

# DBA TIPS AND TRICKS LOCKING

Damir Vadas

<http://damir-vadas.blogspot.com>

TEB Informatika d.o.o.

[www.teb-informatika.hr](http://www.teb-informatika.hr)

HROUG

Rovinj, 10/2010

# Koncept

- O zaključavanju (malo teorije - samo kratko)
  - Razine zaključavanja
  - Modovi zaključavanja
  - Tipovi zaključavanja
- Prezentacija mogućnosti rješenja
  - Single instance 11gR2
  - RAC instance 10GR2
  - PLSQL code locking
- Credits
- Q and A

# USPJEŠAN DBA

**Onaj koji svojim radom uspije  
napraviti balans između fire fighter-a  
i proactive DBA.**

Ova demonstracija pokušat će  
pokazati vezu između oboje.

# O zaključavanju (lock)

- ▣ Lock-ovi su Oracle mehanizam koji sprečava destruktivnu interakciju među transakcijama (nad istim resursom)
  - Data concurrency - korisnici mogu istovremeno pristupati podacima
  - Data consistency - korisnici vide konzistentnu sliku podataka:
    - ▣ Sve promjene iz svoje transakcije
    - ▣ Obzirom na „isolation level“, promjene drugih transakcija, po defaultu samo „commit-irane“.

# O zaključavanju

- ▣ Izolacijske razine („isolation levels”)
  - Read committed (default) (transaction based)
  - Serializable Transactions (query based)
  - Read-only
- ▣ U razmatranju samo default-ni isolation level.
  
- ▣ Oracle podržava dva moda zaključavanja (locking modes):
  1. Exclusive lock mode (X)
  2. Share lock mode (S).

# Exclusive Lock Mode

- ❑ Sprečava da se asocirani resurs dijeli s drugima.
- ❑ Ovaj mod je primjerice kod ažuriranja podataka.
- ❑ Prva transakcija koja postavi ovaj mod može jedina alterirati resurs.

# Share Lock Mode

- ▣ Zavisno od operacija, resurs se dijeli.
- ▣ Više transakcija može postaviti share lock nad istim resursom
- ▣ Njime se sprečava pristup bilo kakvom tipu ažuriranja (ono treba exclusive lock mod).

# Tipovi zaključavanja

- ▣ Oracle tipovi zaključavanja
  - DML locks (data locks)
  - DDL locks (dictionary locks)
  - Oracle Internal Locks/Latches
  - Oracle Distributed Locks
  - Oracle Parallel Cache Management Locks



# DML zaključavanja

- ▣ Zaključavanja nad podacima
- ▣ Čuvaju integritet podataka
- ▣ Dva tipa DML zaključavanja:
  - Razina retka (row level locking)
  - Razina tablice (table level locking)

# Row level locking TX

- ▣ Najniža razina zaključavanja
- ▣ Ova razina osigurava zaključavanje pojedinog retka tablice.
- ▣ Svi DML lock-ovi su automatski row level lock.
- ▣ Oracle ne eskalira lock kada ima row level lock.
- ▣ Zaključani reci tablice mogu se ažurirati samo iz sesije koja je lock postavila
- ▣ Zaključani reci otpuštaju se samo iz sesije koja je lock postavila
- ▣ Modificirani redak je uvijek zaključan ekskluzivno
- ▣ Ažuriranje retka s FK traži postavljanje table lock nad korespondentnom FK tablicom

# Row level locking TX

- Svi reci dostupni su za čitanje ostalim sesijama s time da oni vide sliku retka koja je bila prije nego je ažuriran.

Ovo je poznato kao „consistent read”.

