ORACLE “In-Memory” baza

bez “In-Memory” opcije

Josip Pojatina
mStart d.o.o. (Agrokor ICT)
josip.pojatina@mStart.hr

Sadržaj

- Uvod
- HugePages / Large Page
- LOCK_SGA
- DEFAULT / KEEP / RECYCLE pool
- CACHE / NOCACHE opcija
- Query & function result cache
- RAM disk
- Demo
- Pitanja i odgovori

O tvrtki mStart

- Agrokor ICT 1.7.2010. promijenio naziv u mStart d.o.o.
- Djeluje kao samostalni subjekt unutar Agrokor koncerna s ciljem pružanja podrške za 100+ kompanija unutar Agrokor grupacije

O autoru

- Arhitektura/dizajn/optimizacija/razvoj/administracija
- 15+ godina iskustva s Oracle RDBMS-om
- 10+ godina optimizacija velikih sustava baziranih na Oracle tehnologiji (optimizacija Oracle RDBMS-a, Web Logic servera i Oracle Service Bus (OSB), Java/JRockit JVM)
- Red Hat / Oracle Linux, IBM AIX
- specijalnost Oracle CBO (SQL engine), PL/SQL i Java store procedure
- Oracle Retail
- Oracle eBS

Uvod

- Trend memorijskih baza je započeo sa SAP HANA
- Oracle je otprije imao Times Ten bazu
- Oracle “In Memory” opcija dostupna od drugog PS-a 12c RDBMS-a (25.7.2014)
- Memorija (RAM) je najbrži način pristupa i manipuliranjem podacima
- Memorijske baze ubrzavaju Analitičke (OLAP), Data Warehouse, Reporting (DSS) pa čak i OLTP sustave

Uvod

- Oracle “In Memory” opcija je potpuno (za razliku od Times Ten-a koji je zaseban proizvod) integrirana u Oracle RDBMS i ubrzava upite dohvaćanjem podataka iz memorije umjesto sa diska
- Druga opcija dobivanja top performansi je upgrade storage sustava (“flash storage”) kakve nude mnogi vendori
- Oba rješenja za ubrzavanje performansi imaju zajednički problem:

Uvod

Uvod

- Kombinacijom postojećih funkcionalnosti/tehnologije, moguće je postići “In-Memory” bazu bez dodatnih ulaganja
- Iako neke funkcionalnosti u besplatnom rješenju nedostaju (memorijsko komprimiranje na nivou stupaca), ubrzanje je impresivno
- Pojedine metode za ručnu izradu memorijske baze su stare više od 10 godina, ali su primjenjene u novim okolnostima (niske cijene memorija)
- Oracle, nakon što predstavi neku novu funkcionalnost, ne dokumentira upotrebu određene funkcionalnosti u novim uvjetima

Uvod

- Znanja potrebna za ručnu izradu “In-Memory” baze:
  - Storage subsystem
  - Operacijski sustav (Linux/Unix, Windows)
  - Oracle DBA
  - Oracle Development (SQL, PL/SQL, Java)
  - C / C++ (poželjno)

HugePages / Large Page

- Funkcionalnost dostupna na svim OS-ima (najlakše se konfigurira na Linux-u – od 2.6 verzije integriran u kernel)
- Smanjuje potrošnju CPU-a za upravljanje memorijom (umjesto 4Kb, najmanja jedinica aociranja memorije je bitno veća npr. 2Mb blokovi)
- AMM (Automatic Memory Management) nije kompatibilan sa HugePages funkcionalnošću (ASMM i manualno konfiguriranje memorije su jedino podržani)

HugePages / Large Page

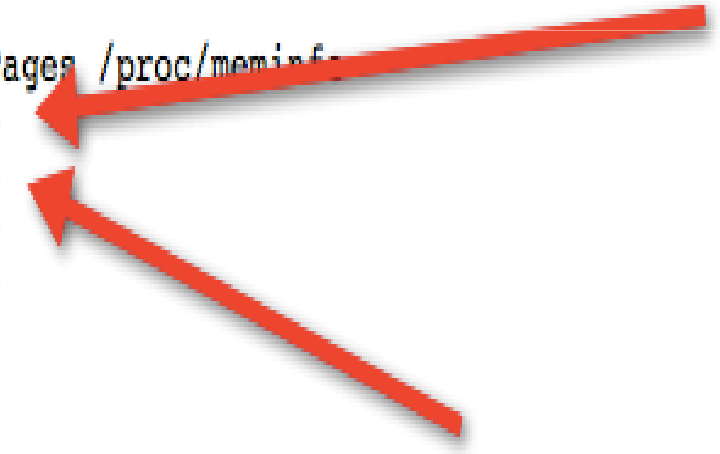
- HugePages nisu “swappable” (nema page-in/page-out-a), pa je time osigurano da je SGA uvijek u RAM-u pinan (lock_sga = true)
- smanjen je page table i page lookup overhead
- ubrzanje performansi memorije
- manje potrebne memorije za kernel OS-a
- poboljšanje se osjeća već na serverima s više od 8 Gb RAM-a

