GREENPLUM DATABASE® 云社区

六节课

快速上手Greenplum

第六节课

异构数据库的迁移

线上直播 | 1月9日 14：00 - 15：00

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

GREENPLUM DATABASE®

云社区

Greenplum 中文社区 公众号

- 技术研讨会PPT
- 活动预告
- 行业热点
- 技术干货
- Greenplum动态

GREENPLUM DATABASE®

云社区

Greenplum中文社区网站

https://cn.greenplum.org

博客 · 资料 · 文档 · 项目 · 活动

GREENPLUM DATABASE®

云社区

全新的技术问答论坛

有问题？askGP！

https://cn.greenplum.org/askgp

原厂专家值班 · 优质内容沉淀

六节课快速上手Greenplum

第六课

Greenplum 迁移指南

辉鸿泛在CTO：阿福

目录

GREENPLUM DATABASE®

云社区

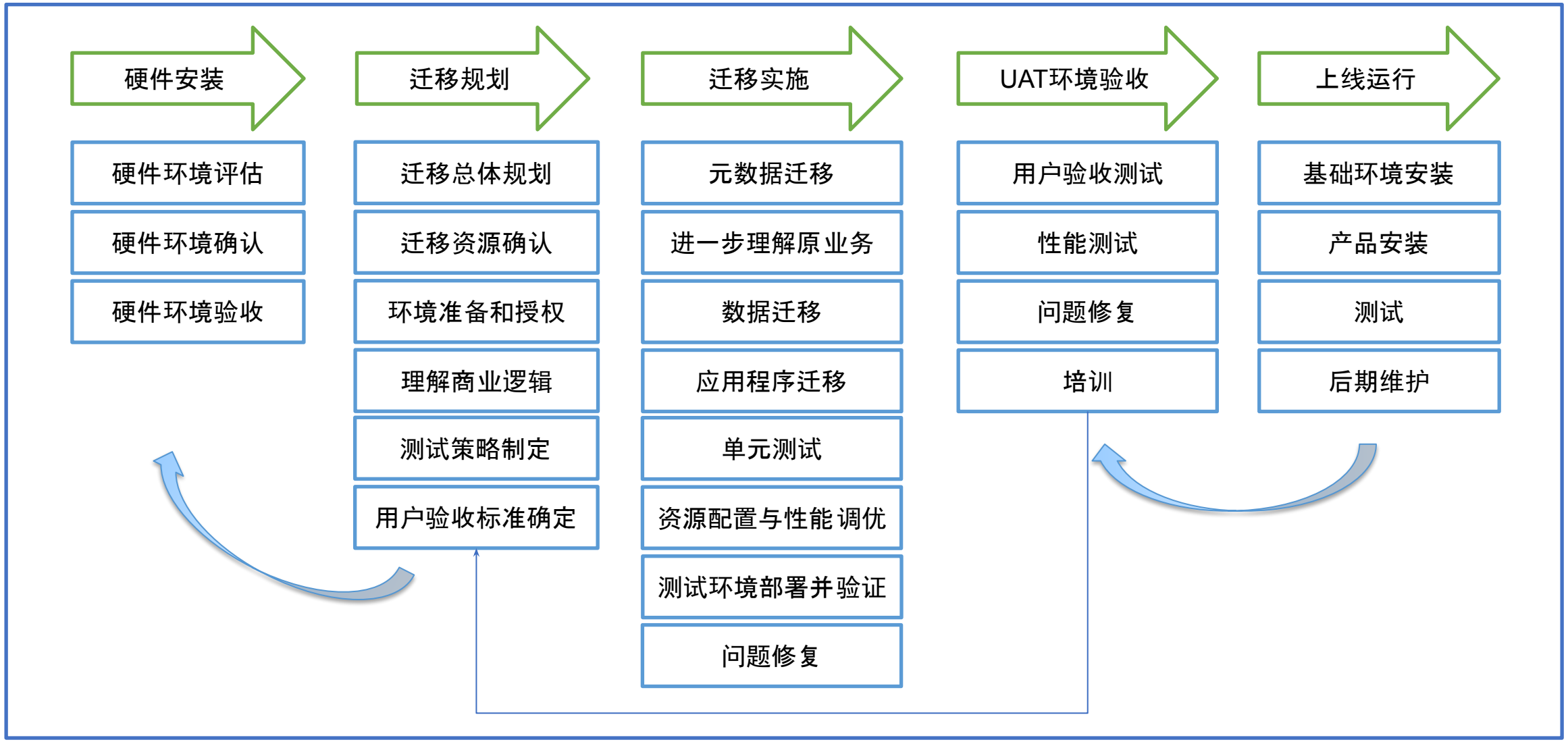
GPDB数据迁移方法论

为什么要进行数据迁移

数据迁移的目的是为了给数据找一个更合适的归宿，让其满足当前及未来某段时间内业务场景的使用要求。使数据更安全、更可靠、更有效的为客户服务。

对于数据库而言，通常为了解决当前数据库遇到的瓶颈，考虑到成本、性能、可用性、未来发展等多个方面因素，进行合理的数据迁移，以求通过新技术的引进，满足未来3-5年时间内业务持续性的要求。

迁移整体流程


硬件安装
迁移规划
迁移实施
UAT环境验收
上线运行
硬件环境评估
硬件环境确认
硬件环境验收
迁移总体规划
迁移资源确认
环境准备和授权
理解商业逻辑
测试策略制定
用户验收标准确定
元数据迁移
进一步理解原业务
数据迁移
应用程序迁移
单元测试
资源配置与性能调优
测试环境部署并验证
问题修复
用户验收测试
性能测试
问题修复
培训
基础环境安装
产品安装
测试
后期维护

GREENPLUM DATABASE®

云社区

GPDB数据迁移工具

迁移工具

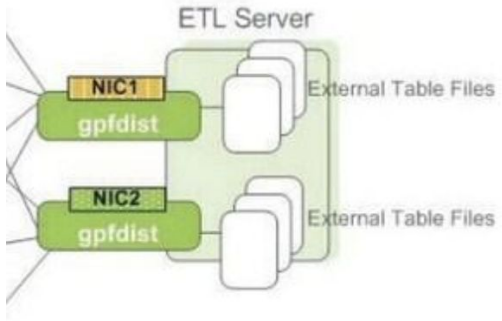
根据第一部分的讲解，大家也理解数据迁移是一个复杂的工作，要求各方配合，多种技术结合使用。

