Hive 案例演示

第1页共18页 出自石山园,博客地址: http://www.cnblogs.com/shishanyuan

目 录

1	HIVE操作演示	3
	1.1 内部表	3
	1.1.1 创建表并加载数据	
	1.1.2 查询行数	4
	1.1.3 包含baidu的数据	5
	1.1.4 查询结果排名第1,点击次序排第2,其中URL包含baidu的数据	5
	1.2 外部表	6
	1.2.1 创建表关联数据	6
	1.2.2 查询行数	7
	1.2.3 显示前10 行	8
	1.2.4 查询结果排名第1,点击次序排第2 的数据	8
	1.2.5 查询次数排行榜	9
2	交易数据演示	
	2.1 准备数据	10
	2.1.1 上传数据	
	2.1.2 在Hive创建数据库和表	
	2.1.3 导入数据	
	2.2 计算所有订单每年的总金额	
	2.2.1 算法分析	
	2.2.2 执行HSQL语句	
	2.2.3 查看结果	
	2.3 计算所有订单每年最大金额订单的销售额	13
	2.3.1 算法分析	
	2.3.2 执行HSQL语句	
	2.3.3 查看结果	
	2.4 计算其他金额	15
	2.4.1 所有订单中季度销售额前10 位	
	2.4.2 列出销售金额在100000 以上的单据	
	2.4.3 所有订单中每年最畅销货品	

Hive 案例演示

1 Hive 操作演示

1.1 内部表

1.1.1 创建表并加载数据

第一步 启动 HDFS、YARN 和 Hive, 启动完毕后创建 Hive 数据库

hive>create database hive;

hive>show databases;

hive>use hive;

I hadoop1 | hadoop2 | hadoop3
hive> show databases;
OK
default
hive
Time taken: 0.059 seconds, Fetched: 2 row(s)
hive> use hive;
OK
Time taken: 0.212 seconds

第二步 创建内部表

由于 Hive 使用了类似 SQL 的语法,所以创建内部表的语句相对 SQL 只增加了行和字段分隔符。

23

.

hive>CREATE TABLE SOGOUQ2(DT STRING,WEBSESSION STRING,WORD STRING,S_SEQ INT,C_SEQ INT,WEBSITE STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' ;



第三步 加载数据

数据文件可以从 HDFS 或者本地操作系统加载到表中,如果加载 HDFS 文件使用 LOAD DATA INPATH,而加载本地操作系统文件使用 LOAD DATA LOCAL INPATH 命令。HIVE 表保存的默 认路径在\${HIVE_HOME}/conf/hive-site.xml 配置文件的 hive.metastore.warehouse.dir 属性 指定,当创建表时会在 hive.metastore.warehouse.dir 指向的目录下以表名创建一个文件夹, 在本演示中表默认指向的是/user/hive/warehouse。 数据文件在本地操作系统将复制到表对应的目录中,而数据文件在 HDFS 中,数据文件将移动 到表对应的目录中,原来的路径将不存在该文件。在这里使用《Spark 编程模型(上)--概念及 Shell 试验》中在本地操作系统中的搜狗日志数据文件:

hive>LOAD DATA LOCAL INPATH '/home/hadoop/upload/sogou/SogouQ2.txt' INTO TABLE SOGOUQ2;

hadoop1 hadoop2 hadoop3
hive> LOAD DATA LOCAL INPATH '/home/hadoop/upload/sogou/SogouQ2.txt' INTO TABLE SOGOUQ2;
Copying data from file:/home/hadoop/upload/sogou/SogouQ2.txt
Copying file: file:/home/hadoop/upload/sogou/SogouQ2.txt
Loading data to table hive.sogouq2
Table hive.sogouq2 stats: [numFiles=1, numRows=0, totalSize=217441417, rawDataSize=0]
OK
Time taken: 54.983 seconds

在/user/hive/warehouse/hive.db/sogouq2 目录下,可以看到 SougouQ2.txt 数据文件:

← → C 🗋 hadoop3:50075/browseDirectory.jsp?dir=%2Fuser%2Fhive%2Fwarehouse%2Fhive.db%2Fsoge

Contents of directory <u>/user/hive/warehouse/hive.db</u>/sogouq2

Goto	oto : /user/hive/warehodse/hive.db go									
<u>Go to</u>	Go to parent directory									
Name		Туре	Size	Replication	Block Si:	e Modification Time	Permission	0wner	Group	
Sogo	uQ2. txt	file	207.37 MB	2	128 MB	2015-07-23 16:01	rw-rr	hadoop	supergroup	

1.1.2 查询行数

可以用 count 关键字查询 SogouQ2.txt 数据行数,查询时会启动 MapReduce 进行计算,Map 的个数一般和数据分片个数对应,在本查询中有 2 个 Map 任务(数据文件有 2 个 Block), 1 个 Reduce 任务。

hive>select count(*) from SOGOUQ2;

```
I hadoop1 hadoop2 hadoop3
Nive> select count(*) from SOGOUQ2;
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 set a constant number of reducers:
set hive.exec.reduces=<number>
In order to set a constant number of reducers:
set mapreduce.job.reduces=<number>
Starting job = job_1437639210711_0001, Tracking URL = http://hadoop1:8088/proxy/application_1437639
210711_0001/
Kill command = /app/hadoop/hadoop-2.2.0/bin/hadoop job -kill job_1437639210711_0001
Hadoop job information for stage-1: number of mappers: 2; number of reducers: 1
2015-07-23 16:16:50,771 stage-1 map = 0%, reduce = 0%
2015-07-23 16:17:50,916 stage-1 map = 0%, reduce = 0%
2015-07-23 16:18:05,526 stage-1 map = 100%, reduce = 0%, cumulative CPU 48.63 sec
2015-07-23 16:18:05,526 stage-1 map = 100%, reduce = 100%, cumulative CPU 48.63 sec
2015-07-23 16:18:05,526 stage-1 map = 100%, reduce = 100%, cumulative CPU 52.6 sec
MapReduce Total cumulative CPU time: 52 seconds 600 msec
Ended job = job_1437639210711_0001
MapReduce Job Launched:
Job 0: Map: 2 Reduce: 1 Cumulative CPU: 52.6 sec HDFS Read: 217572931 HDFS Write: 8 SUCCESS
Total MapReduce CPU Time Spent: 52 seconds 600 msec
GK
200000
Time taken: 113.307 seconds, Fetched: 1 row(s)
```

