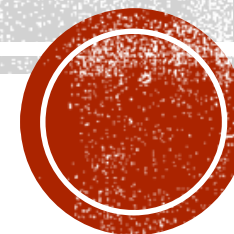


深入浅出 XTTTS 技巧



XTTS 是什么？



XTTS

- Transportable Tablespaces (TTS)
- Cross Platform Transportable Tablespaces (XTTS)

MOS 1389592.1



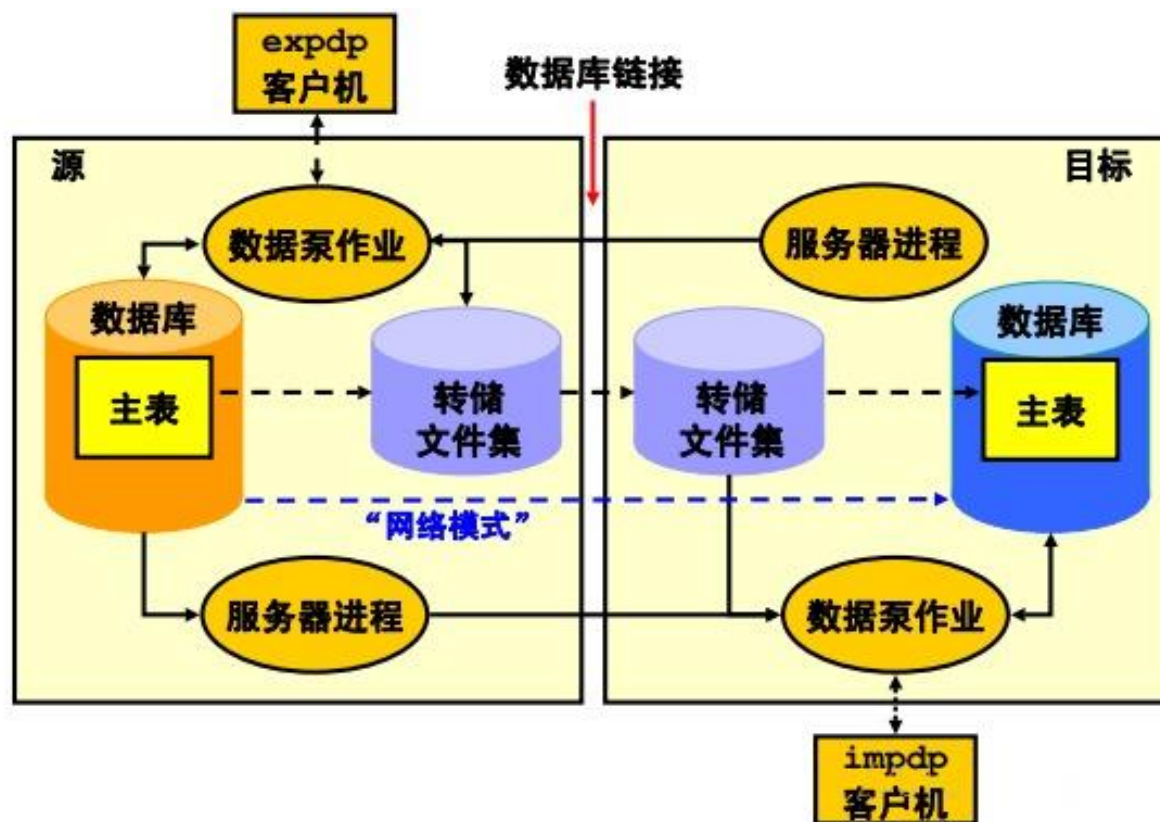
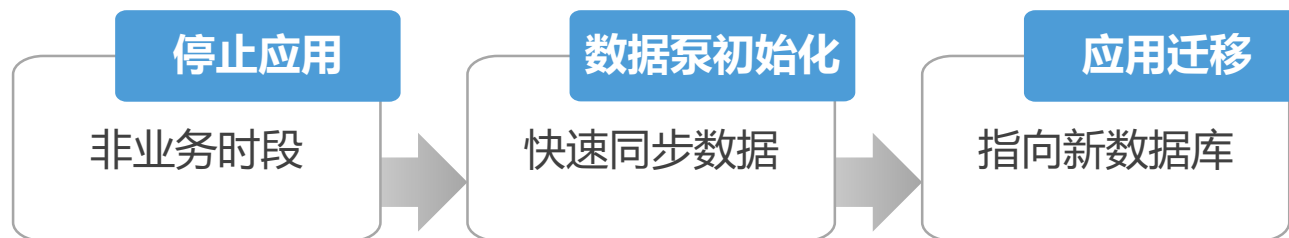
- 但是对于传统的TTS来说，数据量越大，需要的停机时间越长。因此，Oracle提供了一个加强版的XTTS，XTTS可以提供跨平台的增量备份，两者结合可以大大缩减迁移时所需要的停机时间。



