**数据开发优化小节**

# 变更记录

|  |  |  |
| --- | --- | --- |
| 变更内容 | 变更人 | 变更时间 |
| 数据开发优化小节初稿v1.0 | 陈松玉 | 2018-09-18 |
|  |  |  |
|  |  |  |

# 阅读说明

为了让读者更好的理解文档的内容，本文在排版上会采用一些特殊字体与正文区别，下面是一些示例及其说明。

配置文件如下所示：

|  |
| --- |
| <!--Enable Hadoop Security-->  <property>  <name>hadoop.security.authentication</name>  <value>kerberos</value> <!-- A value of "simple" would disable security. -->  </property>  <property>  <name>hadoop.security.authorization</name>  <value>true</value>  </property> |

如果我们希望读者对某些配置文件参数引起重视，则会设置为粗体：

|  |
| --- |
| <!-- MapReduce JobHistory Server security configs -->  <property>  <name> mapreduce.jobhistory.keytab </name>  <value>**/opt/beh/core/hadoop/etc/hadoop/hadoop.keytab**</value>  </property>  <property>  <name> mapreduce.jobhistory.principal </name>  <value>**hadoop/\_HOST@BONC**</value>  </property> |

任何命令行的输入或者输出则如下所示：

|  |
| --- |
| [hadoop@csy001 hadoop]$ zkServer.sh start |

特殊段落的标识如下：

这里随手记下笔记和提示。

这里给出了重要注意事项。

这里给出警告。

# 概述

本文主要介绍Hive在使用中的一些优化总结，希望能够为数据开发人员提供一些有益的帮助。

# 简介

在Hadoop集群中使用MapReduce计算引擎执行任务（比如使用Hive），究竟会衍生出哪些问题呢？以下问题可能都需要我们深入思考：

* 文件存储格式是否合适
* 文件压缩格式是否合适
* 同一段HQL中产生的MapReduce是否合理，是否可以减少job数量
* Map数是否合理
* Reduce数是否合理
* 如何合理地进行小文件合并
* 如何给计算任务合理分配集群资源
* 如何把控整体流程的优化而非单一任务的优化

# 特性

Hive在运行时实际上是把HiveQL翻译成MapReduce程序进行执行，因此在考虑Hive性能优化时，就需要把HiveQL当做MapReduce程序来读，即从M/R的运行角度来考虑优化性能，从更底层思考，而不仅仅局限于逻辑代码的替换层面。

首先要了解Hive和MapReduce的特性：MapReduce将一个大型的任务，拆解成了一个个小型任务，可以比作一个巨大的货轮，出发前需要把所有的集装箱搬移到货轮上，所以MapReduce启动开销很大。

MapReduce任务分为Map阶段和Reduce阶段。

Map阶段主要就是把需要加工的数据读取到各个小任务中，主要的操作是加载数据和过滤等操作。Map阶段关注的是加载数据的文件格式和压缩格式，因为这直接涉及到加载和解析的效率。如何使得Map加载的数据能更充分利用分配给Map任务的计算资源，这是优化的一个关键。

Reduce阶段主要做的是sort及其他操作。同一key值过多，会导致数据倾斜，也就导致某几个Reduce计算的数据量相对其他Reduce会大很多。如何避免数据倾斜，同时又要避免产生过多不必要的Reduce数目，这是优化的另一个关键所在。

在引入Yarn资源管理之后，如何对每个Map任务和Reduce任务分配合适的资源，不仅影响到单个任务的计算效率，还直接关系到整体流程运行的效率。记住，整体优化比单个优化来的更有意义。流程优化的原则是，在任务一定的情况下，如何合理的利用计算资源，将流程整体时间压缩到最短。

另外，Hive和MapReduce中拥有较多在特定情况下优化的特性，如何利用好这些特性，也是优化的关键。

# Hive的特性优化

## 本地模式

### 特性介绍

当一个MapReduce任务的数据量和计算任务很小的时候，在MapReduce框架中Map任务和Reduce任务的启动过程占用了任务执行的大部分时间，真正的逻辑处理其实占用的时间很少，这样给用户的感受就是：即使很小的任务，同样需要执行较长的时间。比如对一张维表进行计算，总时间可能接近1~2分钟，这个对于用户来说，其实感受很差。

在0.7版本之后，Hive引入了本地模式，对于小任务的执行，Hive客户端不再需要到Yarn上申请Map任务和Reduce任务，只需要在本地进行Map和Reduce的执行，大大的加快了小任务的执行时间，通常可以把分钟级别任务的执行时间降低到秒级。

### 参数设置

* 如下参数设置是否开启本地模式

|  |
| --- |
| hive.exec.mode.local.auto  Default Value: **false**  Added In: Hive 0.7.0 with HIVE-1408  Lets Hive determine whether to run in local mode automatically. |

|  |
| --- |
| COO集市中该参数设置为false  hive> set hive.exec.mode.local.auto;  hive.exec.mode.local.auto=false |

### 性能对比

我们以一张小表为例，对比使用非本地模式和本地模式任务的执行时长。

* 非本地模式

|  |
| --- |
| hive> set hive.exec.mode.local.auto=**false**;  hive> select count(\*) from app.app\_org\_bd\_da ;  Query ID = mart\_coo\_20180920154012\_282baefe-811e-4732-8de1-5ea6e992fd2b  Total jobs = 1  Launching Job 1 out of 1  Number of reduce tasks determined at compile time: 1  In order to change the average load for a reducer (in bytes):  set hive.exec.reducers.bytes.per.reducer=<number>  In order to limit the maximum number of reducers:  set hive.exec.reducers.max=<number>  In order to set a constant number of reducers:  set mapreduce.job.reduces=<number>  Start submit job !  Submit job success : job\_1536659973025\_1814562  Starting Job = job\_1536659973025\_1814562, Tracking URL = http://BJLFRZ-10k-234.hadoop.jd.local:50320/proxy/application\_1536659973025\_1814562/  Kill Command = /software/servers/hadoop-2.7.1/bin/hadoop job -kill job\_1536659973025\_1814562  Hadoop job(job\_1536659973025\_1814562) information for Stage-1: number of mappers: 1; number of reducers: 1  2018-09-20 15:40:21,299 Stage-1(job\_1536659973025\_1814562) map = 0%, reduce = 0%  2018-09-20 15:40:28,638 Stage-1(job\_1536659973025\_1814562) map = 100%, reduce = 0%, Cumulative CPU 2.08 sec  2018-09-20 15:40:37,923 Stage-1(job\_1536659973025\_1814562) map = 100%, reduce = 100%, Cumulative CPU 4.33 sec  MapReduce Total cumulative CPU time: 4 seconds 330 msec  Stage-1 Elapsed : 23845 ms job\_1536659973025\_1814562  Ended Job = job\_1536659973025\_1814562  MapReduce Jobs Launched:  Stage-1: job\_1536659973025\_1814562 SUCCESS HDFS Read: 0.000 GB HDFS Write: 0.000 GB Elapsed : 23s845ms  Map: Total: 1 Success: 1 Killed: 0 Failed: 0 avgMapTime: 5s512ms  Reduce: Total: 1 Success: 1 Killed: 0 Failed: 0 avgReduceTime: 3s116ms avgShuffleTime: 3s661ms avgMergeTime: 19ms  JobHistory URL : http://BJLFRZ-10k-212.hadoop.jd.local:19888/jobhistory/job/job\_1536659973025\_1814562  Total MapReduce CPU Time Spent: 4s330ms  Total Map: 1 Total Reduce: 1  Total HDFS Read: 0.000 GB Written: 0.000 GB  OK  168  Time taken: **26.084 seconds**, Fetched: 1 row(s) |

* 本地模式

|  |
| --- |
| hive> set hive.exec.mode.local.auto=**true**;  hive> select count(\*) from app.app\_org\_bd\_da;  **Automatically selecting local only mode for query**  Query ID = mart\_coo\_20180920154151\_42ac4447-1fff-47ef-a7fd-1eba5f8a9378  Total jobs = 1  Launching Job 1 out of 1  Number of reduce tasks determined at compile time: 1  In order to change the average load for a reducer (in bytes):  set hive.exec.reducers.bytes.per.reducer=<number>  In order to limit the maximum number of reducers:  set hive.exec.reducers.max=<number>  In order to set a constant number of reducers:  set mapreduce.job.reduces=<number>  Start submit job !  Submit job success : job\_local485438294\_0001  Job running in-process (local Hadoop)  2018-09-20 15:41:54,601 Stage-1(job\_local485438294\_0001) map = 100%, reduce = 100%  Stage-1 Elapsed : 1021 ms job\_local485438294\_0001  Ended Job = job\_local485438294\_0001  MapReduce Jobs Launched:  Stage-1: HDFS Read: 0.000 GB HDFS Write: 0.134 GB SUCCESS Elapsed : 1s21ms job\_local485438294\_0001  Total MapReduce CPU Time Spent: 0ms  Total Map: -1 Total Reduce: -1  Total HDFS Read: 0.000 GB Written: 0.134 GB  OK  168  Time taken: **2.759 seconds**, Fetched: 1 row(s) |

通过以上对比发现，使用本地模式之后，对于小表的计算查询从26.084秒减少到了2.759秒。

## Mapjoin

### 特性介绍

Mapjoin是指将小表以hash map的方式装入内存，然后用key值跟大表去匹配，分解后的步骤如下所示：

* Local work:

• read records via standard table scan (including filters and projections) from source on local machine

• build hashtable in memory

• write hashtable to local disk

• upload hashtable to dfs

• add hashtable to distributed cache

* Map task

• read hashtable from local disk (distributed cache) into memory

• match records' keys against hashtable

• combine matches and write to output

* No reduce task

使用mapjoin的主要作用就是有效减少一个HQL任务中的Stage层数。同一个HQL任务中，Stage层越少代表执行的效率越快。

### 参数设置

* 如下参数设置是否开启自动转换为mapjoin

|  |
| --- |
| hive.auto.convert.join  Default Value: **false** in 0.7.0 to 0.10.0; **true** in 0.11.0 and later (HIVE-3297)  Added In: 0.7.0 with HIVE-1642  Whether Hive enables the optimization about converting common join into mapjoin based on the input file size. (Note that hive-default.xml.template incorrectly gives the default as false in Hive 0.11.0 through 0.13.1.) |

|  |
| --- |
| 在hive 0.11版本后，默认为true，不过COO集市中该参数设置为false  hive> set hive.auto.convert.join;  hive.auto.convert.join=false |