目前市面上还没有任何一款工具可以灵活高性能的完成到Greenplum的异构数据迁移，并且在迁移过程中需要大量的人工干预，所以通常情况下我们都需要采用多种技术配合来完成这一工作。

通常我们常用的迁移工具如右图。

AWS Schema ConversionTool
https://aws.amazon.com/cn/dms/?nc=sn&loc=1

ora2pg
http://ora2pg.darold.net



gpfdist

迁移工具 - ora2pg

ora2pg 数据库迁移工具

☆ 收藏 72　　评论 1　　分享

ora2pg

授权协议：GPLv3	开发厂商：无
开发语言：Perl	地区　　：不详
操作系统：Linux	提 交 者 ：不详
软件类型：开源软件	适用人群：未知
所属分类：数据库相关、 数据库管理工具	收录时间：2008-09-20

- Ora2Pg安装配置
- ora2pg安装及卸载
- 使用Ora2pg需要注意的问题
- ORACLE 迁移到 PG 之 ora2pg
- Oracle迁移至PostgreSQL工具之Ora2Pg
- adsas数据库去O记

ora2pg是一款功能丰富的工具，用于将oracle/mysql数据库迁移到PostgreSQL。由于Greenplum与PostgreSQL的语法近乎一致性，所以同样也适用于Greenplum。通常情况下，我使用他来做简单的元数据转换及迁移分析。

✔ 相关详细信息、源码及安装使用教程，参考开源中国:https://www.oschina.net/p/ora2pg?hmsr=aladdin1e1

迁移工具 - ora2pg

温馨提示：

离网环境编译过程会比较痛苦，通常需要自行安装以下软件依赖包

- ExtUtils-MakeMaker-7.38
- Test-Simple-1.302168
- gdbm-devel-1.10-8.el7
- Libdb-devel-5.3.21-25
- Pyparsing-1.5.6-9.el7
- System tap-set-devel-4.0-9
- Perl-ExtUtils-ParseXS-3.18-3 —nodeps
- Perl-ExtUtils-install-1.58 —nodeps
- Perl-Test-harness-3.28-3
- Perl-extutils-manifest-1.61-244
- Perl-ExtUtils-MakeMaker-6.68-3
- DBI-1.642

