上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Database optimization on Spark

Yuezhou Liu

May 15th,2017

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

My work in group

- Group : Database group

- Work: Development of Spark cluster and using Spark cluster to optimize database querying.

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Outline

Why Spark

- Compare Spark and Hadoop
- Compare Spark and MySQL

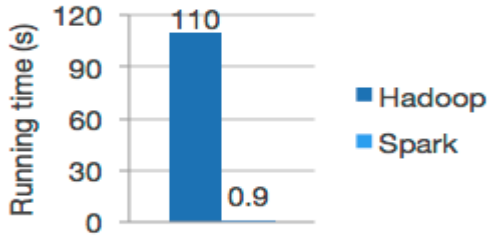
My works

- Work1 :Construction of Spark cluster.
- Work2 : Using Spark for database querying and some optimizations.
- Work3 : Joint operation of Spark and Hive.

- Key words: big data, data computing, data storage, distributed system.
- Both Spark and hadoop have those above features, but Spark performs better in many ways.

Compare Spark and Hadoop

- Spark is much quicker than hadoop.



Logistic regression in Hadoop and Spark

- Spark has more operations, thus, easier to use.

- hadoop: map, reduce.
- Spark: Map、Filter、FlatMap、Sample、GroupByKey、ReduceByKey、Union、Join、Cogroup、MapValues, etc.

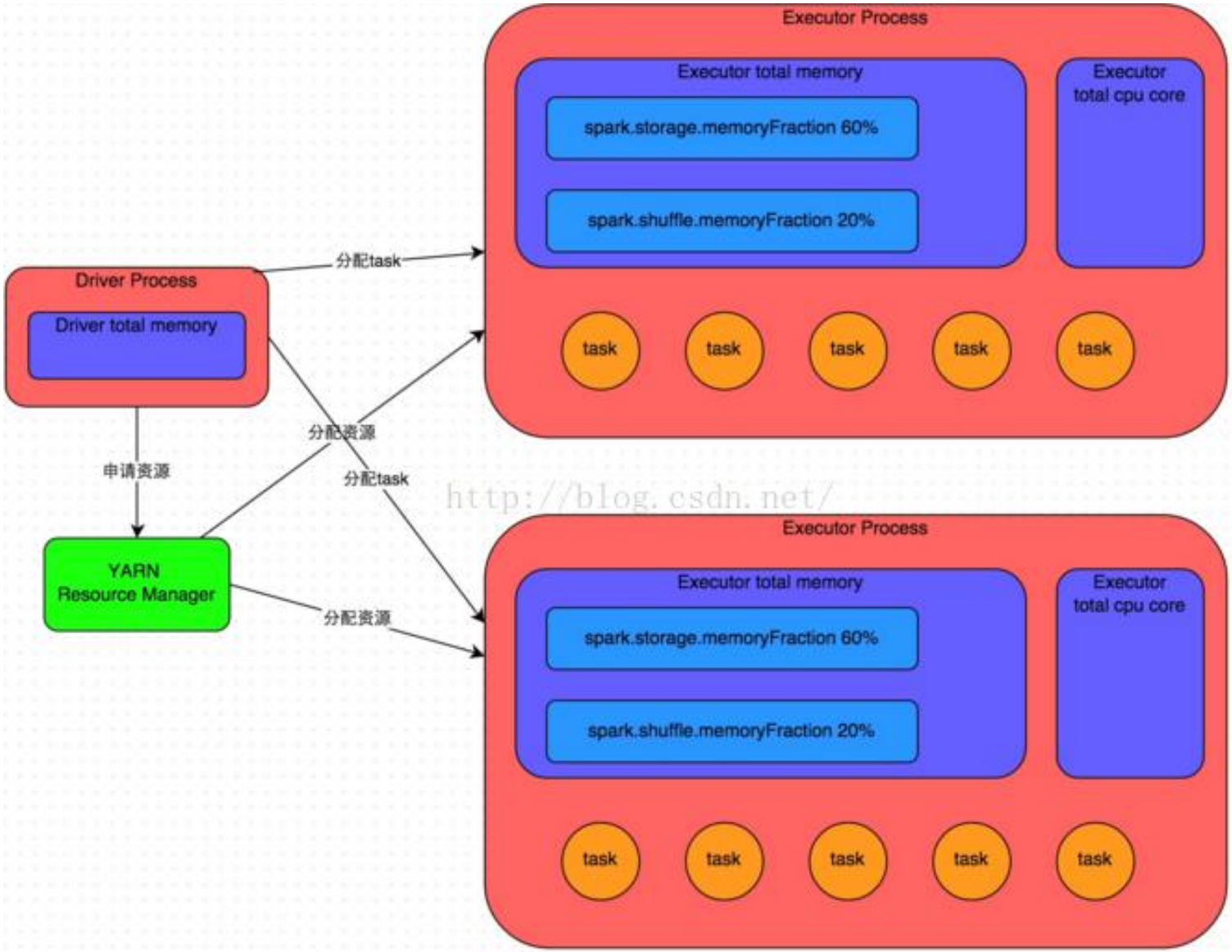
- Other reason to choose Spark.

