



GREENPLUM
DATABASE[®]

一个用于分析, 机器学习和AI的开源大
规模并行数据平台

杨峻峰

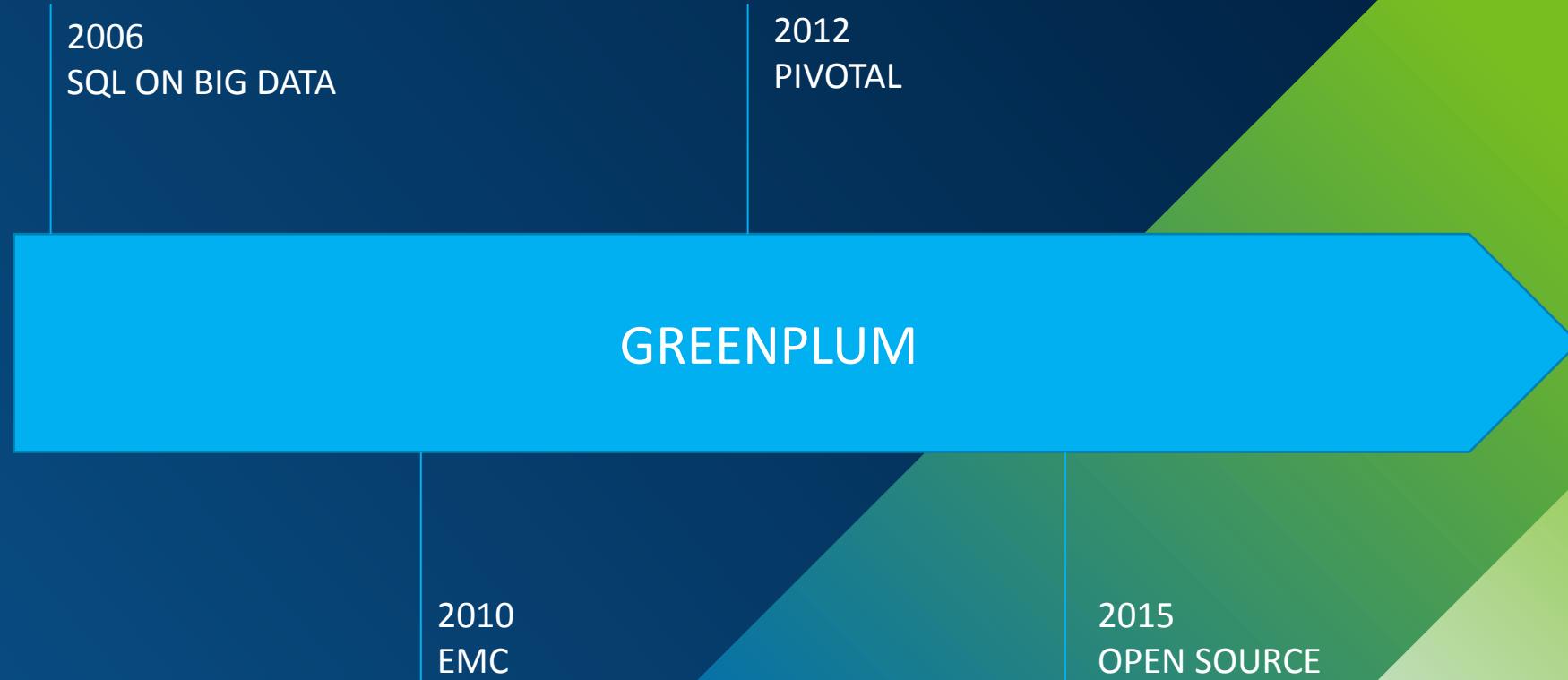
Greenplum内核研发工程师

2021年5月

Agenda

- Greenplum的历史
- Greenplum的基本架构
- 重要扩展组件
- 最近的工作
- 开源社区
- Q & A

Greenplum历史



Greenplum的基本架构

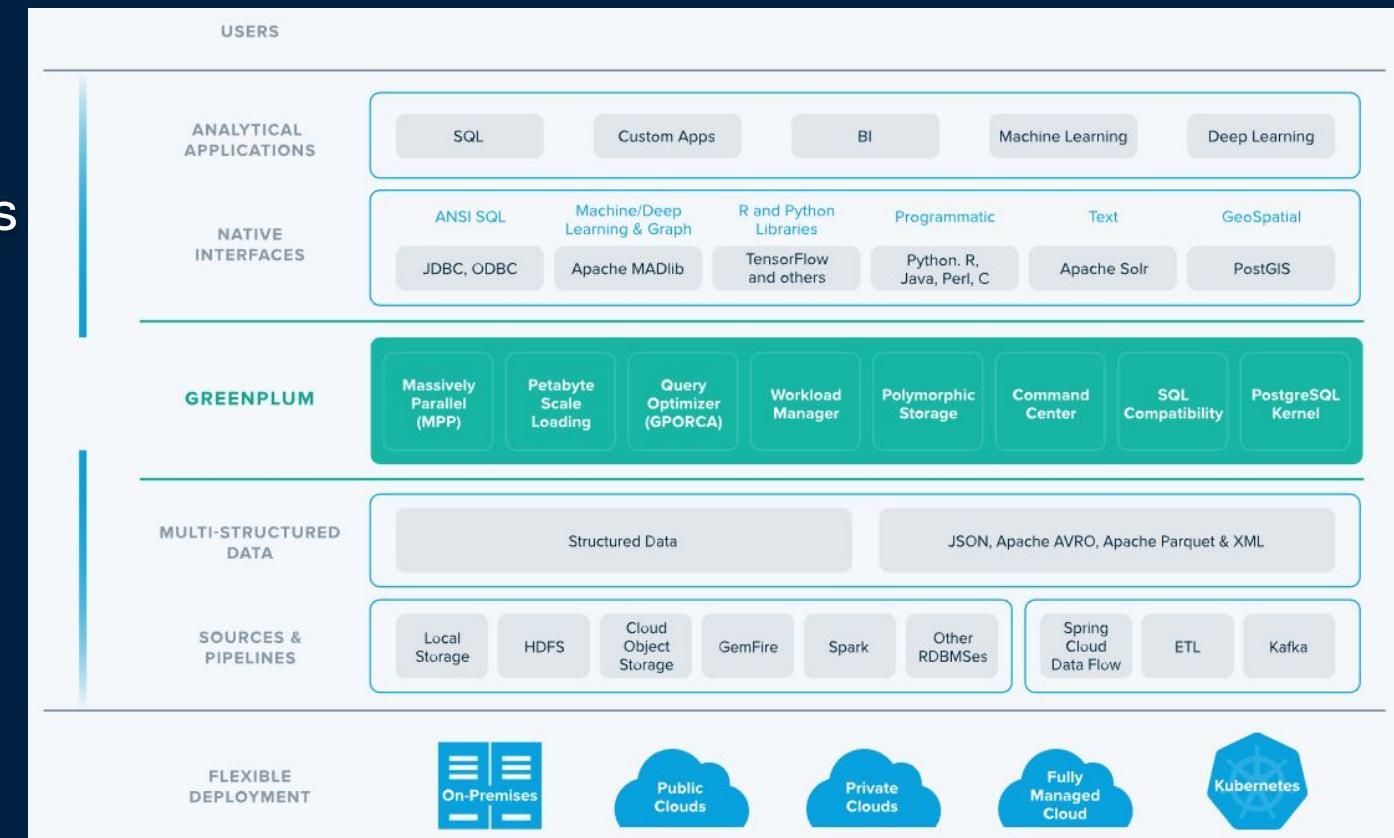
Greenplum是什么

PostgreSQL

- + Massively Parallel Process
- + Extra features
- + And addons

Greenplum 是一个开源项目

Greenplum 是一个开源项目



部署

- 一个Coordinator，数个Segment 实例（根据用户需求）
- 一台主机上可以部署多个Segment实例
 - 取决于主机的相关配置
- Mirroring 实现高可用
- Gb级别的交换机
- 一般部署在防火墙之后

