ILLINOIS DATA SCIENCE INITIATIVE

TECHNICAL REPORT

Introduction to Cassandra

Author:
Bhuvan Venkatesh

March 1, 2017

Introduction to Cassandra

BHUVAN VENKATESH[1] AND ROBERT BRUNNER[2]

[1] *National Center For Supercomputing Applications (NCSA)*
[2] *Laboratory for Computation, Data, and Machine Learning*

Compiled March 1, 2017

https://github.com/lcdm-uiuc

INTRODUCTION

Apache Cassandra is an Open Source distributed database. The characteristics of Cassandra is that it is a high availability and fault tolerant database. The use cases for Cassandra are when you are familiar with the typical SQL databases and you need a way of modeling

INSTALLATION

For those of you on docker to run all you need to do is

```
$ docker pull cassandra
$ docker run cassandra
```

If not, the non-dockerized installation is below.

Prerequisites

1. Java 1.6 or Above
2. Python 2.7 or Above

First you will need to set a new user for cassandra, letting cassandra run under normal users is not recommended, so we create a new user

```
$ useradd cassandra # Name for the user
```

Navigate to *http://cassandra.apache.org/download/* and select the most recent stable version of cassandra (Usually the most recent or the second most recent version) and execute

```
$ wget <url>
$ tar zxvf <file>.tar.gz
$ mv <folder> /opt/cassandra # May need root
$ export CASSANDRA_HOME=/opt/cassandra
$ chown -R cassandra $CASSANDRA_HOME
```

After this, create the directories that cassandra keeps its data in.

```
$ mkdir /var/lib/cassandra
$ chown cassandra /var/lib/cassandra
$ mkdir /var/log/cassandra
$ chown cassandra /var/log/cassandra
```

You can change this in the settings explained in a future section.

To run cassandra switch the cassandra user, and execute the cassandra executable

```
$ sudo su cassandra
$ cd $CASSANDRA_HOME
$ ./bin/cassandra -f # or if you want to run
    it in the background -d
```

And you have cassandra running! To make sure that everything is working on a different terminal

```
$ cd $CASSANDRA_HOME
$ ./bin/cqlsh
>       #If there are no errors, your single
    node cluster is up.
```

CONFIGURING MULTINODE CLUSTERS

Backing up date on previous

First, if you have a single node cluster you **must** complete these instructions or your system will be in a halfway state. Delete all the data associated with your cassandra instance. If you have important data, there are ways to back it up.

```
> ./bin/cqlsh
> COPY table_name TO '/user/you/file_name'
# After the switch to multinode you can do
> COPY table_name FROM '/user/you/file_name'
```

If you are comfortable with your data backup, delete the old datastore

```
sudo rm -rf /var/lib/cassandra/*
```

Setting up the configurations

Got to the $CASSANDRA_HOME/cassandra.yaml file. You can update the following values to set up a multi node system

```
# Find these values in the existing file
cluster_name: 'MustBeNamed'

# Some fields may be added
seed_provider:
  - class_name: org.apache.cassandra.locator.
      SimpleSeedProvider
    parameters:
        - seeds: "this_server_ip,server_ip_2
            ,..."
```

```
# Firewall Addresses
rpc_address: this_server_ip
listen_address: this_server_ip

# Snitches
endpoint_snitch: GossipingPropertyFileSnitch
```

1. cluster_name: The name of your cluster. In order for other nodes to join it must be named the **same thing** on each server.

2. seed_provider: This property is how cassandra knows about which nodes are directly connected to which. It will assume that the nodes are connected in a ring where the first ip is connected to the second, the second to the third and so on and the last is connected to the first.

3. rpc_address, listen_address: These are the two IPs that cassandra will do its communication on. These are usually localhost and default to two ports.

4. endpoint_snitch: A snitch in computing shares metadata about the cluster (the health of each node, the free space etc) by occasionally telling its neighbors (fingertable accessible). This is the gossip protocol in work and that is why we go with the GossipingPropertyFileSnitch and not the SimpleSnitch for fault tolerance.

For those on linux systems iptables are the usualy firewall the kernel has up to block incoming traffic, we will need to circumvent that.

```
# Redo the following command for each
# machine connected to a single machine.

$ iptable -A INPUT \ # We are going to be
    receiving data
  -p tcp -s connected_machine_ip \ # Replace
      the IP as goes
  -m multiport --dports 7000,9042 \ # Default
      port
  -m state --state NEW,ESTABLISHED -j ACCEPT
$ service iptables-persistent restart
```

To see if it worked try the following command.

```
$ $CASSANDRA_HOME/bin/cqlsh ANY_SERVER 9042
```

OVERVIEW OF DATA MODELING

If you are familiar with more traditional Relational Database Management Systems (RDBMS) then you are familiar with the concept of tables and joins where the table represents spreadsheets

CQLSH

TUNING PARAMS

USUAL DO NOTS

Although cassandra gives you a lot of tunability parameters there are practices that should be avoided at all cost because it severly impacts the performance or the reliability of the database.

1. Do not put cassandra's data files on network attached storage

2. Try not to host cassandra's data files directly on hadoop either, cassandra has reliability built into it.

REFERENCES