bitError

昵称：bitError
园龄：1年9个月
粉丝：7
关注：1
+加关注

< 2019年3月 >

日	一	二	三	四	五	六
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

分布式系统(10)
Linux操作系统(8)
C++(7)
PostgreSQL(5)
分布式理论(4)
zookeeper源码(3)
计算机网络(1)
数据库(1)

随笔档案

2017年7月 (7)
2017年6月 (5)
2017年5月 (27)

最新评论

1. Re:Hive和SparkSQL： 基于Hadoop 的数据仓库工具
这篇写得最好。
--李博洋
2. Re:C++11新特性
C 爷牛逼，我辈楷模！
--李博洋

阅读排行榜

1. GreenPlum：基于PostgreSQL的分布式关系型数据库(7956)
2. Hive和SparkSQL： 基于 Hadoop的数据仓库工具(6998)
3. Effective C++读书笔记(6100)
4. Storm：分布式流式计算框架(4233)
5. PostgreSQL事务实现(3649)

评论排行榜

1. C++11新特性(1)
2. Hive和SparkSQL： 基于 Hadoop的数据仓库工具(1)

推荐排行榜

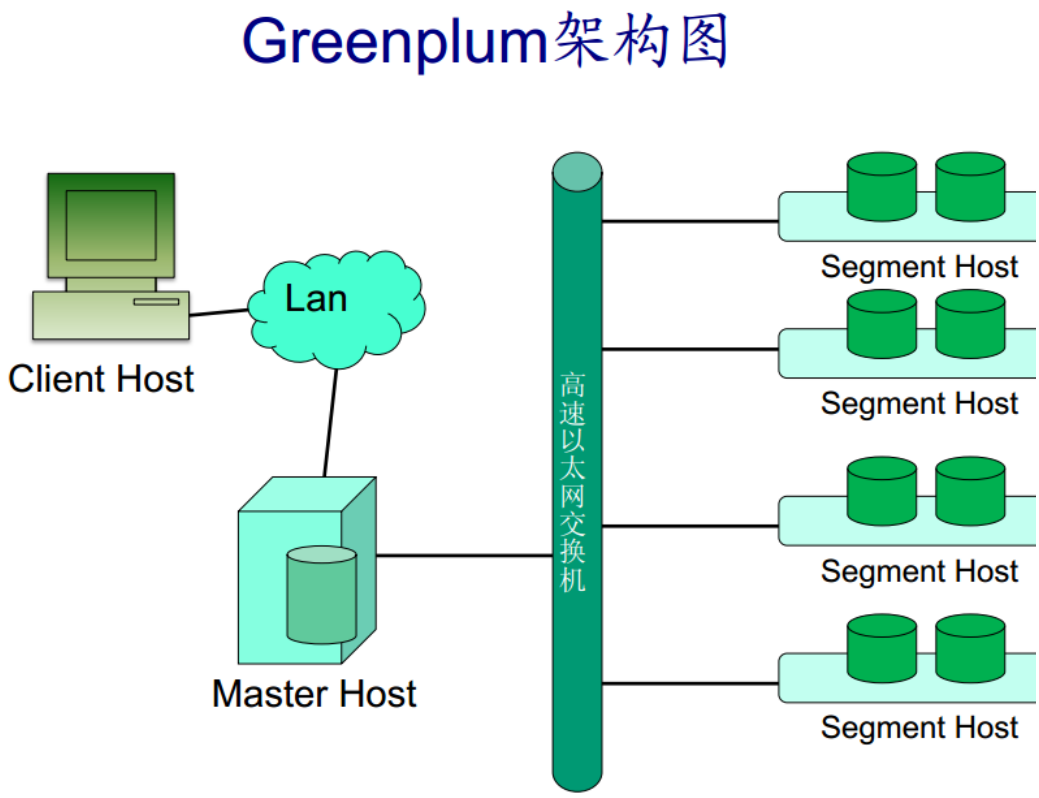
1. Yarn和Mesos：资源管理调度平台(2)
2. Effective C++读书笔记(1)

GreenPlum：基于PostgreSQL的分布式关系型数据库

GreenPlum是一个底层是多台PostgreSQL分表分库的分布式数据库，它有如下特点

- 支持标准SQL，几乎所有PostgreSQL支持的SQL，greenplum都支持
- 支持ACID、分布式事务
- 支持上百台集群(这一点有点不好，hadoop可以万台)