Coordinator/Segment

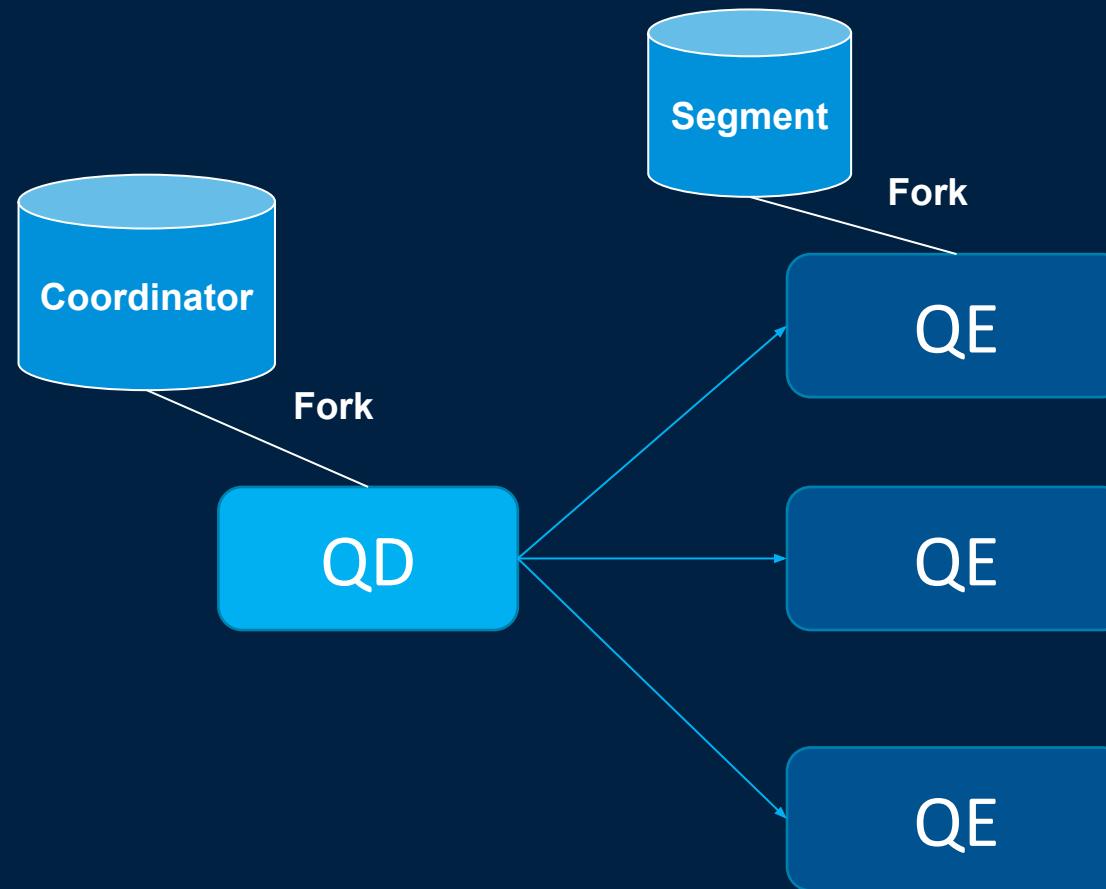
- Coordinator

- 管理元数据
- 接受客户端请求

- Segment

- 存储数据
- 执行计算任务

Query Dispatching



QD = Query Dispatcher

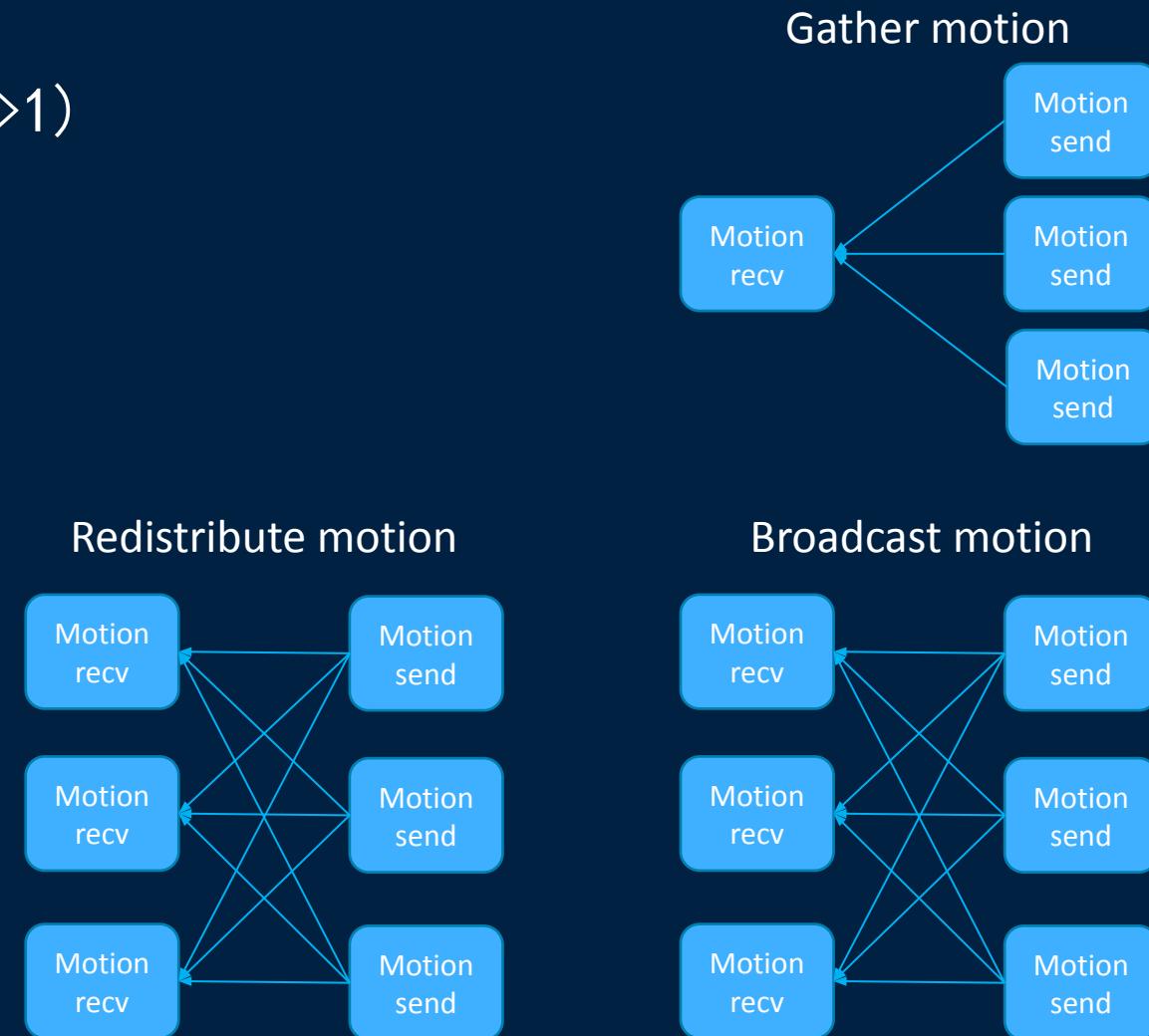
QE = Query Executor

QD和QE之前的通信协议

- 通信链接控制
 - libpq
- 数据传输 Interconnect
 - TCP
 - 自定义 UDP

Motion节点

- Gather motion ($N \rightarrow 1$)
- Broadcast motion ($N \rightarrow N$)
- Redistribute motion ($N \rightarrow N$)



分布式查询计划

Postgres 查询计划：

```
postgres=# explain select * from foo;  
QUERY PLAN
```

```
Seq Scan on foo
```

Greenplum 查询计划

```
postgres=# explain select * from foo;  
QUERY PLAN
```

```
Gather Motion 3:1 (slice1; segments: 3)  
-> Seq Scan on foo
```

