DeepGreen HA/FT Solution

Feng Tian (ftian@vitessedata.com)

April 20, 2017

1 Introduction

DeepGreen introduced a new suite of HA/FT solution with various settings,

1. A synchronous replation solution using Linux DRBD (dg cluster)
2. An asynchronous repliation solution using rsync (dg sync)
3. Data transfer between two deepgreen Cclusters via XDrive (dg transfer)

All three solutions are designed to be robust, efficient and reliable. We tried to make administration simple, so that it is easy to use and more importantly, reduce chance of operation error. We suggest all new installation of DeepGreen to use these new solutions and the old greenplum utilities like gpmirror, gpcrondump, gptransfer, while still supported, are considered deprecated.

2 Synchronous Replication Using dg cluster

The key to HA is multipath to data. By far the most common hardware failure is disk failure, which, a typical DeepGreen installation may use RAID to mask. However, RAID5 won't tolerate double disk failure and other components of server hardware like NIC, Memory/Motherboad/CPU, Power Supply, etc, do fail. DeepGreen provides a HA solution that will automatically failover a host/segment to another node.

2.1 Prerequisite

Before install/initialize the cluster, user must set up each node in the cluster properly.

1. Operating system must be configured properly for DeepGreen. Users should setup kernel limits properly, SELinux should be turned off and firewall should be turned off.
2. Network must be properly configured. Especially, hostname should be setup correctly.

3. ssh key has been exchanged. DeepGreen software must be installed properly on master, master mirror and each segment hosts. DeepGreen user should be able to sudo without password.

4. Create volume group for data storage. See pvcreate, vgcreate manual.

5. Optionally, create a volume group for temp storage.

6. Install DRBD 8.4. Set usage_count to no.

2.2 cluster.toml

To use dg cluster, user need to create a directory on the master node. The directory should contain a cluster.toml file, which describes the cluster configuration. The following is a documented cluster.toml file that will set up a cannonical installtion.

```
# cluster.toml
# DGHome is the deepgreen software installation dir.
DGHome = "/home/deepgreen/deepgreendb"

# User, usually, it is the user running dg cluster
User = "deepgreen"

# Master host and mirror.
MasterHost = "xx0"
MasterMirror = "xx1"

# Segment hosts. Must have at leas two hosts. Master/mirror can run on
# some of the hosts list here.
SegHosts = ["xx0", "xx1"]

# Segments per host. This config has 2 hosts, each has 2 segs.
Therefore, 4 segs total.
SegsPerHost = 2

# Data install dir.
Dir = "/data/dgdata"

# Segment prefix, so, master will be installed in a dir dg-1, segments are in
SegPrefix = "dg"

# Data base encoding, collation, etc. We recommend utf8 and "C" collation.
Encoding = "UNICODE"
Collation = "C"

# Configuration of master, and seg. We will use volume group /dev/dvdg.
In this example,
```

```
# master/mirror has 2GB capacity, each seg has 2GB.  Note that they are repli
# each node, there will need
# 2GB (master) + 2GB (master mirror) + 2GB x 2 (Seg) + 2GB x 2 (Seg mirror) =
# Also, it need temp storage during query execution.  We set it to 1GB.
We do not replicate
# temp storage, so, each node need 1GB (master) + 1GB x 2 = 3GB.
#
# So /dev/dvdg need 15GB to install deepgreen.
[Master]
Port = 5432
ReplicationPort = 10000
DataVG = "/dev/dgvg"
DataSizeGB = 2
TempVG = "/dev/dgvg"
TempSizeGB = 1
[Seg]
PortBase = 40000
ReplicationPortBase = 20000
DataVG = "/dev/dgvg"
DataSizeGB = 2
TempVG = "/dev/dgvg"
TempSizeGB = 1
```

2.3 Installation

Suppose the cluster.toml file is in ./install directory,

```
dg cluster -dir=./install Install_Stage1
```

Stage1 will create necessary directory, configuration file, etc, for gpinitsystem in ./install directory. Run the shell file to init DeepGreen cluster. If init system failed, one should examine gpinitsystem error logs to fix. After successfully inited system, run

```
dg cluster -dir=./install Install_Stage2
```

to finish all replication setup.

2.4 Administion

Cluster created by dg cluster should be started/stoped using dg cluster command.

```
dg cluster -dir=./install Start
dg cluster -dir=./install Stop
```

2.5 Failback

Cluster created b dg cluster will automatically failover if a hosts failed. Once the failed node is repaired, user should failback the node. Failback is manual. For example, to fail back segment 2 and 3, run

```
dg cluster -dir=./install FailbackSegs '2,3'
```

3 Asynchronous Replication Using dg sync

dg sync is a utility that copies a DeepGreen cluster to remote site. In new DeepGreen cluster, we recommend using dg sync to backup, restore, over older gpcrondump tool. dg sync uses rsync to copy data, rsync must be installed in all hosts of the source cluster and sync destination cluster. For RHEL/CentOs user, must also intall net-tools.

3.1 sync.toml

To use dg sync, user need to create a directory on the master node of the sync target cluster. The directory should contain a sync.toml file. The following is an example. sync.toml describes the source and target cluster. The content of the file should be self explainatory except that user need to specifiy the mount point of linux file system where master and segment data directory reside. Pretty much the only requirement of the toml file is that the number of segments of src and tgt cluster mush match.

```
[Src]
DgHome = "/data/deepgreen/work/deepgreen/run/dghome"
[Src.Master]
Host = "dg0"
Port = 5432
DataDir = "/data/mnt/dg/dg-1"
MountPt = "/data/mnt"
[[Src.Segs]]
Host = "dg0"
DataDir = "/data/mnt/dg/dg0"
MountPt = "/data/mnt"
[[Src.Segs]]
Host = "dg0"
DataDir = "/data/mnt/dg/dg1"
MountPt = "/data/mnt"
[[Src.Segs]]
Host = "dg1"
DataDir = "/data/mnt/dg/dg2"
MountPt = "/data/mnt"
[[Src.Segs]]
```

```
Host = "dg1"
DataDir = "/data/mnt/dg/dg3"
MountPt = "/data/mnt"
[Tgt]
DgHome = "/data/deepgreen/work/deepgreen/run/dghome"
PortBase = 40000
[Tgt.Master]
Host = "dg2"
Port = 5432
DataDir = "/data/mnt/dg/dg-1"
MountPt = "/data/mnt"
[[Tgt.Segs]]
Host = "dg2"
DataDir = "/data/mnt/dg/dg0"
MountPt = "/data/mnt"
[[Tgt.Segs]]
Host = "dg2"
DataDir = "/data/mnt/dg/dg1"
MountPt = "/data/mnt"
[[Tgt.Segs]]
Host = "dg3"
DataDir = "/data/mnt/dg/dg2"
MountPt = "/data/mnt"
[[Tgt.Segs]]
Host = "dg3"
DataDir = "/data/mnt/dg/dg3"
MountPt = "/data/mnt"
```

3.2 dg sync -dir=./dir [-full]

If -full flag is specified, dg sync will do a full copy. Otherwise, it will do incremental copy.

4 Copy Database/Schema/Table using dg transfer

We recommend using dg transfer to copy data between two DeepGreen clusters, over older gptransfer tools. See dg command manual for details.