第 4 页 共 18 页 出自石山园,博客地址: http://www.cnblogs.com/shishanyuan

← → C ^a hadoop1:	8088/cluster/a	pps											5 ☆ 🍇 🗉
Ene		D				All A	oplic	ation	S			Logge	d in as: dr.who
- Cluster	Cluster Mer	trics											
<u>About</u> Nodes	Apps Submitted	Apps Pending	Apps Running	A1 Comp	ops Contain leted Runni	ners Memory ng Used	Memory Total	Memory Reserved	Active Nodes	Decommissi Nodes	ioned Lost Nodes	Unhealthy Nodes	Rebooted Nodes
Applications	1	0	0	1	0	0 B	24 GB	0 B	3	<u>0</u>	Q	0	0
NEW SAVING	Show 20 🔻	entries									Search	1:	
SUBMITTED ACCEPTED		ID	Ŧ	User \$	Name 🗘	Application Type \$	Queue \$	StartTime ≎	FinishTime \$	State ≎	FinalStatus \$	Progress ≎	Tracking UI \$
RUNNING REMOVING FINISHING FINISHED FAILED	application	1437639210	0711 0001	hadoop	select count(*) from SOGOUQ2(Stage- 1)	MAPRED UCE	default	Thu, 23 Jul 2015 08:16:29 GMT	Thu, 23 Jul 2015 08:18:16 GMT	FINISHED	SUCCEEDED		<u>History</u>
KILLED Scheduler	Showing 1 to	olofler	ntries								First	Previous 1	Next Last

1.1.3 包含 baidu 的数据

可以用 like 关键字进行模糊查询, Map 的个数一般和数据分片个数对应。

hive>select count(*) from SOGOUQ2 where WEBSITE like '%baidu%';



1.1.4 查询结果排名第1,点击次序排第2,其中 URL 包含 baidu 的数据

hive>select count() from SOGOUQ2 where S_SEQ=1 and C_SEQ=2 and WEBSITE like '%baidu%';*

I hadoop1 hadoop2 hadoop3
Nive> select count(*) from SOGOUQ2 where 5_SEQ=1 and C_SEQ=2 and WEBSITE like '%baidu%';
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>
Starting Job = job_1437639210711_0003, Tracking URL = http://hadoop1:8088/proxy/application_14376392
10711_0003/
Kill Command = /app/hadoop/hadoop-2.2.0/bin/hadoop job -kill job_1437639210711_0003
Hadoop job information for stage-1: number of mappers: 2; number of reducers: 1
2015-07-23 16:25:513,886 stage-1 map = 00%, reduce = 0%, Cumulative CPU 15.96 sec
2015-07-23 16:26:16,202 stage-1 map = 100%, reduce = 100%, Cumulative CPU 18.85 sec
MapReduce Total cumulative CPU time: 18 seconds 850 msec
Finde Job = job_1437639210711_0003
MapReduce Jobs Launched:
Job 0: Map: 2 Reduce: 1 Cumulative CPU: 18.85 sec HDFS Read: 217572931 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 18 seconds 850 msec
OK
S024
Time taken: 83.077 seconds, Fetched: 1 row(s)

Job Name: select count(*) from SOGOUQ2 wh...'%baidu% (Stage-1) State: RUNNING Uberized: false Started: Thu Jul 23 16:25:10 CST 2015 Elapsed: 1mins, 2sec

ApplicationMaster			
Attempt Number	Start Time	Node	Logs
1	Thu Jul 23 16:25:03 CST 2015	hadoop1:8042	<u>logs</u>

Task Type	Task Type Progress		Total	Pending	R	unning	Complete
Map			2	0	0		2
Reduce			1	0	1		0
Attempt Type	New	Ru	nning	Failed	Kille	d	Successful
Maps	<u>0</u>	<u>0</u>	(0	<u>0</u>	<u>2</u>	
Reduces	<u>0</u>	1	(0	0	0	

1.2 外部表

1.2.1 创建表关联数据

第一步 在 HDFS 创建外部表存放数据目录

\$hadoop fs -mkdir -p /class5/sogouq1

\$hadoop fs -ls /class5

```
I hadoop1 | hadoop2 | hadoop3
[hadoop@hadoop1 ~]$
[hadoop@hadoop1 ~]$ hadoop fs -mkdir -p /class5/sogouq1
[hadoop@hadoop1 ~]$ hadoop fs -ls /class5
Found 1 items
drwxr-xr-x - hadoop supergroup 0 2015-07-23 16:46 /class5/sogouq1
[hadoop@hadoop1 ~]$
```

第二步 在 Hive 创建外部表,指定表存放目录

hive>CREATE EXTERNAL TABLE SOGOUQ1(DT STRING,WEBSESSION STRING,WORD STRING,S_SEQ INT,C_SEQ INT,WEBSITE STRING) ROW FORMAT DELIMITED FIELDS

第6页共18页 出自石山园,博客地址: http://www.cnblogs.com/shishanyuan

TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '/class5/sogouq1'; hive>show tables;

观察一下创建表和外部表的区别,会发现创建外部表多了 EXTERNAL 关键字以及指定了表对应存放文件夹 LOCATION '/class5/sogouq1'

【注】在删除表的时候,内部表将删除表的元数据和数据文件;而删除外部表的时候,仅仅删 除外部表的元数据,不删除数据文件



第三步 加载数据文件到外部表对应的目录中

创建 Hive 外部表关联数据文件有两种方式,一种是把外部表数据位置直接关联到数据文件所在 目录上,这种方式适合数据文件已经在 HDFS 存在,另外一种方式是创建表时指定外部表数据 目录,随后把数据加载到该目录下。以下将以第二种方式进行演示:

\$hadoop fs -copyFromLocal /home/hadoop/upload/sogou/SogouQ1.txt

/class5/sogouq1/

\$hadoop fs -ls /class5/sogouq1

\$hadoop fs -tail /class5/sogouq1/SogouQ1.txt

hadoop1 hadoop2 hadoop3									
<pre>[[hadoop@hadoop1 ~]\$ hadoop fs -copyFromLocal /home/hadoop/upload/sogou/SogouQ1.txt /class5/sogouq1/</pre>									
Found 1 items									
-rw-rr- 2 hadoop supergroup 108750574 2015-07-23 17:12 /class5/sogouq1/SogouQ1.txt [hadoop@hadoop1 ~]\$ hadoop fs -tail /class5/sogouq1/SogouQ1.txt /index.html									
20111230114209 baaaf1a5de7d43ead0e8304620b62352 anii.com/gongsi/5484315	长?阿???	1	5	http://www.g					
20111230114209 08802c6e199d82c166bbbabd30c7ee66 sina.com.cn/b/14749937.html	?不??缓?C??	3	1	http://iask.					
20111230114209 f77b6e57e97fd04de7c85fbbc4dd363f zpxwsp/	???? 1	1	http	://www.zpgd.net/					
			1 A A A A A A A A A A A A A A A A A A A						

Contents of directory <u>/class5</u>/sogouq1

1	Goto : /class5/sogouq1 go								
	Go to parent directory								
	Name	Гуре	Size	Replication	Block Size	Modification Time	Permissi on	Owner	Group
	SogouQ1. txt	file	103.71 MB	2	128 MB	2015-07-23 17:12	rw-rr	hadoop	supergroup

1.2.2 查询行数

hive>select count(*) from SOGOUQ1;

第7页共18页 出自石山园,博客地址:http://www.cnblogs.com/shishanyuan



	J	ob Overview
Job Name:	select count(*) from SOGOUQ1(Stage=1)	
State:	RUNNING	
Uberized:	false	
Started:	Thu Jul 23 17:15:05 CST 2015	
Elapsed:	15sec	

	Appricationmas	ter							
	Attempt Numbe	er	Start Time					Node	
1		Th	u Jul 23	17:14:59 CST	2015		hadoop3:	<u>hadoop3:8042</u> <u>log</u>	
	Task Type Progre			s Total Pe		nding	Running	Co	mplete
	Map			1	0		0	1	
	Reduce			1	1		0	0	
	Attempt Type	New		Running	Faile	d	Killed	Succes	ssful
	Maps	<u>0</u>	<u>0</u>		<u>0</u>	<u>0</u>		1	
	Reduces	1	0		0	0		0	

1.2.3 显示前 10 行

hive>select * from SOGOUQ1 limit 10;

hadoop1 hadoop2	hadoop3				
hive> select * 1	From SOGOUQ1 limit 10;				
ок					
20111230000005	57375476989eea12893c0c3811607bcf	?????? 1	1	http://www.qiyi.com/	
20111230000005	66c5bb7774e31d0a22278249b26bc83a	?????????	3	1 http://www.booksky.org/E	3
20111230000007	b97920521c78de70ac38e3713f524b50	???????	1	1 http://www.bblianmeng.com	ן כ
20111230000008	6961d0c97fe93701fc9c0d861d096cd9	???????????????????????????????????????	1	1 http://lib.scnu.edu.cn/	
20111230000008	f2f5a21c764aebde1e8afcc2871e086f	?????? 2	1	http://proxyie.cn/	
20111230000009	96994a0480e7e1edcaef67b20d8816b7	????? 1	1	http://movie.douban.com/review/1	L 📝
20111230000009	698956eb07815439fe5f46e9a4503997	youku 1	1	http://www.youku.com/	
20111230000009	599cd26984f72ee68b2b6ebefccf6aed	?????365??????	1	<pre>1 http://hf.house365.com/</pre>	
20111230000010	f577230df7b6c532837cd16ab731f874	???????????????????????????????????????	1	<pre>1 http://www.kz321.com/</pre>	
20111230000010	285f88780dd0659f5fc8acc7cc4949f2	IQ???? 1	1	http://www.igshuma.com/	
Time taken: 0.12	22 seconds, Fetched: 10 row(s)	-		• • • •	

可以看出 Hive 会根据查询不同任务决定是否生成 Job,获取前 10 条并没有生成 Job,而是得到数据后直接进行显示。

1.2.4 查询结果排名第1,点击次序排第2的数据

hive>select count(*) from SOGOUQ1 where S_SEQ=1 and C_SEQ=2;

hadoop1 hadoop2 hadoop3	
hive> select count(*) from SOGOUQ1 where S_SEQ=1 and C_SEQ=2;	
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></number>	
In order to limit the maximum number of reducers:	
set hive.exec.reducers.max= <number></number>	
In order to set a constant number of reducers:	
set mapreduce.job.reduces= <number></number>	
Starting Job = job_143/639/10/11_0005, Tracking URL = http://hadoop1:8088/proxy/application_143/639	
Kill Command = / App/hadoop/hadoop-2.2.0/bin/hadoop]obKill]ob_143/639210/11_0005	
Hadoop Job Information for Stage-1: number of mappers: 1; number of reducers: 1	
2015 - 07 - 23 17:17:31,207 Stage-1 Map = 0%, reduce = 0%	
2015-07-23 17:17:43,401 Stage-1 map = 100%, reduce = 0%, cumulative CPU 0.70 Sec	
2015-07-25 17.17.57.270 Stage-1 map = 100%, reduce = 100%, cumulative CP0 9.02 Sec	
radad lab - jab 1477520210711 0005	
$\frac{1}{100} = \frac{1}{100} = \frac{1}$	
Tab 0: Man: 1 Poduce: 1 Cumulative CPU: 9 02 sec HDES Pead: 108750774 HDES Write: 6 SUCCESS	
Total ManPedure CPU Time Spent' 9 seconds 20 msec	
ok	
19771	
Time taken: 40.659 seconds, Fetched: 1 row(s)	

