友乾营

专注数据技术的社群

高性能不停机T+0的实现

23 期

www.raqsoft.com.cn/yqy

目录 CONTENTS

友乾营

专注数据技术的社群

T+0查询的实现方案——问题背景

友乾营

什么是T+0查询？常规查询方案的不足

用户越来越关注数据的实时性，希望能基于最新的数据进行实时运算，也就是我们常说的T+0查询。

比如交通大数据应用的场景：需要结合实时数据了解车辆通行密度，合理进行道路规划。

但常规的查询方案：数据仓库+ETL工具很难实现此类需求，往往只能处理昨天、上周甚至是更久以前的数据查询，也就是T+1、T+7、T+n等查询。

T+0查询的实现方案——常规方案的不足

友乾营

困难大概体现在如下三个方面：

1、如果历史数据和最新数据都从生产系统读取，虽然可以实现T+0查询，但是会对生产数据库造成压力，当数据量越来越大时，产生性能瓶颈，直接影响业务；并且大量的历史数据会占用高昂的数据库成本（存储成本和性能成本）。

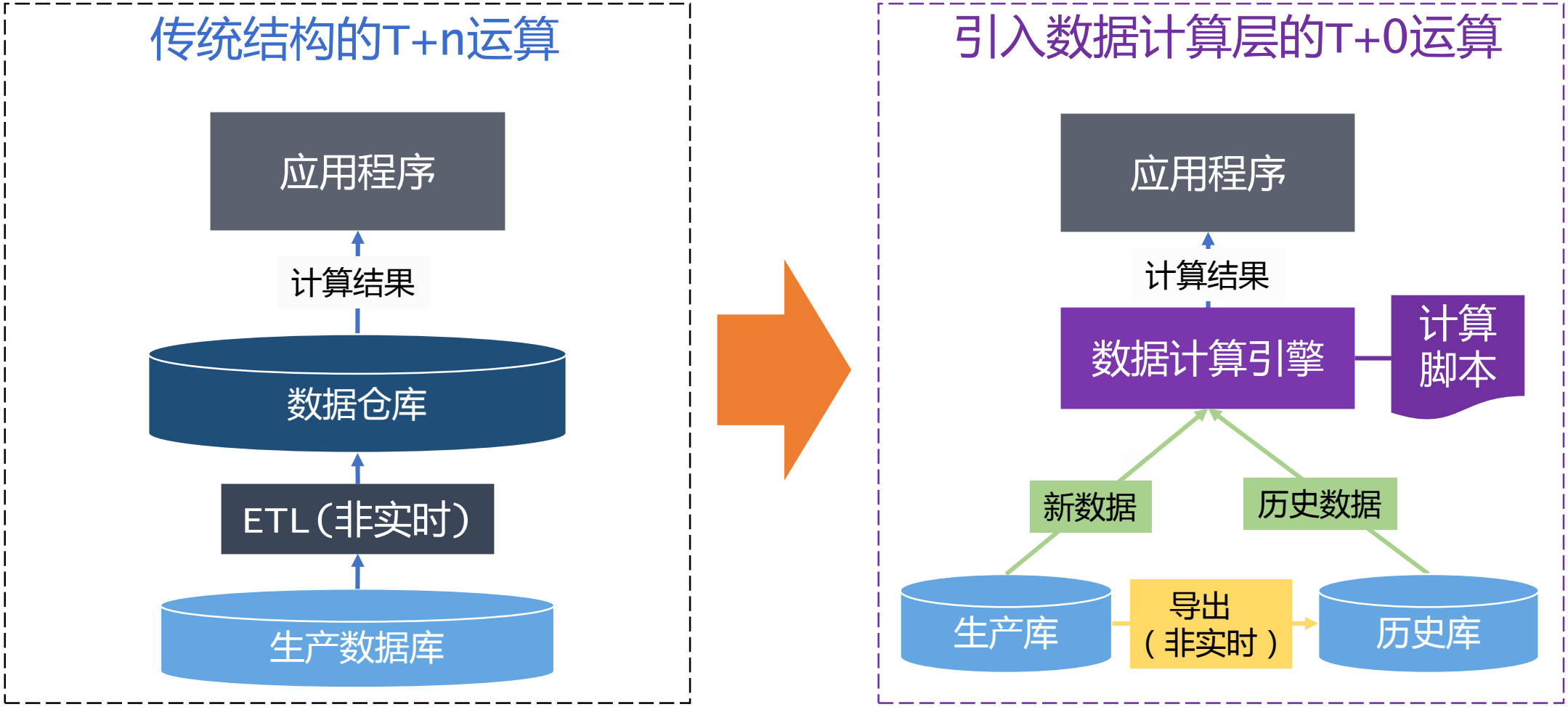
2、如果采用数据仓库的方式，那么ETL从生产库中取出数据，需要较长的“窗口时间”，一般是业务人员下班之后，到第二天早上上班之前，所以能进行分析的最新数据也只能是T+1。

