### 请用Hive完成（手写SQL）（10分）

有50W个京东店铺，每个顾客访问任何一个店铺的任何一个商品时都会产生一条访问日志，访问日志存储的表名为Visit，访客的用户id为user\_id，被访问的店铺名称为shop，访问时间为visit\_time。数据样例：'huawei','1001','2017-02-10'，'apple','1001','2017-02-11'……

请统计：

1. 每个店铺的UV（访客数）
2. 每个店铺访问次数top3的访客信息。输出店铺名称、访客id、访问次数

建表：

drop table if exists Visit;

create table Visit(

shop string COMMENT '店铺名称',

user\_id string COMMENT '用户id',

visit\_time string COMMENT '访问时间'

)

row format delimited fields terminated by '\t';

插入数据：

insert into table Visit values ('huawei','1005','2017-02-10');

insert into table Visit values ('huawei','1005','2017-02-10');

insert into table Visit values ('huawei','1005','2017-02-10');

insert into table Visit values ('huawei','1005','2017-02-10');

insert into table Visit values ('huawei','1004','2017-02-10');

insert into table Visit values ('huawei','1004','2017-02-10');

insert into table Visit values ('huawei','1003','2017-02-10');

insert into table Visit values ('huawei','1003','2017-02-10');

insert into table Visit values ('huawei','1001','2017-02-10');

insert into table Visit values ('huawei','1002','2017-02-10');

insert into table Visit values ('huawei','1006','2017-02-10');

insert into table Visit values ('apple','1001','2017-02-10');

insert into table Visit values ('apple','1001','2017-02-10');

insert into table Visit values ('apple','1001','2017-02-10');

insert into table Visit values ('apple','1001','2017-02-10');

insert into table Visit values ('apple','1002','2017-02-10');

insert into table Visit values ('apple','1002','2017-02-10');

insert into table Visit values ('apple','1005','2017-02-10');

insert into table Visit values ('apple','1005','2017-02-10');

insert into table Visit values ('apple','1006','2017-02-10');

insert into table Visit values ('apple','1004','2017-02-10');

insert into table Visit values ('meizu','1006','2017-02-10');

insert into table Visit values ('meizu','1006','2017-02-10');

insert into table Visit values ('meizu','1006','2017-02-10');

insert into table Visit values ('meizu','1006','2017-02-10');

insert into table Visit values ('meizu','1003','2017-02-10');

insert into table Visit values ('meizu','1003','2017-02-10');

insert into table Visit values ('meizu','1003','2017-02-10');

insert into table Visit values ('meizu','1002','2017-02-10');

insert into table Visit values ('meizu','1002','2017-02-10');

insert into table Visit values ('meizu','1004','2017-02-10');

每个店铺的UV（访客数）

select

shop,

count(distinct(user\_id)) shop\_uv

from Visit

group by shop

order by shop\_uv desc;

或者

select

shop,

count(user\_id) shop\_uv

from

(

select

shop,

user\_id

from Visit

group by shop,user\_id

) t1

group by shop

order by shop\_uv desc;

每个店铺访问次数top3的访客信息。输出店铺名称、访客id、访问次数

select

shop `商店名称`,

user\_id `用户id`,

visit\_num `访问次数`,

rank\_num `忠诚排名`

from

(

select

shop,

user\_id,

visit\_num,

row\_number() over(partition by shop order by visit\_num desc) rank\_num

from

(

select

shop,

user\_id,

count(\*) visit\_num

from Visit

group by shop,user\_id

) t1

) t2

where rank\_num<=3;

### 请用Hive完成（手写SQL）（10分）

已知一个表ORDER\_TBL，有如下字段:Date，Order\_id，User\_id，amount。请给出sql进行统计:数据样例:2017-01-01,10029028,1000003251,33.57。

1）给出 2017年每个月的订单数、用户数、总成交金额。

2）给出2017年11月的新客数(指在11月才有第一笔订单)

