

#### 1. hadoop中两个大表实现join的操作，简单描述。

1. Hive中可以通过分区来减少数据量；
2. 还可以通过优化HQL语句，比如只查询需要的字段，尽量避免全表、全字段查询；

#### 2.Hive中存放是什么？

表。存的是和hdfs的映射关系，hive是逻辑上的数据仓库，实际操作的都是hdfs上的文件，HQL就是用sql语法来写的mr程序。

#### 3.Hive与关系型数据库的关系？

没有关系，hive是数据仓库，不能和数据库一样进行实时的CURD操作。是一次写入多次读取的操作，可以看成是ETL工具。

#### 4.Hive中的排序关键字有哪些？

1. sort by , order by , cluster by , distribute by
2. sort by : 不是全局排序，其在数据进入reducer前完成排序
3. order by : 会对输入做全局排序，因此只有一个reducer(多个reducer无法保证全局有序).只有一个reducer,会导致当输入规模较大时，需要较长的计算时间。
4. cluster by : 当distribute by 和sort by的字段相同时，等同于cluster by.可以看做特殊的distribute + sort
5. distribute by : 按照指定的字段对数据进行划分输出到不同的reduce中

#### 5.大表和小表join

Map side join。将小表存入内存中，将小表复制多份，让每个map task内存中保留一份(比如存放到hash table中)，这样只需要扫描大表。对于大表中的每一条记录key/value，在hash table中查找是否有相同的key，如果有，则连接后输出即可。

#### 6. 数据清洗怎么做的？怎么用spark做数据清洗

1. 数据清洗的目的是为了保证数据质量，包括数据的完整性、唯一性、一致性、合法性和权威性。数据清洗的结果是对各种脏数据进行对应的处理方式，从而得到标准的、干净的、连续的数据，提供给数据统计和数据挖掘使用。
2. 解决数据的完整性问题：
  - a. (1) 通过其他信息不全；
  - b. (2) 通过前后数据不全；
  - c. (3) 如果实在无法不全，虽然可惜，但是还是要剔除掉进行统计。但是没必要删除，后续其他分析可能还需要。
3. 解决数据的唯一性问题：
  - a. (1) 根据主键进行去除，去除重复数据；
  - b. (2) 制定一系列规则，保证根据某种规则下只保存一条数据。
4. 解决数据权威性的问题：
  - a. 选择最权威的数据作为统计和挖掘。
5. 解决合法性的问题：
  - a. 设定判定规则，通过特定的规则来判断字段或者值来确定数据是否需要被清洗。

#### 7. Hadoop中二次排序怎么做？

在hadoop中一般都是按照key进行排序的，但是有时候还需要按照value进行排序。

有两种办法进行二次排序：buffer and int memory sort和value-to-key conversion。

- a. Buffer and in memory sort主要是在reduce()函数中,将每个key对应的value值保存下来，进行排序。但是缺点在于可能会出现out of memory。
- b. Value-to-key conversion主要思想是将key和value拼接成一个组合key，然后进行排序，这样reduce()函数获取的结果就实现了先按照key排序，然后按照value进行排序。需要注意的是，用户需要自己实现partitioner，以便只按照key进行数据划分。

#### 8. hadoop常见的join操作？

- (1) reduce side join: 是最简单的join操作, 主要是在reduce端进行join操作;
- (2) Map side join: 之所以存在reduce side join, 是因为在map端不能获得需要连接的全部的字段。Reduce side join比较低效, 因为shuffle传输数据需要消耗大量的性能。
- (3) Semijoin: 半连接, 对于reduce side join, 跨机器的数据传输量特别大, 成为join的一个瓶颈。如果能在map端过滤掉不会参加join的数据, 那么可以大大节省网络IO。

## 9. hive优化有哪些?

- (1) 数据存储及压缩。

针对hive中表的存储格式通常有orc和parquet, 压缩格式一般使用snappy。相比与textfile格式表, orc占有更少的存储。因为hive底层使用MR计算架构, 数据流是hdfs到磁盘再到hdfs, 而且会有很多次, 所以使用orc数据格式和snappy压缩策略可以降低IO读写, 还能降低网络传输量, 这样在一定程度上可以节省存储, 还能提升hql任务执行效率;

- (2) 通过调参优化。

并行执行, 调节parallel参数;

调节jvm参数, 重用jvm;

设置map、reduce的参数;

开启strict mode模式;

关闭推测执行设置。

- (3) 有效地减小数据集

将大表拆分成子表;

结合使用外部表和分区表。

- (4) SQL优化

大表对大表: 尽量减少数据集, 可以通过分区表, 避免扫描全表或者全字段;

大表对小表: 设置自动识别小表, 将小表放入内存中去执行。

## 10. 分析函数?

`row_number() over(partition by regionX order by nameX desc) as tn 1 93;2 90;3 90` 排名是连续的, 相同的分数会有排名先后, 前100名只有100个

`rank() over(partition by regionX order by nameX desc) as tn 1 93;2 90;2 90;4 89` 排名不是连续的, 相同的分数是同名次, 前100名只有100个

`dense_rank() over() 1 93;2 90;2 90;3 89`排名是连续的, 相同的分数是同名次, 前100名可能多于100个

## 11. 分析下hive数据倾斜问题, 有什么解决方案?

1. 倾斜原因: map输出数据按Key Hash分配到reduce中, 由于key分布不均匀、或者业务数据本身的特点。等原因造成的reduce上的数据量差异过大。

1.1) key分布不均匀

1.2) 业务数据本身的特性

1.3) SQL语句造成数据倾斜

2. 解决方案:

1> 参数调节:

```
1 hive.map.aggr=true
2 hive.groupby.skewindata=true
```

有数据倾斜的时候进行负载均衡, 当选项设定为true, 生成的查询计划会有两个MR Job。第一个MR Job中, Map的输出结果集会随机分布到Reduce中, 每个Reduce做部分聚合操作, 并输出结果, 这样处理的结果是相同Group By Key有可能被分发到不同的Reduce中, 从而达到负载均衡的目的; 第二个MR Job在根据预处理的数据结果按照Group By Key 分布到Reduce中(这个过程可以保证相同的 Group By Key 被分布到同一个Reduce中), 最后完成最终的聚合操作。

2> SQL语句调节:

1) 选用join key 分布最均匀的表作为驱动表。做好列裁剪和filter操作, 以达到两表join的时候, 数据量相对变小的效果。

2) 大小表Join: 使用map join让小的维度表(1000条以下的记录条数) 先进内存。在Map端完成Reduce。

3)大表Join大表：把空值的Key变成一个字符串加上一个随机数，把倾斜的数据分到不同的reduce上，由于null值关联不上，处理后并不影响最终的结果。

4)count distinct大量相同特殊值：count distinct时，将值为空的情况单独处理，如果是计算count distinct，可以不用处理，直接过滤，在做后结果中加1。如果还有其他计算，需要进行group by，可以先将值为空的记录单独处理，再和其他计算结果进行union。

**12. 请说明hive中 sort by， order by， cluster by， distribute by各代表什么意思？**

1. order by：会对输入做全局排序，因此只有一个reducer(多个reducer无法保证全局有序).只有一个reducer,会导致当输入规模较大时，需要较长的计算时间。
2. sort by：不是全局排序，其在数据进入reducer前完成排序
3. distribute by：按照指定的字段对数据进行划分输出到不同的reduce中
4. cluster by：除了具有distribute by 的功能外还兼具sort by 的功能

**13. 描述数据中的null,在hive底层如何存储？**

null在hive底层默认是用"\N"来存储的，所以在sqoop到mysql之前需要将null的数据加工成其他字符，否则sqoop提示错误

**14.Hive中 split、coalesce以及collect\_list函数的用法？**

1. split将字符串转化为数组
2. coalesce(T v1,T v2,...) 返回参数中的第一个非空值；如果所有值都为null,那么返回null
3. collect\_list列出该字段所有的值,不去重 select collect\_list(id) from table;

**15.如何将文件导入到hive表中？**

```
1 load data local inpath '/a.txt' overwrite into table test partition(xx='xx')
```

**16.Hive的分组方式？**

1. row\_number() 是没有重复值的排序(即使两天记录相等也是不重复的),可以利用它来实现分页
2. dense\_rank() 是连续排序,两个第二名仍然跟着第三名
3. rank() 是跳跃排序的,两个第二名下来就是第四名

**17. 请把下一语句用hive方式实现？**

```
1 SELECT a.key,a.value
2 FROM a
3 WHERE a.key not in (SELECT b.key FROM b)
```

答案：

```
1 select a.key,a.value from a where a.key not exists (select b.key from b)
```

**18.Hive内外部表的区别**

删除表是否影响外部数据 内部表影响，外部表不影响

**19. Hive如何做到权限管理**

hive下可以修改配置后创建用户管理,但是仅仅是为了防止误操而已,如果要真的为了安全操作建议使用Kerberos (和Hadoop相同)

**20.说说对桶表的理解**

对数据的细化 取模分开存储数据,提高查询效率

**21.hive有哪些保存元数据的方式，有什么特点**

- 1) Single User Mode:

默认安装hive，hive是使用derby内存数据库保存hive的元数据，这样是不可以并发调用hive的，这种模式时hive默认

的存储模式。

2) Multi User Mode:

通过网络连接到一个数据库中，是最经常使用到的模式。假设使用本机mysql服务器存储元数据。这种存储方式需要在本地运行一个mysql服务器，并作如下配置（需要将mysql的jar包拷贝到\$HIVE\_HOME/lib目录下）。

3) Remote Server Mode:

在服务器端启动一个 MetaStoreServer，客户端利用 Thrift 协议通过 MetaStoreServer 访问元数据库。

## 22.hive能建多个库吗

可以

## 23.请谈一下hive的特点是什么？hive和RDBMS有什么异同？

hive是基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供完整的sql查询功能，可以将sql语句转换为MapReduce任务进行运行。其优点是学习成本低，可以通过类SQL语句快速实现简单的MapReduce统计，不必开发专门的MapReduce应用，十分适合数据仓库的统计分析。

## 24.hive 是如何实现分区的？

1. 建表语句：

```
create table tablename (id) partitioned by (dt string)
```

2. 增加分区：

```
alter table tablename add partition (dt = '2016-03-06')
```

3. 删除分区：

```
alter table tablename drop partition (dt = '2016-03-06')
```

## 25. hive 相对于 Oracle 来说有那些优点？

- 1) 存储，hive 存储在 hdfs 上，oracle 存储在本地文件系统。
- 2) 扩展性，hive 可以扩展到数千节点，oracle 理论上只可扩展到 100 台左右。
- 3) 单表存储，数据量大 hive 可以分区分桶，oracle 数据量大只能分表。

## 26.Hive 的 sort by 和 order by 的区别！

1. order by 会对输入数据做全局排序，只有一个 reduce，数据量较大时，很慢。
2. sort by 不是全局排序，只能保证每个 reduce 有序，不能保证全局有序，需设置mapred.reduce.tasks>1。