GREENPLUM DATABASE®

# Greenplum在运营商领域的HTAP实践

苑泽福（阿福）
Greenplum中文社区

2022年4月29日

vmware®
Confidential | ©2021 VMware, Inc.

# Agenda

1. Greenplum HTAP架构特点介绍

2. 项目整体规划及部署架构

3. 客户选型及测试内容分享

4. 新老系统迁移及数据校验

5. 生产运维经验分享

GREENPLUM DATABASE®

# 1. Greenplum HTAP架构特点介绍

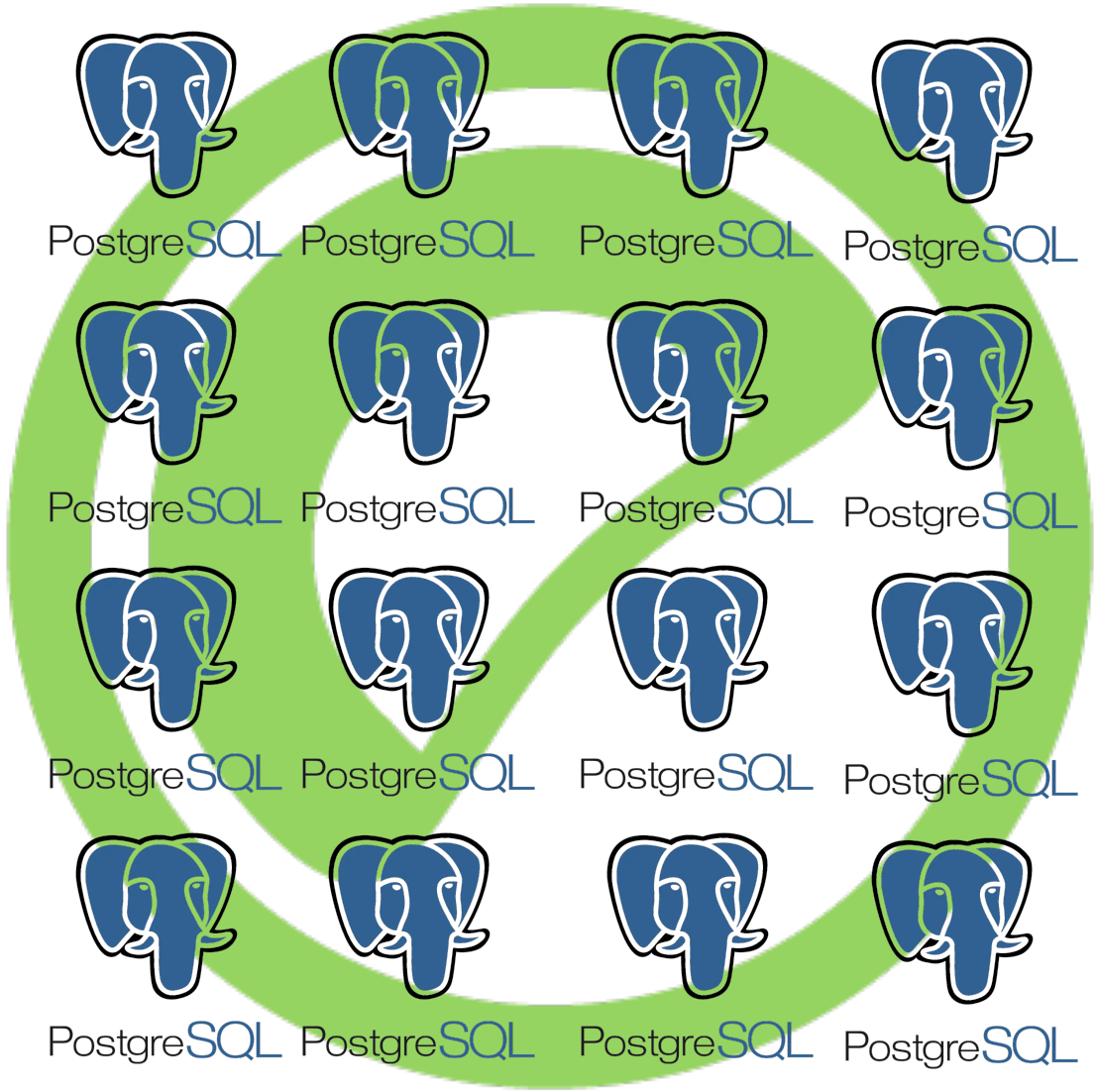
## 世界上最先进的开源MPP数据库

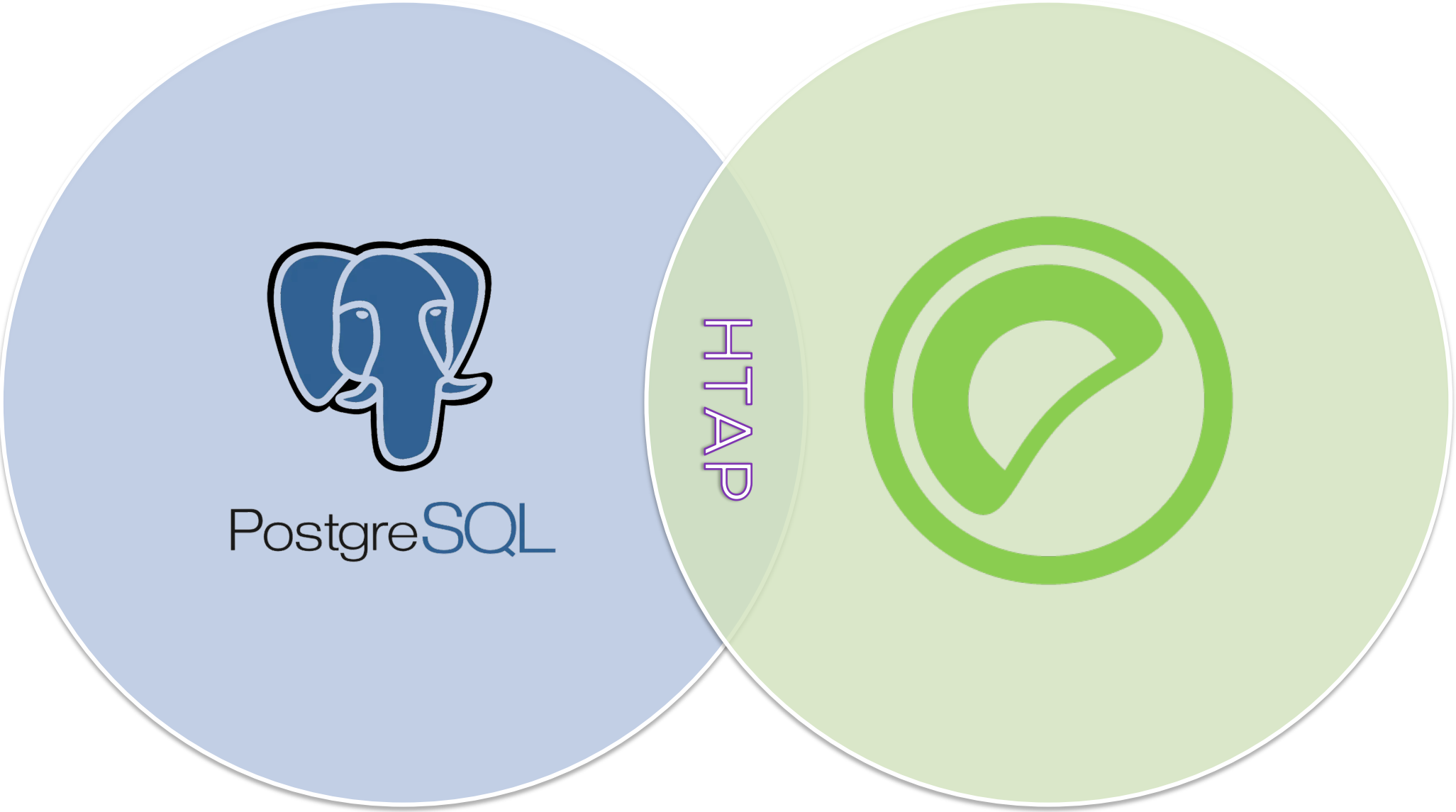

Confidential | ©2021 VMware, Inc.

