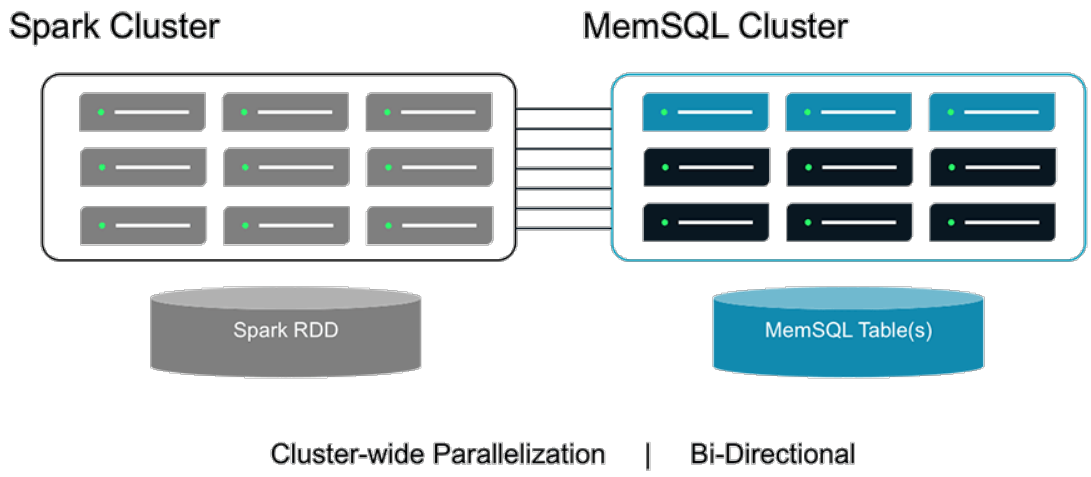
使用Spark和MemSQL Spark连接器运行实时应用

Apache Spark是目前非常强大的分布式计算框架。其简单易懂的计算框架使得我们很容易理解。虽然Spark是在操作大数据集上很有优势，但是它仍然需要将数据持久化存储，HDFS是最通用的选择，和Spark结合使用，因为它基于磁盘的特点，导致在实时应用程序中会影响性能（比如在Spark Streaming计算中）。而且Spark内置就不支持事务提交(commit transactions)。

本文介绍的MemSQL 数据库号称是世界上最快的分布式内存数据库（The World's Fastest In-Memory Database）！它是由Eric Frenkiel（前Facebook员工）和Nikita Shamgunov（前微软SQL Server高级工程师）创建的一款基于内存的分布式关系数据库，它通过将数据存储在内存中，并将SQL语句预编译为C++而获得极速的执行效率。它兼容MySQL，且速度要比MySQL快30倍，能实现每秒150万次事务。

最近在其官方发布的一个MemSQL Spark Connector可以很好地和Spark一起使用，使得Spark用户可以快速地读写数据库中的数据。MemSQL 天生就适合Spark，因为它可以高效地处理大量的读写，而Spark经常需要这样的操作，而且MemSQL可以提供大量的空间足以提供给Spark创建

新的数据。



MemSQL Spark Connector提供了所有Spark和MemSQL交互的各种接口，而且其中做了许多的优化措施，比如并行地从MemSQL读取数据；当 MemSQL和Spark运行在一个物理节点上，Spark直接将数据写入其中。MemSQL提供了两个最主要的组建：MemSQLRDD和saveToMemsql。

MemSQLRDD用于存储从MemSQL查询的数据集；而saveToMemsql将Spark中的RDD数据写入到MemSQL表中。这两个接口和Spark内置的JDBC接口看起来很类似，而且使用方式也很类似（可以看这里《Spark与Mysql(JdbcRDD)整合开发》）。来看看如何使用MemSQLRDD。我们使用从MemSQL读取表数据，并存储在MemSQLRDD中：

```
import com.memsql.spark.connector.rdd.MemSQLRDD

...

val rdd = new MemSQLRDD(
   sc,
   dbHost,
   dbPort,
   dbUser,
   dbPassword,
   dbName,
   "SELECT * FROM iteblog",
   (r: ResultSet) => { r.getString("test_column") })
rdd.first()  // Contains the value of "test_column" for the first row
```

如果你想将RDD写入到Memsql，可以使用saveToMemsql函数：

```
import com.memsql.spark.connector._

...

val rdd = sc.parallelize(Array(Array("www", "iteblog"), Array("com", "qux")))
rdd.saveToMemsql(dbHost, dbPort, dbUser, dbPassword,
   dbName, outputTableName, insertBatchSize=1000)
```

从上面的例子可以看出，使用Memsql和Spark结合是多么的容易。

本文翻译自: http://blog.memsql.com/memsql-spark-connector/