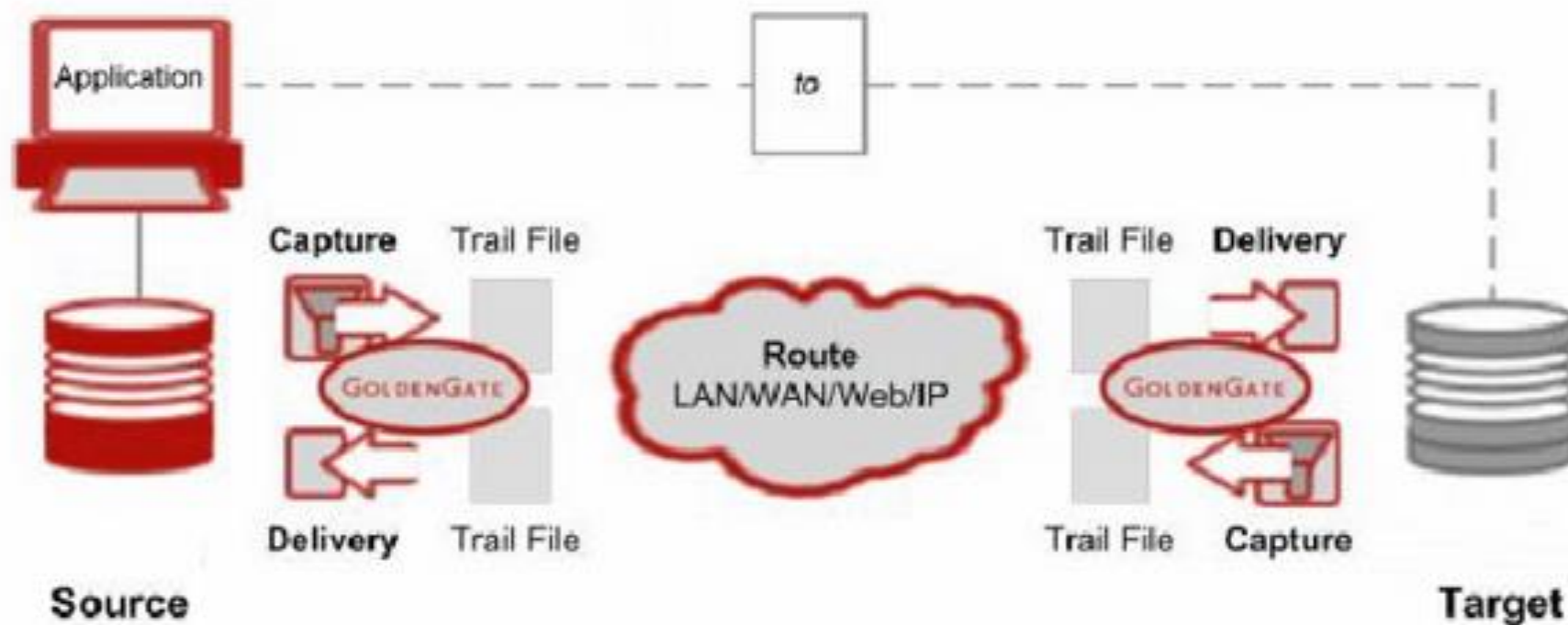
适用场景？



数据泵



GOLDENGATE



XTTTS



最短停机时间 最少数据丢失

	跨版本	跨平台	停机时间	场景
数据泵	支持	支持	48小时	数据量较小，或者停机时间没有要求。
GoldenGate	支持	支持	1小时	没有特殊类型的对象，基本没有DDL操作，对rowid变化没有要求。
XTTS	支持	支持	3-4小时	有一定停机时间，元数据数量可控的系统。



XTTS的基本操作步骤？



TTS的基本步骤

- A、将源端数据库表空间设置为READ ONLY模式。
- B、传输数据文件到目标系统。
- C、转换数据文件为目标系统的字节序。
- D、在源端导出元数据，并在目标端导入。
- E、将目标端的数据库表空间设置为READ WRITE。



XTTS的基本步骤

- A、将源端数据文件传输到目标系统。
- B、转换数据文件为目标系统的字节序。
- C、在源端创建增量备份，并传输到目标端。
- D、在目标端恢复增量备份。
- E、重复多次操作C和D步骤。
- F、将源端数据库表空间设置为READ ONLY模式。
- G、最后一次执行C和D步骤。
- H、在源端导出元数据，并在目标端导入。
- I、将目标端的数据库表空间设置为READ WRITE。