3、虽然理论上可以从历史库中和生产库中同时取数据进行运算，但是一般的数据分析工具都不具备跨库混合计算的能力，其他的跨库计算方案又比较复杂，难以实施，并且性能较低。

T+0查询的实现方案——引入数据计算层

友乾营

引入数据计算层来化解难题



T+0查询的实现方案——常规分库查询回顾

常规异构分库运算需要翻译SQL；分库数据不重复，不用考虑冗余数据

举例： MySQL和Oracle两个异构分库中有销售数据表（sales），按月份和销售员ID分组，统计销售额和订单数。

分析：由于两个异构库的“取月”函数不同，需要先将标准SQL翻译为各库能识别的SQL后，再执行。若用集算器处理，代码可以这样写：

	A	B
1	=[[connect@l("mysql"),"MYSQL"],[connect@l("oracle"),"ORACLE"]]	
2	=SQL="**select month**(orderdate) ordermonth,sellerid,**sum**(amount) totalamount,**count**(amount) totalcount **from** sales **group by month**(orderdate),sellerid"	
3	fork A1	=SQL.**sqltranslate**(A3(2))
4		=A3(1).query(B3)
5	=A3.conj()	

更多分库统计方法可以参照【友乾营第十四期】分库后的统计查询

目录 CONTENTS

友乾营

专注数据技术的社群

数据库作为历史库——冷导出

允许一段“时间窗口”内从生产库取出历史数据追加导出至数据库

数据库作为历史库的冷导出实现思路：

1、初始化工作：

a) 将昨天零点前的所有数据导出至历史库；

b) 导出完成后，删除生产库中所有已导出的冗余数据。

2、每天的“时间窗口”中：

a) 停止提供T+0查询服务；

b) 将昨天一整天的数据导出至历史库；

c) 导出完成后，删除生产库中昨天一整天的冗余数据；

d) 恢复T+0查询服务。

数据库作为历史库的冷导出面临的问题：

每天的“时间窗口”内，T+0查询暂停服务，无法满足不停机业务。在T+0查询服务期间，由于生产库与历史库的数据不存冗余情况，所以查询可以按照上一页所讲的常规分库场景处理。

注意：这里导出的时间范围需要根据实际业务情况而定，不一定是“昨天”

数据库作为历史库——冷导出

友乾营

允许一段“时间窗口”内从生产库取出历史数据追加导出至数据库

生产库（MySQL），历史库（Oracle），初始化历史库

	A
1	=connect("mysql")
2	=A1.cursor@x("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
3	=file("his_temp.txt").export(A2;"|")
4	=system("sqlldr user/password@ORCLPDB control=his_temp.ctl direct=true data=his_temp.txt")
5	=A1.execute("**delete from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
6	>A1.close()
7	=movefile(file("his_temp.txt"))

Oracle的数据加载工具sqlldr需要一个控制文件（his_temp.ctl）来指定数据如何导入，内容如下：

```
LOAD DATA APPEND INTO TABLE SALES
(ORDERID terminated by '|',
CLIENT terminated by '|',
SELLERID terminated by '|',
AMOUNT terminated by '|',
ORDERDATE terminated by '|')
```