- 查询计划从QD分发给每个QE节点
- 查询结果通过Motion从QE节点返回给QD

一个更复杂的查询

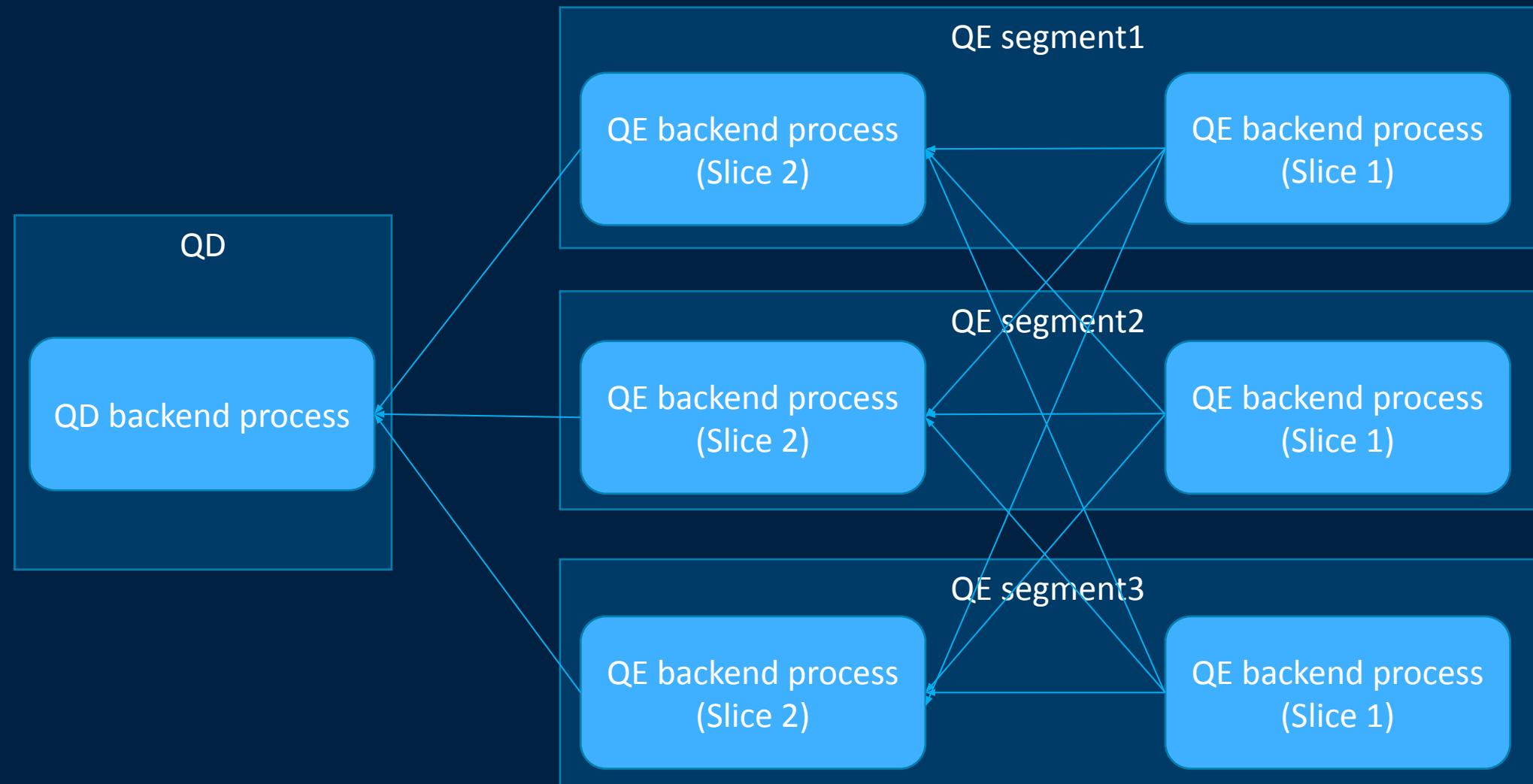
```
create table t1(a int, c int) distributed by (a);  
create table t2(a int, c int) distributed by (a);
```

```
demo=# explain select * from t1, t2 where t1.c = t2.c;  
          QUERY PLAN
```

Gather Motion 3:1 (slice2; segments: 3)

- > Hash Join
 - Hash Cond: t1.c = t2.c
 - > Seq Scan on t1
 - > Hash
 - > Broadcast Motion 3:3 (slice1; segments: 3)
 - > Seq Scan on t2

一个更复杂的查询



一个更复杂的查询

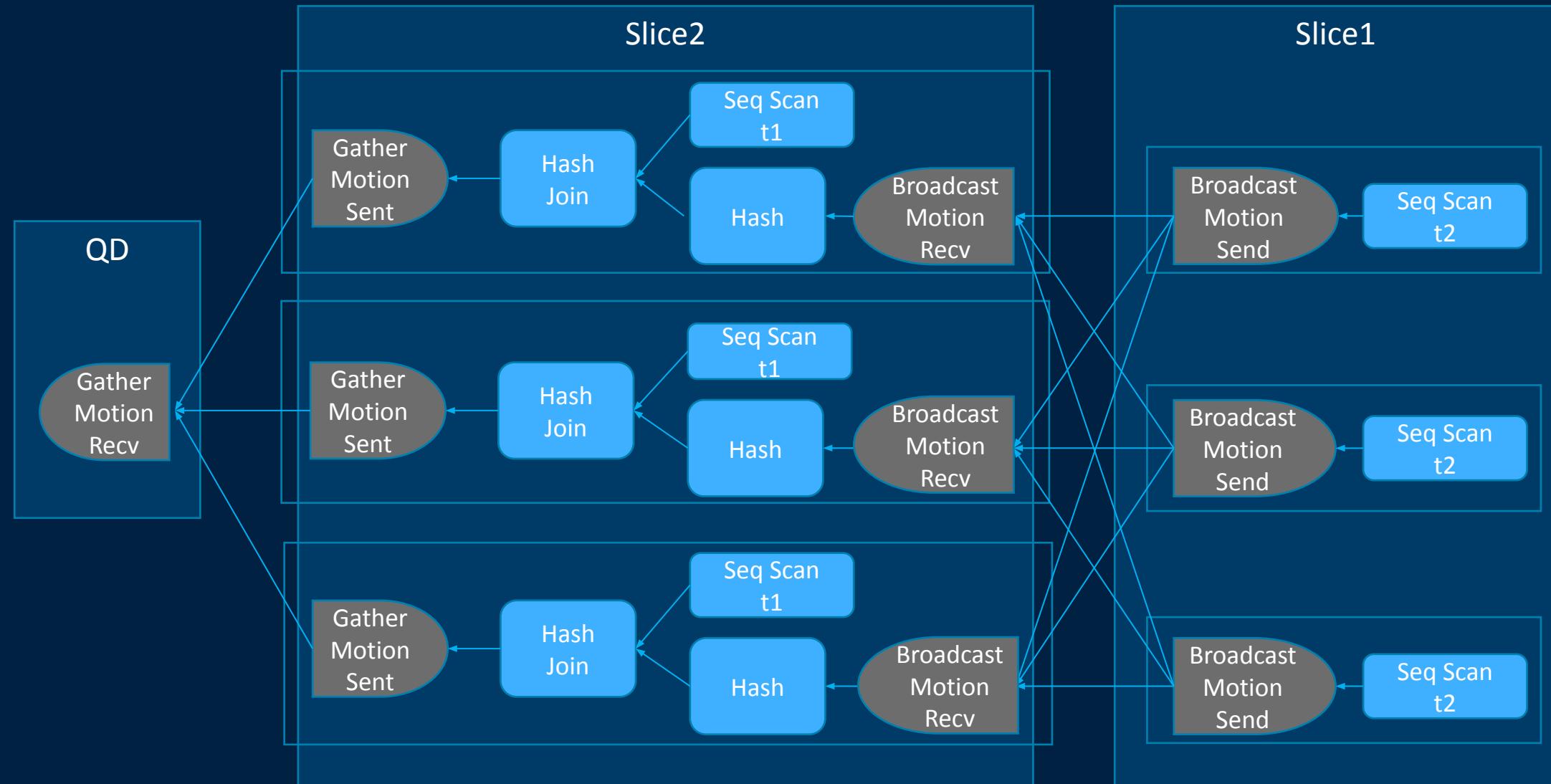
```
create table t1(a int, c int) distributed by (a);  
create table t2(a int, c int) distributed by (a);
```

```
demo=# explain select * from t1, t2 where t1.c = t2.c;  
          QUERY PLAN
```

Gather Motion 3:1 (slice2; segments: 3)

- > Hash Join
 - Hash Cond: $t1.c = t2.c$
 - > Seq Scan on t1
 - > Hash
 - > Broadcast Motion 3:3 (slice1; segments: 3)
 - > Seq Scan on t2

一个更复杂的查询



一个更复杂的查询

```
create table t1(a int, c int) distributed by (a);  
create table t2(a int, c int) distributed by (a);
```

```
demo=# explain select * from t1, t2 where t1.a = t2.c;
```

QUERY PLAN

Gather Motion 3:1 (slice2; segments: 3)