XTTS的参数设置

platformid=17	源端平台号,select * from v\$transportable_platform order by platform_id 语句查询得到
srcdir=SOURCEDIR	使用数据泵导入原数据,network_link方式
dstdir=DESTDIR	使用数据泵导入原数据,network_link方式
srclink=TTSLINK	perl xttdriver.pl -e命令生成impdp迁移脚本使用
dfcopydir=/export/home/oracle/dfcopydir	源端 datafile convert 保存路径
backupformat=/export/home/oracle/backup	源端增量备份保存路径
stageondest=/home/oracle/backup	目标端备份文件存放路径
backupondest=/home/oracle/backup	目标端增量备份集存放路径



XTTS 案例 - 参数设置

storageondest=+DATADG/orcladb11g	目标端恢复到数据库的数据文件保存位置，如果是diskgroup则需要注意还需要对应的数据库名，只在RMAN backup中使用
cnvinst_home=/u01/oracle/app/11.2.4/db	新环境\$ORACLE_HOME
cnvinst_sid=orcladb11g	新环境ORACLE_SID
asm_home=/home/grid/app/11.2.0/grid	如果是直接恢复到diskgroup，则需要指定对应的grid_home和ASM实例
asm_sid=+ASM	如果是直接恢复到diskgroup，则需要指定对应的grid_home和ASM实例
parallel=3	设置并行，建议设置为6或者8，根据CPU情况设置
rollparallel=2	roll forward 并行,如果数据库大的话建议设置为8
getfileparallel=4	getfile 并行度，建议设置为6或者8，根据CPU情况设置
desttmpdir=/home/oracle/xtts	设置目标端tmpdir目录,建议指向xtts目录)



XTTS 案例 - 准备

- 源端运行perl脚本，操作命令

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl -S
```

该操作将生成xttnewdatafiles.txt、getfile.sql两个文件。



XTTS 案例 - 数据文件拷贝

- 目标端执行命令：

```
scp oracle@source:/home/oracle/xtt/xttnewdatafiles.txt /home/oracle/xtt  
scp oracle@source:/home/oracle/xtt/getfile.sql /home/oracle/xtt
```

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl -G
```



XTTS 案例 - 进行第1次增量备份

```
$ORACLE_HOME/perl/bin /perl xttdriver.pl -i
```

- 该命令将对xtt.properties参数文件中指定的表空间，使用进行一个增量备份，同时会生成tsbkupmap.txt、incrbackups.txt、xttplan.txt三个文件。
- 备份的数据是从做xttdriver.pl -S时在xttplan.txt文件中记录的SCN开始的。备份完成后需将这3个文件连同增量备份集一起传输到目标端。



XTTTS 案例 - 进行第1次增量恢复

```
scp oracle@source:/home/oracle/xtt/xttplan.txt /home/oracle/xtt
```

```
scp oracle@source:/home/oracle/xtt/tsbkupmap.txt /home/oracle/xtt
```

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl -r
```

- 但是如果一套库上有多个实例的话，在执行该步骤之前，需要对环境变量进行确认，如检查当前ORACLE_SID是否是需要执行的SID，否则可能会恢复到其他实例上。（并非真实的恢复，因为其他实例跟这个备份集没有任何关系，但恢复的过程会在其他实例上进行一遍，如关闭/启动数据库，包括增量恢复的日志都会在另一个数据库上显示。）如果发生了这种事情，不用紧张，调整好环境变量，再执行一次perl xttdriver.pl -r即可。误操作的实例不受影响。



XTTS 案例 - 进行 SCN 推进

```
$ORACLE_HOME/perl/bin/ perl xttdriver.pl -s
```

- 该命令将修改FROM_SCN，用于确定下一次增量备份的起点。
建议在【目标端】每次做完recover动作后，【源端】就执行一次该命令，以免遗忘。



XTTS 案例 - 最后的增量备份和恢复

- 【原库端】表空间设置为READ ONLY

```
alter tablespace XXXX read only;
```

- 【原库端】做最后一次增量备份

```
perl xttdriver.pl -i
```

- 【目标端】做最后一次增量恢复

```
perl xttdriver.pl -r
```

- 在执行完恢复操作后，脚本会自动将目标库重启，不需要人工干预，如果出现到mount状态出现异常，根据情况手工执行后续命令。



XTTS 案例 - 元数据同步

- expdp/impdp 导出导入元数据

```
expdp system dumpfile=expdat.dmp  
directory=data_pump_dir transport_tablespaces=xxx1,xxx2 logfile=export.log
```