系统架构



Master Host

- 处理用户请求，生成执行计划，以及在执行计划执行必要的聚合操作(avg)或者排序
- 内部有一个PostgreSQL数据库，保存所有的元数据，索引信息
- 监控所有segment的状态信息

Segment host

- 每台Segment host有多个segment，一般segment等于core数
- segment是一个PostgreSQL数据库，负责存储具体数据

内部网络

GreenPlum内部使用udp网络，但是Greenplum会对数据包进行校验，因此可靠性等同于TCP。使用TCP的时候，最多支持1000个segment

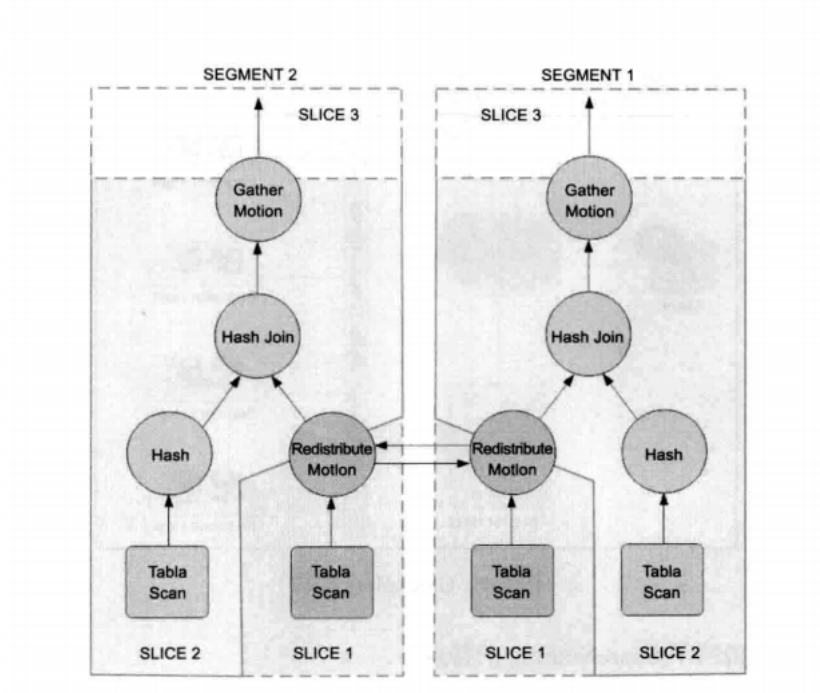
执行计划

当master接受到一条SQL语句，会将这条语句解析为执行计划DAG，将DAG中不需要进行数据交换的划分为slice，多表连接，aggerate，sort的时候，都会涉及到slice的重分布，会有一个motion任务来执行数据的重分布。将slice下发到涉及到的相关segment中。

我认为slice类似与Spark中的stage的概念，不需要进行数据shuffle

motion方式

- gather motion(N->1)：在master节点上把所有segment数据聚集起来，一般是sort，sort group，sort join
- boardcast motion(N->N)：每个segment把数据广播给其余所有segment
- redistribute motion(N->N)：每个segment把数据按照hash的方式重新分布



