

Exercise 10 – Gossip

In this exercise, you will:

- Understand how Apache Cassandra™ uses gossip.
- Understand how gossip information propagates through a cluster.
- Understand how a gossip exchange works.

In a fully distributed system such as Apache Cassandra™, there is no single repository that contains the state of all the nodes in the cluster. Clearly, such a repository would be a single point of failure. Instead, Apache Cassandra™ uses the Gossip protocol to distribute nodes' status amongst its peers.

In this exercise, we will examine the gossip information for our two-node cluster.

Steps

- 1) Be sure both nodes are up and running using `dsetool status`. Start your nodes if necessary.
- 2) Execute the following command:

```
/home/ubuntu/node1/resources/cassandra/bin/nodetool gossipinfo
```

Note the gossip information for both nodes as discussed in the slides.

```
ubuntu@ds201-node1:~$ /home/ubuntu/node1/resources/cassandra/bin/nodetool gossipinfo
WARN 19:05:01,684 Could not determine sector size for xvdb, assuming 512
/127.0.0.1
  generation:1523905164
  heartbeat:412
  STATUS:32:NORMAL, -116815900718359043
  LOAD:374:92637.0
  SCHEMA:75:172ca292-9d0a-37fb-8f32-70c87ba0891e
  DC:44:Cassandra
  RACK:18:rack1
  RELEASE_VERSION:4:4.0.0.2151
  NATIVE_TRANSPORT_ADDRESS:3:127.0.0.1

X_11_PADDING:394:{"dse_version":"6.0.0","workloads":"Cassandra","workload":"Cassandra",
"active":"true","server_id":"06-A4-68-36-AA-A2","graph":false,"health":0.1}
  NET_VERSION:1:256
  HOST_ID:2:9e2954c0-29ce-450f-9983-b0584d3e28fc
  NATIVE_TRANSPORT_READY:78:true
  NATIVE_TRANSPORT_PORT:6:9041
```

```

NATIVE_TRANSPORT_PORT_SSL:7:9041
STORAGE_PORT:8:7000
STORAGE_PORT_SSL:9:7001
JMX_PORT:10:7199
TOKENS:31:<hidden>
/127.0.0.2
  generation:1523905226
  heartbeat:325
  STATUS:55:NORMAL, -1186054891257956901
  LOAD:291:165183.0
  SCHEMA:35:172ca292-9d0a-37fb-8f32-70c87ba0891e
  DC:63:Cassandra
  RACK:18:rack1
  RELEASE_VERSION:4:4.0.0.2151
  NATIVE_TRANSPORT_ADDRESS:3:127.0.0.1

X_11_PADDING:271:{"dse_version":"6.0.0", "workloads":"Cassandra", "workload": "Cassandra",
"active":"true", "server_id":"06-A4-68-36-AA-A2", "graph":false, "health":0.1}
  NET_VERSION:1:256
  HOST_ID:2:1d611123-2cd6-44ae-b37f-b32198134e5d
  NATIVE_TRANSPORT_READY:77:true
  NATIVE_TRANSPORT_PORT:6:9042
  NATIVE_TRANSPORT_PORT_SSL:7:9042
  STORAGE_PORT:8:7000
  STORAGE_PORT_SSL:9:7001
  JMX_PORT:10:7299
  TOKENS:54:<hidden>

```

Notice that even though we executed this command on node1, node1 still knows node2's gossip state. Also notice the node state consists of key-values pairs as discussed in the slides.

- 3) Rerun your `nodetool gossipinfo` command a few times and notice the heartbeat values increasing for both nodes.
- 4) Run `nodetool gossipinfo` on node2 instead of node1. Notice the gossip data is the same.
- 5) Terminate your second node by executing:

```
/home/ubuntu/node2/resources/cassandra/bin/nodetool stopdaemon
```

- 6) Check the `gossipinfo` on node1. Notice node2's gossip information is still present as it is part of the cluster, but its STATUS state is shutdown.
- 7) Restart your second node.
- 8) Rerun `nodetool gossipinfo` on either node, and notice the generation values are the same as before for node1, but are now different for node2 since we restarted node2.

- 9) Start `cqlsh` and execute the following query of the `system.peers` table which stores some gossip data about a node's peers.

```
SELECT peer, data_center, host_id, preferred_ip, rack,  
release_version, rpc_address, schema_version  
FROM system.peers;
```

Notice the values here are some of the same values you saw in the terminal. Also notice that a node does not store a row of peer data for itself. By default, `cqlsh` connects to `127.0.0.1`.