### 性能对比

我们以同一段HQL在启用mapjoin前后的执行时长，对比其性能优劣。

* 未使用mapjoin

|  |
| --- |
| hive> set hive.auto.convert.join=**false**;  hive> select count(\*)  > from (select a.delv\_center\_num  > from (select delv\_center\_num, store\_id  > from dim.dim\_wms\_store  > where store\_id not in ('2,8,9,52')  > and dim\_delv\_center\_name like '%武汉%'  > and dim\_store\_name like '%图书%') a  > inner join (select dept\_id, sid, sku\_id  > from fdm.fdm\_stock\_core\_stocknum\_chain  > where dp = 'ACTIVE') b  > on a.delv\_center\_num = b.dept\_id  > and a.store\_id = b.sid  > left join (select item\_sku\_id  > from gdm.gdm\_m03\_item\_sku\_da  > where dt = sysdate(-1)  > and sku\_valid\_flag = '1') c  > on b.sku\_id = c.item\_sku\_id) d;  WARNING: Comparing a bigint and a string may result in a loss of precision.  Query ID = mart\_coo\_20180920165836\_ee4400e4-7dde-4c52-bc2b-95fbc07fafc0  Total jobs = 3  Launching Job 1 out of 3  Number of reduce tasks not specified. Estimated from input data size: 19  In order to change the average load for a reducer (in bytes):  set hive.exec.reducers.bytes.per.reducer=<number>  In order to limit the maximum number of reducers:  set hive.exec.reducers.max=<number>  In order to set a constant number of reducers:  set mapreduce.job.reduces=<number>  Start submit job !  Submit job success : job\_1536659973025\_1823330  Starting Job = job\_1536659973025\_1823330, Tracking URL = http://BJLFRZ-10k-234.hadoop.jd.local:50320/proxy/application\_1536659973025\_1823330/  Kill Command = /software/servers/hadoop-2.7.1/bin/hadoop job -kill job\_1536659973025\_1823330  Hadoop job(job\_1536659973025\_1823330) information for Stage-1: number of mappers: 315; number of reducers: 19  2018-09-20 16:59:09,869 Stage-1(job\_1536659973025\_1823330) map = 0%, reduce = 0%  2018-09-20 16:59:52,745 Stage-1(job\_1536659973025\_1823330) map = 11%, reduce = 0%, Cumulative CPU 280.8 sec  2018-09-20 16:59:53,782 Stage-1(job\_1536659973025\_1823330) map = 20%, reduce = 0%, Cumulative CPU 526.89 sec  2018-09-20 16:59:54,814 Stage-1(job\_1536659973025\_1823330) map = 45%, reduce = 0%, Cumulative CPU 1245.52 sec  2018-09-20 16:59:55,844 Stage-1(job\_1536659973025\_1823330) map = 54%, reduce = 0%, Cumulative CPU 1543.28 sec  2018-09-20 16:59:56,876 Stage-1(job\_1536659973025\_1823330) map = 59%, reduce = 0%, Cumulative CPU 1690.53 sec  2018-09-20 16:59:57,907 Stage-1(job\_1536659973025\_1823330) map = 67%, reduce = 0%, Cumulative CPU 1919.65 sec  2018-09-20 16:59:58,937 Stage-1(job\_1536659973025\_1823330) map = 79%, reduce = 0%, Cumulative CPU 2250.19 sec  2018-09-20 16:59:59,968 Stage-1(job\_1536659973025\_1823330) map = 84%, reduce = 0%, Cumulative CPU 2417.74 sec  2018-09-20 17:00:00,998 Stage-1(job\_1536659973025\_1823330) map = 85%, reduce = 0%, Cumulative CPU 2434.74 sec  2018-09-20 17:00:02,031 Stage-1(job\_1536659973025\_1823330) map = 86%, reduce = 0%, Cumulative CPU 2474.0 sec  2018-09-20 17:00:03,062 Stage-1(job\_1536659973025\_1823330) map = 87%, reduce = 0%, Cumulative CPU 2513.85 sec  2018-09-20 17:00:04,092 Stage-1(job\_1536659973025\_1823330) map = 88%, reduce = 0%, Cumulative CPU 2538.81 sec  2018-09-20 17:00:05,122 Stage-1(job\_1536659973025\_1823330) map = 89%, reduce = 0%, Cumulative CPU 2569.8 sec  2018-09-20 17:00:07,183 Stage-1(job\_1536659973025\_1823330) map = 90%, reduce = 0%, Cumulative CPU 2594.22 sec  2018-09-20 17:00:08,214 Stage-1(job\_1536659973025\_1823330) map = 91%, reduce = 0%, Cumulative CPU 2634.66 sec  2018-09-20 17:00:12,359 Stage-1(job\_1536659973025\_1823330) map = 92%, reduce = 0%, Cumulative CPU 2659.76 sec  2018-09-20 17:00:13,389 Stage-1(job\_1536659973025\_1823330) map = 93%, reduce = 0%, Cumulative CPU 2695.16 sec  2018-09-20 17:00:14,419 Stage-1(job\_1536659973025\_1823330) map = 94%, reduce = 0%, Cumulative CPU 2704.43 sec  2018-09-20 17:00:17,512 Stage-1(job\_1536659973025\_1823330) map = 95%, reduce = 0%, Cumulative CPU 2729.33 sec  2018-09-20 17:00:18,544 Stage-1(job\_1536659973025\_1823330) map = 96%, reduce = 0%, Cumulative CPU 2771.64 sec  2018-09-20 17:00:19,582 Stage-1(job\_1536659973025\_1823330) map = 98%, reduce = 0%, Cumulative CPU 2829.85 sec  2018-09-20 17:00:20,613 Stage-1(job\_1536659973025\_1823330) map = 99%, reduce = 0%, Cumulative CPU 2875.32 sec  2018-09-20 17:00:21,644 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 0%, Cumulative CPU 2875.56 sec  2018-09-20 17:00:32,105 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 5%, Cumulative CPU 2896.66 sec  2018-09-20 17:00:34,171 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 24%, Cumulative CPU 2987.29 sec  2018-09-20 17:00:35,203 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 48%, Cumulative CPU 3085.77 sec  2018-09-20 17:00:37,269 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 51%, Cumulative CPU 3113.99 sec  2018-09-20 17:00:38,300 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 55%, Cumulative CPU 3154.4 sec  2018-09-20 17:00:40,360 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 58%, Cumulative CPU 3192.82 sec  2018-09-20 17:00:41,392 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 62%, Cumulative CPU 3230.52 sec  2018-09-20 17:00:43,458 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 65%, Cumulative CPU 3275.2 sec  2018-09-20 17:00:44,488 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 68%, Cumulative CPU 3315.23 sec  2018-09-20 17:00:46,548 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 71%, Cumulative CPU 3369.33 sec  2018-09-20 17:00:47,578 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 73%, Cumulative CPU 3401.6 sec  2018-09-20 17:00:49,646 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 77%, Cumulative CPU 3470.42 sec  2018-09-20 17:00:50,679 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 78%, Cumulative CPU 3485.53 sec  2018-09-20 17:00:51,709 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 79%, Cumulative CPU 3490.16 sec  2018-09-20 17:00:52,783 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 83%, Cumulative CPU 3553.91 sec  2018-09-20 17:00:54,843 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 84%, Cumulative CPU 3567.13 sec  2018-09-20 17:00:55,872 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 88%, Cumulative CPU 3604.43 sec  2018-09-20 17:00:58,958 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 91%, Cumulative CPU 3655.96 sec  2018-09-20 17:01:02,057 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 93%, Cumulative CPU 3689.21 sec  2018-09-20 17:01:03,087 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 94%, Cumulative CPU 3693.06 sec  2018-09-20 17:01:05,147 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 95%, Cumulative CPU 3719.15 sec  2018-09-20 17:01:06,174 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 96%, Cumulative CPU 3722.41 sec  2018-09-20 17:01:08,244 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 97%, Cumulative CPU 3742.6 sec  2018-09-20 17:01:11,331 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 98%, Cumulative CPU 3761.6 sec  2018-09-20 17:01:17,515 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 99%, Cumulative CPU 3781.43 sec  2018-09-20 17:01:30,910 Stage-1(job\_1536659973025\_1823330) map = 100%, reduce = 100%, Cumulative CPU 3801.18 sec  MapReduce Total cumulative CPU time: 0 days 1 hours 3 minutes 21 seconds 180 msec  Stage-1 Elapsed : 149709 ms job\_1536659973025\_1823330  Ended Job = job\_1536659973025\_1823330  Launching Job 2 out of 3  Number of reduce tasks not specified. Estimated from input data size: 1009  In order to change the average load for a reducer (in bytes):  set hive.exec.reducers.bytes.per.reducer=<number>  In order to limit the maximum number of reducers:  set hive.exec.reducers.max=<number>  In order to set a constant number of reducers:  set mapreduce.job.reduces=<number>  Start submit job !  Submit job success : job\_1536659973025\_1823676  Starting Job = job\_1536659973025\_1823676, Tracking URL = http://BJLFRZ-10k-234.hadoop.jd.local:50320/proxy/application\_1536659973025\_1823676/  Kill Command = /software/servers/hadoop-2.7.1/bin/hadoop job -kill job\_1536659973025\_1823676  Hadoop job(job\_1536659973025\_1823676) information for Stage-2: number of mappers: 6019; number of reducers: 1009  2018-09-20 17:01:43,336 Stage-2(job\_1536659973025\_1823676) map = 0%, reduce = 0%  2018-09-20 17:02:43,467 Stage-2(job\_1536659973025\_1823676) map = 0%, reduce = 0%  2018-09-20 17:03:43,695 Stage-2(job\_1536659973025\_1823676) map = 0%, reduce = 0%  2018-09-20 17:04:44,697 Stage-2(job\_1536659973025\_1823676) map = 0%, reduce = 0%  2018-09-20 17:05:20,287 Stage-2(job\_1536659973025\_1823676) map = 1%, reduce = 0%, Cumulative CPU 1651.69 sec  2018-09-20 17:05:23,485 Stage-2(job\_1536659973025\_1823676) map = 4%, reduce = 0%, Cumulative CPU 8861.51 sec  2018-09-20 17:05:24,551 Stage-2(job\_1536659973025\_1823676) map = 5%, reduce = 0%, Cumulative CPU 10822.17 sec  2018-09-20 17:05:25,643 Stage-2(job\_1536659973025\_1823676) map = 7%, reduce = 0%, Cumulative CPU 13945.76 sec  2018-09-20 17:05:26,748 Stage-2(job\_1536659973025\_1823676) map = 11%, reduce = 0%, Cumulative CPU 18316.26 sec  2018-09-20 17:05:27,816 Stage-2(job\_1536659973025\_1823676) map = 15%, reduce = 0%, Cumulative CPU 21153.45 sec  2018-09-20 17:05:28,884 Stage-2(job\_1536659973025\_1823676) map = 18%, reduce = 0%, Cumulative CPU 23199.48 sec  2018-09-20 17:05:29,959 Stage-2(job\_1536659973025\_1823676) map = 21%, reduce = 0%, Cumulative CPU 25842.43 sec  2018-09-20 17:05:31,035 Stage-2(job\_1536659973025\_1823676) map = 23%, reduce = 0%, Cumulative CPU 27078.08 sec  2018-09-20 17:05:32,111 Stage-2(job\_1536659973025\_1823676) map = 25%, reduce = 0%, Cumulative CPU 28052.99 sec  2018-09-20 17:05:33,206 Stage-2(job\_1536659973025\_1823676) map = 26%, reduce = 0%, Cumulative CPU 29715.53 sec  2018-09-20 17:05:34,276 Stage-2(job\_1536659973025\_1823676) map = 27%, reduce = 0%, Cumulative CPU 31556.21 sec  2018-09-20 17:05:35,337 Stage-2(job\_1536659973025\_1823676) map = 28%, reduce = 0%, Cumulative CPU 33259.59 sec  2018-09-20 17:05:36,401 Stage-2(job\_1536659973025\_1823676) map = 30%, reduce = 0%, Cumulative CPU 37137.21 sec  2018-09-20 17:05:37,463 Stage-2(job\_1536659973025\_1823676) map = 32%, reduce = 0%, Cumulative CPU 39630.22 sec  2018-09-20 17:05:38,541 Stage-2(job\_1536659973025\_1823676) map = 33%, reduce = 0%, Cumulative CPU 41773.75 sec  2018-09-20 17:05:39,624 Stage-2(job\_1536659973025\_1823676) map = 37%, reduce = 0%, Cumulative CPU 44434.43 sec  2018-09-20 17:05:40,712 Stage-2(job\_1536659973025\_1823676) map = 40%, reduce = 0%, Cumulative CPU 46098.93 sec  2018-09-20 17:05:41,792 Stage-2(job\_1536659973025\_1823676) map = 43%, reduce = 0%, Cumulative CPU 48321.98 sec  2018-09-20 17:05:42,867 Stage-2(job\_1536659973025\_1823676) map = 46%, reduce = 0%, Cumulative CPU 