我们可以猜一猜上面的执行计划代表什么：A表和B表进行join连接，然后它们又进行sort或者聚合。

算子实现

索引

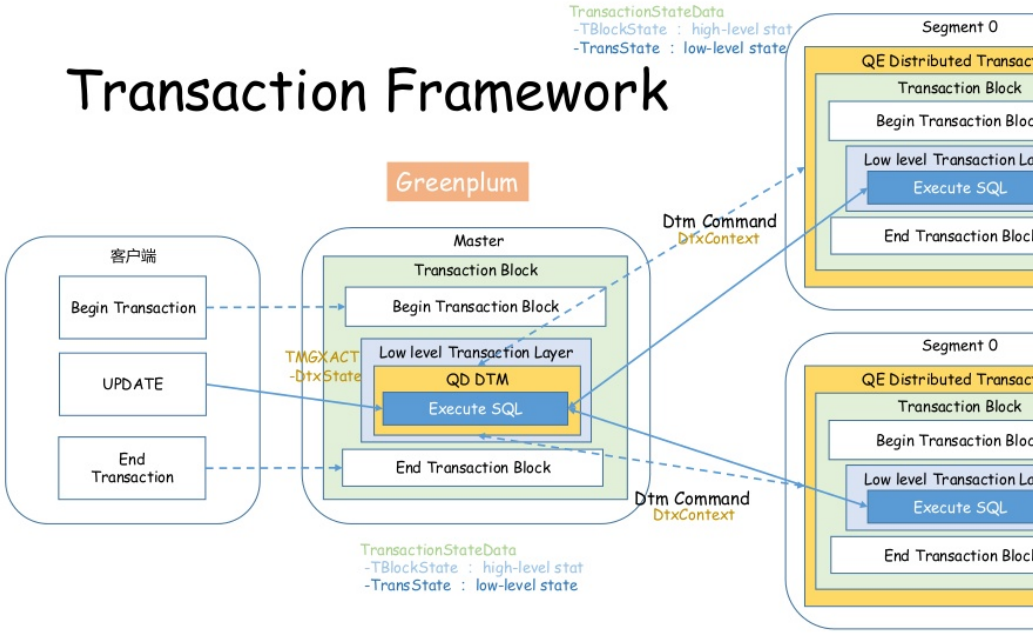
Greenplum支持所有postgresql的索引，另外还支持位图索引

Join方式

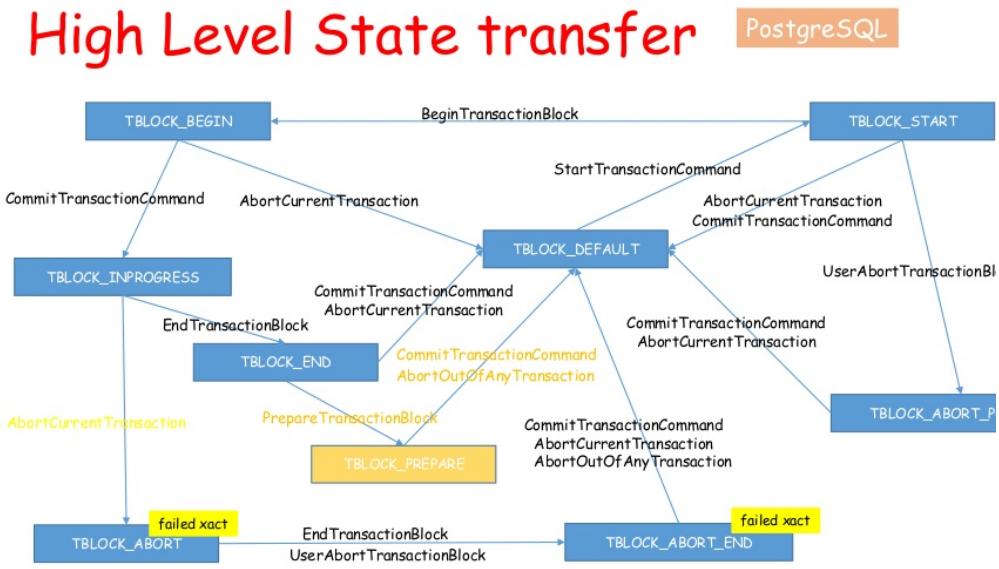
1. Hash join：
2. nestloop join：笛卡儿积必须nestloog join
3. merge join

分布式事务

Greenplum虽然是面向OLAP的数据库，但是也提供了插入，删除，更新数据的接口，利用两阶段提交协议支持分布式事务，提供强一致性，支持ACID，支持的隔离级别是(读已提交，可串行化)。



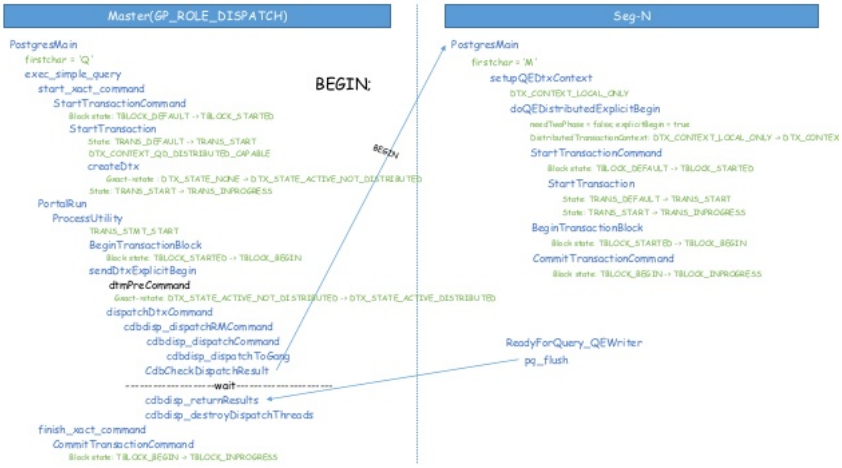
Greenplum采用和Postgresql类似的方式，上层事务块控制事务状态转换，底层事务负责执行具体的语句以及和相关segment交互。



