

# PART 04 ▶

## 数据仓库实例讲解 ——Apache HAWQ 系统架构和基本原理

# 目录

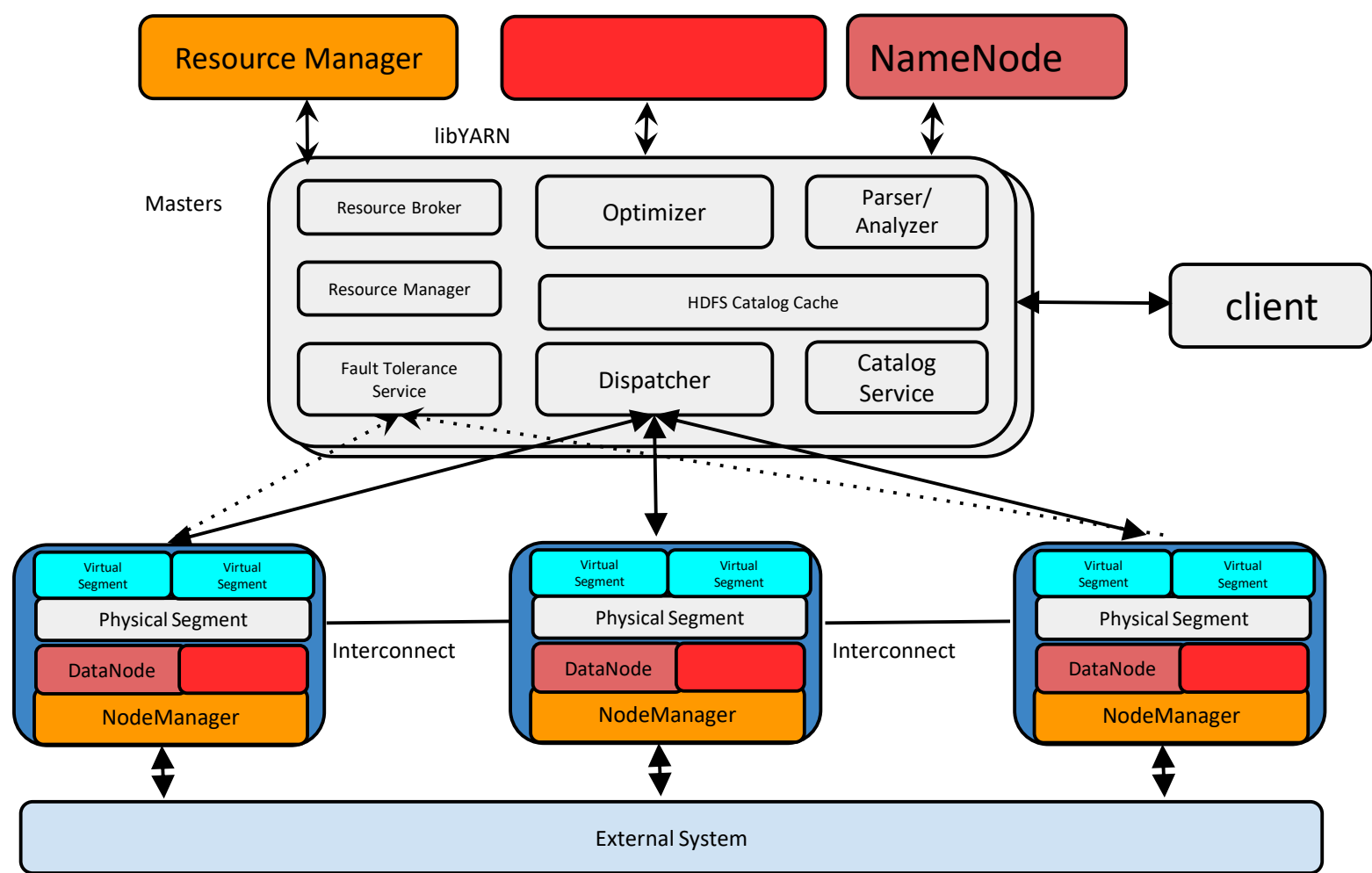
## CONTENT

- 01** | Apache HAWQ基本组件
- 02** | Apache HAWQ部署架构
- 03** | Apache HAWQ进程架构
- 04** | Apache HAWQ查询执行
- 05** | Apache HAWQ查询计划
- 06** | 外部接口
- 07** | 调试

# 01

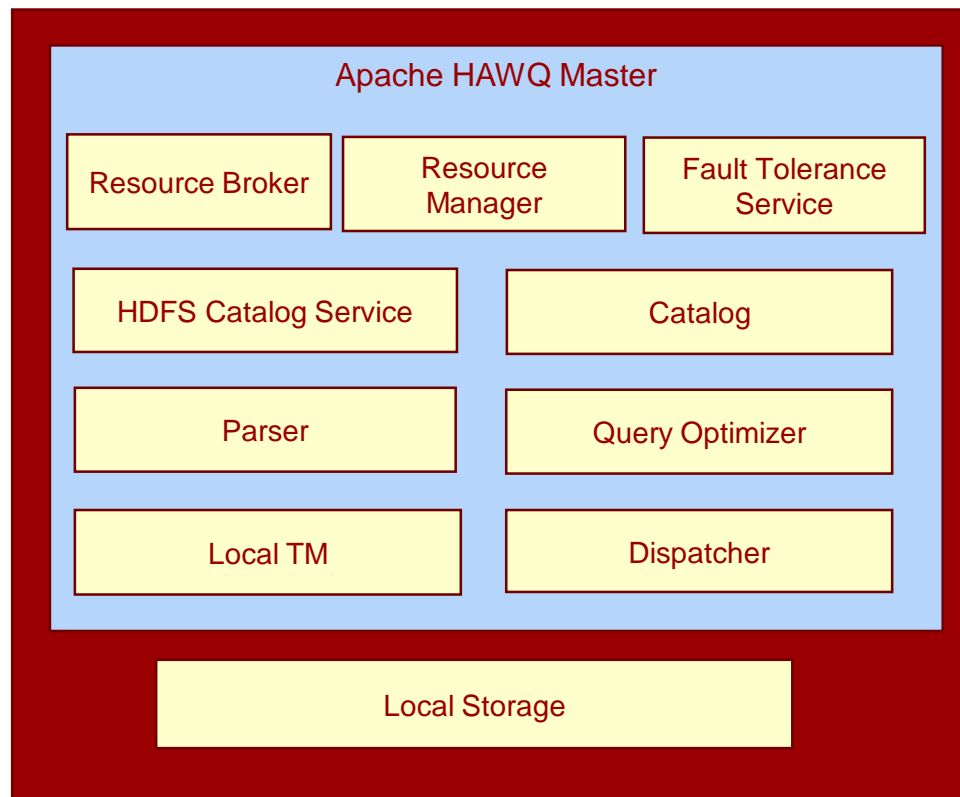
## Apache HAWQ 基本组件

# Apache HAWQ基本组件



# Apache HAWQ Master

- 元数据信息仅存储在master的本地存储中
- 认证客户和接受客户的要求
- 将SQL转换为解析树数据结构
- SQL优化器决定最有效的并行查询执行
- 将查询计划分发segment
- 将结果返回给客户端
- 执行事务管理



# Apache HAWQ Physical Segment & Virtual Segment

- 一个 Apache HAWQ segment host 是一个计算节点
- 每个segment host可以有多个虚拟segment,一般为 (2, 4, 6, 8) , 缺省为8
- 一个虚拟segment是一个并行查询的最基本计算单元 (资源的container, 并行度的弹性)
- 多个部分协同工作, 形成一个单一的并行查询处理系统。
- Apache HAWQ segments向HDFS传递请求所需的数据节点完成数据查询。
- Segment和Master的区别:
  - 无状态
  - 不存储数据库表元数据
  - 不存储本地文件系统中的数据

# Apache HAWQ基本组件

- HDFS (可选)
  - Namenode: 管理文件系统的元数据, 在一个集群中配置两个不同的机器来提供HA的服务
  - Datanode: 管理该节点上的数据存储
- Zookeeper (可选)
  - 负责为分布式应用提供一致性服务。
  - Zookeeper 会实时获取HDFS 两个Namenode 的心跳信息, 从而在必要时触发namenode 的切换。

