

基于BCV和LogMiner实现增量数据抽取的应用

杨从法¹ 宋兴彬² (1.淄博市国家税务局信息中心 山东 淄博 255047;

2.山东省国家税务局信息中心 山东 济南 250002)

摘要: 介绍了不同数据抽取方式的优缺点,基于 EMC 存储联机业务持续性卷 BCV 数据备份和 Oracle LogMiner 分析归档日志原理,分析了数据抽取复制方案,实现了省级数据集中应用生产环境到分析平台中增量数据抽取。

关键词: 数据抽取;业务持续性卷;重做日志;归档日志;增量抽取

Application of Incremental Data Extraction Based on BCV and LogMiner

YANG Cong-Fa¹, SONG Xing-Bin² (1. Information Center of Zibo State Tax Bureau, Zibo 255047, China;

2. Information Center of Shandong State Tax Bureau, Jinan, 250002, China)

Abstract: The advantages and disadvantages of different kinds of data extraction are introduced. The detailed incremental data extraction scheme based on Business Continuance Volume of EMC and Oracle LogMiner are analyzed. The incremental data extraction from production platform to data analysis platform in provincial data centralized is achieved.

Keywords: data extraction; business continuance volume; redo log; archived log; incremental data extraction

1 引言

山东省国税局实现省级数据大集中后,为满足数据增值应用,需要将生产环境数据抽取到分析利用平台 ODS(操作型数据存储)中,用于后续加工处理。出于安全考虑,不允许直连生产数据库进行数据抽取,前期我们基于 EMC 存储业务持续性卷 BCV 备份,采取每天全量数据抽取方式,满足了数据分析工作需要。随着生产数据不断增加,全量抽取后,每天全部数据汇总和再加工一晚上不能完成,影响了白天应用。为进一步满足抽取效率和增量数据处理需要,我们研究利用 Oracle LogMiner 日志分析实现了从 BCV 备份环境到数据分析利用平台的增量数据抽取,满足了分析预警和纳税评估工作需要。

2 数据抽取

数据抽取是从源数据系统抽取部分或全部数据到

目标系统,从而在目标系统再进行数据加工利用的过程。数据抽取分为全量抽取和增量抽取多种方式,实现方式不同,数据抽取效率也不一样,全量抽取较为简单,在此不做介绍,增量数据抽取目前方式主要有以下几种方式^[1]:

2.1 时间戳方式

时间戳是一种基于快照变化的数据捕获方式,需要在源表上增加时间戳列,更新数据表数据时,同时修改时间戳列值。数据抽取时,通过比较系统时间与时间戳列值来决定抽取变化数据,实现增量抽取。时间戳方式性能较好,抽取相对简单,缺点是无法捕获时间戳以前数据 delete 和 update 操作,在数据准确性上受到一定限制。

2.2 日志表方式

该方式通过分析数据库自身在线日志判断变化数据。在对源数据表进行 insert、update 或 delete 操

收稿时间:2009-08-07;收到修改稿时间:2009-09-04

作同时就可提取数据,变化数据保存在日志表中,通过这种方式捕获变化数据,然后利用视图方式提供给目标系统。如 Oracle 提供的物化视图、DSG 和 GoldenGateTDM 等第三方数据复制工具都采用了该方式,其优点是数据抽取性能高,缺点是数据操作时要同时修改数据表和日志表数据,对业务系统性能有一定影响。

2.3 全表比对方式

全表比对方式要事先为抽取的表建立结构类似的临时表,临时表记录源表主键以及根据列数据计算出来的校验码。每次进行数据抽取时,对源表和临时表进行校验,决定源表数据是 insert、update 还是 delete 操作。该方式优点是对源系统影响较小,缺点是性能较差,表中没有主键或唯一列且含有重复记录时准确性更差。

2.4 触发器方式

需要在源数据表上建立 insert、update 和 delete 等触发器,当源数据变化时,相应触发器将变化数据写入临时表,抽取线程从临时表中抽取数据,临时表中抽取过的数据被标记或删除。如 InforEAI 就是采用该方式实现增量抽取,现正在我省国税系统出口退税审核系统数据集中使用。其优点是数据抽取效率高,缺点是要在业务表建触发器,对业务系统性能和安全性有一定影响。