注意：代码中先将历史数据导出成文件，再使用数据库的数据加载工具是因为jdbc的insert效率很差，比较当前做法，往往慢上几十倍。

数据库作为历史库——冷导出

友乾营

允许一段“时间窗口”内从生产库取出历史数据追加导出至数据库

每天的“时间窗口”内做的处理 ：

	A
1	=connect("mysql")
2	=A1.cursor@x("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))
3	=file("his_temp_yday.txt").export(A2;"|")
4	=system("sqlldr user/password@ORCLPDB control=his_temp.ctl direct=true data=his_temp_yday.txt")
5	=A1.execute("**delete from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))
6	>A1.close()
7	=movefile(file("his_temp_yday.txt"))

Oracle的数据加载工具sqlldr需要一个控制文件（his_temp.ctl）来指定数据如何导入，内容如下：

```
LOAD DATA APPEND INTO TABLE SALES
(ORDERID terminated by '|',
CLIENT terminated by '|',
SELLERID terminated by '|',
AMOUNT terminated by '|',
ORDERDATE terminated by '|')
```

数据库作为历史库——热导出

所谓热导出，是相对于冷导出而言的。热导出要保证查询系统永不停机，在导出数据的过程中有查询请求进来，依然能够工作。热导出一般适用于实时查询场景要求较高的情况。

数据库作为历史库的热导出实现思路：

1、初始化工作：

a) 将昨天零点前的所有数据导出至历史库；
b) 导出完成后，删除生产库中所有已导出的冗余数据；
c) 建立一个“时间节点表”（timenode）包含一个日期字段（timenode），用于区分查询的取数时间段，并赋初值（日期应为昨天）。

2、每天的定时导出任务中：

a) 将昨天一整天的数据导出至历史库；
b) 导出完成后，删除生产库中昨天一整天的冗余数据；
c) 再将“时间节点表”（timenode）的日期改为今天即可。

数据库作为历史库的热导出面临的问题：

利用关系型数据库的事务一致性，化解高并发时面临的导出数据期间，因“时间节点”变化造成查询数据可能有误的问题。但查询数据阶段需要对常规分库方案做一些更改。

数据库作为历史库——热导出

所谓热导出，是相对于冷导出而言的。热导出要保证查询系统永不停机，在导出数据的过程中有查询请求进来，依然能够工作。热导出一般适用于实时查询场景要求较高的情况。

生产库（MySQL），历史库（Oracle），初始化历史库：

	A
1	=connect("mysql")
2	=A1.cursor@x("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
3	=file("his_temp.txt").export(A2;"|")
4	=system("sqlldr user/password@ORCLPDB control=his_temp.ctl direct=true data=his_temp.txt")
5	=A1.execute("**delete from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
6	>A1.close()
7	=movefile(file("his_temp.txt"))
8	=connect("oracle")
9	=A8.execute("create table timenode(timenode date)")
10	=A8.execute("insert into timenode values (?)",after(date(now()),-1))
11	>A8.close()

数据库作为历史库——热导出

所谓热导出，是相对于冷导出而言的。热导出要保证查询系统永不停机，在导出数据的过程中有查询请求进来，依然能够工作。热导出一般适用于实时查询场景要求较高的情况。

每天的定时导出任务中做的处理 ：

	A
1	=connect("mysql")
2	=A1.cursor@x("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))
3	=file("his_temp_yday.txt").export(A2;"|")
4	=system("sqlldr user/password@ORCLPDB control=his_temp.ctl direct=true data=his_temp_yday.txt")
5	=A1.execute("**delete from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))
6	>A1.close()
7	=movefile(file("his_temp_yday.txt"))
8	=connect("oracle")
9	=A8.execute("update timenode set timenode=?",date(now()))
10	>A8.close()