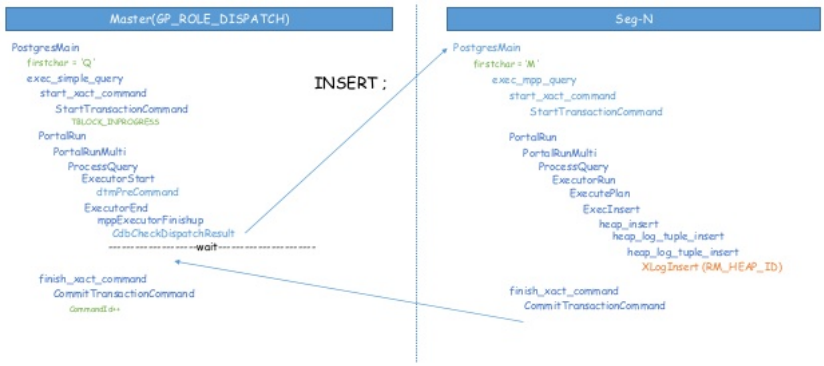
与单机事务相比，多了TBLOCK_PREPARE状态，代表两阶段提交协议中的中间状态。除此之外，分布式事务也有一套以DXT开头的分布式状态

例子

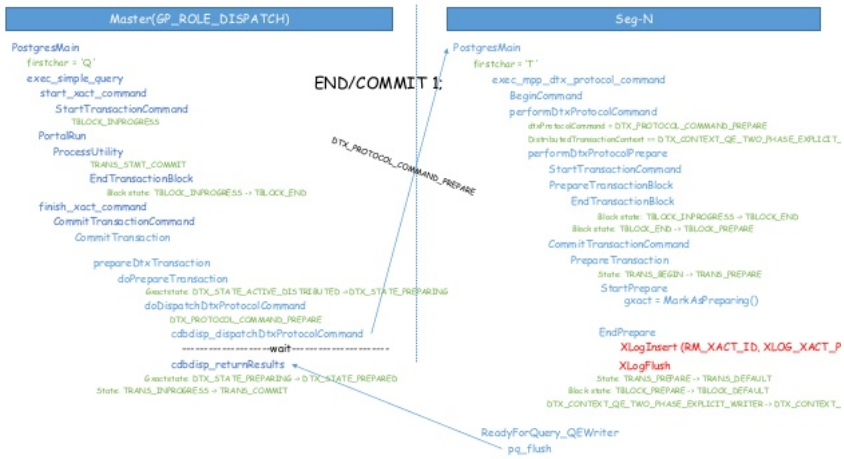
正常流程



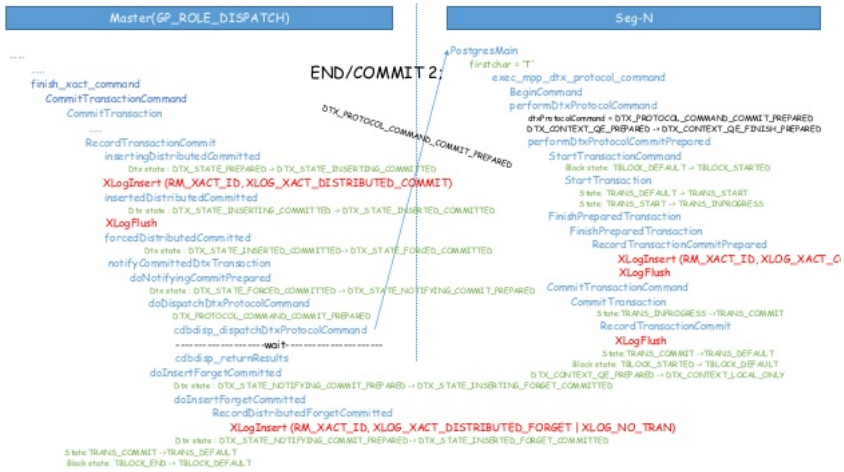
- 在所有segment都启动一个事务块，状态TBLOCK_BEGIN



- 执行一条插入语句，状态TBLOCK_INPRGRESS



- END命令，状态为DXT_STATE_PREPARED。这里master状态为TBLOCK_END，slave segment状态为TBLOCK_DEFAULT(初始状态)

