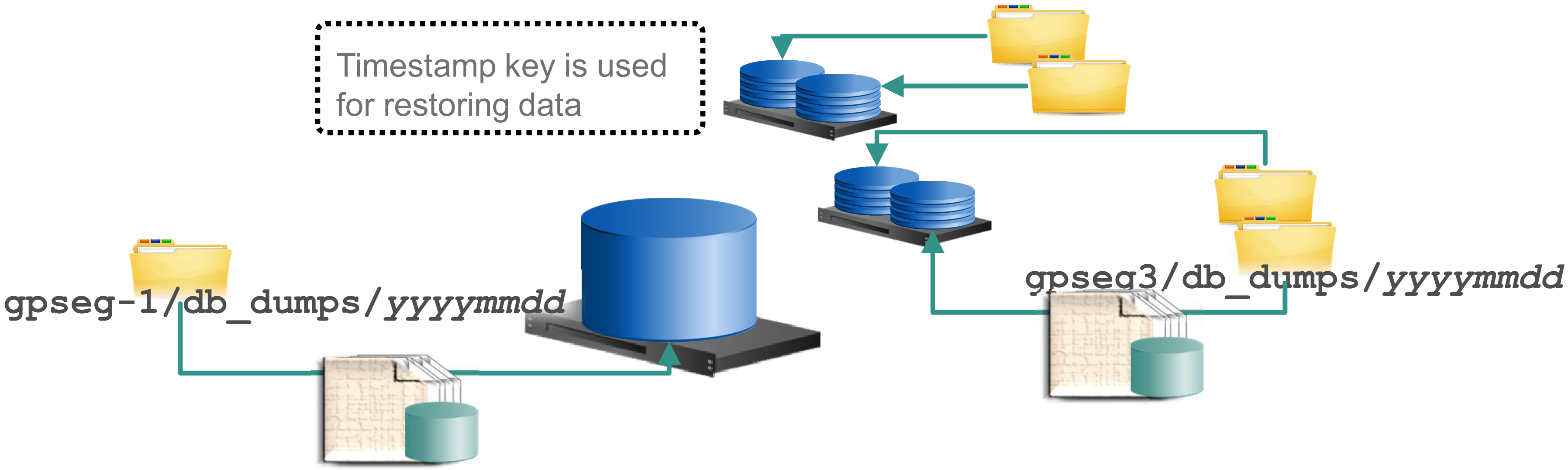
# Dump Files Created During Parallel Backup

| Master Segment Dump File | Description |
|---|---|
| gp_cdatabase_1_<dbid>_<timestamp> | CREATE DATABASE statement |
| gp_dump_1_<dbid>_<timestamp> | Database schemas |
| gp_dump_status_1_<dbid>_<timestamp> | Log file |
| gp_dump_1_<dbid>_<timestamp>_post_data | Post database setup |
| gp_dump_1_<dbid>_<timestamp>_ao_state_file | List of append-optimized tables |
| gp_dump_1_<dbid>_<timestamp>_co_state_file | List of column-oriented tables |
| gp_dump_<timestamp>.rpt | Database dump report |

| Primary Segments Dump File | Description |
|---|---|
| gp_dump_0_<dbid>_<timestamp> | Data for the segment |
| gp_dump_status_0_<dbid>_<timestamp> | Log file |

**Note:** Each backup set shares the same unique timestamp.
This timestamp is required for restoring a backup set.

# Performing Parallel Restores



Timestamp key is used for restoring data

gpseg-1/db_dumps/*yyyymmdd*

gpseg3/db_dumps/*yyyymmdd*

# Scheduling Routine Backups – `gpcrondump`

The `gpcrondump` utility:

- Can be called directly or can schedule using `cron`
- Should be scheduled on the master host
- Sends email notifications
- Flexible dump options
- Can copy configuration files
- Can dump system catalogs
- Can dump global objects
- Can include a post-dump script

# Restoring Archived Data

The `gpdbrestore` utility:

- Restores `gpcrondump` files
- Reconfigures for compression
- Validates the number of dump files
- Restores to active segment instances even with a failed segment
- Does not require you to retrieve the timestamp key
- Can restore from an archive host
- Does not require dump files to be placed on segments
- Identifies the database name automatically
- Detects the type of backup set available

# Agenda

- Performing Backups and Restores
- **Incremental backups**
- Defining the strategy for backups

