Digging Deeper

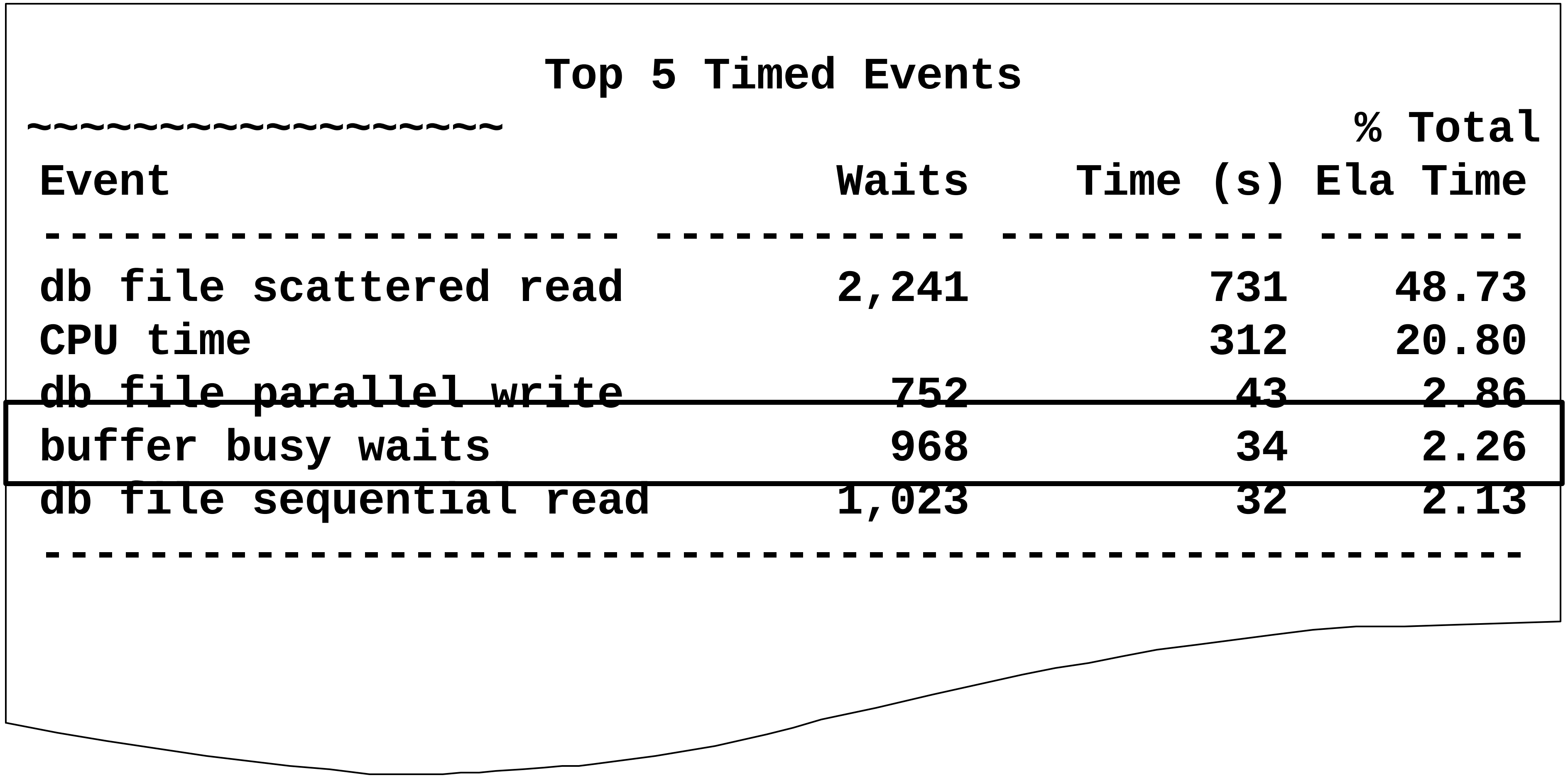
Arup Nanda

Proligence, Inc.

