

# Greenplum

## 开源MPP数据仓库介绍

李晓亮 <adlee@vmware.com>

Greenplum工程师、内核团队经理

# Agenda

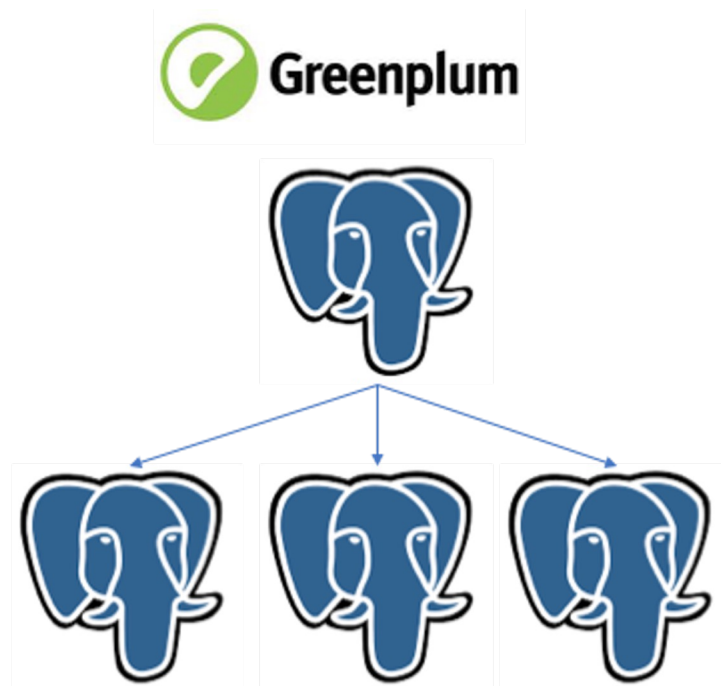
- Greenplum简介
- Greenplum的MPP架构
- 分布式优化器: Postgres planner 和 ORCA
- 分布式事务和执行
- Greenplum存储
- Greenplum生态
- Greenplum 7

# Greenplum简介：什么是Greenplum？

基于PostgreSQL、开源、分布式MPP、ACID完备、为OLAP优化的关系型数据仓库。

<https://greenplum.org>

<https://github.com/greenplum-db/gpdb>



# Greenplum的历史



- 2003年，Luke Loneragan 和 Scott Yara 发起 Greenplum项目，从 PostgreSQL 8 分支，做成 MPP架构
- 2010年被EMC收购
- 2012年成为Pivotal的一部分
- 2015年开源，可能是世界上第一个成熟商用的开源 MPP数据仓库
- 2019年底跟随Pivotal被VMware收购

# 谁在用Greenplum?

- 500多付费企业客户
- 成千上万的开源用户
- 支撑巨大的生产集群:
  - ❑ 250+ servers
  - ❑ 10+ PetaBytes
- 十几个甚至几十个国内国外的衍生项目（我们是真开源，欢迎大家贡献）

## Federal & Government Services



## Financial Services



## Industrial Manufacturing



## Telco & Media



# Greenplum的MPP架构

## ➤ Massively:

- ❑ PB级的数据，单台主机无法处理
- ❑ 所以数据分布在多个主机上
- ❑ 高效、灵活的数据分布，和实际业务相关

## ➤ Parallel:

- ❑ 数据并行处理计算
- ❑ 通过网络进行数据交换和汇总

# 执行架构

## ➤ Coordinator:

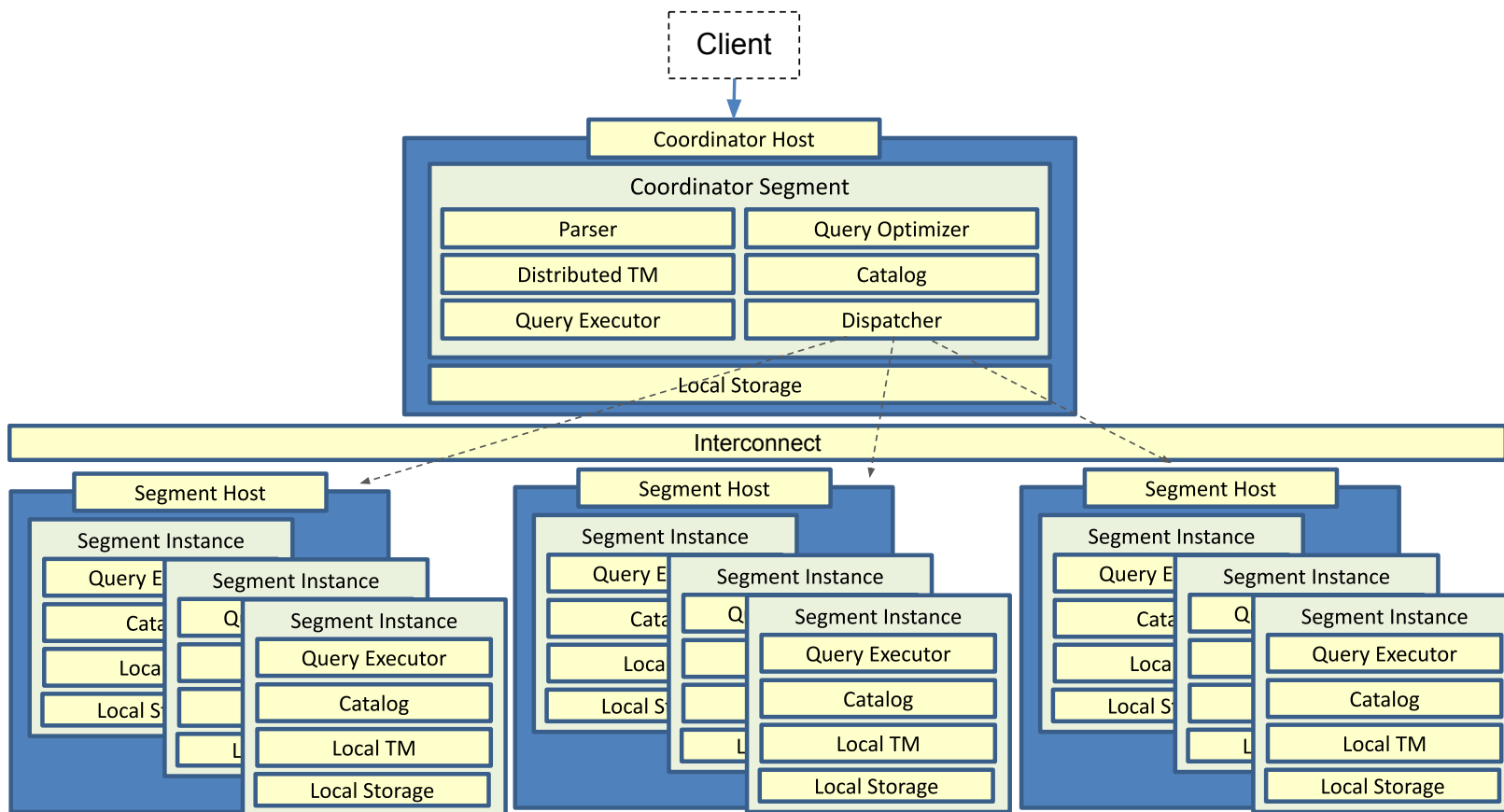
- ❑ 管理其它节点
- ❑ 生成分布式计划
- ❑ 下发计划和汇总结果
- ❑ 管理分布式事务

