

Oracle9i 分区技术在 BOSS 数据库中的应用研究

The applying & research of oracle9i's partition technics in BOSS databse

邓德贵 (南京联创科技股份有限公司 南京 210006)

摘要:本文介绍了移动 BOSS 的 CRM 数据库生命周期设计方法,并充分阐述了 ORACLE9I 的分区表和索引技术在移动 BOSS CRM 数据库中的应用。

关键词:oracle 分区 BOSS 数据库

1 前言

BOSS - Business & Operation Support System, 移动业务运营支撑系统的简称。BOSS 系统通常包括营业 (CRM)、帐务、计费、清单查询、统计、结算、联机采集等一系列业务模块。作为其核心应用系统 - 营业系统或称 CRM 系统,是一个典型的以 OLTP 系统为主的,并有一定的 OLAP 操作的大型综合型系统,是移动 BOSS 系统严格要求 7 × 24 在线的生产系统。由于其业务和数据的复杂性,该系统必须既要在最短的时间内响应前台营业的需要,又要在最短的时间响应用户和后台维护人员的查询需要。该系统的数据库量是每天不断增长的,而 ORACLE 等后台数据库会由于数据量的不断增长和数据底层存储由于数据的经常性的进行 DML 操作而造成的变更导致数据库系统性能下降,影响移动 BOSS 系统的正常运行。因此,对于类似系统的核心大表的物理设计是非常关键的,在此探讨一下移动 BOSS CRM 数据库的生命周期设计方法和 ORACLE9I 的分区表和索引技术在移动 BOSS CRM 数据库中的应用。

2 数据生命周期设计

根据移动 BOSS 数据库的特点,在 CRM 库一般都是存放资源表,资料表,台帐表,帐单表等核心大表,这些表的数据量都是非常大的,基本所有业务的操作都是围绕这些核心表进行频繁 DML 和 SELECT 操作的。同时由于业务的需要,这些表在线保存的时间周期还是比较长的。因此结合业务和数据操作特点,我们在开始进行 BOSS 系统 CRM 数据库总体设计时就重点考虑这些核心大表的数据生命周期。数据生命周期设计即是指在数据从产生到消亡的整个生命周期中,

数据需要不同级别的性能、可用性、迁移、保留和废弃处置,我们如何界定和保持数据的时效性,可用性和准确性的设计方法。因此,根据这些数据及业务的特点,我们可以将他们分为:

表 1 生命周期阶段划分

数据类型	生命周期阶段划分			
	空闲	占有	废弃	
台帐日志	当前	历史	在线备份	库外备份
缴费日志	当前	历史	在线备份	库外备份
用户资料	当前	历史		
帐户资料	当前	历史		
客户资料	所有			
帐单	当前	历史	在线备份	库外备份

如表 1,生命周期阶段的第 1 个阶段主要操作是 DML 和 SELECT 操作,而第 2 个阶段主要是 SELECT 操作和少许的 DML 操作,而第 3 个阶段为前台不进行 SELECT 和 DML 操作、后台少许查询操作,第 4 阶段为基本不进行操作可以清出数据库进行库外备份了。我们可以对不同类型的表的不同生命周期阶段进行不同的物理设计,将它们放置在不同的表空间(数据文件)上,确保资源得到最高效的利用。同时可以根据上表的生命周期阶段给 BOSS 的其他模块系统或经营分析系统提供数据,而每个模块系统所占用的主机和存储资源也是不同的,这样就良性的纳入信息生命周期管理系统环节(ILM)中了。因此我们可以根据表一对移动 BOSS CRM 库的核心大表进行分表或分区设计,来确保 CRM 数据库的性能需求。同时我们根据表一生