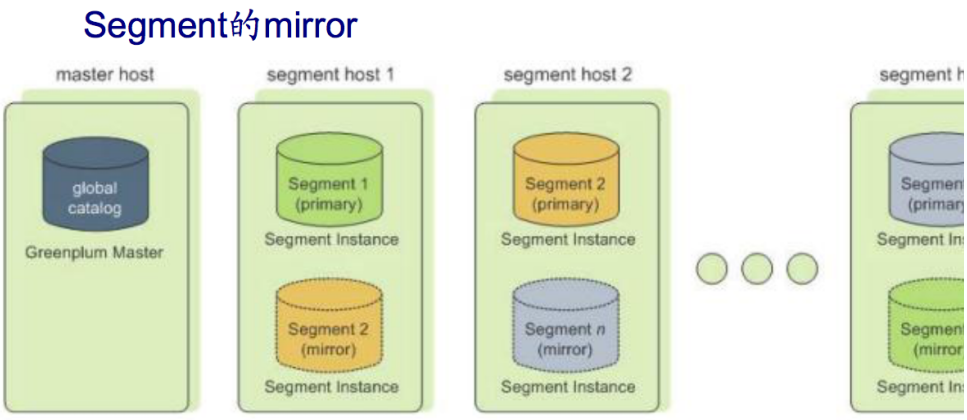


- 第二阶段，开始正式提交。DXT_STATE_PREPARED->DXT_STATE_INSGRETE_FORGET_COMMIT。master状态为TBLOCK_END->TBLOCK_DEFATULT，slave segmeng又重新经历一轮所有状态

容错

slave segment容错

每台segment都在其他机器上有备机



Primary Segment 与对应 Mirror Segment 之间的数据基于文件级别同步备份。Mirror Segment 不直接参与数据库事务和控制操作。

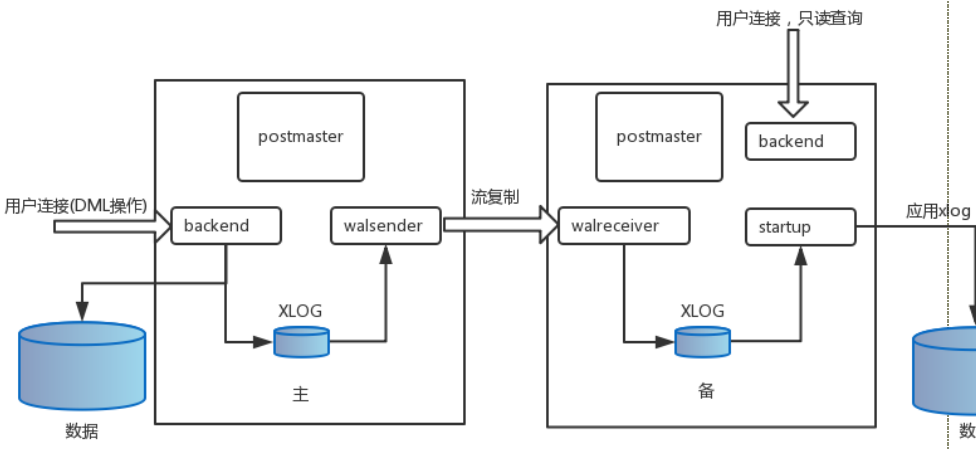
为什么采用文件同步的机制：mirror库数据直接获取primary的文件(日志文件)和数据(修改的数据页)。

恢复流程

发生宕机时，greenplum有两种恢复模式，"read-only"和"continue"。

- read-only：也就是说如果一个segment坏了，整个greenplum会变成只读，不能写了
- continue：由mirror正常提供服务，master节点会把新增数据记录下来，等待primary恢复后同步