		Job Overview
Job Name:	select count(*) from SOGOUQ1 whereC_SEQ=2(Stage=1)	
State:	RUNNING	
Uberized:	false	
Started:	Thu Jul 23 17:17:30 CST 2015	
Elapsed:	15sec	

ApplicationMas	ter							
Attempt Numbe	er		Start	Node		Logs		
1	Thu	Jul 23 1	7:17:22 CST 2	015		<u>hadoop1:8042</u>		<u>logs</u>
Task Type	Progress	5	Total	Total Pending		Running	Complete	
Map			1	0	0		1	
Reduce			1	1	0		0	
Attempt Type	New	1	Running	Failed	Ki	illed	Successf	ul
Maps	Q	<u>0</u>		<u>0</u>	<u>0</u>	1		
Reduces	<u>1</u>	<u>0</u>		<u>0</u>	<u>0</u>	<u>0</u>		

1.2.5 查询次数排行榜

按照 session 号进行归组,并按照查询次数进行排序,最终显示查询次数最多的前10条。

hive>select WEBSESSION,count(WEBSESSION) as cw from SOGOUQ1 group by

WEBSESSION order by cw desc limit 10;

hadoop1 hadoop2 hadoop3	
hives select WEBSESSION, count (WEBSESSION) as cw from SOGOUQ1 group by WEBSESSION order by cw desc limit 10;	
Total jobs = 2	
Launching Job 1 out of 2	
Number of reduce tasks not specified. Estimated from input data size: 1	
In order to change the average load for a reducer (in bytes):	
set nive.exec.reducers.bytes.per.reducer= <number></number>	
In order to limit the maximum number of reducers:	
set nive.exec.reducers.max= <number></number>	
In order to set a constant number of reducers:	
set mapreduce.job.reduces= <number></number>	
<pre>[starting Job = Job_143/639210/11_0006, Tracking URL = http://hadoop1:8088/proxy/application_143/639210/11_0 local</pre>	
1006/ Will command (and the deep 2 2 0 /big /badeep inb 1411 inb 1427(20010711 000)	
KIII Command = / app/hadoop/hadoop-2.2.0/bin/hadoop job -KIII job_143/639210/II_0006	
Hadoop Do information for stage-1: number of mappers: 1; number of reducers: 1	
2015-07-23 17:20:11,230 State-1 map = 0%, reduce = 0%	
2015-07-23 17:20:20,959 Stage-1 map = 100%, reduce = 0%, cumulative CPU 12.59 Sec	
ManBedice Total cumulative CPU trainer 20 escende 20 mset	
raded to a total cumulative to time. 20 seconds 500 msec	
leurer 200 - 200 - 1422 0225 021 - 0000	

2 交易数据演示

2.1 准备数据

2.1.1 上传数据

交易数据存放在该系列配套资源的/class5/saledata 目录下,在/home/hadoop/upload 创建 class5 目录用于存放本周测试数据

\$cd /home/hadoop/upload \$mkdir class5

创建新文件夹后使用,使用 SSH Secure File Transfer 工具上传到 /home/hadoop/upload/class5目录下,如下图所示:

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>O</u> peration <u>W</u> indow <u>H</u> elp									
🖬 📕 🍠 🛍 🛍 🎽	🖶 🔎 🏂 🖻 💼 🧾 🍋 🦠 🧈 û 🔚 🖭 🗄 🏥 🏢 abş 101 🎉 🐼 🧇 💦								
📗 🗾 Quick Connect 🛛 📄 Pr	👔 Quick Connect 💼 Profiles								
🛛 🔁 🖄 🛍 🗢 🖬 🛣 🗙	F:\1.Spark\炼	数成≝ <mark>▼</mark> Add Add	🖄 🖻 🌣 📑 🕻	× <mark>/</mark> h	ome/hadoop	/upl - Add			
Local Name 🛆 Size	Type Mod	lified Remote Name	7	Size	Туре	Modified			
퉬 saledata	文件夹 201	.5/02/ 🌗 saledata			Folder	2015/02/04 14			
퉬 wiki_parquet	<u>\\\</u> /#-#- 201	📫 🔰 🛯 wiki_parqu	et		Folder	2015/02/04 14			
people.txt 32	TXT 文件 201	4/06/ people.txt		32	TXT 文件	2014/06/18 18			
• [III						•			
Transfer Oueue									

2.1.2 在 Hive 创建数据库和表

启动 Hadoop 集群,进入 Hive 命令行操作界面,使用如下命令创建三张数据表:

- tbDate 定义了日期的分类,将每天分别赋予所属的月份、星期、季度等属性,字段分别为日期、年月、年、月、日、周几、第几周、季度、旬、半月;
- tbStock 定义了订单表头,字段分别为订单号、交易位置、交易日期;
- tbStockDetail 文件定义了订单明细,该表和 tbStock 以交易号进行关联,字段分别为订单 号、行号、货品、数量、金额:

hive>use hive;

hive>CREATE TABLE tbDate(dateID string,theyearmonth string,theyear string,themonth string,thedate string,theweek string,theweeks string,thequot string,thetenday string,thehalfmonth string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY ',' (

hive>CREATE TABLE tbStock(ordernumber STRING,locationid string,dateID string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'; hive>CREATE TABLE tbStockDetail(ordernumber STRING,rownum int,itemid string,qty int,price int ,amount int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