- Ostali reci su slobodni za ažuriranje
- Proces zaključavanja:
  1. Prvo se postavlja DML lock na redak. Ovime se sprječava da ostali procesi počnu ažurirati retke.
  2. Zatim se postavlja DDL (Data Dictionary Language) lock nad cijelom tablicom koji sprečava strukturne promjene (primjerice alter, drop).
  3. Oba lock-a otpuštaju se na kraju transakcije (commit, rollback, kill session)

# Table level locking TM

- ▣ Omogućuje DML i sprečava DDL zaključavanja.
- ▣ Samo sesija koja je postavila table level lock može ažurirati lock ili bilo koji redak tablice (niti jedna druga sesija ne može ažurirati ijedan redak)
- ▣ Svi reci tablice slobodni su za čitanje ostalim sesijama
- ▣ Postoji pet modova za TM:
  - row share (RS),
  - row exclusive (RX),
  - share (S),
  - share row exclusive (SRX),
  - exclusive (X).

# Table level locking TM

- ▣ Kompatibilnost table lock-a (modovi koji se mogu postaviti) i primjer SQL naredbe:

SQL	Lock Mode	RS	RX	S	SRX	X	OK?
SELECT		Y	Y	Y	Y	Y	Y
INSERT	RX	Y	Y	N	N	N	N
UPDATE	RX	Y*	Y*	N	N	N	N
DELETE	RX	Y*	Y*	N	N	N	N
SELECT.. FOR UPDATE OF..	RS	Y*	Y*	Y*	Y*	N	N
LOCK TABLE <table_name>							
IN ROW SHARE MODE	RS	Y	Y	Y	Y	N	N
IN ROW EXCLUSIVE MODE	RX	Y	Y	N	N	N	N
IN SHARE MODE	S	Y	N	Y	N	N	N
IN SHARE ROW EXCLUSIVE MODE	SRX	Y	N	N	N	N	N
IN EXCLUSIVE MODE	X	N	N	N	N	N	N

\* Čeka ukoliko druga transakcija ima lock

# Pregled što ima sada

- ▣ Sva rješenja (koja sam našao) imaju jednu odliku:
  - Koriste DBA\_\*, v\$\* view-ove čime podržavaju samo jednu instancu (nema RAC podrške) .
- ▣ Oracle je od 6.x inačice kreirao utllockt.sql skriptu (\$ORACLE\_HOME/rdbms/admin) koja prilično lijepo daje rezultat za single instance

# Single instance 11gR2

```
SQL> @my_sid
```

USER	INSTANCE	SID	SERIAL#
USER1	1 xe	19	342

```
SQL> update user1.qqq set c1='aa' where c1='3';
```

1 row updated.

```
SQL> @my_sid
```

USER	INSTANCE	SID	SERIAL#
USER2	1 xe	22	138

```
SQL> delete user1.qqq;
```

# Single instance 11gR2

Oracle version: 11.2.0.1.0 (11.1.0.0.0)

Blocker Module	Inst	SID	Serial	[sec]	Lock Type/Req. lock	Status
1.USER1 SQL*Plus	1	19	342	57	Transaction	INACTIVE
USER2 SQL*Plus	1	22	138	21	Exclusive	ACTIVE

```

----- Blocking records information -----
USER1 (1 '19,342')
      USER2 (1 22,138)          21 sec          Exclusive          ACTIVE          SQL*Plus
      SELECT C1 FROM USER1.QQQ WHERE rowid = 'AAAVsLAAGAAAAe2AAC' ;
      (delete user1.qqq)
  
```

To kill first from the list, perform:

```

ALTER SYSTEM DISCONNECT SESSION '19,342,@1' IMMEDIATE;
ALTER SYSTEM KILL SESSION '19,342,@1' IMMEDIATE;
  
```

PL/SQL procedure successfully completed.

```

SQL> SELECT C1 FROM USER1.QQQ WHERE rowid = 'AAAVsLAAGAAAAe2AAC' ;
  
```

C1

-----

3



# RAC instance 10gR2

USER1@hact1> @my\_sid

USER	INSTANCE	SID	SERIAL#
USER1	1 HACT1	113	19735

USER1@hact1> update user1.qqq set c1='1';

3 rows updated.