## 开源解决方案 for OLTP

PostgreSQL

## 开源解决方案 for OLAP



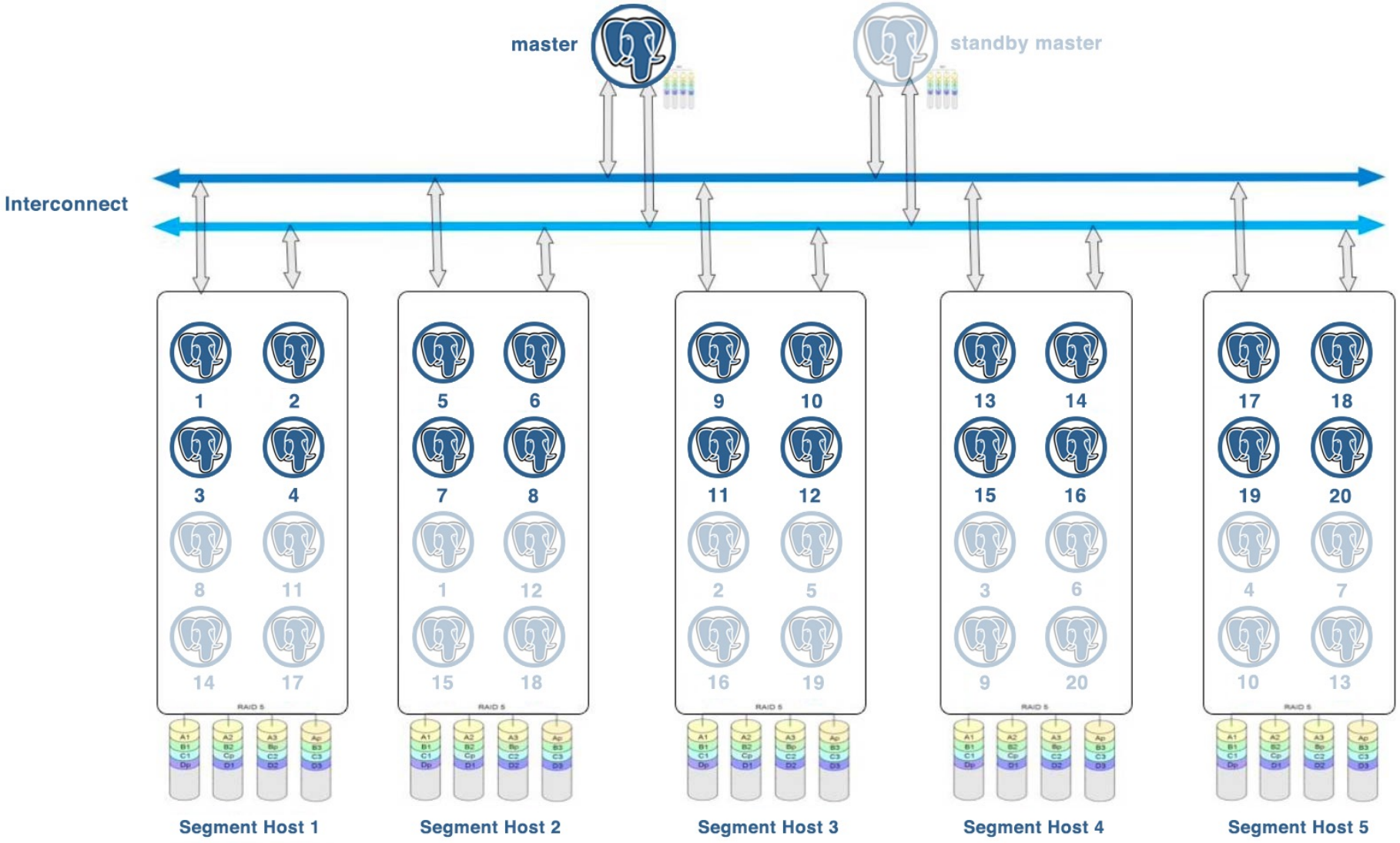




Confidential | ©2021 VMware, Inc.

GREENPLUM DATABASE

# 基于PostareSQL的MPP架构



# 新版本向HTAP进化过程中的一些优化工作

在向HTAP进化过程中，Greenplum内核团队还顺便写了一片论文《Greenplum: A Hybrid Database for Transactional and Analytical Workloads》，该论文入选了2021 西安 SIGMOD，大家可以到网上去寻找正华之前分享的内容，了解更多。

- 全局死锁检测(GDD)
- 事务优化
- 复制表
- 内核升级

老版本上update使用的是表锁，性能较差，通过全局死锁检测技术，降低为行级锁，性能大幅提升。GDD的基本逻辑是收集每个segment锁依赖关系图，在master上重建整个集群的锁依赖关系图，并检测是否有环。

只读事务不需要分布式快照，也不需要2PC； 而单节点查询也可以对两阶段提交进行优化。

复制表即每个segment都有表的全部数据，因而不需要网络数据传输，还可使用索引。

2005年研发之初时其内核版本是PostgreSQL 7；

2017年发布的Greenplum 5之前一直是PostgreSQL 8.2版本；2017年9月发布的Greenplum 5，内核升级到了8.3；

当前广泛使用的Greenplum 6，内核升级到9.4

正在研发的Greenplum 7，内核将会升级到12。

GREENPLUM DATABASE

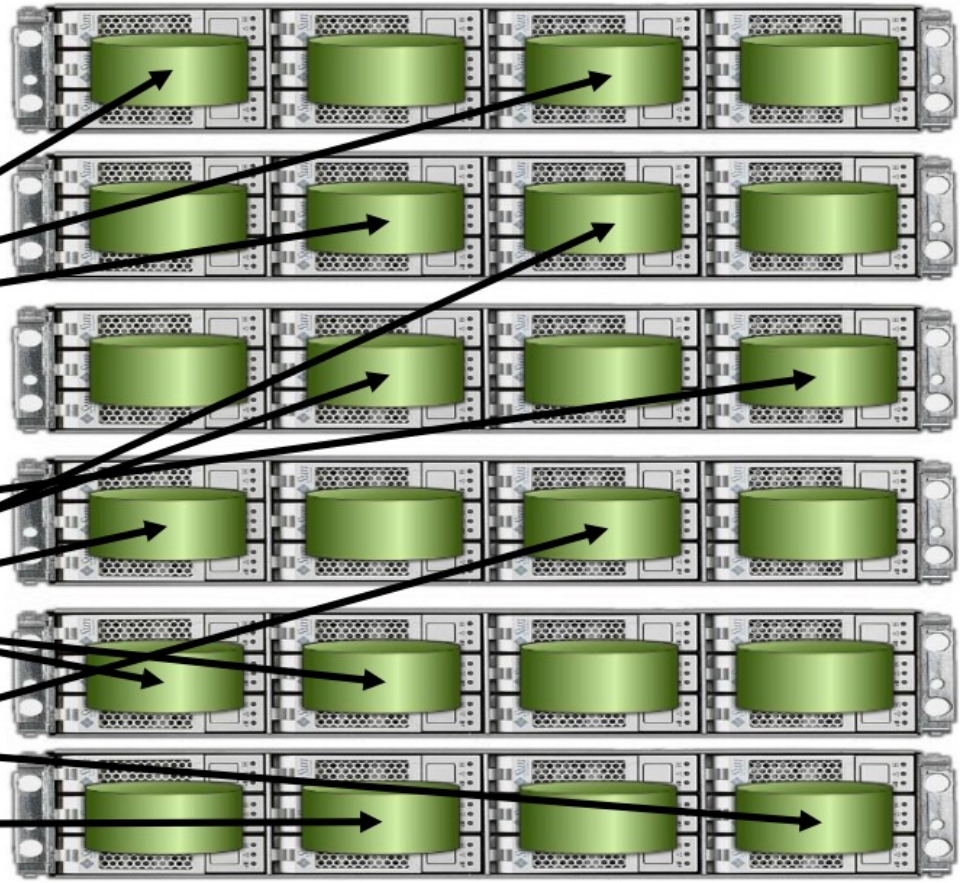
# 新版本向HTAP进化过程中的一些优化工作

| 特性 | Resource group | Resource queue |
|---|---|---|
| 并发控制单元 | transaction | statement |
| 并发控制死锁 | no | yes |
| CPU 分配 | Percentage, cgroup | Priority |
| CPU: Burst | yes | some |
| CPU: CPUSET 绑定 | yes | no |
| Memory: 严格内存控制 | yes | no |
| Memory: 组内内存共享 | yes | no |
| Memory: 什么时候等待 | No concurrency slot or memory quota | No concurrency slot |
| 磁盘配额 | 是 | 否 |