I hadoop1 hadoop2 hadoop3
hive> CREATE TABLE tbDate(dateID string,theyearmonth string,theyear string,themonth string,thedate string,t
heweek string,theweeks string,thequot string,thetenday string,thehalfmonth string) ROW FORMAT DELIMITED FIE
LDS TERMINATED BY ',' LINES TERMINATED BY '\n';
OK
Time taken: 1.574 seconds
hive> CREATE TABLE tbStock(ordernumber STRING,locationid string,dateID string) ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' LINES TERMINATED BY '\n';
OK
Time taken: 0.099 seconds
hive> CREATE TABLE tbStockDetail(ordernumber STRING,rownum int,itemid string,qty int,price int ,amount int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';
OK
Time taken: 0.131 seconds
hive> show tables;
OK
sogouq1
sogouq2
TbDate
tbStock
tbStock
tbStock
tbStock
tbStock
tbStock

2.1.3 导入数据

从本地操作系统分别加载日期、交易信息和交易详细信息表数据

hive>use hive;

hive>LOAD DATA LOCAL INPATH '/home/hadoop/upload/class5/saledata/tbDate.txt' INTO TABLE tbDate:

hive>LOAD DATA LOCAL INPATH '/home/hadoop/upload/class5/saledata/tbStock.txt' INTO TABLE tbStock;

hive>LOAD DATA LOCAL INPATH '/home/hadoop/upload/class5/saledata/tbStockDetail.txt' INTO TABLE tbStockDetail;

I hadoop1 hadoop2 hadoop3
hive> LOAD DATA LOCAL INPATH '/home/hadoop/upload/class5/saledata/tbDate.txt' INTO TABLE tbDate;
Copying data from file:/home/hadoop/upload/class5/saledata/tbDate.txt
Loading data to table hive.tbdate
Table hive.tbdate stats: [numFiles=1, numRows=0, totalSize=171512, rawDataSize=0]
OK
Time taken: 1.863 seconds
hive> LOAD DATA LOCAL INPATH '/home/hadoop/upload/class5/saledata/tbStock.txt' INTO TABLE tbStock;
Copying data from file:/home/hadoop/upload/class5/saledata/tbStock.txt
Copying file: file:/home/hadoop/upload/class5/saledata/tbStock.txt
Copying data to table hive.tbstock
Time taken: 0.672 seconds
hive> LOAD DATA LOCAL INPATH '/home/hadoop/upload/class5/saledata/tbStockDetail.txt' INTO TABLE tbStockDetail:
Copying file: file:/home/hadoop/upload/class5/saledata/tbStockDetail.txt
Copying file: file:/home/hadoop/upload/class5/saledata/tbStockDetail.txt
Loading data to table hive.tbstockdetail
Copying file: file:/home/hadoop/upload/class5/saledata/tbStockDetail.txt
Copying file: file:/home/hadoop/upload/class5/saledata/tbStockDetail.txt
Loading data to table hive.tbstockdetail
Table hive.tbstockdetail stats: [numFiles=1, numRows=0, totalSize=11992131, rawDataSize=0]
OK
Table hive.tbstockdetail stats: [numFiles=1, numRows=0, totalSize=11992131, rawDataSize=0]
OK

查看 HDFS 中相关 SALEDATA 数据库中增加了三个文件夹,分别对应三个表:

第 11 页 共 18 页 出自石山园,博客地址: http://www.cnblogs.com/shishanyuan

← → C 🗋 hadoop1:50075/browseDirectory.jsp?dir=%2Fuser%2Fhive%2Fwarehouse%2Fhive.db&nam

Contents of directory /user/hive/warehouse/hive.db

Goto : /user/hive/	oto : /user/hive/warehouse/hive.db go													
Name	Туре	Size	Replication	Block	Size	Modification Time	Permission	0wner	Group					
sogouq2	dir					2015-07-23 16:01	rwxr-xr-x	hadoop	supergroup					
<u>tbdate</u> 📕	dir					2015-07-23 22:00	rwxr-xr-x	hadoop	supergroup					
tbstock	dir					2015-07-23 22:00	rwxr-xr-x	hadoop	supergroup					
tbstockdetail	dir					2015-07-23 22:00	rwxr-xr-x	hadoop	supergroup					

2.2 计算所有订单每年的总金额

2.2.1 算法分析

要计算所有订单每年的总金额,首先需要获取所有订单的订单号、订单日期和订单金信息, 然后把这些信息和日期表进行关联,获取年份信息,最后根据这四个列按年份归组统计获取所 有订单每年的总金额。

2.2.2 执行 HSQL 语句

hive>use hive;

hive>select c.theyear, sum(b.amount) from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and a.dateid=c.dateid group by c.theyear order by c.theyear;

hive> select c.theyear, sum(b.amount) from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber an d a.dateid=c.dateid group by c.theyear order by c.theyear; Total jobs = 2 15/07/23 22:04:57 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-04-50_855_1652573098814475079-1/-lo cal-10009/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification.max.retry.interval; Ignoring. 15/07/23 22:04:57 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-04-50_855_1652573098814475079-1/-lo cal-10009/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification.max.attempts; Ignori ng. 15/07/23 22:04:57 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.re duces 15/07/23 22:04:57 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.inpu t.fileinputformat.split.minsize 15/07/23 22:04:57 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead is mapreduce.reduce.speculative 15/07/23 22:04:57 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapred 15/07/23 22:04:57 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapred 15/07/23 22:04:57 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapred 15/07/23 22:04:57 INFO Configuration.deprecation: mapred.min.split.size.per.node is deprecated. Instead, use mapred uce.input.fileinputformat.split.minsize.per.node

