IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

NCRTET-17

National Conference on Recent Trends in Engineering & Technology

Government Polytechnic College Kasaragod, Kerala

Vol. 4, Special Issue 1

Apache Cassandra-The Data Storage Framework for Hadoop

Sereena.M.V[1], Zahid Ansari[2]
Student, Department of Computer Science and Engineering, P. A. College of Engineering, Mangalore, India[1].
Dean Research, Department of Computer Science and Engineering, P. A. College of Engineering, Mangalore, India[2]

Abstract—The primary data storage for business applications use RDBMSs (Traditional relational databases) for 20 years. Today, another revolutionize is required because most of the applications must now scale to levels that were unbelievable just a few years ago. But scaling alone isn't enough; companies also require that their applications are always available and scattering fast. Hence Apache Cassandra is a massively scalable Distributed database (NoSQL) that allows for amazing performance at extreme data. This paper provides a brief overview of the Apache Cassandra.

Keywords— Hadoop, HDFS, MapReduce, Cassandra, CQL.

I. INTRODUCTION

Big data [1] refers to datasets whose size is further than the ability of typical database software tools to capture, store, manage and analyze. As the technology advances over time, the size of dataset will also increased. Today, big data span in many sectors and it will range from a few dozen terabytes to multiple petabytes. The main characteristics of big data are volume, velocity and variety. There are many Big Data technologies are used for handling Big Data, but Apache Hadoop [2], [3] is one technology that has been the most famous of Big Data talk. Hadoop is an open-source platform for storage and processing of various data types like structured, semi structured and unstructured data .The main components of Hadoop are: HDFS and Map Reduce.The Hadoop Distributed File System (HDFS) [4], [5] is a distributed file system designed to run on commodity hardware. That is HDFS is the storage system for a Hadoop cluster. HDFS provides efficient access to application data and is suitable for applications having big data sets. Hadoop MapReduce [6], proposed by Google which is the most popular open source implementation of the Map Reduce framework. Generally, MapReduce is a programming model for processing the variety of data. It consists of two user-defined functions: map and reduce. The input of a Hadoop Map Reduce is a set of key-value pairs (key, value) and the map function is called for each of these pairs. A NoSQL database (called as Not Only SQL) is a distributed database that provides a mechanism to store and retrieve data other than the tabular relations used in relational databases. These databases are schema-free, support easy replication, and it can handle large amounts of data. Apache Cassandra [8] is a type of NoSQL database and it is highly scalable, high-performance distributed database designed to handle large amounts of data across many servers. It has no single point of failures. Next section describes the methodology of Cassandra. Section 3 describes data model. Section 4 describes cloud support in Cassandra and last section describes the developing Cassandra applications.

II. OVERVIEW OF CASSANDRA

Cassandra [9], [10] is an open source, non-relational, decentralized, column oriented, distributed database, developed for storing large amounts of unstructured data over commodity servers. It is developed by Facebook and it is a column oriented database that use distribution design on Amazon's Dynamo [8] and its data model on Google's big table. Cassandra is based on a peer-to-peer model, which makes it tolerant against single points of failure and provides horizontal scalability. The Dynamo-style replication model is implemented in Cassandra without any single point of failure, but it adds a more powerful "column family" data model. Cassandra is used at some of the most popular site on the web. That is used by some of the biggest companies such as Facebook, Twitter, Cisco, Rack space, Twitter, Netflix, and more. In this section, we discuss features, architecture, components and applications.

A. Features of Cassandra

Cassandra becomes more popular because of it has outstanding technical features. The Table I show the features [10] of Cassandra.

TABLE I
FEATURES OF CASSANDRA

Features	Meaning
Elastic scalability	It provides high scalability. As per the requirement of the data it allows to add more hardware to accommodate more customers and more data.
Availability	It is continuously available for business-critical applications without any single point of failure.
Fast linear-scale performance	It is linearly scalable, i.e., if we increase the number of nodes in the cluster then it will increase the throughput. Therefore it maintains a quick response time.
Flexible data storage	It can store all possible data formats.
Easy data distribution	It provides the elastically to distribute data where you need by replicating data across multiple datacenters.
Transaction (ACID) support	It supports transaction properties (ACID) like Atomicity, Consistency, Isolation, and Durability.
Fast writes	It was designed to run on cheap commodity hardware. Cassandra can store hundreds of terabytes of data by performing fast writes, without sacrificing the read efficiency.