迁移工具 - aws schema conversion tool

AWS Schema Conversion Tool（简称AWS SCT）是为了方便用户数据上云，由AWS提供的图形化自动转换工具，可以在本地部署安装，安装部署过程简单，能生成详细的分析报告，并且支持多种数据平台的语法转换。根据我们在用户环境的验证，大概可以完成将近70%的语法自动转化工作。关于AWS SCT安装和使用，可参考如下链接：

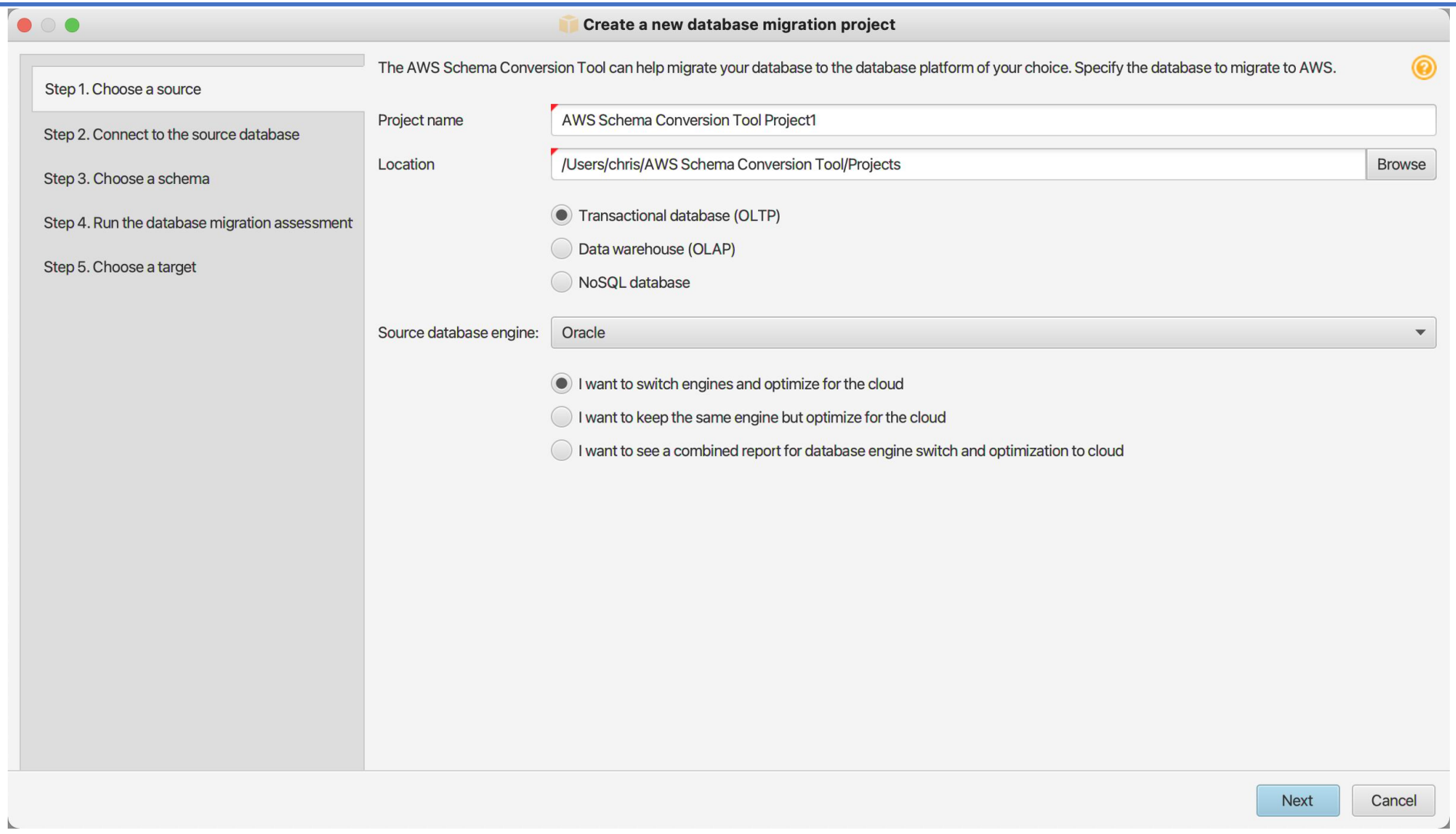
- ✔ https://docs.aws.amazon.com/zh_cn/SchemaConversionTool/latest/userguide/Welcome.html。