命周期阶段划分原则来设计这些核心大表的数据转移方式。

3 数据转移方式设计

结合以上表 1 数据生命周期设计和数据特点,我们可以设计 BOSS CRM 数据库核心大表的数据转移方式。

表 2 数据转移方式

数据类型	生命周期阶段划分				转移方式
	空闲	占有	废弃		
资源					delete
台帐日志	当前	历史	在线备份	库外备份	分区操作
缴费日志	当前	历史	在线备份	库外备份	分区操作
用户资料	当前	历史			delete
帐户资料	当前	历史			delete
客户资料	所有				
帐单	当前	历史	在线备份	库外备份	delete

通过表 2 制定的数据转移方式,结合 ORACLE9I 分区表和索引的特性,同时考虑在线系统使用的有效性和高性能性,我们可以对那些分区操作转移方式的表进行 EXCHANGE 分区操作。因为在 oracle9i 中,EXCHANGE 分区操作仅仅是数据库内部进行数据字典的变更,而不是真正意义上的表的数据的 DML 操作,从而避免大量的 REDO 和 UNDO 的日志操作,最大限度的减少了数据库性能的损耗和数据转移时间的消耗,保证了 CRM 在线系统的高效运行。至于数据转移的时间点则应根据不同类型的表的不同业务情况和分区设计来制定。

4 分区表和索引的设计

分区表的设计主要是根据业务操作类型和数据量的特点,并结合以上表二的数据转移方式来设计的。分区表的主要分区方式有 RANGE、HASH、LIST 以及 RANGE、LIST、RANGE、HASH 复合分区。如数据转移方式为分区操作的日志类表可以设计为按时间方式的 RANGE 分区,而数据转移方式为 DELETE 操作的资料类表可以设计为按 Xxid 方式的 RANGE 分区等。HASH 分区则适用于那些查询比较多,但是插入等操作相对少些的大表。根据业务级别的不同要求,需要对分区表的每个分区的存储进行不同的物理设计。

分区索引的使用主要是相对分区表而言,对于分区表可以建立 GLOBAL、LOCAL 分区索引和普通索引三种情况。我们对分区索引的设计应充分结合分区表的分区方式,数据转移方式,以及业务及数据操作特点来综合考虑。

结合以上表 2 数据转移方式来设计索引的分区方式,如台帐日志类型的表,由于数据转移方式是利用 ORACLE9I 分区表的 EXCHANGE 功能,而又由于这些表都是在线生产表(7×24),因此我们必须将此类表的索引方式设定为 LOCAL 型的,因为你所交换的分区不能影响其他分区的数据和索引。否则会由于 EXCHANGE 交换分区的数据转移,导致这些表的全局分区索引和普通索引无效,从而影响在线系统的应用。为表述清楚,特举例如下:

```

create table TF_A_PAYLOG
(
    PARTITION_ID NUMBER(4) not null,
    ACCT_ID NUMBER(16) not null
)
partition by range (PARTITION_ID)
(
    partition PAR_TF_A_PAYLOG_1 values less than (2),
    partition PAR_TF_A_PAYLOG_2 values less than (3),
    partition PAR_TF_A_PAYLOG_3 values less than (4),
    partition PAR_TF_A_PAYLOG_4 values less than (5),
    partition PAR_TF_A_PAYLOG_5 values less than (6),
    partition PAR_TF_A_PAYLOG_6 values less than (7),
    partition PAR_TF_A_PAYLOG_7 values less than (8),
    partition PAR_TF_A_PAYLOG_8 values less than (9),
    partition PAR_TF_A_PAYLOG_9 values less than (10),
    partition PAR_TF_A_PAYLOG_10 values less than (11),
    partition PAR_TF_A_PAYLOG_11 values less than (12),
    partition PAR_TF_A_PAYLOG_12 values less than (MAX-
VALUE)
)
;
create index IDX_TF_A_PAYLOG_ACCT_ID on TF_A_PAY-
LOG (ACCT_ID);
insert into tf_a_paylog values(9,2005090902975248);
commit;
create table tmp_tf_a_paylog (partition_id number(4)
not null,acct_id number(16) not null);
SQL > select status from user_indexes where index_