GREENPLUM DATABASE

# 常规的OLAP特性 - 数据分布

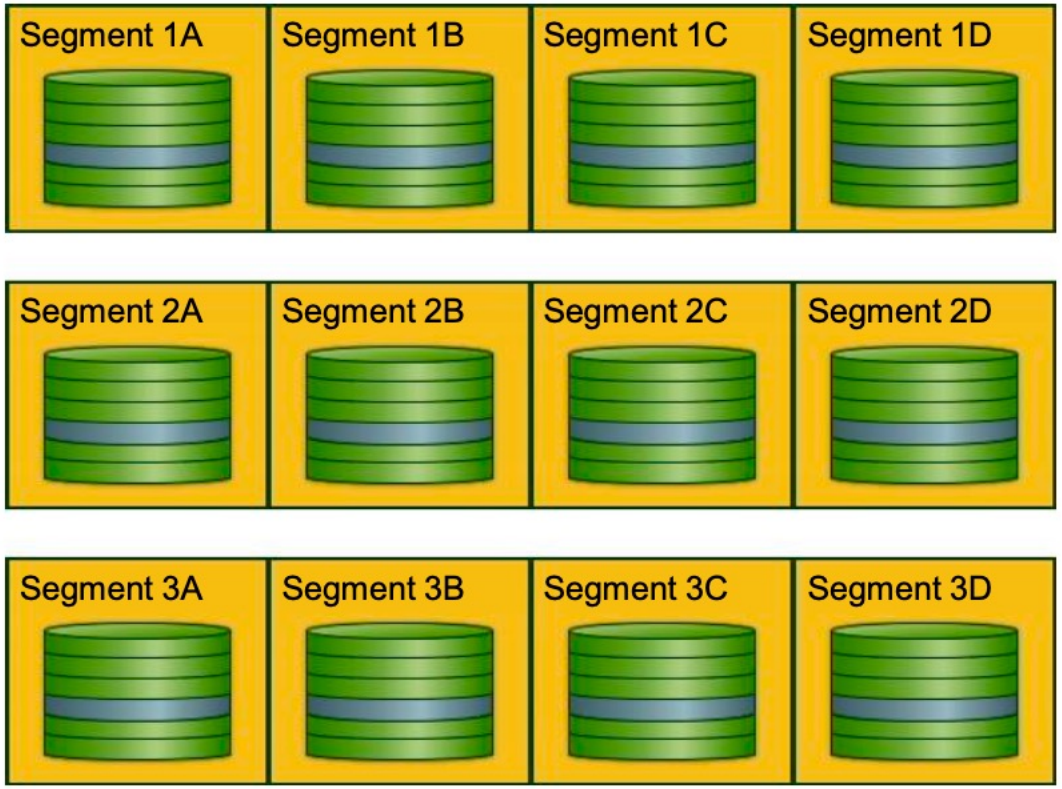
| Order | | |
|---|---|---|
| Order # | Order Date | Customer ID |
| 43 | Oct 20 2005 | 12 |
| 64 | Oct 20 2005 | 111 |
| 45 | Oct 20 2005 | 42 |
| 46 | Oct 20 2005 | 64 |
| 77 | Oct 20 2005 | 32 |
| 48 | Oct 20 2005 | 12 |
| 50 | Oct 20 2005 | 34 |
| 56 | Oct 20 2005 | 213 |
| 63 | Oct 20 2005 | 15 |
| 44 | Oct 20 2005 | 102 |
| 53 | Oct 20 2005 | 82 |
| 55 | Oct 20 2005 | 55 |



GREENPLUM DATABASE

# 常规的OLAP特性 - 多级分区

```
SELECT COUNT(*)
  FROM orders
 WHERE order_date >= 'Oct 20 2007'
   AND order_date <  'Oct 27 2007'
```



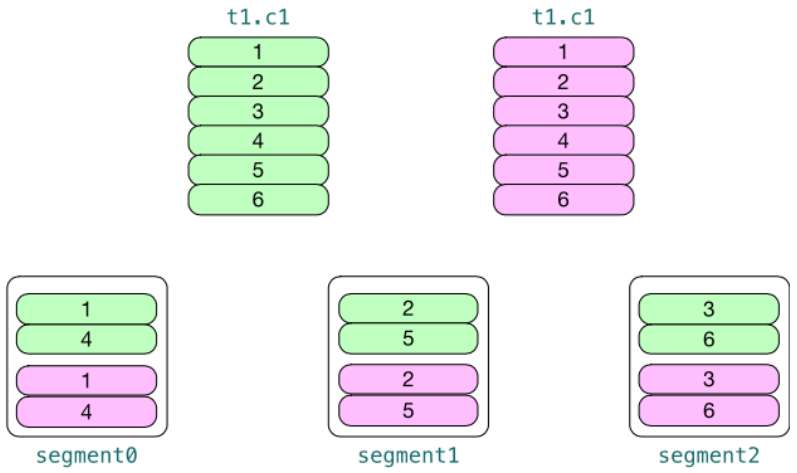
# 常规的OLAP特性 - 多模存储

- 适合 OLTP 业务
- 适合频繁更新或者访问大部分字段的场景

- 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，zstd

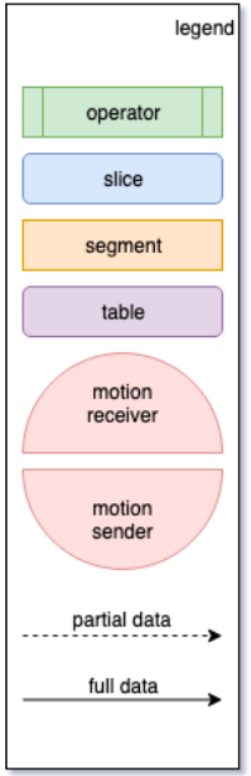
- 历史数据和不常访问的数据存储在 HDFS 或者其他外部系统中
- 无缝查询所有数据
- Text，CSV，Binary，Avro，Parquet，ORC 格式

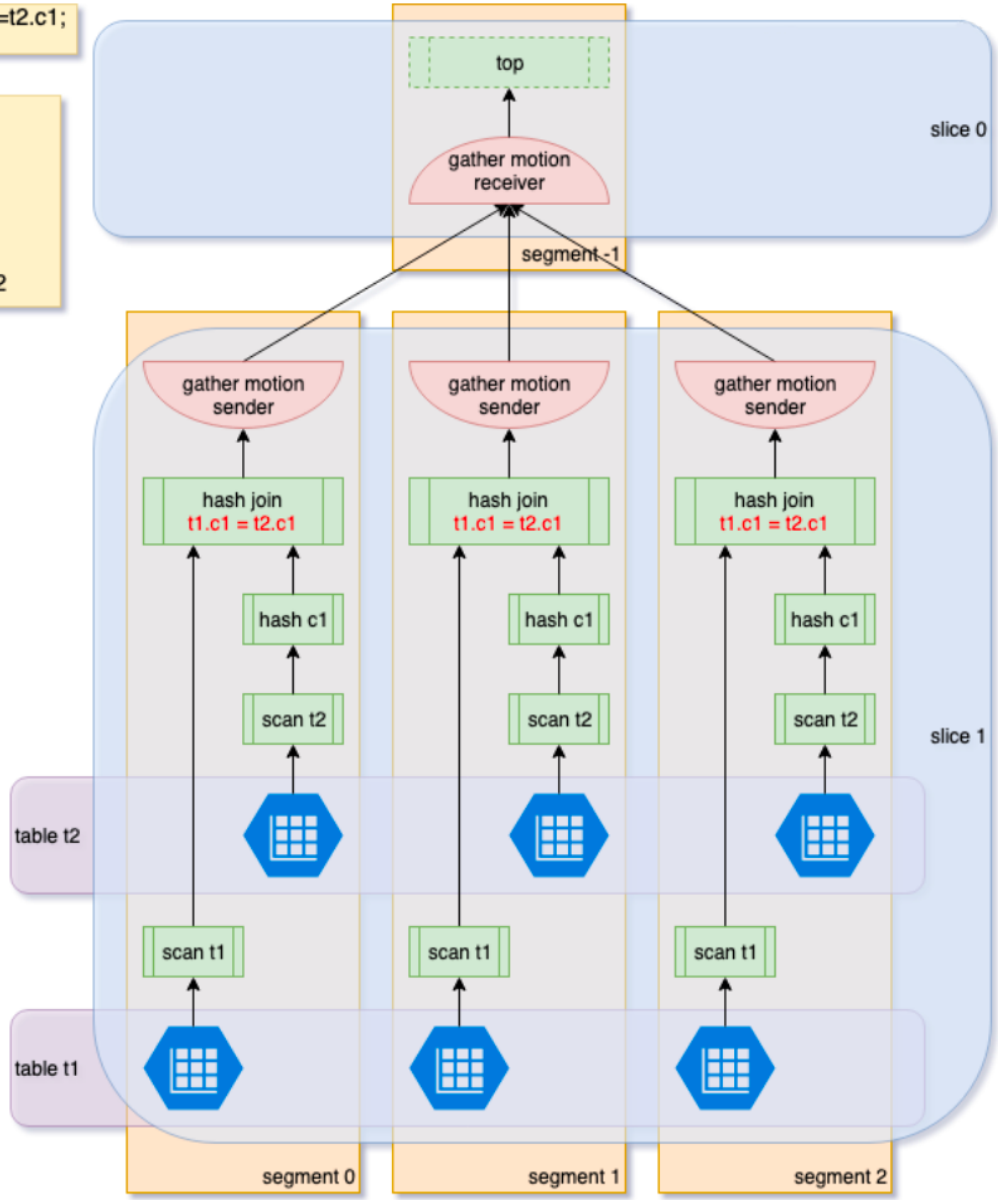
# 常规的OLAP特性 - 并行




select * from t1 join t2 on t1.c1=t2.c1;
(Distribution key: t1.c1, t2.c1)