## ➤ Segments:

- ❑ 存储数据, share-nothing
- ❑ 产生计算进程

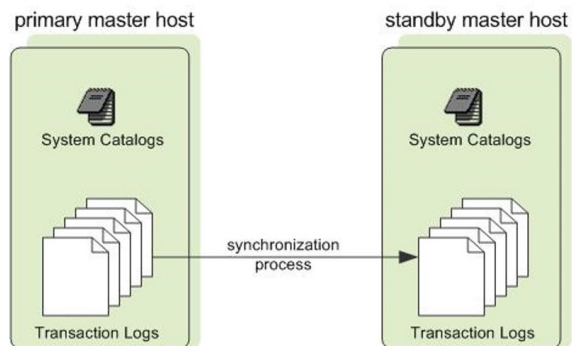
## ➤ Libpq: 控制信道

## ➤ Interconnect: 数据交换信道



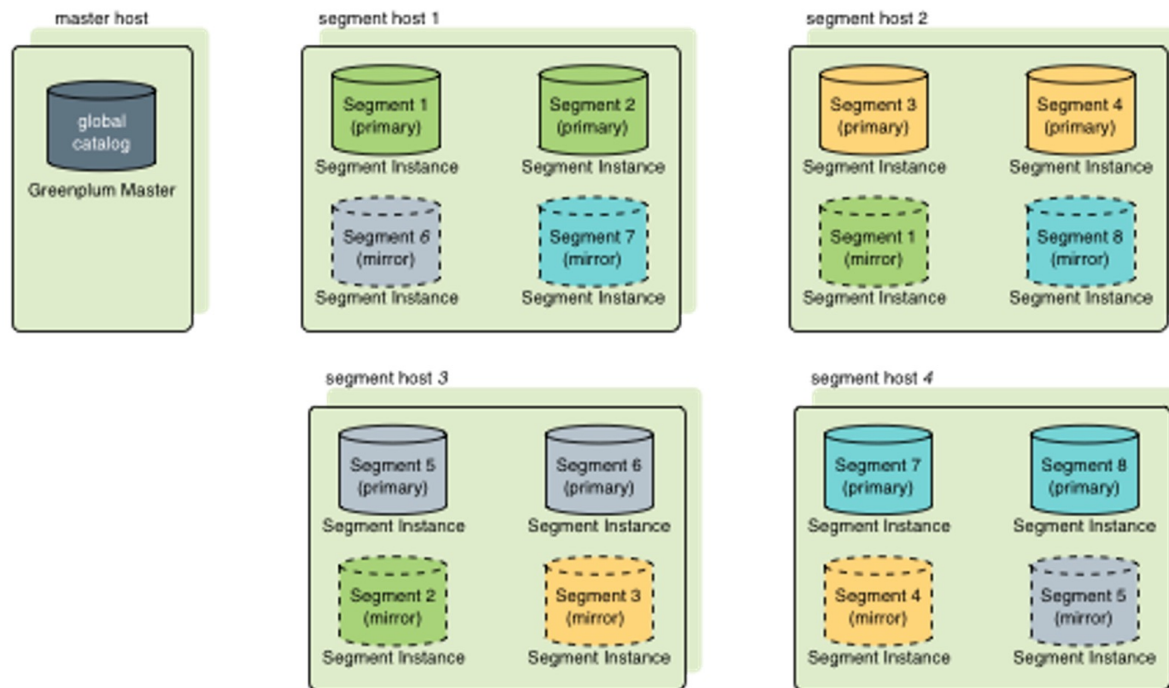
# Greenplum的高可用

## Coordinator View



- 数据存两份，Coordinator有standby
- 自动同步数据 (WAL replication)
- 自动灾难恢复 (FTS，主备切换)

## Segment View





# 分布式优化器：OLAP

- OLTP系统的SQL语句相对简单（CURD）
- OLAP系统的SQL语句就复杂得多（OLTP则尽量避免）
  - ❑ Join 很复杂(多表, outer join, lateral...)
  - ❑ 子查询、子链接
  - ❑ 聚集 (grouping sets, 多阶段聚集...)
  - ❑ 窗口函数, (Recursive) CTE
  - ❑ Procedure Languages（Python, R, Perl.....）
- 优化器非常非常重要
- 基于规则优化和基于代价优化