50890.63 sec  2018-09-20 17:05:43,929 Stage-2(job\_1536659973025\_1823676) map = 48%, reduce = 0%, Cumulative CPU 52964.49 sec  2018-09-20 17:05:45,001 Stage-2(job\_1536659973025\_1823676) map = 50%, reduce = 0%, Cumulative CPU 55161.46 sec  2018-09-20 17:05:46,068 Stage-2(job\_1536659973025\_1823676) map = 51%, reduce = 0%, Cumulative CPU 56993.73 sec  2018-09-20 17:05:47,139 Stage-2(job\_1536659973025\_1823676) map = 53%, reduce = 0%, Cumulative CPU 59327.15 sec  2018-09-20 17:05:48,219 Stage-2(job\_1536659973025\_1823676) map = 55%, reduce = 0%, Cumulative CPU 61223.37 sec  2018-09-20 17:05:49,291 Stage-2(job\_1536659973025\_1823676) map = 56%, reduce = 0%, Cumulative CPU 62438.02 sec  2018-09-20 17:05:50,357 Stage-2(job\_1536659973025\_1823676) map = 59%, reduce = 0%, Cumulative CPU 64234.63 sec  2018-09-20 17:05:51,431 Stage-2(job\_1536659973025\_1823676) map = 62%, reduce = 0%, Cumulative CPU 66587.86 sec  2018-09-20 17:05:52,497 Stage-2(job\_1536659973025\_1823676) map = 63%, reduce = 0%, Cumulative CPU 67483.49 sec  2018-09-20 17:05:53,564 Stage-2(job\_1536659973025\_1823676) map = 64%, reduce = 0%, Cumulative CPU 68294.95 sec  2018-09-20 17:05:54,634 Stage-2(job\_1536659973025\_1823676) map = 65%, reduce = 0%, Cumulative CPU 69056.08 sec  2018-09-20 17:05:55,703 Stage-2(job\_1536659973025\_1823676) map = 66%, reduce = 0%, Cumulative CPU 69557.01 sec  2018-09-20 17:05:56,765 Stage-2(job\_1536659973025\_1823676) map = 67%, reduce = 0%, Cumulative CPU 70349.94 sec  2018-09-20 17:05:57,823 Stage-2(job\_1536659973025\_1823676) map = 68%, reduce = 0%, Cumulative CPU 73268.01 sec  2018-09-20 17:05:58,883 Stage-2(job\_1536659973025\_1823676) map = 69%, reduce = 0%, Cumulative CPU 74442.43 sec  2018-09-20 17:06:01,026 Stage-2(job\_1536659973025\_1823676) map = 71%, reduce = 0%, Cumulative CPU 78121.2 sec  2018-09-20 17:06:02,089 Stage-2(job\_1536659973025\_1823676) map = 72%, reduce = 0%, Cumulative CPU 78672.94 sec  2018-09-20 17:06:03,153 Stage-2(job\_1536659973025\_1823676) map = 74%, reduce = 0%, Cumulative CPU 79863.12 sec  2018-09-20 17:06:04,217 Stage-2(job\_1536659973025\_1823676) map = 76%, reduce = 0%, Cumulative CPU 81481.09 sec  2018-09-20 17:06:05,298 Stage-2(job\_1536659973025\_1823676) map = 78%, reduce = 0%, Cumulative CPU 82017.05 sec  2018-09-20 17:06:06,369 Stage-2(job\_1536659973025\_1823676) map = 79%, reduce = 0%, Cumulative CPU 82751.61 sec  2018-09-20 17:06:07,446 Stage-2(job\_1536659973025\_1823676) map = 80%, reduce = 0%, Cumulative CPU 83101.75 sec  2018-09-20 17:06:10,746 Stage-2(job\_1536659973025\_1823676) map = 81%, reduce = 0%, Cumulative CPU 83540.5 sec  2018-09-20 17:06:18,279 Stage-2(job\_1536659973025\_1823676) map = 82%, reduce = 0%, Cumulative CPU 85218.85 sec  2018-09-20 17:06:22,564 Stage-2(job\_1536659973025\_1823676) map = 83%, reduce = 0%, Cumulative CPU 86319.31 sec  2018-09-20 17:06:25,742 Stage-2(job\_1536659973025\_1823676) map = 84%, reduce = 0%, Cumulative CPU 87073.0 sec  2018-09-20 17:06:31,073 Stage-2(job\_1536659973025\_1823676) map = 85%, reduce = 0%, Cumulative CPU 88382.41 sec  2018-09-20 17:06:35,337 Stage-2(job\_1536659973025\_1823676) map = 86%, reduce = 0%, Cumulative CPU 89336.93 sec  2018-09-20 17:06:39,596 Stage-2(job\_1536659973025\_1823676) map = 87%, reduce = 0%, Cumulative CPU 90080.04 sec  2018-09-20 17:06:52,350 Stage-2(job\_1536659973025\_1823676) map = 88%, reduce = 0%, Cumulative CPU 90914.59 sec  2018-09-20 17:07:12,507 Stage-2(job\_1536659973025\_1823676) map = 89%, reduce = 0%, Cumulative CPU 92182.98 sec  2018-09-20 17:07:22,069 Stage-2(job\_1536659973025\_1823676) map = 90%, reduce = 0%, Cumulative CPU 93411.95 sec  2018-09-20 17:07:30,599 Stage-2(job\_1536659973025\_1823676) map = 91%, reduce = 0%, Cumulative CPU 94549.51 sec  2018-09-20 17:07:40,185 Stage-2(job\_1536659973025\_1823676) map = 92%, reduce = 0%, Cumulative CPU 95469.07 sec  2018-09-20 17:07:57,260 Stage-2(job\_1536659973025\_1823676) map = 93%, reduce = 0%, Cumulative CPU 96797.87 sec  2018-09-20 17:08:04,694 Stage-2(job\_1536659973025\_1823676) map = 94%, reduce = 0%, Cumulative CPU 97740.27 sec  2018-09-20 17:08:13,226 Stage-2(job\_1536659973025\_1823676) map = 95%, reduce = 0%, Cumulative CPU 98805.75 sec  2018-09-20 17:08:24,915 Stage-2(job\_1536659973025\_1823676) map = 96%, reduce = 0%, Cumulative CPU 99728.64 sec  2018-09-20 17:08:52,557 Stage-2(job\_1536659973025\_1823676) map = 97%, reduce = 0%, Cumulative CPU 100701.67 sec  2018-09-20 17:09:17,990 Stage-2(job\_1536659973025\_1823676) map = 98%, reduce = 0%, Cumulative CPU 101822.83 sec  2018-09-20 17:10:08,757 Stage-2(job\_1536659973025\_1823676) map = 99%, reduce = 0%, Cumulative CPU 102839.08 sec  2018-09-20 17:11:09,129 Stage-2(job\_1536659973025\_1823676) map = 99%, reduce = 0%, Cumulative CPU 103832.62 sec  2018-09-20 17:12:09,552 Stage-2(job\_1536659973025\_1823676) map = 99%, reduce = 0%, Cumulative CPU 104233.03 sec  2018-09-20 17:13:10,054 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 0%, Cumulative CPU 104317.17 sec  2018-09-20 17:13:22,907 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 1%, Cumulative CPU 104754.01 sec  2018-09-20 17:13:23,971 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 34%, Cumulative CPU 118030.41 sec  2018-09-20 17:13:26,101 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 39%, Cumulative CPU 120165.8 sec  2018-09-20 17:13:27,160 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 48%, Cumulative CPU 122871.64 sec  2018-09-20 17:13:29,302 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 50%, Cumulative CPU 123432.07 sec  2018-09-20 17:13:30,365 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 58%, Cumulative CPU 126016.59 sec  2018-09-20 17:13:32,495 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 63%, Cumulative CPU 128774.98 sec  2018-09-20 17:13:33,560 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 67%, Cumulative CPU 131330.23 sec  2018-09-20 17:13:35,682 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 76%, Cumulative CPU 137515.35 sec  2018-09-20 17:13:36,748 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 78%, Cumulative CPU 139204.44 sec  2018-09-20 17:13:37,812 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 80%, Cumulative CPU 139779.05 sec  2018-09-20 17:13:38,873 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 92%, Cumulative CPU 143141.15 sec  2018-09-20 17:13:39,937 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 94%, Cumulative CPU 143730.31 sec  2018-09-20 17:13:41,000 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 96%, Cumulative CPU 144201.43 sec  2018-09-20 17:13:42,085 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 99%, Cumulative CPU 145162.98 sec  2018-09-20 17:13:47,384 Stage-2(job\_1536659973025\_1823676) map = 100%, reduce = 100%, Cumulative CPU 145512.48 sec  MapReduce Total cumulative CPU time: 1 days 16 hours 25 minutes 12 seconds 480 msec  Stage-2 Elapsed : 733084 ms job\_1536659973025\_1823676  Ended Job = job\_1536659973025\_1823676  Launching Job 3 out of 3  Number of reduce tasks determined at compile time: 1  In order to change the average load for a reducer (in bytes):  set hive.exec.reducers.bytes.per.reducer=<number>  In order to limit the maximum number of reducers:  set hive.exec.reducers.max=<number>  In order to set a constant number of reducers:  set mapreduce.job.reduces=<number>  Start submit job !  Submit job success : job\_1536659973025\_1824951  Starting Job = job\_1536659973025\_1824951, Tracking URL = http://BJLFRZ-10k-234.hadoop.jd.local:50320/proxy/application\_1536659973025\_1824951/  Kill Command = /software/servers/hadoop-2.7.1/bin/hadoop job -kill job\_1536659973025\_1824951  Hadoop job(job\_1536659973025\_1824951) information for Stage-3: number of mappers: 900; number of reducers: 1  2018-09-20 17:13:56,744 Stage-3(job\_1536659973025\_1824951) map = 0%, reduce = 0%  2018-09-20 17:14:34,688 Stage-3(job\_1536659973025\_1824951) map = 18%, reduce = 0%, Cumulative CPU 287.42 sec  2018-09-20 17:14:35,727 Stage-3(job\_1536659973025\_1824951) map = 55%, reduce = 0%, Cumulative CPU 936.3 sec  2018-09-20 17:14:36,760 Stage-3(job\_1536659973025\_1824951) map = 58%, reduce = 0%, Cumulative CPU 995.94 sec  2018-09-20 17:14:38,823 Stage-3(job\_1536659973025\_1824951) map = 59%, reduce = 0%, Cumulative CPU 1013.71 sec  2018-09-20 17:14:43,982 Stage-3(job\_1536659973025\_1824951) map = 60%, reduce = 0%, Cumulative CPU 1027.13 sec  2018-09-20 17:14:46,043 Stage-3(job\_1536659973025\_1824951) map = 63%, reduce = 0%, Cumulative CPU 1075.67 sec  2018-09-20 17:14:47,073 Stage-3(job\_1536659973025\_1824951) map = 64%, reduce = 0%, Cumulative CPU 1095.92 sec  2018-09-20 17:14:48,103 Stage-3(job\_1536659973025\_1824951) map = 87%, reduce = 0%, Cumulative CPU 1493.9 sec  2018-09-20 17:14:49,135 Stage-3(job\_1536659973025\_1824951) map = 98%, reduce = 0%, Cumulative CPU 1694.16 sec  2018-09-20 17:14:50,167 Stage-3(job\_1536659973025\_1824951) map = 99%, reduce = 0%, Cumulative CPU 1722.83 sec  2018-09-20 17:14:51,198 Stage-3(job\_1536659973025\_1824951) map = 100%, reduce = 0%, Cumulative CPU 1735.16 sec  2018-09-20 17:14:57,400 Stage-3(job\_1536659973025\_1824951) map = 100%, reduce = 100%, Cumulative CPU 1741.09 sec  MapReduce Total cumulative CPU time: 29 minutes 1 seconds 90 msec  Stage-3 Elapsed : 68145 ms job\_1536659973025\_1824951  Ended Job = job\_1536659973025\_1824951  MapReduce Jobs Launched:  Stage-1: job\_1536659973025\_1823330 SUCCESS HDFS Read: 0.822 GB HDFS Write: 0.145 GB Elapsed : 2m29s709ms  Map: Total: 315 Success: 315 Killed: 0 Failed: 0 avgMapTime: 33s951ms  Reduce: Total: 19 Success: 19 Killed: 1 Failed: 0 avgReduceTime: 19s615ms avgShuffleTime: 5s956ms avgMergeTime: 13s167ms  JobHistory URL : http://BJLFRZ-10k-212.hadoop.jd.local:19888/jobhistory/job/job\_1536659973025\_1823330  Stage-2: job\_1536659973025\_1823676 SUCCESS HDFS Read: 124.206 GB HDFS Write: 0.000 GB Elapsed : 12m13s84ms  Map: Total: 6019 Success: 6019 Killed: 30 Failed: 0 avgMapTime: 54s792ms  Reduce: Total: 1009 Success: 1009 Killed: 0 Failed: 0 avgReduceTime: 8s318ms avgShuffleTime: 6s298ms avgMergeTime: 10s185ms  JobHistory URL : http://BJLFRZ-10k-212.hadoop.jd.local:19888/jobhistory/job/job\_1536659973025\_1823676  Stage-3: job\_1536659973025\_1824951 SUCCESS HDFS Read: 0.001 GB HDFS Write: 0.000 GB Elapsed : 1m8s145ms  Map: Total: 900 Success: 900 Killed: 0 Failed: 0 avgMapTime: 5s908ms  Reduce: Total: 1 Success: 1 Killed: 0 Failed: 0 avgReduceTime: 840ms avgShuffleTime: 2s688ms avgMergeTime: 39ms  JobHistory URL : http://BJLFRZ-10k-212.hadoop.jd.local:19888/jobhistory/job/job\_1536659973025\_1824951  Total MapReduce CPU Time Spent: 1d17h57m34s750ms  Total Map: 7234 Total Reduce: 1029  Total HDFS Read: 125.030 GB Written: 0.145 GB  OK  7238890  Time taken: **984.427 seconds**, Fetched: 1 row(s) |

