Managing the Greenplum Database

Pivotal® Greenplum Database

Pivotal™

Agenda

- **System Administration Tasks Overview**
- Starting and stopping the services
- Verifying status
- Logs
- Assessing data skew and data storage
- Troubleshooting

System Administration Tasks – Overview

The following are routine administrative tasks:

- Starting and stopping Greenplum
- Obtaining the state of Greenplum
- Checking for data skew
- Accessing log files and parameters
- Maintaining the system catalog and reclaiming disk space
- Recover Down Segments

Agenda

- System Administration Tasks Overview
- **Starting and stopping the services**
- Verifying status
- Logs
- Assessing data skew and data storage
- Troubleshooting

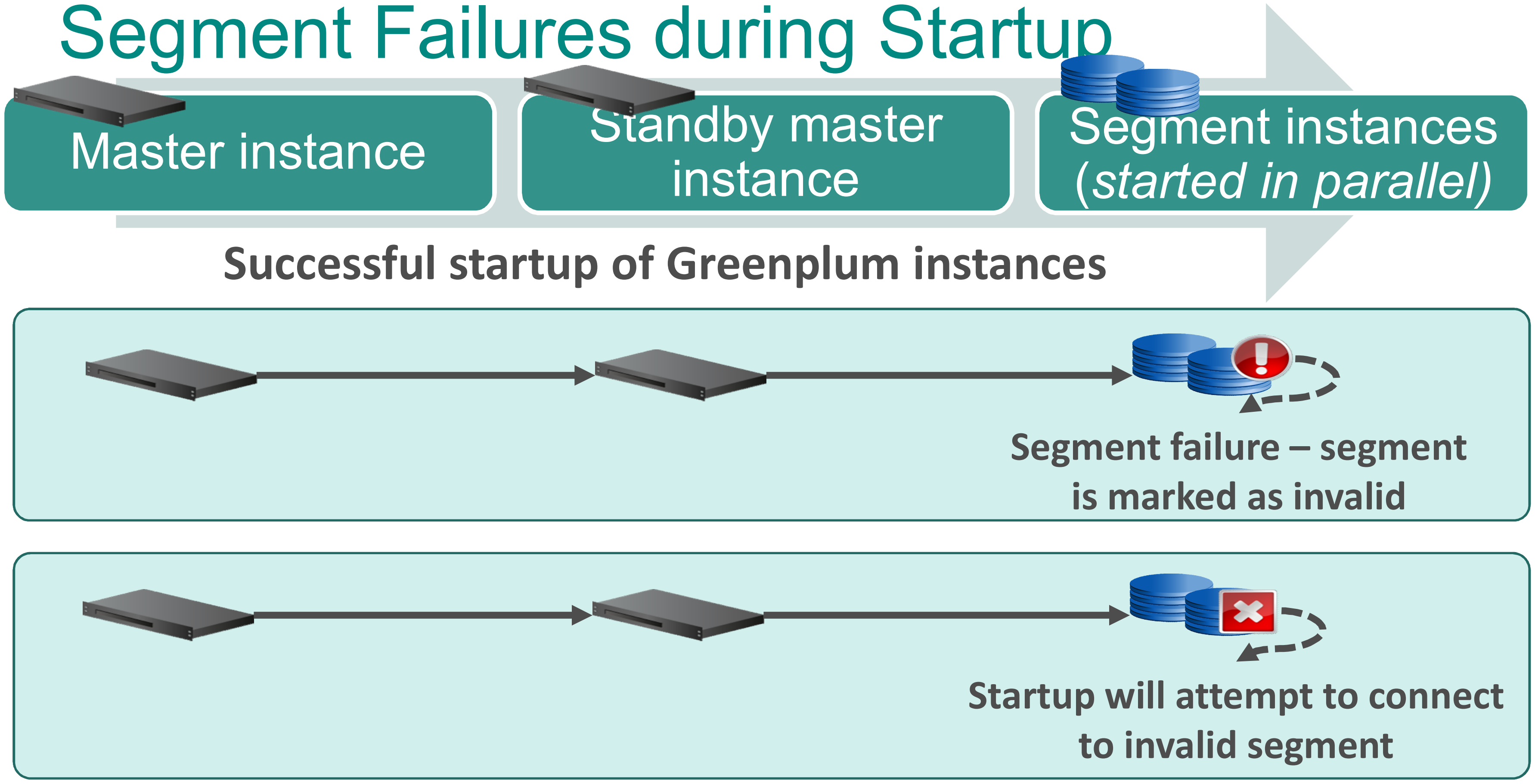
Pivotal

Starting and Stopping Greenplum

The following commands start and stop Greenplum:

Action	Greenplum Application
Start the Greenplum Database	`gpstart`
Stop the Greenplum Database	`gpstop`
Restart the Greenplum Database	`gpstop -r`
Start the Greenplum Database in restricted mode	`gpstart -R`
Reload master postgresql.conf and pg_hba.conf	`gpstop -u`
Start the master in utility mode	`gpstart -m`
Stop the master that was started in utility mode	`gpstop -m`

Segment Failures during Startup


Master instance
Standby master instance
Segment instances (*started in parallel*)
Successful startup of Greenplum instances
Segment failure – segment is marked as invalid
Startup will attempt to connect to invalid segment

Agenda

- System Administration Tasks Overview
- Starting and stopping the services
- **Verifying status**
- Logs
- Assessing data skew and data storage
- Troubleshooting

Checking System State

The following commands are used to check the Greenplum state:

Action	Greenplum Application
Check system status	`gpstate`
Show complete system configuration and status	`gpstate -s`
Ports used by the system	`gpstate -p`
Segment mirror configuration	`gpstate -m`
Primary to mirror mapping	`gpstate -c`
Show details on primary/mirror segment pairs that have potential issues	`gpstate -e`
Obtain Greenplum Database version information	`gpstate -i`

Recovering Down Segments

Recover Greenplum Database primary and mirror segments with the following:

Action	Greenplum Application
Recovers a primary or mirror segment instance that has been marked as down.	`gprecoverseg`
Rebalances primary and mirror segments by returning them to their preferred roles.	`gprecoverseg -r`
Performs a full copy of the active segment instance in order to recover the failed segment. The default is to only copy over the incremental changes that occurred while the segment was down.	`gprecoverseg -F`

Pivotal

Agenda

- System Administration Tasks Overview
- Starting and stopping the services
- Verifying status
- **Logs**
- Assessing data skew and data storage
- Troubleshooting

Log Files

Greenplum Database server log files:

- Are located in the data directory
- Are stored daily as CSV files
- Are stored on the master in `$MASTER_DATA_DIRECTORY/pg_log`
- Are stored on each segment in `/`*`segment_datadir`*`/gpseg#/pg_log`
- For management scripts, log files are located in `/`*`superuser_home`*`/gpAdminLogs`
- Can be searched and filtered with the `gplogfilter` command

Logging Configuration Parameters

The following are commonly used for log configuration:

Log Parameter	Description
`client_min_messages`	Controls which message levels are sent to the client; accepts range from `DEBUG5` to `PANIC`
`log_min_messages`	Controls which message levels are written to the server log; accepts range from `DEBUG5` to `PANIC`
`log_connections`	Each successful connection is logged; accepts `on|off`
`log_statement`	Controls which SQL statements are logged; accepts `NONE`, `DDL`, `MOD`, or `ALL`
`log_rotation_age`	Determines the maximum lifetime of an individual log file; accepts values in minutes
`log_rotation_size`	Determines the maximum size of an individual log file; accepts values in kilobytes
`log_duration`	Causes the duration of every completed statement to be logged; useful for query profiling

Pivotal™

Agenda

- System Administration Tasks Overview
- Starting and stopping the services
- Verifying status
- Logs
- **Assessing data skew and data storage**
- Troubleshooting

Checking for Data Distribution Skew

Check for data skew with the following:

- `gp_toolkit` administrative schema offers two views:
  - `gp_toolkit.gp_skew_coefficients`
  - `gp_toolkit.gp_skew_idle_fractions`
- To view the number of rows on each segment, run the

```
SELECT gp_segment_id, count(*)
FROM table_name GROUP BY gp_segment_id;
```