配置好Oracle和PostgreSQL的URL连接串后，SCT会自动检索并进行分析，生成评估转换报告。

下载连接：

https://aws.amazon.com/cn/dms/schema-conversion-tool/?nc=sn&loc=2

迁移工具 - aws schema conversion tool



迁移工具 - sqluldr2

sqluldr2是一款Oracle数据快速导出工具，包含32、64位程序，sqluldr2在大数据量导出方面速度超快，能导出亿级数据为excel文件，另外它的导入速度也是非常快速，功能是将数据以TXT/CSV等格式导出。

他支持Windows和Linux平台，通常用来配合gpfdist做大批量存量数据迁移。

也可以用来构建自己的数据迁移工具。

✔ 具体的使用方法可以参考博客

:https://blog.csdn.net/cheng_feng_xiao_zhan/article/details/83620593

迁移工具 - sqluldr2

简要配置步骤：

1.安装Oracle客户端

下载instantclient_19_3.zip

unzip instantclient_19_3.zip

mv instantclient_19_3 /usr/local/instantclient_19_3

export LD_LIBRARY_PATH=/usr/local/instantclient_19_3/

2.测试sqluldr2

下载sqluldr2

./sqluldr2_linux64_10204.bin

3.如果有结果返回，证明环境正常，尝试连接数据库卸载数据

迁移工具 - sqldata

SQLines Data 是一款开源(Apache License 2.0)，可伸缩，并行高性能的data传输、schema转换工具，可以用作数据库迁移和ETL处理。

支持数据库:

- Oracle and Oracle Exadata
- Microsoft SQL Server and Microsoft Azure SQL
- MySQL
- MariaDB
- PostgreSQL
- Netezza
- Greenplum
- IBM DB2 LUW, iSeries (AS/400) and zSeries (S/390)
- Sybase Adaptive Server Enterprise, Sybase SQL Anywhere, Sybase IQ and Sybase Advantage
- Informix
- Teradata
- Vertica
- SAP HANA

该工具特别适合数据量在TB以下级别的小型数据库迁移，速度快、省时省力。

✔ 程序获取方式见：https://github.com/luzhihua407/SQLines-Data

迁移工具 - sqldata

温馨提示：

1.简单使用示例：

./sqldata sqldata.cfg -sd=oracle,tigger/tigger@192.168.0.12:1521/orcl
-td=pg,tigger/tigger@192.168.0.13:5432/postgres -qf=sqlines_qf.txt -ss=5
-topt=truncate &

2.bug

- 目标端td的数据库名字必须和用户名一致，如果不一致，会以用户名为准，而不以数据库名称为准
- 单表迁移数据量超过21亿，结果报告展示会显示负数

GREENPLUM DATABASE®

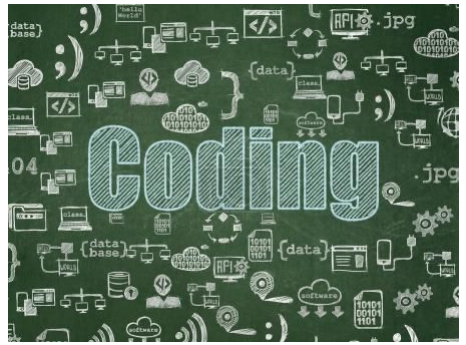
云社区

实现自己的数据迁移程序

数据迁移全周期功能

Step1 : Get the information about source schema.
Step2 : Generate DDL for Greenplum schema from Oracle schema
Step3: Generate CSV data dump for oracle tables.
Step4: Load the database using GPFDist
step5: Validate the data
1 - Test Oracle Database Connectivity
2 - Oracle Database Information Report
3 - Oracle Table Rows Count Report
4 - Oracle Table Checksum Report
5 - Generate Greenplum Schema Table DDL corresponding to Oracle Schema
6 - Generate Greenplum External Table DDL corresponding to Oracle Schema
7 - Generate Load data insert table scripts to insert data into Greenplum table
8 - Generate Select count DML scripts to count no of rows in greenplum internal and external tables
9 - Export Oracle Table Data in CSV Format consumed by Greenplum External Table
10 - **Export very large partitioned tables data in parallel and store in different location
11 - **Generate External table DDL of large partitioned tables

21 - Test Greenplum Database Connectivity
22 - Create table in Greenplum using DDL generated from option 5
23 - Create external table in Greenplum using DDL generated from option 6 or option 10
24 - Load Data in Greenplum
25 - Generate table counts DML script
26 - Create Checksum Report of Migrated data in Greenplum
27 - Compare Oracle and Greenplum Checksum Report



