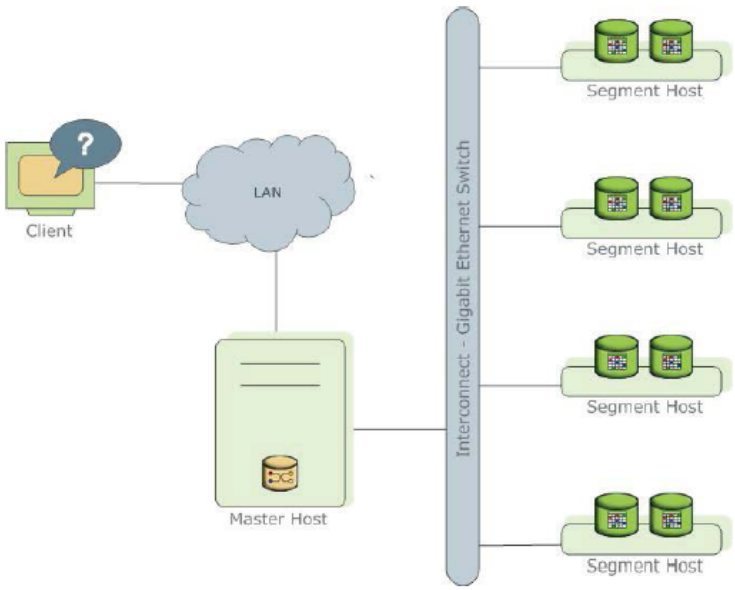
ArthurQin

Greenplum 的分布式框架结构

Greenplum 的分布式框架结构

1.基本架构

Greenplum（以下简称 GPDB）是一款典型的 Shared-Nothing 分布式数据库系统。GPDB 拥有一个中控节点（ Master ）统筹整个系统，并在整个分布式框架下运行多个数据库实例（ Segment ）。Master 是 GPDB 系统的访问入口，其负责处理客户端的连接及 SQL 命令、协调系统中的其他 Segment 工作，Segment 负责管理和处理用户数据。而每个 Segment 实际上是由多个独立的 PostgreSQL 实例组成，它们分布在不同的物理主机上，协同工作。

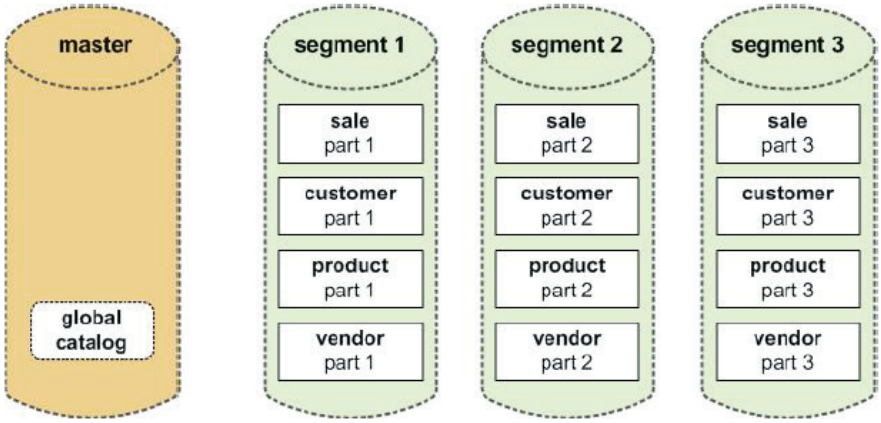


主节点与子节点

GPDB中，数据通过复杂的HASH 算法或随机拆分成无重叠的记录集合，分布到所有 Segment 上。仅 Master 完成与用户和客户端程序的直接交互。因此但对于用户来说，使用 GPDB 系统如同使用一个单机数据库。

Master上存储全局系统表（Global System Catalog ），但不存储任何用户数据，用户数据只存储在 Segment 上。Master 负责客户端认证、处理 SQL 命令入口、在Segment 之间分配工作负、整合 Segment 处理结果、将

