"Real Time" Fragmentation

Managing Old Data

Jack Parker
Jack.parker4@verizon.net

Aging off old data

- Banks retain data for 7 years
- Marketing data typically has 3 years worth of data
- “Rolling” data

Conceptually…

- Archive (perhaps)
- Delete

Worst case I've ever seen

- SELECT … FROM table
- If date_column < delete_date
- DELETE where current

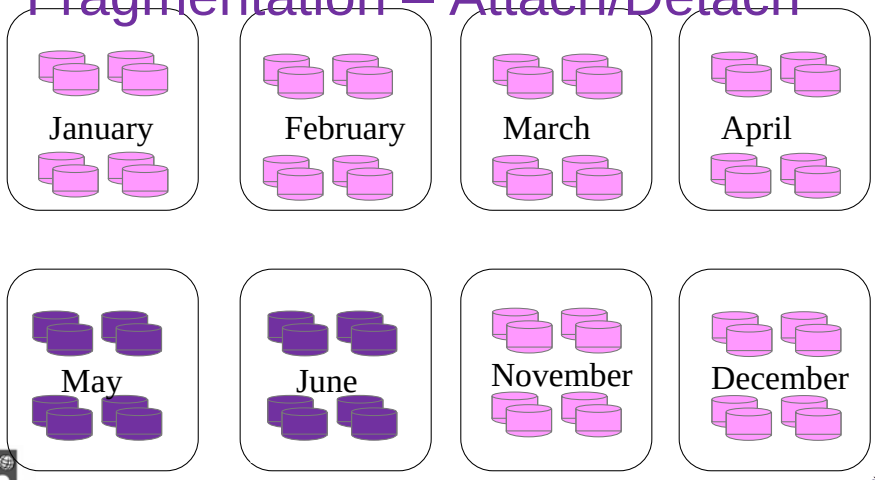
Also not so good…

- DELETE FROM table WHERE date_col < delete_date
- Slow
- Heavy I/O
- Potential Long transaction
- Leaves holes in the table – which you have to reorg to recover

Nibble Strategy

- Select first 10,000 rowid from table into temp t1 with no log;
- Delete from table where rowid in (select rowid from t1);
- Rinse and repeat
- Pre-create t1 for better performance.
- Still slow etc.

Fragmentation – Attach/Detach



Advantages

- Fast – about 1 second
- No messy holes in the data
- No I/O
- Parallelism
  - (that's another talk)
- Fragment elimination

Added advantage

- The new partition is an added extent to the table
- Its size is controlled by systables.extsize
- You can control the size with an ALTER TABLE … NEXT SIZE – (for the second extent)

Warning

- If you fragment by an ‘update_date’
- every time the row is updated…
- it will migrate from one fragment to another.

V10

- Has a partition and a fragment
- Partitions can reside in the same dbspace

Syntax

- Create table foo (
- …)
- Fragment by expression
- Partition xxxx_YYMMDD Date_col=1/1/2001 in dbspc1,
- Partition xxxx_YYMMDD Date_col=1/2/2001 in dbspc1,
- Remainder Partition xxxx_rmdr in dbspc1

Syntax

- ALTER FRAGMENT ON TABLE foo DROP partition/fragment
- … well not exactly
- DROP partition drops the partition from under your data
- The data then has to migrate to another partition

Syntax (2)

- ALTER FRAGMENT ON TABLE foo DETACH partition new_table;
- DROP new_table;
- ALTER FRAGMENT ON TABLE foo ATTACH partition new_partition (expression) IN dbspace;

Performance note

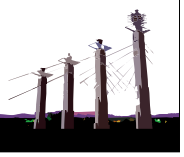
- Expressions are evaluated from top down
  - Date_col < 2/1/2000 and Date_col >= 1/1/2000
  - Date_col < 3/1/2000 and Date_col >= 2/1/2000
  - Date_col < 4/1/2000 and Date_col >= 3/1/2000
- Put current/future partitions at the beginning
  - Date_col >= 3/1/2000 and date_col < 4/1/2000
  - Date_col >= 2/1/2000 and date_col < 3/1/2000
  - Date_col >= 1/1/2000 and date_col < 2/1/2000
- ALTER FRAGMENT ….. ATTACH partition (expression) IN dbspace BEFORE other_partition

Performance note - Correction

- Please excuse the tip provided on the previous slide, the presenter did not know what he was talking about. It is expensive to insert a new fragment before existing fragments.

Other fun things you can do with fragmentation

- Detach
  - (Splitter process)
- Attach a pre-built table
  - Schema must match
  - Including defaults, NOT NULL, indexes



Issues

- ROWID
- Foreign/Primary Keys

ROWIDs

- Are not fragmented by date
- Means that the ROWID “index” has to be rebuilt each time
  - expensive

Foreign Keys

- Cannot detach a partition from the parent table
- Even if the tables are empty
- Primary Keys/Unique Indexes must include the column used in the fragmentation expression.
- Build indices underneath FK constraints

Turn off constraints

- SELECT t.tabname, c.constrname
  FROM systables t, sysconstraints c
  WHERE t.tabid = c.tabid
  AND constrtype = ‘R’ (also ‘P’)
- Save them off first, then drop them.
- After you are done, re-add them.
- Expensive.