Quick Poll

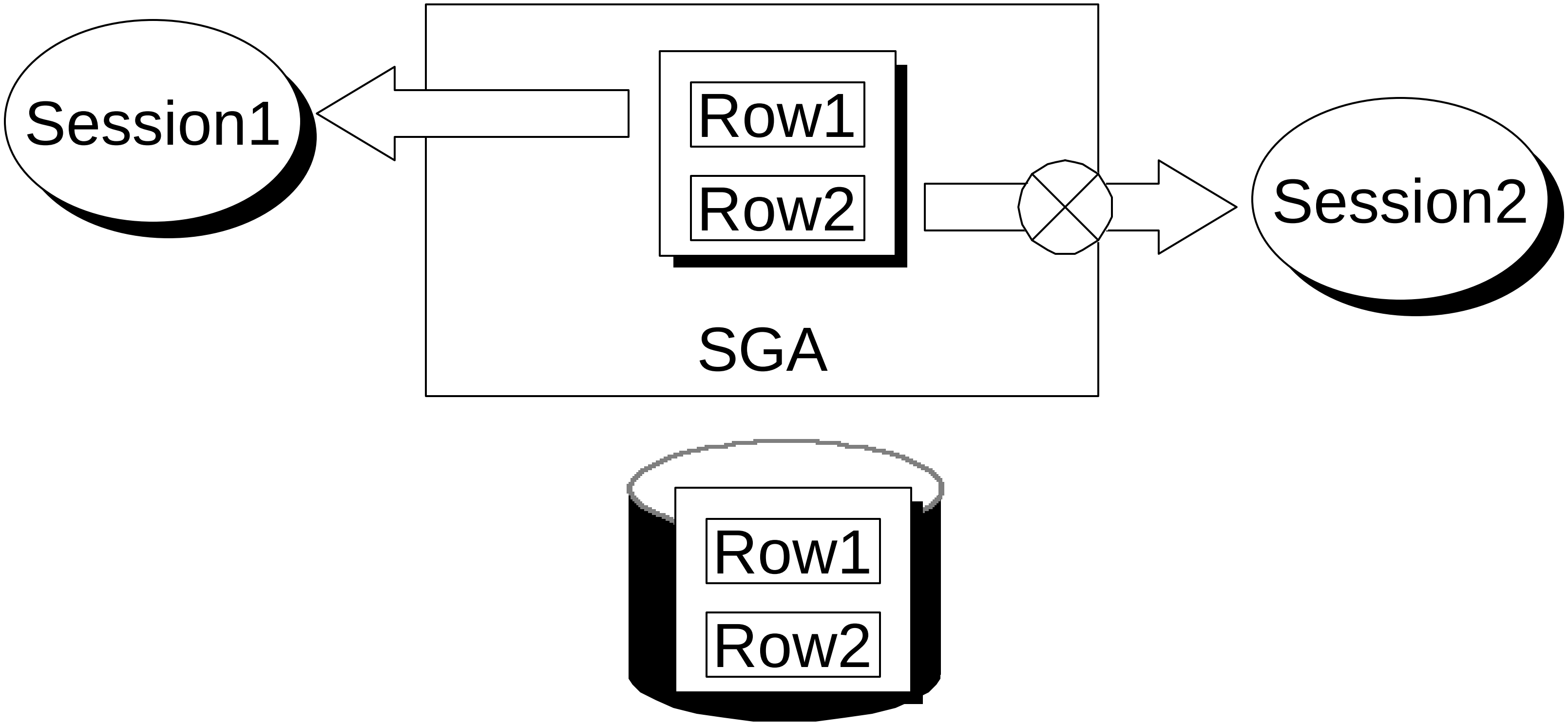
- DBAs
- Developers

STATSPACK Report



```
                   Top 5 Timed Events
~~~~~~~~~~~~~~~~~~                                     % Total
Event                              Waits    Time (s) Ela Time
---------------------- ------------ ----------- --------
db file scattered read            2,241         731    48.73
CPU time                                        312    20.80
db file parallel write              752          43     2.86