- Check for processing skew with the following query:

```
SELECT gp_segment_id, count(*) FROM table_name
WHERE column='value' GROUP BY gp_segment_id;
```

Agenda

- System Administration Tasks Overview
- Starting and stopping the services
- Verifying status
- Logs
- Assessing data skew and data storage
- **Troubleshooting**

Greenplum Server Log Troubleshooting

Use the following tips to troubleshoot server problems:

- Check the master log to find the relevant log entry
  - Log lines have the format of:
    ```
    timestamp | user | database | statement_id |
    con# cmd# |:-MESSAGE _TYPE: <log_message>
    ```
  - The following is a sample of a log entry:
    ```
    2006-08-19 19:00:58 PDT|lab1|names|11085|con107
    cmd1|:-LOG:  statement: select * from topten
    where year='2005' and gender='F' order by rank;
    ```
- Search the segment logs `gpssh` and `gplogfilter`

```
gpssh -f seg_host_file
=> source /usr/local/greenplum-db/greenplum_path.sh
=> gplogfilter -n 3 /gpdata/*/pg_log/gpdb*.log
```

Maintaining the System Catalog and Reclaiming Disk Space

- Growth in the system catalog size:
- Can be caused by numerous database updates with `CREATE` and `DROP` commands
- Can be controlled with the `VACUUM` command or `vacuumdb` Greenplum application for regular system maintenance
- Can be controlled with `VACUUM FULL` for intensive system maintenance

Script for Reclaiming Disk Space

The following script can be used for periodic maintenance:

```
#!/bin/bash
DBNAME="<database_name>"
VCOMMAND="VACUUM"
psql -tc "select '$VCOMMAND' || ' pg_catalog.' ||
relname || ';' from pg_class a,pg_namespace b where
a.relnamespace=b.oid and b.nspname= 'pg_catalog'
and a.relkind='r'" $DBNAME | psql -a $DBNAME
```

Checking Database Object Sizes and Disk Space

`gp_toolkit` schema views for disk usage (in bytes):

Action	gp_toolkit View
Total size of all indexes for a table	`gp_size_of_all_table_indexes`
Size of a database	`gp_size_of_database`
Total size of an index	`gp_size_of_index`
Size of schemas in this database	`gp_size_of_schema_disk`
Disk size of a table	`gp_size_of_table_disk`
Uncompressed table size for append-only tables	`gp_size_of_table_uncompressed`
Amount of disk free, as determined by the OS `df` command	`gp_disk_free`

Note: Objects are shown by their object IDs. Look up the relation name in the `pg_class` table.

Pivotal™

Detecting Bloated Tables and Tables with Missing Statistics

The following views provide information on bloated tables and tables missing statistics:

Action	gp_toolkit View
List tables that have bloat	`gp_bloat_diag`
List tables that do not have statistics	`gp_stats_missing`

