

交换分区在业务支撑系统数据库迁移中的应用^①

Application of Oracle9i Exchanging Partitions in Migrating Business & Operation Support System's Database

邓德贵 (联创科技(南京)股份有限公司 江苏 南京 210006)

摘要: 为减少移动业务支撑系统(以下简称 boss)数据库迁移的停业时间,提出了交换分区在 boss 数据库迁移过程中的应用。本文同时介绍了移动 boss 的 crm 数据库迁移的多种方法,并充分阐述了 ORACLE9I 的交换分区在移动 boss 的 crm 数据库迁移的设计和应用。实际证明,交换分区的应用切实减少了数据库迁移过程的停业时间。

关键词: 交换分区 迁移 oracle 分区 boss 数据库

1 前言

随着 boss 系统功能的不断完善,电信运营商的需求不断满足,boss 系统的数据量迅速增长;同时数据库管理系统和操作系统的不断更新换代,以及主机和存储硬件平台的不断升级,这样软件和硬件平台的不断发展和升级就给后台数据库的迁移提出了需求。在此主要简述一下 oracle 数据库的各种迁移方式,以及重点介绍一下交换分区方式在数据迁移中的应用。

2 迁移方式简介

数据迁移分为同构数据库系统和异构数据库系统之间迁移,两种不同数据库系统之间迁移的方式是不同的,在此仅讨论同构数据库系统之间的迁移。Oracle 数据库系统目前在移动 boss 核心系统中使用基本是 oracle9i 版本,少量使用 oracle10g 版本,不过目前 oracle11g 版本已经正式发布商用。也就是说即使同构的 oracle 数据库系统之间还存在着不同操作系统平台,不同存储阵列以及不同数据库版本间的差异。在迁移过程中必须确保数据资源的准确性,同时必须保证在尽可能短的时间内完成迁移,因此对于操作人员来说数据迁移极具挑战性。为了确保数据的平滑迁移,必须周密计划和充分准确,并按照一定步骤来进行,一般是设计数据迁移方案,进行模拟数据迁移,测试数据迁移,压力测试,正式数据迁移,应

用迁移...等步骤。其中制定数据迁移方案至关重要,必须结合实际的数据环境和应用环境来综合考虑。下面就基于移动 boss 的相关数据和应用环境讨论一下几种情况下的数据迁移方式。

移动 boss 系统的数据库大致主要分为省中心数据库,crm 营业数据库,帐务数据库,计费数据库,结算数据库,清单查询数据库,统计数据库,一级 boss 数据库,统一电子渠道数据库,网管数据库,bi 数据库等等。每个省的 boss 系统数据库步署略有不同,其中 crm 营业数据库最为重要和复杂,其操作主要为 oltp 和 olap 型操作结合(这个和每个省的 boss 系统步署相关)。省中心数据库,统一电子渠道和一级 boss 数据库主要为 oltp 型,其他数据库主要为 olap 型。其中涉及营业相关的数据库都是 7×24 小时的数据库,因此对于 crm 营业数据库的迁移必须是在最短的时间内准确迁移。下面对 crm 营业数据库的几种迁移方式进行一下简述,其他数据库可参照执行。

(1)采用 oracle 的 dg 方式进行迁移:采用 oracle 的 data guard 的物理或者逻辑步署不仅仅是容灾考虑,同时也可作为数据迁移的一种考虑方式。在实施 dg 方式迁移时须注意 oracle 不同版本的考虑。

(2)采用操作系统级直接 copy 或 dd 数据库文件的方式:该方式对于数据库比较小的方式进行数据迁移非常方便,对于同平台同版本的小型数据库是最佳

① 收稿时间:2008-10-18

的数据迁移方式之一。

(3)采用 oracle 的 rman 工具进行迁移: 此为 oracle 的物理迁移方式, 必须考虑版本和平台差异。如在 oracle10g 前, rman 工具不支持跨平台迁移, 也就是说源系统是 hp-ux 的操作系统, 就不能通过 rman 工具迁移到 ibm 的 aix 操作系统上。但是对于同平台和同版本的数据库而言, 此迁移方式为较好的迁移方式。因为该迁移为物理迁移方式, 避免了所有逻辑上的整理和考虑, 但是由于是完全物理方式迁移, 因此迁移前后并没有对数据库中的表和索引等逻辑对象的存储进行重整, 也就是说 rman 迁移前后, 数据库中的表和索引的存储并不会通过迁移来改善。

(4)采用 oracle 的 sql*loader 工具进行数据迁移: 前提是必须在迁移的目的库事先建好和源库完全一样的数据库用户和对象。该方式优点是非常灵活, 对于不同格式的文本文件都可以导入到 oracle 数据库中。此种方法在其他非 oracle 数据库迁移至 oracle 数据库是非常有用的。