buffer busy waits                   968          34     2.26
db file sequential read           1,023          32     2.13
-------------------------------------------------------
```

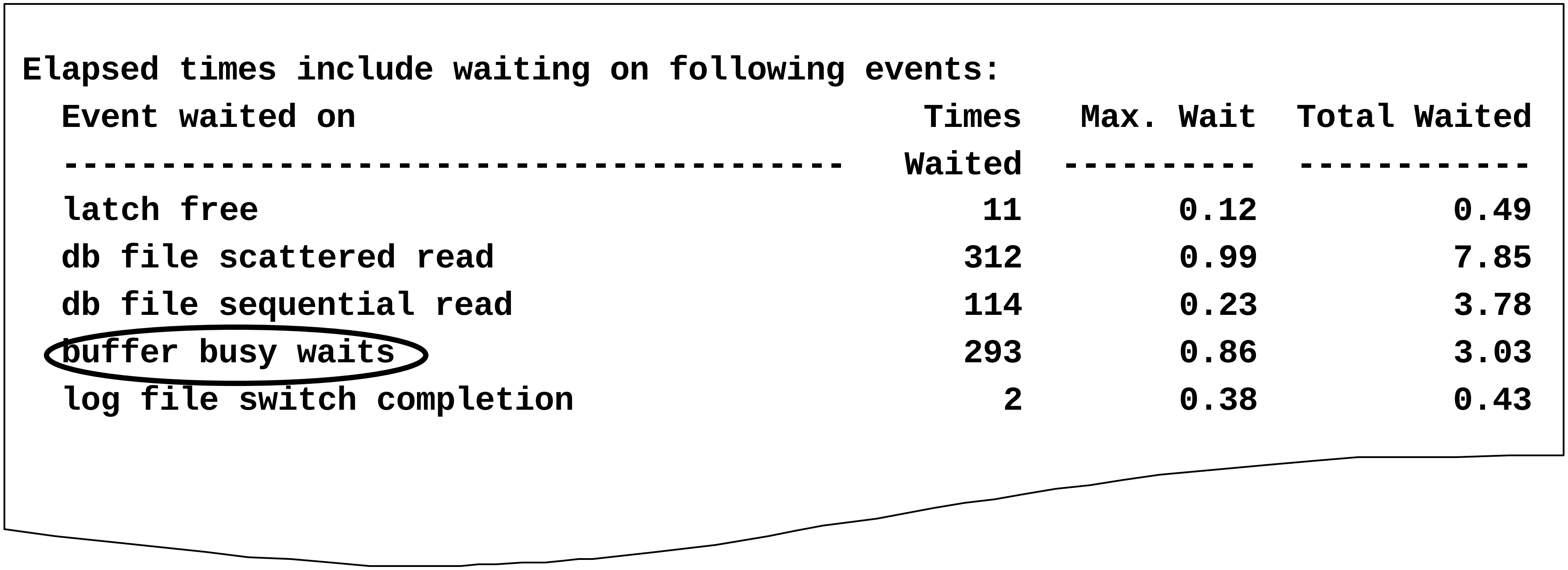
Buffer Busy Waits



Event

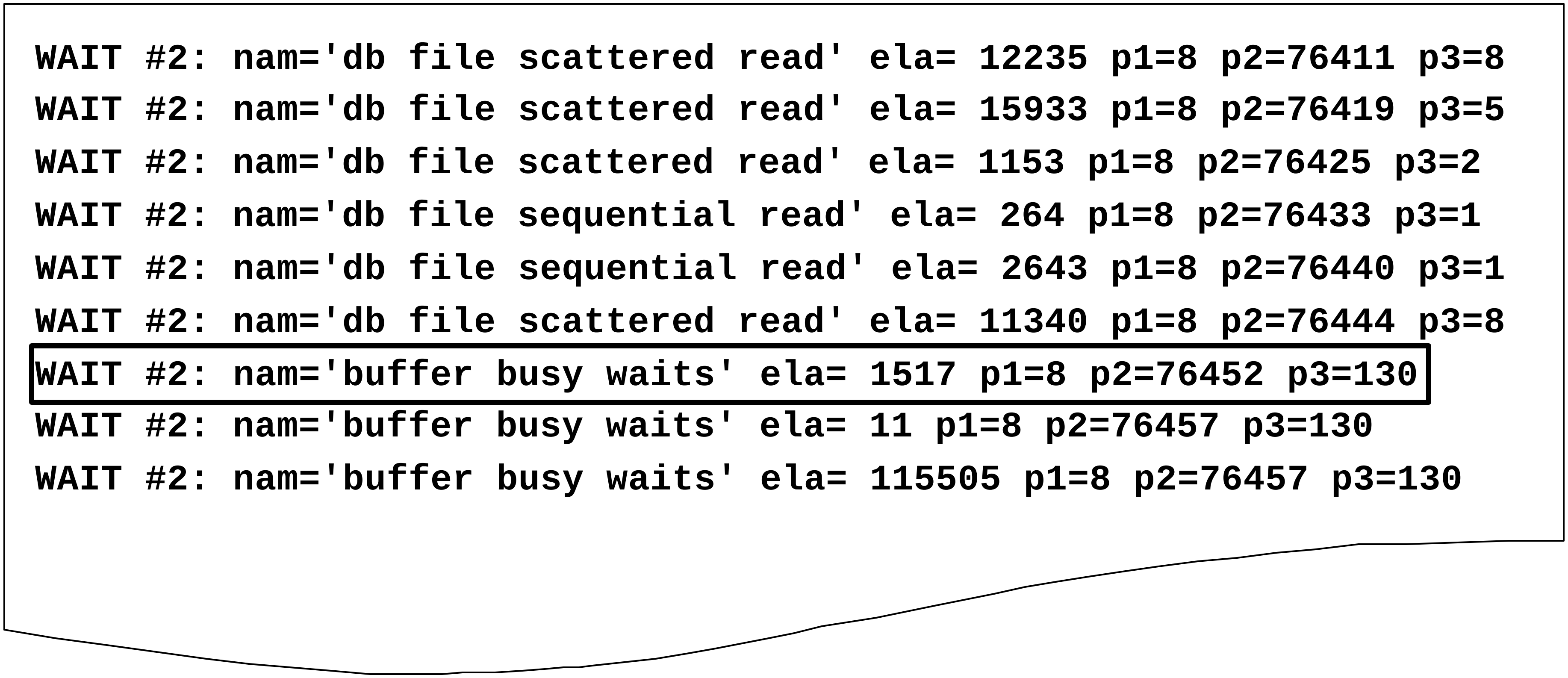
- Event 10046 can be set to get all waits for a session
- Alter session set events ‘10046 trace name context forever, level 8’
- You can also use DBMS_SUPPORT or DBMS_SYSTEM

Tk Prof Output