-> Hash Join

 Hash Cond: t2.c = t1.a

 -> Redistribute Motion 3:3 (slice1; segments: 3)

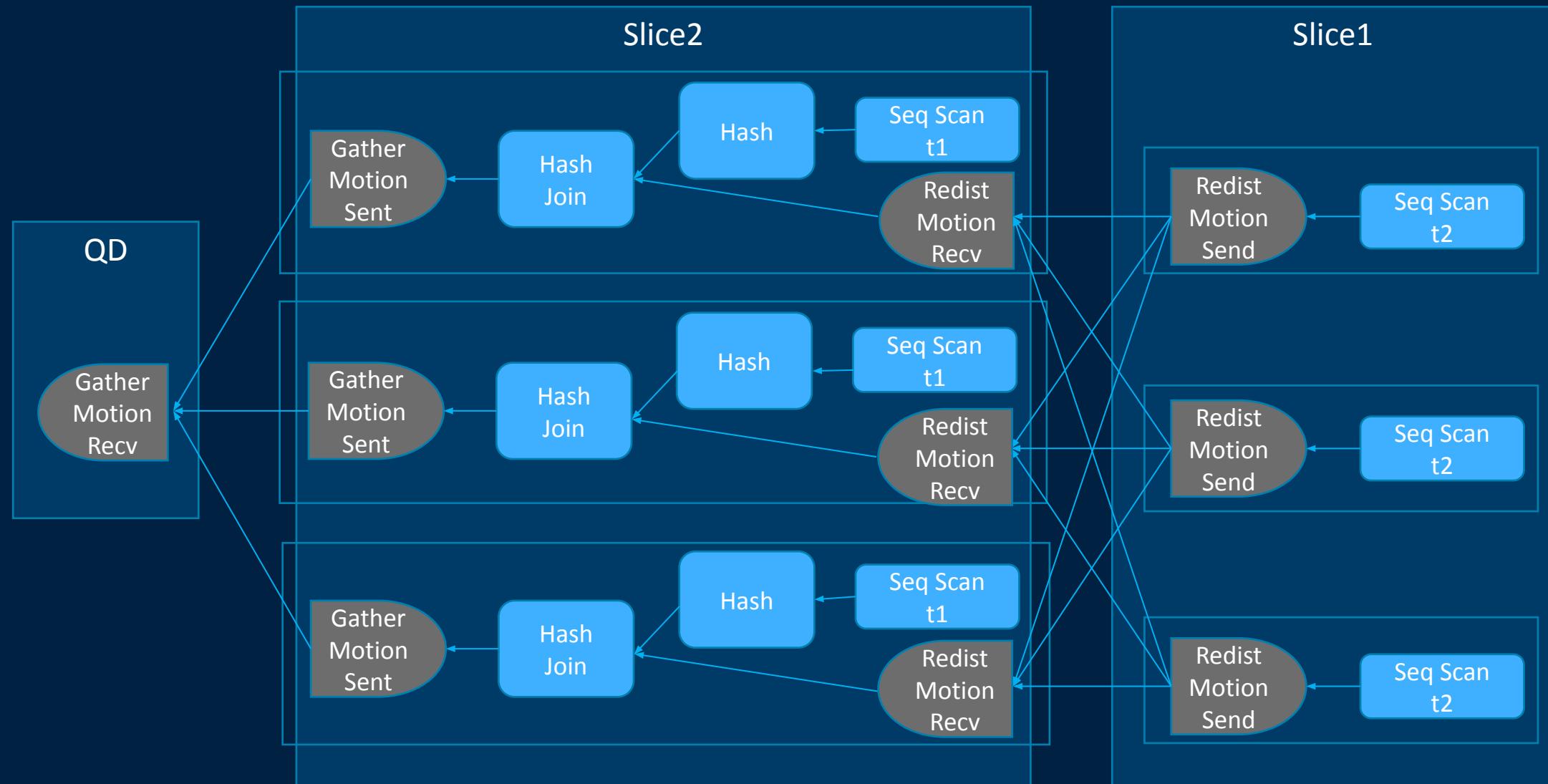
 Hash Key: t2.c

 -> Seq Scan on t2

 -> Hash

 -> Seq Scan on t1

一个更复杂的查询



一个更复杂的查询

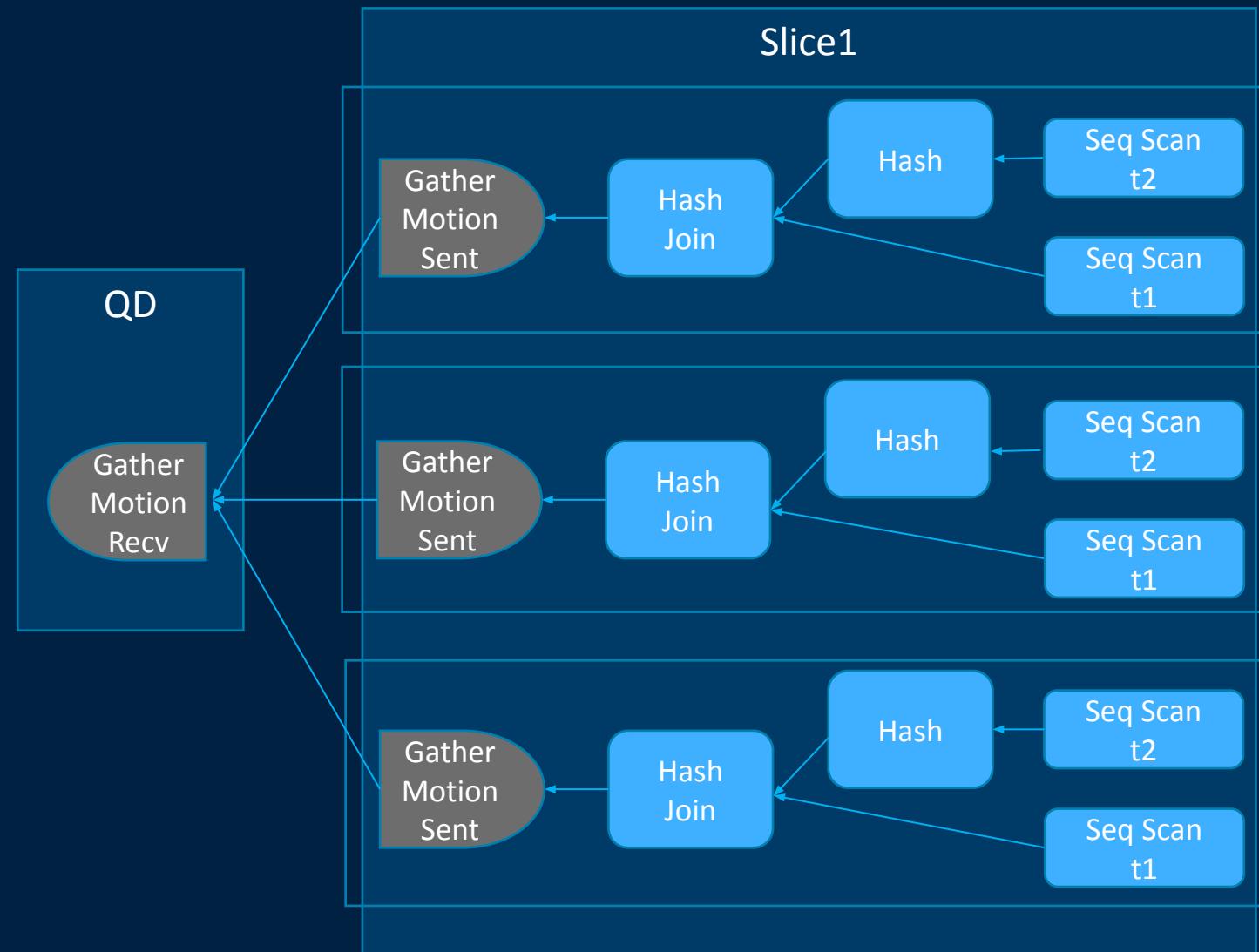
```
create table t1(a int, c int) distributed by (a);  
create table t2(a int, c int) distributed by (a);
```

```
demo=# explain select * from t1, t2 where t1.a = t2.a;  
QUERY PLAN
```

Gather Motion 3:1 (slice1; segments: 3)

- > Hash Join
 - Hash Cond: t1.a = t2.a
 - > Seq Scan on t1
 - > Hash
 - > Seq Scan on t2