(5)采用 oracle 的 exp/imp 工具进行数据迁移: 使用该工具可以进行数据库对象及其数据的所有数据库逻辑对象的迁移, 该工具也是使用范围最广, 使用最方便的工具之一。该工具有全库方式, owner 用户方式, table 表方式 3 种级别的逻辑对象方式导出, 同时可仅仅导出导入数据库逻辑定义, 也可数据一并导出导入。同时该工具还有 transport tablespace 的传输表空间的方式导出—这原则上讲应该是一种物理存储的方式了, 当然这种传输表空间方式对于需要导出导出的表空间有一些限制, 具体使用可参见 oracle 的联机文档。传输表空间方式的导出导入是在 oracle8i 版本后提供的。在 oracle10g 版本中还提供了数据泵-data pump(impdp/expdp)方式, 也即加强版的 exp/imp 方式; 该数据泵方式的导出导入的速度远远超出 imp/exp 方式, 其实际运行也是在数据库服务端进行, 而不是以前的 exp/imp 程序在数据库客户端进行的方式, 具体差异见 oracle 的联机文档。在使用 exp/imp 方式进行数据库迁移时务必遵守一个原则是低出高入, 也就是说对于不同的数据库版本而言, 务必使用低版本的数据库客户端 exp 工具

进行逻辑导出, 而使用高版本的数据库客户端 imp 工具进行逻辑导入。同时在使用 exp/imp 逻辑导出导入工具时, 必须注意字符集环境变量的设置了。关于 exp/imp 的具体使用参数可参见 oracle 的联机文档。

(6)采用 oracle 的 dblink 方式进行数据迁移: 前提也是须将目的端的数据库用户和其他逻辑对象建立好, 然后采用 insert into tab_xx from tab_xx@dblink_xx ;commit;的方式进行。同时为加快数据插入速度, 可增加 hint parallel 或 append 的方式。由于此种方式的方便使用, 同时避免了 exp/imp 逻辑导出导入方式的额外文件系统的开销, 该方式在数据迁移中也使用广泛。

(7)采用 oracle 的 on prebuilt table 物化视图方式: 该方式和普通的增量物化视图一样操作, 待数据迁移完成时删除对应的物化视图, 保留同名的逻辑表即可。该方式的数据迁移可极大的减少数据迁移过程中的停止业务的时间。

(8)采用第 3 方工具进行 oracle 数据迁移: 如 dsq, quest, goldengate, powerbuilder, sqlserver 等都有相应的工具来完成 oracle 的数据迁移。

同时对于以上列出的各种 oracle 数据迁移方法都可以根据实际情况进行组合使用。下面重点介绍一下交换分区在 boss 数据库数据迁移中的应用。

3 交换分区的应用

Boss crm 营业数据库是 7×24 小时的数据库, 其存储容量一般也是 t 级别的。因此对于 crm 营业数据库的迁移必须制定详尽的方案, 确保在最短的时间内迁移完成。对于此种情况的数据迁移, 结合文章《Oracle9i 分区技术在 BOSS 数据库中的应用研究》中提到的数据生命周期设计以及分区表的设计, 我们可以将数据迁移分两步进行: 先提前 2 天左右将历史数据导入, 所谓历史数据即指不影响这两天正常营业的静态数据, 这些数据必须和表的设计以及业务紧密联系, 谨慎确认, 这类数据一般会占有比较大的比例; 割接当晚才导入在线数据, 在线数据即为正常营业涉及到的所有动态数据。

为了缩短数据迁移时间, 对于分区表的导出导入,

我们可以采用交换分区的方式进行。具体步骤大致如下:

(1)将认定为静态数据的分区表的分区在割接当晚提前两天作为历史数据导入,然后建立 local 分区索引。

(2)对这些分区表及其索引采用 dbms_stat 包进行分析。

(3)写个过程对这些分区表的每个分区建立对应的普通表和普通索引,其存储存放的表空间及其他参数也必须和对应的分区表和索引的存储存放表空间以及参数是一致的。

(4)将所有这些分区表中提前导入的分区及其索引全部采用 alter table tab_xx exchange partition par_xx with table tab_exg_xx including indexes;方式交换到第 2 步建立的对应的临时表和索引中。

(5)在割接当晚导入这些分区表对应的在线数据分区。

(6)建立这些分区表的 local 分区索引,由于大部分的已经作为历史数据的分区已经导入且已经交换出去了,因此对这些在线数据的分区建立分区索引的时间相对而言就很短了。

(7)将交换出去的分区表及其索引仍然采用第 4 步中的 alter table tab_xx exchange partition par_xx with table tab_exg_xx including indexes;方式再把交换出去的那些历史数据及索引分区交换回来。

(8)对这些在线分区表及其索引采用 dbms_stat 包进行分析。

从上我们可以明显看出的是采用交换分区方式进行数据迁移,减少了那些历史分区的索引建立时间。对于我们 crm 营业数据库中的营帐的台帐日志类大表而言节约了大量时间,减少了业务的停机时间。具体我们见下面的例子说明,同时我们对 alter table tab_xx exchange partition par_xx with table tab_exg_xx including indexes;的两种方式进行了跟踪分析。

3.1 测试环境

主机: ibm p595 20 颗 2.1g 主频 cpu(双核), 68g

内存

存储: emc dmx-3 存储阵列 raid10 磁盘划分

3.2 软件环境: aix 5307 oracle9208(单机)