数据库作为历史库——热导出

所谓热导出，是相对于冷导出而言的。热导出要保证查询系统永不停机，在导出数据的过程中有查询请求进来，依然能够工作。热导出一般适用于实时查询场景要求较高的情况。

查询时，需要根据“时间节点表”中的时间，区分“历史”与“新”数据的时间区间

	A	B	C
1	=real=connect("mysql")	=his=connect("oracle")	=timenode=his.query@1("select * from timenode")
2	=SQL_real="**select** sellerid, **sum**(amount) totalamount, **count**(amount) countamount,**max**(amount) maxamount,**min**(amount) minamount **from** sales **where** orderdate **>=**? **group by** sellerid"		
3	=SQL_his="**select** sellerid, **sum**(amount) totalamount, **count**(amount) countamount,**max**(amount) maxamount,**min**(amount) minamount **from** sales **where** orderdate **<**? **group by** sellerid"		
4	=real.cursor@x(SQL_real,timenode)		
5	=his.cursor@x(SQL_his,timenode)		
6	=[A4,A5].conjx().groups(sellerid;sum(totalamount):totalamount,sum(countamount):countamount,max(maxamount):maxamount,min(minamount):minamount)		
7	=A6.new(sellerid,totalamount,countamount,if(countamount==0,null,round(totalamount/countamount)):avgamount,maxamount,minamount)		

目录 CONTENTS

友乾营

专注数据技术的社群

文件作为历史库

友乾营

数据库与文件作为历史库的差异（优缺点）比较

数据库作为历史库：

优点：关系型数据库支持事务一致性，数据写入的同时仍然可以很好地支持查询。

缺点：牺牲一部分性能，当每天导出的数据量较多时对资源占用相当巨大（因为数据库回滚段会很大）

文件作为历史库：

优缺点与数据库相反，所以一致性和高性能在一定程度上是矛盾的。

文件作为历史库——冷导出

允许一段“时间窗口”内从生产库取出历史数据追加导出至文件

文件作为历史库的冷导出实现思路：

1、初始化工作：

a) 将昨天零点前的所有数据导出至历史库；

b) 导出完成后，删除生产库中所有已导出的冗余数据；

2、每天的“时间窗口”中：

a) 停止提供T+0查询服务；

b) 将昨天一整天的数据以追加方式导出至文件；

c) 导出完成后，删除生产库中昨天一整天的冗余数据；

d) 恢复T+0查询服务。

文件作为历史库的冷导出面临的问题：

每天的“时间窗口”内，T+0查询暂停服务，无法满足不停机业务。在T+0查询服务期间，由于生产库与历史库的数据不存冗余情况，所以查询可以按照上一页所讲的常规分库场景处理。

文件作为历史库——冷导出

允许一段“时间窗口”内从生产库取出历史数据追加导出至文件

生产库（MySQL），历史库（文件），初始化历史库：

	A
1	=connect("mysql")
2	=A1.cursor@x("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
3	=file("his_sales.btx").export@b(A2)
4	=A1.execute("**delete from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
5	>A1.close()

代码中使用集算器的集文件作为历史库。集文件中的数据经过压缩处理，适用于高效地查询访问场景。

文件作为历史库——冷导出

允许一段“时间窗口”内从生产库取出历史数据追加导出至数据库

每天的“时间窗口”内做的处理 ：

	A
1	=connect("mysql")
2	=A1.cursor@x("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))
3	=file("his_sales.btx").export@ab(A2)
4	=A1.execute("**delete from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))
5	>A1.close()