# Incremental Backups

Incremental backups:

- Were Introduced in the 4.2.5 release and above
- Allow users to specify a point in time to restore the database to
- Are supported with:
  - Column- and row-oriented append-only tables
  - At the partition level of AO tables
- Back up an AO table if one of the following operations is performed:
  - ALTER TABLE, INSERT, TRUNCATE, DELETE, UPDATE
  - DROP and then re-create the table
- Cannot be used with Data Domain Boost

# Managing Incremental Backups

- To create an incremental backup with `gpcrondump`, include the `--incremental` option
- To restore data from an incremental backup you need a complete backup set, which consists of the following:
  - The last full backup before the current incremental backup
  - All incremental backups created between the time of the full backup the current incremental backup
  - The full backup and incremental backups need to be in the same directory location (the `gpcrondump -u` option will ensure this)

# Incremental Backup Example

**Example: Creating a full backup of the `faa` database to `/backupdir`**

```
$ gpcrondump -x faa -u /backupdir
```

**Example: Creating a series of incremental backups to `/backupdir`**

```
$ gpcrondump -x faa -u /backupdir --incremental
```

# Incremental Backup Example (Cont)

- Full and incremental backups are saved in the user specified directory and named with an appropriate timestamps. After a series of backups you might see something like this in the backup directory:
  - 20120514054532 (full backup)
  - 20120714095512
  - 20120914081205
  - 20121114064330 (full backup)
  - **20130114051246**
- Restore a backup by specifying a point in time that corresponds to an existing incremental backup:

```
gpdbrestore -t 20130114051246 -u /backupdir
```

- The result of this command will be to restore the database using the last full backup (20121114064330) and the last incremental backup (20130114051246)

# Agenda

- Performing Backups and Restores
- Incremental backups
- **Defining the strategy for backups**

# Non-Parallel Backups and Restores

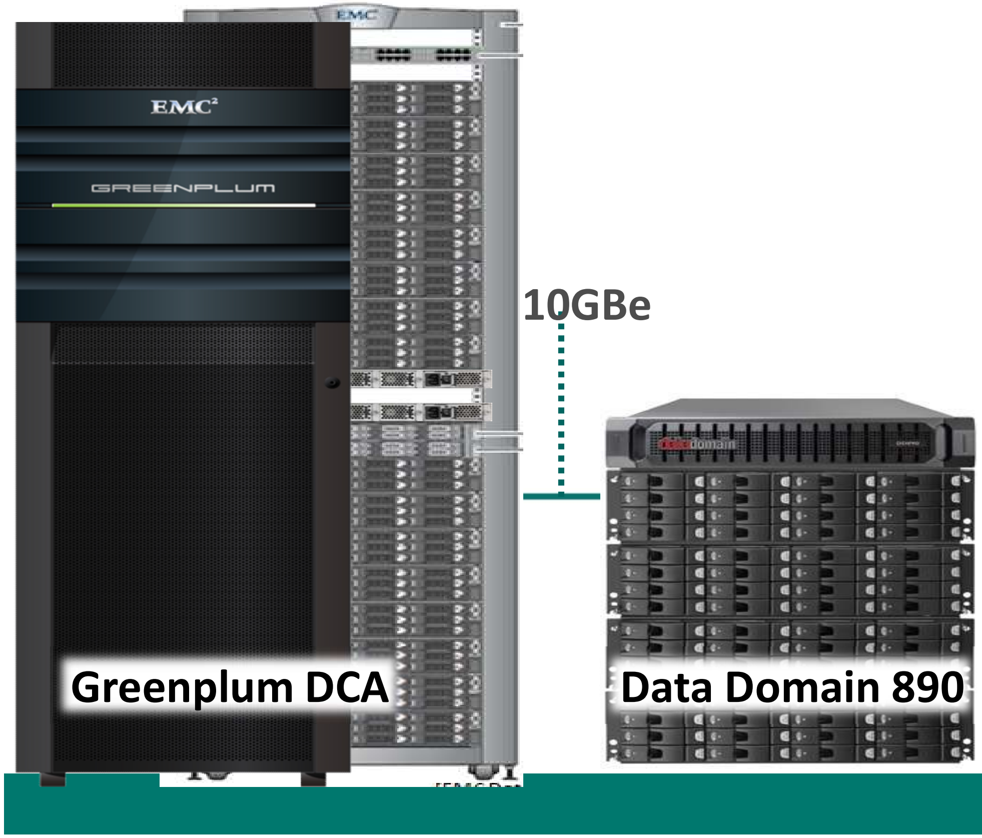
Non-parallel backups and restores:

- Are supported with the `pg_dump` and `pg_restore` utilities
- Are useful for migrating data to and from other DBMS

The `pg_dump` utility:

- Creates a single dump file
- Can be slow on very large databases
- Should be run at low-usage times
- Supports compression
- Can dump data as `INSERT` or `COPY` commands
- Includes the `DISTRIBUTED BY` statements in DDL with the `––gp–`syntax option

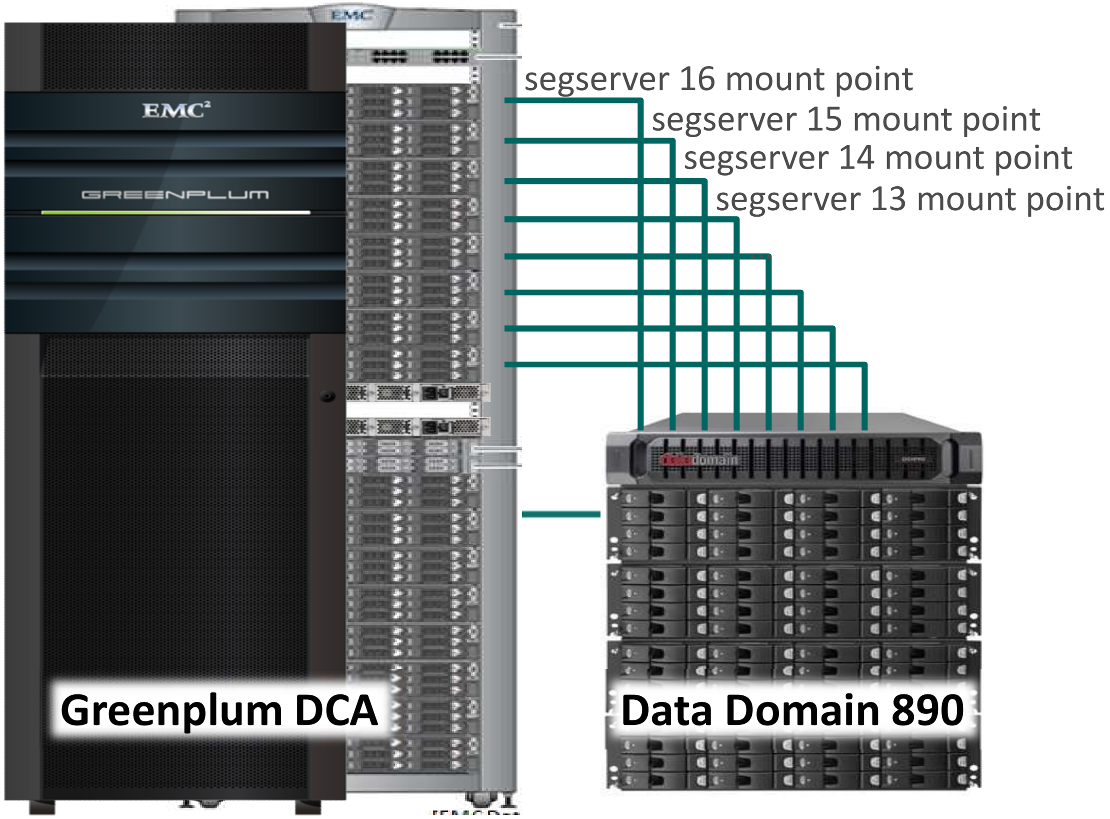
# EMC Greenplum DCA and the Data Domain Solution



Data Domain:

- Provides backup and recovery with Greenplum DB 4.1+
- Offers deduplication
- Supports:
  - NFS mounts with GPDB 4.1
  - DDBoost with GPDB 4.2
- Leverages `gpcrondump` and `gpdbrestore`
- Must be connected on the interconnect
- Provides access to each Greenplum Database instance