```

```

name =IDX_TF_A_PAYLOG_ACCT_ID';
STATUS
-----
VALID
SQL > alter table TF_A_PAYLOG exchange partition PAR_
TF_A_PAYLOG_9 with table tmp_tf_a_paylog ;
Table altered.
SQL > select status from user_indexes where index_
name =IDX_TF_A_PAYLOG_ACCT_ID';
STATUS
-----
UNUSABLE
从上可以看出,建成普通索引的话就导致整个分区表的
索引状态处于 UNUSABLE 状态。
SQL > drop index IDX_TF_A_PAYLOG_ACCT_ID;
Index dropped.
SQL > create index IDX_TF_A_PAYLOG_ACCT_ID on TF_A
_PAYLOG (ACCT_ID) local;
Index created.
SQL > select status from user_ind_partitions where index
_name =IDX_TF_A_PAYLOG_A
CCT_ID';
STATUS
-----
USABLE
SQL > alter table TF_A_PAYLOG exchange partition PAR_
TF_A_PAYLOG_9 with table t
_tf_a_paylog ;
Table altered.
SQL > select status from user_ind_partitions where index
_name =IDX_TF_A_PAYLOG
CCT_ID';

```

从上可以看出,建成 LOCAL 索引方式的话,只影响所交换的那个分区,不影响其他分区的正常使用。大家还可以关注 EXCHANGE 的 INCLUDING INDEXES 选项,关于 EXCHANGE 的详细使用,可以参考 ORACLE 的官方文档,在此不作详细论述。

而资料表的数据转移方式是 DELETE 方式,不是通过 EXCHANGE 交换分区方式,因此这些表的索引设定方式可以不受在线数据转移方式的影响。因此需要结合业务及数据操作特点来设计这类分区表的索引方式,具体可以参考下节分区索引对执行计划的影响来设计。

值得注意的是,我们还应结合数据转移方式的实现,考虑在 ORACLE9I 系统中分析数据随着 EXCHANGE 的分区数据的交换而造成分区统计数据交换而带来的后果的处理,在此不作详述。

5 分区索引对执行计划的影响

对于分区表上的普通索引对执行计划的影响与普通表上的普通索引对执行计划影响是类似的,在此不作讨论。对于分区表上的 LOCAL 分区索引,必须注意 WHERE 条件的写法,如写法不恰当的话,极有可能会造成分区索引的 PARTITION ALL 方式 RANGE SCAN,这样对于分区数较多的分区表的话,会比较严重的消耗系统性能,这点在系统比较繁忙的情况表现尤为明显。也即,在 WHERE 条件中要能显式的赋值给 PARTITIONID 为一个明确的传入参数值或明确的值的限定范围,这样 ORACLE9I 才能在单个 PARTITION 或范围 PARTITION 内进行索引的 RANGE SCAN,减少索引 PARTITION ALL 的扫描,减少索引扫描的 IO 次数,从而达到减少系统性能的消耗。

6 总结

BOSS 数据库无论从数据量和交易量的方面来看,在商用数据库应用方面都是遥遥领先的,其对数据库系统的性能要求更是严格的电信级要求。作为这么大的一个数据库应用系统,不仅仅是在上线时能满足业务性能要求,而且必须在后期的使用过程中能持续满足业务的性能要求,这样就要求我们在数据库的设计时就应考虑全面,减少后期调优和维护的人力和物力成本。基于以上原则,本文详细阐述了 BOSS CRM 数据库的核心大表的生命周期设计方法,以及如何结合 ORACLE9I 的分区表和索引的特点来设计这些核心大表。希望对其他类似 BOSS CRM 的生产系统的数据库设计有所借鉴和帮助,来共同提高我们的数据库设计和应用水平。

参考文献

- 1 Gaja Krishna Vaidyanatha, Kirtikumar Deshpande, John Kostelac. oracle 性能优化技术内幕,北京 机械工业出版社,2002.5.
- 2 SQL. PDF. Oracle9i 官方电子文档。