* 使用mapjoin

|  |
| --- |
| hive> set hive.auto.convert.join=**true**;  hive> select count(\*)  > from (select a.delv\_center\_num  > from (select delv\_center\_num, store\_id  > from dim.dim\_wms\_store  > where store\_id not in ('2,8,9,52')  > and dim\_delv\_center\_name like '%武汉%'  > and dim\_store\_name like '%图书%') a  > inner join (select dept\_id, sid, sku\_id  > from fdm.fdm\_stock\_core\_stocknum\_chain  > where dp = 'ACTIVE') b  > on a.delv\_center\_num = b.dept\_id  > and a.store\_id = b.sid  > left join (select item\_sku\_id  > from gdm.gdm\_m03\_item\_sku\_da  > where dt = sysdate(-1)  > and sku\_valid\_flag = '1') c  > on b.sku\_id = c.item\_sku\_id) d;  WARNING: Comparing a bigint and a string may result in a loss of precision.  Query ID = mart\_coo\_20180920165838\_ee9505b1-b802-4c7a-9531-7c1241c80f4c  Total jobs = 4  [2018-09-20T16:59:06.929+08:00] [WARN] hive.conf.HiveConf.initialize(HiveConf.java 2890) [main] : DEPRECATED: hive.metastore.ds.retry.\* no longer has any effect. Use hive.hmshandler.retry.\* instead  Execution log at: /tmp/mart\_coo/mart\_coo\_20180920165838\_ee9505b1-b802-4c7a-9531-7c1241c80f4c.log  2018-09-20 16:59:07 Starting to launch local task to process map join; maximum memory = 2058354688  2018-09-20 16:59:08 Dump the side-table for tag: 0 with group count: 3 into file: file:/tmp/mart\_coo/004b5491-43e6-4e89-b766-aa50797082c6/hive\_2018-09-20\_16-58-38\_800\_1262124839758227307-1/-local-10008/HashTable-Stage-8/MapJoin-mapfile10--.hashtable  2018-09-20 16:59:08 Uploaded 1 File to: file:/tmp/mart\_coo/004b5491-43e6-4e89-b766-aa50797082c6/hive\_2018-09-20\_16-58-38\_800\_1262124839758227307-1/-local-10008/HashTable-Stage-8/MapJoin-mapfile10--.hashtable (359 bytes)  2018-09-20 16:59:08 End of local task; Time Taken: 1.489 sec.  Execution completed successfully  MapredLocal task succeeded  Launching Job 1 out of 4  Number of reduce tasks is set to 0 since there's no reduce operator  Start submit job !  Submit job success : job\_1536659973025\_1823340  Starting Job = job\_1536659973025\_1823340, Tracking URL = http://BJLFRZ-10k-234.hadoop.jd.local:50320/proxy/application\_1536659973025\_1823340/  Kill Command = /software/servers/hadoop-2.7.1/bin/hadoop job -kill job\_1536659973025\_1823340  Hadoop job(job\_1536659973025\_1823340) information for Stage-8: number of mappers: 314; number of reducers: 0  2018-09-20 16:59:21,797 Stage-8(job\_1536659973025\_1823340) map = 0%, reduce = 0%  2018-09-20 17:00:07,175 Stage-8(job\_1536659973025\_1823340) map = 2%, reduce = 0%, Cumulative CPU 40.9 sec  2018-09-20 17:00:08,204 Stage-8(job\_1536659973025\_1823340) map = 4%, reduce = 0%, Cumulative CPU 91.59 sec  2018-09-20 17:00:09,235 Stage-8(job\_1536659973025\_1823340) map = 19%, reduce = 0%, Cumulative CPU 386.87 sec  2018-09-20 17:00:10,267 Stage-8(job\_1536659973025\_1823340) map = 44%, reduce = 0%, Cumulative CPU 939.29 sec  2018-09-20 17:00:11,298 Stage-8(job\_1536659973025\_1823340) map = 52%, reduce = 0%, Cumulative CPU 1099.14 sec  2018-09-20 17:00:12,328 Stage-8(job\_1536659973025\_1823340) map = 60%, reduce = 0%, Cumulative CPU 1244.72 sec  2018-09-20 17:00:13,365 Stage-8(job\_1536659973025\_1823340) map = 66%, reduce = 0%, Cumulative CPU 1367.45 sec  2018-09-20 17:00:14,396 Stage-8(job\_1536659973025\_1823340) map = 71%, reduce = 0%, Cumulative CPU 1459.94 sec  2018-09-20 17:00:15,428 Stage-8(job\_1536659973025\_1823340) map = 76%, reduce = 0%, Cumulative CPU 1558.28 sec  2018-09-20 17:00:16,463 Stage-8(job\_1536659973025\_1823340) map = 80%, reduce = 0%, Cumulative CPU 1649.94 sec  2018-09-20 17:00:17,496 Stage-8(job\_1536659973025\_1823340) map = 84%, reduce = 0%, Cumulative CPU 1725.98 sec  2018-09-20 17:00:18,537 Stage-8(job\_1536659973025\_1823340) map = 86%, reduce = 0%, Cumulative CPU 1775.53 sec  2018-09-20 17:00:19,567 Stage-8(job\_1536659973025\_1823340) map = 87%, reduce = 0%, Cumulative CPU 1785.98 sec  2018-09-20 17:00:20,598 Stage-8(job\_1536659973025\_1823340) map = 88%, reduce = 0%, Cumulative CPU 1807.01 sec  2018-09-20 17:00:21,630 Stage-8(job\_1536659973025\_1823340) map = 91%, reduce = 0%, Cumulative CPU 1847.72 sec  2018-09-20 17:00:22,661 Stage-8(job\_1536659973025\_1823340) map = 93%, reduce = 0%, Cumulative CPU 1894.25 sec  2018-09-20 17:00:24,739 Stage-8(job\_1536659973025\_1823340) map = 94%, reduce = 0%, Cumulative CPU 1912.4 sec  2018-09-20 17:00:25,770 Stage-8(job\_1536659973025\_1823340) map = 96%, reduce = 0%, Cumulative CPU 1945.09 sec  2018-09-20 17:00:26,808 Stage-8(job\_1536659973025\_1823340) map = 97%, reduce = 0%, Cumulative CPU 1969.46 sec  2018-09-20 17:00:30,937 Stage-8(job\_1536659973025\_1823340) map = 98%, reduce = 0%, Cumulative CPU 1991.59 sec  2018-09-20 17:00:31,969 Stage-8(job\_1536659973025\_1823340) map = 99%, reduce = 0%, Cumulative CPU 2004.51 sec  2018-09-20 17:00:34,067 Stage-8(job\_1536659973025\_1823340) map = 100%, reduce = 0%, Cumulative CPU 2033.66 sec  MapReduce Total cumulative CPU time: 33 minutes 53 seconds 660 msec  Stage-8 Elapsed : 83063 ms job\_1536659973025\_1823340  Ended Job = job\_1536659973025\_1823340  Stage-9 is filtered out by condition resolver.  Stage-2 is selected by condition resolver.  Launching Job 2 out of 4  Number of reduce tasks not specified. Estimated from input data size: 1009  In order to change the average load for a reducer (in bytes):  set hive.exec.reducers.bytes.per.reducer=<number>  In order to limit the maximum number of reducers:  set hive.exec.reducers.max=<number>  In order to set a constant number of reducers:  set mapreduce.job.reduces=<number>  Start submit job !  Submit job success : job\_1536659973025\_1823490  Starting Job = job\_1536659973025\_1823490, Tracking URL = http://BJLFRZ-10k-234.hadoop.jd.local:50320/proxy/application\_1536659973025\_1823490/  Kill Command = /software/servers/hadoop-2.7.1/bin/hadoop job -kill job\_1536659973025\_1823490  Hadoop job(job\_1536659973025\_1823490) information for Stage-2: number of mappers: 6303; number of reducers: 1009  2018-09-20 17:00:50,045 Stage-2(job\_1536659973025\_1823490) map = 0%, reduce = 0%  2018-09-20 17:01:50,610 Stage-2(job\_1536659973025\_1823490) map = 0%, reduce = 0%  2018-09-20 17:02:51,245 Stage-2(job\_1536659973025\_1823490) map = 0%, reduce = 0%  2018-09-20 17:03:51,444 Stage-2(job\_1536659973025\_1823490) map = 0%, reduce = 0%  2018-09-20 17:04:28,012 Stage-2(job\_1536659973025\_1823490) map = 1%, reduce = 0%, Cumulative CPU 192.22 sec  2018-09-20 17:04:30,152 Stage-2(job\_1536659973025\_1823490) map = 2%, reduce = 0%, Cumulative CPU 1949.84 sec  2018-09-20 17:04:31,215 Stage-2(job\_1536659973025\_1823490) map = 4%, reduce = 0%, Cumulative CPU 3790.45 sec  2018-09-20 17:04:32,277 Stage-2(job\_1536659973025\_1823490) map = 6%, reduce = 0%, Cumulative CPU 6405.56 sec  2018-09-20 17:04:33,343 Stage-2(job\_1536659973025\_1823490) map = 9%, reduce = 0%, Cumulative CPU 12501.79 sec  2018-09-20 17:04:34,422 Stage-2(job\_1536659973025\_1823490) map = 15%, reduce = 0%, Cumulative CPU 22750.68 sec  2018-09-20 17:04:35,511 Stage-2(job\_1536659973025\_1823490) map = 20%, reduce = 0%, Cumulative CPU 31410.59 sec  2018-09-20 17:04:36,589 Stage-2(job\_1536659973025\_1823490) map = 25%, reduce = 0%, Cumulative CPU 36356.25 sec  2018-09-20 17:04:37,677 Stage-2(job\_1536659973025\_1823490) map = 33%, reduce = 0%, Cumulative CPU 44088.1 sec  2018-09-20 17:04:38,814 Stage-2(job\_1536659973025\_1823490) map = 41%, reduce = 0%, Cumulative CPU 49740.37 sec  2018-09-20 17:04:39,905 Stage-2(job\_1536659973025\_1823490) map = 47%, reduce = 0%, Cumulative CPU 52795.76 sec  2018-09-20 17:04:40,994 Stage-2(job\_1536659973025\_1823490) map = 52%, reduce = 0%, Cumulative CPU 55664.42 sec  2018-09-20 17:04:42,085 Stage-2(job\_1536659973025\_1823490) map = 54%, reduce = 0%, Cumulative CPU 57153.61 sec  2018-09-20 17:04:43,171 Stage-2(job\_1536659973025\_1823490) map = 56%, reduce = 0%, Cumulative CPU 58307.97 sec  2018-09-20 17:04:44,246 Stage-2(job\_1536659973025\_1823490) map = 57%, reduce = 0%, Cumulative CPU 59574.78 sec  2018-09-20 17:04:45,333 Stage-2(job\_1536659973025\_1823490) map = 58%, reduce = 0%, Cumulative CPU 60039.9 sec  2018-09-20 17:04:46,407 Stage-2(job\_1536659973025\_1823490) map = 59%, reduce = 0%, Cumulative CPU 60608.56 sec  2018-09-20 17:04:48,564 Stage-2(job\_1536659973025\_1823490) map = 60%, reduce = 0%, Cumulative CPU 61434.6 sec  2018-09-20 17:04:50,706 Stage-2(job\_1536659973025\_1823490) map = 62%, reduce = 0%, Cumulative CPU 65682.1 sec  2018-09-20 17:04:51,785 Stage-2(job\_1536659973025\_1823490) map = 63%, reduce = 0%, Cumulative CPU 68784.69 sec  2018-09-20 17:04:53,966 Stage-2(job\_1536659973025\_1823490) map = 67%, reduce = 0%, Cumulative CPU 78701.69 sec  2018-09-20 17:04:55,090 Stage-2(job\_1536659973025\_1823490) map = 70%, reduce = 0%, Cumulative CPU 86362.6 sec  2018-09-20 17:04:56,184 Stage-2(job\_1536659973025\_1823490) map = 71%, reduce = 0%, Cumulative CPU 87110.82 sec  2018-09-20 17:04:57,285 Stage-2(job\_1536659973025\_1823490) map = 80%, reduce = 0%, Cumulative CPU 93942.45 sec  2018-09-20 17:04:58,378 Stage-2(job\_1536659973025\_1823490) map = 83%, reduce = 0%, Cumulative CPU 95914.07 sec  2018-09-20 17:04:59,448 Stage-2(job\_1536659973025\_1823490) map = 88%, reduce = 0%, Cumulative CPU 98835.29 sec  2018-09-20 17:05:00,540 Stage-2(job\_1536659973025\_1823490) map = 94%, reduce = 0%, Cumulative CPU 102476.7 sec  2018-09-20 17:05:01,628 Stage-2(job\_1536659973025\_1823490) map = 97%, reduce = 0%, Cumulative CPU 103366.33 sec  2018-09-20 17:05:02,710 Stage-2(job\_1536659973025\_1823490) map = 98%, reduce = 0%, Cumulative CPU 104266.24 sec  2018-09-20 17:05:03,785 Stage-2(job\_1536659973025\_1823490) map = 99%, reduce = 0%, Cumulative CPU 104730.71 sec  2018-09-20 17:05:11,415 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 0%, Cumulative CPU 105088.81 sec  2018-09-20 17:05:24,542 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 37%, Cumulative CPU 122150.52 sec  2018-09-20 17:05:25,623 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 38%, Cumulative CPU 122869.82 sec  2018-09-20 17:05:26,690 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 41%, Cumulative CPU 123937.71 sec  2018-09-20 17:05:27,756 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 49%, Cumulative CPU 126851.73 sec  2018-09-20 17:05:29,929 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 54%, Cumulative CPU 128139.37 sec  2018-09-20 17:05:30,995 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 59%, Cumulative CPU 130052.86 sec  2018-09-20 17:05:33,128 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 65%, Cumulative CPU 134625.06 sec  2018-09-20 17:05:34,204 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 68%, Cumulative CPU 136350.95 sec  2018-09-20 17:05:36,378 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 79%, Cumulative CPU 141520.75 sec  2018-09-20 17:05:37,455 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 81%, Cumulative CPU 142526.42 sec  2018-09-20 17:05:38,528 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 83%, Cumulative CPU 143122.9 sec  2018-09-20 17:05:39,590 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 93%, Cumulative CPU 146137.05 sec  2018-09-20 17:05:40,658 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 95%, Cumulative CPU 147024.02 sec  2018-09-20 17:05:41,719 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 96%, Cumulative CPU 147452.34 sec  2018-09-20 17:05:42,781 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 98%, Cumulative CPU 148034.26 sec  2018-09-20 17:05:43,845 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 99%, Cumulative CPU 148221.79 sec  2018-09-20 17:05:48,092 Stage-2(job\_1536659973025\_1823490) map = 100%, reduce = 100%, Cumulative CPU 148513.95 sec  MapReduce Total cumulative CPU time: 1 days 17 hours 15 minutes 13 seconds 950 msec  Stage-2 Elapsed : 308025 ms job\_1536659973025\_1823490  Ended Job = job\_1536659973025\_1823490  Launching Job 3 out of 4  Number of reduce tasks determined at compile time: 1  In order to change the average load for a reducer (in bytes):  set hive.exec.reducers.bytes.per.reducer=<number>  In order to limit the maximum number of reducers:  set hive.exec.reducers.max=<number>  In order to set a constant number of reducers:  set mapreduce.job.reduces=<number>  Start submit job !  Submit job success : job\_1536659973025\_1824268  Starting Job = job\_1536659973025\_1824268, Tracking URL = http://BJLFRZ-10k-234.hadoop.jd.local:50320/proxy/application\_1536659973025\_1824268/  Kill Command = /software/servers/hadoop-2.7.1/bin/hadoop job -kill job\_1536659973025\_1824268  Hadoop job(job\_1536659973025\_1824268) information for Stage-3: number of mappers: 885; number of reducers: 1  2018-09-20 17:05:57,275 Stage-3(job\_1536659973025\_1824268) map = 0%, reduce = 0%  2018-09-20 17:06:40,221 Stage-3(job\_1536659973025\_1824268) map = 10%, reduce = 0%, Cumulative CPU 163.93 sec  2018-09-20 17:06:42,286 Stage-3(job\_1536659973025\_1824268) map = 26%, reduce = 0%, Cumulative CPU 442.8 sec  2018-09-20 17:06:43,320 Stage-3(job\_1536659973025\_1824268) map = 45%, reduce = 0%, Cumulative CPU 759.88 sec  2018-09-20 17:06:44,355 Stage-3(job\_1536659973025\_1824268) map = 53%, reduce = 0%, Cumulative CPU 912.51 sec  2018-09-20 17:06:45,411 Stage-3(job\_1536659973025\_1824268) map = 58%, reduce = 0%, Cumulative CPU 995.04 sec  2018-09-20 17:06:46,446 Stage-3(job\_1536659973025\_1824268) map = 60%, reduce = 0%, Cumulative CPU 1035.96 sec  2018-09-20 17:06:48,512 Stage-3(job\_1536659973025\_1824268) map = 61%, reduce = 0%, Cumulative CPU 1055.17 sec  2018-09-20 17:06:50,582 Stage-3(job\_1536659973025\_1824268) map = 62%, reduce = 0%, Cumulative CPU 1071.87 sec  2018-09-20 17:06:55,761 Stage-3(job\_1536659973025\_1824268) map = 63%, reduce = 0%, Cumulative CPU 1092.64 sec  2018-09-20 17:06:59,889 Stage-3(job\_1536659973025\_1824268) map = 68%, reduce = 0%, Cumulative CPU 1175.89 sec  2018-09-20 17:07:01,957 Stage-3(job\_1536659973025\_1824268) map = 91%, reduce = 0%, Cumulative CPU 1569.06 sec  2018-09-20 17:07:02,992 Stage-3(job\_1536659973025\_1824268) map = 99%, reduce = 0%, Cumulative CPU 1701.58 sec  2018-09-20 17:07:05,066 Stage-3(job\_1536659973025\_1824268) map = 100%, reduce = 0%, Cumulative CPU 1723.95 sec  2018-09-20 17:07:15,441 Stage-3(job\_1536659973025\_1824268) map = 100%, reduce = 100%, Cumulative CPU 1731.45 sec  MapReduce Total cumulative CPU time: 28 minutes 51 seconds 450 msec  Stage-3 Elapsed : 85949 ms job\_1536659973025\_1824268  Ended Job = job\_1536659973025\_1824268  MapReduce Jobs Launched:  Stage-8: job\_1536659973025\_1823340 SUCCESS HDFS Read: 0.821 GB HDFS Write: 0.145 GB Elapsed : 1m23s63ms  Map: Total: 314 Success: 314 Killed: 1 Failed: 0 avgMapTime: 33s922ms  Reduce: Total: 0 Success: 0 Killed: 0 Failed: 0 avgReduceTime: 0ms avgShuffleTime: 0ms avgMergeTime: 0ms  JobHistory URL : http://BJLFRZ-10k-212.hadoop.jd.local:19888/jobhistory/job/job\_1536659973025\_1823340  Stage-2: job\_1536659973025\_1823490 SUCCESS HDFS Read: 124.213 GB HDFS Write: 0.000 GB Elapsed : 5m8s25ms  Map: Total: 6303 Success: 6303 Killed: 2 Failed: 0 avgMapTime: 18s522ms  Reduce: Total: 1009 Success: 1009 Killed: 0 Failed: 0 avgReduceTime: 7s892ms avgShuffleTime: 5s794ms avgMergeTime: 10s660ms  JobHistory URL : http://BJLFRZ-10k-212.hadoop.jd.local:19888/jobhistory/job/job\_1536659973025\_1823490  Stage-3: job\_1536659973025\_1824268 SUCCESS HDFS Read: 0.001 GB HDFS Write: 0.000 GB Elapsed : 1m25s949ms  Map: Total: 885 Success: 885 Killed: 0 Failed: 0 avgMapTime: 7s681ms  Reduce: Total: 1 Success: 1 Killed: 0 Failed: 0 avgReduceTime: 2s397ms avgShuffleTime: 5s509ms avgMergeTime: 41ms  JobHistory URL : http://BJLFRZ-10k-212.hadoop.jd.local:19888/jobhistory/job/job\_1536659973025\_1824268  Total MapReduce CPU Time Spent: 1d18h17m59s60ms  Total Map: 7502 Total Reduce: 1010  Total HDFS Read: 125.036 GB Written: 0.145 GB  OK  7238890  Time taken: **520.148 seconds**, Fetched: 1 row(s) |