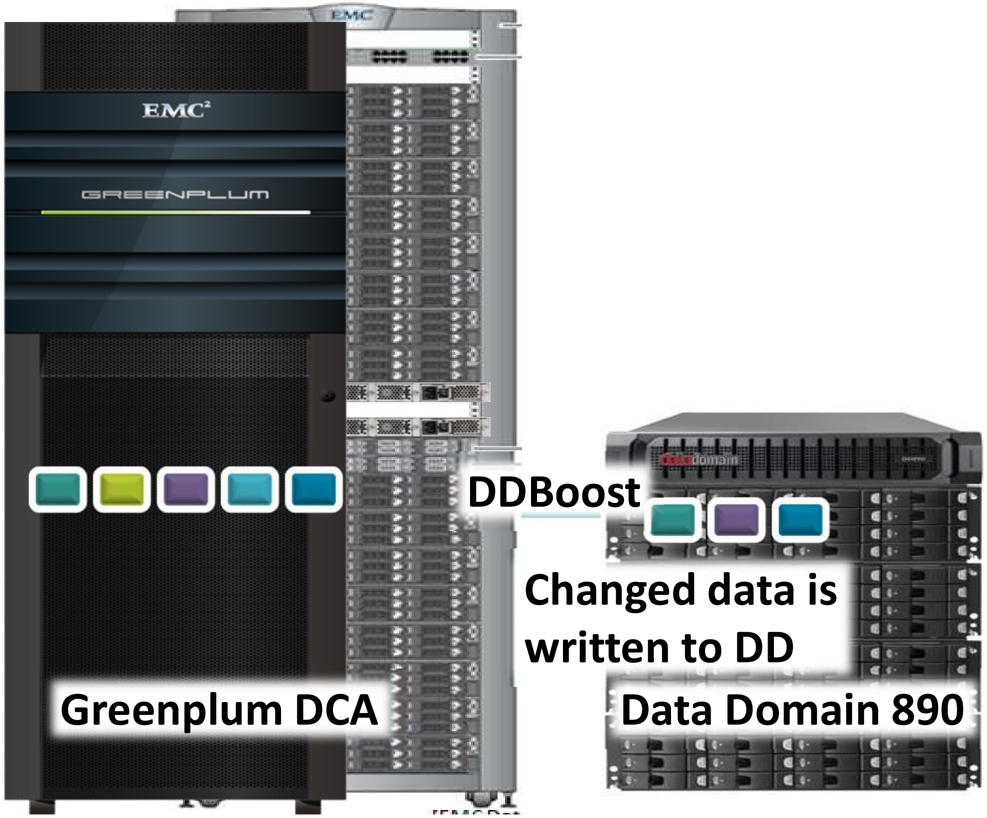
# Data Domain Integration: NFS Solution



NFS integration:

- Is available to GPDB 4.1 and 4.2
- Requires each server has its own mount point
- Performs deduplication and compression after data is sent over the network

# Data Domain Integration: DD Boost Solution

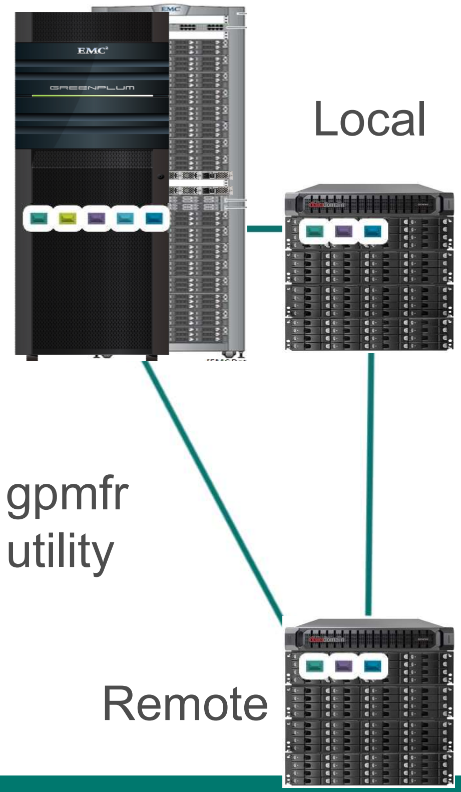


DD Boost integration:

- Is a client library integrated with GPDB
- Uses native communication protocol
- Performs deduplication on the segments and master
- Only captures changed data
- Takes advantage of MPP design