GREENPLUM DATABASE®

云社区

Oracle到GPDB的数据迁移

为什么要从Oracle迁移到GPDB

客户通常从别的平台迁移到Greenplum的原因有：

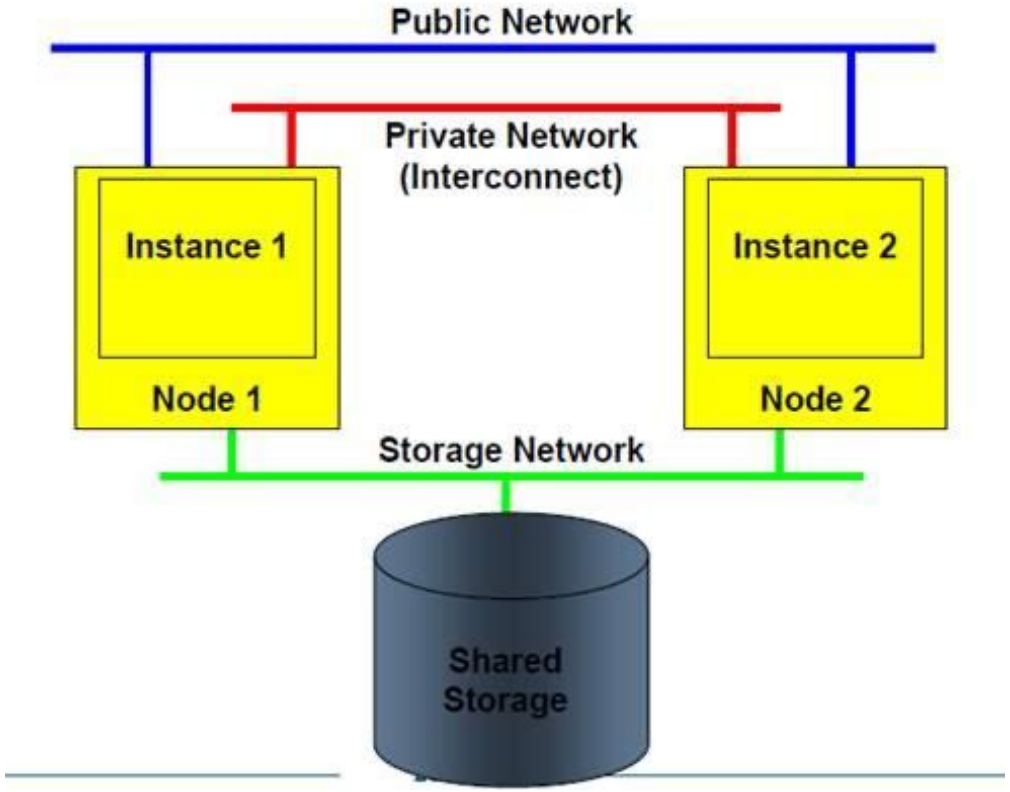
1.成本：Greenplum相对于Teradata、Oracle Exadata等一体机设备，不需要购买专有硬件设备，有明显的成本优势；

2.性能：Greenplum相比传统关系型数据库有明显的性能提升，多个用户从Oracle迁移到Greenplum后，性能有几十倍的提升。

3.易用性：Greenplum相比Hadoop平台，SQL表达能力更为突出，应用改造成本要小很多。

针对Oracle而言，Oracle并不是专门为分析型场景设计，其体系架构主要是应对数据经常变动的OLTP高并发、低时延场景。

针对分析型应用，一般在Oracle上运行数小时的分析型应用，在Greenplum上只用数分钟或者秒级返回



迁移场景

大部分场景都可以直接迁移到Greenplum，但也有部分场景（如高并发事务型场景）不太适合迁移到目前的Greenplum版本。具体的迁移建议如下：

Oracle中的应用场景	Oracle中的响应时间	迁移到Greenplum建议
分析型场景	1秒以上	此类应用完全可以迁移至Greenplum，迁移后性能会有较明显提升
并发小查询场景	1秒以内	并发小查询场景包括小表全表扫描和大表索引扫描场景，迁移至Greenplum性能在同一量级，但因为数据节点交互，延迟会略有增加
并发数据加载场景	1秒以内	可以迁移至Greenplum，需要将逐笔插入操作改为微批量插入，由于Greenplum MPP架构优势，加载性能会有较明显提升
低并发事务型场景	1秒以内	可以迁移至Greenplum，需要做适当业务改造，将逐笔操作改为微批量操作
高并发事务型场景	1秒以内	不建议迁移到Greenplum，由于数据跨节点的网络交互和锁的问题，会导致性能有较大的损失，甚至无法满足业务要求。请关注Greenplum的研发进展和新版本特性，Greenplum社区正在不断增强高并发事务型特性。