在程序中 mode=1 为分区表的分区交换出至普通表, mode=2 为普通表交换回分区表的分区。具体交换出的语句为上面提到的 alter table tab_xx exchange partition par_xx with table tab_exg_xx including indexes; 测试的数据存储为 88g, 索引存储为 50g, 用上面语句交换出的时间大致为 30 秒完成。在这么短时间即可完成交换出, 我们可以推断 oracle 的内部肯定是仅仅修改一些内部的系统表来完成的, 因此我们对此采用 alter session set events '10046 trace name context forever,level 12' 语句来进行跟踪, 从日志中我们可以发现, oracle 内部通过 9 个步骤的系统表更新来完成此次的交换出操作, 由于篇幅限制这里就不贴出日志文件中获取的 sql 语句。

再次采用语句 alter table tab_xx exchange partition par_xx with table tab_exg_xx including indexes;将分区交换回来的时间大致为 10 分钟左右, 经检查跟踪日志发现和上面的交换出是一样的, 而这次时间为何要 10 分钟之多, 分析日志发现在其中的一步过程中 select 1 from "UCR_CRM1"."CRM1_TEMP51" where TBL\$OR\$ IDX\$PART\$ NUM("UCR_CRM1"."TF_BH_TRADE",0,0,65535,"ACCEPT_MONTH") != :1 的执行计划为 "TABLE ACCESS FULL CRM1_TEMP51", 因此对于交换出的时候, 由于交换中间表 CRM1_TEMP51 为空, 执行速度很快; 但是对于交换回来的时候, 由于交换中间表 CRM1_TEMP51 的数据很多, 高水位比较高, 因此全表扫描的速度就比较慢了。那如何来避免这种全表扫描的情况呢, 我们发现对于 alter table tab_xx exchange partition par_xx with table tab_exg_xx including indexes;语句实行分区交换的时候默认选项是 oracle 对交换中间表的数据进行确认是否符合交换进的那个分区的数据条件, 它采用全表扫描的方式对交换中间表的数据进行确认。我们可以使用语句 alter table tab_xx exchange partition par_

xx with table tab_exg_xx including indexes without validation;这时我们跟踪其日志可以发现其中一步全表扫描的 sql 变为"select /*+ first_rows(1) ordered */ 1 from "UCR_CRM1"." CRM1_TEMP51""A","UCR_CRM1"."TF_BH_TRADE""B"where "A"."CHARGE_ID" = "B"."CHARGE_ID" and "A"."PARTITION_ID" = "B"."PARTITION_ID"and tbl\$or\$idx\$part\$num("UCR_CRM1"."TF_BH_TRADE",0,0,0,"B"."PARTITION_ID") < 5 or tbl\$or\$idx\$part\$num("UCR_CRM1"."TF_BH_TRADE",0,0,0,"B"."PARTITION_ID")>5)and tbl\$or\$idx\$part\$num("UCR_CRM1"."TF_BH_TRADE",0,0,0,"A"."PARTITION_ID") <> 5 and rownum < 2" 的执行计划为"INDEX FULL SCAN PK_CRM1_TEMP51",从而避免了上面的"TABLE ACCESS FULL CRM1_TEMP51",缩短了分区表交换进时间。至于仍然要走主建的全索引扫描则是 oracle 的内部程序,无从查起。通过 without validation 方式的交换进分区则只用了大概 2.5 分钟即可完成。同时在此交换过程中未出现任何索引和其他对象的失效或不可用,且交换分区表的数据和索引分区的分析数据仍然保留。因此,如果可提前导入的分区越多,通过这种交换分区的使用,在割接当晚停业务的时间就越短。就拿某省的 boss crm 营业数据库割接时间来看,用这种方式可以节约 50

分钟以上的建立索引时间,对于近 2t 存储的 crm 营业数据库迁移在 5 个小时内的停业时间来看是非常可观的,可以将节约出来的时间来加强业务和数据测试以确保迁移的完整和正确性。

4 结论

从上面的分析和案例可以看出,在大数据量和停业时间短的情况下的数据迁移,采用分区交换的方式切实减少了当晚割接的建立索引时间,从而减少割接停业时间,确保 boss 业务的持续正常运营。希望以上分析能对其他类似系统的迁移提供可参考之处。同时从上也可以看出,对于这种大数据量且要求 7×24 小时服务的数据库系统而言,表和索引的源头设计是至关重要,它不仅要求能长时间的持续提供快速的实时业务响应,还要求能在最短的时间内进行系统迁移,文章《Oracle9i 分区技术在 BOSS 数据库中的应用研究》中关于表的设计也可向大家提供一些参考。希望我们的分析和应用能对我们国家的数据不断增长的大型数据库应用的设计和迁移提供参考。

参考文献

- 1 邓德贵.Oracle9i 分区技术在 BOSS 数据库中的应用研究.计算机系统应用,2006,15(10):73-75.
- 2 Lorentz D, Gregoire J. SQL.PDF.oracle 官方电子文档:Oracle Corporation, March 2002:11-2-11-100.