# ORCA

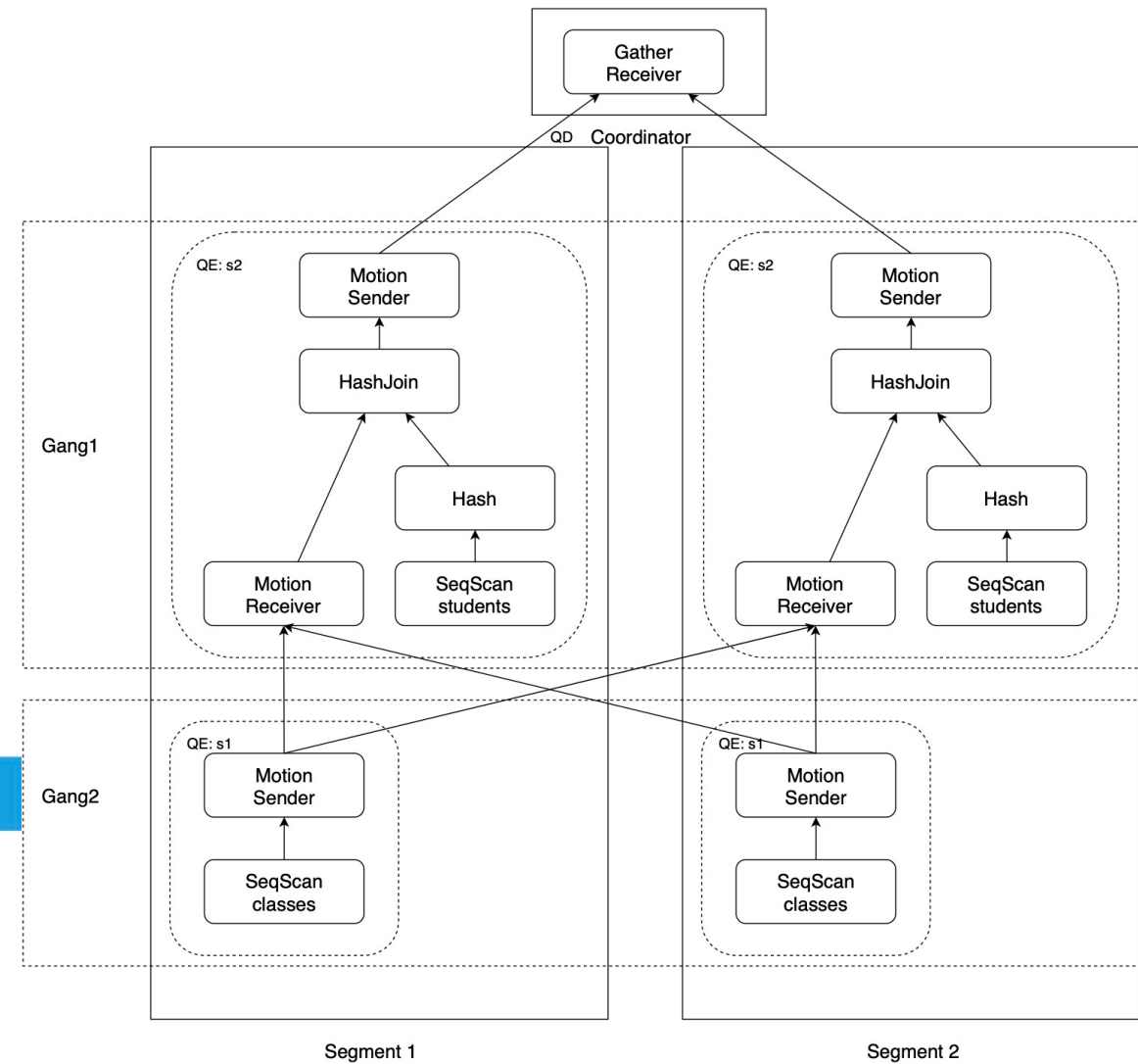
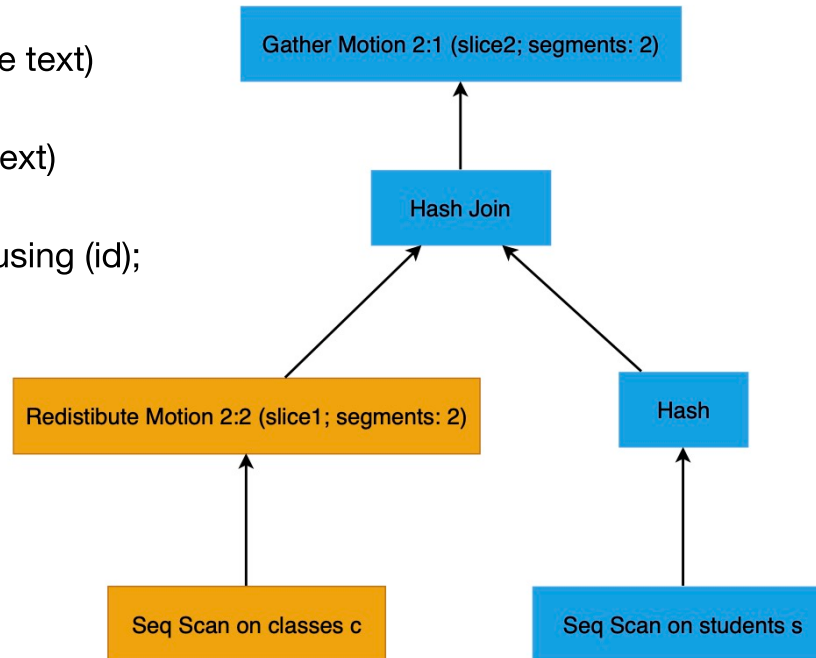
- 历时十年，独立开发
- Cascades 架构
- OLAP性能很棒
- <https://db.cs.cmu.edu/events/vaccination-2022-orca-a-modular-query-optimizer-architecture-for-vmware-greenplum-venky-raghavan/>