HugePages / Large Page

- Provjera da Oracle koristi large pages

```
--Dokaz da Oracle koristi large pages je da HugePages_Free < HugePages_Total
grep HugePages /proc/meminfo

root@jp-oel:~>grep HugePages /proc/meminfo
HugePages_Total:     255
HugePages_Free:       83
HugePages_Rsvd:       79
HugePages_Surp:        0
```



HugePages / Large Page

- Pregled shared memory segmenata

```
--Izlist share memory segmenata
root@jp-oel:~>ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 131072     admjpojati 600        393216     2          dest
0x00000000 163841     admjpojati 600        393216     2          dest
0x00000000 196610     admjpojati 600        393216     2          dest
0x00000000 229379     admjpojati 600        393216     2          dest
0x00000000 262148     admjpojati 600        393216     2          dest
0x00000000 294917     admjpojati 600        393216     2          dest
0x00000000 327686     admjpojati 600        393216     2          dest
0x00000000 360455     admjpojati 600        393216     2          dest
0x00000000 393224     admjpojati 600        393216     2          dest
0x00000000 425993     admjpojati 600        393216     2          dest
0x00000000 458762     admjpojati 600        393216     2          dest
0x00000000 491531     admjpojati 600        393216     2          dest
0x00000000 557068     oracle     660        8388608    23
0x00000000 589837     oracle     660        515899392  23
0x4ceb1850 622606     oracle     660        2097152    23
```

LOCK_SGA parametar

- Osigurava da se SGA nalazi u glavnoj memoriji (zaključava SGA u RAM) i uvijek top performanse
- Sprečava pojavu swap-a
- Nije kompatibilan sa MEMORY_TARGET i MEMORY_MAX_TARGET parametrima

DEFAULT/KEEP/RECYCLE pool

- DEFAULT cache određen db_cache_size parametrom i uvijek je kreiran
- Koristi LRU algoritam za izbacivanje blokova segmenata iz memorije
- Iako prisutan dugi niz godina, zbog promijenjenih okolnosti (pada cijene memorije), postaje ponovno aktualan, jer se povećanjem DB_CACHE_SIZE parametra može cashirati veći broj segmenata/blokova u memoriji

DEFAULT/KEEP/RECYCLE pool

- KEEP i RECYCLE cache omogućuju DBA veću kontrolu nad cashiranjem objekata
- Koriste isti algoritam izbacivanja blokova iz cache-a kao i DEFAULT cache
- Segmenti koji su “warm” možemo staviti u KEEP pool da ne budu izbačeni
- Segmente za koje nije bitno da su u cache-u (npr. LOB-ovi) treba staviti u RECYCLE pool (inače bi išli u DEFAULT pool i izbacili bitne segmente – tablice/indekse)

DEFAULT/KEEP/RECYCLE pool

- DB_KEEP_CACHE_SIZE (ex. buffer_pool keep)
- create/alter table/index ... storage (buffer_pool keep)

- DB_RECYCLE_CACHE_SIZE (ex. buffer_pool recycle)
- create/alter table/index ... storage (buffer_pool recycle)
- alter system set db_recycle_cache_size=5m scope=both;

CACHE / NOCACHE opcija

- create/alter table/index ... cache
- Ako je tablica kreirana s cache opcijom, to ne znači da će dotična tablica biti u memoriji.
- Cache opcija je samo signal da se blokovi tablice s cache opcijom tretiraju drugačije (kod full table scan-a umjesto da blokovi dođu na LRU dio liste DEFAULT pool-a, oni će doći na MRU – most recent used dio liste, čime se smanjuje šansa da će biti izbačeni iz cache-a)

