

某 ERP 数据库性能优化案例

王飞鹏

13811817203

本优化文档针对的 DB2 版本为 V9.7fp6,包括两部分内容,一部分是已经采取的优化操作,另一部分是对后续运维建议。

1. 所实施的优化操作

1.1 JCC 驱动一定要和 DB2 版本匹配

例如, DB2 服务器是 V9.76,那么就需要从 db2 的安装包 /opt/IBM/db2/V9.7/java 下面复制 db2jcc.jar 和 db2jcc_license_cu.jar 到 JDBC 库路径下面。否则会由于 JDBC 驱动不同版本的兼容问题,会出现一些不可预见的问题,这个一定要注意。

1.2 参数调整

调整后的:

```
update db cfg using MAXLOCKS 80;
update db cfg using LOCKLIST 40960;
update db cfg using LOCKTIMEOUT 30;
update db cfg using CATALOGCACHE_SZ 102400;
update db cfg using LOGBUFSZ 10240;
update db cfg using PCKCACHESZ 102400;
update db cfg using sheapthres_shr 1638400;
update dbm cfg using SHEAPTHRES 0;
update db cfg using sortheap 819200;
update db cfg using STMT_CONC LITERALS;
```

update dbm cfg using DIAGSIZE 1024;

调整前的:

Percent. of lock lists per application	(MAXLOCKS) = 80
Max storage for lock list (4KB)	(LOCKLIST) = 10240
Lock timeout (sec)	(LOCKTIMEOUT) = -1
Catalog cache size (4KB)	(CATALOGCACHE_SZ) = 3072
Log buffer size (4KB)	(LOGBUFSZ) = 2048
Package cache size (4KB)	(PCKCACHESZ) = AUTOMATIC(2400)
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR)	= 1638400
Sort heap threshold (4KB)	(SHEAPTHRES) = 40000
Sort list heap (4KB)	(SORTHEAP) = AUTOMATIC(4096)
Statement concentrator	(STMT_CONC) = OFF

1.3 解决多表关联字段不匹配问题

当多表关联时, 优化器选择哈希连接, 由于字段类型不匹配导致查询性能非常低, 这种情况在这个项目中应该存在多处, 随着目标表数据量的变大, 查询的性能问题逐渐暴露出来。

解决办法是, 从设计上调整表设计, 将关联字段调整为同一类型, 如果是字符串, 长度也要求完全一致。

下面这个例子就是一个典型, 调整好, 性能提高了 10 倍:

```
CREATE TABLE
  HN_INVRETURNTIME
  (
    PK_INVBASDOC VARCHAR(20),
    PK_INVVCL VARCHAR(20),
    DATEBEGIN TIMESTAMP,
    DATEEND TIMESTAMP
  )
```

```
CREATE TABLE
  HN_JINGAOMX
  (
    VDEF9 VARCHAR(100),
    VDEF8 VARCHAR(100),
    VDEF7 VARCHAR(100),
    VDEF6 VARCHAR(100),
    VDEF5 VARCHAR(100),
    VDEF4 VARCHAR(100),
```

```

VDEF3 VARCHAR(100),
VDEF20 VARCHAR(100),
VDEF2 VARCHAR(100),
VDEF1 VARCHAR(100),
PK_CUMANDOC CHARACTER(20),
VDEF19 VARCHAR(100),
VDEF18 VARCHAR(100),
VDEF17 VARCHAR(100),
PK_CHANNEL CHARACTER(20),
VDEF16 VARCHAR(100),
VJINGAOCODE VARCHAR(50),
VDEF15 VARCHAR(100),
TRETURNTIME CHARACTER(19),
VDEF14 VARCHAR(100),
NRETURNNUMBER DECIMAL(20,8),
VDEF13 VARCHAR(100),
VDEF12 VARCHAR(100),
VGOODSCODE CHARACTER(20),
VDEF11 VARCHAR(100),
VCODE CHARACTER(20),
VDEF10 VARCHAR(100),
VBILLTYPE VARCHAR(20),
PK_JINGAOMX CHARACTER(20) NOT NULL,
NWAITUSENUMEBER DECIMAL(20,8),
NDISCOUNT DECIMAL(20,8),
NPRICE DECIMAL(20,8),
TS CHARACTER(19),
DR SMALLINT DEFAULT 0,
NSUMMONEY DECIMAL(20,8),
NMONEY DECIMAL(20,8) DEFAULT 0,
NWAITUSENUMEBER_REF DECIMAL(20,8) DEFAULT 0,
NSALENUMBER DECIMAL(20,8) DEFAULT 0,
CONSTRAINT PK_JINGAOMX PRIMARY KEY (PK_JINGAOMX)
)
SELECT
    HN_JINGAOMX.pk_cumandoc,
    SUM((DOUBLE(HN_JINGAOMX.ndiscount)*DOUBLE(HN_JINGAOMX.nprice)))
FROM
    HN_INVRETURNTIME,
    HN_JINGAOMX
WHERE
    HN_INVRETURNTIME.pk_invbasdoc=HN_JINGAOMX.vgoodscode
AND (
    HN_JINGAOMX.treturtime BETWEEN HN_INVRETURNTIME.datebegin