Compare Spark and MySQL

- MySQL: uses only one CPU core for a single query.
- Spark: uses all available CPU cores.

My works

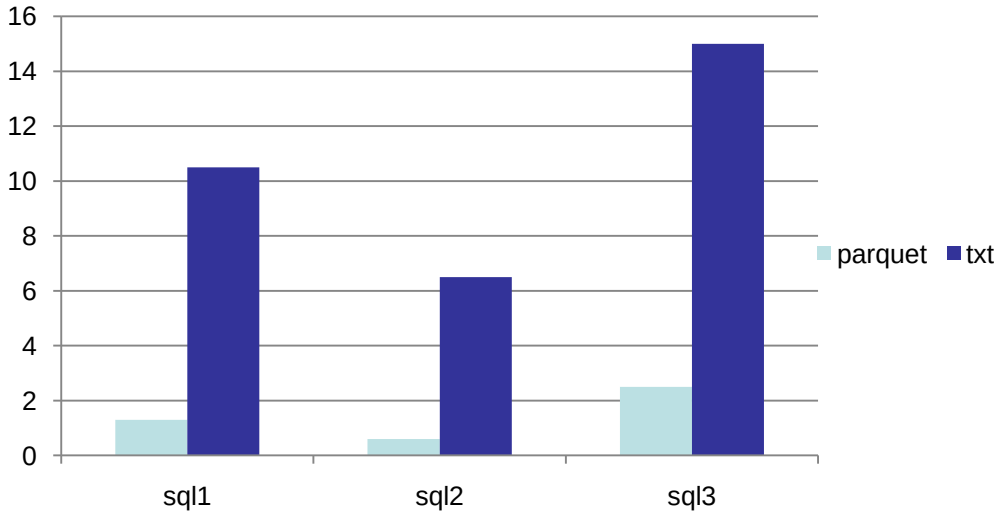
How Spark works?



Pre-work of data migration

- Using sqoop to migrate some database from MySQL to HDFS(hadoop).

- Sqoop can migrate data into different types of files, so I compare the performance of Spark on different files.



Two ways to do a querying

- Using operations for RDD

```
// Select people older than 21
df.filter($"age" > 21).show()
// +---+----+
// |age|name|
// +---+----+
// | 30|Andy|
// +---+----+

// Count people by age
df.groupBy("age").count().show()
// +----+-----+
// | age|count|
// +----+-----+
// |  19|    1|
// |null|    1|
// |  30|    1|
// +----+-----+
```