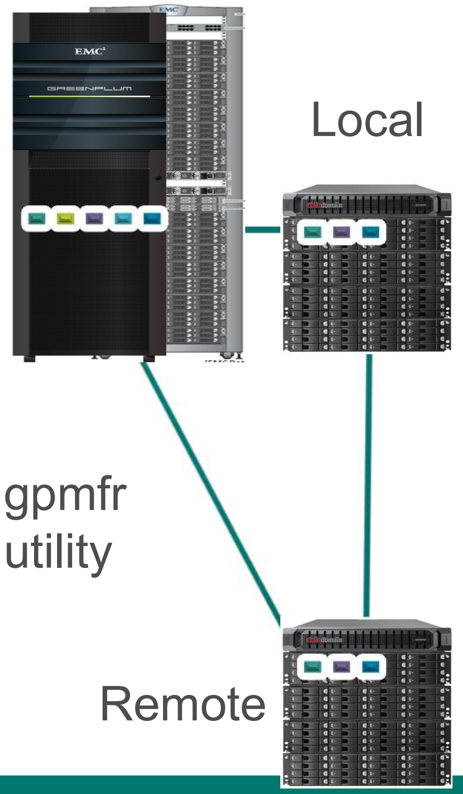
Pivotal

# Data Domain Integration: Managed File Replication



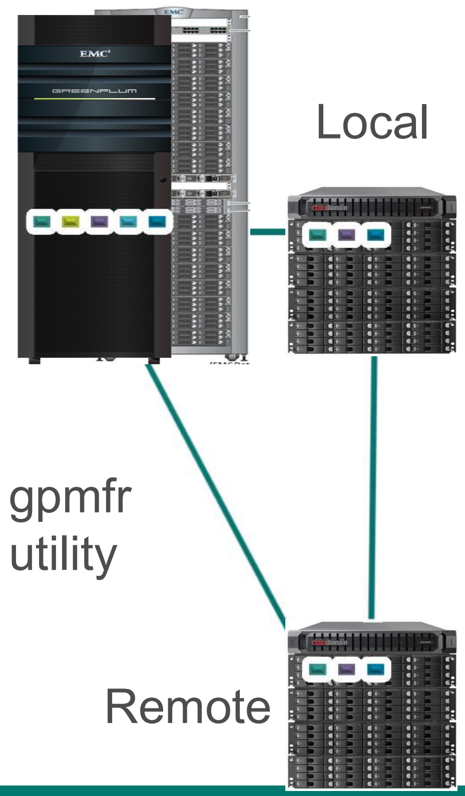
- Managed File Replication (MFR)
  - Introduced with the 4.2.5 release of GP
  - Allows replication of Greenplum Database backup images stored on a local Data Domain system to a remote Data Domain system.
  - Data Domain login credentials have to be configured with `gpcrondump` utility on both the local and remote Data Domain systems.
  - The master segment must be able to connect to both the local Data Domain system and the remote Data Domain system.

# Data Domain Integration: Managed File Replication



- The gpmfr utility manages the Greenplum Database backup sets on the local and remote Data Domain systems.
- The gpmfr utility provides these capabilities:
  - Lists the backup data sets that are on the local or the remote Data Domain system.
  - Replicates a backup data set that is on the local Data Domain system to the remote system.
  - Recovers a backup data set that is on the remote Data Domain system to the local system.
  - Deletes a backup data set that is on the local or the remote Data Domain system.

# Data Domain Integration: MFR Example



The following example replicates the latest backup set on the local Data Domain sever to the remote server. The maximum number of I/O streams that can be used for the replication is 30.

```
gpmfr --replicate
LATEST --max-streams
30
```

# Comparing the EMC Data Domain Integration Solutions for Greenplum DCA

| Feature | NFS | DD Boost |
|---|---|---|
| Deduplication | Deduplication on Data Domain appliance | Deduplication on Greenplum DB segment and master instance |
| CPU Usage on Segments | As needed for NFS | Increased CPU usage on GPDB due to de-duplication and compression |
| Network Utilization | All data is sent over the network | Only changed, deduplication data is sent over the network |
| Scalability | Increasing the number of racks can result in saturation of DD appliance or network | Minimal data transfer |
| Management | Each segment server and master server requires its own mount point | Integrated native solution with no static system configuration |
| Backup Performance | Full backup | Initial backup is full; follow-on backups are incremental |
| Data Domain Replication | Directory level | Collection and Managed File |

# Wrapping Up

In this module we covered:

- The process of parallel backup
- The process of parallel restore
- The process of non-parallel backup
- The commands used to perform backup and restoration of data
- Command options to perform incremental archival