最终结果呈现给客户端程序。



用户 Table 和相应的 Index 都分布在 GPDB 中各 Segment 上，每个 Segment 只存储其中属于本节点的那部分数据。用户不能够直接跳过 Master 访问 Segment，而只能通过 Master 来访问整个系统。在 GPDB 推荐的硬件配置环境下，每个有效的 CPU 核对应一个 Segment，比如一台物理主机配备了2个双核的 CPU，那么每个主机配置4个主实例（Segment Primary）。

网络链接

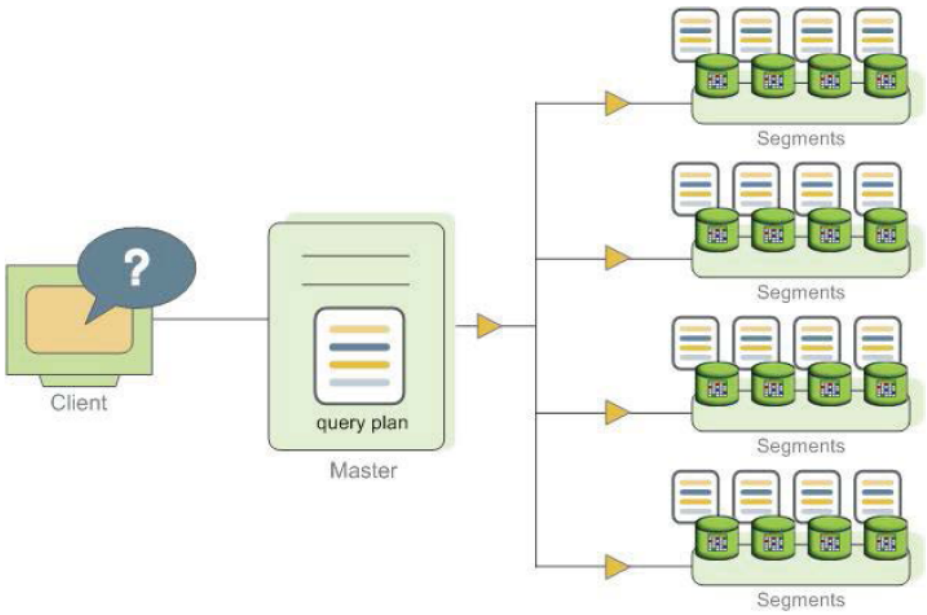
网络层组件（Interconnect）是 GPDB的重要组件。在用户执行查询时，每个 Segment 都需要执行相应的处理，因此物理主机间需要进行控制信息和数据的高效传递。网络层的作用就是实现物理主机之间的通信、数据传递，以及备份。在默认情况下，网络层使用 UDP 协议。GPDB 自己会为 UDP 协议做数据包校验，其可靠性与 TCP 协议一致，但其性能和扩展性远好于TCP协议。

2.查询执行机制

系统启动后，用户通过客户端程序（例如 psql）连接到的 Master 主机并提交查询语句。GP 会创建多个 DB 进程来处理查询。在 Master 上的称为执行分发器（Query Dispatcher/QD）。QD 负责创建、分发查询计划，汇总呈现最终结果。在 Segment 上，处理进程被称为查询执行器（Query executor/QE）。QE负责完成自身部分的处理工作以及与其他处理进程之间交换中间结果。

查询计划生成与派发

查询被 Master 接收处理（QD身份）。QD 将查询语句依据所定义的词法和语法规则创建原始查询语法树。接着在查询分析阶段，QD 将原始语法树转换为查询树。然后进入查询改写阶段，QD 将查询树依据系统中预先定义的规则对查询树进行转换。QD 最终调用优化器接受改写后的查询树，并依据该查询树完成查询逻辑优化和物理优化。GPDB 是基于成本的优化策略：评估若干个执行计划，找出最有效率的一个。但查询优化器必须全局的考虑整个集群，在每个候选的执行计划中考虑到节点间移动数据的开销。至此 QD 创建一个并行的或者定向的查询计划（根据查询语句决定）。之后Master将查询计划分发到相关的 Segment 去执行，每个 Segment 只负责处理自己本地的那部分数据操作。大部分的操作—比如扫表、关联、聚合、排序都是同时在 Segment 上并行被执行。每个具体部分都独立于其他 Segment 执行（一旦执行计划确定，比如有 join，派发后 join 是在各个节点分别进行的，本机只和本机的数据 join）。