Two ways to do a querying

- Using SparkSQL

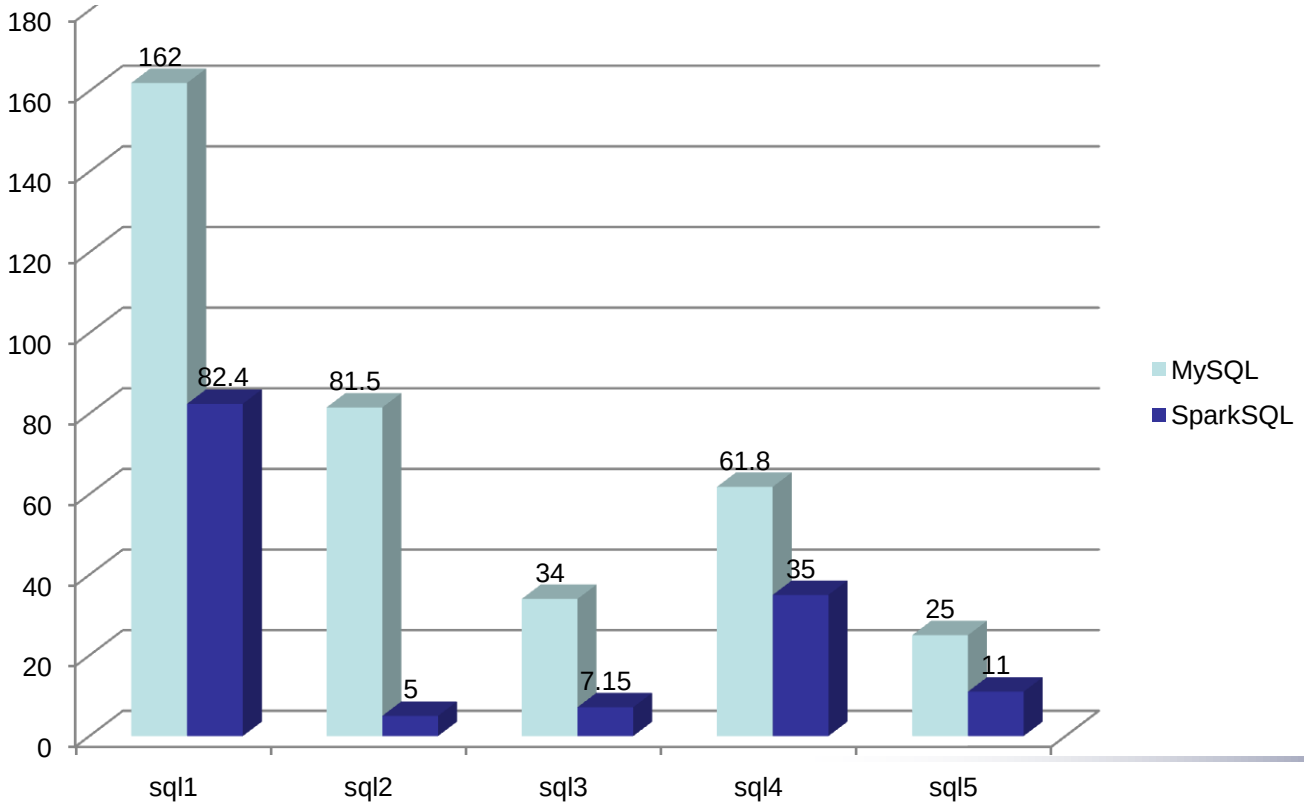
```
val df2=spark.read.parquet("PaperReferences/0511fd00-726c-4e1e-be51-3f0c59b7419e.parquet")
df2.createOrReplaceTempView("PaperReferences")

val df4=spark.read.parquet("FieldsOfStudy/57aa959f-1a88-48e4-aa58-ab6097ad5d79.parquet")
df4.createOrReplaceTempView("FieldsOfStudy")

val df5=spark.read.parquet("PaperKeywords/5349782b-f111-4491-8168-670f440fa09c.parquet")
df5.createOrReplaceTempView("PaperKeywords")

//sql2.4
val sqldf=spark.sql("SELECT FieldsOfStudyID,FieldsOfStudyName,FieldCitation from FieldsOfStudy INNER JOIN (s
sqldf.show()
```

Optimization

- number of executors & executor memory
- executor cores
- Parallelism
- Other settings for SparkSQL

Results

	MySQL	SparkSQL	SparkSQL-optimized
Sql1	162	120	82.4
Sql2	81.5	25	5
Sql3	34	18	7.1
Sql4	61.8	44	35
Sql5	25	16	11



Other work

- Use Spark together with Hive