通过对以上增量数据抽取方式分析,本着不直接从生产数据库进行抽取的原则,我们利用已经建立的 BCV 备份数据库进行增量数据抽取。

3 日志分析 LogMiner

我省主要省局集中应用系统采取 BCV 作为一种备份策略^[2],BCV(Business Continuance Volume)是 EMC 存储具有的独立寻址访问的联机业务持续性卷的简称,作为备份系统效率高,可达到数据卷镜像拷贝的效果,缺点是基于存储设备覆盖方式,每次备份后原先在备份库上做的数据抽取操作全被覆盖,以上列举分析的增量抽取方式都无法在此备份数据库上实现。

LogMiner 是 Oracle 8i 后提供的日志分析工具,由一组 PL/SQL 包和部分动态视图组成,可以分析在线日志文件、归档日志文件和重做日志文件 redo log^[3]。

Oracle 所有逻辑变化都记录在重做日志 redo log 中,在归档模式下,redo log 被写为归档日志文

件,通过分析日志可以查明数据库逻辑更改情况,获得数据库更改历史、更改类型(Insert、Update、Delete 和 DDL 等),Oracle 通过 redo log 保证数据库事务可以被重新提交或回滚,从而在数据库发生故障后可以做到完整恢复^[4]。

为保证生产数据库安全和受 BCV 备份条件限制,我们在 BCV 备份数据库上利用 LogMiner 分析归档日志进行增量抽取。

4 归档日志分析

根据 BCV 备份原理,其上的数据库同生产环境数据库数据块、字符集等完全相同,打开 BCV 数据库系统,利用 LogMiner 分析 BCV 归档日志,获得在生产数据库上提交的事务。以下是主要工作步骤:

4.1 安装 LogMiner

以 sys 用户运行两个脚本,创建分析日志文件的 DBMS_LOGMNR 包和创建数据库字典文件的 DBMS_LOGMNR_D 包。

```
sql>@?/rdbs/admin/dbmslm.sql
```

```
sql>@?/rdbs/admin/dbmslmd.sql
```

4.2 修改生产数据库参数 log_parallelism

初始化静态参数 log_parallelism 指定 redo 分配的并发级别,将值设为 1 后重启数据库^[5]。

4.3 生产数据库启用追加日志

基于数据库级启用追加日志,指示生产数据库向重做日志增加主键和唯一索引信息,从而可以在分析平台数据库里正确标识相同行^[6]。在生产数据库中,执行下面语句将主键和唯一索引添加到归档日志后重启数据库。

```
alter database add supplemental log  
data(primary key,unique index) columns;
```

4.4 修改 BCV 数据库初始化参数

在 BCV 数据库参数文件 init.ora 中,指定数据字典文件位置,添加 UTL_FILE_DIR 参数值为服务器中放置数据字典文件目录,之后重启数据库,使新参数生效。

```
alter system set UTL_FILE_DIR='/ctais2/  
oracle' scope=spfile;
```

4.5 从 BCV 数据库导出数据字典文件

数据字典文件是 dbms_logmnr_d 包将数据库中数据字典导出的外部文本文件,用于存放表、对象与 id 号之间的对应关系。LogMiner 依赖于外部文件提供的

数据字典,可方便分析其他数据库重做日志^[7]。生产数据库数据字典是不断变化的,每次利用 Log-Miner 分析归档日志时,需执行以下语句重新导出字典文件。

```
Begin
  dbms_logmnr_d.build(
    dictionary_filename=>'dict.ora',
    dictionary_location=>'/ctais2/oracle');
end;
```

4.6 创建数据库链接 DBLINK

创建分析归档日志数据库到 BCV 数据库的连接:
create database link redosql connect to ctais2 identified by oracle using 'ctais';

创建分析平台数据库到 BCV 数据库连接,用于不一致数据的修复。

```
create database link ctais_dbl_main connect to
ctais2 identified by oracle using 'ctais_BCV'
```