# Greenplum的一些概念

- MPP、分布式系统最重要的点是什么？
- 一个整体的分布式系统，和中间件的区别在哪？
- Motion
  - ❑ 跨节点的数据交换
  - ❑ Gather汇集 (n:1), Broadcast广播 (n:n), Redistribute重分布 (n:n)
- Slice
  - ❑ Motion把计划切片
  - ❑ 每一片叫Slice，每一个Slice的实体是一组存在于各个节点上的进程
- Locus
  - 数据的分布模式

# 分布式计划举例

create table student (id int, name text)  
distributed by (id);  
create table class (id int, name text)  
distributed randomly;  
select \* from student join class using (id);



# 分布式执行和事务

- 火山/流水线模型
- QD(query dispatcher)负责下发查询，QE(query executor)负责执行查询
- 查询的生命周期:
  - 1) 客户端连接coordinator, coordinator fork出QD
  - 2) QD 拿到纯文本的查询，解析、优化、生成一个树形结构的分布式计划
  - 3) QD 生成slice结构，生成每个slice的一系列进程结构（Gang）
  - 4) QD 连接segment节点，segment节点fork出QE，QE执行分布式计划
  - 5) QD 从QE归集结果，返回给客户端

# 分布式执行和事务

## ➤ MVCC

- ❑ Xmin, Xmax 是节点本地的

## ➤ 分布式快照

- ❑ QD生成, 下发给QE
- ❑ segment本地事务异步两阶段提交, 保持一致性

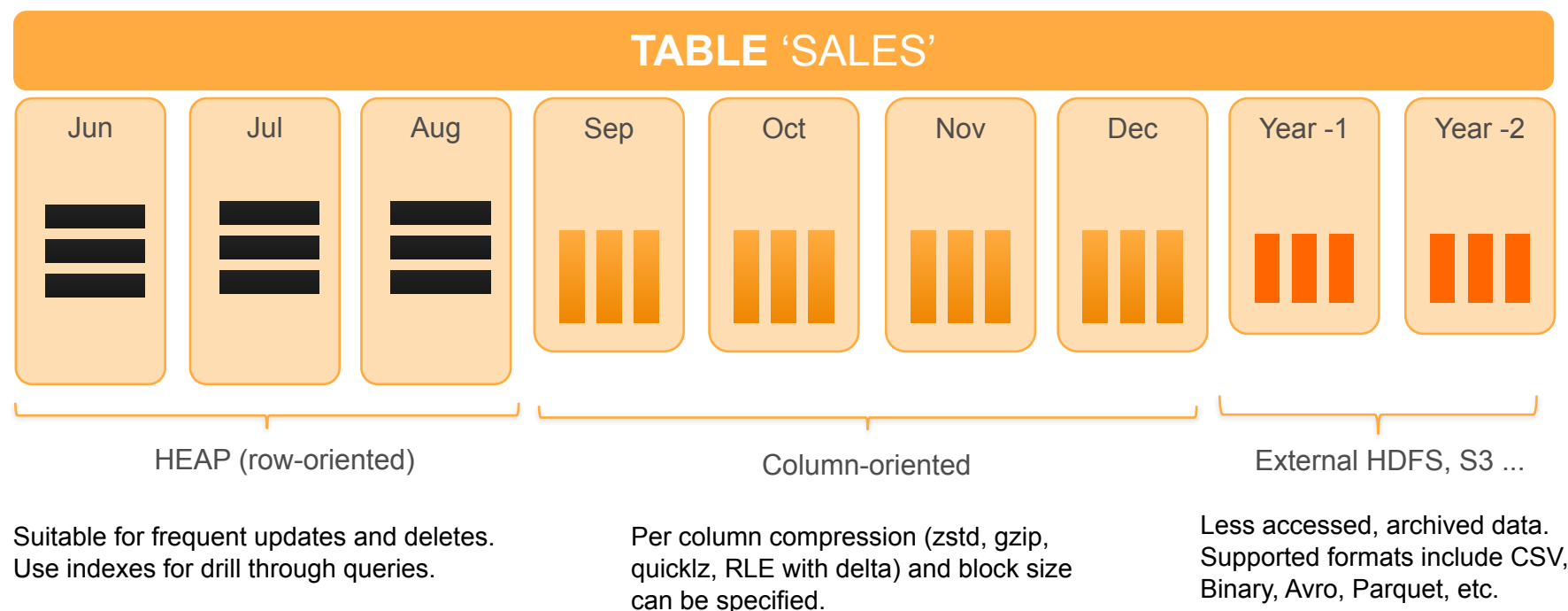
## ➤ HTAP 优化

- ❑ 全局死锁检测
- ❑ 只读事务、只涉及到某个节点的操作、vacuum

## ➤ SIGMOD 2021: *Greenplum: A Hybrid Database for Transactional and Analytical Workloads.*

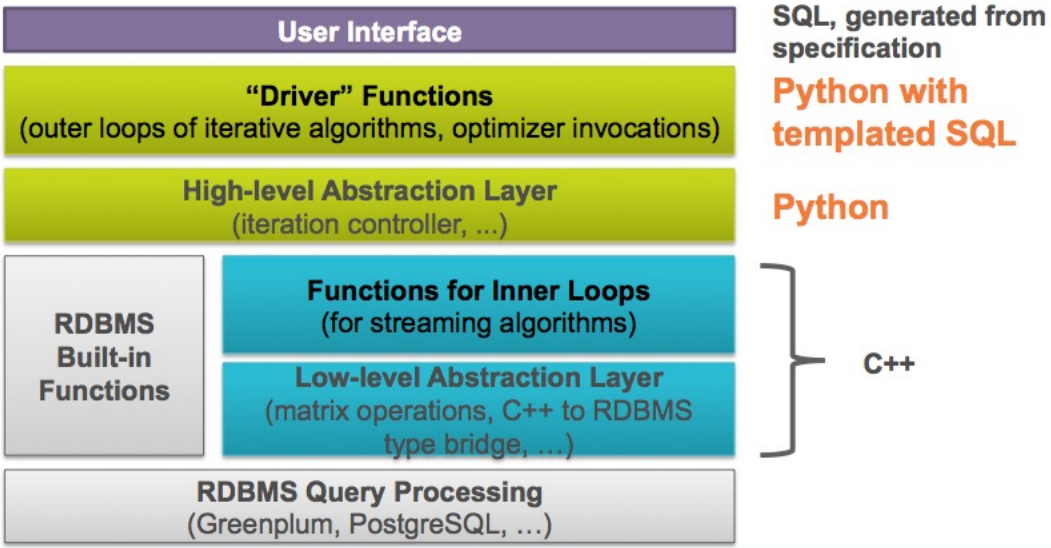
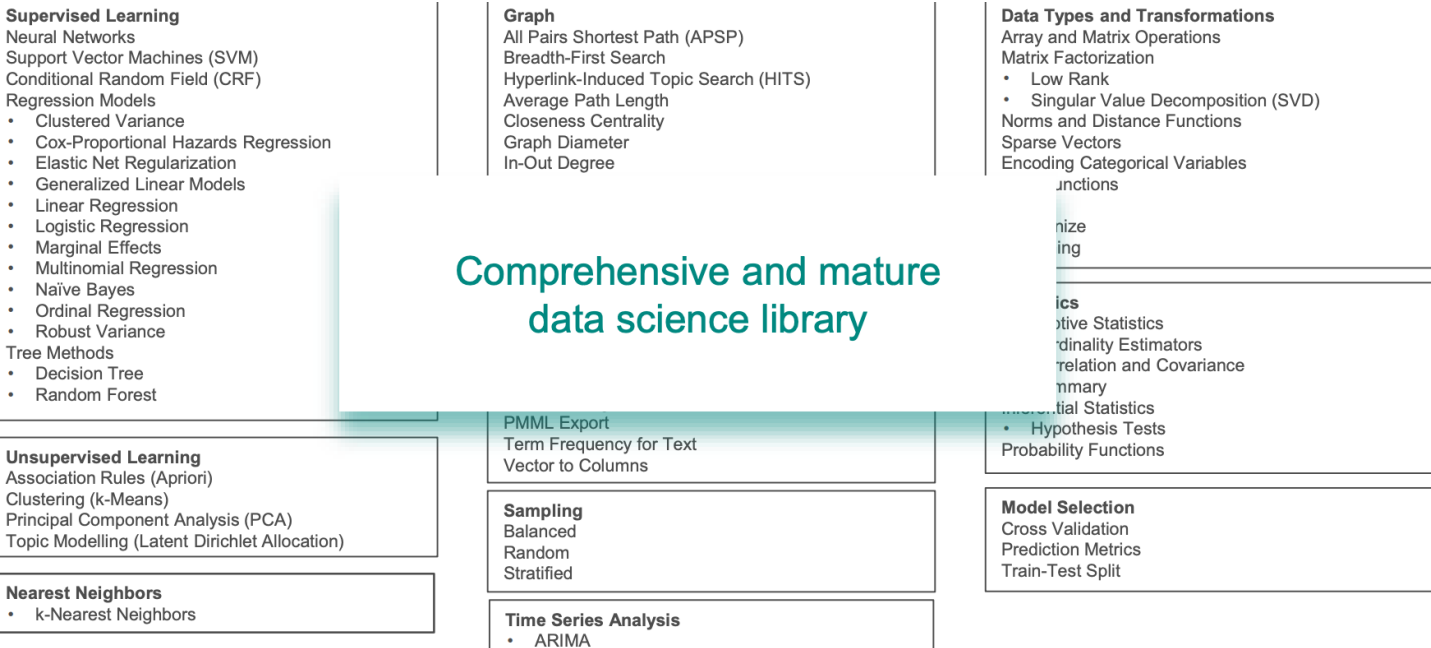
# Greenplum的存储

- Heap: 源自PG, 固定页面大小, 适合OLTP
- Append Optimized: 没有页面的概念, 变长, 行存、列存、压缩, 适合OLAP
- 外部表: HDFS, S3, 文件, 网络, 命令, 流式数据...