B. Architecture

The main objective of Cassandra is to handle big data workloads across multiple nodes without any single point of failure. Cassandra support peer-to-peer distributed system across its nodes, in which data is distributed among all the nodes in a cluster. All the nodes in a cluster perform the same job. Every node in the cluster act as an independent node and at the same time they are interconnected to each other nodes. Each node in a cluster can accept read and write requests, in spite of where the data is actually located in the cluster. When a node becomes fail, the read/write requests can be served from other nodes in the network.

C. Data Replication in Cassandra

In Cassandra, some of the nodes play as replicas in a cluster for a given piece of data. Sometimes, some of the nodes answered with an out-of date value. If it is detected, Cassandra will return the most recent value to the client. After that Cassandra performs a read repair that is running on the background to update the old values. Fig. 1 explains how the data replications work.

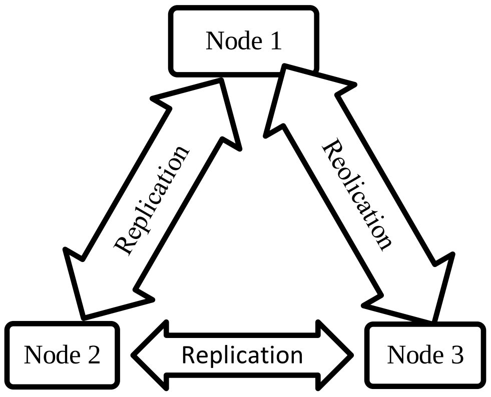

Fig. 1 Schematic view of how Cassandra uses data replication

D. Components

The key components of Cassandra are as shown in Table II.

TABLE II
COMPONENTS OF CASSANDRA

Components	Description
Node	It is the place of data where it is stored.
Data center	It is a collection of related nodes.
Cluster	It is a component that contains one or more data centers.
Commit log	It is used for crash-recovery mechanism. Data is written to the commit log after every write operation.
Mem table	A Mem-table is a memory-resident data structure.The data will be written to the Mem-table after commit log.
Sorted String Table(SST)	It is a disk file. If Mem-table exceeds some threshold value then it will be flushed to SSTable.
Bloom filter	It is a special kind of cache. These are quick, nondeterministic, algorithms for checking whether an element is a member of a set. These are accessed after every query.

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

NCRTET-17

National Conference on Recent Trends in Engineering & Technology

Government Polytechnic College Kasaragod, Kerala

Vol. 4, Special Issue 1

E. Applications of Cassandra

Some of the applications are:

- Real-time, big data workloads.
- Media streaming management (e.g., music, movies).
- Social media (i.e., unstructured data) input and analysis.
- Online web retail (e.g., shopping carts, user transactions).
- Real-time data analytics.
- Online gaming (e.g., real-time messaging).
- Software as a Service (SaaS) applications that utilize web services.
- Online portals (e.g., healthcare provider/patient interactions).
- Most write-intensive systems.

III. DATA MODEL

The data model [1], [12] of Cassandra is significantly different from traditional relational database (RDBMS). This provides an overview of how Cassandra the data is stored.

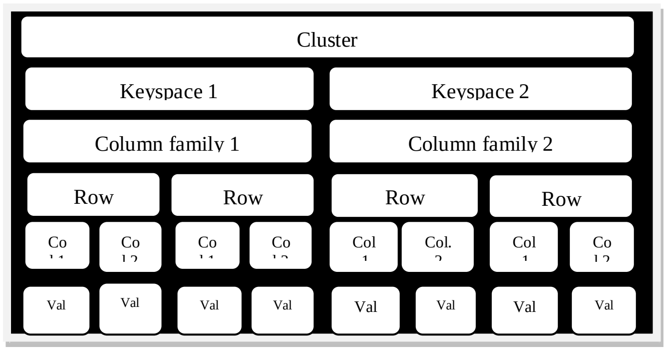

Fig. 2 Hierarchy of the Cassandra data model.

A Cassandra cluster can contains many key spaces, which can contain various column families. These contain rows, which are further contains one or more columns. The number of columns can vary between rows. Those are shown in Fig. 2.Cassandra database is distributed over several machines that operate together. The most outermost container is known as the Cluster, sometimes called the ring, because Cassandra assigns data to nodes in the cluster by arranging them in a ring. For handling failures, every node contains a replica, and in case of a failure, the replica takes charge. Next we discuss the data model of a Cassandra.