4.7 建立归档日志文件名列表

(1) 从分析数据库中获得已被分析过的归档日志文件名,放到表 archlog 中保存。

```
drop table ctais2.archlog;
create table ctais2.archlog as select name from
archlog;
```

(2) 从生产环境获得归档日志文件名表

生产环境归档日志文件名存到视图 v\$archived_log 中,通过 BCV 后,视图 v\$archived_log 复制到 BCV 数据库环境,内容不再变动,将 NAME 列拷贝到 archlog 表中:

```
drop table ctais2.archlog;
create table ctais2.archlog as select name from
v$archived_log@redosql;
```

(3) 需要分析的归档日志文件名表

从生产环境获得归档日志文件名表减去已经分析的归档日志文件名表,获得需要分析的归档日志文件名表。

```
create table ctais2.archminus as (select name
from ctais2.archlog) minus (select name from
ctais2.archlog);
```

(4) 添加归档日志文件到列表

将需要分析的日志放到日志文件列表中,每次分析的日志文件个数由机器性能决定。

```
create or replace procedure add_logfiles
as
```

```
cursor c_log is select name from
ctais2.archminus;
count pls_integer:=0;
my_option pls_integer:=dbms_logmnr.new;
begin
  for c_log_rec in c_log
  loop
    dbms_logmnr.add_logfile(logfilename=>c_log_r
ec.name,options=>my_option);
    my_option:=dbms_logmnr.addfile;
    dbms_output.put_line('added logfile'||c_log_
rec.name);
  end loop;
end;
```

4.8 利用 LogMiner 进行分析

LogMiner 可以进行无限制或有限制条件的日志分析,通过设置 dbms_logmnr.start_logmnr 参数,可以控制要分析日志文件的范围^[8]。每执行一次 dbms_logmnr.start_logmnr 分析 2 个日志文件,得到需要的 sql 语句保存起来,生产环境当天归档日志全部分析完后,将得到的 sql 语句按照表空间进行分组,可并行执行提高效率。

(1) 启动 LogMiner

```
execute
dbms_logmnr.start_logmnr(DictFileName=>'/cta
is2/oracle/dict.ora',options=>dbms_logmnr.commit
ted_data_only);
```

执行后动态性能视图 v\$logmnr_contents 包含 LogMiner 分析得到的所有信息。

(2) 创建 redosql 表保存 sql_redo,sql_undo

v\$logmnr_contents 包含 LogMiner 分析得到重做和回滚语句,这里仅抽取部分信息。

```
create table ctais2.redosql as select
sql_redo,sql_undo from v$logmnr_contents
where username='CTAIS2' and sql_redo not like
'%SYS%';
```

(3) 结束分析,释放系统分配给 LogMiner 的所有资源。

```
exec dbms_logmnr.end_logmnr();
```

(4) 删除 sql 中 rowid。由于生产数据库表中 rowid 和分析平台 rowid 不相同,删除 sql 中 rowid。

```
update redosql1 set sql_redo=rpads(sql_
redo,instr(sql_redo,'ROWID')-5,' ')||';'
```

```
where instr(sql_redo,'ROWID')>0;
```

(5) 抽取需要表的 sql 语句

运行脚本 create_redo.sql, 从表 redosql 中抽取分析平台需要表的 sql, 以两个表为例:

```
create table ctais2.redosql1 as select sql_redo
from ctais2.redosqlz
```

```
where sql_redo like '%"CTAIS2". "DJ_NSRXX"%'
or sql_redo like '%"CTAIS2". "RD_NSRZG_LSXX"%';
```

(6) 运行脚本 redo_table.sql, 按照表所在表空间进行分组。

```
drop table ctais2.redosql3;
create table ctais2.redosql3 as select
sql_redo from ctais2.redosql1
where sql_redo like '%"CTAIS2". "DJ_BG"%'
or sql_redo like '%"CTAIS2". "DJ_BGXM"%';
```