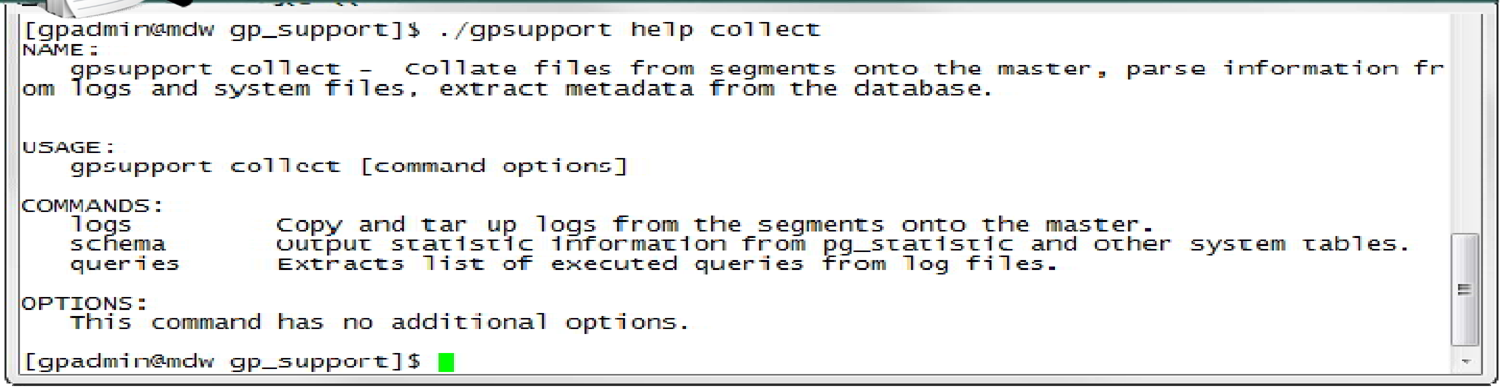
Pivotal™

Data Gathering for Troubleshooting with `gpsupport`

System log information:	Process listing
	Free memory
	Hosts file
	RPM packages
	`sysctl.conf`
Executed queries	Obtained from parsed log files
Metadata	Schemas
	Statistics
	Configuration information

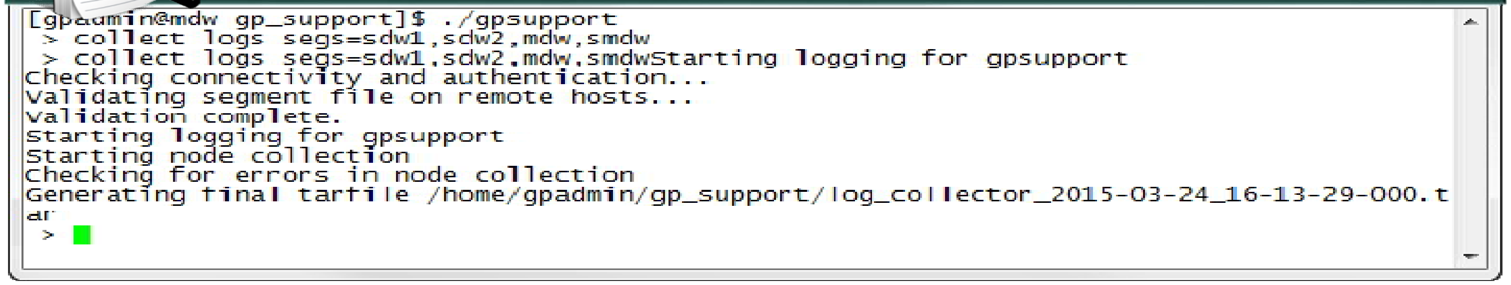
`gpsupport` Collection Examples

Example: Obtain help on the collect command



```
[gpadmin@mdw gp_support]$ ./gpsupport help collect
NAME:
   gpsupport collect -  collate files from segments onto the master, parse information fr
om logs and system files, extract metadata from the database.


USAGE:
   gpsupport collect [command options]

COMMANDS:
   logs           Copy and tar up logs from the segments onto the master.
   schema         Output statistic information from pg_statistic and other system tables.
   queries        Extracts list of executed queries from log files.

OPTIONS:
   This command has no additional options.

[gpadmin@mdw gp_support]$
```

Example: Collect logs from several segments



```
[gpadmin@mdw gp_support]$ ./gpsupport
 > collect logs segs=sdw1,sdw2,mdw,smdw
 > collect logs segs=sdw1,sdw2,mdw,smdwStarting logging for gpsupport
Checking connectivity and authentication...
Validating segment file on remote hosts...
Validation complete.
Starting logging for gpsupport
Starting node collection
Checking for errors in node collection
Generating final tarfile /home/gpadmin/gp_support/log_collector_2015-03-24_16-13-29-000.t
ar
 >
```

Wrapping Up

In this module we covered:

- Starting and stopping the Greenplum Database
- Verifying the state of the Greenplum Database
- Using the Greenplum administrative schema to view which tables are exhibiting data skew
- Accessing log files and logging parameters
- Maintaining the system catalog and reclaim physical disk space
- Using `gpsupport` to gather troubleshooting data (future `gpmt`)