USER1@hact1>.

USER2@hact2> @my\_sid

USER	INSTANCE	SID	SERIAL#
USER2	2 HACT2	118	11178

USER2@hact2> update user1.qqq set c1='2';

# RAC instance 10gR2

Blocker	Inst	SID	Serial	[sec]	Lock Type/Req. lock	Status	Module
1.USER1	1	113	19735	78	Transaction	INACTIVE	SQL*Plus
USER2	2	118	11178	28	Exclusive	ACTIVE	SQL*Plus

----- Blocking records information -----

```

USER1 (1 '113,19735')
  USER2 (2 118,11178)      31 sec      Exclusive      ACTIVE      SQL*Plus
      SELECT C1 FROM USER1.QQQ WHERE rowid = 'AAAtNcAAJAACckFAAA' ;
      ( update user1.qqq set c1='2')
  
```

To kill first from the list, perform:

NON RAC (or RAC logged on that node):

```

-----
ALTER SYSTEM DISCONNECT SESSION '113,19735' IMMEDIATE;
ALTER SYSTEM KILL          SESSION '113,19735' IMMEDIATE;
  
```

RAC (logged on any node) :

```

-----
declare
  v_job binary_integer;
begin
  DBMS_JOB.submit ( job      =>v_job
,what      =>'begin execute immediate 'ALTER SYSTEM
DISCONNECT SESSION '113,19735'' IMMEDIATE''; end; '
,instance=>1
);
  commit;
end;
/
  
```

PL/SQL procedure successfully completed.

```

TOOLS@hact3> SELECT C1 FROM USER1.QQQ WHERE rowid = 'AAAtNcAAJAACckFAAA' ;
  
```

```

C1
-----
1
  
```

# Prezentacija mogućnosti

- ▣ Detektirati „žive“ blocking session-e\*

Blocker	Inst	SID	Serial	[sec]	Lock Type	Status	Module
1. ANAG	1	282	6661	7981	Transaction	<b>KILLED</b>	?
ZLSA	2	372	126	1128	Exclusive	ACTIVE	?
SLPE	2	368	48	6836	Exclusive	ACTIVE	?
DRI2	3	237	232	6727	Exclusive	ACTIVE	?
2. DRI2	3	237	232	6724	Transaction	<b>KILLED</b>	?
DRIA	2	355	195	216	Exclusive	ACTIVE	?

All valid blocker sessions already killed. Try kill with OS level command!

Dakle, rješenje ne prikazuje „disconnect session“ jer je proces „marked for kill“ i čeka da ga PMON očisti što može potrajati! Lock je i dalje tu!

OS naredbom (**kill**, **ora\_kill**) ubiti takav proces.

# Prezentacija live-preduvjeti

```
SQL> show parameter smtp
```

NAME	TYPE	VALUE
smtp_out_server	string	mail.iskon.hr

```
SQL>
```

```
-- za tools usera (koji će biti vlasnik procedure):  
grant execute on SYS.UTL_MAIL to tools;
```

```
-- ACL za 11g baze
```

```
Rješenje ACL problema kod slanja mail-a (11g baza)
```

<http://blog.whitehorses.nl/2010/03/17/oracle-11g-access-control-list-and-ora-24247/>

```
-- 2 usera i tablice sa tri reda:
```

```
drop user user1 cascade;  
drop user user2 cascade;  
create user user1 identified by qw;  
create user user2 identified by qw;  
grant connect, resource to user1;  
grant connect to user2;
```

```
grant select, insert, delete, update, alter, index on user1.qqq to user2;
```

# Prezentacija live-preduvjeti

```
create table user1.qqq  
(c1 varchar2 (10));
```

```
grant select, insert, delete, update, alter,  
index on user1.qqq to user2;
```

```
SQL> select * from user1.qqq;
```

```
C1
```

```
-----
```

```
1
```

```
2
```

```
3
```

```
SQL>
```