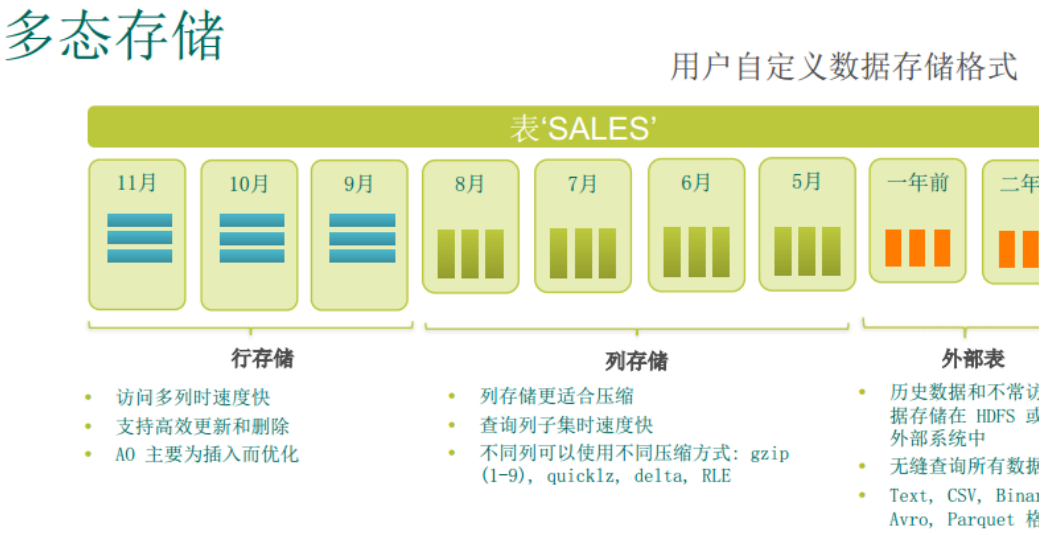
primary segment容错



基于数据流通过WAL同步，由postgresql提供的容错。

负载均衡和数据组织方式

数据组织方式



- 有一类特殊的表，称为append-only表，支持列存储，表压缩
- 通过gpfdist插件，可以支持外部表

负载均衡

Greenplum通过分布和分区的方式，使得庞大的数据分布在不同的segment上。严格来说，分布才是拆表，分区只是为了加快查询速度。

- 分布：是从物理上把数据分散到各个SEGMENT上，Greennplum提供hash函数
- 分区：segment内部按照规则将数据组织在一起

分布

1. hash分布：distributed by (column_name)，可以指定多个分布键。相同的hash值分布到同一个segment
2. 随机分布：distributed randomly，相同的记录可能分布到不同的segment

建议：

- 分布列尽量选择需要经常JOIN的列，这类查询的并发越高，越应该考虑
- 尽量选择分布均匀的列，或者多列
- 不要轻易使用随机分布

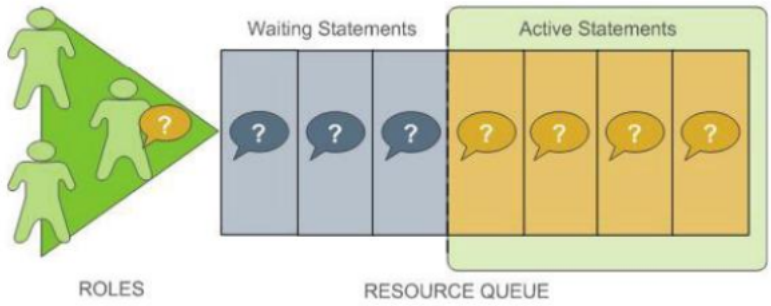
分区

1. range partition：按照数据的范围
2. list partition：按照List中的值
3. 多级分区

建议：

- 尽量选择和查询条件相关的字段，缩小QUERY需要扫描的数据
- 当有多个查询条件时，可以使用子分区，进一步缩小需要扫描的数据

资源控制



1. 限制正在执行的所以SQL的最大cost
2. 限制最多运行多少SQL
3. 控制正在运行的SQL的优先级

参考资料

Greenplum分布式事务，很详细

主从同步

标签: 分布式系统

bitError
关注 - 1
粉丝 - 7
+加关注

0 0

« 上一篇：Linux网络子系统
» 下一篇：Storm：分布式流式计算框架

posted on 2017-05-26 18:06 bitError 阅读(7959) 评论(0) 编辑 收藏