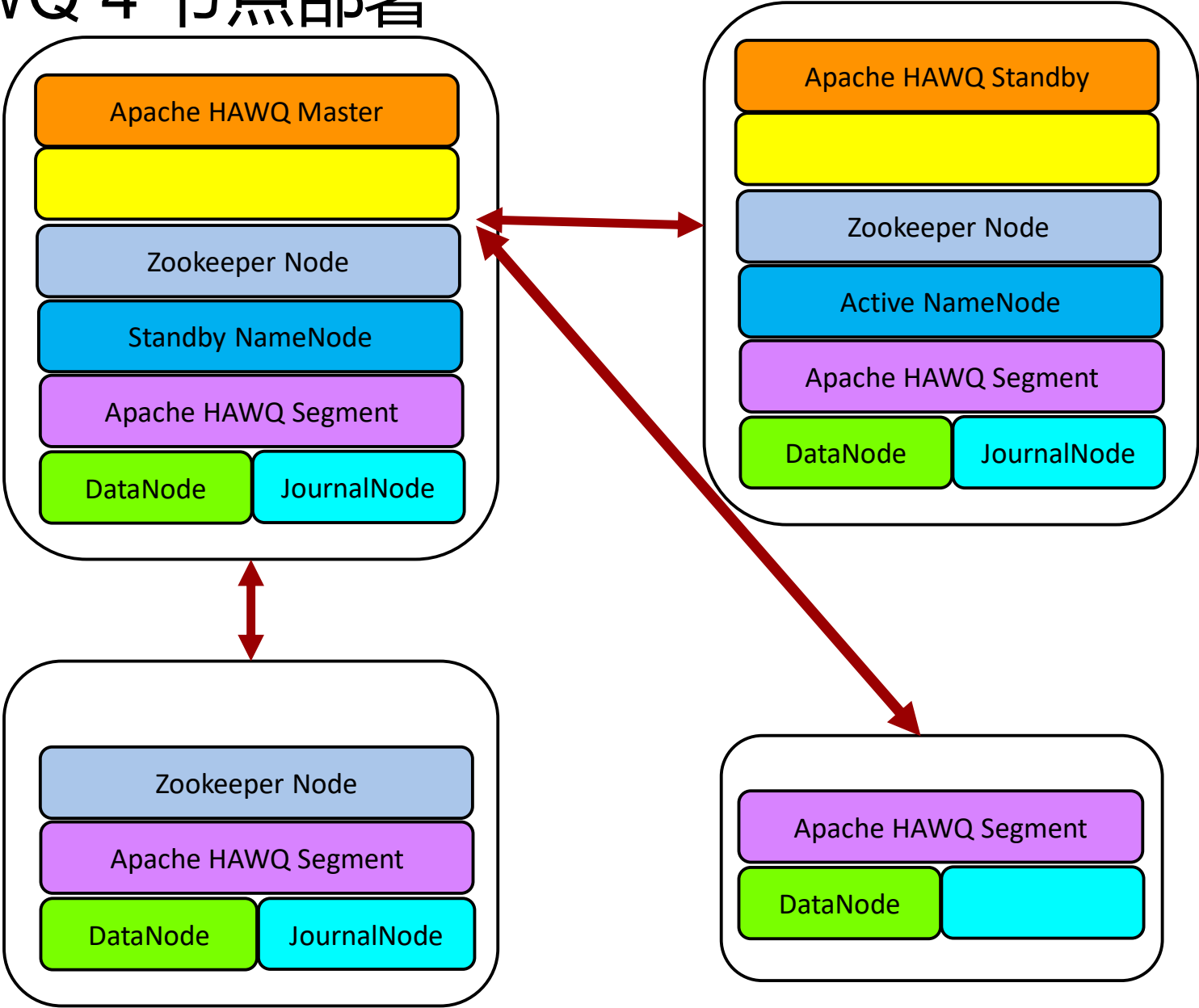
# Apache HAWQ基本组件

- Yarn (可选)
- Apache HAWQ 提供了Standalone 和Yarn 两种模式来进行资源管理
- Standalone: 缺省模式, 独占系统资源。
- Yarn: Apache HAWQ 作为Yarn 的一个应用程序, 通过libyarn向Yarn 来申请资源。
- Resource Manager
  - 全局的资源管理, 负责整个系统的资源管理和分配。
- Node Manager
  - 负责本地节点资源管理

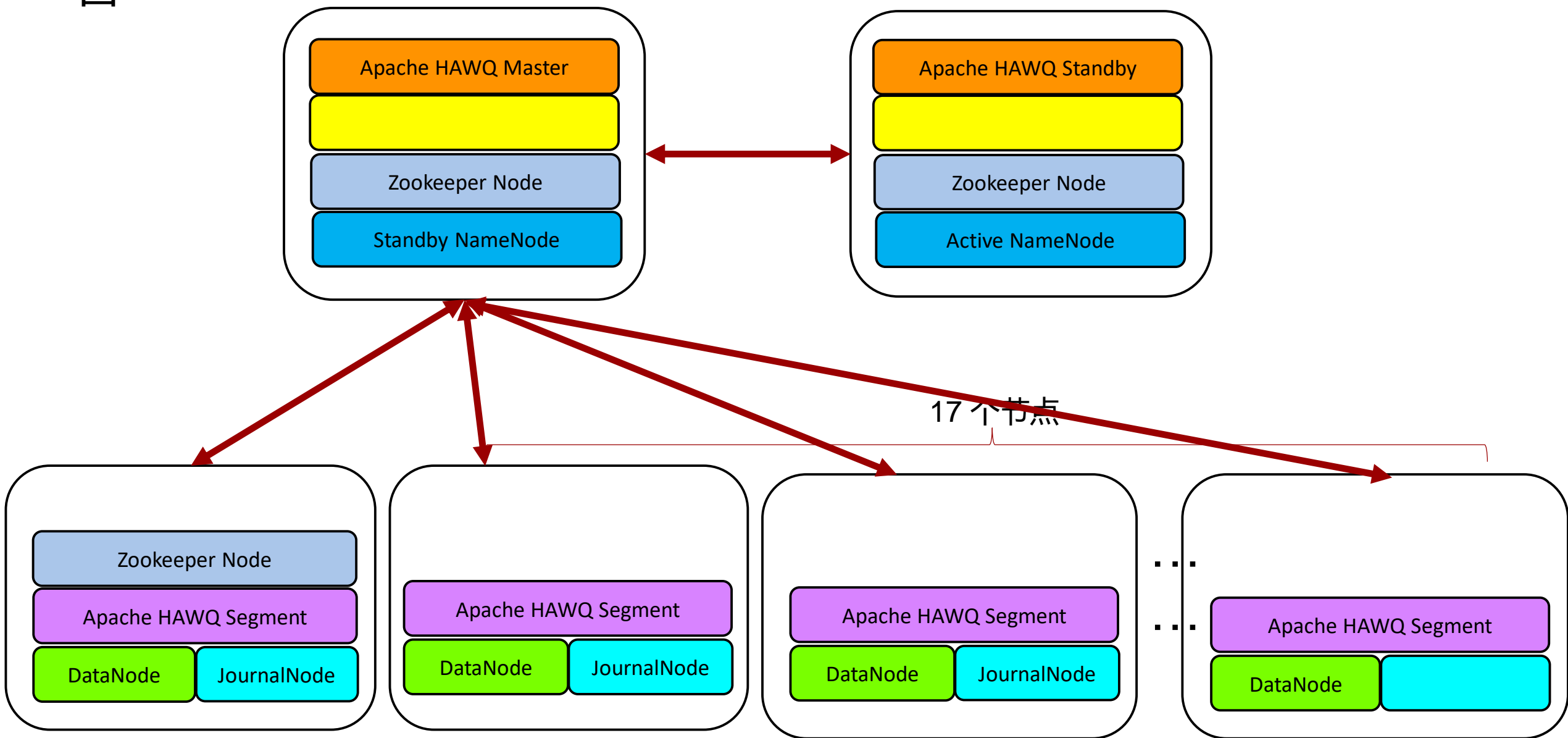


# 02 Apache HAWQ 部署架构

# Apache HAWQ 4 节点部署

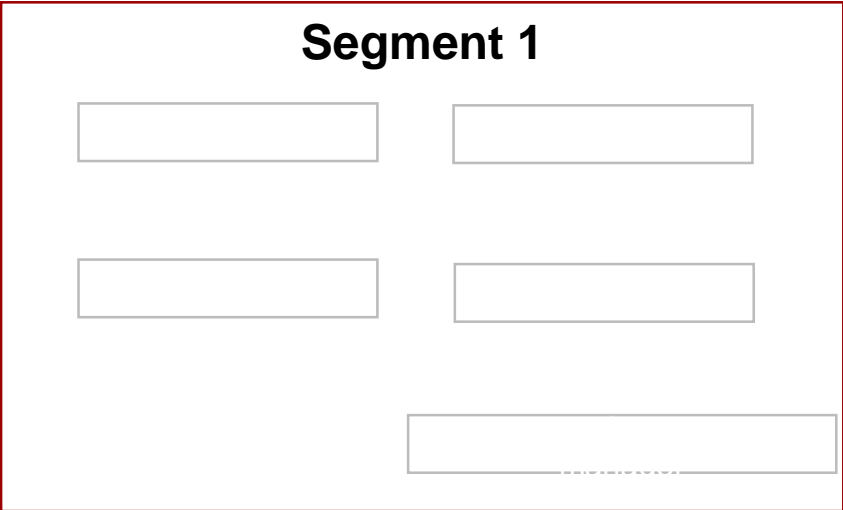
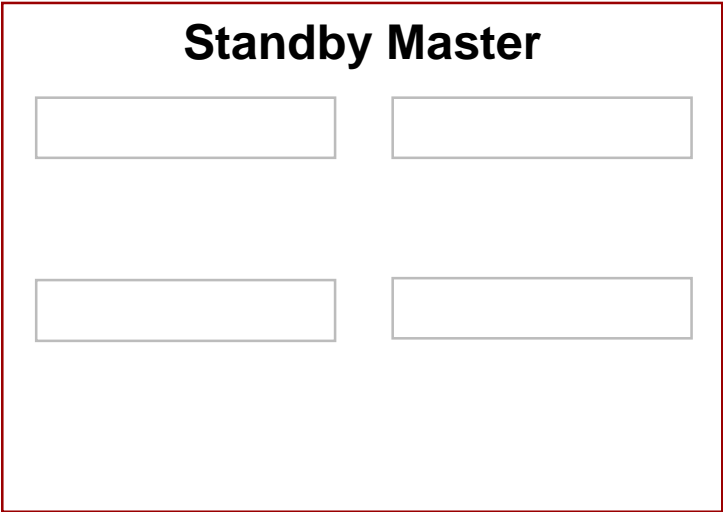
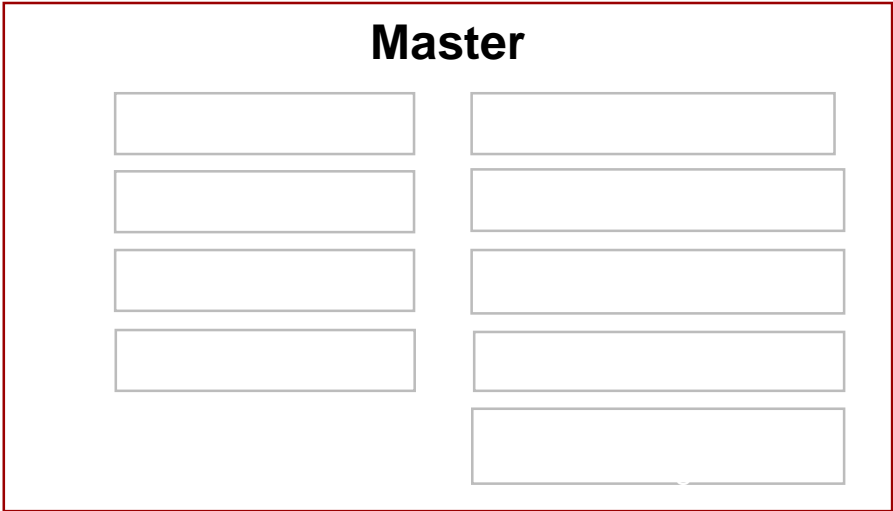


