



2017第八届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2017

# spark替代HIVE实现ETL作业

---河狸家数据小二黄伟伦  
danding@helijia.com

# 目录

#使用HIVE的一些问题？

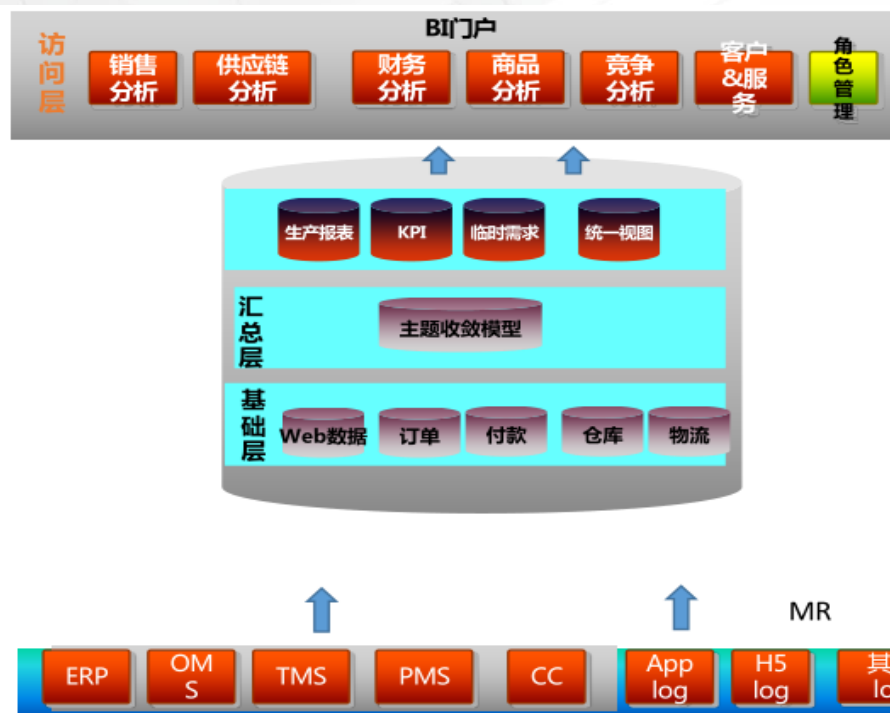
#SPARK针对ETL场景配置优化

#SPARK执行复杂SQL遇到的问题与解决

#利用SPARK应对未来ETL场景

# 使用HIVE产生的问题

## 新旧架构的冲突



# 使用HIVE产生的问题

## HIVE的性能瓶颈

HIVE依赖hadoop提供的MR引擎，执行计算逻辑。硬件需求小，吞吐量大。但是对比目前DAG调度的计算框架，很多过程无法进行优化，有着明显的性能差距。

# 使用HIVE产生的问题

## SPARK-SQL替换HIVE的优势

- 1.spark-sql 与hive非常相似，曾经SHARK演化而来。版本的迭代过程中不断的支持hive特性。支持hive-sql以及hive-server。
- 2.spark引擎本身对比hadoop-MR，在相同配置下，有明显的性能优势。
- 3.支持yarn模式方便资源的调度。
- 4.支持json， csv等原生数据格式。
- 5.项目更新进度非常快，尤其是spark-sql，社区火热。

# 使用HIVE产生的问题

## 平滑切换spark-sql到hive

- 1.部署spark接入hive源数据，将hive-site.xml移动到spark的conf目录下即可。
- 2.确认HIVE元数据的版本，如果使用server模式的话需要注意server的用户。
- 3.利用hive作业脚本，建立spark作业脚本。替换hive参数,为spark版本
- 4.hive与spark双跑，对比结果！！
- 5.切换线上作业用spark作业代替hive。

# spark-sql针对ETL的优化

## 针对大数据量计算（大量冗余，多字段）

- 1.使用列式存储格式（如：parquet）减少IO消耗。
- 2.开启spark.sql.optimizer.metadataOnly，直接只用元数据计算。
- 3.开启kryo序列化。



# spark-sql针对ETL的优化

## 针对大数据量关联查询，汇总

### 1.offheap内存的使用

在yarn模式下适当增加spark.yarn.executor.memoryOverhead。SPARK-SQL中汇总排序都是使用offheap内存。增加堆外内存上限避免被yarn杀掉。同时使用off-heap内存，理论上就有可能出现memory corruption得现象，需要留意监控。