```
Elapsed times include waiting on following events:
  Event waited on                             Times   Max. Wait  Total Waited
  ----------------------------------------   Waited  ----------  ------------
  latch free                                     11        0.12          0.49
  db file scattered read                        312        0.99          7.85
  db file sequential read                       114        0.23          3.78
  buffer busy waits                             293        0.86          3.03
  log file switch completion                      2        0.38          0.43
```

Output



```
WAIT #2: nam='db file scattered read' ela= 12235 p1=8 p2=76411 p3=8
WAIT #2: nam='db file scattered read' ela= 15933 p1=8 p2=76419 p3=5
WAIT #2: nam='db file scattered read' ela= 1153 p1=8 p2=76425 p3=2
WAIT #2: nam='db file sequential read' ela= 264 p1=8 p2=76433 p3=1
WAIT #2: nam='db file sequential read' ela= 2643 p1=8 p2=76440 p3=1
WAIT #2: nam='db file scattered read' ela= 11340 p1=8 p2=76444 p3=8
WAIT #2: nam='buffer busy waits' ela= 1517 p1=8 p2=76452 p3=130
WAIT #2: nam='buffer busy waits' ela= 11 p1=8 p2=76457 p3=130
WAIT #2: nam='buffer busy waits' ela= 115505 p1=8 p2=76457 p3=130
```

Event

- Long reports – difficult to assess
- All wait events
- Performance hit
- After effect

Oracle Trace

- Different from sql_trace
- Integration with OEM
- Database reporting

Oracle Trace - ??

- After effect
- Poor documentation

Release 2

```
select owner, object_name
from v$segment_statistics
where statistic_name = ‘buffer busy waits’
and value > 0
```

Objective

- Setting up the collection
- Collection Types
- Extensions
- Live Case Study

Setting Up

- STATISTICS_LEVEL in init.ora
- Can be set up via alter system
- Three levels
  - BASIC
  - TYPICAL
  - ALL

TYPICAL

- Buffer Cache
- Mean Time to Recover
- Shared Pool Sizing
- PGA Target
- Timed Statistics
- Segment Level Statistics

ALL

- All of TYPICAL plus
  - Row Execution Stats
  - Timed OS Stats

Current Level

```
SELECT ACTIVATION_LEVEL,
   STATISTICS_NAME,
   SYSTEM_STATUS,
   SESSION_STATUS
FROM V$STATISTICS_LEVEL
ORDER BY ACTIVATION_LEVEL,
   STATISTICS_NAME;
```

Current Stat Levels…