运行过程中创建两个 Job ,分别为 job_1437659442092_0001 和 job_1437659442092_0002 , 运行过程如下: Starting Job = job_1437659442092_0001, Tracking URL = http://hadoop1:8088/proxy/application_1437659442092_0001/ Kill command = /app/hadoop/hadoop-2.2.0/bin/hadoop job -kill job_1437659442092_0001 Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1 2015-07-23 22:05:18,947 Stage-3 map = 00%, reduce = 0% 2015-07-23 22:05:35,954 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 8.88 sec 2015-07-23 22:05:45,629 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 11.01 sec MapReduce Total cumulative CPU time: 11 seconds 10 msec Ended Job = job_1437659442092_0001 Launching Job 2 out of 2 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 change the average load for a reducers: set hive.exec.reducers.max=<number> In order to save<number of reducers: set napreduce.job.reduces=<number> Starting Job = job_1437659442092_0002, Tracking URL = http://hadoop1:8088/proxy/application_1437659442092_0002/ Kill Command = /app/hadoop/hadoop-2.2.0/bin/hadoop job _kill job_1437659442092_0002 Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1 2015-07-23 22:06:10,483 Stage-4 map = 00%, reduce = 0%, Cumulative CPU 1.69 sec 2015-07-23 22:06:10,085 stage-4 map = 100%, reduce = 0%, Cumulative CPU 3.78 sec MapReduce Total cumulative CPU time: 3 seconds 780 msec

在 YARN 的资源管理器界面中可以看到如下界面:

← → C 🗋 hadoop	1 :8088/cluste	er													區 🏠 🐴 🗏
<u>Ened</u>		D					All Ap	plic	atio	าร					Logged in as: dr.who
 ▼ Cluster 	Cluster Met	trics													
About Nodes	Apps Submitted	Apps Pending	Apps Runnin;	g Comp	pps pleted	Containers Running	Memory Used	Memory Total	Memory Reserve	Active d Nodes	Decomm	nissioned odes	Lost Nodes	Unheal Node	thy Rebooted s Nodes
Applications	2	0	1	1		2	3 GB	24 GB	0 B	3	<u>0</u>	Ś	2	<u>0</u>	<u>0</u>
NEW SAVING	Show 20 🔻	entries											Searc	h:	
SUBMITTED ACCEPTED		ID	~	User \$	ŀ	Name 🗘	Application Type ≎	Queue \$	StartTime ≎	FinishTime \$	State ≎	FinalStatu	s Prog	ress \$	Tracking UI ≎
REMOVING FINISHING FINISHED FAILED	application	1437659442	<u>092_0002</u>	hadoop	select sum(b.a c.th 4)	c.theyear, mount) eyear(Stage-	MAPREDUCE	default	Thu, 23 Jul 2015 14:05:49 GMT	N/A	RUNNING	UNDEFINED			<u>ApplicationMaster</u>
<u>KILLED</u> Scheduler	application	1437659442	<u>092 0001</u>	hadoop	select sum(b.a c.th 3)	c.theyear, mount) eyear(Stage-	MAPREDUCE	default	Thu, 23 Jul 2015 14:05:05 GMT	Thu, 23 Jul 2015 14:05:45 GMT	FINISHED	SUCCEEDED			<u>History</u>
 Tools 	Showing 1 to	o 2 of 2 en	tries										Firs	t Previou	ıs 1 Next Last

2.2.3 查看结果

整个计算过程使用了 91.51 秒,结果如下:

MapReduce Job 0: Ma Job 1: Ma Total Ma	e Jobs Launched: ap: 1 Reduce: 1 ap: 1 Reduce: 1 pReduce CPU Time	Cumulative CPU: 11.01 sec Cumulative CPU: 3.78 sec Spent: 14 seconds 790 msec	HDFS Read: 11992364 HDFS Write: 278 SUCCESS HDFS Read: 641 HDFS Write: 94 SUCCESS	
2004 2005 2006 2007	3265696 13247234 13670416 16711974			
2008 2009 <u>2010</u> Time tak	14670698 6322137 210924 en: 91.515 second	ds, Fetched: 7 row(s)		Ш

2.3 计算所有订单每年最大金额订单的销售额

2.3.1 算法分析

该算法分为两步:

- 1. 按照日期和订单号进行归组计算,获取所有订单每天的销售数据;
- 把第一步获取的数据和日期表进行关联获取的年份信息,然后按照年份进行归组,使用 Max 函数,获取所有订单每年最大金额订单的销售额。

2.3.2 执行 HSQL 语句

//所有订单每年最大金额订单的销售额

//第一步:

hive>use hive;

hive>select a.dateid,a.ordernumber,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b where a.ordernumber=b.ordernumber group by

a.dateid,a.ordernumber;

//第二步:

hive>select c.theyear,max(d.sumofamount) from tbDate c,(select a.dateid,a.ordernumber,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b where a.ordernumber=b.ordernumber group by a.dateid,a.ordernumber) d where c.dateid=d.dateid group by c.theyear sort by c.theyear;

hive> select c.theyear,max(d.sumofamount) from tbDate c,(select a.dateid,a.ordernumber,sum(b.amount) as sumofam ount from tbStock a,tbStockDetail b where a.ordernumber=b.ordernumber group by a.dateid,a.ordernumber) d where c.dateid=d.dateid group by c.theyear sort by c.theyear; Total jobs = 2 15/07/23 22:13:02 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-12-57_688_5621731284397117739-1/ -local-10011/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification.max.retry.inter 15/07/23 22:13:02 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-12-57_688_5621731284397117739-1/ -local-10011/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification.max.attempts; Ignoring. 15/07/23 22:13:02 INFO configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job .reduces 15/07/23 22:13:02 INFO configuration.deprecation: mapred.min.split.size is deprecated. Instead, use mapreduce.io nput.fileinputformat.split.minsize

运行过程中创建两个 Job ,分别为 job_1437659442092_0004 和 job_1437659442092_0005 , 运行过程如下:

In order to set a constant number of reducers: set mapreduce.job.reduces=<number> Starting Job = job_1437659442092_0004, Tracking URL = http://hadoop1:8088/proxy/application_1437659442092_0004/ Hadoop job information for Stage-4 map = 0%, reduce = 0% 2015-07-23 22:13:18,591 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 9.26 sec 2015-07-23 22:13:14,025 Stage-4 map = 100%, reduce = 100%, Cumulative CPU 14.89 sec MapReduce Total cumulative CPU time: 14 seconds 890 msec Ended Job = job_1437659442092_0004 15/07/23 22:13:50 WARN conf.configuration: file:/tmp/hadoop/hive_2015-07-23_22-12-57_688_5621731284397117739-1/ -local-10015/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification.max.retry.inter val; Ignoring. In order to set a constant number of reducers: set mapreduce.job.reduces=<number> starting Job = job_1437659442092_0005, Tracking URL = http://hadoop1:8088/proxy/application_1437659442092_0005/ Kill Command = /app/hadoop/hadoop-2.2.0/bin/hadoop job -kill job_1437659442092_0005/ Hadoop job information for Stage-2 map = 0%, reduce = 0% 2015-07-23 22:14:16,799 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 6.11 sec 2015-07-23 22:14:26,321 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 9.01 sec MapReduce Total cumulative CPU time: 9 seconds 10 msec Ended Job = job_1437659442092_0005

在 YARN 的资源管理器界面中可以看到如下界面:

品公 春日

Logged in as: dr.who

All Applications

Cluster Meti	rics													
Apps Submitted	Apps Pending	Ru	Apps mning	Apps Containers Memory Completed Running Used			Memory Total	Memory Reserve	d No	tive De des	commissione Nodes	ed Lost Nodes	Unheal th Nodes	ny Rebooted Nodes
5	0	1		4	2	3 GB	24 GB	0 B	<u>3</u>	<u>0</u>		<u>0</u>	<u>0</u>	<u>0</u>
Show 20 🔻	entries												Search:	
	ID	*	User ≎		Name	\$	Application Type ≎	Queue	StartTime ≎	FinishTime \$	State ≎	FinalStatus \$	Progress ≎	Tracking UI 💲
application :	1437659442092	0005	hadoop	select c.theyear,max 2)	(d. sumof amou c	. theyear (Stage-	MAPREDUCE	default	Thu, 23 Jul 2015 14:13:55 GMT	N/A	RUNNING	UNDEFINED		<u>ApplicationMaster</u>
application :	1437659442092	0004	hadoop	select c.theyear,max 4)	(d. sumof amou c	. theyear (Stage-	MAPREDUCE	default	Thu, 23 Jul 2015 14:13:07 GMT	Thu, 23 Jul 2015 14:13:44 GMT	FINISHED	SUCCEEDED		<u>History</u>
application :	1437659442092	0003	hadoop	select a.dateid, a.or 2)	ddateid, a.oro	ernumber(Stage-	MAPREDUCE	default	Thu, 23 Jul 2015 14:11:33 GMT	Thu, 23 Jul 2015 14:12:10 GMT	FINISHED	SUCCEEDED		<u>History</u>
application :	1437659442092	<u>0002</u>	hadoop	select c.they c.theyear(ear, sum(b.amour Stage-4)	t)	MAPREDUCE	default	Thu, 23 Jul 2015 14:05:49 GMT	Thu, 23 Jul 2015 14:06:20 GMT	FINISHED	SUCCEEDED		<u>History</u>
application :	1437659442092	0001	hadoop	select c.they c.theyear(ear, sum(b.amour Stage-3)	t)	MAPREDUCE	default	Thu, 23 Jul 2015 14:05:05 GMT	Thu, 23 Jul 2015 14:05:45 GMT	FINISHED	SUCCEEDED		<u>History</u>
Showing 1 to	howing 1 to 5 of 5 entries First Previous 1 Next Last													

其中 job_1437659442092_0005 运行的具体情况如下:

						Job Overview			
Job Name: s	elect c.theyear,ma	ct c.theyear,max(d.sumofamouc.theyear(Stage-2)							
State: H	UNNING								
Uberized: f	alse								
Started: 1	hu Jul 23 22:14:02	CST 2015							
Elapsed: 1	.7sec								
ApplicationMast	ter								
Attempt Numbe:	r		Start Time					Logs	
1	T	hu Jul 23 22:13	:57 CST 2015			hadoop3:804	hadoop3:8042		
Task Type	Prog	ress	Total	Total Pending		Running		Complete	
Map			1	0		0	1		
Reduce			1	0		1	0		
Attempt Type	Ne	w	Running	Faile	d	Killed		Successful	
Maps	<u>0</u>	<u>0</u>		<u>0</u>	<u>0</u>		1		
Reduces	<u>0</u>	1		0	<u>0</u>		<u>0</u>		

2.3.3 查看结果

整个计算过程使用了 285 秒,结果如下:

MapReduce Jobs Launched: Job 0: Map: 1 Reduce: 1 Cumulative CPU: 14.89 sec Job 1: Map: 1 Reduce: 1 Cumulative CPU: 9.01 sec Total MapReduce CPU Time Spent: 23 seconds 900 msec OK 2004 23612 2005 38180 2006 36124 2007 159126 2008 55828 2009 25810 2010 13063 Time taken: 90.004 seconds, Fetched: 7 row(s)

2.4 计算其他金额

2.4.1 所有订单中季度销售额前 10 位

//所有订单中季度销售额前 10 位 hive>use hive;

第 15 页 共 18 页 出自石山园,博客地址: http://www.cnblogs.com/shishanyuan

hive>select c.theyear,c.thequot,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and

a.dateid=c.dateid group by c.theyear,c.thequot order by sumofamount desc limit 10;