查询执行

2014年3月(1)
2013年12月(1)

阅读排行榜

1. Greenplum 源码安装教程 —— 以 CentOS 平台为例(9493)
2. Greenplum 的分布式框架结构(3685)
3. 为 Greenplum 增加 Zstandard 压缩功能(1317)
4. 一种安全云存储方案设计（下）——基于Lucene的云端搜索与密文基础上的模糊查询(1065)
5. 一种安全云存储方案设计（上）——基于二次加密的存储策略与加密图文混合检索(732)

评论排行榜

1. Greenplum 源码安装教程 —— 以 CentOS 平台为例(3)

推荐排行榜

1. Greenplum 源码安装教程 —— 以 CentOS 平台为例(5)

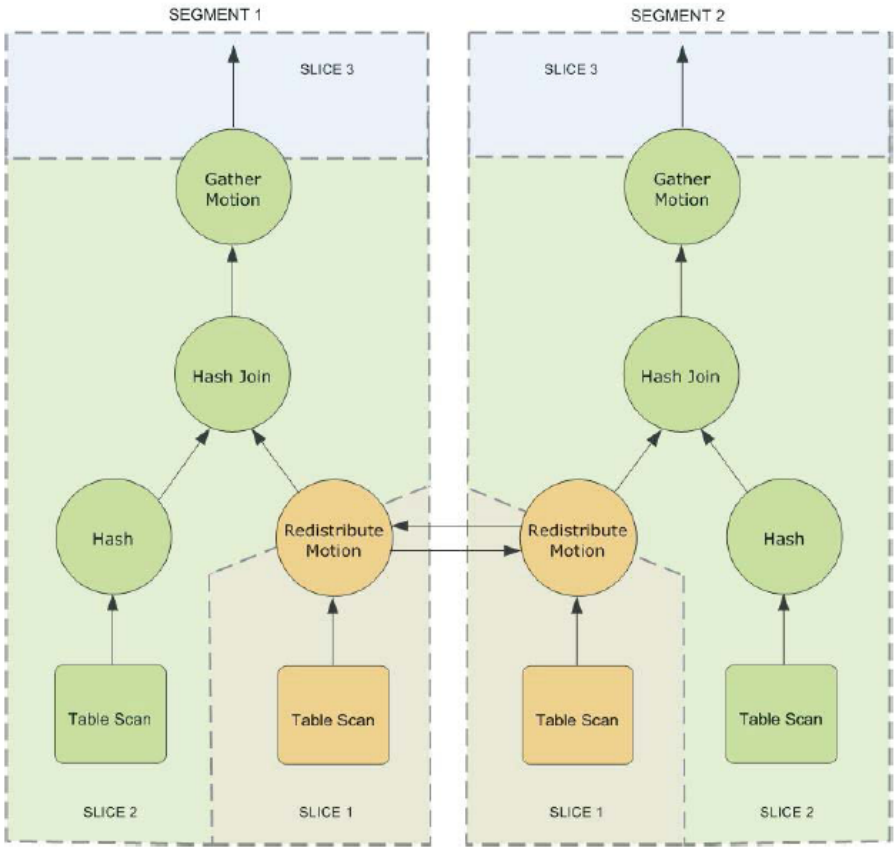
最新评论

1. Re:Greenplum 源码安装教程 —— 以 CentOS 平台为例
多谢楼主分享 帮了大忙
--shysky123
2. Re:Greenplum 源码安装教程 —— 以 CentOS 平台为例
@ 孤鸿寄语@ 孤鸿寄语谢谢回复，第一个空格的问题已修正。第二个问题也是我大意了，之前是安在那里的，后来改地址了。已修正，非常感谢。...
--ArthurQin
3. Re:Greenplum 源码安装教程 —— 以 CentOS 平台为例
修正1:gpscp -f /home/gpadmin/conf/seg_hosts /home/gpadmin/gp.tar =:/home/gpadmin(=:前应有空格)修正2: source /...
--孤鸿寄语