```

```
AND HN_INVRETURNTIME.dateend)
AND NVL(HN_JINGAOMX.dr,0)=0
GROUP BY
    HN_JINGAOMX.pk_cumandoc
```

1.4 HP-UNIX 下 CHOWN 实例目录导致 DB2 实例 profile 失效

使用 ROOT 账户，执行 `chown -R db2inst1:db2grp1 /home/db2inst1` 命令后，随后使用命令 `su - db2inst1` 切入实例用户账户下，这时执行 `db2set`、`db2ilist` 等命令，会报如下错误：

```
$ db2set
DBI1306N  The instance profile is not defined.
Explanation:
The instance is not defined in the target machine registry.
User response:
Specify an existing instance name or create the required instance.
```

这时连接 DB2 失败，用命令行连接 `db2`，也会报如下错误：

```
$ db2 connect to PERFDB user db2inst1 using db2inst1
SQL30082N  Security processing failed with reason "42" ("ROOT CAPABILITY
REQUIRED").  SQLSTATE=08001
```

查询出错信息，如下所示，但是在 `db2 V9.76` 下面根本没有 `db2rfe` 命令：

```
$ db2 ? SQL30082N
To enable local client or server authentication for non-root
    installations, the system administrator must run the db2rfe
    script
```

解决办法是，进入 root 账号，重新刷新实例的 profile，故障消失：

```
# ./db2iupdt db2inst1
```

1.5 HP-UNIX 下 load 大表失败，db2osconf 推荐值偏小

本来修改 HP-UX 内核参数是根据：`db2osconf` 的建议值来设置：

```
***** System Resources *****
```

```
Physical RAM ==> 137259171840
```

```
Number of CPUS ==> 16
```

```
System Scale Factor ==> 8192.00
```

```
* Recommended HP-UX Kernel Parameters Based on this System *
```

```
nproc ==> 8208
```

```

maxuprc ==> 2736
max_thread_proc ==> 32767
maxfiles ==> 4096
maxfiles_lim ==> 65535
nflocks ==> 16384
ninode ==> 11469
msgmnb ==> 65535
msgtql ==> 6554
msgmni ==> 6554
semmns ==> 16385
semmni ==> 8192
semmnu ==> 8192
shmmax ==> 123533254656
shmmni ==> 8192
maxdsiz_64bit ==> 1900000000

```

这些参数都已经按建议值修改过，但是还报错。

按下面内核值联系 HP 进行修改，再执行 load 问题解决。

msgmnb (max # of bytes on a message queue, i.e max-msg-qbytes in Solaris 10+) = 2097152

msgmni (max # of message queues, i.e max-msg-ids in Solaris 10+) = 32767

msgtql (max # of message on all message queues, i.e max-msg-messages in Solaris 10+) = 262144

2. 运维和优化建议

2.1 缓冲池设计

目前 PERFDB 数据库共有 3 个数据表空间，分别为 NNC_DATA01、NNC_DATA02 和 NNC_DATA03，也有 3 个索引表空间，NNC_INDEX01、NNC_INDEX02 和 NNC_INDEX03。但是，它们竟然都共享一个默认的缓冲池 IBMDEFAULTBP，这个默认缓冲池大小为 60G，这不是一个好的缓冲池设计。正确的设计应该是每个表空间分配一个缓冲池，再根据冷热程度为其指定合理大小。如下表所示：