元数据迁移

- Oracle到Greenplum没有现成的迁移工具，可以借助部分自动化转换工具先将Oracle语法转换为PostgreSQL语法，再通过脚本替换，最终转换成Greenplum语法。
- Oracle到PostgreSQL常用的迁移工具有ora2pg以及AWS Schema ConversionTool。ora2pg为命令行工具，只能从Oracle转换到PostgreSQL；而AWS Schema Conversion Tool（简称AWSSCT）是为了方便用户数据上云，由AWS提供的图形化自动转换工具，可以在本地部署安装，安装部署过程简单，能生成详细的分析报告，并且支持多种数据平台的语法转换。
- 根据我们在用户环境的验证，大概可以完成将近70%的语法自动转化工作。
- 存储过程的迁移属于难点。
- 关于AWS SCT安装和使用，可参考如下链接：https://docs.aws.amazon.com/zh_cn/SchemaConversionTool/latest/userguide/Welcome.html。

元数据迁移

SCT会自动进行类型转换，如果你想了解Oracle和Greenplum中不同数据类型的映射关系如右表

Oracle	Greenplum	说明
VARCHAR2(n)	VARCHAR(n)	在Oracle中n代表字节数，在Greenplum中n代表字符数
CHAR(n)	CHAR(n)	同上
NUMBER(n,m)	NUMERIC(n,m)	number可以转换成numeric，但真实业务中数值类型可以用smallint、int或bigint等代替，性能会有较大提升
NUMBER(4)	SMALLINT	
NUMBER(9)	INT	
NUMBER(18)	BIGINT	
NUMBER(n)	NUMERIC(n)	如果n>19，则可以转换成numeric类型
DATE	TIMESTAMP(0)	Oracle和Greenplum都有日期类型，但Oracle的日期类型会同时保存日期和时间，而Greenplum只保存日期
TIMESTAMP WITH LOCAL TIME ZONE	TIMESTAMPTZ	
CLOB	TEXT	PostgreSQL中TEXT类型不能超过1GB
BLOBRAW(n)	BYTEA	在Oracle中BLOB用于存放非结构化的二进制数据类型，BLOB最大可以存储128TB，而PostgreSQL中BYTEA类型最大可以存储1GB，如果有更大的存储要求，可以使用Large Object类型

数据迁移

数据迁移包括全量和增量数据迁移。进行全量迁移时，可以用sqluldr2工具先把数据以csv格式导出，然后再通过gpfdist加载到Greenplum中。

增量迁移一般借助golden gate等cdc软件尽量做到数据的实时捕获，再通过GPFDIST加载到Greenplum中。曾经有用户以250ms的间隔通过GPFDIST实时加载数据到Greenplum中，在8个计算节点的集群上速度可达到200万/s。

将数据导成csv文件，命令如下：

```
        export NLS_LANG="SIMPLIFIED CHINESE"
_CHINA.ZHS16GBK
        sqluldr2  user='username'/'password'
@tnsname  query="select  /*  parallel(2)  *
/  *  from
        PICCPROD.T_CONTRACT_MASTER" text=CSV
file="/data/ods/test.dat" log=/tmp/sqluldr2.
log
```

创建外部表进行数据导入，命令如下：

```
        create external table ext.t_contract
_master_ext (like t_contract_master)
        LOCATION('gpfdist://10.111.224.1:999
9/T_CONTRACT_MASTER.dat') FORMAT 'CSV' (HEAD
ER
        delimiter as ', ') ENCODING 'GB18030
' LOG ERRORS SEGMENT REJECT LIMIT 2 rows;
```

数据校验

数据校验通常有以下集中方式：

- count值校验
- 部分字段汇总校验
- md5校验

通常情况下，对校验方式的选择还是根据客户的要求来做，前两种的效率较高，md5校验的成本可能更高、但是准确度也高。

```
        with oracle_checksum as (select id,
md5(textin(record_out(foo2))) as row_md5 fro
m
    ora_hello as foo2),
        gp_checksum as (select id, md5(texti
n(record_out(foo3))) as row_md5 from gp_hell
o
    as foo3),
        compare_result as (select oracle_che
cksum.id as ora_id , oracle_checksum.row_md5
as
    ora_md5, gp_checksum.id as gp_id, gp_che
cksum.row_md5 as gp_md5 from oracle_checksum
full outer
    join gp_checksum on oracle_checksum.row_
md5=gp_checksum.row_md5)
        select * from compare_result where o
ra_md5 is null or gp_md5 is null;
```