# Greenplum生态: Madlib

- 在数据库内做机器学习
- 非常多的算法库



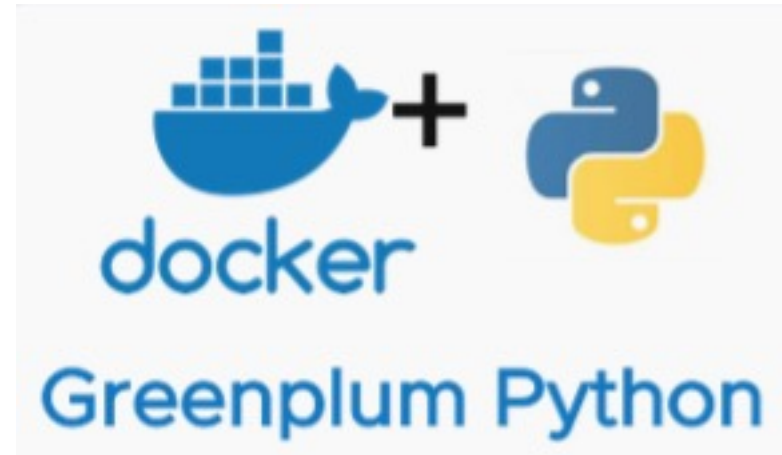
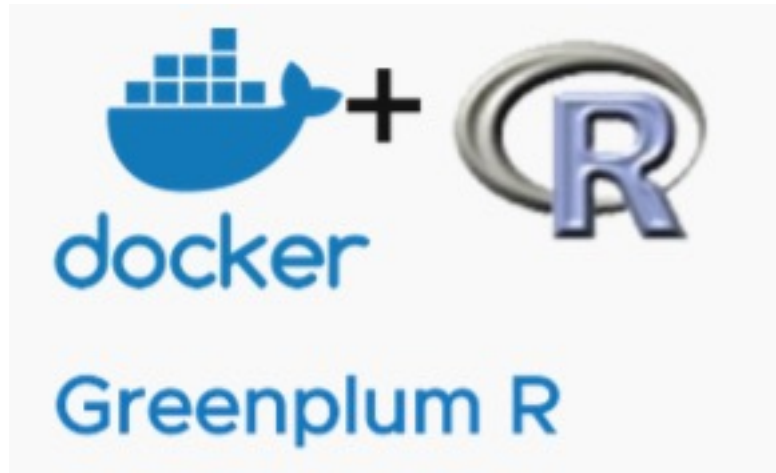


# GPTText

- MPP化的Apache Solr，用于全文检索和文本分析
- 举例：
  - ❑ 十个单词内包含Alan和Clinton: “Alan Clinton”~10

```
842613703 | WASHINGTON -- President Clinton's press secretary got a lesson on how sensitive a subject the Federal Reserve can be. Disclosi
ng that Fed Chairman Alan Greenspan met with Mr. Clinton Wednesday night, Dee Dee Myers told reporters that the Fed chief said passage of t
he president's deficit-reduction plan would help keep interest rates low. But that apparently sounded too much like Mr. Greenspan was endor
sing the specifics of the Clinton plan, which he hasn't. Ms. Myers later issued a written clarification saying she hadn't meant to imply th
at Mr. Greenspan was endorsing the president's plan. Rather, he told the president that "any credible deficit-reduction plan would help to
keep longterm interest rates low." A Fed spokesman wouldn't say if Mr. Greenspan had complained. "I can tell you that the clarification is
correct," he said. Separately, Treasury Secretary Lloyd Bentsen ducked a reporter's query about the possibility that the Fed may raise shor
t-term rates. "It's the Federal Reserve's province insofar as short-term interest rates" are concerned, he said.
(1 row)
```