一个更复杂的查询



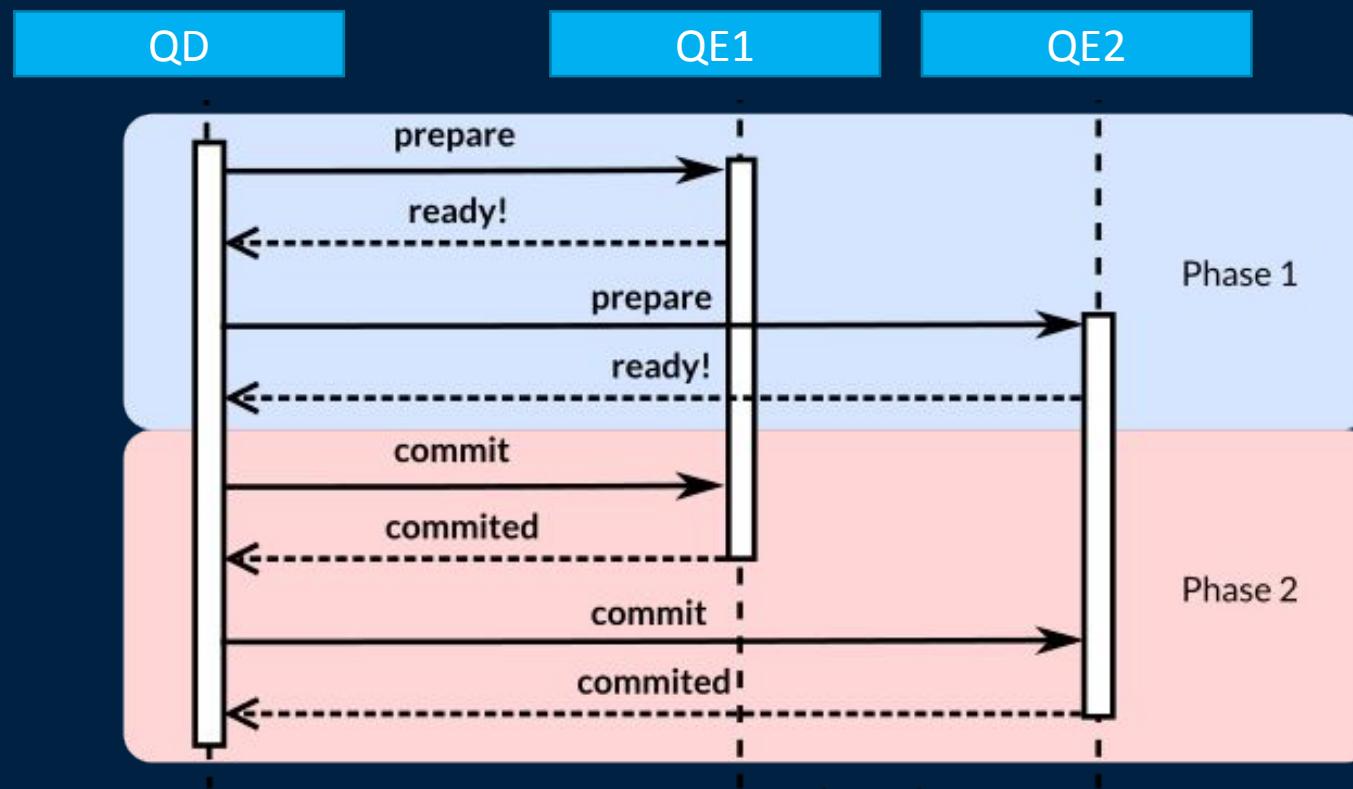
分布式查询计划

- 生成一个单机查询计划
- 跟踪每一步的distribution keys
- 在需要重分布的位置加入motion nodes
- 在聚合函数上实现多阶段聚合

分布式事务

- 两阶段提交
- QD作为两阶段提交的协调者
- 分布式快照保证一致性

两阶段提交

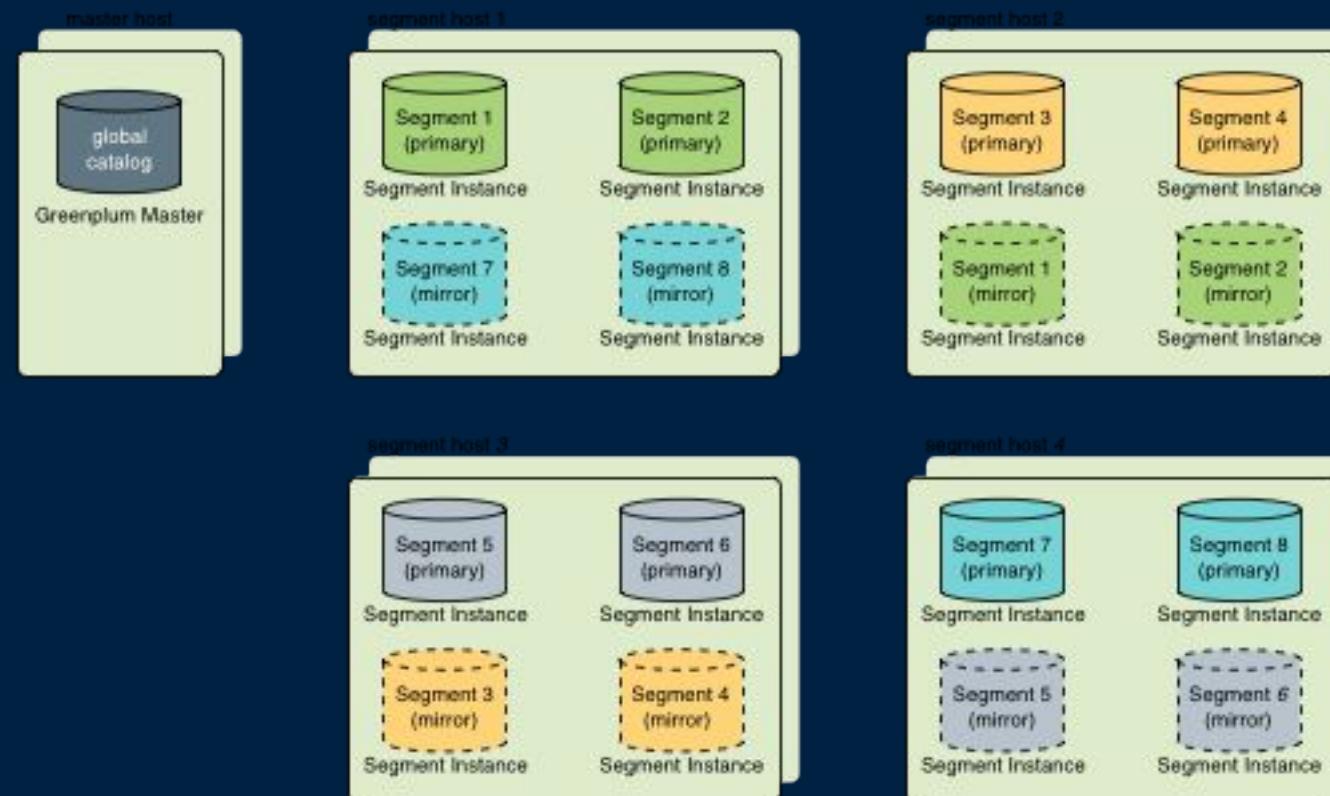


分布式快照

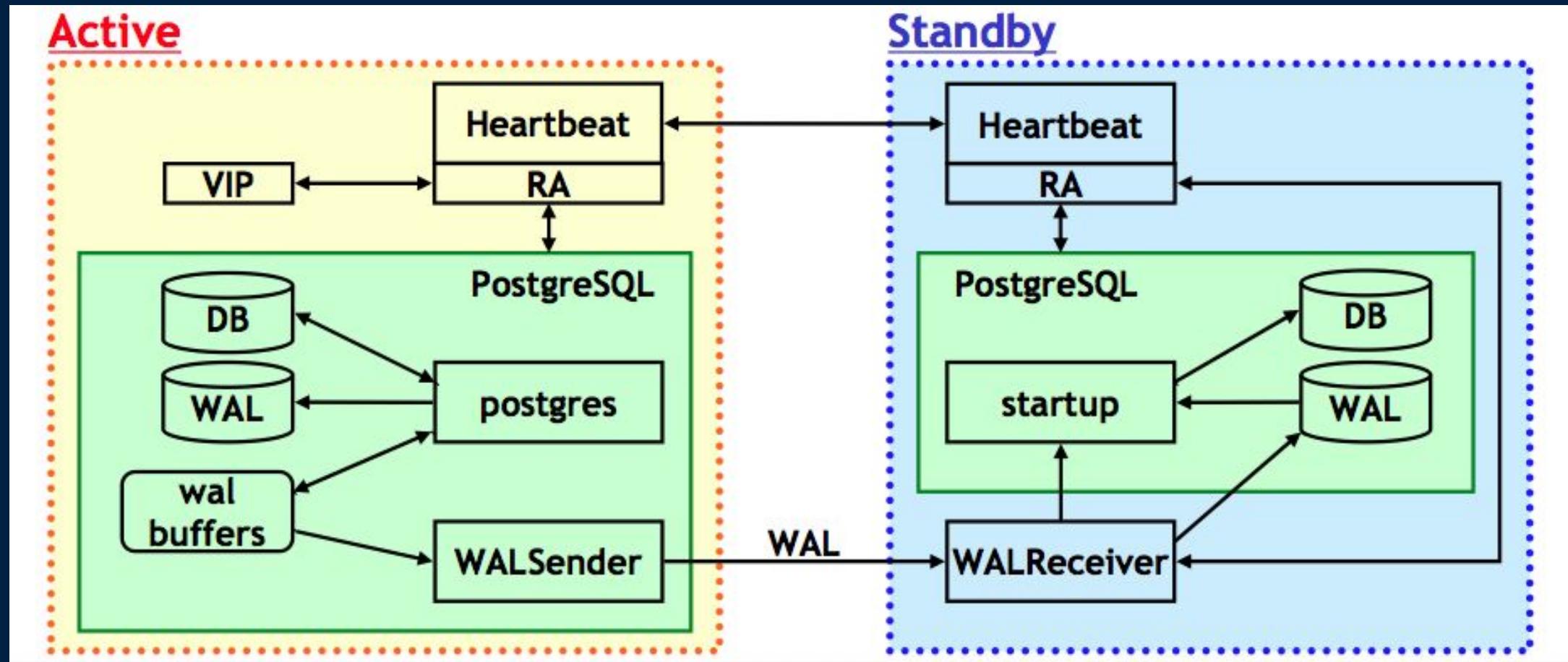
distribuTransactionTimestamp	The unique times for this start of DTM
distribSnapshotId	Snap shot id
xmin	XID < xmin are visible to me
xmax	XID >= xmax are invisible to me
count	Lenth of xid array
inProgressXidArray	Array of distributed transactions in progress

Mirroring

- 每个segment有一个mirror segment 作为备份
- Coordinator有一个standby 作为备份
- Wal replication



Wal replication



Partitioning

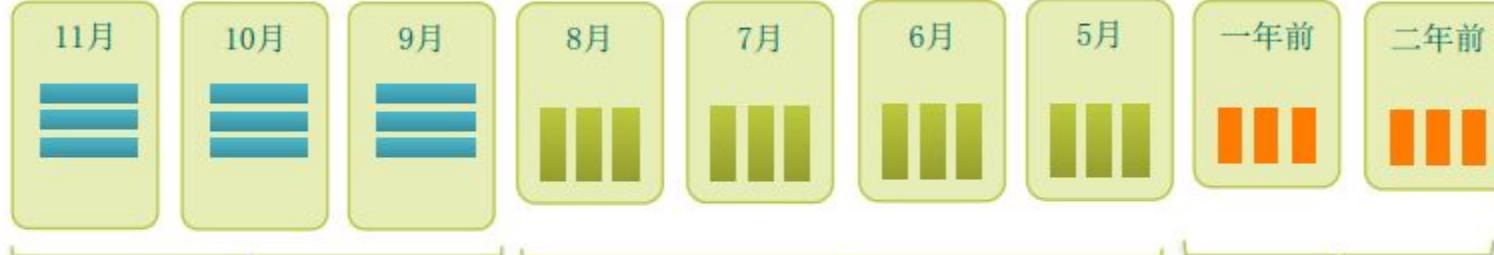
```
CREATE TABLE partition_test
(
    col1 int,
    col2 decimal,
    col3 text
)
distributed by (col1) partition by list(col2) (
    partition part1 VALUES(1,2,3,4,5,6,7,8,9,10),
    partition part2 VALUES(11,12,13,14,15,16,17,18,19,20),
    partition part3 VALUES(21,22,23,24,25,26,27,28,29,30),
    partition part4 VALUES(31,32,33,34,35,36,37,38,39,40),
    default partition def
);
```

自定义存储格式

多态存储

用户自定义数据存储格式

表‘SALES’



行存储

- 访问多列时速度快
- 支持高效更新和删除
- AO 主要为插入而优化

列存储

- 列存储更适合压缩
- 查询列子集时速度快
- 不同列可以使用不同压缩方式: gzip (1-9), quicklz, delta, RLE

外部表

- 历史数据和不常访问的数据存储在 HDFS 或者其他外部系统中
- 无缝查询所有数据
- Text, CSV, Binary, Avro, Parquet 格式

Greenplum的重要扩展组件

- GPORCA
- GPSS
- Apache MADlib
- GPText
- ...

GPORCA

- 一种新式的查询优化器
- 使用C++实现
- 可以同时适配greenplum和HAWQ
- <https://github.com/greenplum-db/gporca>
- 对于复杂查询，需要更长的计算时间，但是通常能够给出更好的计划

Greenplum Streaming Server

- GPSS是一种ETL(提取, 转换, 加载)工具。
- GPSS服务器的实例使用Greenplum数据库可读的外部表从一个或多个客户端获取流数据, 以将数据转换并将其插入到目标Greenplum表中。
- 数据源和数据格式特定于客户端。

MADlib

- 一个基于SQL的数据库内置的可扩展机的器学习库
- 强大的分析能力
- 将机器学习逻辑与数据库特定的实现细节分开
- 充分利用MPP架构处理海量数据