通过以上对比我们可以看出对于同一段HQL，启用mapjoin后虽然job数量和stage层数增加了，但是执行时间却从984.427秒缩短到了520.148秒。究其原因可以从两个任务的执行日志中看出端倪，在未启用mapjoin的时候，虽然job数少，但是reduce数目相对较多，导致最后的计算时间有所延长，启用mapjoin之后，map数增多，reduce端的计算操作部分转移到map端，减少了reduce端的计算量，因此执行时间有所加快。

## Parallel

### 特性介绍

Parallel特性是指某些任务中的stage子任务可以并行执行，相对于串行执行stage任务来说有效的提升资源利用率。Parallel特性主要针对如下几种情况：

* 多个数据源表关联
* 插入多个目标表
* UNION ALL

### 参数设置

* 如下参数设置是否开启自动转换为Parallel

|  |
| --- |
| hive.exec.parallel  Default Value: **false**  Added In: Hive 0.5.0  Whether to execute jobs in parallel. Applies to MapReduce jobs that can run in parallel, for example jobs processing different source tables before a join. As ofHive 0.14, also applies to move tasks that can run in parallel, for example moving files to insert targets during multi-insert. |

* 如下参数设置最大并行度，默认值为8

|  |
| --- |
| hive.exec.parallel.thread.number  Default Value: **8**  Added In: Hive 0.6.0  How many jobs at most can be executed in parallel. |

|  |
| --- |
| COO集市中以上两个参数均为默认值  hive> set hive.exec.parallel;  hive.exec.parallel=false  hive> set hive.exec.parallel.thread.number;  hive.exec.parallel.thread.number=8 |

### 性能对比

我们以TPC-DS中的q11的HQL代码为例，测试使用parallel特性前后，任务执行效率是否有所不同。

HQL代码如下：

|  |
| --- |
| with year\_total as  (select c\_customer\_id customer\_id,  c\_first\_name customer\_first\_name,  c\_last\_name customer\_last\_name,  c\_preferred\_cust\_flag customer\_preferred\_cust\_flag,  c\_birth\_country customer\_birth\_country,  c\_login customer\_login,  c\_email\_address customer\_email\_address,  d\_year dyear,  sum(ss\_ext\_list\_price - ss\_ext\_discount\_amt) year\_total,  's' sale\_type  from customer, store\_sales, date\_dim  where c\_customer\_sk = ss\_customer\_sk  and ss\_sold\_date\_sk = d\_date\_sk  group by c\_customer\_id,  c\_first\_name,  c\_last\_name,  c\_preferred\_cust\_flag,  c\_birth\_country,  c\_login,  c\_email\_address,  d\_year  union all  select c\_customer\_id customer\_id,  c\_first\_name customer\_first\_name,  c\_last\_name customer\_last\_name,  c\_preferred\_cust\_flag customer\_preferred\_cust\_flag,  c\_birth\_country customer\_birth\_country,  c\_login customer\_login,  c\_email\_address customer\_email\_address,  d\_year dyear,  sum(ws\_ext\_list\_price - ws\_ext\_discount\_amt) year\_total,  'w' sale\_type  from customer, web\_sales, date\_dim  where c\_customer\_sk = ws\_bill\_customer\_sk  and ws\_sold\_date\_sk = d\_date\_sk  group by c\_customer\_id,  c\_first\_name,  c\_last\_name,  c\_preferred\_cust\_flag,  c\_birth\_country,  c\_login,  c\_email\_address,  d\_year)  select t\_s\_secyear.customer\_id,  t\_s\_secyear.customer\_first\_name,  t\_s\_secyear.customer\_last\_name,  t\_s\_secyear.customer\_email\_address  from year\_total t\_s\_firstyear,  year\_total t\_s\_secyear,  year\_total t\_w\_firstyear,  year\_total t\_w\_secyear  where t\_s\_secyear.customer\_id = t\_s\_firstyear.customer\_id  and t\_s\_firstyear.customer\_id = t\_w\_secyear.customer\_id  and t\_s\_firstyear.customer\_id = t\_w\_firstyear.customer\_id  and t\_s\_firstyear.sale\_type = 's'  and t\_w\_firstyear.sale\_type = 'w'  and t\_s\_secyear.sale\_type = 's'  and t\_w\_secyear.sale\_type = 'w'  and t\_s\_firstyear.dyear = 2001  and t\_s\_secyear.dyear = 2001 + 1  and t\_w\_firstyear.dyear = 2001  and t\_w\_secyear.dyear = 2001 + 1  and t\_s\_firstyear.year\_total > 0  and t\_w\_firstyear.year\_total > 0  and case  when t\_w\_firstyear.year\_total > 0 then  t\_w\_secyear.year\_total / t\_w\_firstyear.year\_total  else  0.0  end > case  when t\_s\_firstyear.year\_total > 0 then  t\_s\_secyear.year\_total / t\_s\_firstyear.year\_total  else  0.0  end  order by t\_s\_secyear.customer\_id,  t\_s\_secyear.customer\_first\_name,  t\_s\_secyear.customer\_last\_name,  t\_s\_secyear.customer\_email\_address limit 100; |