- 2005 3 3304243

hadoop1 hadoop2 hadoop3 hive> select c.theyear,c.thequot,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and a.dateid=c.dateid group by c.theyear,c.thequot order by sumofamo unt desc limit 10; unt desc limit 10; Total jobs = 2 15/07/23 22:35:27 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-35-22_672_5143929466869 932-1/-local-10009/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification. max.retry.interval; Ignoring. 15/07/23 22:35:27 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-35-22_672_5143929466869 932-1/-local-10009/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification. max.attempts; Ignoring. 15/07/23 22:35:27 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead. use mapreduce.job.reduces educe.job.reduces 15/07/23 22:35:27 INFO Configuration.deprecation: mapred.min.split.size is deprecated. Instead, use ma preduce.input.fileinputformat.split.minsize MapReduce Jobs Launched: Cumulative CPU: 11.68 sec Cumulative CPU: 3.24 sec Job 0: Map: 1 Reduce: 1 Job 1: Map: 1 Reduce: 1 HDFS Read: 11992364 HDFS Write: 767 SUCCESS HDFS Read: 1127 HDFS Write: 150 SUCCESS Total MapReduce CPU Time Spent: 14 seconds 920 msec <u>ок</u> 2008 2006 1 2

2.4.2 列出销售金额在 100000 以上的单据

taken: 79.628 seconds, Fetched: 10 row(s)

//列出销售金额在100000以上的单据

hive>use hive;

Time

hive>select a.ordernumber,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b where a.ordernumber=b.ordernumber group by a.ordernumber having sumofamount>100000;

I hadoop1 | hadoop2 | hadoop3
hives select a. ordernumber, sum(b. amount) as sumofamount from tbStock a,tbStockDetail b where a. ordernumber-b. ordernumber group by a. ordernumber having sumofamount>10000;
Total jobs = 1
15/07/23 22:40:09 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-40-04_321_4258109466309
171632-1/-local-10007/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notificati
0.7/23 22:40:09 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-40-04_321_4258109466309
171632-1/-local-10007/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notificati
0.7/23 22:40:10 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduces
15/07/23 22:40:10 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated.
15/07/23 22:40:10 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated.
Kill command = /app/hadoop/hadoop-2.2.0/bin/hadoop job -kill job_1437659442092_0016
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2015-07-23 22:40:45,722 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 8.0 sec
2015-07-23 22:40:45,722 Stage-2 map = 100%, reduce = 100%, cumulative CPU 13.41 sec
MapReduce Total cumulative CPU time: 13 seconds 410 msec
Finde Job = job_1437659442092_0016
MapReduce Instead: 1199286 HDFS Write: 42 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 410 msec
MilsL00009024 119058
HMJSL00009024 119058
HMJSL000

2.4.3 所有订单中每年最畅销货品

//所有订单中每年最畅销货品

第一步:

hive>use hive;

hive>select c.theyear,b.itemid,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and a.dateid=c.dateid group by c.theyear,b.itemid;

第二步:

hive>select d.theyear,max(d.sumofamount) as maxofamount from (select c.theyear,b.itemid,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and a.dateid=c.dateid group by c.theyear,b.itemid) d group by d.theyear ;

第三步:

hive>select distinct e.theyear,e.itemid,f.maxofamount from (select c.theyear,b.itemid,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and a.dateid=c.dateid group by c.theyear,b.itemid) e , (select d.theyear,max(d.sumofamount) as maxofamount from (select c.theyear,b.itemid,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and a.dateid=c.dateid group by c.theyear,b.itemid) d group by d.theyear) f where e.theyear=f.theyear and e.sumofamount=f.maxofamount order by e.theyear; 2004 JY424420810101 53374 2005 24124118880102 56569

第 17 页 共 18 页 出自石山园,博客地址: http://www.cnblogs.com/shishanyuan

2006 JY425468460101 113684

2007 JY425468460101 70226

2008 E2628204040101 97981

2009 YL327439080102 30029

2010 SQ429425090101 4494

hadoop1 hadoop2 hadoop3

hive> select distinct e.theyear,e.itemid,f.maxofamount from (select c.theyear,b.itemid,sum(b.amount)	
as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.ordernumber=b.ordernumber and a.dateid=	
c.dateid group by c.theyear,b.itemid) e , (select d.theyear,max(d.sumofamount) as maxofamount from (se	
lect c.theyear,b.itemid,sum(b.amount) as sumofamount from tbStock a,tbStockDetail b,tbDate c where a.o	
rdernumber=b.ordernumber and a.dateid=c.dateid group by c.theyear,b.itemid) d group by d.theyear) f wh	
ere e.theyear=f.theyear and e.sumofamount=f.maxofamount order by e.theyear;	
Total jobs = 7	
15/07/23_22:42:45 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-42-40_641_7821765305197	
2220-1/-local-10020/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification	
.max.retry.interval; Ignoring.	
15/07/23_22:42:45 WARN conf.Configuration: file:/tmp/hadoop/hive_2015-07-23_22-42-40_641_7821765305197	
2220-1/-local-10020/jobconf.xml:an attempt to override final parameter: mapreduce.job.end-notification	
.max.attempts; Ignoring.	
15/07/23 22:42:45 INFO Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapr	
educe. job. reduces	
MapReduce Jobs Launched:	
JOD 0: Map: 1 Reduce: 1 Cumulative CPU: 20.52 sec HDFS Read: 11992364 HDFS Write: 2/4 SUCCESS	
JOD 1: Map: 1 Reduce: 1 Cumulative CPU: 22.39 Sec HDFS Read: 1192304 HDFS Write: 005004 SUCCESS	
JOD 2: Map: 1 Cumulative CPU: 5.6/ Sec HDFS Read: 603365 HDFS Write: 3/9 SUCCESS	
Job 3: Map: 1 Reduce: 1 Cumulative CPU: 5.79 Sec. HDFS Read: 740 HDFS Write: 379 SOCCESS	
JOD 4: Map: 1 Reduce: 1 Cumulative CPU: 0.0 Sec HDFS Read: 740 HDFS WRITE: 182 SUCCESS	
Total MapReduce CPO Time Spent: I minutes o seconds 970 msec	
UK 2004 2V424420810101 E2274	
2004 J1424420810101 J3374	
2003 24124110000102 30309	
2007 1423468460101 113064	
2007 51725406400001 70220	
2010 50/20/25000101 4/0/	
$\frac{2}{100}$ $\frac{1}{100}$ $\frac{1}$	-
The caken 240.155 seconds, recencer / Tow(5)	