Query & function result cache

- sprema rezultat SQL upita ili funkcije u memoriji kako bi se spremljene vrijednosti mogle ponovno iskoristiti prilikom slijedećih upita (funkcijskih poziva)
- result_cache_max_size parametar određuje max. veličinu memorije namijenjenu pohrani SQL & function result cache-a
- ALTER SYSTEM SET result_cache_max_size = 100M SCOPE = BOTH;
- result_cache_max_result određuje max. postotak cache-a kojeg može zauzeti jedno cashiranje (resultset)

Query & function result cache

```
select /*+ result_cache */ * from scott.emp;

Plan hash value: 3956160932

| Id | Operation           | Name                    | Rows | Bytes | Cost (%CPU)| Time     |
|  0 | SELECT STATEMENT    |                         |      |       |     3 (100)|          |
|  1 |  RESULT CACHE       | 81bhwtsjthkar4ww8pgffp20su |      |       |            |          |
|  2 |   table access full | emp                     |   14 |   532 |     3   (0)| 00:00:01 |
```

```
with subq as (select /*+ result_cache */ * from scott.emp)
select * from scott.dept where deptno in (select deptno from subq);

Plan hash value: 2862707078

| Id | Operation                    | Name                    | Rows | Bytes | Cost (%CPU)| Time     |
|  0 | SELECT STATEMENT             |                         |      |       |     6 (100)|          |
|  1 |  MERGE JOIN SEMI             |                         |    3 |    99 |     6  (17)| 00:00:01 |
|  2 |   TABLE ACCESS BY INDEX ROWID| DEPT                    |    4 |    80 |     2   (0)| 00:00:01 |
|  3 |    INDEX FULL SCAN           | PK_DEPT                 |    4 |       |     1   (0)| 00:00:01 |
|* 4 |   SORT UNIQUE                |                         |   14 |   182 |     4  (25)| 00:00:01 |
|  5 |    VIEW                      | VW_NSO_1                |   14 |   182 |     3   (0)| 00:00:01 |
|  6 |     VIEW                     |                         |   14 |    28 |     3   (0)| 00:00:01 |
|  7 |      RESULT CACHE            | 81bhwtsjthkar4ww8pgffp20su |      |       |            |          |
|  8 |       TABLE ACCESS FULL      | EMP                     |   14 |   532 |     3   (0)| 00:00:01 |
```

RAM disk

- memorijski disk
- Postoje dva tipa: ramfs i tmpfs
- ramfs
  - ne može mu se ograničiti veličina
  - uvijek koristi RAM memoriju
- tmpfs
  - može mu se ograničiti veličina (kao kod običnog diska)
  - može pored memorije koristiti i swap space

RAM disk

- Testna tablica: cca. 4 mil. redaka, 448Mb
- Kreirana na bazi dba_objects view-a
- Rezultati testova provedeni na Virtualbox-u na host računalu s 4Gb RAM-a i jednim SSD diskom 256 Gb
- Oracle 11.2.0.3 EE baza

RAM disk –Test 1

```
SYS@orcl> select * from scott.objects_disk where object_type = 'TABLE' and object_name like 'I%';

1378 rows selected.

Elapsed: 00:00:03.45

Statistics
----------------------------------------------------------
    1  recursive calls
    1  db block gets
   56849  consistent gets
   56793  physical reads
    0  redo size
   63716  bytes sent via SQL*Net to client
  666  bytes received via SQL*Net from client
   15  SQL*Net roundtrips to/from client
    0  sorts (memory)
    0  sorts (disk)
    1378  rows processed
```

RAM disk –Test 1

```
SYS@orcl> select * from scott.objects_ram where object_type = 'TABLE' and object_name like 'I%';

1378 rows selected.

Elapsed: 00:00:01.19

Statistics
----------------------------------------------------------
   21  recursive calls
    0  db block gets
   56863  consistent gets
   56956  physical reads
    0  redo size
   63716  bytes sent via SQL*Net to client
  666  bytes received via SQL*Net from client
   15  SQL*Net roundtrips to/from client
    0  sorts (memory)
    0  sorts (disk)
    1378  rows processed
```

RAM disk –Test 2

```
SYS@orcl> select * from scott.objects_disk;

3986395 rows selected.

Elapsed: 00:01:19.00

Statistics
----------------------------------------------------------
      1  recursive calls
      1  db block gets
     96120  consistent gets
     56793  physical reads
      0  redo size
  414902516  bytes sent via SQL*Net to client
     439016  bytes received via SQL*Net from client
      39865  SQL*Net roundtrips to/from client
      0  sorts (memory)
      0  sorts (disk)
    3986395  rows processed
```