- Apache ASF上的顶级开源项目

MADlib 功能



The screenshot shows the MADlib feature page with sections for Supervised Learning, Unsupervised Learning, Graph, Data Types and Transformations, and other advanced topics like Text Analysis and Model Selection. It includes a red banner stating "复杂，成熟的数据科学学习库".

Supervised Learning

- Neural Networks
- Support Vector Machines (SVM)
- Regression Models
 - Clustered Variance
 - Cox-Proportional Hazards Regression
 - Elastic Net Regularization
 - Generalized Linear Models
 - Linear Regression
 - Logistic Regression
 - Marginal Effects
 - Multinomial Regression
 - Naïve Bayes
 - Ordinal Regression
 - Robust Variance
 - Tree Methods
 - Decision Tree
 - Random Forest
 - Conditional Random Field (CRF)

Unsupervised Learning

- Association Rules (Apriori)
- Clustering (k-Means)
- Topic Modelling (Latent Dirichlet Allocation)

Graph

- All Pairs Shortest Path (APSP)
- Breadth-First Search
- Average Path Length
- Closeness Centrality
- Graph Diameter
- In-Out Degree
- PageRank

Data Types and Transformations

- Array and Matrix Operations
- Matrix Factorization
 - Low Rank
 - Singular Value Decomposition (SVD)
- Norms and Distance Functions
- Sparse Vectors
- Principal Component Analysis (PCA)
- Categorical Variables

Text Frequency for Text Analysis

Time Series Analysis

- ARIMA

Model Selection

- Cross Validation
- Prediction Metrics
- Train-Test Split

复杂，成熟的数据科学学习库

Sept 2017

知乎 @IT大咖说

<https://github.com/apache/madlib>

GPText

- 海量文本数据处理
- 高速并行全文本检索
- 支持半结构化和结构化数据（社交媒体，文档等）
- 易用的SQL接口
- 复杂文本分析（实体识别，情感分析等）
- 内置自然语言处理工具

示例 MADLib + GPText

KMeans example:

```
SELECT * FROM madlib.kmeans_plusplus(  
    '<tfidf_table>',  
    '<tfidf_column>',  
    '<document_id>',  
    ... MADLib Kmeans parameters ...  
);
```

