Master Note:

1260804.1

Prepare Environment:

Adjust glogin.sql:

vi $ORACLE_HOME/sqlplus/admin/glogin.sql

```
define _EDITOR=vi
set timing on feedback on echo on pagesize 50000 linesize 150 trimsp on verify off
```

Create tablespace:

50 datafiles at 100M = 5000M (5GB) start, 1500000M (1 ½ TB) max

sqlplus "/ as sysdba"

```
DROP TABLESPACE TS_TROY INCLUDING CONTENTS AND DATAFILES;

CREATE TABLESPACE TS_TROY DATAFILE
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
```

```
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M,
'+DATA_XOR2' SIZE 100M AUTOEXTEND ON NEXT 100M MAXSIZE 30000M
BLOCKSIZE 8K
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;

    Tablespace created.
    Elapsed: 00:00:12.39
```

Create User:

sqlplus "/ as sysdba"

```
create user troy identified by soug
  default tablespace TS_TROY;
grant dba to TROY;
exit
```

Create some dummy data:

```
sqlplus /nolog
--
connect troy/soug
--
drop table sales;
create table sales as
select
rownum                                      as id,
rownum + 1                                  as flag,
'Oracle Enterprise Edition'                 as product,
mod(rownum,5)                               as channel_id,
mod(rownum,1000)                            as cust_id ,
5000                                        as amount_sold,
trunc(sysdate - 10000 + mod(rownum,10000)) as order_date,
trunc(sysdate - 9999 + mod(rownum,10000))  as ship_date,
to_date('01.'||lpad(to_char(mod(rownum,12)+1),2,'0')||'.2010','dd.mm.yyyy')
                                            as time_id
from dual
connect by level<=2e7;
--
alter table sales nologging;
insert /*+ append */ into sales select * from sales;
commit;
insert /*+ append */ into sales select * from sales;
commit;
insert /*+ append */ into sales select * from sales;
commit;
insert /*+ append */ into sales select * from sales;
commit;
insert /*+ append */ into sales select * from sales;
commit;
insert /*+ append */ into sales select * from sales;
commit;
insert /*+ append */ into sales select * from sales;
commit;
--
select count(*) from sales;
select bytes/1024/1024/1024 GB from user_segments;
--
exec dbms_stats.gather_table_stats('TROY','SALES')
--
    This gives us 80million rows and a little over 4gig.
--
create table sales_query   compress for query   high as select * from sales;
create table sales_archive compress for archive high as select * from sales;
--
create table sales_cache     as select * from sales_query;
create table sales_nocache   as select * from sales_query;
```

OFFLOADING Queries to Storage Cells (smartscan):

OFF:

```
connect troy/soug
alter session set cell_offload_processing=false;
alter system flush buffer_cache;
alter system flush shared_pool;
@mystats
--
select /* NO_SMART_SCAN */ count(*) from sales where channel_id=1;
/
--
select plan_table_output from table (dbms_xplan.display_cursor);
@mystats

-- standard block access
```

ON:

```
connect troy/soug
alter session set cell_offload_processing=true;
@mystats
--
select /* WITH_SMART_SCAN */ count(*) from sales where channel_id=1;
/
--
select plan_table_output from table (dbms_xplan.display_cursor);
@mystats

-- shows smart scan bytes saved by the storage cells only returning result set
```

Storage Indexes:

Netezza ZoneMap is a Storage Index
DataWarehouse - don't need indexes, a la, shared nothing
OLTP - do traditional indexes and the cells still gain from off-loading and flashcache

Storage cells avoiding I/O:

```
-- think of it as an anti-index telling you where things are NOT.
-- might give a false positive (value may be there), but never a false negative
--
-- connect again to initialize session stats
connect troy/soug
@mystats
select * from sales where id=4711;
@mystats

-- there was not a traditional index, yet the storage index gave the performance
--
-- turn off storage indexes
connect troy/soug
alter session set "_kcfis_storageidx_disabled" = TRUE;
@mystats
select * from sales where id=4711;
@mystats
-- this gives traditional painful non-indexed performance
```

Storage cells offloading I/O:

```
--
-- connect again to initialize session stats
connect troy/soug
select count(*) from sales where channel_id=1;
```

```
@mystats

-- shows that smart scan only returned the results I needed
```

Hybrid Columnar Compression:

QUERY - less cpu but more diskspace

```
--create table sales_query compress for query high as select * from sales;
-- I've already done this
     Table created.
     Elapsed: 00:07:49.02
```

ARCHIVE - more cpu but less diskspace

```
--create table sales_archive compress for archive high as select * from sales;
-- I've already done this



col segment_name format a15
col GB           format 99,999.999

select segment_name, sum(bytes/1024/1024/1024)GB from user_segments group by segment_name
order by 1;

    SEGMENT_NAME              GB
    --------------- -----------
    SALES                207.604
    SALES_ARCHIVE           .602
    SALES_QUERY            1.250

select count(*) from sales;

  COUNT(*)
----------
2560000000

Elapsed: 00:00:51.97

select count(*) from sales_QUERY;

  COUNT(*)
----------
2560000000

Elapsed: 00:00:37.56

select count(*) from sales_ARCHIVE;

  COUNT(*)
----------
2560000000

Elapsed: 00:00:44.29
```

FLASH Cache:

```
create table sales_cache    as select * from sales_query;
alter  table sales_cache   storage (cell_flash_cache keep);
select count(*) from sales_cache;

create table sales_nocache  as select * from sales_query;
alter  table sales_nocache storage (cell_flash_cache none);

col segment_name format a15
select segment_name,cell_flash_cache from user_segments;

SEGMENT_NAME    CELL_FL
--------------- -------
SALES           DEFAULT
SALES_QUERY     DEFAULT
SALES_ARCHIVE   DEFAULT
SALES_CACHE     KEEP
SALES_NOCACHE   NONE


alter system flush buffer_cache;
connect troy/soug
@mystats
select count(*) from sales_cache;
@mystats
select count(*) from sales_nocache;
@mystats
```