建表：

drop table if exists ORDER\_TBL;

create table ORDER\_TBL

(

`Date` String COMMENT '下单时间',

`Order\_id` String COMMENT '订单ID',

`User\_id` String COMMENT '用户ID',

`amount` decimal(10,2) COMMENT '金额'

)

row format delimited fields terminated by '\t';

插入数据：

insert into table ORDER\_TBL values ('2017-10-01','10029011','1000003251',19.50);

insert into table ORDER\_TBL values ('2017-10-03','10029012','1000003251',29.50);

insert into table ORDER\_TBL values ('2017-10-04','10029013','1000003252',39.50);

insert into table ORDER\_TBL values ('2017-10-05','10029014','1000003253',49.50);

insert into table ORDER\_TBL values ('2017-11-01','10029021','1000003251',130.50);

insert into table ORDER\_TBL values ('2017-11-03','10029022','1000003251',230.50);

insert into table ORDER\_TBL values ('2017-11-04','10029023','1000003252',330.50);

insert into table ORDER\_TBL values ('2017-11-05','10029024','1000003253',430.50);

insert into table ORDER\_TBL values ('2017-11-07','10029025','1000003254',530.50);

insert into table ORDER\_TBL values ('2017-11-15','10029026','1000003255',630.50);

insert into table ORDER\_TBL values ('2017-12-01','10029027','1000003252',112.50);

insert into table ORDER\_TBL values ('2017-12-03','10029028','1000003251',212.50);

insert into table ORDER\_TBL values ('2017-12-04','10029029','1000003253',312.50);

insert into table ORDER\_TBL values ('2017-12-05','10029030','1000003252',412.50);

insert into table ORDER\_TBL values ('2017-12-15','10029032','1000003255',612.50);

1）给出 2017年每个月的订单数、用户数、总成交金额。

select

date\_format(`date`,'yyyy-MM') `date`,

count(order\_id) `订单数`,

count(distinct(user\_id)) `用户数`,

sum(amount) `总成交金额`

from ORDER\_TBL

group by date\_format(`date`,'yyyy-MM');

2）给出2017年11月的新客数(指在11月才有第一笔订单)

select

t1.user\_id

from

(

select

user\_id

from ORDER\_TBL

where date\_format(`date`,'yyyy-MM') = '2017-11'

group by user\_id

) t1

left join

(

select

user\_id

from ORDER\_TBL

where date\_format(`date`,'yyyy-MM') < '2017-11'

group by user\_id

) t2

on t1.user\_id = t2.user\_id

where t2.user\_id is null;

或者

select

count(User\_id) `11月新客数`

from

(

SELECT

User\_id,

Order\_id,

`Date`,

LAG (`DATE`,1,0) over(partition by User\_id order by `Date`) preOrderDate

FROM

ORDER\_TBL

) t1

where date\_format(`date`,'yyyy-MM')='2017-11' and preOrderDate=0;

### 数仓项目服务器选型，物理机还是云主机，请简述理由。（5分）

服务器使用物理机还是云主机？

1）机器成本考虑：

（1）物理机：以128G内存，20核物理CPU，40线程，8THDD和2TSSD硬盘，单台报价4W出头，需考虑托管服务器费用。一般物理机寿命5年左右

（2）云主机，以阿里云为例，差不多相同配置，每年5W

2）运维成本考虑：

（1）物理机：需要有专业的运维人员

（2）云主机：很多运维工作都由阿里云已经完成，运维相对较轻松

*评分标准：5分 每条一分，全部答对5分*

### 请列举五条以上Linux常用命令（5分）

|  |  |  |
| --- | --- | --- |
| 序号 | 命令 | 命令解释 |
| 1 | top | 查看内存 |
| 2 | df -h | 查看磁盘存储情况 |
| 3 | iotop | 查看磁盘IO读写(yum install iotop安装） |
| 4 | iotop -o | 直接查看比较高的磁盘读写程序 |
| 5 | netstat -tunlp | grep 端口号 | 查看端口占用情况 |
| 6 | uptime | 查看报告系统运行时长及平均负载 |
| 7 | ps -aux | 查看进程 |