RAM disk –Test 2

```
SYS@orcl> select * from scott.objects_ram;

3986395 rows selected.

Elapsed: 00:00:32.95

Statistics
----------------------------------------------------------
         5  recursive calls
         0  db block gets
     96104  consistent gets
     56729  physical reads
         0  redo size
 414902516  bytes sent via SQL*Net to client
    439016  bytes received via SQL*Net from client
     39865  SQL*Net roundtrips to/from client
         0  sorts (memory)
         0  sorts (disk)
   3986395  rows processed
```

RAM disk –Test 3

```
SYS@orcl> select count(*) from scott.objects_disk;

Elapsed: 00:00:03.31

Statistics
----------------------------------------------------------
      1  recursive calls
      1  db block gets
          56832  consistent gets
          56793  physical reads
      0  redo size
    529  bytes sent via SQL*Net to client
    523  bytes received via SQL*Net from client
      2  SQL*Net roundtrips to/from client
      0  sorts (memory)
      0  sorts (disk)
      1  rows processed
```

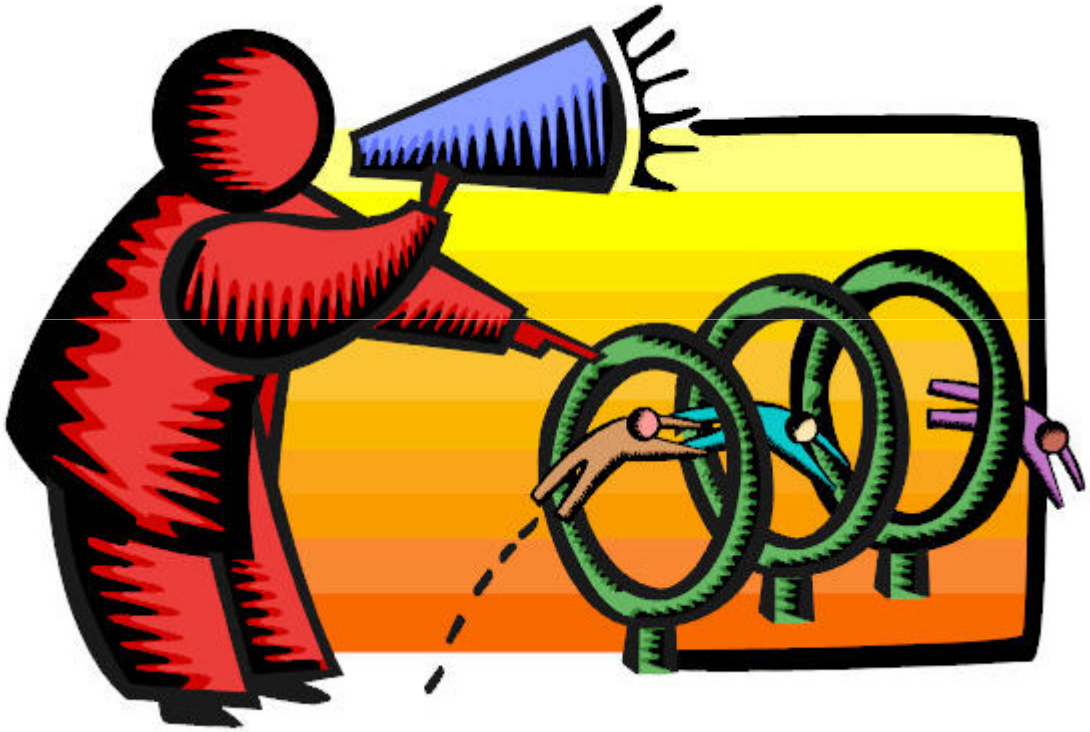
RAM disk –Test 3

```
SYS@orcl> select count(*) from scott.objects_ram;

Elapsed: 00:00:00.27

Statistics
----------------------------------------------------------
     4  recursive calls
     0  db block gets
         56817  consistent gets
         56729  physical reads
     0  redo size
   529  bytes sent via SQL*Net to client
   523  bytes received via SQL*Net from client
     2  SQL*Net roundtrips to/from client
     0  sorts (memory)
     0  sorts (disk)
     1  rows processed
```

Demo



Optimizacija SQL-a na Oracle Support način