表空间	缓冲池	尺寸（仅是举例）
NNC_DATA01	BUF_DATA01	16G
NNC_INDEX01	BUF_INDEX01	4G
NNC_DATA02	BUF_DATA02	16G
NNC_INDEX02	BUF_INDEX02	4G
NNC_DATA03	BUF_DATA03	16G

NNC_INDEX03	BUF_INDEX03	4G
-------------	-------------	----

改动脚本如下所示：

```
ALTER TABLESPACE NNC_DATA01 BUFFERPOOL BUF_DATA01
ALTER TABLESPACE NNC_INDEX01 BUFFERPOOL BUF_INDEX01
ALTER TABLESPACE NNC_DATA02 BUFFERPOOL BUF_DATA02
ALTER TABLESPACE NNC_INDEX02 BUFFERPOOL BUF_INDEX02
ALTER TABLESPACE NNC_DATA03 BUFFERPOOL BUF_DATA03
ALTER TABLESPACE NNC_INDEX03 BUFFERPOOL BUF_INDEX03
```

2.2 遇到 SQL 性能问题，通过分析访问计划

```
connect to PERFDDB;
explain plan with snapshot for <query_text>;
!db2exfmt -d PERFDDB -1 -o query_plan.txt;
disconnect PERFDDB;
```

2.3 如何查看故障日志

一共是两个文件：db2diag.log 和 db2inst1.nfy。

例如，你可以使用命令“db2diag -time 2013-04-12 -level "Severe, Error"”查看 4 月 12 日以来的错误信息。

另外，为了避免日志文件过大，本次调整开启了日志轮转方式，总大小为 1000M，每个日志文件大约 100M 左右。

```
update dbm cfg using DIAGSIZE 1024;
```

2.4 在业务清淡时，例如晚上对大表手工 reorg 和 runstats 操作

例如对大表 BD_INVBASDOC 进行 reorg 和 runstats 操作：

```
reorg table BD_INVBASDOC;
reorg indexes all for table BD_INVBASDOC;
runstats on table BD_INVBASDOC with distribution on all columns and indexes all;
```

例如，目前 PERFDDB 中的大表如下：

```
DB2INST1.BD_INVBASDOC
DB2INST1.BD_INVMANDOC
DB2INST1.HN_BKARRIVAL_B
```

```
DB2INST1.HN_BKARRIVAL_D
DB2INST1.HN_JINGAOMX
DB2INST1.HN_JINGAORDERMX
DB2INST1.HN_HISTORYMX
DB2INST1.IC_GENERAL_B
DB2INST1.HN_UNONLINEBILL
```

[2.5 降低表空间高水位](#)

PERFDB 数据库中，6 个表空间都是 automatic storage，并且是 autoresize yes，可以使用下面的语句，降低表空间高水位：

```
ALTER TABLESPACE NNC_DATA01 REDUCE MAX;
ALTER TABLESPACE NNC_INDEX01 REDUCE MAX;
ALTER TABLESPACE NNC_DATA02 REDUCE MAX;
ALTER TABLESPACE NNC_INDEX02 REDUCE MAX;
ALTER TABLESPACE NNC_DATA03 REDUCE MAX;
ALTER TABLESPACE NNC_INDEX03 REDUCE MAX;
```

[2.6 如何查看死锁](#)

使用下面的命令，可以查看当前数据库是否存在死锁：

```
db2evmon -db PERFDB -evm db2detaildeadlock > deadlock.txt
```

用编辑器打开 deadlock.txt，随后从文件尾部向上搜索"update"，重点检查 update 语句导致的死锁，随后结合应用调整。

[2.7 如何查看最耗资源的 SQL 语句](#)

下面的语句取出前 10 大最耗资源的 SQL 语句：

```
select (TOTAL_USR_CPU_TIME + TOTAL_SYS_CPU_TIME) as totaltime, NUM_EXECUTIONS,
SORT_OVERFLOWS, TOTAL_SORT_TIME, STMT_TEXT from sysibmadm.snapdyn_sql order by
totaltime desc fetch first 10 rows only
```