# Apache HAWQ 20 节点部署

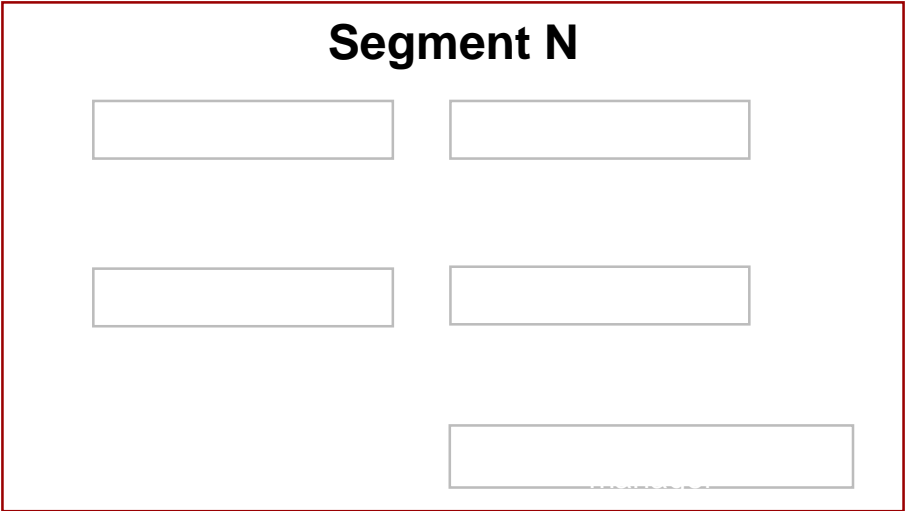


# 03 Apache HAWQ 进 程架构

# 进程架构



...  
...

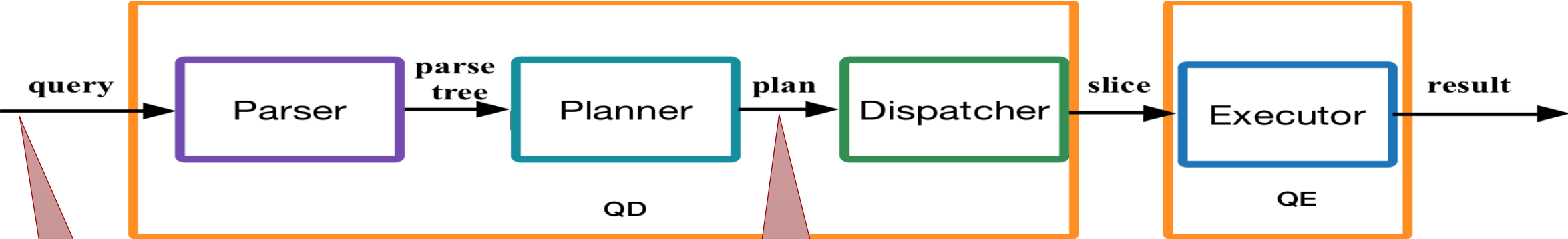


# 进程架构

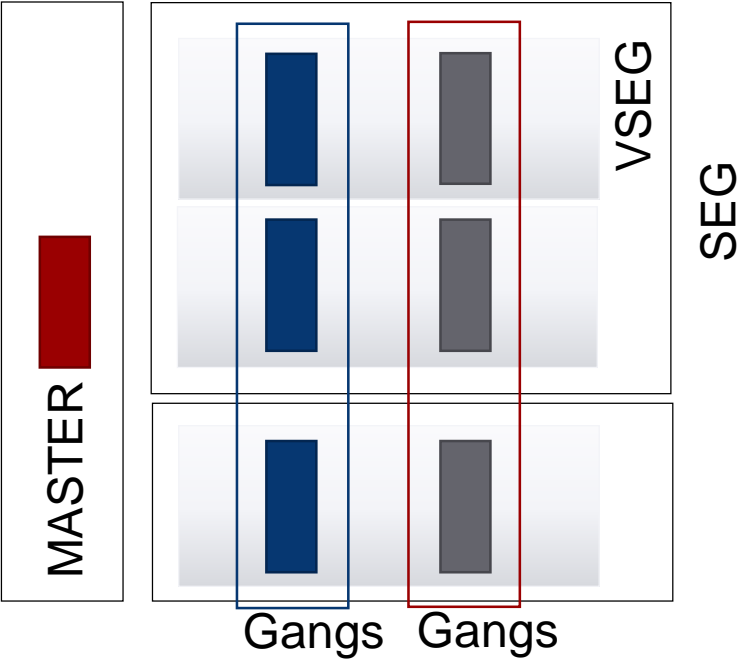
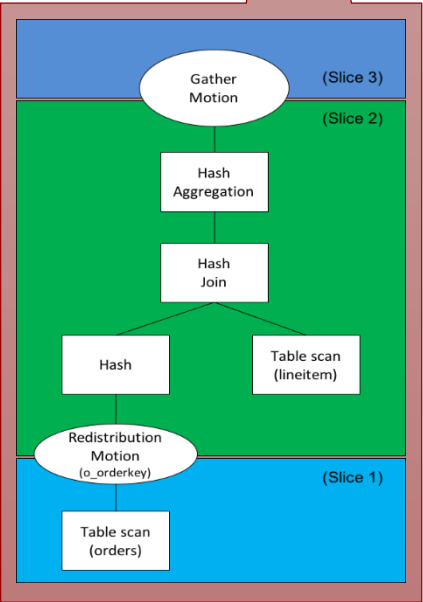
- Postmaster: 监听用户请求的守护进程，为每个客户端连接fork单独的postgres服务，当postgres 进程出错时尝试修复。
- Logger: 负责收集各个子进程的输出，并将其写到pg\_log。
- Stats collector: 统计数据收集进程, 会生成描述数据库，表等统计信息的.stat文件。
- Writer: 就是定期将共享内存的数据写到磁盘上的进程
- Master resource manager: 管理 / 分配 / 回收资源，定期查询 / 接收 / 处理segment心跳信息，从而获取集群可用的节点。
- Segment resource manager: 查询本节点状态，发送 segment心跳信息
- DFS metadata cache: 读取并缓存block location，方便计算data locality信息，从而提高生成查询计划时决定哪个节点上读取哪些数据的速度。
- Checkpoint: 负责周期性做checkpoint或响应常规的checkpoint请求
- Sequenceserver: 负责产生序列的进程
- Wal send server: 负责把write ahead log发给standby master
- Sync agent: 负责和master上wal send server通信的进程，处理Master 和standby 节点状态

# 04 Apache HAWQ 查询执行过程

# 查询执行流程



```
SELECT l_orderkey,
count(l_quantity)
FROM lineitem, orders
WHERE l_orderkey=o_orderkey
AND l_tax>0.01
GROUP BY l_orderkey;
```