1) Keyspace: It is the outermost container for data. The Fig. 3 show the schematic view of Keyspace [13]. The basic attributes in Cassandra are:

- Replica placement strategy: It is the strategy to place replicas in the ring. The strategies includes such as simple strategy (rack aware strategy), old network topology strategy (rack-aware strategy), and network topology strategy (datacenter-shared strategy).
- Replication factor: It counts the number of machines in the cluster that will receive copies of the same data.
- Column families: Keyspace is a container for a list of one or more column families. A column family is a container of a collection of rows. Each row represents the ordered columns. Column families represent the structure of your data. Each Keyspace has at least one and often many column families.

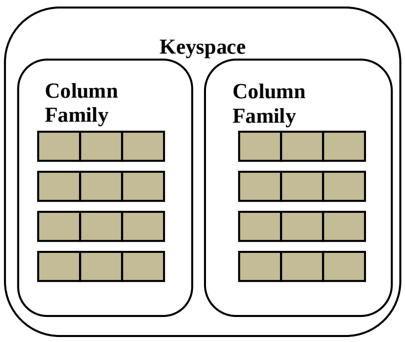

Fig. 3 Schematic view of Keyspace

The syntax of creating a Keyspace is as follows:

CREATE KEYSPACE Keyspace1 name WITH replication = {'class': 'Simple Strategy','replication_factor': 4};

2) *Column Family*: The column family structure is as shown in figure 4.A column family is a container for an ordered collection of rows. Each row is an ordered collection of columns. The Table III describes major difference between the relational table and column family.

TABLE III

RELATIONAL TABLE VS CASSANDRA COLUMN FAMILY

Relational Table	Cassandra Column Family
A schema is fixed. Once we define certain columns for a table, while inserting data, in every row all the columns must be filled at least with a null value.	First we define the column families, not columns. So we can freely add any column to any column family at any time.
It defines only columns and the user fills in the table with values.	A table contains columns, or can be defined as a super column family.

Fig. 4 shows the structure of Cassandra column family. The column family has the following attributes:

- Rows_cached: It represents the number of rows whose entire contents will be cached in memory.
- Keys_cached: It represents the number of locations to keep cached per SSTable.
- Preload_row_cache: It specifies whether we want to pre-populate the row cache.

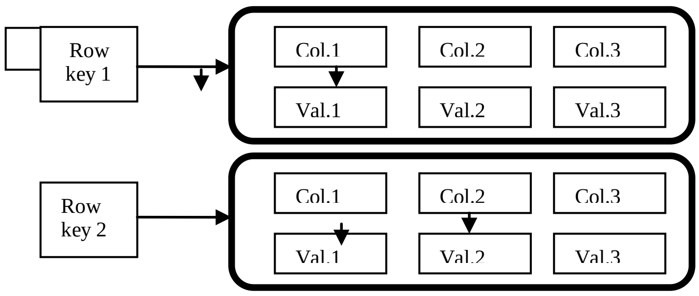

Fig. 4 Cassandra column family

3) *Column*: It is the basic data structure of Cassandra with three values, namely column name or key, value, and a time stamp. Fig. 5 shows the structure of a column.

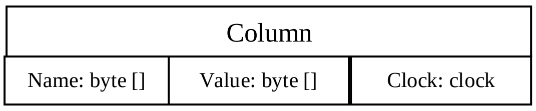

Fig. 5 Column

4) *Super Column*: A super column is a special column structure i.e.; a key-value pair. But a super column stores a map of sub-columns. Generally column families are stored on disk in individual files. The structure of a super column is as shown in Fig. 6.

Super Column	
Name: byte []	Cols: map<byte [], column>

Fig. 6 Super column

IV. COMPARISONS OF CASSANDRA & RDBMS

The Table IV lists down the points that differentiate the data model of Cassandra from that of an RDBMS [14].

TABLE IV

RDBMS VS CASSANDRA

RDBMS	Cassandra
It deals with structured data.	It deals with unstructured data.
It has a fixed schema.	It has a flexible schema.
A table is an array of structures.(ROW x COLUMN)	A table is a list of "nested key-value pairs". (ROW x COLUMN key x COLUMN value)
Database is the outermost container that contains data corresponding to an application.	Keyspace is the outermost container that contains data corresponding to an application.
Tables are the main entities of a database.	Tables or column families are the main entity of a Keyspace.
Row (tuple) is an individual record.	Row represents a unit of replication.
The attributes of a relation are represented by Column.	Column represents unit of storage.
It supports the concepts of foreign keys, joins, etc.	Relationships are represented using collections.

