

Elasticsearch加速SparkSQL查询

李振炜

2016-12-10

- 设计动机
- 查询详解
- 性能分析
- 后续计划

- **Hive作业迁移至Spark**

在360内部，已经基本把各个业务部门全部的Hive作业迁移到Spark，每天有几万的例行报表作业和不定的SQL分析任务稳定运行在Spark平台上

- **SQL执行效率提升至少五六倍**

从实际的迁移过程来看，SQL作业在Spark平台上效率至少有五六倍的提升，大大提升了集群的利用率

公司内部SQL作业可分为两类：


Case 1: 数据清洗和日志统计

Case 2: 交互式实时查询

- 虽然现在SQL性能有较大的提升，但是对于业务部门来说，没有最快，只有更快。
- 特别是对一些Ad-hoc、OLAP，希望查询结果快速响应

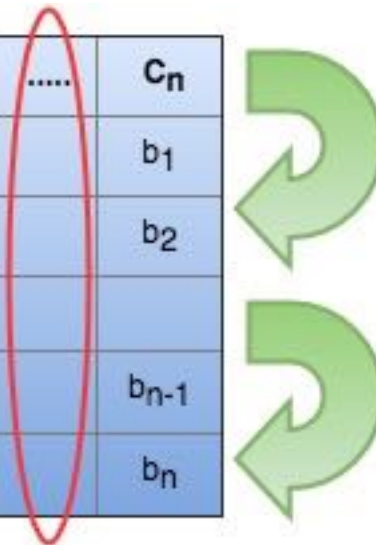
Case1