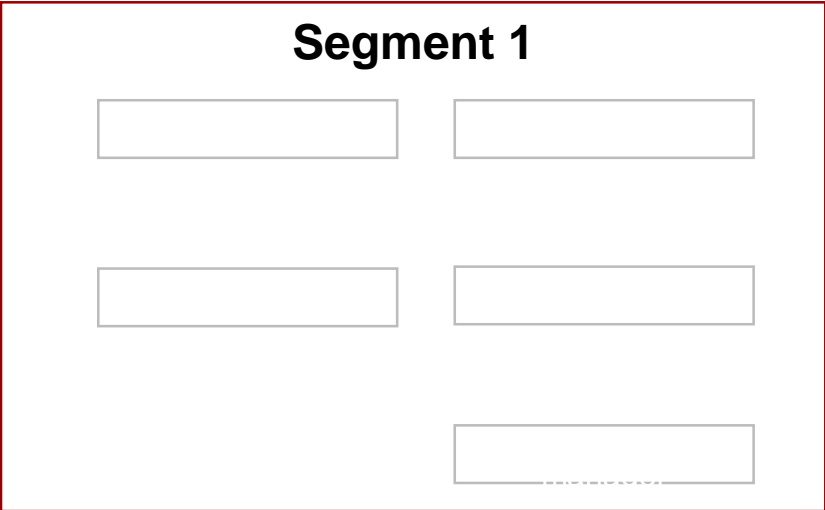
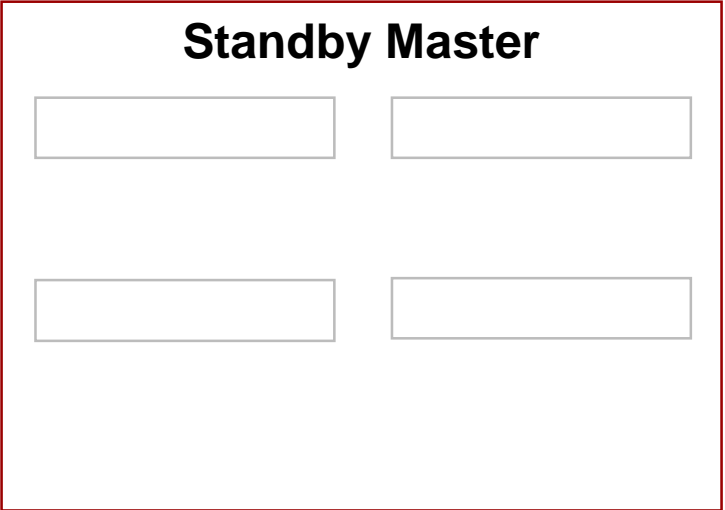
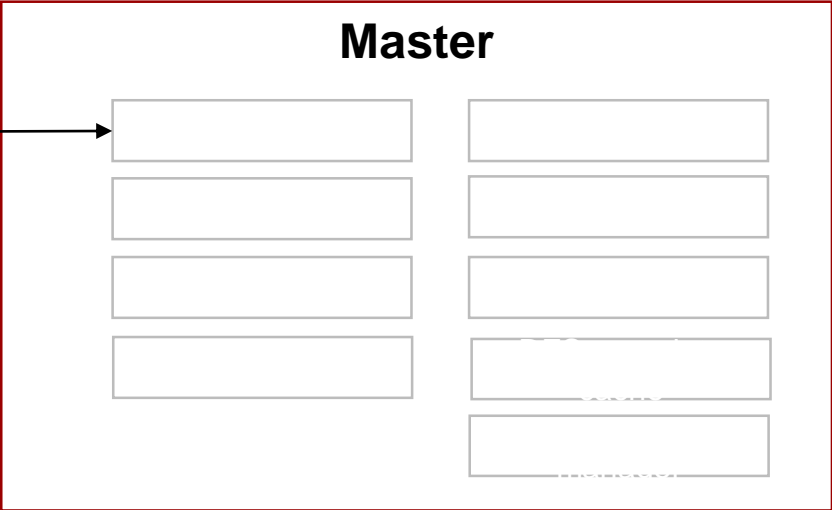


# 查询执行流程

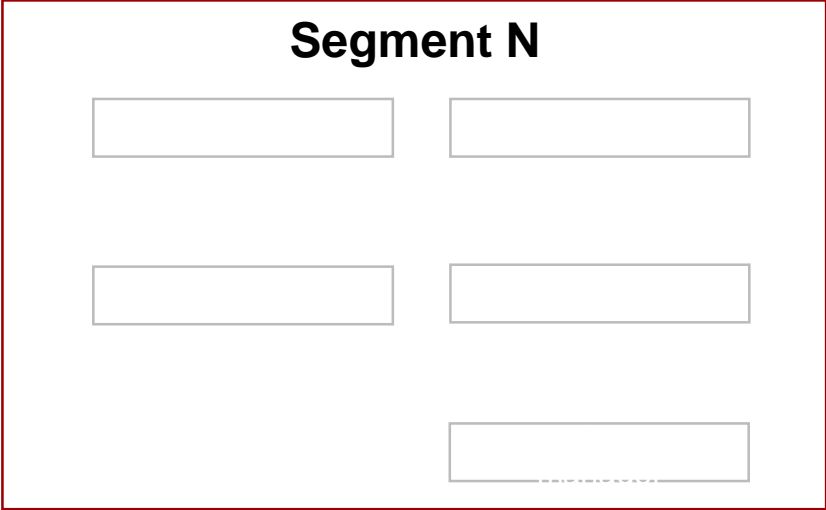


1 connect

客户端连接到  
master节点上的  
postmaster  
进程



...



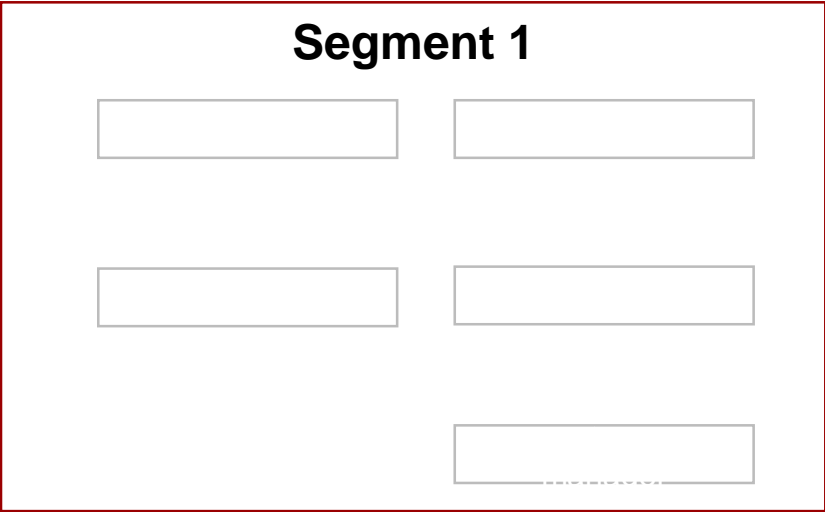
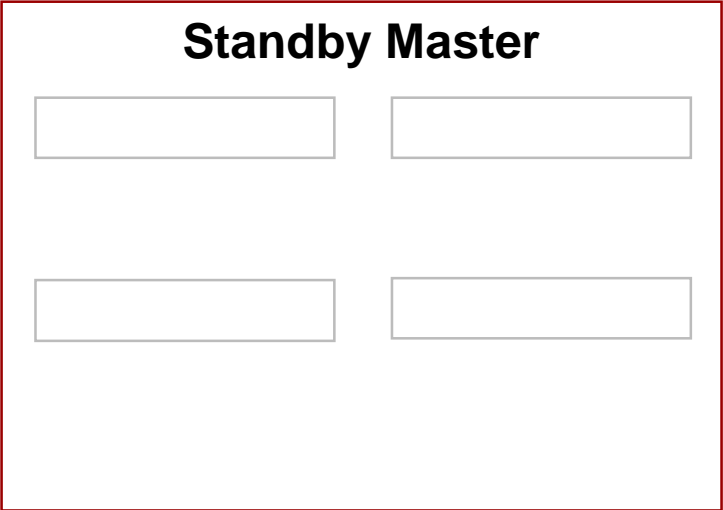
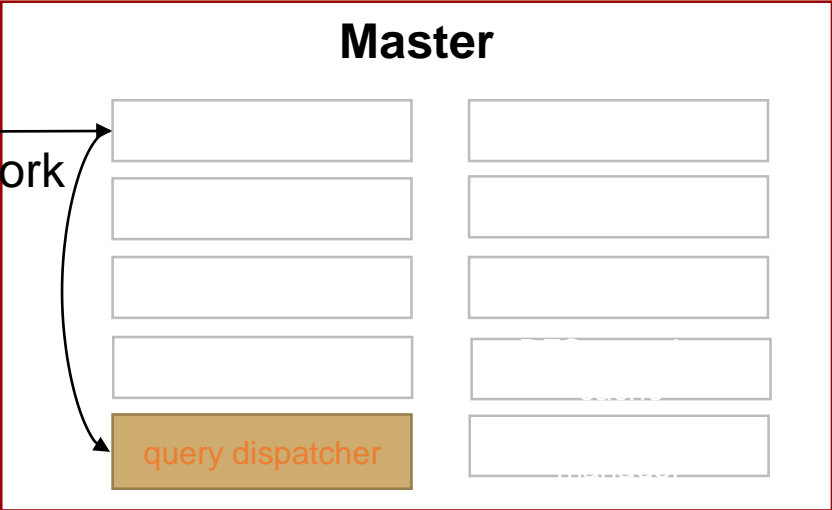
# 查询执行流程



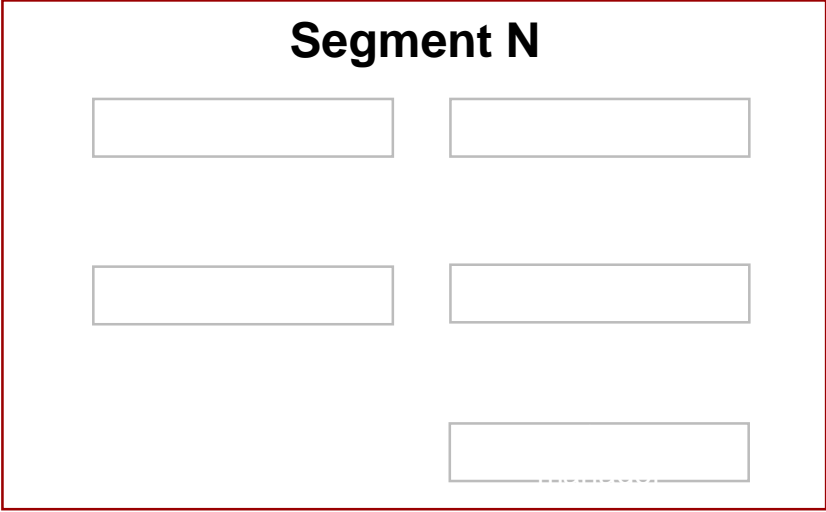
1 connect

2 fork

master节点上的  
postmaster  
fork出query  
dispatcher进程，  
称之为QD 进程



...