```
ACTIVAT STATISTICS_NAME                SYSTEM_S SESSION_
------- ------------------------------ -------- --------
ALL     Plan Execution Statistics      DISABLED DISABLED
ALL     Timed OS Statistics            DISABLED DISABLED
TYPICAL Buffer Cache Advice            ENABLED  ENABLED
TYPICAL MTTR Advice                    ENABLED  ENABLED
TYPICAL PGA Advice                     ENABLED  ENABLED
TYPICAL Segment Level Statistics       ENABLED  ENABLED
TYPICAL Shared Pool Advice             ENABLED  ENABLED
TYPICAL Timed Statistics               ENABLED  ENABLED
```

V$SEGSTAT

TS#

OBJ#

DATAOBJ#

STATISTIC_NAME

STATISTIC#

VALUE

V$SEGMENT_STATISTICS

OWNER

OBJECT_NAME

SUBOBJECT_NAME

TABLESPACE_NAME

OBJECT_TYPE

Examining the Stats

```
SELECT STATISTIC_NAME, VALUE
FROM V$SEGMENT_STATISTICS
WHERE OWNER = 'SCOTT'
AND OBJECT_NAME = 'SALES';
```

…Examining the Stats

```
STATISTIC_NAME                                VALUE
--------------------------------- ----------
logical reads                               1363168
buffer busy waits                              1649
db block changes                            1430448
physical reads                               238620
physical writes                               15572
physical reads direct                           300
physical writes direct                            0
global cache cr blocks served                     0
global cache current blocks served                0
ITL waits                                         4
row lock waits                                    0
```

Extensions

- X$KSOLSFT
  - FTS_STMP
  - INST_ID
- Adding
  - Instance Number
  - Timestamp

New View: segstat_with_time

```
create or replace view segstat_with_time as
select s.inst_id        Instance_id,
    u.name                          Owner,
    o.name                          Object_name,
    o.subname                       Sub_object_name,
    ts.name                         Tablespace_name,
    decode(o.type#,
            0, 'NEXT OBJECT',
            57, 'SECURITY PROFILE',
            'UNDEFINED')            Object_type,
    s.fts_statnam                   Statistic_name,
    s.fts_staval                    Value,
    to_char(fts_stmp, 'mm/dd/yyyy hh24:mi:ss')
                                    time_stamp
from obj$ o, user$ u, x$ksolsfts s, ts$ ts
where o.owner# = u.user#
and s.fts_inte = 0
and s.fts_objn = o.obj#
and s.fts_tsn = ts.ts#
and s.fts_objd = o.dataobj#
and o.linkname is null
and
(o.type# not in (1, 10) or          (o.type# = 1
    and 1 =             (select 1 from ind$ I where i.obj# = o.obj#
            and i.type# in (1, 2, 3, 4, 6, 7, 9)
            )
    )
)
and o.name != '_NEXT_OBJECT'
and o.name != '_default_auditing_options_'
```

s.inst_id   Instance_id,

to_char (fts_stmp, 'mm/dd/yyyy hh24:mi:ss') time_stamp

Case Study

- OLTP System
- Table SALES

Table: SALES

```
SQL> desc sales
 Name                       Null?    Type
 -------------------------- -------- ------------
 SALES_TRANS_ID             NOT NULL NUMBER
 CUSTOMER_ID                         NUMBER(2)
 PRODUCT_ID                          CHAR(10)
 PRICE                               NUMBER(10,2)
 QUANTITY                            NUMBER(5)
 COMMENTS                            VARCHAR2(20)
```

Stress.sql

```
declare
  v_cust_id       number(6) := 0;
begin
  for v_cust_id in 1..60 loop
      update sales
      set comments = 'CHANGED by &&1'
      where customer_id = v_cust_id
      and mod(sales_trans_id,2) = &&1;
      commit;
  end loop;
end;
```

Stressing

- Running from SQL*Plus Concurrently

```
SQL> @stress 0
SQL> @stress 1
```

Putting it all together

- STATSPACK Report
- Segment Statistics

Resolution

- Packing Factor Reduction
- Increase PCFREE, INITRANS, MAXTRANS, FREELISTS, FREELIST GROUPS
- Reloading the Data

Table Design

- Increase PCTFREE, INITRANS, MAXTRANS, FREELISTS and FREELIST GROUPS

```
pctfree 20
pctused 70
storage (freelists 7
 freelist groups 3 )
initrans 4 maxtrans 30
```

In Conclusion

- Only a few stats are collected. Hope for new stats.
- Provides insight into segment level stats hitherto impossible to access.

More Information

- Oracle Documentation

  http://otn.oracle.com/docs/products/oracle9i/doc_library/release2/server.920/a96533/instance.htm#34509

- Proligence Website (for updated copy and support)

  http://www.proligence.com

- Contact Me

  arup@proligence.com

Questions?

proligence

www.proligence.com