V. DEVELOPING CASSANDRA APPLICATIONS-CQL

Using the Google Big-table model, Cassandra provides more flexibility to store structured, semi-structured, and unstructured data. The primary container of data is a Keyspace, which is like a database in an RDBMS. Inside a Keyspace are one or more column families, which are like relational tables, but they are dynamic in structure. Column families have one to many thousands of columns, with both primary and secondary indexes on columns being supported. In Cassandra, objects are created, data is inserted and manipulated, and information queried via CQL – the Cassandra Query Language [16], which is identical to SQL.CQL can use standard commands (e.g., INSERT, SELECT) to interact with objects and data stored in Cassandra. Users can access Cassandra through its nodes using Cassandra Query Language (CQL). CQL treats the database (Key space) as a container of tables. Programmers use cqlsh: a prompt to work with CQL or separate application language drivers. Clients approach any of the nodes for their read-write operations. That node (coordinator) plays a proxy between the client and the nodes holding the data.

VI. CONCLUSIONS

This paper concludes that a modern application requires storing of large amount of data for handling big data workloads. So Apache Cassandra is the one of the data storage frame work for Hadoop and it is a type of NoSQL database. Its main advantage is the ability to offer a fast linear scale performance when we add more nodes in the cluster. Users can access the Cassandra by using CQL. It provides very efficient read and write access for big data applications.

ACKNOWLEDGMENT

We would like to thank to all staffs and friends at P.A College of engineering, Mangalore for their feedback and support.

REFERENCES

[1] Dede, Elif, et al. "*An Evaluation of Cassandra for Hadoop*." IEEE CLOUD 2013 (2013): 494-501.

[2] Dede, Elif, et al. "*A processing pipeline for cassandra datasets based on Hadoop streaming*." Big Data (BigData Congress), 2014 IEEE International Congress on. IEEE, 2014.

[3] Chris Eaton, Dirk Deoos, Tom Deutsh, George Lapis, Paul Zikopolous-Undertsanding Big data (Analytics for Enterprise class Hadoop and streaming data), Mc Graw Hill.

[4] Konstantin Shvachko, Hairng Kuang, Sanjay Radia, Robert Chansler-The Hadoop Distributed File System.

[5] Karun, A. Kala, and K. Chitharanjan. "*A review on Hadoop—HDFS infrastructure extensions*." Information & Communication Technologies (ICT), 2013 IEEE Conference on. IEEE, 2013.

[6] Jens Dittrich Jorge-Arnulfo Quian´e-Ruiz, Efficient Big Data Processing in Hadoop Map Reduce.

[7] Tom White,"*Hadoop: The definitive guide*", Third edition, O'reoilley, 2012.

[8] Eben Hewitt-Foreword by Jonathan Ellis, Apache Cassandra Project chair, Cassandra, The definitive guide (Distributed data at web scale), O'Reilly.

[9] A. Lakshman and P. Malik. Cassandra: structured storage system on a p2p network. In Proceedings of the 28th ACM symposium on Principles of distributed computing, PODC '09, pages 5–5, New York, NY, USA, 2009. ACM.

[10] Ruchira A. Kulkarni, Evaluating Cassandra Data-sets with Hadoop Approaches, International Journal of Trend in Research and Development, Volume 2(5), ISSN 2394-9333.www.ijtrd.com.

[11] www.tutorilaspoint.com/cassandra_tutorial-simplyeasylearning.

[12] Himadri Sekhar Ray1, Kausik Naguri1, Poly Sil Sen2 and Nandini Mukherjee1, Comparative Study of Query Performance in a Remote Health Framework using Cassandra and Hadoop

[13] Bagade, Prasanna, Ashish Chandra, and Aditya B. Dhende. "*Designing performance monitoring tool for NoSQL Cassandra distributed database*." Education and e-Learning Innovations (ICEELI), 2012 International Conference on. IEEE, 2012.

[14] Chebotko, Artem, Andrey Kashlev, and Shiyong Lu. "*A big data modeling methodology for Apache Cassandra*." Big Data (BigData Congress), 2015 IEEE International Congress on. IEEE, 2015.

[15] DataStax-Introduction to Apache Cassandra.

[16] http://cassandra.apache.org/.