由于 GPDB 采用 Shared-Nothing 架构，为了最大限度的实现并行化处理，当节点间需要移动数据时，查询计划将被分割，最终一个查询会分为多个切片（ slice ），每个切片都涉及不同处理工作。即：先执行一步分操作，然后执行数据移动，再执行下一步分操作。在查询执行期间，每个 Segment 会根据查询计划上 slice 的划分，创建多个 postgres 工作进程，并行的执行查询。每个 slice 对应的进程只处理属于自己部分的工作，且这些处理工作仅在本 Segment 上执行。slice 之间为树形结构，其整体构成整个查询计划。不同 Segment 之间对应的查询计划上的同一个 slice 处理工作称为一个簇（ gang ）。在当前 gang 上的工作完成后，数据将向上传递，直到查询计划完成。Segment之间的通信涉及到 GPDB 的网络层组件（ Interconnect ）。

QE 为每个 slice 开启独立进程，在该进程内执行多个操作。每一步代表着特定的 DB 操作，比如：扫表、关联、聚合、排序等。Segment 上单个 slice 对应进程的执行算子从上向下调用，数据从下向上传递。

与典型的 DB 操作不同的是，GPDB 有一个特有的算子：移动（ motion ）。移动操作涉及到查询处理期间在 Segment 之间移动数据。motion 分为广播（ broadcast ）和重分布（ redistribute motion ）两种。正是 motion 算子将查询计划分割为一个个 slice ，上一层 slice 对应的进程会读取下一层各个 slice 进程广播或重分布的数据，然后进行计算。



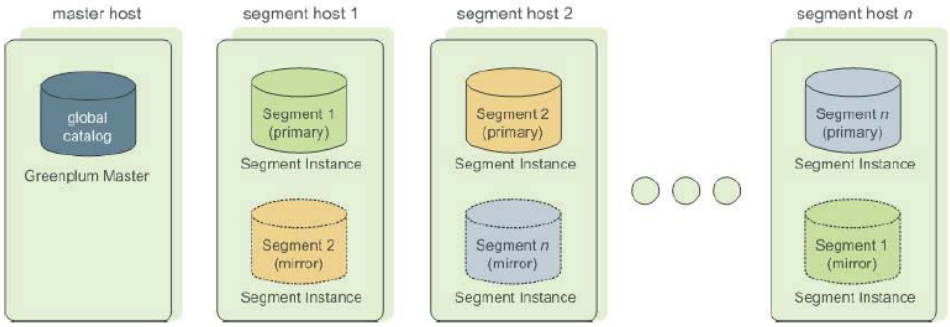
Greenplum 同 PostgreSQL 一样，采用元组流水方式获取和处理数据。我们按需取出单条元组，在处理本条元组后，系统将会取出下一条满足条件的元组，直到取出所有满足条件的元组为止。slice 间的 motion 操作同样以元组为单位收发数据，并通过下层 slice 缓冲构成生产消费模型，但不会阻断整个查询的流水。最终，各 Segment 的查询结果同样通过 motion 传给 Master，Master 完成最终处理后返回查询结果。

3.容错机制

节点镜像与故障容错

GPDB 支持为 Segment 配置镜像节点，单个Primary Segment 与对应的 Mirror Segment 配置在不同的物理主机上。同一物理主机可同时混合装载多个对应不同实例的 Primary Segment 和 Mirror Segment 。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。



当 Primary Segment 不可访问时，系统会自动切换到其对应的 Mirror Segment 上，此时，Mirror Segment 取代 Primary Segment 的作用。只要剩余的可用 Segment 能够保证数据完整性，在个别 Segment 或者物理主机宕机时，GPDB系统仍可能通过 Primary/Mirror 身份切换，来保持系统整体的可用状态。

其具体切换过程是，每当 Master 发现无法连接到某 Primary Segment 时（FTS系统），会在 GPDB 的系统日志表中标记失败状态，并激活/唤醒对应的 Mirror Segment 取代原有的 Primary Segment 继续完成后续工作。失败的 Primary Segment 可以等恢复工作能力后，在系统处于运行状态时切换回来。

扩展阅读

Greenplum Database Administrator Guide
Greenplum 源码安装教程

标签: greenplum , postgres , mpp , 并行 , 分布式 , 容错 , 数据库

« 上一篇： 读书笔记——寻找道德
» 下一篇： 读书笔记——泪与笑间寻正念