使用gptext.terms得到terms vector, 值为tfidf。

结合MADLib就可以去做更多的machine learning处理

最近的工作

- [SIGMOD paper] Greenplum: A Hybrid Database for Transactional and Analytical Workloads
- PG 12 merge
 - Partition logic (optimizer improvement)
 - BRIN index on AO
 - ...
- GP Auto Failover
- Auto Vacuum/Analyze

Greenplum开源社区

源码地址及代码结构

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

`gpMgmt /`

包含用于管理集群的特定于Greenplum的命令行工具。像gpinit, gpstart, gpstop之类的脚本都在这里。它们主要是用Python编写的。

`gpAux /`

包含特定于Greenplum的版本管理脚本和依赖。还包含一些子模块。

`gpcontrib /`

与PostgreSQL contrib /目录非常相似，该目录包含gpfdist, PXF和gpmapreduce之类的扩展，它们是Greenplum特有的。

`doc /`

在PostgreSQL中，用户手册位于此处。在Greenplum中，用户手册是单独维护的，此处仅用于构建手册页的参考页。

源码地址及代码结构

gpdb-doc /

包含DITA XML格式的Greenplum文档。有关如何构建和使用文档的信息，请参考 gpdb-doc / README.md。

ci /

包含GPDB持续集成系统的配置文件。

src /backend/ cdb /

Greenplum特定的后端模块。例如，segment之间的通信，将计划转变为可并行化计划，mirroring，分布式事务和快照管理等。cdb代表集群数据库-它是早期使用的工作名称。该名称不再使用，但保留cdb前缀。

src /backend/ gpopt /

包含所谓的翻译器库，用于将GPORCA优化器与Greenplum一起使用。转换器库是用C++代码编写的，并且包含用于在GPORCA使用的DXL格式与PostgreSQL内部表示之间转换计划和查询的结合代码。

源码地址及代码结构

`src/backend/gporca/`

包含GPORCA优化器代码和测试。这是用C++编写的。有关更多信息以及如何对GPORCA进行单元测试，请参见README.md。

`src/backend/fts/`

FTS是在Coordinator节点中运行的进程，并定期轮询以维护每个segment节点的状态。

`src/backend/*`

其他内核代码，可根据文件夹名大致理解其功能模块，例如optimizer主要包含查询优化器相关代码。executor主要包含执行器相关代码。

丰富的社区活动



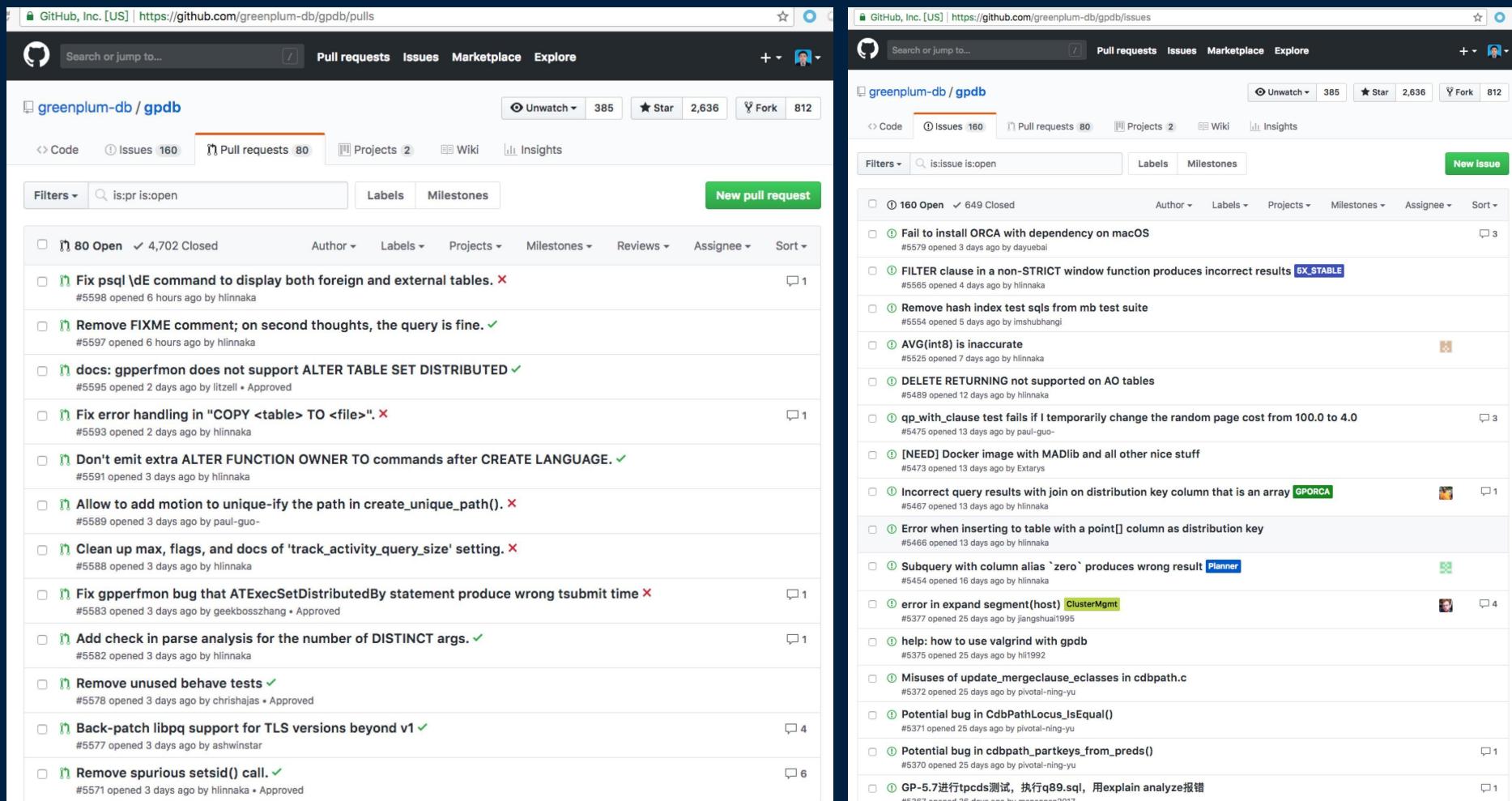
Greenplum内核课程

- <https://space.bilibili.com/489184136>



Greenplum的开发

- Github Account
- Fork repository
- Code change and add tests.
- Pull requests
- Code review, validate checks and CI



The image shows two screenshots of the Greenplum GitHub repository (`greenplum-db/gpdb`) illustrating the development process.

Pull Requests: The left screenshot displays the pull requests page with 80 open pull requests. The pull requests are listed with their titles, descriptions, and status (e.g., Open, Closed). Some pull requests have green checkmarks indicating they are approved or merged.

Issue #	Title	Status
#5598	Fix psql \dE command to display both foreign and external tables.	Closed
#5597	RemoveFIXME comment; on second thoughts, the query is fine.	Closed
#5595	docs: gpperfmon does not support ALTER TABLE SET DISTRIBUTED	Closed
#5593	Fix error handling in "COPY <table> TO <file>".	Closed
#5591	Don't emit extra ALTER FUNCTION OWNER TO commands after CREATE LANGUAGE.	Closed
#5589	Allow to add motion to unique-ify the path in create_unique_path().	Closed
#5588	Clean up max, flags, and docs of 'track_activity_query_size' setting.	Closed
#5583	Fix gpperfmon bug that ATExecSetDistributedBy statement produce wrong tsubmit time	Closed
#5582	Add check in parse analysis for the number of DISTINCT args.	Closed
#5578	Remove unused behave tests	Closed
#5577	Back-patch libpq support for TLS versions beyond v1	Closed
#5571	Remove spurious setsid() call.	Closed

Issues: The right screenshot displays the issues page with 160 open issues. The issues are listed with their titles, descriptions, and status (e.g., Open, Closed).