Gather Motion 3:1 (slice1; segments: 3)
-> Hash Join
Hash Cond: t1.c1 = t2.c1
-> Seq Scan on bar t1
-> Hash
-> Seq Scan on bar t2



GREENPLUM DATABASE

# 2. 项目整体规划及部署架构

## 对Greenplum在项目中的位置整体了解

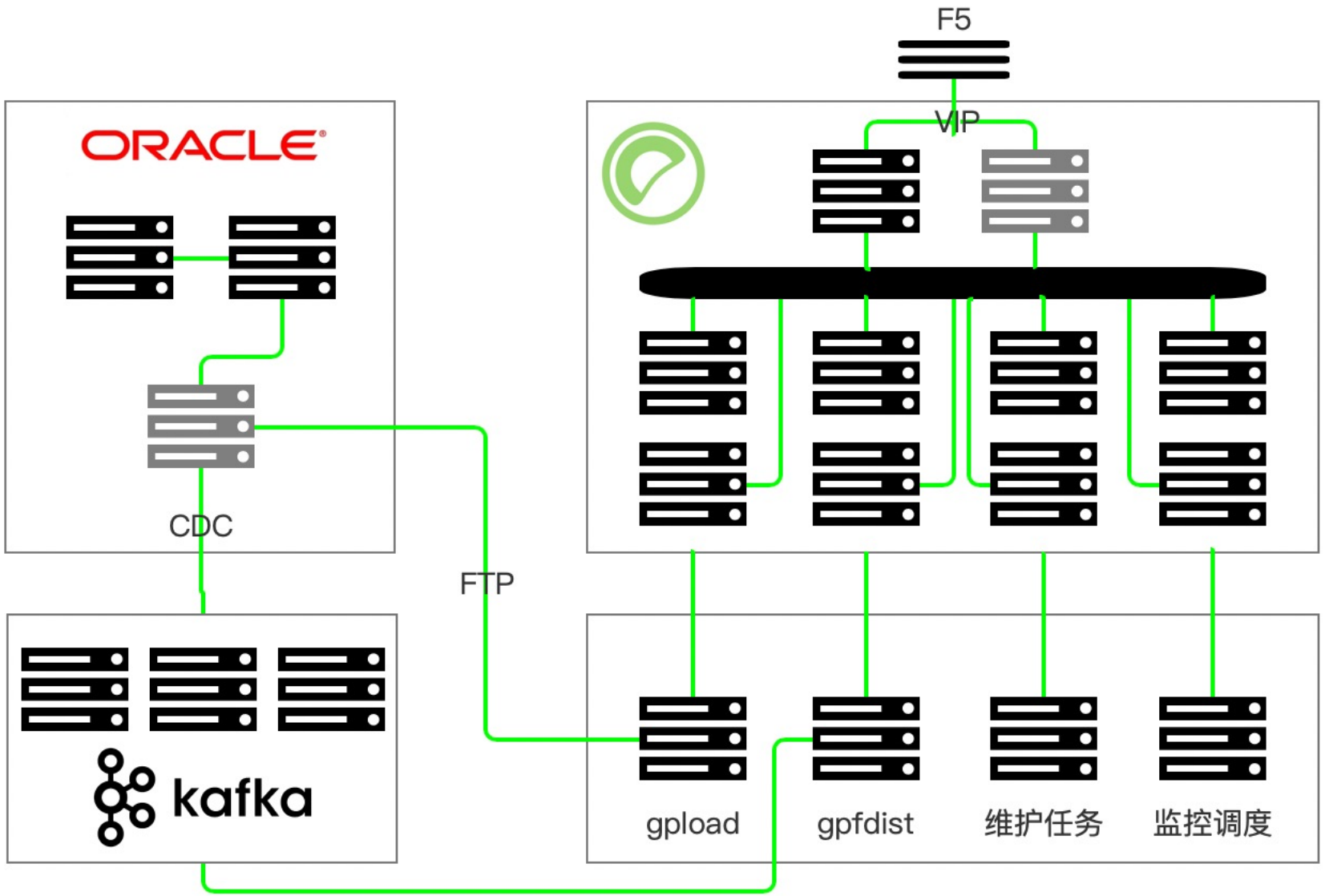
# 规划与部署

机器配置

- Master 2台
- Segment 8台
- ETL 4台
- Kafka 3台

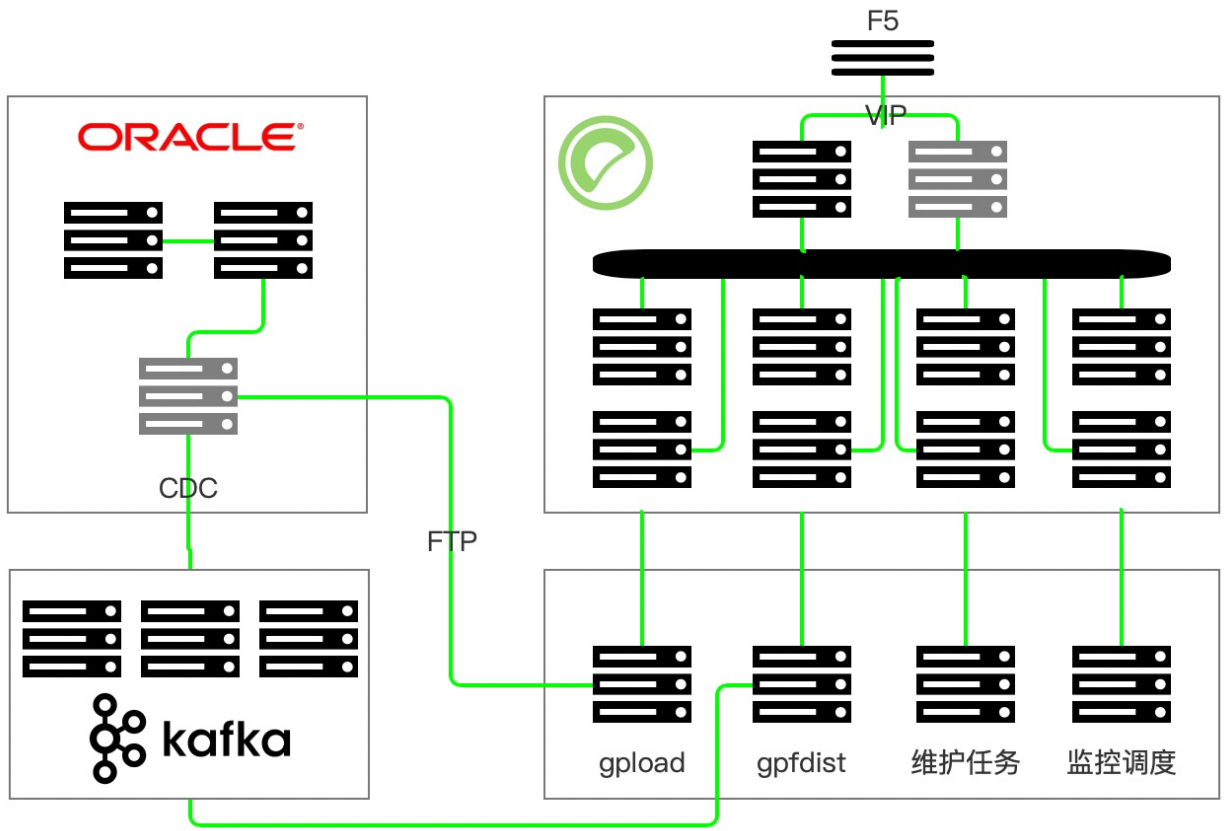
- CPU 48核
- 内存 256GB
- 双RAID5 50TB
- 双网卡绑定