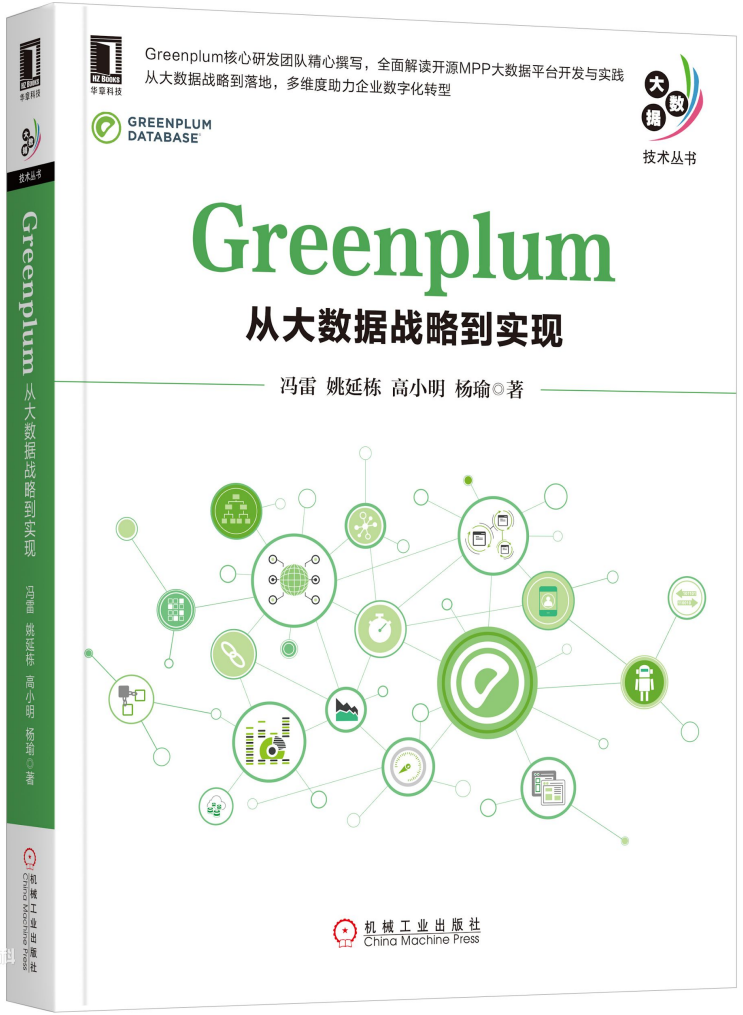
参考书籍

本节主要介绍了Oracle到Greenplum的迁移思路，由于篇幅有限，内容仅做了简单的涉猎。如果大家有对应的需求，可以购买右侧的书籍。

书中关于迁移的部分内容有：

- 14章 Greenplum同构集群迁移
- 15章 Oracle到Greenplum的迁移
- 16章 Teradata到Greenplum的迁移

书中分享了原厂工程师大量的实战经验，值得一读。



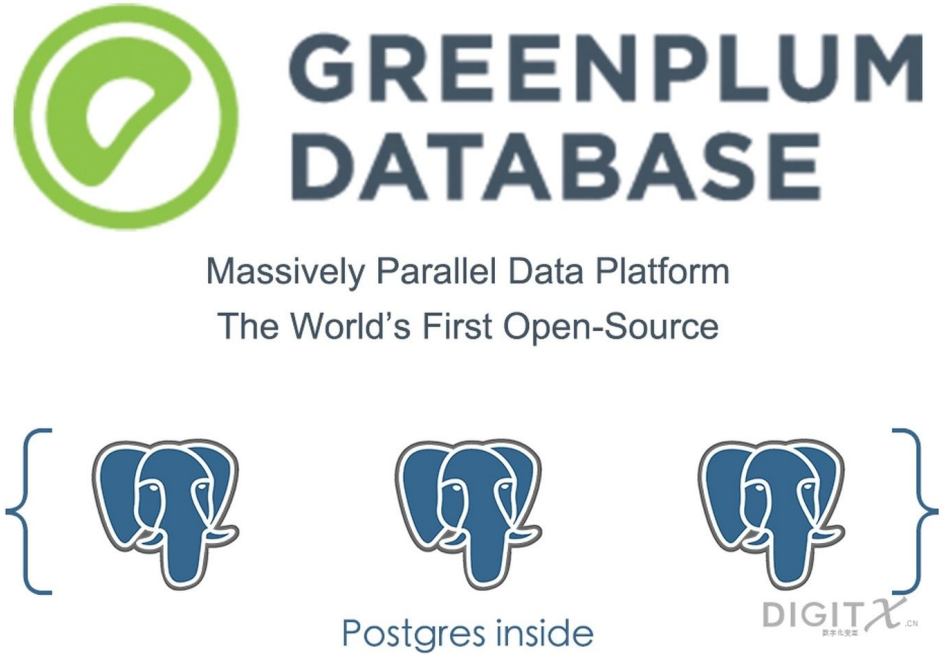
GREENPLUM DATABASE®

云社区

PostgreSQL到GPDB的数据迁移

一种平滑的解决方案

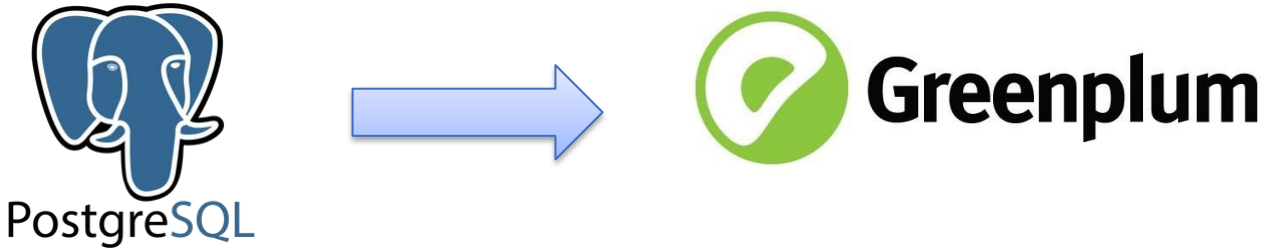
Greenplum与PostgreSQL无论在语法还是使用方式上，都基本相似。所以从PostgreSQL迁移到Greenplum，通常是TP、AP拆分的一种平滑解决方案。由于均属于开源软件，既能节约成本，又能很好的相互结合。



元数据迁移

元数据迁移直接从过pg_dump导出后修改导入即可，通常只需要以下三步

- pg_dump -s schema.sql sourcedb
- 手工介入，修改脚本中对应的分布建、分区等语法、优化存储过程
- psql -f schema.sql -d targetdb



数据迁移

数据迁移可以选用前面提到的sqldata工具，也可以自己编写全量增量迁移工具，通常情况下，自己编写工具会采用copy + gpfdist的组合，以最大限度的发挥两个数据库的特点。

将数据导成csv文件，命令如下：

```
$ COPY user(name,password) TO
'/tmp/data/test.csv' WITH csv;
```

创建外部表进行数据导入，命令如下：

```
        create external table ext.t_contract
_master_ext (like t_contract_master)