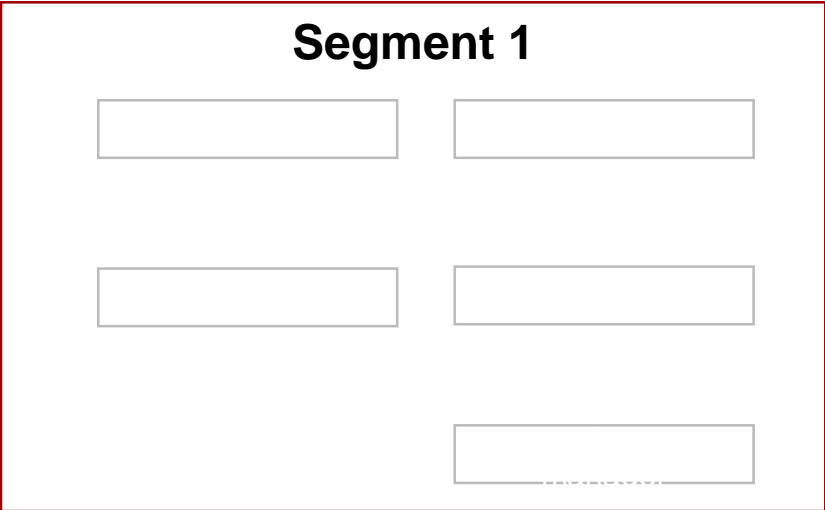
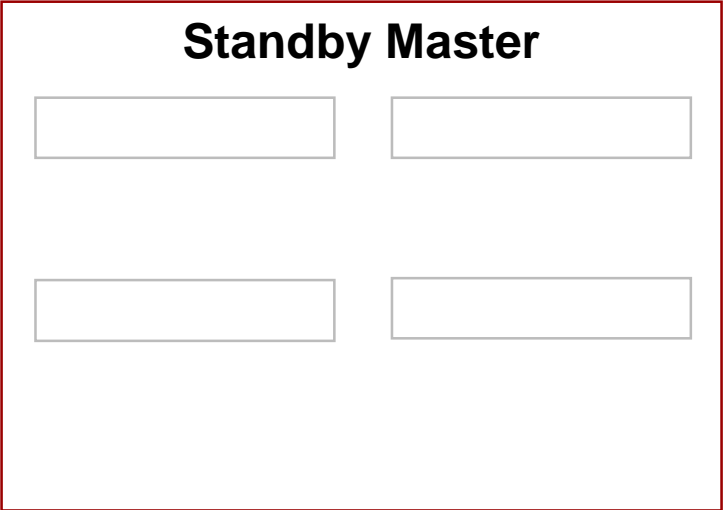
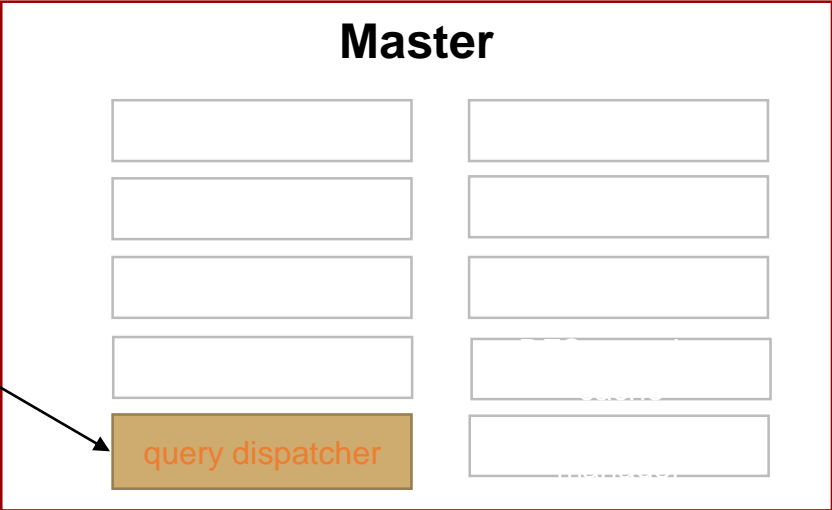


# 查询执行流程

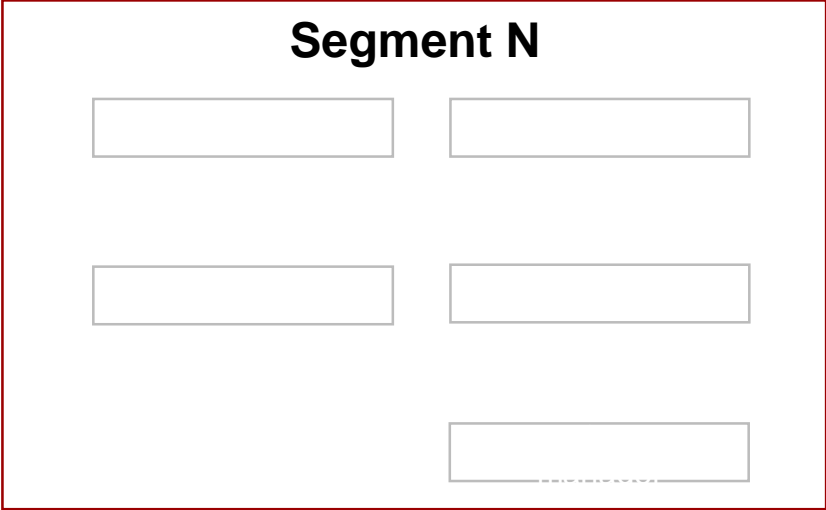


3 connect

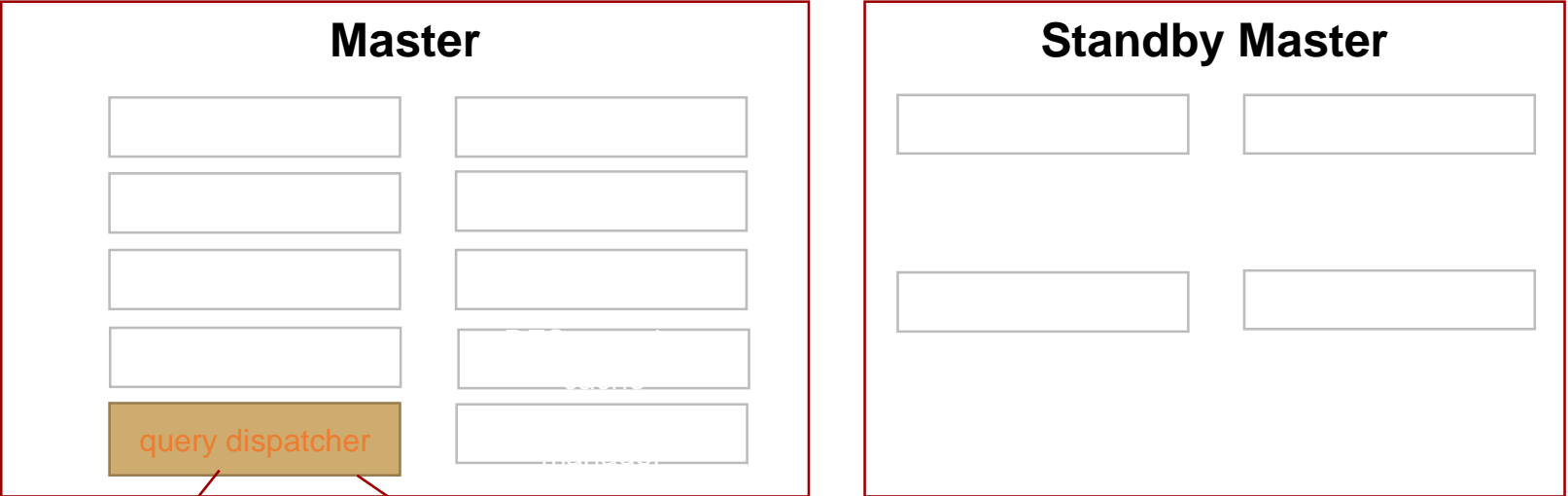
建立client和query dispatcher的连接



...