- 4P+4M



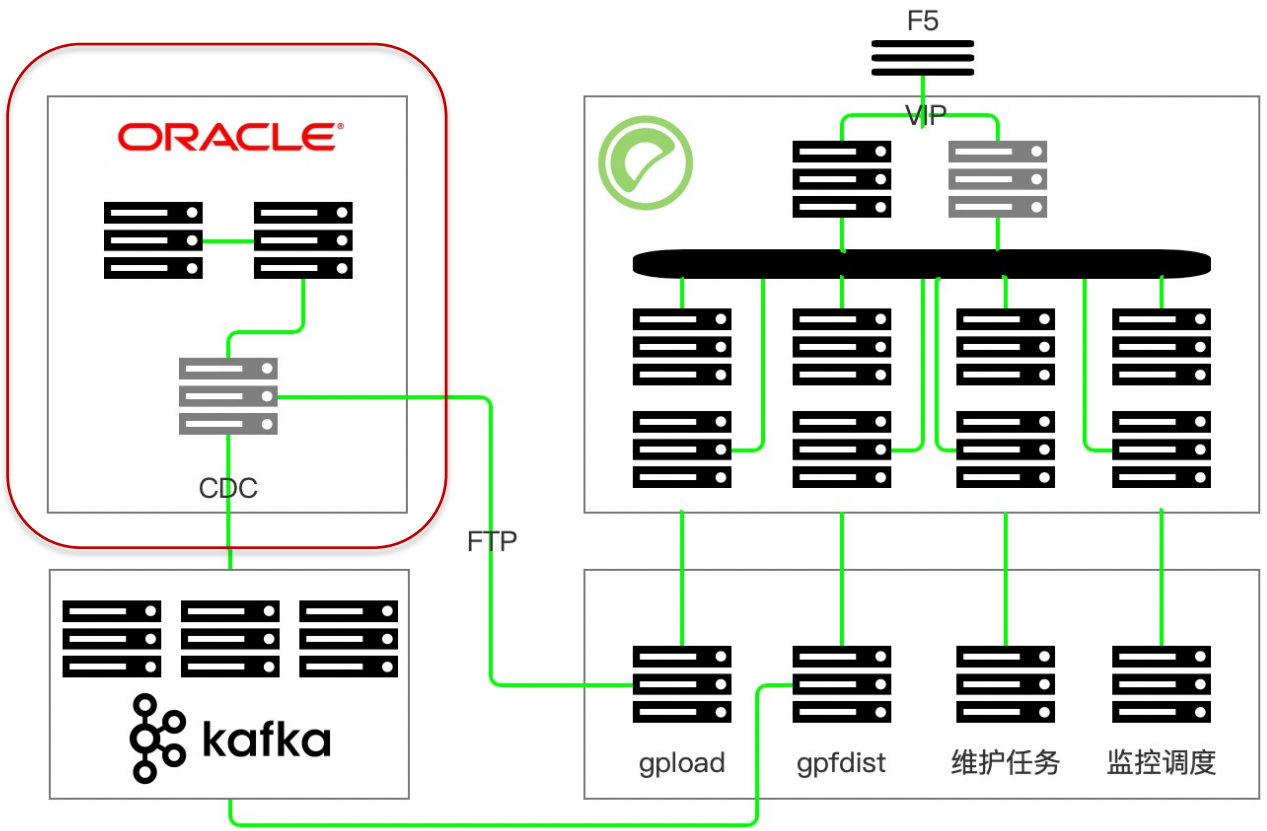
# 规划与部署 - 整体架构的考虑

- 首先客户是体量巨大的运营商，数据量是可观的，所以三、五台机器只能应急，不能满足长远数据增长的规划，所以机器的数量设置为2+8+4；
- 其次，客户是对现有系统进行Oracle替换改造，所以可能有长时间的并行运行要求，选用CDC与FTP是为了保证数据的实时性与双写；
- 3节点Kafka集群基本可以保证3个月的增量日志，方便数据冗余和追溯；
- FTP服务器充分考虑了高可用的要求，gpload和gpfdist入库均做了冗余。



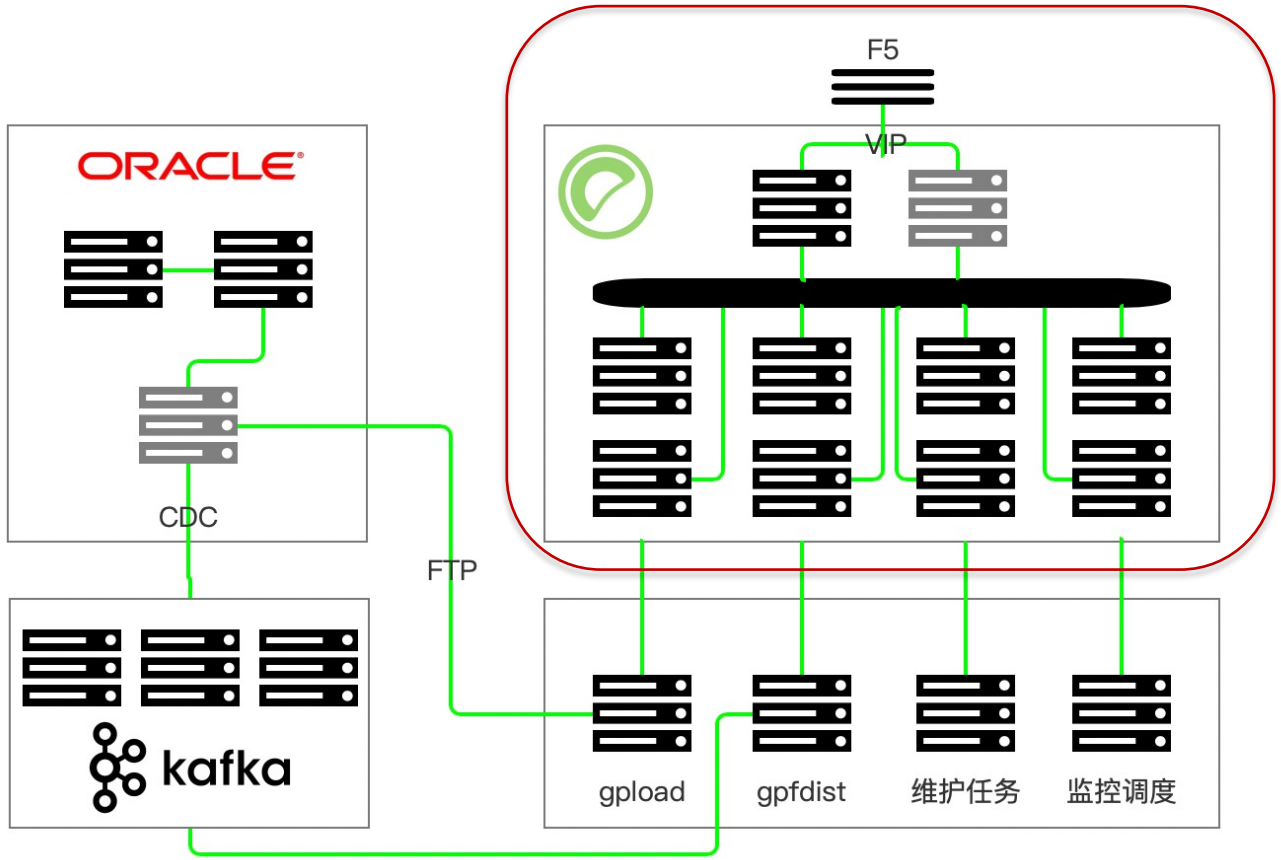
# 规划与部署 - Oracle的现状

- 客户当前Oracle服务器环境是双机RAC，能够，暂时还能满足业务要求，但是由于每天的跑批任务耗费时间巨大，有时到中午跑批任务仍然不能完成；
- 经常出现莫名其妙的Oracle卡顿故障，多次调优仍然不能解决；
- 经过清理的存量数据仍然达到100TB左右，对存储和备份的要求都很高，每天增量数据当时位于500～800GB，并有递增趋势；
- 事实导致已经不得不考虑替代方案。