* 未启用parallel特性

|  |
| --- |
| ……  MapReduce Jobs Launched:  Stage-Stage-3: Map: 15 Reduce: 16 Cumulative CPU: 739.29 sec HDFS Read: 4003246772 HDFS Write: 13748588 SUCCESS  Stage-Stage-10: Map: 6 Reduce: 6 Cumulative CPU: 165.05 sec HDFS Read: 1513085545 HDFS Write: 576 SUCCESS  Stage-Stage-15: Map: 15 Reduce: 16 Cumulative CPU: 699.89 sec HDFS Read: 4003249140 HDFS Write: 26495831 SUCCESS  Stage-Stage-21: Map: 6 Reduce: 6 Cumulative CPU: 166.3 sec HDFS Read: 1513086229 HDFS Write: 576 SUCCESS  Stage-Stage-26: Map: 15 Reduce: 16 Cumulative CPU: 488.0 sec HDFS Read: 4003252055 HDFS Write: 1536 SUCCESS  Stage-Stage-32: Map: 6 Reduce: 6 Cumulative CPU: 217.88 sec HDFS Read: 1513086637 HDFS Write: 4719924 SUCCESS  Stage-Stage-37: Map: 15 Reduce: 16 Cumulative CPU: 478.58 sec HDFS Read: 4003251265 HDFS Write: 1536 SUCCESS  Stage-Stage-43: Map: 6 Reduce: 6 Cumulative CPU: 211.87 sec HDFS Read: 1513086271 HDFS Write: 4739457 SUCCESS  Stage-Stage-47: Map: 8 Cumulative CPU: 57.69 sec HDFS Read: 172769643 HDFS Write: 4372580 SUCCESS  Stage-Stage-5: Map: 1 Reduce: 1 Cumulative CPU: 3.27 sec HDFS Read: 662060 HDFS Write: 523870 SUCCESS  Total MapReduce CPU Time Spent: 53 minutes 47 seconds 820 msec  OK  Time taken: **743.388 seconds**, Fetched: 100 row(s) |

* 启用parallel特性

|  |
| --- |
| ……  **Launching Job 1 out of 14**  **Launching Job 2 out of 14**  **Launching Job 3 out of 14**  **Launching Job 4 out of 14**  **Launching Job 5 out of 14**  **Launching Job 6 out of 14**  **Launching Job 7 out of 14**  **Launching Job 8 out of 14**  ……  MapReduce Jobs Launched:  Stage-Stage-26: Map: 15 Reduce: 16 Cumulative CPU: 526.66 sec HDFS Read: 4003253807 HDFS Write: 1536 SUCCESS  Stage-Stage-37: Map: 15 Reduce: 16 Cumulative CPU: 499.5 sec HDFS Read: 4003251281 HDFS Write: 1536 SUCCESS  Stage-Stage-3: Map: 15 Reduce: 16 Cumulative CPU: 731.74 sec HDFS Read: 4003252802 HDFS Write: 13748588 SUCCESS  Stage-Stage-21: Map: 6 Reduce: 6 Cumulative CPU: 183.95 sec HDFS Read: 1513087243 HDFS Write: 576 SUCCESS  Stage-Stage-43: Map: 6 Reduce: 6 Cumulative CPU: 215.95 sec HDFS Read: 1513085941 HDFS Write: 4739457 SUCCESS  Stage-Stage-15: Map: 15 Reduce: 16 Cumulative CPU: 769.94 sec HDFS Read: 4003252628 HDFS Write: 26495831 SUCCESS  Stage-Stage-32: Map: 6 Reduce: 6 Cumulative CPU: 216.31 sec HDFS Read: 1513086979 HDFS Write: 4719924 SUCCESS  Stage-Stage-10: Map: 6 Reduce: 6 Cumulative CPU: 174.82 sec HDFS Read: 1513087231 HDFS Write: 576 SUCCESS  Stage-Stage-47: Map: 8 Cumulative CPU: 62.69 sec HDFS Read: 179400672 HDFS Write: 4451516 SUCCESS  Stage-Stage-5: Map: 1 Reduce: 1 Cumulative CPU: 3.19 sec HDFS Read: 662038 HDFS Write: 523977 SUCCESS  Total MapReduce CPU Time Spent: 56 minutes 24 seconds 750 msec  Time taken: **600.607 seconds**, Fetched: 100 row(s) |

|  |
| --- |
| 由于运行过程中日志信息过多，故只截取了其中的关键部分。 |

在启用parallel特性之后，任务执行时长从原来的743.388秒，减少到了600.607秒。在并行任务数据量较大，集群资源充足，计算较复杂的情况下，任务执行效率提升会更明显。

# 文件存储格式

截止到目前，Hive中总共支持五种文件存储格式，分别是：textfile、sequencefile、rcfile、orc、parquet。当然hive中除了这几种常见的存储格式外，还支持自定义存储格式（用户可以通过实现inoutformat和outputformat来定义输入输出格式，实际使用很少）。其中textfile为默认格式，建表时默认为这个格式，导入数据时会直接把数据文件拷贝到hdfs上不进行处理。sequencefile、rcfile、orc、parquet格式的表不能直接从本地文件导入数据，数据要先导入到textfile格式的表中，然后再从textfile表中用insert导入到sequencefile、rcfile、orc、parquet表中。

## 五种文件存储格式

### textfile

textfile是hive中默认的文件存储格式，每一行都是一条记录，每行都以换行符（\ n）结尾。数据不做压缩，磁盘开销大，数据解析开销大。

### sequencefile

sequencefile是Hadoop API 提供的一种二进制文件，它将数据以<key,value>的形式序列化到文件中。这种二进制文件内部使用Hadoop的标准的Writable接口实现序列化和反序列化。它与Hadoop API中的MapFile是互相兼容的。Hive中的SequenceFile继承自Hadoop API 的SequenceFile，不过它的key为空，使用value存放实际的值， 这样是为了避免MR 在运行map阶段的排序过程。

三种压缩选择：NONE, RECORD, BLOCK。 Record压缩率低，一般建议使用BLOCK压缩。使用时设置参数

|  |
| --- |
| set hive.exec.compress.output=true;  set io.seqfile.compression.type=BLOCK; |

### rcfile

rcfile是一种行列存储相结合的存储方式。首先，其将数据按行分块，保证同一个record在一个块上，避免读一个记录需要读取多个block。其次，块数据列式存储，有利于数据压缩和快速的列存取。

理论上具有高查询效率（但hive官方说效果不明显，只有存储上能省10%的空间，所以不好用，可以不用）。

RCFile结合行存储查询的快速和列存储节省空间的特点：

1. 同一行的数据位于同一节点，因此元组重构的开销很低；
2. 块内列存储，可以进行列维度的数据压缩，跳过不必要的列读取。

查询过程中，在IO上跳过不关心的列。实际过程是，在map阶段从远端拷贝仍然拷贝整个数据块到本地目录，也并不是真正直接跳过列，而是通过扫描每一个row group的头部定义来实现的。

但是在整个HDFS Block 级别的头部并没有定义每个列从哪个row group起始到哪个row group结束。所以在读取所有列的情况下，rcfile的性能反而没有sequencefile高。

### orc

hive/spark都支持orc这种存储格式，它存储的方式是采用数据按照行分块，每个块按照列存储，其中每个块都存储有一个索引。特点是数据压缩率非常高，是hive 0.11版里引入的新的存储格式，是对之前的rcfile存储格式的优化。orc文件格式提供了一种将数据存储在Hive表中的高效方法。这个文件系统实际上是为了克服其他Hive文件格式的限制而设计的。Hive从大型表读取，写入和处理数据时，使用orc文件可以提高性能。

### parquet

parquet是一个面向列的二进制文件格式，也是spark-sql中默认的输出格式。parquet具有较快的处理效率，虽然要牺牲点存储（相对于orc格式），但是计算速度可以很大提升，加快响应速度，提供交互式查询。对于扫描特性表格中得特定列的查询，parquet特别有用。

## 存储格式对比

|  |  |  |
| --- | --- | --- |
| 存储格式 | 存储方式 | 特点 |
| textfile | 行存储 | 存储空间消耗比较大，并且压缩的text无法分割和合并；查询的效率最低；可以直接存储，加载的速度最快 |
| sequencefile | 行存储 | 存储空间消耗最大，压缩的文件可以分割和合并，查询效率高，需要通过textfile文件转化来加载 |
| rcfile | 数据按行分块，每块按照列存储 | 存储空间小，查询效率高，需要通过textfile文件转化来加载，加载的速度最低 |
| orc | 数据按行分块，每块按照列存储 | 存储空间最小，查询的最高 ，需要通过text文件转化来加载，加载的速度最低，是rcfile的改良版 |
| parquet | 列存储 | 相对于orc，parquet压缩比较低，查询效率较低，但是parquet支持impala查询引擎 |

## 建表语句

COO集市中的表绝大部分都是以textfile和orc这两种存储格式创建的，因此我们我们也主要以这两种格式举例说明。

### textfile

|  |
| --- |
| ########################创建表############################  create table Addresses (  name string,  street string,  city string,  state string,  zip int)  PARTITIONED BY (dt string comment '日期分区')  ROW FORMAT DELIMITED FIELDS TERMINATED BY '\001'  NULL DEFINED AS ''  **stored as textfile**; **--将hive表存储格式定义为textfile**  ######################向表中插入数据###########################  **set mapreduce.map.output.compress=true;**  **set mapreduce.map.output.compress.codec=org.apache.hadoop.io.compress.LzoCodec;**  **set hive.exec.compress.intermediate=true;**  **set mapreduce.output.fileoutputformat.compress=true;**  **set mapreduce.output.fileoutputformat.compress.type=BLOCK;**  **set mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.LzoCodec;**  **set hive.exec.compress.output=true;** **--启用压缩格式**  insert overwrite table textfile\_table select \* from T\_Name; |

### orc

|  |
| --- |
| ########################创建表############################  create table Addresses (  name string,  street string,  city string,  state string,  zip int)  PARTITIONED BY (dt string comment '日期分区')  ROW FORMAT DELIMITED FIELDS TERMINATED BY '\001'  NULL DEFINED AS ''  **stored as orc** **--将hive表存储格式定义为orc**  tblproperties ('orc.compress'='SNAPPY');  ######################向表中插入数据###########################  **set mapreduce.map.output.compress=true;**  **set mapreduce.map.output.compress.codec=org.apache.hadoop.io.compress.SnappyCodec;**  **set hive.exec.compress.intermediate=true;**  **set mapreduce.output.fileoutputformat.compress=true;**  **set mapreduce.output.fileoutputformat.compress.type=BLOCK;**  **set mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.SnappyCodec;**  **set hive.exec.compress.output=true;** **--启用压缩格式**  insert overwrite table textfile\_table select \* from T\_Name; |

# 压缩格式

我们在上文中其实已经对压缩格式有所提及，那为什么我们要对文件进行压缩呢？在Hadoop中，文件需要存储、传输、读取磁盘、写入磁盘等等操作，而文件的大小，直接决定了这些这些操作的速度。除了上面提到的LZO和SNAPPY两种压缩方式外，实际上我们目前常用的压缩格式包括：SNAPPY，GZIP，BZIP2，LZO等。但是在实际使用场景中其实非常严格，切勿乱使用。