5 数据装载

将 sql 语句传输到分析平台, 执行后生成与生产环境相同的数据。

(1) 执行 redo_tran.sql 脚本, 进行 sql 语句传输:

```
drop table redosql3;
```

```
create table redosql3 as select * from redosql3@
redosql;
```

(2) 在分析平台创建执行 sql 语句的存储过程

```
create or replace procedure p_redosql3
as
begin
declare
sql_str varchar2(4000);
l_redo number;
begin
for c in (select * from redosql3) loop
sql_str:=replace(c.sql_redo,';','');
begin
execute immediate sql_str ;
insert into redosql32 values(c.sql_redo,
sysdate);
exception
when others then
```

```
insert into redosql33 values(c.sql_redo,
sysdate);
```

```
commit;
```

```
end;
```

```
commit;
```

```
end loop;
```

```
commit;
```

```
end;
```

```
commit;
```

```
end;
```

(3) 在分析平台 crontab 中定时执行计划, 减少人工干预, 为提高效率可利用多组存储过程在分析平台并行执行。

```
variable jobno number;
begin
DBMS_JOB.SUBMIT(:jobno,'p_redosql3;'.sy
sdate);
commit;
end;
```

6 数据完整性保证

我们通过以下措施保证抽取数据一致性和完整性:

(1) 事务先后顺序保证。通过 LogMiner 分析归档日志, 严格按照生产环境产生归档日志顺序进行分析, 保证事务在分析平台完全按照在生产环境中顺序执行。

(2) 日志跟踪连续性检查。从生产平台获得归档日志表减去已经分析了的归档日志表, 获得需要分析的日志表。

(3) 数据一致性检查。每天凌晨 BCV 完成后, 统计比较 BCV 数据库和分析平台数据库表记录数是否一致, 在分析平台删除不一致的表, 重新创建该表和索引。以 DJ_NSRXX 表为例:

```
drop table DJ_NSRXX;
create table DJ_NSRXX as select * from
ctais2.DJ_NSRXX@ctais_dbl_main;
CREATE UNIQUE INDEX "CTAIS2"."PK_DJ_NSRXX"
ON "CTAIS2"."DJ_NSRXX" ("NSRDZDAH")
PCTFREE 10 INITRANS 2 MAXTRANS 255
COMPUTE STATISTICS
STORAGE(INITIAL 1048576 NEXT 1048576
```

```
MINEXTENTS 1 MAXEXTENTS 2147483645  
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1  
BUFFER_POOL DEFAULT)  
TABLESPACE "CTAIS2_IDX";
```

7 结语

BCV 备份条件下, 我们利用 LogMiner 实现了增量数据抽取, 满足了增量刷新和数据分析需要。其优点是数据变化 sql 语句传输量小, 自动化程度高, 无需另外采购第三方抽取工具, 对于生产环境、BCV 备份环境基本没有影响, 具有较强的应用价值, 缺点维护较复杂, 第一次安装需要较长时间, 技术细节较多, 要有熟悉 Oracle 管理人员专门负责维护。

参考文献

1 Zctitan. 面向数据集成的 ETL 技术研究. [2008-03-05] <http://data-base.ctocio.com.cn/tips/263/7832263.shtml>

2 宋兴彬, 杨从法. 基于 BCV 和 VPD 技术实现数据分发的应用研究. 计算机工程与设计, 2007, 28(13): 3170 - 3271.

3 Shining_forever. 总结 LogMiner 使用及各种问题处理. [2005-06-19] <http://www.itpub.net/380732.html>

4 盖国强. 深入浅出 Oracle: DBA 入门、进阶与诊断案例. 北京: 人民邮电出版社, 2009.

5 Yorking. LOG_PARALLELISM 的设置影响 LogMiner 使用. [2007-07-05] <http://www.oracle.com.cn/view-thread.php?tid=121466>

6 Oracle 如何配置逻辑备用数据库. [2008-02-09] <http://www.know-sky.com/390166.html>

7 Thomas Kyte. Apress_Expert one on one Oracle. <http://download.csdn.net/source/1295145>.

8 lyd. LOGMNR 终结版. [2005-03-22] <http://blog.oracle.com.cn/html/46/t-50746.html>