Databases

Harpreet Singh

August 16, 2015

1. Basics

Definition 1.1. High-availability clusters are groups of computers that support server applications that can be reliably utilized with a minimum of down-time. [1] They operate by using high availability software to harness redundant computers in groups or clusters that provide continued service when system components fail.

Definition 1.2. A database is an organized collection of data.

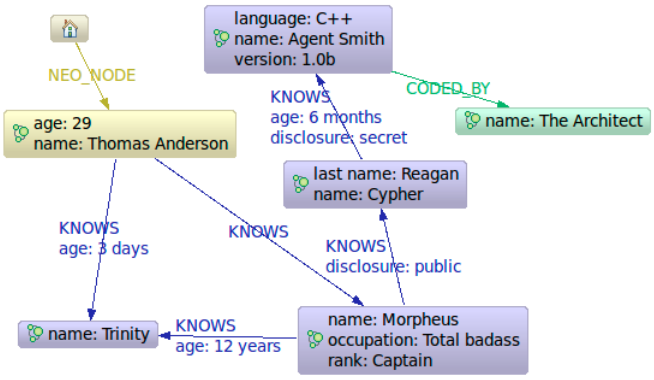
Definition 1.3. A partition is a division of a logical database or its constituent elements into distinct independent parts. Database partitioning is normally done for manageability, performance or availability reasons.Horizontal partitioning involves putting different rows into different tables. [2]

Definition 1.4. A database shard is a horizontal partition of data in a database or search engine. Each individual partition is referred to as a shard or database shard [3]. Each shard is held on a separate database server instance, to spread load.

2. Some Databases and their configuration

2.1 Titandb

Definition 2.1. A graph database is a database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data. A graph database is any storage system that provides index-free adjacency.This means that every element contains a direct pointer to its adjacent elements and no index lookups are necessary [4] Example-



[5]

Titan is a scalable graph database maintained by Aurelius [6]

2.1.1 General usage of Titandb on local machine

Download suitable Titandb version from Aurelius github page and then its very simple to add,remove or edit nodes and vertices using Gremlin shell commands. Few examples can be seen here and here from slide-41.

2.1.2 Setting up Titan for Highly available and distributes cluster [7]

Horizontal scaling is very trivial and easy in Titan graph databases. Following instructions are with respect to titan-cassandra-0.3.0

- Have some machines/servers to host data.
- In configuration files of Titandb folders, change localhost to your machine's own IP address. Difference can be seen here. You can simply use this command

```
titan-cassandra-0.3.0$grep -R localhost * | xargs sed -i
's/localhost/<ip address >/g'
```

- in casaandra.yaml file from config directory,change seed address from "127.0.0.0" to "<primary/master ip address>". So, the common relating thing on defferent servers would be this seed address
- Start titandb from all servers one by one using

```
titan-cassandra-0.3.0$sudo bin/titan.sh config/titan-server-rexster.xml
config/titan-server-cassandra.properties
```

2.1.3 Description of above configuration [7]

By default Replication factor in these clusters is 3. It can be updated using apache-cassandra by this command

```
apache-cassandra-2.1.6$ bin/cassandra-cli -h <Primary/master ip address>
```

You can view the status of the cluster ring by

```
apache-cassandra-2.1.6$ bin/nodetool ring
```

- In a replica set,Each of the 3 Titan Servers are the primary host of approximately one-third of the vertices in the graph while, at the same time, each maintains a replica of the primary data of the other two servers. In this way, a highly-available, master-master setup is rendered. With this model, there is no single point of failure. If one of the database machines goes down, the other two are able to serve the primary data of the dead machine. If two of the machines go down, the remaining machine can serve data albeit not with the same throughput possible when all three machines are available.
- With more servers,say 9,With the replication factor still set to 3, each server does not maintain a full replica of the graph. Instead, each server only replicates a third of the full graph (3/9). At this point, no single server has a full picture of the graph. However, because there are more servers, more transactions can be served and more data can be stored.

Bibliography

[1] Highly-availability clusters. https://en.wikipedia.org/wiki/High-availability_cluster.

[2] Partitioning. https://en.wikipedia.org/wiki/Partition_(database).

[3] Shards. https://en.wikipedia.org/wiki/Shard_(database_architecture).

[4] Graph databases. https://en.wikipedia.org/wiki/Graph_database.

[5] Graph databases example. https://jasperpeilee.files.wordpress.com/2011/11/image4.png.

[6] Courtesy of aurelius. http://thinkaurelius.github.io/titan/.

[7] Instructions for setting up titan clusters. http://thinkaurelius.com/2013/03/30/titan-server-from-a-single-server-to-a-highly-available-cluster/.