* BZIP2

特殊情况下考虑BZIP2，对文件压缩率有很高的要求，主要用来应对文件网络传输。目前主要用途是生成上报文件，低带宽下进行传输使用，其他地方不用考虑。大数据平台上几乎无用武之地。因为压缩率的问题可以通过文件存储格式来解决（无论是使用orc还是parquet存储，都能将文件的空间大小减少到一个合适的范围）。

* GZIP

GZIP的场景主要是用在文件入库（后面我们会再重点讲解入库的概念）阶段，处理GZIP的文件时由于gzip命令较为常用，并且gzip文件可以直接多个进行叠加且不影响读取，所以可用在plumber直通车数据接入时做小文件合并使用（后面我们再详细阐述）。

* SNAPPY

SNAPPY的压缩和解压缩效率在所有场景下基本都是最高的，不会过多的占用有限的CPU资源。同时SNAPPY也不会影响Map的分片和Reduce合并，能够有效避免资源利用不足和过多小文件。实际上SNAPPY的压缩比要比其他压缩格式要小，但是压缩起到重要作用的地方是MapReduce中间各Stage层的磁盘读写。文件系统上长期存储的文件，可以通过设置文件格式为ORC或者PARQUET达到很好的空间利用率。

## 入库

**入库：数据由本地文件系统上传到HDFS，或者通过hive load命令将本地文件数据加载到HDFS上。**

上传文件格式需与Hive中表的存储定义格式匹配。

入库文件的压缩格式要求并不是特别严格，主要就是压缩之后上传文件的大小是否合适。单个文件太大会导致单个Map处理时间过长，太小又会导致占用太多Container，浪费过多计算资源，Map启动的代价过高，过多的Map导致任务执行时间都浪费在Container的申请和Map的启动上了。

由于GZIP文件在Map处理时并不会自动切分，也就是说一个GZIP文件对应一个Map(如果你处理单个100G的GZIP文件，那么一个4G内存的Map处理一个100G的压缩文件，效率之低突破天际)。

## 加工

**加工：来源表和目标表的存储都在Hive或者HDFS。**

### Map阶段

在5.3章节中我们已经对一些压缩参数有过简单的介绍了，在此我们再对这些参数的含义做出详细解释。

通过以下参数设置是否对map任务输出进行压缩，以及相应的压缩格式。COO集市设置了对map阶段的输出结果进行压缩，且使用Lzo压缩。

|  |
| --- |
| <!--##控制hive的查询结果输出是否进行压缩 -->  <property>  <name>hive.exec.compress.outpu</name>  <value>**true**</value>  </property>  <!--##map端计算结果是否输出压缩 -->  <property>  <name>mapreduce.map.output.compress</name>  <value>**true**</value>  </property>  <!--##设置map端输出结果的压缩方式，COO集市设置的是LZO压缩-->  <property>  <name>mapreduce.map.output.compress.codec</name>  <value>**com.hadoop.compression.lzo.LzoCodec**</value>  </property> |

|  |
| --- |
| com.hadoop.compression.lzo.LzoCodec该参数值在hadoop2.0 以上版本中已更新为org.apache.hadoop.io.compress.LzoCodec这种写法。但是COO集市目前还是沿用的老式写法。 |

### Shuffle阶段

Shuffle其实是描述着数据从map task输出到reduce task输入的这段过程。Map任务执行完成之后需要将结果数据完整地拉取到reduce端，因此启用中间数据压缩就是一件很有必要的事情。在跨节点拉取数据时，这就可以尽可能地减少对带宽的不必要消耗，也就是减少磁盘IO对任务执行的影响。以下参数设置是否启用中间数据压缩

|  |
| --- |
| <property>  <name>hive.exec.compress.intermediate</name>  <value>**true**</value>  </property> |

|  |
| --- |
| COO集市中hive.exec.compress.intermediate该参数值为false |

### Reduce阶段

通过以下参数设置是否对reduce任务输出进行压缩，以及相应的压缩格式。COO集市设置了对map阶段的输出结果进行压缩，且使用Lzo压缩。

|  |
| --- |
| <!--##控制hive的查询结果输出是否进行压缩 -->  <property>  <name>hive.exec.compress.outpu</name>  <value>**true**</value>  </property>  <!--##reduce端计算结果是否输出压缩 -->  <property>  <name>mapreduce.output.fileoutputformat</name>  <value>**true**</value>  </property>  <!--##控制输出使用的压缩方式，可配置值有：NONE、RECORD、BLOCK。默认值是RECORD（COO集市也是设置为RECORD），即针对每条记录进行压縮。如果将其改为BLOCK,将针对一组记录进行压缩，这是推荐的压縮策略，因为它的压缩效率更高。-->  <property>  <name>mapreduce.output.fileoutputformat.compress.type</name>  <value>**RECORD**</value>  </property>  <!--##设置reduce端输出结果的压缩方式，COO集市设置的是默认值DefaultCode-->  <property>  <name>mapreduce.output.fileoutputformat.compress.codec</name>  <value>**org.apache.hadoop.io.compress.DefaultCodec**</value>  </property> |

## 出库

**出库：将HDFS上Hive中的数据通过insert方式导出至本地文件系统或者直接将HDFS文件get到本地文件系统。**

* 使用Insert的方式进行数据导出，则会启动MapReduce任务，这时候可以忽略来源表的存储格式和压缩格式，但是可以设置输出文件的压缩格式，任务个数由表中文件的个数决定（也就是Map数目）。
* 使用hadoop fs -get方式来将HDFS文件拖拽到本地时，由于命令只是单纯的拷贝，所以来源表中的存储格式和压缩格式不会改变。如果你需要get出来的文件是TEXTFILE文件那么需要在来源表创建定义时进行设置。如果你需要get出来的文件是GZIP压缩，那么需要指定Hive输出为压缩且设置压缩编码为GZIP。

# 小文件合并

当HDFS上存在大量小文件，且文件数目非常多的时候，容易使得HDFS上的Block块数目相对较多，给NameNode带来较大压力，影响HDFS文件读写性能，而且导致MapReduce任务执行效率低下。因此在MapReduce任务的Map阶段和Reduce阶段都要进行文件合并，一方面可以更加合理地利用集群资源，另一方面也能提高任务的执行效率。

|  |
| --- |
| 小文件是指文件size小于HDFS上block块大小的文件。这样的文件会给hadoop的扩展性和性能带来严重问题。首先，在HDFS中，任何block块，文件或者目录在内存中均以对象的形式存储，每个对象约占150byte，如果有1000 0000个小文件，每个文件占用一个block块，则NameNode大约需要2G空间。如果存储1亿个文件，则NameNode需要20G空间。这样NameNode内存容量严重制约了集群的扩展。  其次，访问大量小文件速度远远小于访问几个大文件。HDFS最初是为流式访问大文件开发的，如果访问大量小文件，需要不断的从一个DataNode跳到另一个DataNode，严重影响性能。最后，处理大量小文件速度远远小于处理同等大小的大文件的速度。每一个小文件要占用一个slot，而task启动将耗费大量时间甚至大部分时间都耗费在启动task和释放task上。 |

## 入库合并

比如我们有500G的GZIP文件需要入库（数据加载到HDFS上）。以下面几种方式来处理：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 单个文件大小 | 文件个数 | Map个数 | 执行时间 | 问题及说明 |
| 50G | 10 | 10 | 20X | 启动Map太少，单个Map执行时间过长，整体资源利用很低，任务执行时间过长。 |
| 5G | 100 | 100 | 3X | 启动Map较少，单个Map执行时间较长，整体资源利用率低，任务执行时间很长。 |
| 1G | 500 | 500 | X | 启动Map合适，单个Map在4G内存的Map下执行合适，整体资源利用率较好，任务执行时间可接受。 |
| 250M | 2000 | 2000 | X | 启动Map较多，单个Map在1~2G内存的Map下执行时间合适，整体资源利用率高，任务执行时间可接受。 |
| 20M | 25000 | 25000 | 20X | 启动Map太多，单个Map执行时间短，启动过多Map导致时间都浪费在container申请和Map任务启动上。资源利用率异常高，导致其他任务无法获取container，任务执行时间过长。 |

## Map合并

Map阶段如果使用合适的文件格式和压缩方式，input读取的都是小文件，使用以下参数就会自动进行文件合并，合并后一般会接近Block块的大小。

|  |
| --- |
| -----在map-only的任务结束之后合并小文件，如果设置为true，会启动一个merge job来合并map端输出结果。默认值为true  set hive.merge.mapfiles = true;  -----执行map前进行小文件合并  set hive.input.format=org.apache.hadoop.hive.ql.io.CombineHiveInputFormat;  -----设置每个map处理的最大文件大小，单位为byte  set mapred.max.split.size=256000000;  -----设置每个DataNode节点中可处理的最小的文件大小，默认值为1byte  set mapred.min.split.size.per.node=100000000;  -----设置每个机架中可处理的最小文件的大小，默认值为1byte  set mapred.min.split.size.per.rack=100000000;  -----当输入文件包括大量小文件或者文件目录，造成Splitmetainfo文件超过默认上限100000000 。设置为-1为取消该默认设置  set mapreduce.jobtracker.split.metainfo.maxsize=-1; |

## Reduce合并

Reduce阶段如果不做小文件合并，将有如下影响：

1. 大量小文件给NameNode造成巨大压力；
2. Reduce任务过多造成集群资源浪费；
3. 文件作为来源表继续向上加工时可能出现过多map，造成集群资源浪费。

使用如下参数对Reduce阶段的数据进行小文件合并

|  |
| --- |
| ------在MapReduce任务结束之后合并小文件，默认值为false  set hive.merge.mapredfiles = true;  ------当输出文件的平均大小小于该值时，启动一个独立的MapReduce任务进行文件合并。前提是map-only任务的hive.merge.mapfiles值必须设置为true，MapReduce任务的hive.merge.mapredfiles值必须设置为true。默认值为16000000 Byte  set hive.merge.smallfiles.avgsize=16000000;  ------合并文件的大小，该值必须要大于hive.merge.smallfiles.avgsize的 值。默认值为256000000 Byte  set hive.merge.size.per.task=256000000;  ------设置每个reduce处理的数据量，默认值为1000000000 Byte≈1GB  set hive.exec.reducers.bytes.per.reducer=1000000000;  ------设置reduce的数量。可以通过控制Reduce数目来达到合并小文件的作用，如果生成了100个几M的文件，可以将reduce数目设置为1，这样最后会生成一个几百M的文件，对于整体流程来说有极大的优化作用。默认值为1，如果设置为-1，Hive将自行决定需要分配的reduce数量  set mapred.reduce.tasks=n; (n为整数) |

## HiveTask及合并小文件工具