*评分标准：5分 每条一分，全部答对5分*

### 列举出五个以上Hadoop集群常用端口号（5分）

dfs.namenode.http-address:50070

dfs.datanode.http-address:50075

SecondaryNameNode辅助名称节点端口号：50090

dfs.datanode.address:50010

fs.defaultFS:8020 或者9000

yarn.resourcemanager.webapp.address:8088

历史服务器web访问端口：19888

*评分标准：5分 每条一分，全部答对5分*

### 请画图说明MapReduce的Shuffle过程（画图题）（10分）





*评分标准：10分 必须画图，不画图0分。有图的话，酌情扣分。*

### 简述Flume组成（结合数仓项目说明）（5分）

Flume组成，Put事务，Take事务

Taildir Source：断点续传、多目录。Flume1.6以前需要自己自定义Source记录每次读取文件位置，实现断点续传。

File Channel：数据存储在磁盘，宕机数据可以保存。但是传输速率慢。适合对数据传输可靠性要求高的场景，比如，金融行业。

Memory Channel：数据存储在内存中，宕机数据丢失。传输速率快。适合对数据传输可靠性要求不高的场景，比如，普通的日志数据。

Kafka Channel：减少了Flume的Sink阶段，提高了传输效率。

Source到Channel是Put事务

Channel到Sink是Take事务

*评分标准：5分 Take,Put,Source,Channel,Sink，见点给分，如果结合数仓项目说明了，直接满分*

### 自定义Flume拦截器步骤（5分）

a）实现 Interceptor接口

b）重写四个方法

initialize 初始化

public Event intercept(Event event) 处理单个Event

public List<Event> intercept(List<Event> events) 处理多个Event，在这个方法中调用Event intercept(Event event)

close 方法

c）静态内部类，实现Interceptor.Builder

*评分标准：5分 参考上述答案，见点给分*

### HDFS Sink小文件处理问题（5分）

（1）HDFS存入大量小文件，有什么影响？

元数据层面：每个小文件都有一份元数据，其中包括文件路径，文件名，所有者，所属组，权限，创建时间等，这些信息都保存在Namenode内存中。所以小文件过多，会占用Namenode服务器大量内存，影响Namenode性能和使用寿命

计算层面：默认情况下MR会对每个小文件启用一个Map任务计算，非常影响计算性能。同时也影响磁盘寻址时间。

（2）数仓项目中，HDFS Sink是怎么避免生产小文件的，用了哪几个参数？

官方默认的这三个参数配置写入HDFS后会产生小文件，hdfs.rollInterval、hdfs.rollSize、hdfs.rollCount,基于以上hdfs.rollInterval=3600，hdfs.rollSize=134217728，hdfs.rollCount =0，hdfs.roundValue=10，hdfs.roundUnit= second几个参数综合作用，效果如下：

（1）tmp文件在达到128M时会滚动生成正式文件

（2）tmp文件创建超10秒时会滚动生成正式文件

举例：在2018-01-01 05:23的时侯sink接收到数据，那会产生如下tmp文件：

/atguigu/20180101/atguigu.201801010520.tmp

即使文件内容没有达到128M，也会在05:33时滚动生成正式文件

*评分标准：5分*

### 请画图说明Kafka的架构（10分）

.