	C_1	C_n
R_1	a_1		b_1
R_2	a_2		b_2
....			
R_{n-1}	a_{n-1}		b_{n-1}
R_n	a_n		b_n



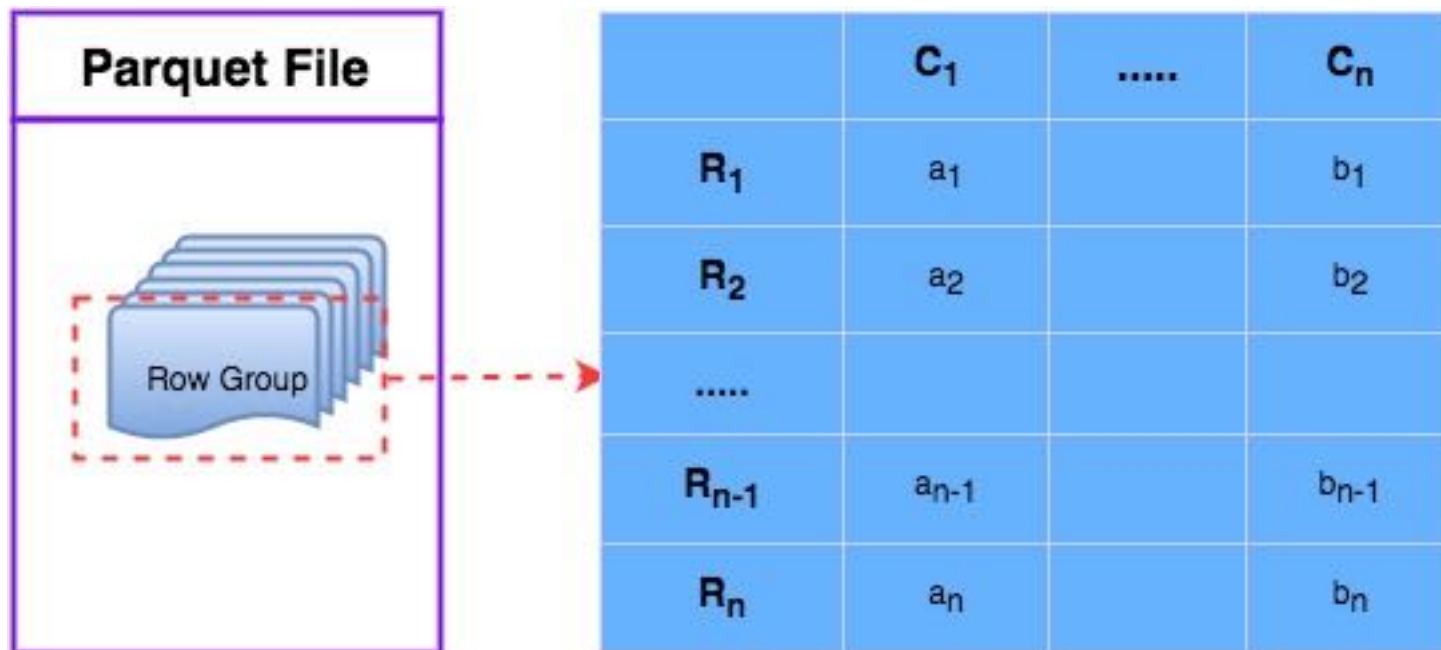
Case2

	C_1	C_n
R_1	a_1		b_1
R_2	a_2		b_2
....			
R_{n-1}	a_{n-1}		b_{n-1}
R_n	a_n		b_n



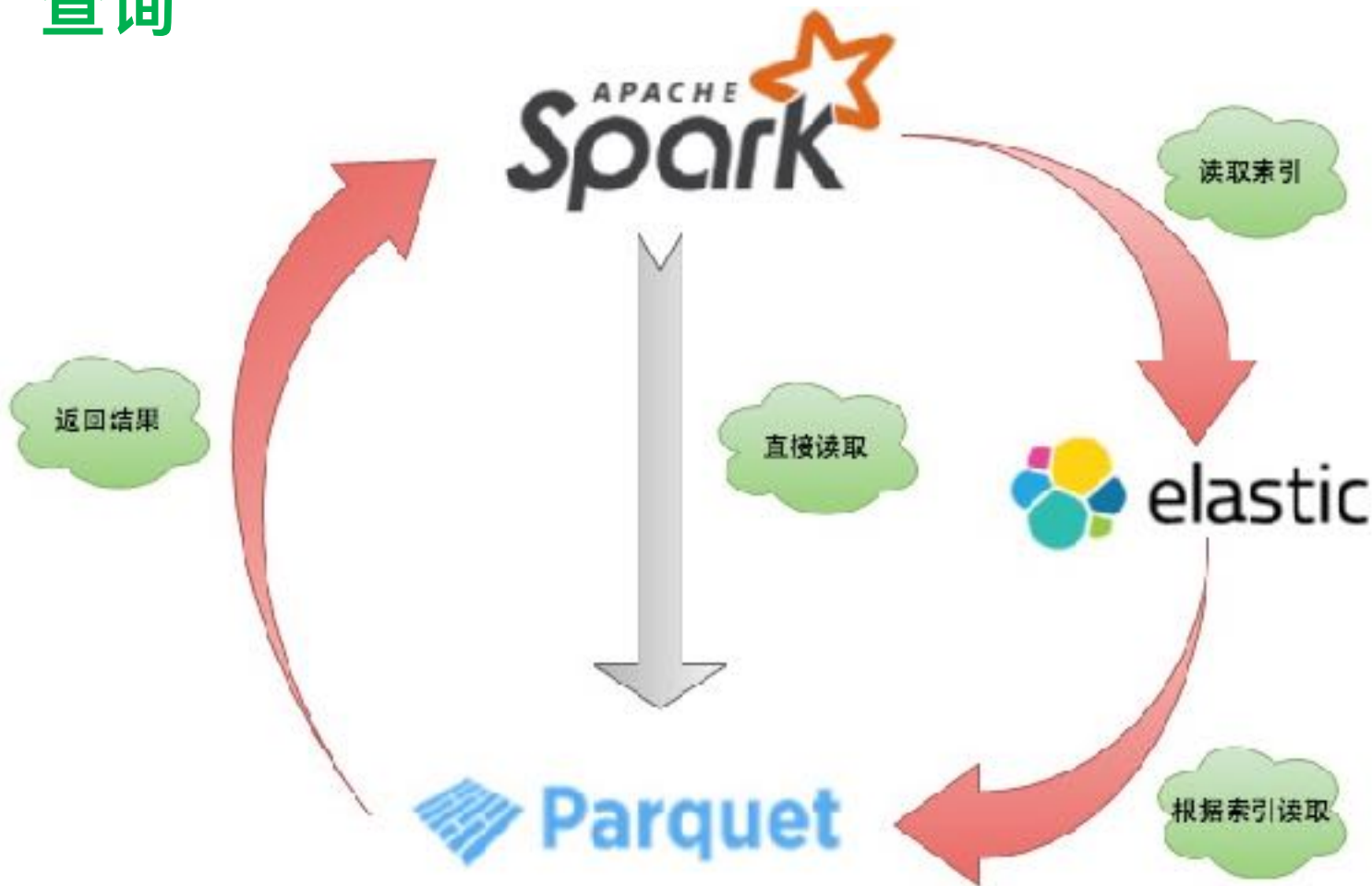
- 对于case1, Spark + ORC/Parquet方案可以满足生产需求
- 对于case2, 现有的Spark + ORC/Parquet会全表扫描数据。浪费大量的CPU和IO, 在海量数据的查询中, 很难在秒级响应
- 为了满足case2的需求, 同时保留原始数据, 我们对数据建立外部索引, 把数据的索引存储在ES中