# Prezentacija live

- ▣ Session 1 (user1)
  - update user1.qqq  
set c1='22'  
where c1='2';
- ▣ Session 2 (user2)
  - update user1.qqq set c1='tt';
- ▣ Session 3 (damirv)
  - sta sb2 FALSE (console)
  - sta sb2 TRUE (console + mail)
  - sta sbr (console)
  - sta sb (console + mail)

# Prezentacija live 2

- ▣ Session 1 (user1)
  - `update user1.qqq  
set c1='22'  
where c1='2';`
- ▣ Session 2 (user2)
  - `update user1.qqq set c1='tt';`
- ▣ Session 3
  - `delete user1.qqq;`
- ▣ Session 3 (damirv)
  - `sta sb` (console + mail)

# DDL lock

- ▣ Rijedak ali vrlo „nezgodan” jer se teško detektira uzrok (SID koji ga drži)
- ▣ Problem nastaje kada se treba alterirati procedura/ fnc/ package, a useri drže share lock-ove nad njima
- ▣ U11g DDL\_LOCKOUT\_TIMEOUT
- ▣ Sreća je da Oracle definira red (queue) i kada jednom transakcija uđe u njega, čeka svoj red.
- ▣ Nezgoda u svemu jest da ovaj tip lock-a može zakočiti puno objekata baze\*



# Prezentacija live DDL

```
CREATE OR REPLACE PROCEDURE user1.demo_wait (p_sec IN INTEGER)
IS
BEGIN
    DBMS_LOCK.SLEEP (p_sec);
END;
/
```

```
grant execute on user1. demo_wait to public;
```

Session 1

```
SQL> exec user1.demo_wait (3600);
```

Session 2

```
SQL> exec user1.demo_wait (3600);
```

Session 3

```
SQL> alter procedure user1.demo_wait compile; -> block
```

# Prezentacija live DDL

```
SQL> sta sbc
```

```
Oracle version: 11.2.0.1.0 (11.1.0.0.0)
```

```
-----  
Blocked object:USER1.DEMO_WAIT (Share)
```

```
On instance: "1" perform:
```

```
alter system disconnect session '139,652' immediate ;
```

```
-----  
Blocked object:USER1.DEMO_WAIT (Share)
```

```
On instance: "1" perform:
```

```
alter system disconnect session '149,17' immediate ;
```

```
-----  
PL/SQL procedure successfully completed.
```

```
SQL>
```

# Prezentacija live DDL

- ▣ Zakočiti puno objekata baze\*

Session 1

```
exec user1.demo_wait (3600);
```

Session 2

sta my\_sid -> **da saznam SID koji će imati probleme**

```
alter procedure user1.demo_wait compile; -> block
```

Session 3

```
BEGIN
```

```
  update user1.qqq set c1='1' where c1='2';
```

```
  user1.demo_wait (5);
```

```
END;
```

```
/
```

```
-> block !!! ???
```

```
SQL> update user1.qqq set c1='1' where c1='2'; -> block!
```

- ▣ Zanimljivo je primijetiti SESSION 3 block!

# Prezentacija mogućnosti

- ▣ Rješenje omogućuje:
  - Detektira blocking/waiter session-e za DML i DDL lock-ove
  - Detektira waiter SQL-ove
  - Detektira retke koji su zaključani (ako može)
  - Create kill statement za 10/11g version RAC i single instance (**fire fighter**)
  - Detektirati „žive“ blocking sessione\*
  - Izvršavat se iz job-a i slati mail kada se lock pojavi (**proactive**)

# Credits

- ▣ Oracle Metalink:
  - Detecting Blocking Sessions in RAC and non-RAC (Enqueue Locks) Environments [ID 398519.1]
  - FAQ about Detecting and Resolving Locking Conflicts [ID 15476.1]
  - How to know which row is locked by what user [ID 132629.1]
- ▣ Google
- ▣ Moje 10 godišnje iskustvo u Oracle-u

# Q and A