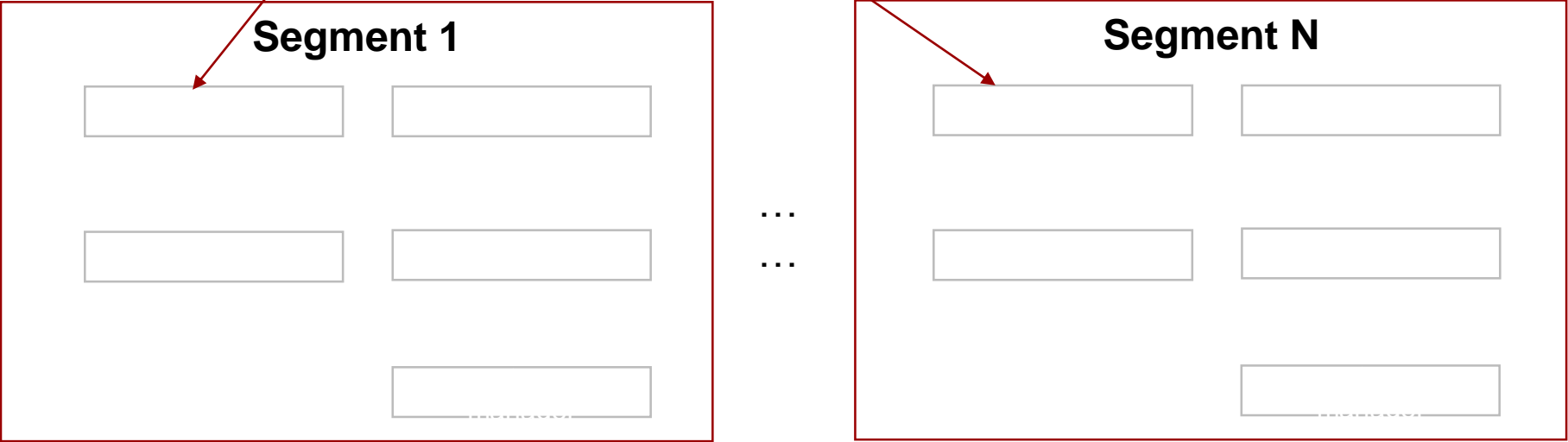


# 查询执行流程

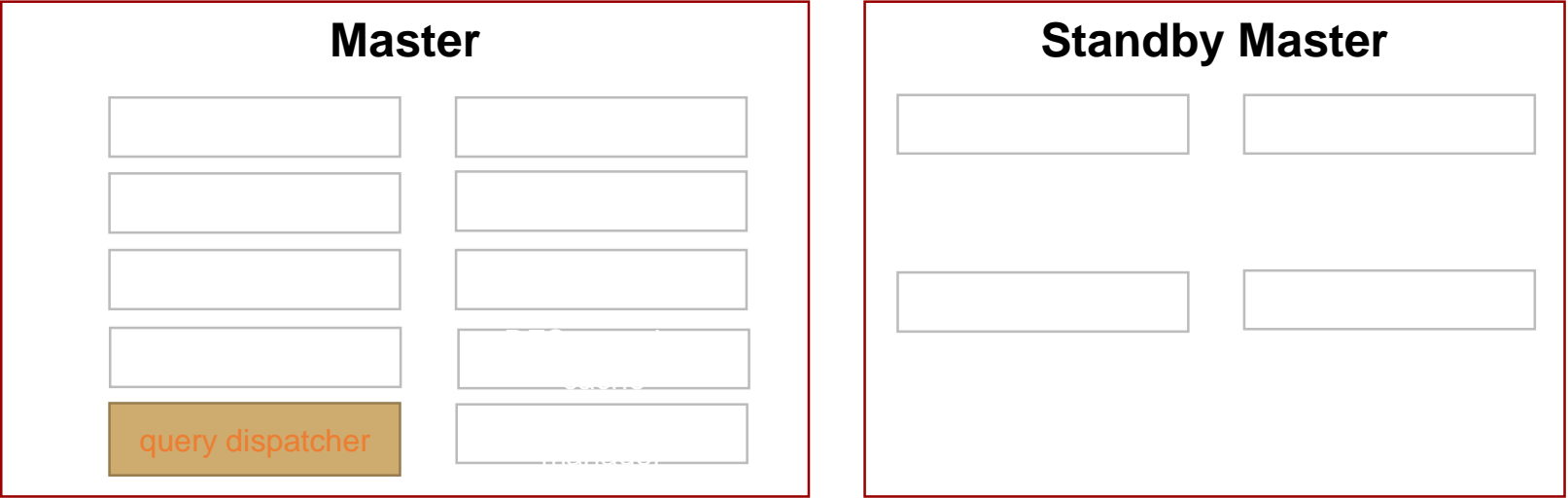


客户端在当前 session 进行 parse, plan, dispatch 等过程将查询计划下发到 segment 节点

4 connect

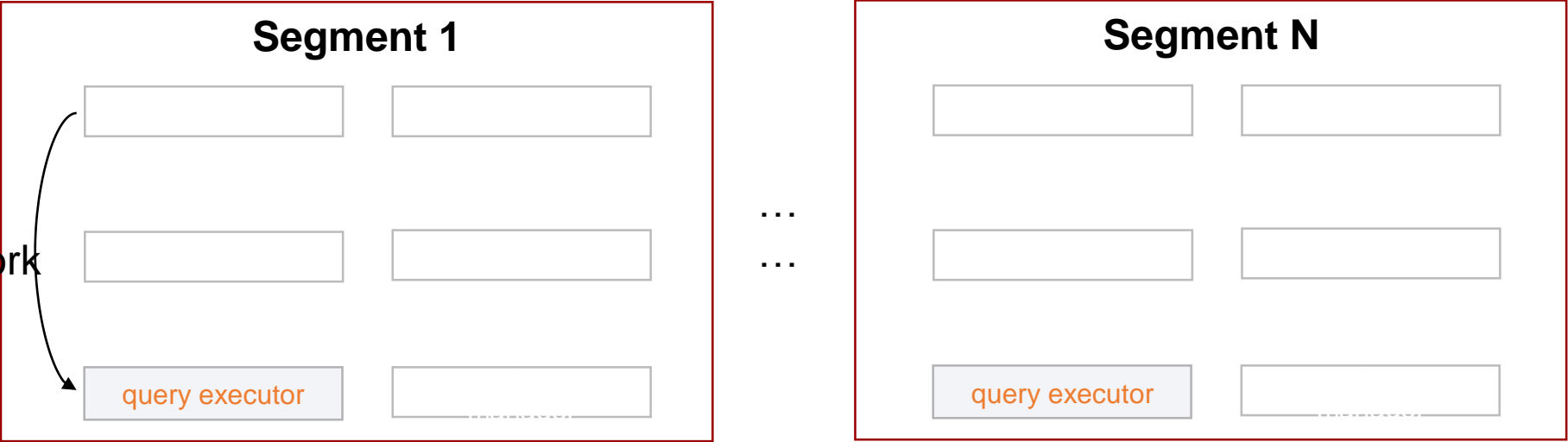


# 查询执行流程

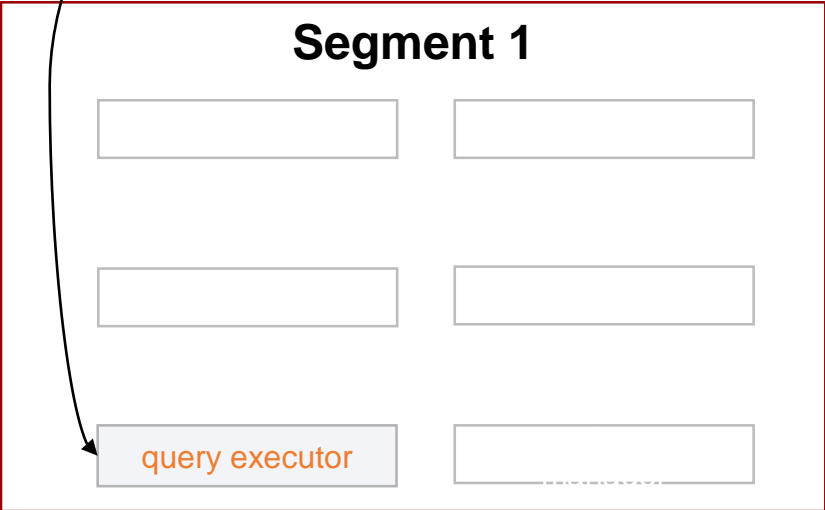
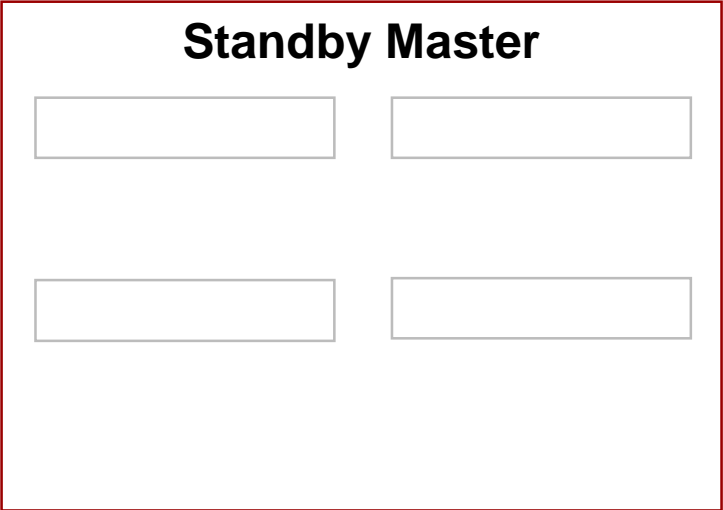
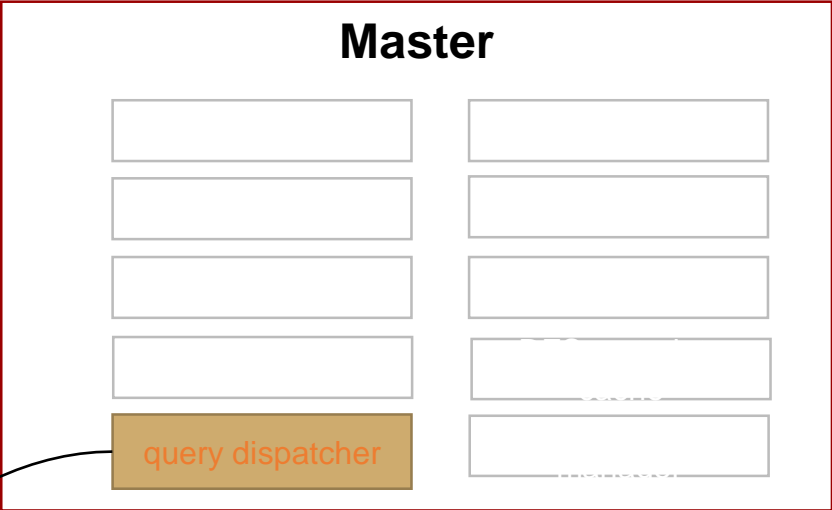