### 2.调整shuffle参数

增加spark.shuffle.file.buffer参数，减少磁盘访问次数



# spark-sql针对ETL的优化

## 针对数据倾斜

spark-sql虽然使用spark引擎，但是同样会和mr引擎一样产生倾斜问题。不过对应的HIVE能用  
的手段spark也同样适用。

一. 调整并行，调整广播join参数

二. 利用脚本

1 预先处理倾斜的数据

2 利用随机值增加数据离散度

三. 引擎优化

1 实现了skew join

# spark-sql针对ETL的优化

## 对spark-server建立完善的监控机制

- 一. 利用spark UI 人肉监控
- 二. 调度引擎内部监控作业的执行时间。增加session关闭同时杀死相关作业的逻辑。
- 三. 利用Spark UI定时监控各个计算节点状态，重点关注失败作业数量，平均作业时间，以及GC耗时。针对有问题的节点及时杀掉重启。
- 四. spark blacklist的特性。

# SPARK执行复杂SQL遇到的问题与解决

## 单个作业处理数据量过大导致作业卡死

与hive不同，spark-sql处理reduce阶段的并行度数量需要人为通过设定spark.sql.shuffle.partitions参数决定。是个session级别的参数。可以根据实际情况针对每一个脚本设定，针对shuffle总数据量大的作业按比例增大。但是过多的并行度会带来小文件和写磁盘的压力需要考虑。

# SPARK执行复杂SQL遇到的问题与解决

大量小文件，生成过多的task，影响Driver的性能。

1.SPARK在2.0版本增加了小文件合并功能。针对会产生小文件的作业通过设置spark.sql.files.openCostInBytes参数。用一个额外的脚本，将大量小文件合并成一个。可以减少因为缓存目录给Driver带来的内存和GC压力。同时也能更高效的执行作业。

2.调整ui参数减少UI缓存对driver得内存压力

spark.ui.retainedJobs

spark.ui.retainedStages

spark.ui.retainedTasks

# SPARK执行复杂SQL遇到的问题与解决

## BUG到feature

在使用spark执行hsql的时候有时会遇到一些很硬的bug。比如create table ... 使用多个参数的函数。但是也遇到了一些情景，是因为hive和spark的细节处理不同。

多个union时经常会出现脚本异常。而且主要集中在，WidenSetOperationTypes 规则。

[SPARK-18622][SQL] Fix the datatype of the Sum aggregate function

由于这个异常我们发现当时spark版本对于小数计算会同时使用double和Decimal，可能会有精度问题。后续spark更新，全部使用Decimal来实现小数运算。

# SPARK执行复杂SQL遇到的问题与解决

```
select *  
from (  
  select 3.0 as a union all  
  select 2.0*1.0 as a  
  union all  
  select 1.0 as a  
)
```

```
-- Union  
:- Project [a#3]  
: +- Union  
:   :- Project [cast(a#0 as decimal(5,2)) AS a#3]  
:   : +- Project [3.0 AS a#0]  
:   :   +- OneRowRelation$  
:   +- Project [a#1]  
:   +- Project [checkOverflow((promote_precision(cast(2.0 as decimal(2,1))) * promote_precision(cast(1.0 as decimal(2,1))))), DecimalType(5,2)) AS a#1]  
:   +- OneRowRelation$  
+- Project [cast(a#2 as decimal(5,2)) AS a#4]  
+- Project [1.0 AS a#2]  
+- OneRowRelation$
```

在SPARK-SQL,在运算过程中会根据DecimalPrecision 规则，将DecimalType字段类型精度不断提高。有可能导致超过上限Decimal精度上限导致null。我们实际ETL场景对于精度的要求是有规范的。并不需要无限的扩展，因此增加了小数精度上限的规则。



# SPARK应对未来ETL场景

## 使用Catalyst优化器针对网络环境优化

互联网企业，发展速度飞快。跨城市，跨地区是发展第必经之路。未来的ETL的框架需要能够处理。跨网络环境的ETL流程。通过使用spark内置的Catalyst优化器，可以将网络成本加入优化逻辑。比如JOIN表的过程中对执行逻辑不同IO的情况也会不同。





# THANKS

SequeMedia  
盛拓传媒

IT168.com

ITPUB

ChinaUnix