### 请简述Kafka的ISR副本同步队列是什么？（5分）

ISR（In-Sync Replicas），副本同步队列。ISR中包括Leader和Follower。如果Leader进程挂掉，会在ISR队列中选择一个服务作为新的Leader。有replica.lag.max.messages（延迟条数）和replica.lag.time.max.ms（延迟时间）两个参数决定一台服务是否可以加入ISR副本队列，在0.10版本移除了replica.lag.max.messages参数，防止服务频繁的进去队列。

任意一个维度超过阈值都会把Follower剔除出ISR，存入OSR（Outof-Sync Replicas）列表，新加入的Follower也会先存放在OSR中。

### 请简述Kafka的分区分配策略以及它们的区别（5分）

在 Kafka内部存在两种默认的分区分配策略：Range和 RoundRobin。

Range是默认策略。Range是对每个Topic而言的（即一个Topic一个Topic分），首先对同一个Topic里面的分区按照序号进行排序，并对消费者按照字母顺序进行排序。然后用Partitions分区的个数除以消费者线程的总数来决定每个消费者线程消费几个分区。如果除不尽，那么前面几个消费者线程将会多消费一个分区。

例如：我们有10个分区，两个消费者（C1，C2），3个消费者线程，10 / 3 = 3而且除不尽。

C1-0 将消费 0, 1, 2, 3 分区

C2-0 将消费 4, 5, 6 分区

C2-1 将消费 7, 8, 9 分区

RoundRobin：前提：同一个Consumer Group里面的所有消费者的num.streams（消费者消费线程数）必须相等；每个消费者订阅的主题必须相同。

第一步：将所有主题分区组成TopicAndPartition列表，然后对TopicAndPartition列表按照hashCode进行排序，最后按照轮询的方式发给每一个消费线程。

### 简述Hive中内部表和外部表的区别？（5分）

1）管理表：当我们删除一个管理表时，Hive也会删除这个表中数据。管理表不适合和其他工具共享数据。

2）外部表：删除该表并不会删除掉原始数据，删除的是表的元数据

### 数仓项目中，你在Sqoop导入导出过程中遇到哪些问题？（5分）

Hive中的Null在底层是以“\N”来存储，而MySQL中的Null在底层就是Null，为了保证数据两端的一致性。在导出数据时采用--input-null-string和--input-null-non-string两个参数。导入数据时采用--null-string和--null-non-string。

### Hadoop的优化（10分）

0）HDFS小文件影响

（1）影响NameNode的寿命，因为文件元数据存储在NameNode的内存中

（2）影响计算引擎的任务数量，比如每个小的文件都会生成一个Map任务

1）数据输入小文件处理：

（1）合并小文件：对小文件进行归档（Har）、自定义Inputformat将小文件存储成SequenceFile文件。

（2）采用ConbinFileInputFormat来作为输入，解决输入端大量小文件场景。

（3）对于大量小文件Job，可以开启JVM重用。

2）Map阶段

（1）增大环形缓冲区大小。由100m扩大到200m

（2）增大环形缓冲区溢写的比例。由80%扩大到90%

（3）减少对溢写文件的merge次数。

（4）不影响实际业务的前提下，采用Combiner提前合并，减少 I/O。

3）Reduce阶段

（1）合理设置Map和Reduce数：两个都不能设置太少，也不能设置太多。太少，会导致Task等待，延长处理时间；太多，会导致 Map、Reduce任务间竞争资源，造成处理超时等错误。

（2）设置Map、Reduce共存：调整slowstart.completedmaps参数，使Map运行到一定程度后，Reduce也开始运行，减少Reduce的等待时间。

（3）规避使用Reduce，因为Reduce在用于连接数据集的时候将会产生大量的网络消耗。

（4）增加每个Reduce去Map中拿数据的并行数

（5）集群性能可以的前提下，增大Reduce端存储数据内存的大小。

4）IO传输

（1）采用数据压缩的方式，减少网络IO的的时间。安装Snappy和LZOP压缩编码器。

（2）使用SequenceFile二进制文件

5）整体

（1）MapTask默认内存大小为1G，可以增加MapTask内存大小为4-5g

（2）ReduceTask默认内存大小为1G，可以增加ReduceTask内存大小为4-5g

（3）可以增加MapTask的cpu核数，增加ReduceTask的CPU核数

（4）增加每个Container的CPU核数和内存大小

（5）调整每个Map Task和Reduce Task最大重试次数

*评分标准：10分 每大点2分*