Migration from relational database like MySQL to nosql database like Cassandra is necessary and how to migrate it using spark

Dr. Kishor Atkotiya [1]
Department of Statistics, Saurashtra University - Rajkot

Parag Shukla [2]
Atmiya Institute of Technology & Science, Rajkot

Abstract: For more than 15 years, Oracle MySQL has been a real structure piece in web technology and its applications, enjoying large acceptance. This is oftentimes permanently reason: MySQL offers a strong database which enables firms to make system that execute comfortably in various use cases. Yet, still it's strongest supporters acknowledge that its not architected to appurtenances the new curve of monolithic knowledge application. Dashing businesses that enjoin to control use cases square evaluate delivery to forge a different set of technologies to alternate MySQL. This article examines the why and how migrating from Oracle MySQL to those new large knowledge technologies.

Keywords*:* MySQL, Cassandra, Migration, BigData Analytics, Spark, Migration from MySQL to Cassandra using SPARK, NoSQL Database

I. INTRODUCTION

MySQL provides massive content electronic database on-line database machine database electronic accumulation service manage method (RDBMS) capabilities in affiliate open supplying packet that allows firms to create systems that execute comfortably from a database appearance in various common goal use cases. In 2008 MySQL was non inherited by Sun Microsystems and was later in Gregorian calendar month of 2010 formally non inherited by Oracle (via its acquisition of Sun Microsystems). Currently a part of Oracle stable of info products, MySQL continues to be promoted and sold-out through Oracle. Variety of diff vendors acquire either ambiguous MySQL to form other info sharing or area unit usage MySQL as a part of a specialised service sharing. ex. Amazon RDS, Calpont, Infobright, Monty Program AB, embody Percona etc.

While Oracle MySQL remain an genuine RDBMS that performs fine for the utilization cases it had been organized for, even it's strongest supporters allow that its not architected to tackle the new curve of massive knowledge application being formed on those days. In fact, the requirements lately 21st century internet firms helped offer born to MySQL and drive it's success, trandy businesses that demand to manage massive knowledge use cases are delivery to forge a definite set of tech that are clemency MySQL in serveral area.

This article examines the why and how migrating from Oracle MySQL to new Big data technoligy, such as Apache Hadoop abd Cassandra.

Why Migrate From MySQL ?

MySQL could be a widespread alternative for brand new comes. it is a versatile info that is simple to line up and begin querying. there is various documentation, examples and frameworks it works with, like Wordpress, Ruby on Rails, and Django the matter arises once you have to be compelled to scale past one server or have high accessibility wants. MySQL's answer to each of those wants is replication. Replication is ok at handling browse serious workloads during a single datacenter, however it falls on it's face underneath serious writes or if you wish multiple datacenters. as luck would have it prophetess excels at quantifiability and high accessibility. it is a common story for folks to migrate from a computer database to prophetess for one or each of those reasons.

Example to Migrate From MySQL to CASSANDRA

MySQL that store consumer event information that references staff members and accumulation from a distinct table

```
create table store(
store_namevarchar(32) primary key,
locationvarchar(32),
store_typevarchar(10));

create table staff(
namevarchar(32)
primary key,
favourite_colourvarchar(32),
job_titlevarchar(32));

create table customer_events(
id MEDIUMINT NOT NULL AUTO_INCREMENT
PRIMARY KEY,
customervarchar(12),
time timestamp,
event_typevarchar(16),
storevarchar(32),
staffvarchar(32),
foreign key fk_store(store) references store(store_name),
foreign key fk_staff(staff) references staff(name));
```

Insert a few rows

```
insert into staff(name, favourite_colour, job_title) values ('Parag', 'Blue', 'Awesome Marketer');

insert into store(store_name, location, store_type) values ('atmiya', 'US', 'WEB');

insert into customer_events(customer, time, event_type, store, staff) values ('chbatey', now(), 'BUY_MOVIE', 'atmiya', 'Parag');

insert into customer_events(customer, time, event_type, store, staff) values ('chbatey', now(), 'WATCH_MOVIE', 'atmiya', 'Parag');
```

Okay therefore we tend to solely have a couple of rows however imagine we tend to had several numerous client event & within the order of many employees member & store.

Now lets have a look at however we are able to transmigrate to prophetess with couple of line of Spark code.