```
impdp system dumpfile=expdat.dmp directory=data_pump_dir transport_datafiles=xxx1.dbf,xxx2.dbf logfile=import.log
```

- 注意在导出时，TRANSPORT_TABLESPACES参数需要的表空间要写全，别遗漏。
- 在目标端进行IMPDP时，transport_datafiles需要将所有的数据文件添加进去。
- 如果数据文件很多，人为添加容易出错，可以使用perl xttdriver.pl -e获取全部数据文件列表。



XTTS 案例 - 表空间设置为 READ WRITE

alter tablespace XXXXX read write。

- 到此整个迁移的主要步骤就完成了，剩下的就是要对数据库进行各种检查校验工作了。



XTTS 案例 2- 运营商环境

- 目标端为生产库
- 数据量20TB+
- 每天归档量1TB+
- 本地空间不足2TB
- 网络单进程35MB/S
- 业务中断时间3小时内



XTTTS 每次恢复都需要重启？

pfile.ora

```
*.audit_file_dest='/home/u02/app/oracle/admin/xtt/adump'  
*.db_name='xtt'  
*.compatible='11.2.0.4.0'  
*.db_block_size=16384  
*.db_file_multiblock_read_count=64  
*.db_files=8000  
*.memory_target=21474836480  
*.open_cursors=3000  
*.processes=8000  
*.undo_tablespace='UNDOTBS1'
```



如何加速XTTS?



增量备份提速 6 小时 → 1 小时

ORACLE BCT(BLOCK CHANGE TRACKING)

```
alter database enable block change tracking using file '/oradata/Oracle_change.trace';  
select status, filename from v$block_change_tracking;
```

incremental backup的目的是只备份那些自上次备份以来发生过改变的block。然而，即使只有一小部分发生改变，incremental backup也要读取完整的数据文件。block change tracking功能解决了这个问题。它使用change tracking writer (CTWR) 后台进程，在change tracking file文件中，记录所有数据库中变化的物理位置。启动block change tracking功能后，level 0级的incremental backup依然要扫描整个数据文件，因为change tracking file还没有映射到block的状态。对于后续级别的incremental backups，RMAN使用change tracking data决定哪些需要读取。通过消除对整个数据文件的read，提高了性能。



增量备份传输提速

RMAN> configure device type disk parallelism 8;

- 每个表空间对应8个分片
- 排序分组发送



元数据导入加速

第一次导入

```
impdp directory=DESTDIR1 logfile=tts_imp.log
network_link=ttslink transport_full_check=yes
transport_tablespaces=XXX
transport_datafiles='/XXX/xxx.dbf'
```

exclude=statistics

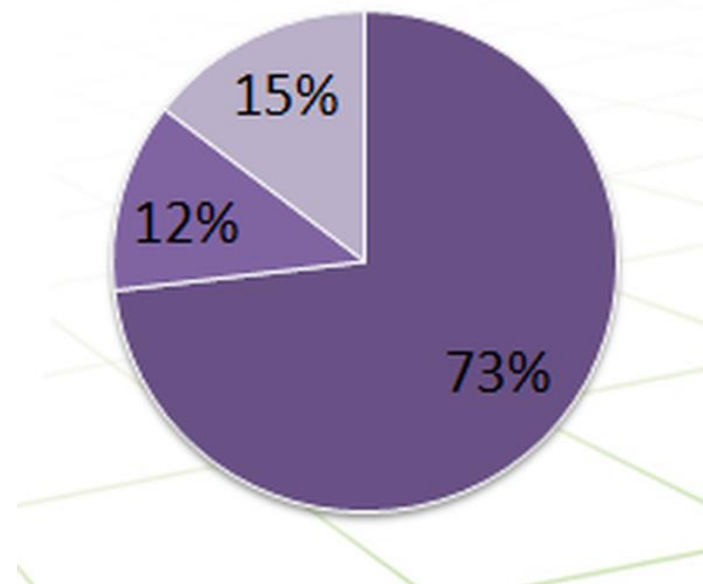
第二次导入

```
impdp directory=DESTDIR1 logfile=tts_imp_2.log
network_link=ttslink schemas=XXX
content=metadata_only exclude=index,table,constraint
parallel=8
```

将过程，视图，包，触发器，统计信息导入，开启并行

元数据导入

- 统计信息导入
- 系统表空间对象导入
- 对象信息导入



迁移前的准备

- 迁移对象统计
- 数据库字符集检查
- 检查原环境是否存在空段
- 失效对象检查
- 基于XMLSchema的XMLType对象检查
- 目标端创建检查用dblink
- 检查源数据库和目标库具有重复名称的表空间
- 检查是否存在应用用户建在system,sysaux,users上的情况



迁移前的准备

- 表空间自包含检查
- 比对新旧环境role
- 比对新旧环境profile
- 在新环境中比对并创建用户
- 生成恢复用户默认表空间和临时表空间的脚本
- 创建非默认的temp表空间
- 生成为应用用户赋对象权限脚本
- 软件包上传



总结

XTTs支持扩字节序迁移

操作灵活简便

停机时间较短

迁移时尽量减少批次，操作越多越容易出错



谢谢大家

上海新炬网络技术有限公司