segment节点上的  
postmaster  
fork出query  
executor进程

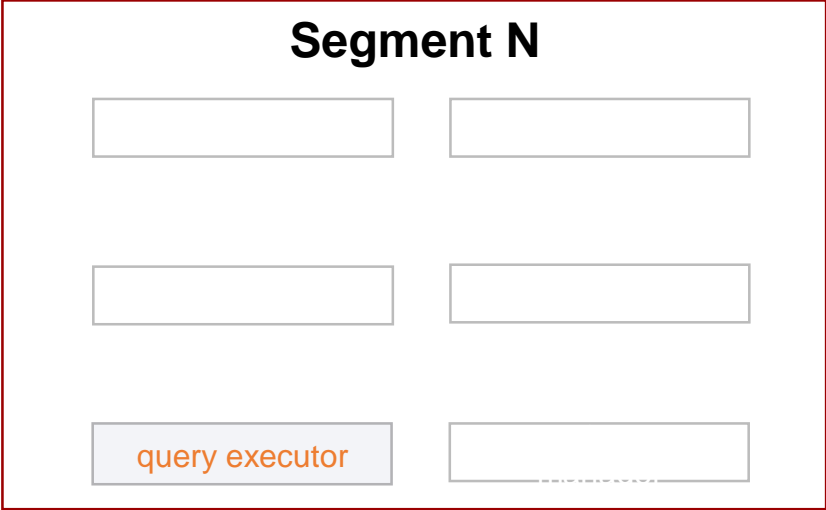
5 fork



# 查询执行流程



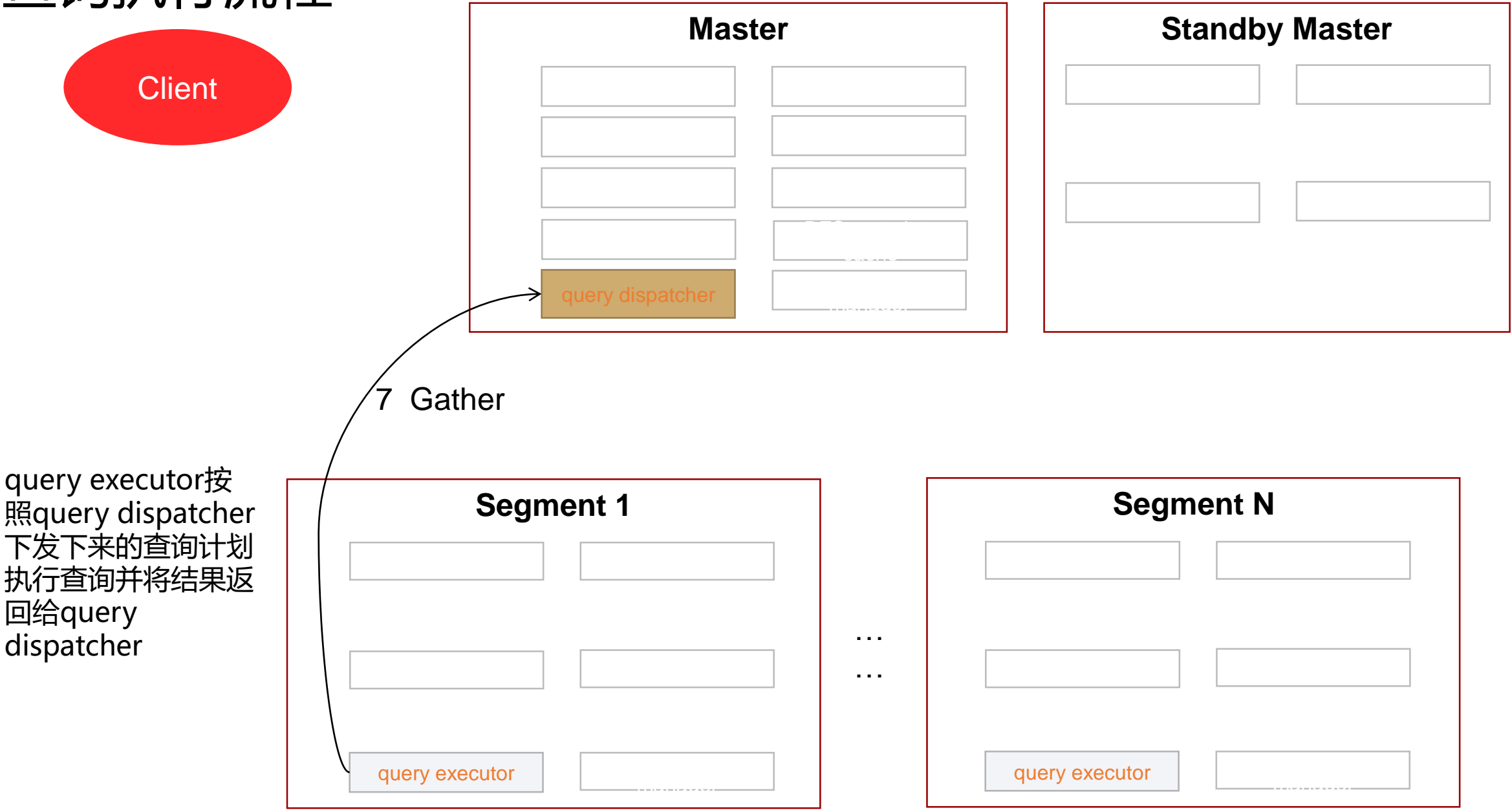
...



query dispatcher到  
postmaster,  
postmaster到  
query executor的  
连接转为query  
dispatcher和query  
executor的连接

6 connect

# 查询执行流程

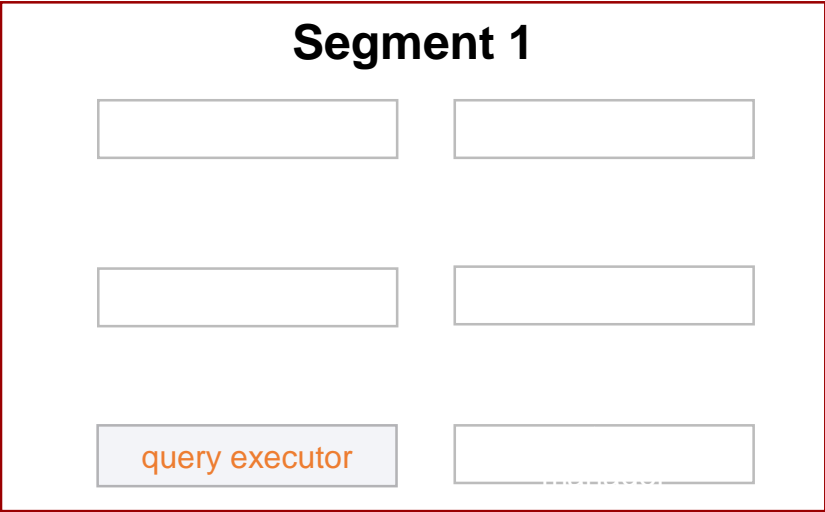
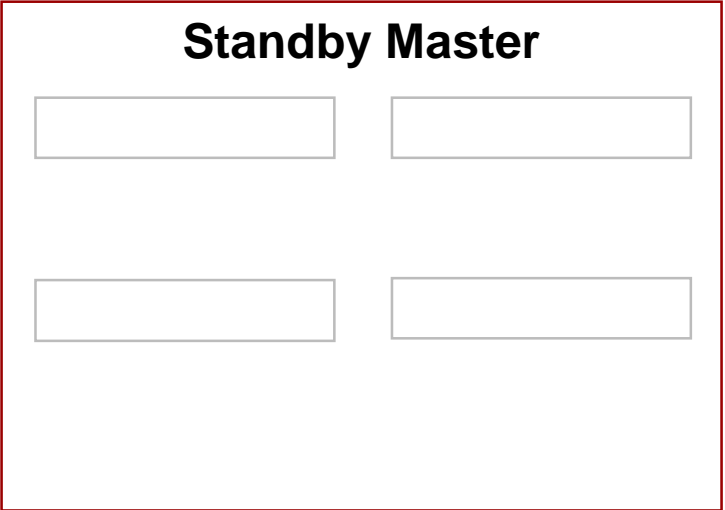
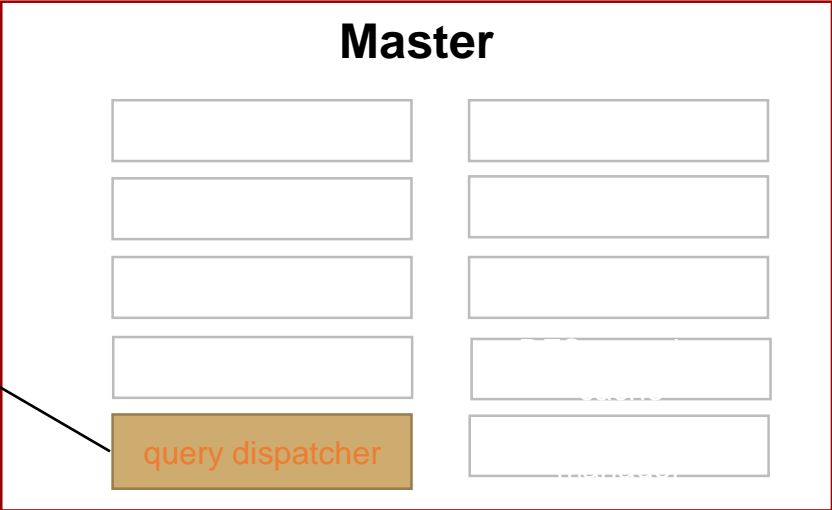