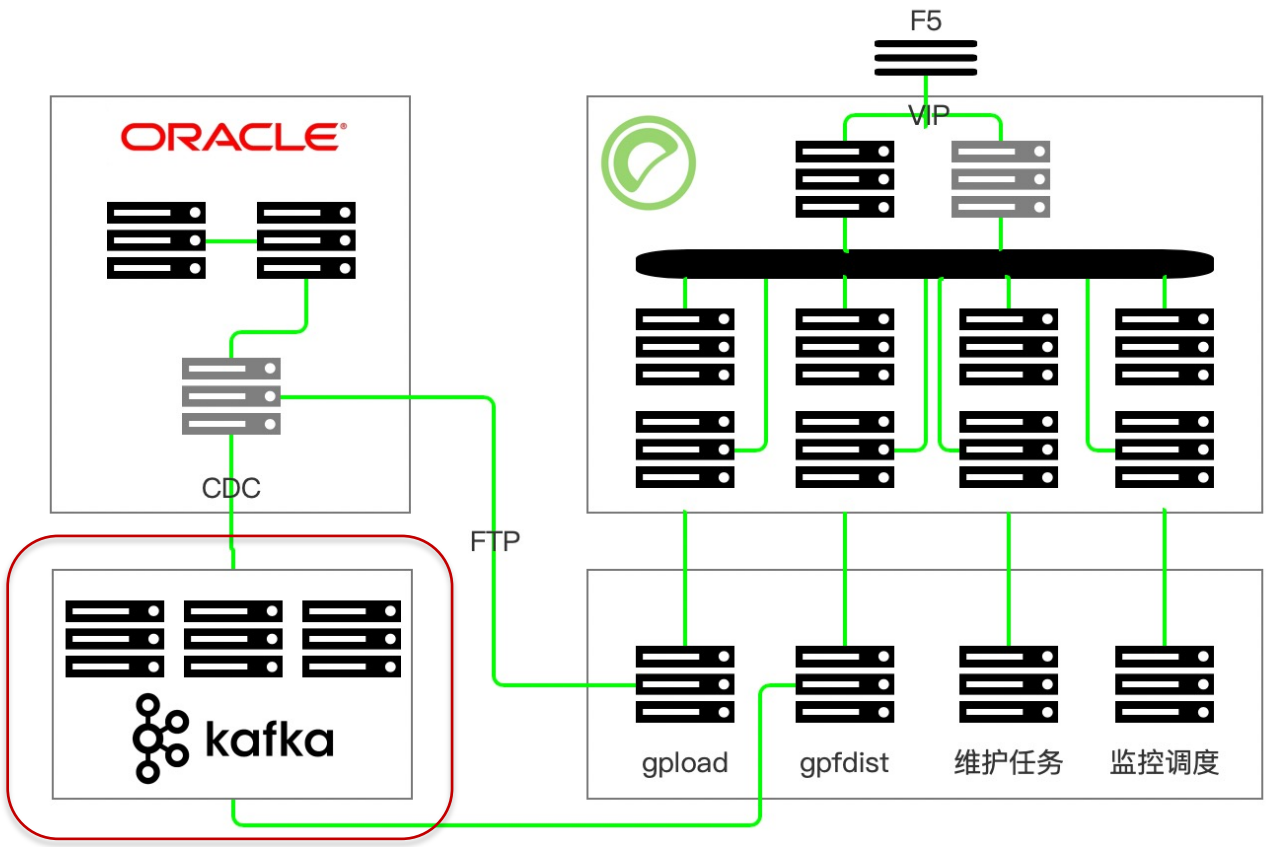
# 规划与部署 - GPDB详细规划

- 集群采用2+8的部署方式，2台Master采用keepalive做了高可用，提供虚拟IP对接F5对外提供服务（F5此处可省略），正式上线运行有2年多时间，未发生过一次切换；Master节点的可用性尚能接受；
- 网络层面万兆双网卡做了冗余，防止硬件故障；
- 8台Segment主机方便后期以4台机器为单位进行扩容，当前8台50TB的存储可满足备份双副本及备份需求，备份机制为选择性备2删1。



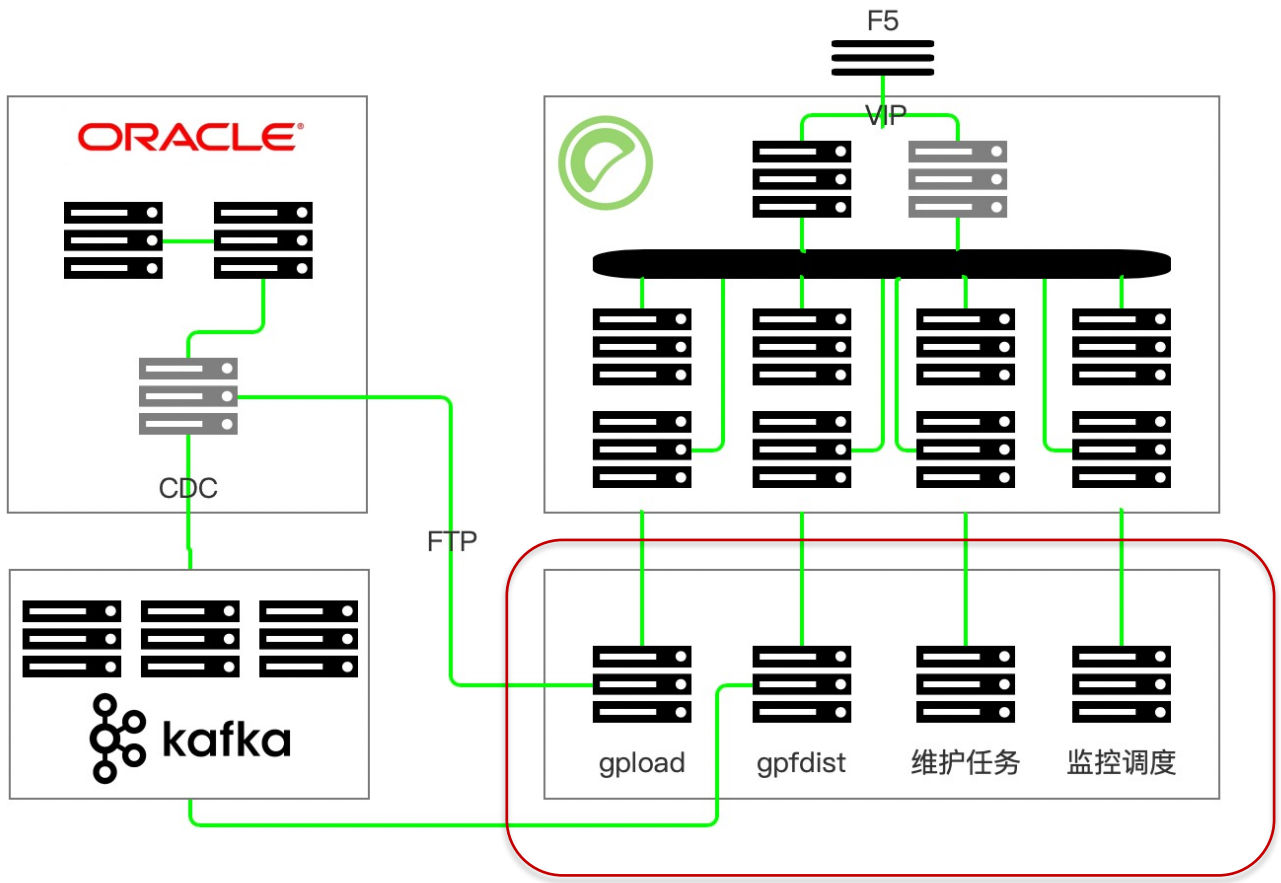
# 规划与部署 - Kafka规划

- Kafka规划比较简单，按照高可用的要求，选择了最少的3台机器组建集群；
- 机器磁盘为单盘挂载方式；没有组合成大盘；
- 磁盘冗余主要做了3个月日志冗余的准备；
- Kafka主要承接CDC软件传输过来的部分实时表数据，然后采用gpfdist尽量做到准实时入库。



# 规划与部署 - ETL服务器规划

- ETL服务器基本承担了跟数据有关的所有任务，包括：FTP数据入库、Kafka数据入库、日常维护工作、监控与调度；
- gpload服务器与维护任务服务器互为备份，gpfdist与监控调度服务器互为备份；高可用通过自开发程序实现；
- FTP数据文件量巨大，采用gpload入库，入库成功文件自动移动到NAS做备份，失败则保留在本地方便补录；
- Kafka数据通过自研程序调度gpfdist实现数据微批入库，支持更新操作。



GREENPLUM DATABASE

# 3. 客户选型及测试内容分享

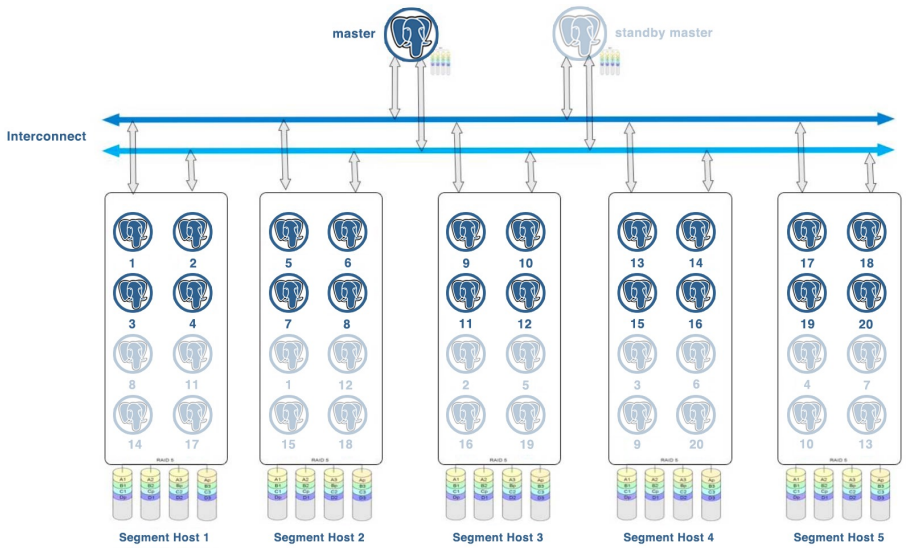
## 为什么选择Greenplum作为替代方案

GREENPLUM DATABASE

# 选型的考虑

- 考虑到数据量大且增长率高，需要选用分布式架构数据库，支持横向扩容；
- 基于当前的业务构建测试模型，测试结果需满足性能3倍于当前Oracle方案的要求；FTP入库及跑批需在工作日上班时间前完成；CDC入库近乎实时；
- 主要定位与分析型场景，因为业务偏向大规模跑批；但是需要支持300左右并发，同时提供OLTP处理能力；
- 迁移成本尽量低。