Spark has in built support for database which have JDBC driver via JdbcRDD. prophetess have good support for Spark via DataStax open supply connecter. We will be exploitation the 2 along to migrate knowledge from MySQL to prophetess.

```
importcom.datastax.spark.connector._
importcom.datastax.spark.connector.cql.CassandraConnector
importorg.apache.spark._
importorg.apache.spark.rdd.JdbcRDD
```

After that we can create Spark Context and adds Cassandra methods to the context and to RDDs.

```
valconf = new
SparkConf().set("spark.cassandra.connection.host",
"127.0.0.1")
valsc = new SparkContext("local[2]",
"MigrateMySQLToCassandra", conf)
valmysqlJdbcString: String =
s"jdbc:mysql://10.9.150.45/customer_events?user=root&p
assword=password"
Class.forName("com.mysql.jdbc.Driver").newInstance
```

My MySQL server is running on IP 10.9.150.45 and connecting securely with username and password.

Next we'll create the new Cassandra table, if yours already exists skip this part.

```
CassandraConnector(conf).withSessionDo
{
session =>
session.execute("CREATE KEYSPACE IF NOT
EXISTS test WITH replication =
{'class': 'SimpleStrategy', 'replication_factor': 1 }")
session.execute("CREATE TABLE IF NOT EXISTS
test.customer_events(   customer_id text,   time
timestamp, id uuid,  event_type text, " +
    "store_name text, store_type text, store_location
text, staff_name text, staff_title text,  PRIMARY KEY
((customer_id), time, id))")
  }
```

For Migration

```
valcustomerEvents = new JdbcRDD(sc, () => {
DriverManager.getConnection(mysqlJdbcString)},
  "select * from customer_eventsce, staff, store where
ce.store = store.store_name and ce.staff = staff.name "
+
     "and ce.id >= ? and ce.id <= ?", startingId,
highestId, numberOfPartitions,
   (r: ResultSet) => {
     (r.getString("customer"),
r.getTimestamp("time"),
UUID.randomUUID(),
r.getString("event_type"),
r.getString("store_name"),
r.getString("location"),
r.getString("store_type"),
r.getString("staff"),
r.getString("job_title")
       )
   })

customerEvents.saveToCassandra("test",
"customer_events",
SomeColumns("customer_id", "time", "id",
"event_type", "store_name", "store_type",
"store_location", "staff_name", "staff_title"))
```

First produce a JdbcRDD permitting MySQL to try to be part of. You would like to grant Spark to partition the MySQL table, therefore you provide it a press release with variable in and a beginning index and a final index. You furthermore might tell Spark what percentage partitions to separate it into, you would like this to be larger than the amount of cores in your Spark cluster therefore these will happen at the same time.

Finally we tend to put it aside to Cassandra.The possibilities area unit this migration are going to be bottle polo-neck by the queries to MySQL. If the shop and employees table area unit comparatively tiny it might be value transferal them utterly in to memory, either as associate degree RDD or as associate degree actual map so MySQL does not ought to be part of for each partition.

Assuming your Spark staff area unit running on constant servers as your Cassandra nodes the partitions are accomplishment to be isolated and inserted regionally to each node in your cluster.

Who's Using CASSANDRA?

One advantage that MySQL users have enjoyed could be a anomaly group of users UN agency have deployed the info in various production environment. Cassandra, likewise, is working in various industries for current applications which require scale, performance, knowledge, flexibility and common knowledge distribution.

Conclusion

No argument that Oracle MySQL may be a sensible RDBMS – and that serves the utilization cases that it had been originally designed. except for IT professionals United Nations agency square measure either coming up with new huge information applications or existing MySQL systems that have begun to interrupt down beneath huge information work loads, a move to DataStax Enterprise & prophetess makes each business and technical sense. Change to a contemporary, huge information platform like DataStax Enterprise can future proof any application, and provide confidence that the system can scale and perform good each currently and into a rigorous future.

References

1. White Paper by DataStax Corporation - 2012.
2. https://en.wikipedia.org/wiki/Comparison_of_structured_storage_software
3. http://image.slidesharecdn.com/introtocassandra-141020133424-conversion-gate02/95/intro-to-cassandra-25-638.jpg?cb=1414049596
4. http://vschart.com/compare/apache-cassandra/vs/mysql
5. https://wiki.apache.org/cassandra/ArchitectureOverview
6. Katarina Grolinger1, Michael Hayes1, Wilson A. Higashino1,2, Alexandra L'Heureux1 David S. Allison1,3,4,5, Miriam A.M. Capretz1. "Challenges for MapReduce in Big Data " 2014 IEEE 10th World Congress.