Real World

- Windows
- SPL
- OS Shell
- Partition period is each day

More Real World

- Infrastructure Product
- 24x7 - No maintenance window
  - (Remember “Real Time”?)
- Variable and unknown retention period
- ... Measured in days

SPL

- There is no dynamic SQL
- Which we need to pass in table names and other parameters.
- Options are
  - esql/c, 4gl,
  - BATCH (cygwin/Rexx),
  - SPL to generate SPL
  - Exec datablade

BATch Issues

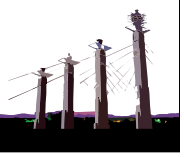
- … From the people that brought you edlin
- Foo.bat Parameter1 “parm2a parm2b” “parm3a, parm3b” parm4a, parm4b
- Seen as 7 parameters

Handling variable parameters

- set tab=%1
- set frag=%2
- set part=%3
- set dbspace=%4
- SHIFT
- SHIFT
- SHIFT
- SHIFT
- set expstrg=
- set SEP=
- :loop
- if "%1" == "" GOTO END
- set expstrg=%expstrg%%SEP%%1
- set SEP=,
- SHIFT
- GOTO LOOP
- :END

BATch Issues (1)
Capturing the Error Message

```
dbaccess -a <database> tmp.sql 1>stdout 2>stderr
set RETVAL=%ERRORLEVEL%
IF %RETVAL%==0 GOTO END2
findstr ":" stderr > ldfile
echo load from ldfile insert into log(message); >
   tmp.sql
dbaccess <database> tmp.sql
```



Capturing return codes with Batch (2)

- Output in stderr
- Echo set variable= > tmp1.bat
- Copy tmp1.bat + stderr tmp2.bat
- Execute tmp2.bat
- Variable is now set

- Hackery

Capturing Return code (3)

```
for /f "tokens=1 delims=:" %%i in (ldfile) do set
   RETVAL=%%i
:END2
del stderr
del stdout
del tmp.sql
del ldfile
EXIT /B %RETVAL%
   /B closes window
```

In Shell

```
retcode=`grep ":" stderr |
  head –1 | awk –F":" '{print $1}'`
exit $retcode
```

Cleaner method

- Use Paul Brown's Exec datablade (IIUG)
- Allows dynamic SQL within a stored procedure.

Identifying Fragments/Partitions to drop

SELECT tabname, b.partition, **c.npused,** a.tabid

FROM sysfragments b, **sysmaster:sysptnhdr c,** systables a

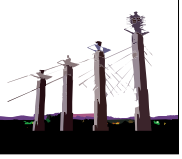
WHERE **indexname is null**

AND a.tabid=b.tabid

AND b.partn=c.partnum

AND **c.nrows=0**

AND **b.partition[6,9]!='rmdr'**



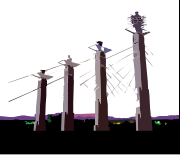
Of interest…

- Sysptnhdr has actual data (doesn't rely on statistics)
- Nrows tells you if the fragment is empty
- Npused is the number of pages used
- Sysfragments.Indexname – sysfragments keeps index fragments in there too – linked to their index.
- Partition[6,9] != rmdr
  - Our convention xxxx_YYMMDD or xxxx_rmdr

```
-- Have to ensure that we are not grabbing the last
   two.
AND evalpos < (select max(evalpos) from
   sysfragments d
                WHERE b.tabid=d.tabid
                  AND indexname IS NULL)
    AND date(partition[8,9] || '/' || partition[10,11] ||
   '/' || partition[6,7])<(today)
AND evalpos != 0
```

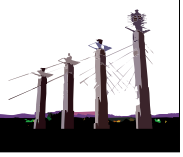
Sysfragments

fragtype T
tabid 239
indexname
colno 0
partn 2097160
strategy E
location L
servername
evalpos 0
Exprtext (date_col = DATE ('10/24/2006'))
dbspace dbsp1
levels 0
npused 0
nrows 0
clust 0
partition jp_1024



Evalpos

- Date_col=1/3/2000 ➔ evalpos 0
- Date_col=1/2/2000 ➔ evalpos 1
- Date_col=1/1/2000 ➔ evalpos 2
- Remainder ➔ evalpos 3



Of interest

- Max(evalpos) is the remainder fragment
- Evalpos=0 is our oldest fragment
- Have to leave at least two fragments or the table stops being a fragmented table and you have to INIT or ATTACH a new partition.
  - ALTER FRAGMENT ON TABLE foo INIT
  - ALTER FRAGMENT ON TABLE foo ATTACH

Locks

- Kill user session
  - Wander through syslocks and figure out who the culprit is
- LOCK MODE WAIT
- Maintenance window

The Remainder Fragment

- If data gets in here, it is no longer in the management scheme.
  - Have to get it out.
- Options
  - Detach remainder
  - Migrate remainder
- Ignore it
  - Mark date it occurred,
  - Drop and create a new remainder when the time comes

Detach issues

- Indexes are dropped
- Column constraints and defaults are dropped
- Have to rebuild these before re-attaching
- While you are doing this, the data is offline.

Attaching a fragment

- If empty, gets the schema of the table, including indices
- If not empty, must match the table.
  constraints and defaults
- Pre-build indices for optimal performance

MODIFY FRAGMENT

- ALTER FRAGMENT ON TABLE <tab> MODIFY PARTITION <frag> TO PARTITION <part> (date_col in (<expstrg>)) IN <dbspace>;
- ALTER FRAGMENT ON TABLE <tab> ADD PARTITION <part> **REMAINDER** in <dbspace>

Modify Fragment issues

- Is a migration
  - Data is inserted/deleted
  - Indices are inserted/deleted
  - Is a single transaction
  - Count on 6x data size for logical log size
  - 2x for Long Transaction HWM

Pitfalls

- Prepared statements have to be re-prepared (-710, -751) “Table has changed”.
- If operating without a Remainder, may run into (-772) “no fragment for data” error.

Summary

- Very fast way to roll off old data.
- Steer clear of:
  - Referential Integrity
    - Or be prepared to turn it off/on
  - ROWIDS
  - Do not let data get into the remainder
    - You don't have to have one at all.
- RTFM
  - Syntax (ALTER), Performance Guide