ORACLE®

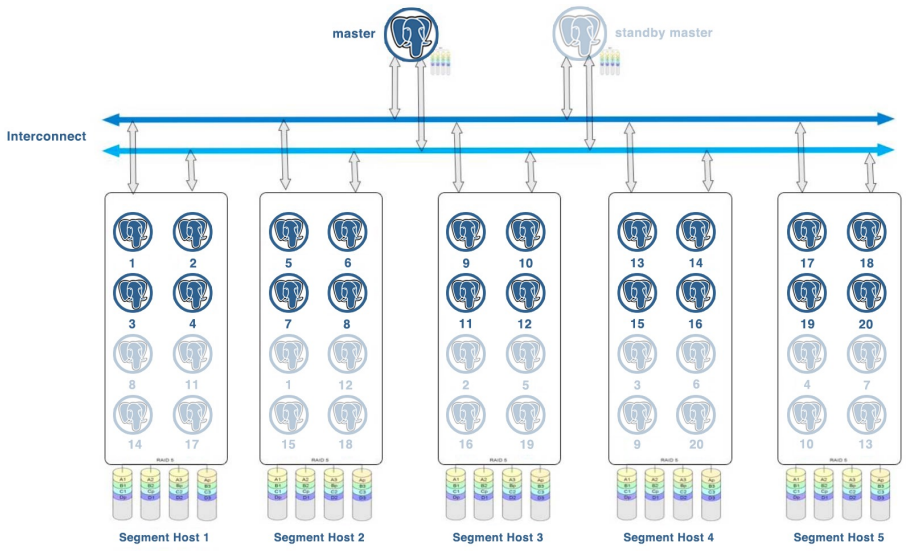


GREENPLUM DATABASE

# 测试内容 - 功能测试

- 存储过程迁移工作量，基本属于平行迁移，不复杂；很多Oracle中的窗口函数在GP中均已支持，例如：ROW_NUMBER() OVER(PARTITION BY xxx ORDER BY length(xxx) asc)；分区表迁移改动量少；
- 完整的ACID支持；标准的SQL查询，改动较少；
- 支持Kafka入库，支持文本文件入库；
- 提供完整的数据生命周期管理软件。

ORACLE®

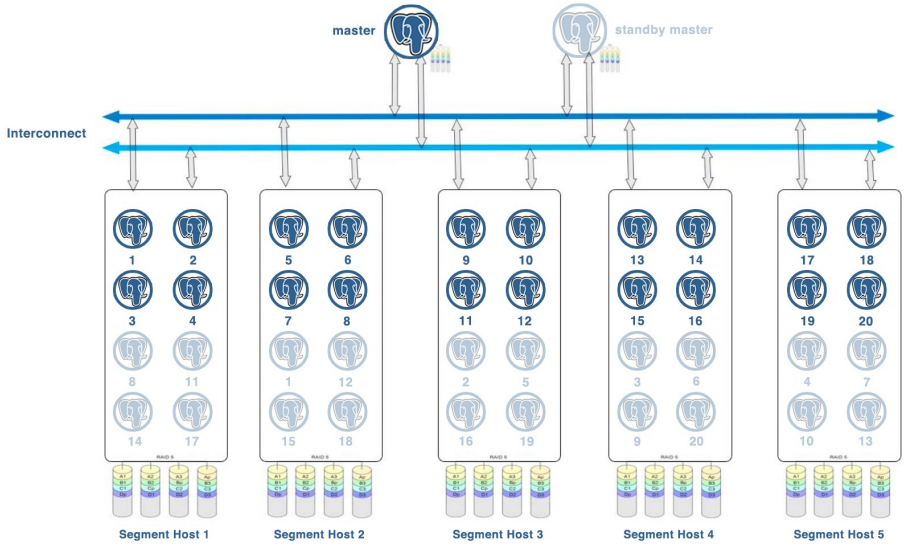


GREENPLUM DATABASE

# 测试内容 - 性能测试

- 存储过程跑批任务，由10小时缩短到2小时；
- FTP文本文件高速入库，gpload，如上；
- CDC数据入库，缩短到10分钟以内；
- JMeter并发压测，30个测试SQL并发压到300，平均响应时间在10s以内，大部分响应时间在3s左右，部分稍复杂SQL拖长平均时间；
- 数据迁移采用并行化方式，gpfdist入库效率高，瓶颈基本在Oracle及网络端。