        LOCATION('gpfdist://10.111.224.1:999
9/T_CONTRACT_MASTER.dat') FORMAT 'CSV' (HEAD
ER
        delimiter as ', ') ENCODING 'GB18030
' LOG ERRORS SEGMENT REJECT LIMIT 2 rows;
```

数据校验

数据校验通常有以下集中方式：

- count值校验
- 部分字段汇总校验
- md5校验

通常情况下，对校验方式的选择还是根据客户的要求来做，前两种的效率较高，md5校验的成本可能更高、但是准确度也高。

```
        with oracle_checksum as (select id,
md5(textin(record_out(foo2))) as row_md5 fro
m
    ora_hello as foo2),
        gp_checksum as (select id, md5(texti
n(record_out(foo3))) as row_md5 from gp_hell
o
    as foo3),
        compare_result as (select oracle_che
cksum.id as ora_id , oracle_checksum.row_md5
as
    ora_md5, gp_checksum.id as gp_id, gp_che
cksum.row_md5 as gp_md5 from oracle_checksum
full outer
    join gp_checksum on oracle_checksum.row_
md5=gp_checksum.row_md5)
        select * from compare_result where o
ra_md5 is null or gp_md5 is null;
```

感谢观看

Greenplum中文社区公众号

Greenplum微信技术讨论群

GREENPLUM DATABASE®

云+社区