Issue #	Title	Status
#5579	Fail to install ORCA with dependency on macOS	Closed
#5565	FILTER clause in a non-STRRICT window function produces incorrect results	Closed
#5554	Remove hash index test sqls from mb test suite	Closed
#5525	AVG(int8) is inaccurate	Closed
#5489	DELETE RETURNING not supported on AO tables	Closed
#5475	qp_with_clause test fails if I temporarily change the random page cost from 100.0 to 4.0	Closed
#5473	[NEED] Docker image with MADlib and all other nice stuff	Closed
#5467	Incorrect query results with join on distribution key column that is an array	Closed
#5466	Error when inserting to table with a point[] column as distribution key	Closed
#5454	Subquery with column alias `zero` produces wrong result	Closed
#5377	error in expand segment(host)	Closed
#5370	help: how to use valgrind with gpdb	Closed
#5372	Misuses of update_mergeclause_eclasses in cdbpath.c	Closed
#5371	Potential bug in CdbPathLocus_IsEqual()	Closed
#5370	Potential bug in cdbpath_partkeys_from_preds()	Closed
#5367	GP-5.7进行tpcds测试，执行q89.sql，用explain analyze报错	Closed

贡献Greenplum社区

<https://github.com/greenplum-db/gpdb/pull/10515>

Enable autoanalyze in Greenplum #10515

Merged JunfengYang merged 14 commits into greenplum-db:master from hidva:auto-analyze on 31 Aug 2020

Conversation 108 · Commits 14 · Checks 0 · Files changed 22 · +1,068 -35

hidva commented on 24 Jul 2020 · edited by JunfengYang

Currently, we have many users inserting data into Greenplum in a streaming way. Optimizers often generate bad execution plans due to outdated statistics because there is no auto-analyze. `gp_autostats_mode` does not work well in such streaming insertion scenarios.

According to our research, the main reason why Greenplum does not enable auto analyze is that the global statistics information is not visible on the master, so the master does not know when to issue an analyze/vacuum. This means that if we can make the statistics, especially `PgStat_StatTabEntry::n_dead_tuples` and `PgStat_StatTabEntry::changes_since_analyze`, on the master accurate, and let the auto vacuum worker work in the `GP_ROLE_DISPATCH` mode, then the autovacuum worker on master will

- issue an analyze when `changes_since_analyze > analyze base threshold + analyze scale factor * number of tuples` is true.
- issue a vacuum when `TransactionIdPrecedes(classForm->relfrozenxid, xidForceLimit)` or `MultiXactIdPrecedes(classForm->relminmxid, multiForceLimit)` is true. It means that we will vacuum the table every `autovacuum_freeze_max_age-vacuum_freeze_min_age` transaction. Considering that the XID allocation on the master and the segment occurs at the same time(does it?), this should avoid the occurrence of xid wraparound warnings on the segment.
- issue a vacuum when `n_dead_tuples > vacuum base threshold + vacuum scale factor * number of tuples` is true. In fact, in our production environment, this kind of vacuum will only be triggered within a specified time period configured by our user according to their business characteristics.

Current PR only focuses on enabling ANALYZE in autovacuum on Master.

Reviewers: JunfengYang, hlinnaka, baobao0206, ashwinstar

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

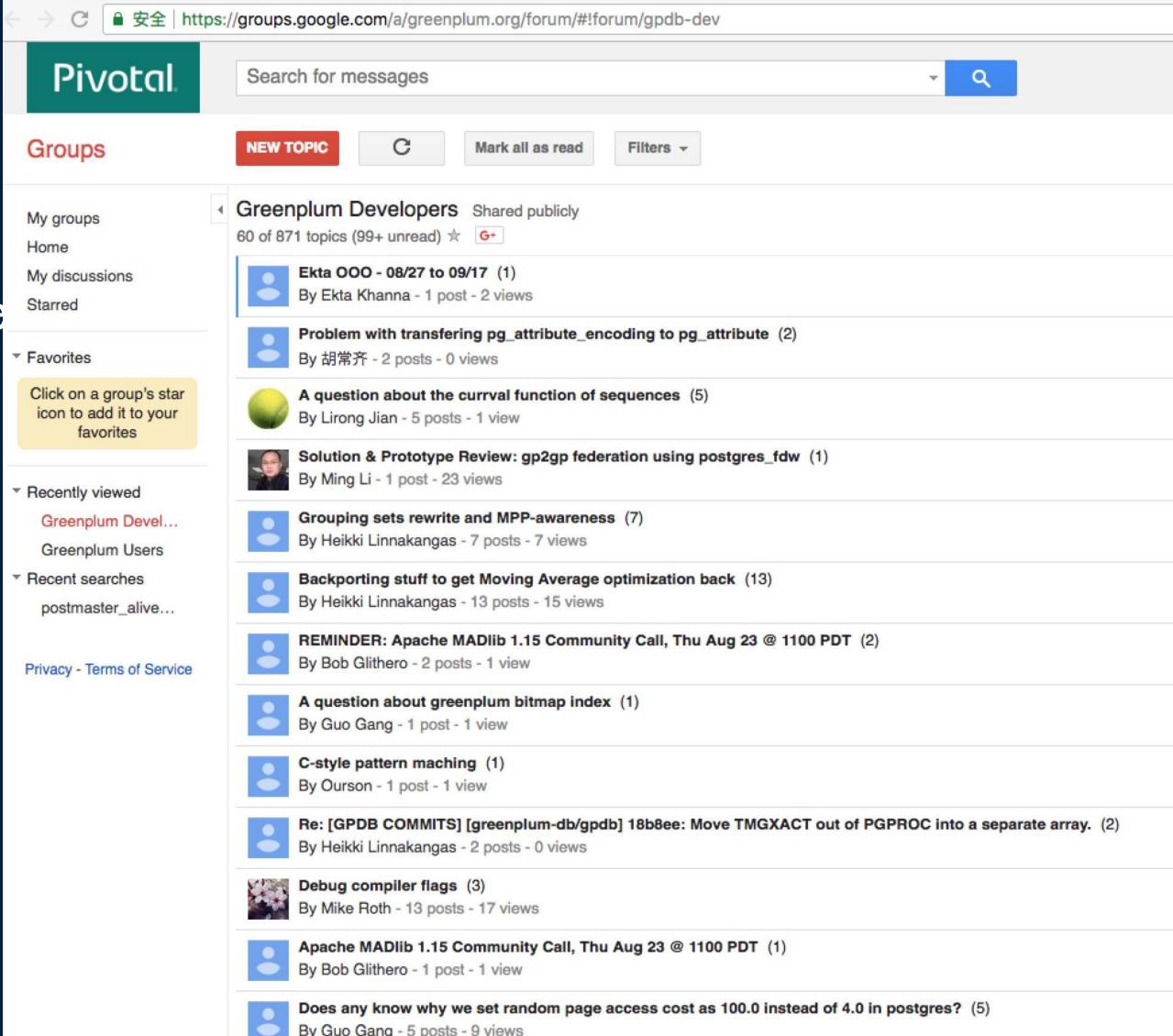
Greenplum参与社区

Github Issues

Public mailing lists on Google Groups:

- gpdb-dev
- gpdb-users

<https://groups.google.com/a/greenplum.org/forum/#!forum/gpdb-dev>



The screenshot shows the Google Groups interface for the "Greenplum Developers" mailing list. The list is shared publicly and has 60 of 871 topics. The topics are listed below:

- Ekta OOO - 08/27 to 09/17 (1) By Ekta Khanna - 1 post - 2 views
- Problem with transferring pg_attribute_encoding to pg_attribute (2) By 胡常齐 - 2 posts - 0 views
- A question about the curval function of sequences (5) By Lirong Jian - 5 posts - 1 view
- Solution & Prototype Review: gp2gp federation using postgres_fdw (1) By Ming Li - 1 post - 23 views
- Grouping sets rewrite and MPP-awareness (7) By Heikki Linnakangas - 7 posts - 7 views
- Backporting stuff to get Moving Average optimization back (13) By Heikki Linnakangas - 13 posts - 15 views
- REMINDER: Apache MADlib 1.15 Community Call, Thu Aug 23 @ 1100 PDT (2) By Bob Glithero - 2 posts - 1 view
- A question about greenplum bitmap index (1) By Guo Gang - 1 post - 1 view
- C-style pattern matching (1) By Ourson - 1 post - 1 view
- Re: [GPDB COMMITS] [greenplum-db/gpdb] 18b8ee: Move TMGXACT out of PGPROC into a separate array. (2) By Heikki Linnakangas - 2 posts - 0 views
- Debug compiler flags (3) By Mike Roth - 13 posts - 17 views
- Apache MADlib 1.15 Community Call, Thu Aug 23 @ 1100 PDT (1) By Bob Glithero - 1 post - 1 view
- Does any know why we set random page access cost as 100.0 instead of 4.0 in postgres? (5) By Guo Gang - 5 posts - 9 views

greenplum.org

- News, events
- Links to the github, project, mailing list

Q & A

Thank You