集文件的追加导出实现简单，只需要使用@a选项即可完成文件数据的追加工作。

文件作为历史库——冷导出

允许一段“时间窗口”内从生产库取出历史数据追加导出至数据库

文件与数据库的混合查询：

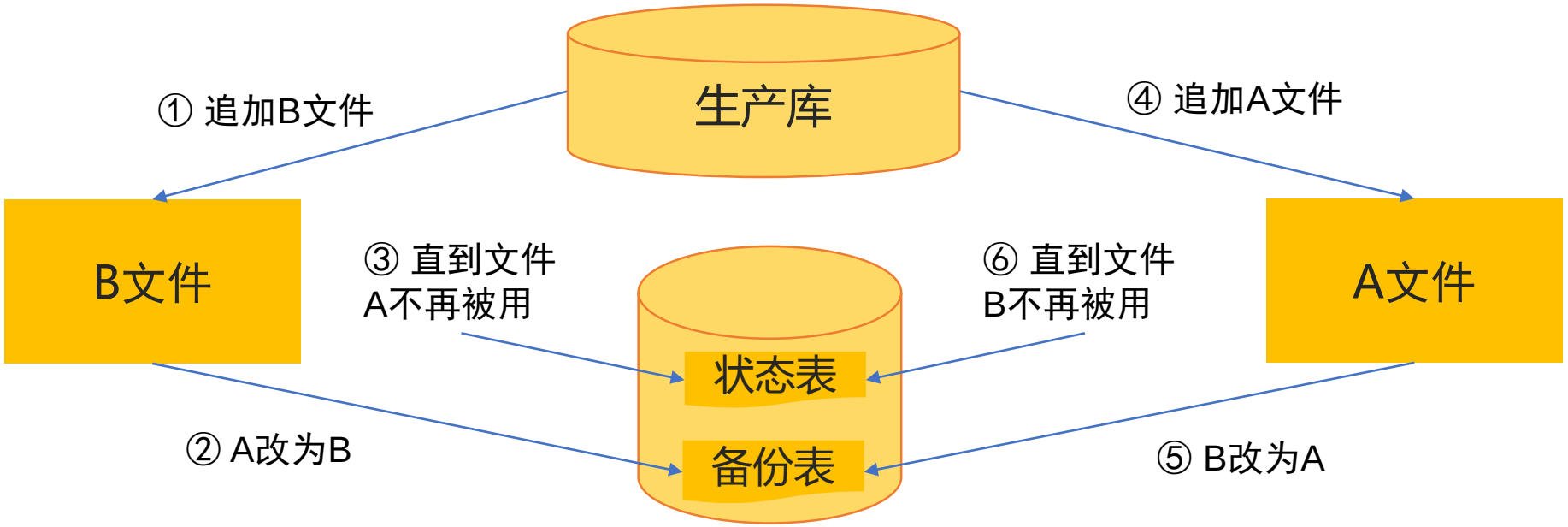
	A	B	C
1	=real=connect("mysql")	=his=connect("esprocODBC")	
2	=real.cursor@x("**select** sellerid, **sum**(amount) totalamount, **count**(amount) countamount,**max**(amount) maxamount,**min**(amount) minamount **from** sales **group by** sellerid")		
3	=his.cursor("**select** sellerid, **sum**(amount) totalamount, **count**(amount) countamount,**max**(amount) maxamount,**min**(amount) minamount **from** his_sales.btx **group by** sellerid")		
4	=[A2,A3].conjx().groups(sellerid;sum(totalamount):totalamount,sum(countamount):countamount,max(maxamount):maxamount,min(minamount):minamount)		
5	=A6.new(sellerid,totalamount,countamount,if(countamount==0,null,round(totalamount/countamount)):avgamount,maxamount,minamount)		

文件作为历史库——热导出

文件热导出方案中，由于文件无法同时读写，需要两套文件切换使用

文件作为历史库的热导出的难点：

在追加数据导出到历史文件的这段时间里，文件不可读，查询也就无法进行。冷导出在“时间窗口”内停止了查询业务，不存在这个问题。但要提供不停机查询业务，就需要两份历史文件来切换使用。