ORACLE®



GREENPLUM DATABASE

# 4. 新老系统迁移及数据校验

## 项目落地的过程

# 迁移范围、原则及目标

## 数据迁移范围

现有生产环境Oracle数据库数据：101张数据表，筛选后总存储容量约需50TB。

## 数据迁移目标

将现有Oracle生产环境**存量数据**选择性迁入外地数据中心GP数据库中；

从Kafka和FTP分别接入增量数据；并校验数据一致性。

## 数据迁移原则

不改动现有业务平台Oracle数据库的配置和数据；

数据迁移在非业务时间进行，避免对生产环境业务及城市间现有业务传输造成太大影响。

# 迁移步骤

①应用和数据库可迁移性评估

②规划迁移到Greenplum的过程

③模式转换：数据类型映射，迁移表结构/视图

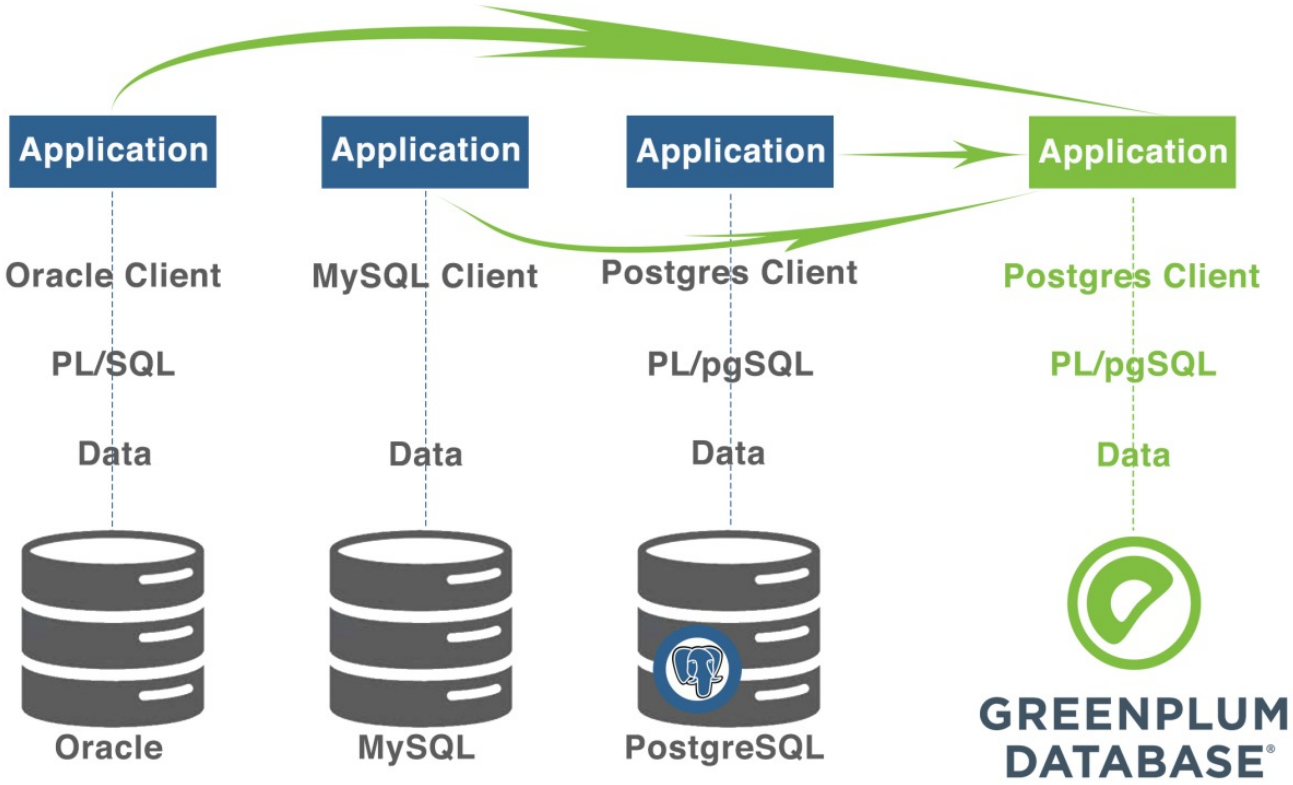
④函数/存储过程迁移（不支持触发器）

⑤应用程序SQL语句适配

⑥迁移数据到Greenplum，数据校验

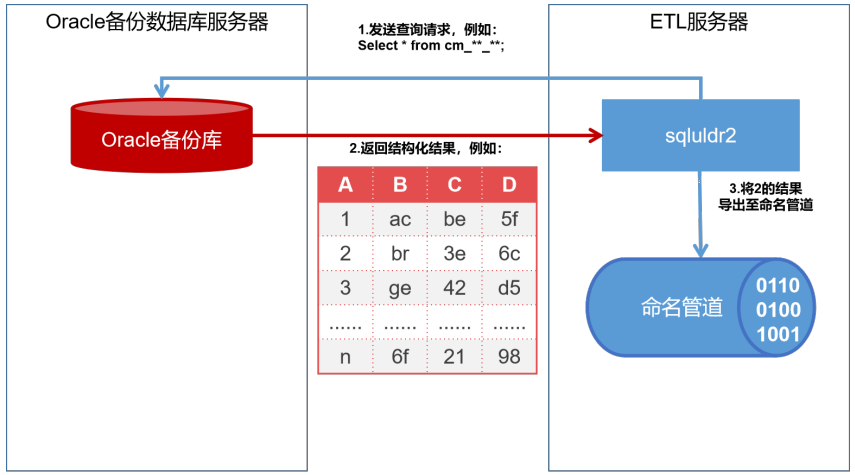
⑦应用测试

⑧生产切换



# 数据迁移逻辑 - 采用封装好的sqluldr2+gpfdist

- segment并行入库，性能高；
- TB以上数据量推荐；
- 自动化成都低；
- 命名管道损坏问题；
- 字符编码问题；
- 自行替换表结构。




| A | B | C | D |
|---|---|---|---|
| 1 | ac | be | 5f |
| 2 | br | 3e | 6c |
| 3 | ge | 42 | d5 |
| …… | …… | …… | …… |
| n | 6f | 21 | 98 |

**前提条件：**

做好以下准备工作：

- 本实例假设，Oracle源端为A，执行同步软件服务器为B，目标端Greenplum服务器为C
- 将sqluldr2软件包、oracle instantclient软件包、greenplum软件包这3个文件包一并上传到执行服务器B
  - 在服务器B上，将三个软件安装并配置环境变量（不需要启动任何程序，sqluldr2执行时会用到oracle的库文件，gpfdist这里采用完整安装包配置最简单）

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/instantclient_19_3/:/usr/local/greenplum-db/lib/
export PATH=$PATH:/iot/instantclient_19_3/
export NLS_LANG=AMERICAN_AMERICA.ZHS16GBK（迁移时出现乱码时需要调整该环境变量）
```

- A、B数据库环境、用户访问及查询权限均开放（Oracle仅需要查询权限）

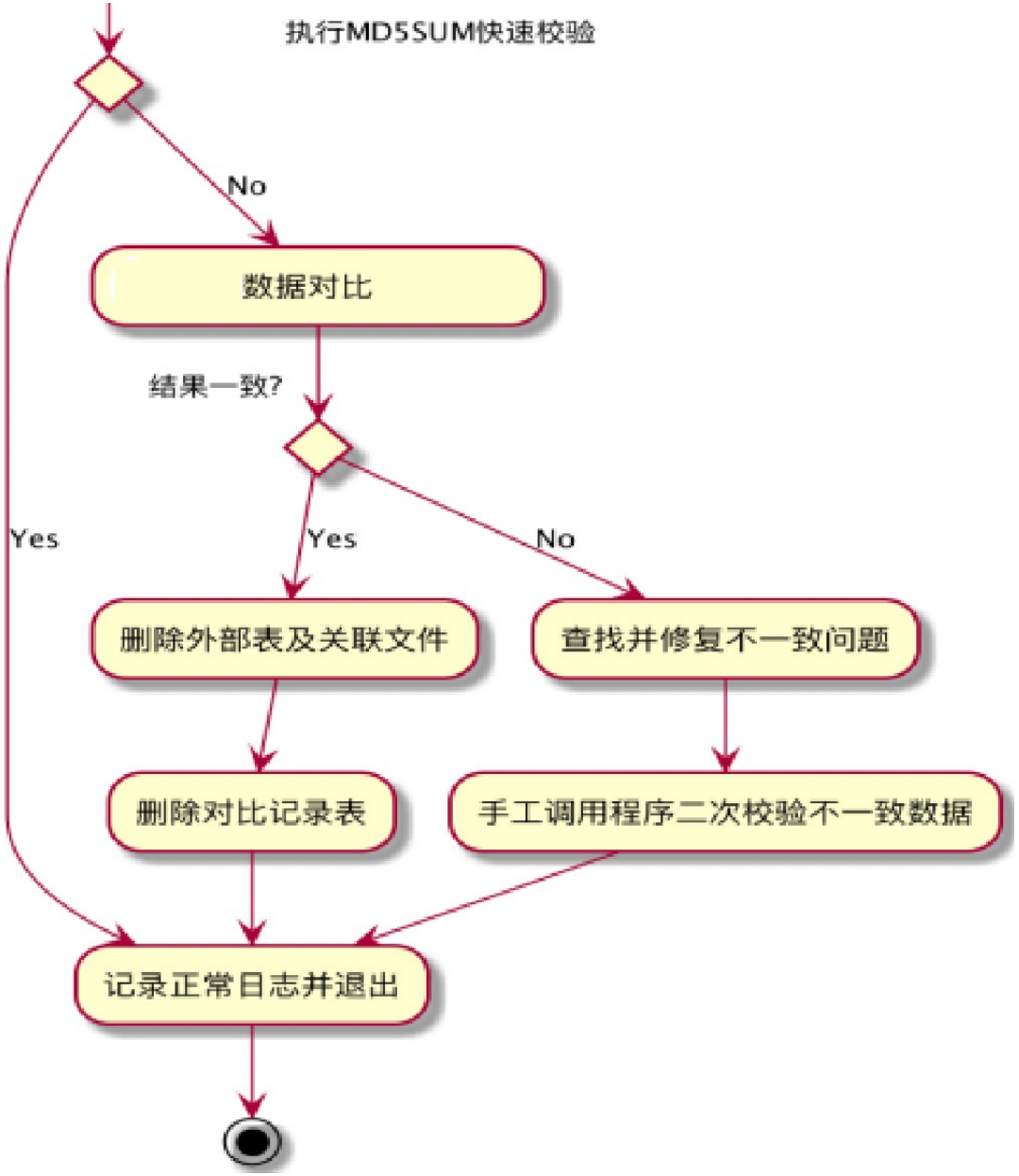
**执行同步：**

**创建外部表**

```
create external table EMPLOYEES_EXT (like EMPLOYEES) location('gpfdist://192.168.0.200:8080/EMPLOYEES.dat') format 'csv';
insert into EMPLOYEES select * from EMPLOYEES_EXT;
```

# 数据校验逻辑 - MD5+数据比对

- 首先进行MD5比对，如果一致记录结果；
- MD5不一致进行数据比对，从远端拉取数据到GP端，利用并行的优势将差异数据筛选出来并修复；
- 清理临时表和数据；
- FTP数据比对因为FTP有T+1的时间差，所以较好控制；
- Kafka数据需要选取合适的时间点，例如周末，通过暂停Oracle备份库的机制进行多次比对。



GREENPLUM DATABASE

# 5. 生产运维经验分享

## 一些应该尽量遵从的最佳实践

# 必须要提前明确的内容

- Greenplum是分布式架构，所以数据分布均匀是所有计算的前提，选择好分布键至关重要；
- 合理利用分区和数据分布，可以事半功倍；
- 清理数据，能选择Drop或Truncate操作，就别选择Delete操作；
- 数据压缩要做合理选择，否则性能差别很大；
- 要想执行计划更合理，及时收集统计信息；
- 合理利用索引，提升OLTP性能；
- 数据倾斜产生的木桶效应可能会拖慢集群的整体计算能力；
- 检测膨胀，系统表定期vacuum是必须要做的工作；
- group mirroring是gpinitsystem和gpaddmirrors的默认选项。

# 生产踩过的坑

- 发现RAID5正在重建磁盘时，可以将单个segment切换到它的镜像上；
- 当活动的主segment不能访问其镜像时，复制会停止并且主segment的状态会改为“Change Tracking”。主segment会把没有被复制到镜像的更改保存在一个系统表中，等到镜像重新在线时这些更改会被复制到镜像；根据这个现象一定要及时恢复，不然可能撑爆你的磁盘；
- 恢复故障segment采用gprecoverseg，恢复后记得角色复原rebalance操作；
- 有必要在segment主机上保留足够的内存和CPU资源，以允许承担了主segment的镜像实例能在故障时提供对应的活动负载。
- 生产上，一定要做好监控工作，无论是采用GPCC还是开源方案，在出现问题时，监控日志好似最好的帮手。

GREENPLUM DATABASE®

微信技术讨论群
微信搜索添加“gp_assistant”
加入技术讨论

微信公众号
搜索添加“Greenplum中文社区”
技术干货、行业热点、活动预告

# Thank You


Confidential | ©2021 VMware, Inc.