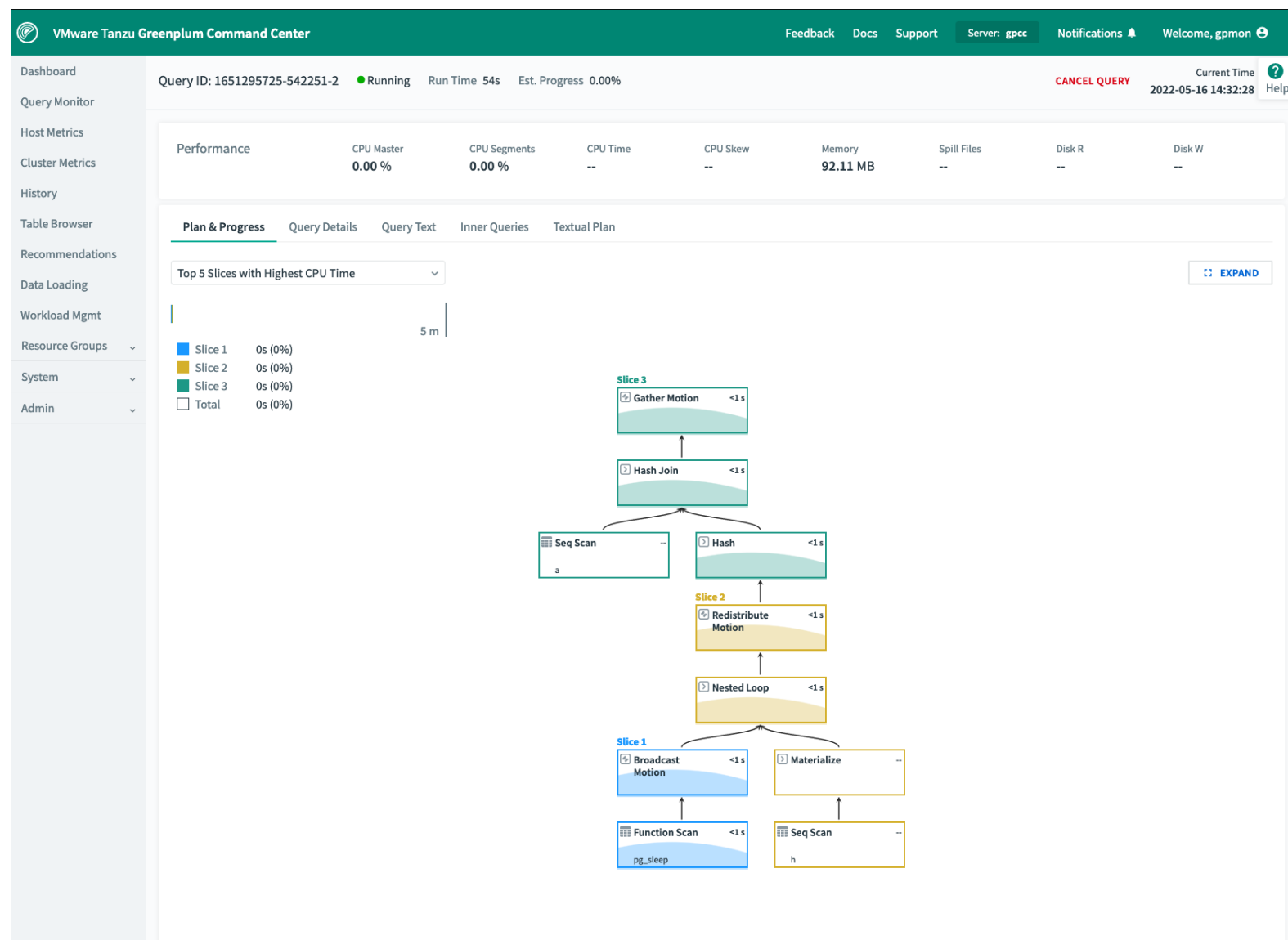
# PL Languages/Container



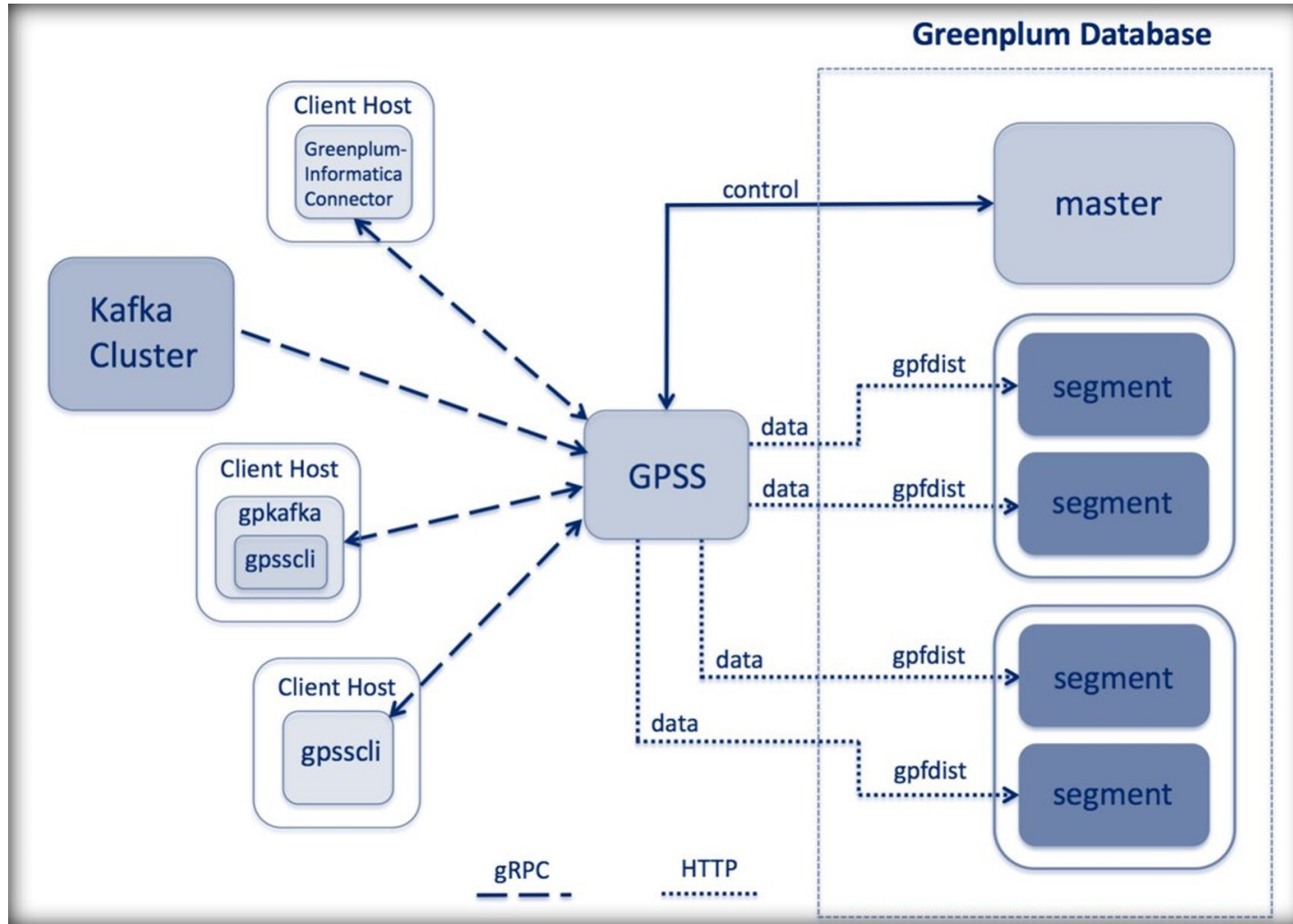
# GPCC

## Greenplum Command Center

- Web UI 监控和管理
- 实时性能监控
- 可视化计划
- 基于规则的任务管理
- 向客户推荐性能优化操作
- 报警和通知



# Greenplum Streaming Server



- ETL工具 (10+TB/hour)
- 并行导入流式数据
- Kafka和其它流式来源

# Greenplum 7的亮点: PostgreSQL v12 和新特性

- 6000+ 冲突
- 从9.4升级到12
- 80多万行改动
- Upsert, BRIN, JIT, ...

The screenshot shows a GitHub pull request interface. At the top, the title is "Merge with PostgreSQL v12 #10862". Below the title, it says "Merged" and "hlinnaka merged 6,529 commits into greenplum-db:master from hlinnaka:iteration\_REL\_12 on Sep 22, 2020".

On the left, there are statistics: "Conversation 11", "Commits 250", "Checks 0", and "Files changed 5,000+". On the right, there is a color-coded bar showing "+817,898" in green and "-741,497" in red.