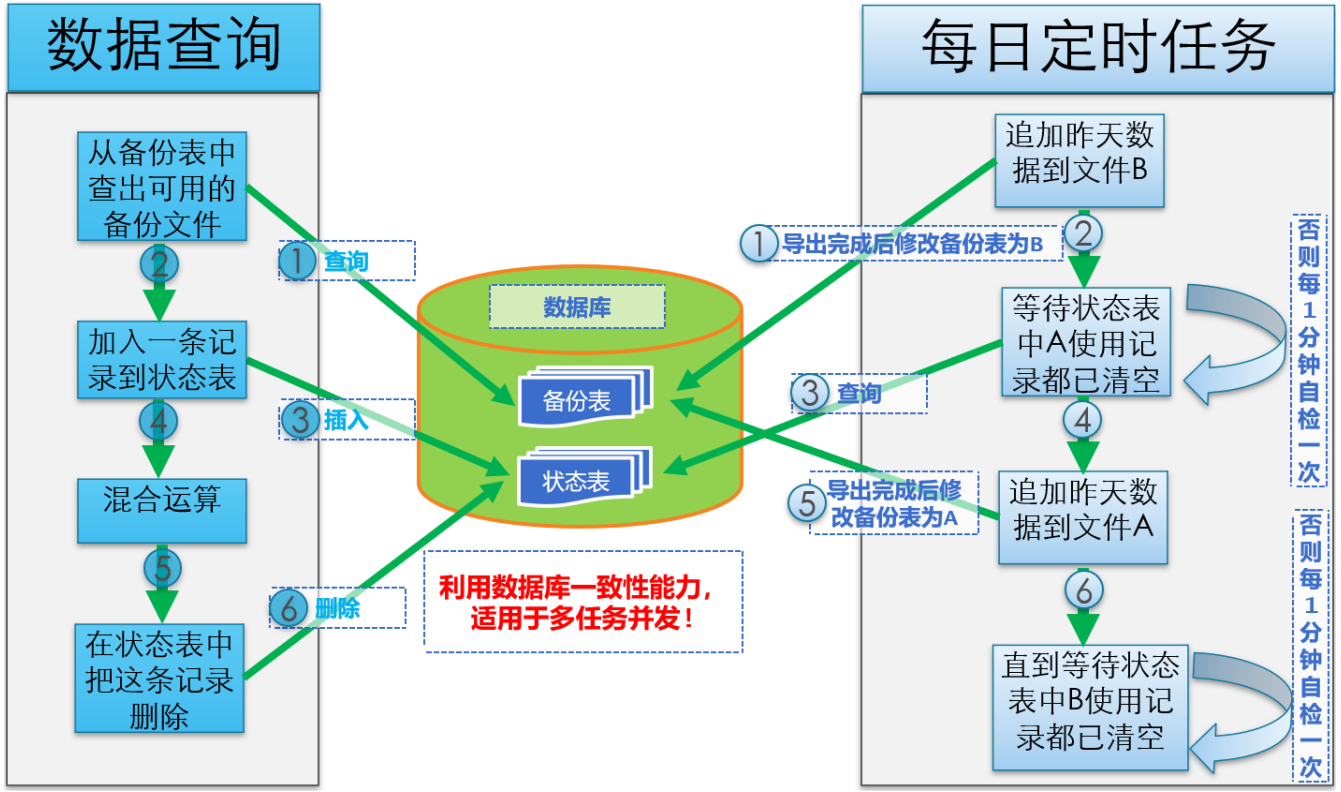
注意：备份表、状态表必须用数据库作为媒介，从而利用数据库的一致性；不能用文件记录备份表、状态表的内容，因为文件无法保持一致性，当多任务并发时可能就乱了。

文件作为历史库——热导出

友乾营

文件热导出方案中，由于文件无法同时读写，需要两套文件切换使用

热导出需要利用文件的备份机制结合数据库的一致性来实现热切换动作，流程如下：



文件作为历史库——热导出——初始化

在数据库中，定义”备份表”（backup），包含三个字段(文件名称/边界时间/标识)；定义”查询状态表”（status），包含三个字段(唯一标识/文件名称/当前系统时间，其中定义唯一标识为自增列)，数据结构分别如下图示：

Table: backup

Column Information

Field	Type	Comment
name	varchar(10)	
crashtime	datetime	
flag	varchar(20)	

Table: status

Column Information

	Field	Type	Comment
🔑	uniques	bigint(20)	
	name	varchar(10)	
	time	datetime	

Index Information

	Indexes	Columns	Index_Type
🔑	PRIMARY	uniques	Unique

DDL Information

create table

```
CREATE TABLE `status` (
  `uniques` bigint(20) NOT NULL AUTO_INCREMENT,
  `name` varchar(10) DEFAULT NULL,
  `time` datetime DEFAULT NULL,
  PRIMARY KEY (`uniques`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1
```