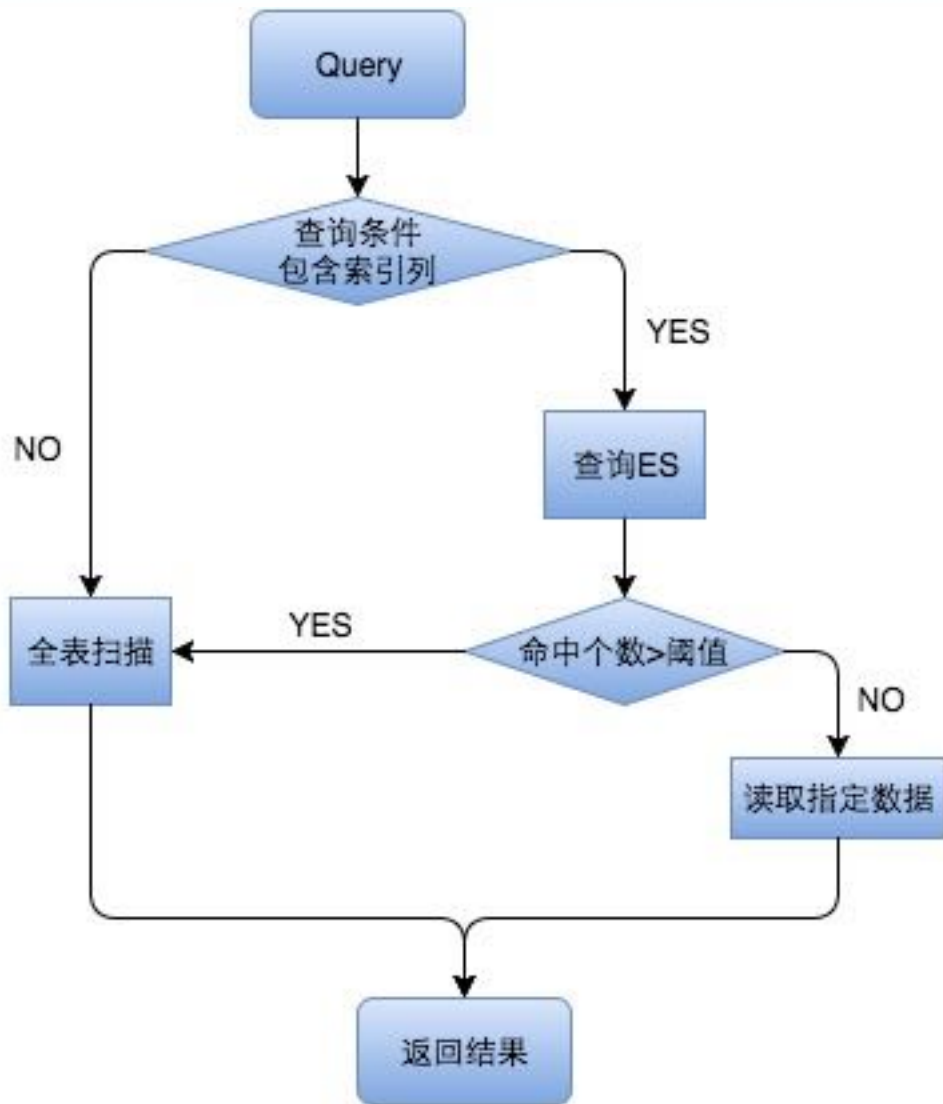
- 对数据建索引



位置	Row Group Offset	索引列 C_1	索引列 C_n
RowID	offset	a_1		b_n

- 查询



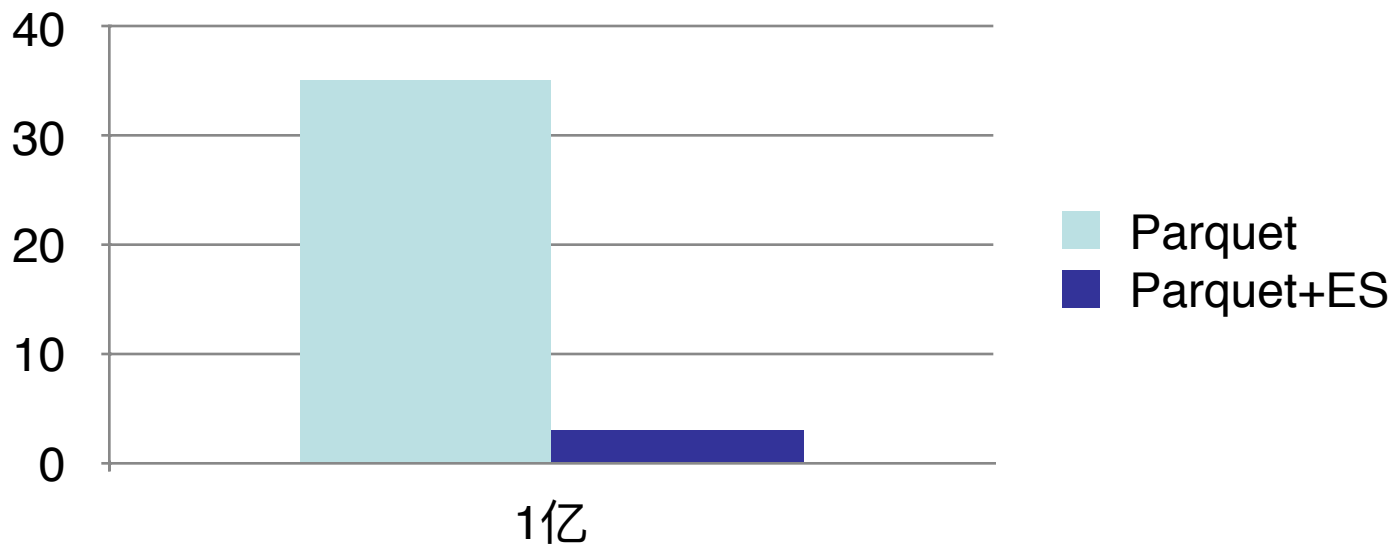


- 全表扫描过程

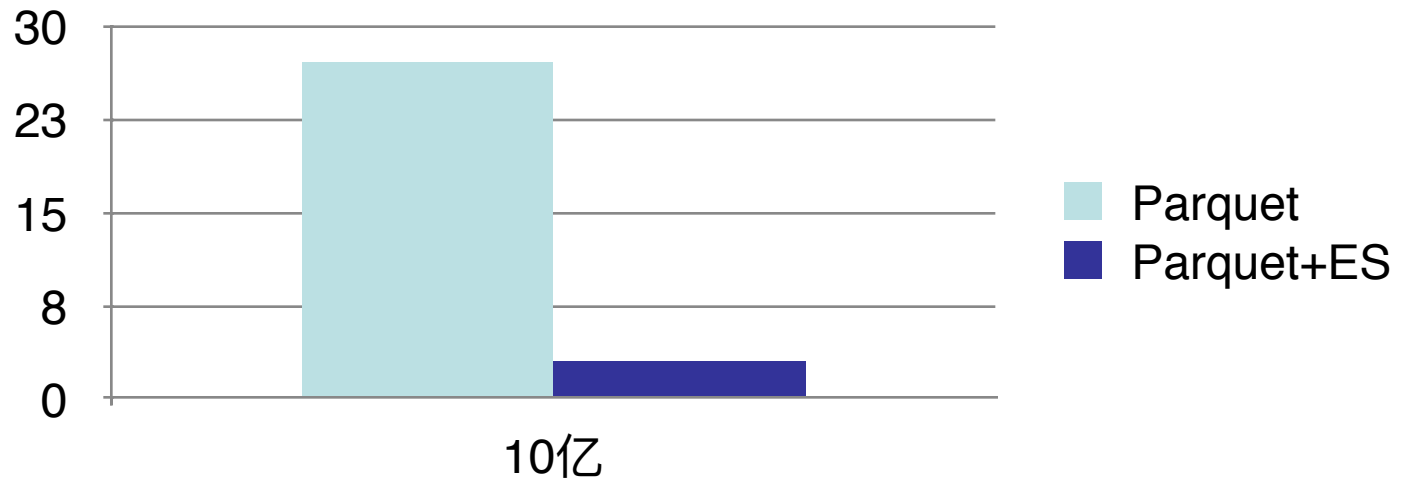
在Parquet 每个RowGroup中，全表扫描需要顺序读取数据，然后通过下推的filter筛选数据

- 加入ES外部索引之后
 - a. 可以根据索引直接读取指定的行
 - b. 实现like表达式下推
 - c. 分布式查询

```
select * from parquet_table where cint in (1,12,123,1234,12345)
```



```
select * from parquet_table where cstring like 'LRrA%'
```



- 充分利用ES的检索，丰富SparkSQL表达式下推
- 完成Spark和Kylin的整合。

谢谢