The main content area shows a comment from "hlinnaka" dated "Sep 22, 2020". The comment text is:

Will be squashed into one gigantic merge commit before pushing. The purpose of this PR is to:

1. Be a heads up to everyone that this is about to land soon
2. Get review of the proposed commit message. Did I miss something?
3. Get one last run through the PR pipeline before pushing.

Merge commit message follows:

Merge with PostgreSQL version 12 (up to a point between beta2 and beta3).

This is the point where PostgreSQL REL\_12\_STABLE was branched off the master branch and v13 development started.

See PostgreSQL release notes for v10, v11 and v12 for information on the upstream changes included in this merge:

<https://www.postgresql.org/docs/release/10.0/>  
<https://www.postgresql.org/docs/release/11.0/>  
<https://www.postgresql.org/docs/release/12.0/>

The two most notable upstream features that had a big impact on GPDB code are:

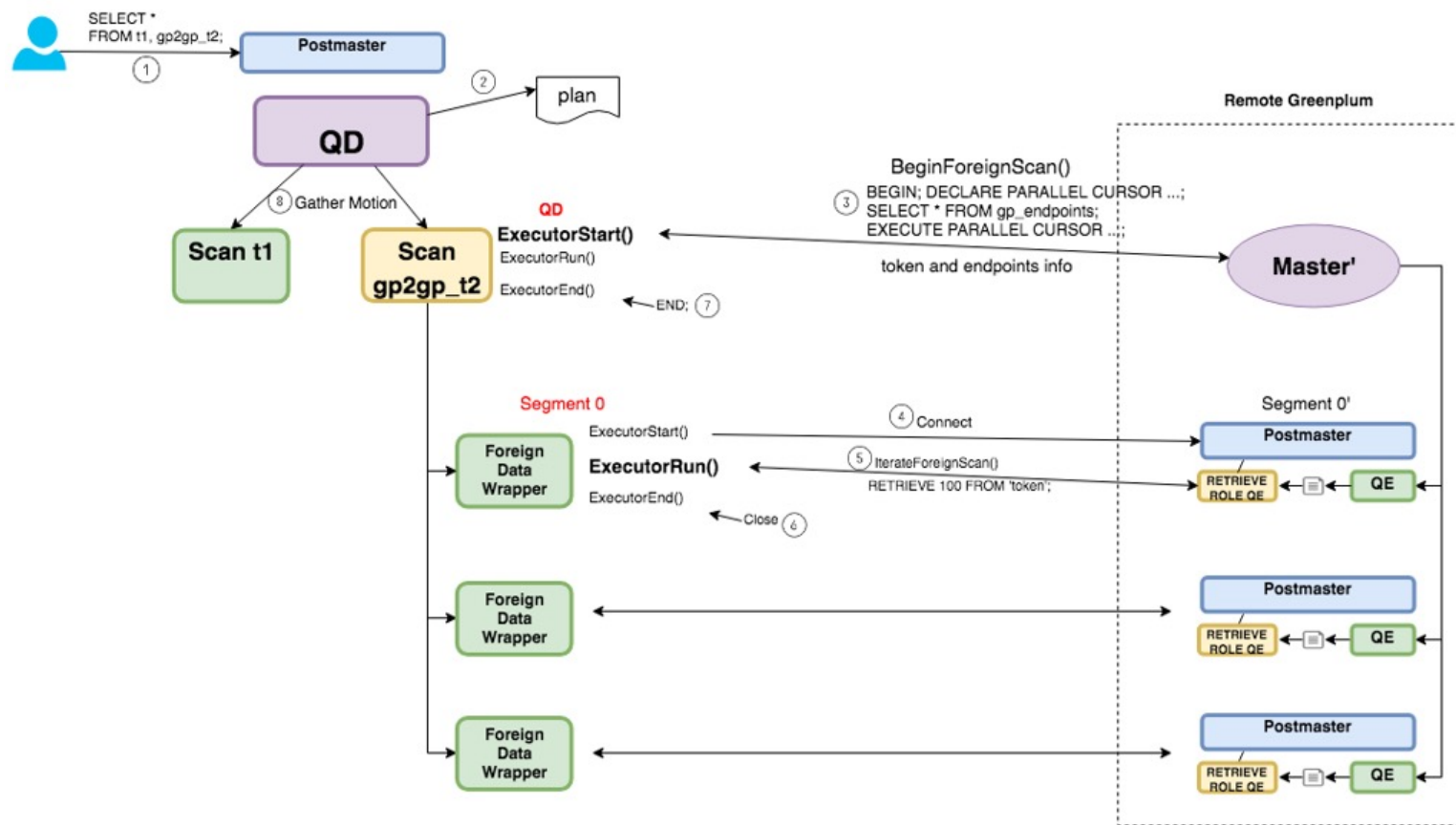
1. Partitioning
2. Table AM API.

The old GPDB partitioning support was completely ripped out and replaced with the upstream code, and new glue code was written to keep the old syntax working on top of the new implementation. Similarly, the AO and AOCS table code was refactored to be table access methods, working under the new Table AM API.

In addition to those big-ticket items, there are many, many smaller changes, detailed in the sections below. This isn't a comprehensive list of all upstream changes, I have only noted items that had a special impact on GPDB, because they work somehow differently from upstream, or they affected existing GPDB code or tests somehow. Also, there are many little things marked with GPDB\_12\_MERGE\_FIXME comments in the code that will need to be addressed after the merge.

On the right side of the pull request, there are sections for "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), "Development" (Successfully merging this pull request may close these issues. None yet), "Notifications" (Unsubscribe), and "22 participants" (a grid of 22 avatars).

## Greenplum 7的亮点: Greenplum to Greenplum



- 集群间节点直传
- 一套纯SQL的API
- 正在和其它分布式系统进行对接

谢谢！有问题吗？