注意：备份表、状态表必须用数据库作为媒介，从而利用数据库的一致性；不能用文件记录备份表、状态表的内容，因为文件无法保持一致性，当多任务并发时可能就乱了。

文件作为历史库——热导出——初始化

通过文件A备份一个文件B，然后在“备份表”中记录可查询的备份文件为A，并设定从数据库中取的热数据边界时间(定义为每日的零点)。

	A
1	=connect("mysql")
2	=A1.cursor@x("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
3	=file("his_sales_A.btx").export@b(A2)
4	=A1.execute("**delete from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')<?",after(date(now()),-1))
5	=movefile@c(file("his_sales_A.btx"),"his_sales_B.btx")
6	>A1.execute("insert into backup (name,crashtime,flag) values (?,?,?)","A",date(now()),"WORKING_STATUS")
7	>A1.close()

文件作为历史库——热导出——同步数据与热切换

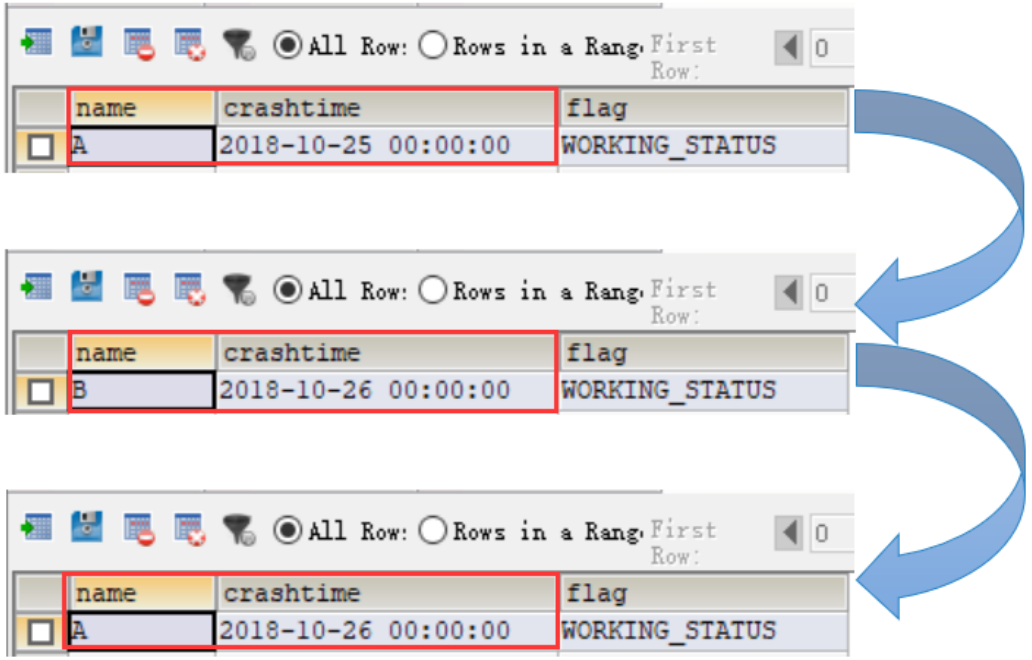
同步数据与热切换时，要注意状态表中记录，判断是否存在接下来要追加的数据文件还在被使用的情况，若存在，每隔1分钟检查一次状态，直到要追加的文件不再被使用了，再追加该文件。避免发生读写错误。

	A	B
1	=connect("mysql")	
2	=A1.cursor("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))	
3	=file("his_sales_B.btx").export@ab(A2)	
4	>A1.execute("**update** backup **set** name= ?,crashtime=? **where** flag='WORKING_STATUS'","B",date(now()))	
5	for connect("mysql").query@1x("**select count**(*) **from** status **where** name='A'")>0	>sleep(60*1000)
6	=A1.cursor("**select * from** sales **where** DATE_FORMAT(orderdate,'%Y-%m-%d')=?",after(date(now()),-1))	
7	=file("his_sales_A.btx").export@ab(A6)	
8	>A1.execute("**update** backup **set** name= ?,crashtime=? **where** flag='WORKING_STATUS'", "A",date(now()))	
9	for connect("mysql").query@1x("**select count**(*) **from** status **where** name='B'")>0	>sleep(60*1000)
10	>A1.close()	