# 查询执行流程

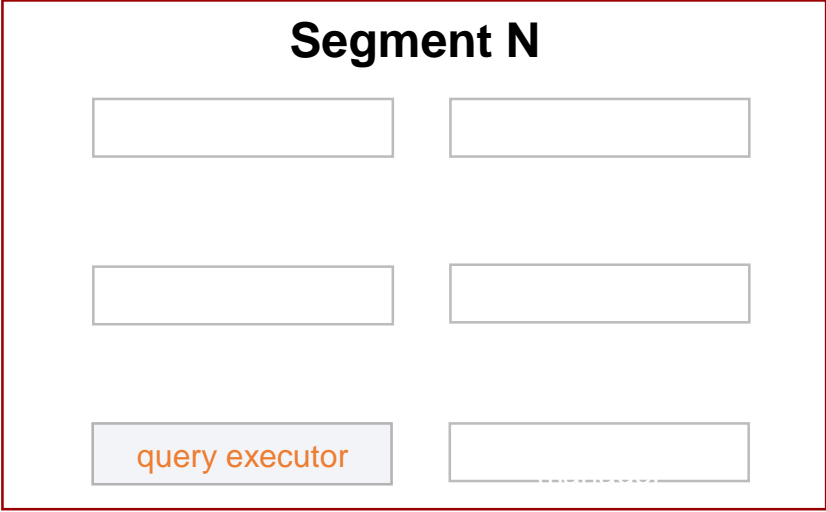


8 return

query dispatcher将  
汇总后的结果返回给  
客户端，从而完成整  
个query的执行



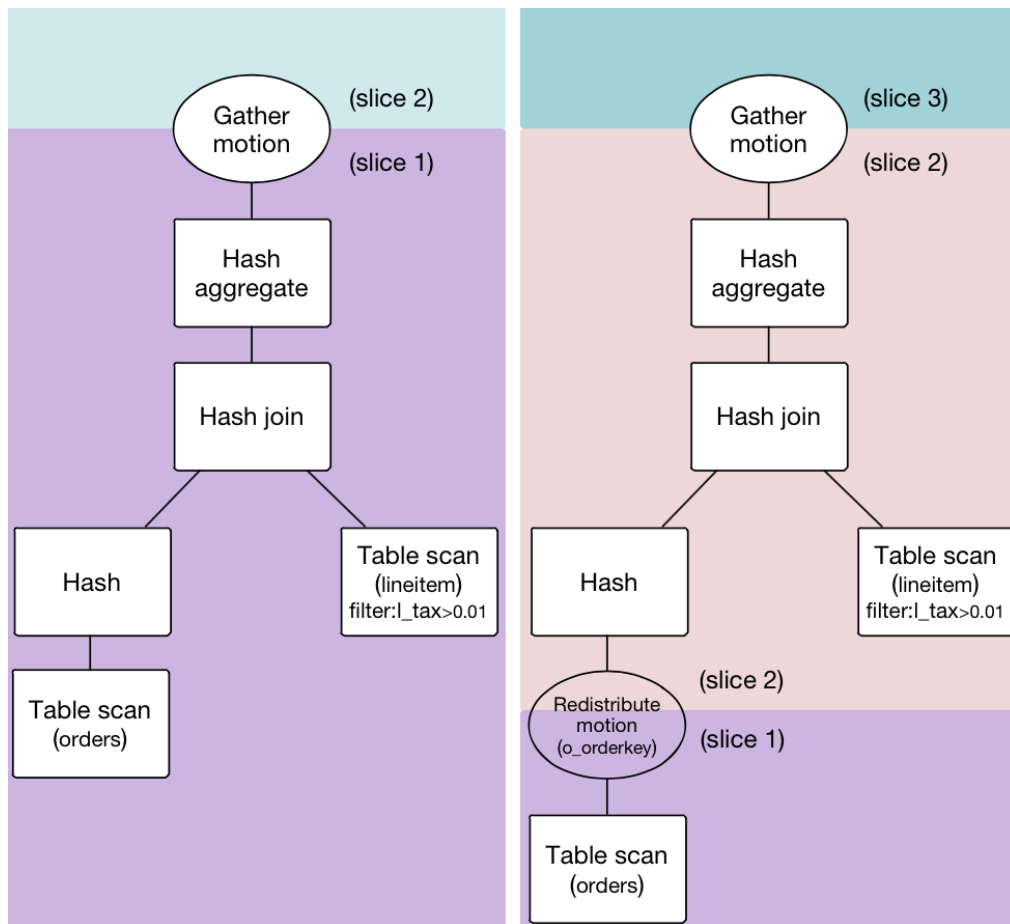
...





# 05 Apache HAWQ 查询例子

# 查询例子

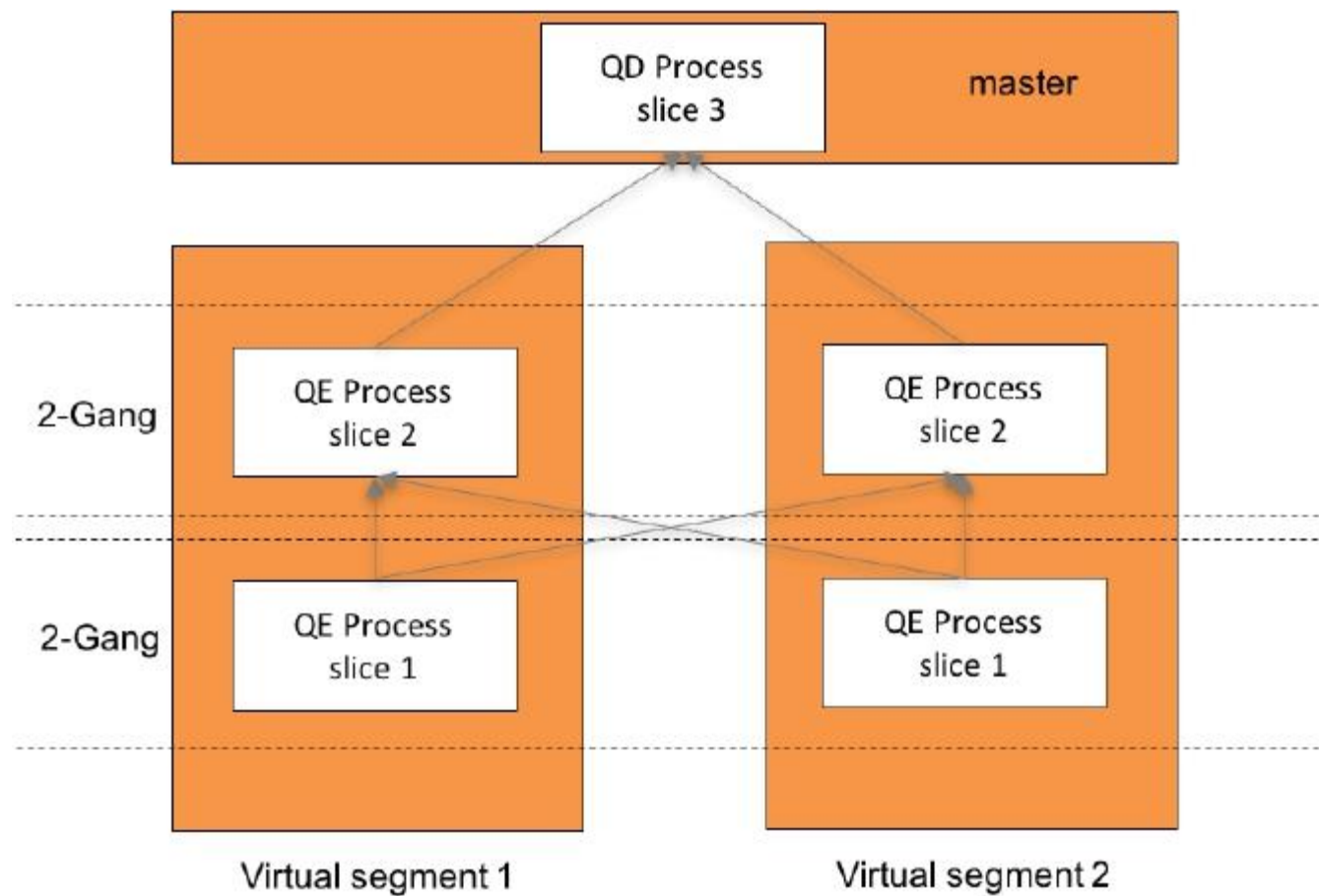


Motion:

- Redistribution
- Broadcast
- Gather
- 每个Motion 划分一个slice

```
SELECT l_orderkey, count(l_quantity)
FROM lineitem, orders
WHERE l_orderkey=o_orderkey AND l_tax>0.01
GROUP BY l_orderkey;
```

# 查询执行流程



# 06 外部接口

# 客户端工具

- PSQL
  - PSQL是 PostgreSQL 中的一个命令行交互式客户端工具。 安装的Apache HAWQ binary 里面会自动安装了PSQL。
- JDBC
  - [可以兼容PG官方JDBC](#)
- ODBC
  - 目前没有提供单独的ODBC, [可以兼容PG官方ODBC](#)
- PGAdmin
  - Postgres 官方UI 客户端

# 07 调试

# 日志

- 初始化 / 启动 / 停止日志
  - /home/gpadmin/hawqAdminLogs/
  - hawq\_init\_日期.log / hawq\_start\_日期.log / hawq\_stop\_日期.log / hawq\_config\_日期.log
- Master 日志
  - 在配置的hawq\_master\_directory 下的pg\_log, 每天一个新的文件, 每次重启一个新的文件。
- Segment 日志
  - 在配置的hawq\_segment\_directory 下的pg\_log, 当query 出错时, 会出现在某个节点的segment 出现问题。这时候需要去对应的segment 上查看相应的log。

# 报使用问题所需要的信息

- 问题描述
  - 重现步骤与预期行为
  - 配置：
  - 节点数 (Master / Segment 个数)
  - 影响的版本号 `select version()` 的输出
  - 如果有coredump，打出错误栈 上传coredump 和对应的binary
  - 出错时间段前后的log。根据出错信息里来决定拿出错节点的对应的log (hawq master log/hawq segment log)



# Q & A