文件作为历史库——热导出——同步数据与热切换

备份表记录变化详解：

当历史数据同步追加到文件B上，修改数据库“备份表”的记录为使用文件B，同时修改从数据库中取热数据的边界日期范围（上一页的A4格）。
当文件A的数据追加完成后，再修改数据库“备份表”的记录为使用文件A，以后新产生的查询就会再使用文件A（上一页的A8格）。



文件作为历史库——热导出——查询

以文件作为历史库，使用热导出方案的查询脚本如下：

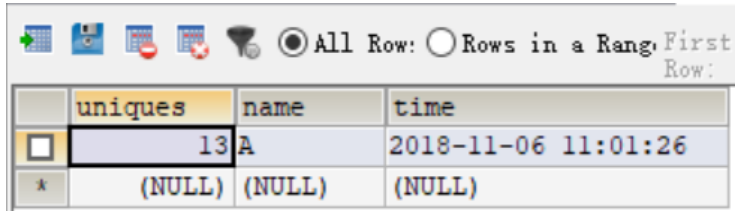
	A	B
1	=connect("mysql")	=connect("esprocODBC")
2	=A1.query@1("**select** name,crashtime **from** backup **where** flag**=**'WORKING_STATUS'")	=name=A2(1),crashtime=A2(2),filename="his_sales_"+name+".btx"
3	=A1.cursor@x("**select** sellerid, **sum**(amount) totalamount, **count**(amount) countamount,**max**(amount) maxamount,**min**(amount) minamount **from** sales **where** orderdate **>**? **group by** sellerid",crashtime)	>A1.execute("**insert into** status (name,**time**) **values** (?,?)",name,now()),uniques=A1.query@1("select @@identity")
4	=B1.cursor("**select** sellerid, **sum**(amount) totalamount, **count**(amount) countamount,**max**(amount) maxamount,**min**(amount) minamount **from** "+filename+" **where** orderdate **>**? **group by** sellerid",crashtime)	
5	=[A3,A4].conjx().groups(sellerid;sum(totalamount):totalamount,sum(countamount):countamount,max(maxamount):maxamount,min(minamount):minamount)	
6	=A5.new(sellerid,totalamount,countamount,if(countamount==0,null,round(totalamount/countamount)):avgamount,maxamount,minamount)	
7	>A1.execute("**delete from** status **where** uniques**=**?",uniques)	>A1.close()
8	return A6	

文件作为历史库——热导出——查询

友乾营

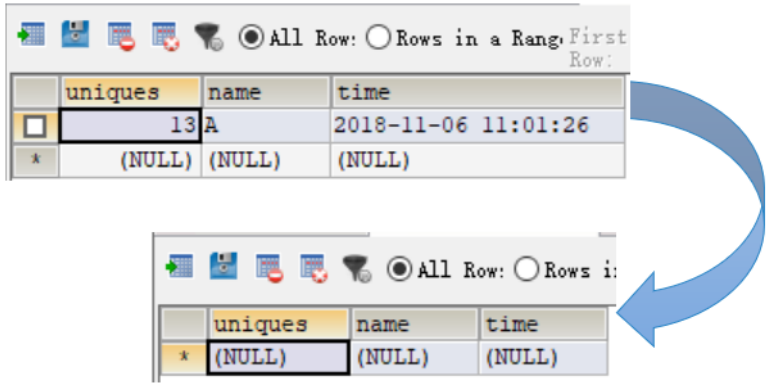
状态表记录变化详解：

查询开始，状态表写入一条记录，表明当前该备份文件正有一个查询，其中uniques为自增列，并需要记录该uniques，以便查询完成删除该记录（上一页的B3格）



	uniques	name	time
☐	13	A	2018-11-06 11:01:26
*	(NULL)	(NULL)	(NULL)

查询完成，根据变量uniques的值作为条件，在状态表中把这条记录删除（上一页的A7格）



	uniques	name	time
☐	13	A	2018-11-06 11:01:26
*	(NULL)	(NULL)	(NULL)

	uniques	name	time
*	(NULL)	(NULL)	(NULL)

润乾线上直销系统

玩转好多乾

http://www.raqsoft.com.cn/wx/hdq-strategy.html