

På innsiden av Oracle optimalisering.

Kan vi stole på at de riktige vurderingene blir tatt?

Ingemar Jansson Haverstad

Om foredragsholderen



Ingemar Jansson Haverstad
dbWatch Services

- Arbeidet med Oracle siden 1985
- Flere og flere «hull» i kompetansen...
- Konsulent, instruktør og foredragsholder

ingemar@oraklet.no
www.oraklet.no/foredrag

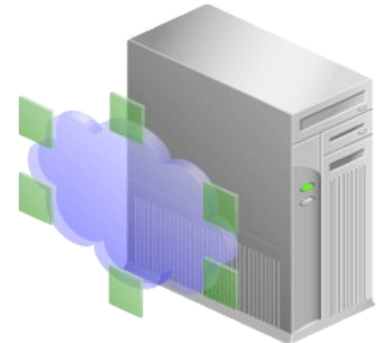
ORACLE
Certified Professional

Agenda

- Hvordan vet vi at den beste eksekveringsplanen blir benyttet når en SQL skal evalueres?
- Hvordan veies tabellsøk opp mot bruk av indekser, hvilke algoritmer ligger i bunn?
- Når to tabeller føyes sammen, hvilken type sorteringsmekanisme bør brukes: «**Nested Loop**», «**Sort Merge join**» eller «**Hash join**»?
- Hvilke alternativer foreligger når eksekveringsplanen ikke fungerer?

Bakgrunn

- **Datavarehus i Skatteetaten**
 - Påbegynt i 2003
- **IBM/AIX Unix server**
 - 16 CPU
 - 72 GB internminne
- Oracle10g R2 (10.2.0.4)
- Største tabellen har 6.000.000.000 rader
- Temporær tabellplass er 1 TB



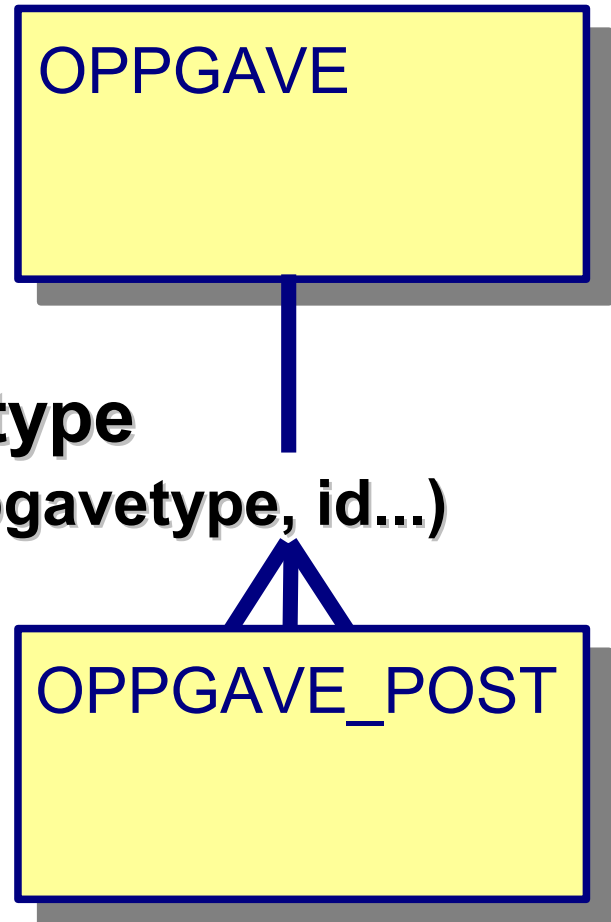
To sentrale tabeller

- **OPPGAVE**

- 400.000.000 rader
- Partisjonert – Range: **År**
- Subpartisjonert – List: **Oppgavetype**
- To unike indekser: **Lokalt (År, oppgavetype, id...)**

- **OPPGAVE_POST**

- 6.000.000.000 rader
- Partisjonert – Range: **År**
- Subpartisjonert – List: **Oppgavetype**
- Unikt indeks: **Lokalt (År, oppgave_id, oppgavetype...)**



OPPGAVE_POST - range

```
DVH@DEMO10G SQL> SELECT partition_name, num_rows, last_analyzed
FROM user_tab_partitions
WHERE table_name = 'OPPGAVE_POST';
```

PARTITION_NAME	NUM_ROWS	LAST_ANALYZED
OP_1999	0	05-MAR-09
OP_2000	356966370	01-MAR-09
OP_2001	463351240	01-MAR-09
OP_2002	633242498	01-MAR-09
OP_2003	711124142	01-MAR-09
OP_2004	816936253	01-MAR-09
OP_2005	873603550	01-MAR-09
OP_2006	920591868	01-MAR-09
OP_2007	948448860	01-MAR-09
OP_2008	14840741	01-MAR-09
OP_2009	12768	01-MAR-09
OP_2010	0	05-MAR-09
OP_DEFAULT	0	05-MAR-09

13 rows selected.

OPPGAVE_POST - list

```
DVH@DEMO10G SQL> SELECT subpartition_name,
                        high_value,
                        num_rows,
                        last_analyzed
FROM user_tab_subpartitions
WHERE partition_name = 'OP_2007';
```

SUBPARTITION_NAME	HIGH_VALUE	NUM_ROWS	LAST_ANALYZED
OP_AAR_2007_GA	'GA'	29124657	01-MAR-09
OP_AAR_2007_DEFAULT	DEFAULT	84771628	01-MAR-09
OP_AAR_2007_LFPUB	'LFPUB'	0	05-MAR-09
OP_AAR_2007_SL	'SL'	110963067	01-MAR-09
OP_AAR_2007_SLN	'SLN'	122369653	01-MAR-09
OP_AAR_2007_GB	'GB'	0	05-MAR-09
OP_AAR_2007_DSB	'DSB'	118014974	01-MAR-09
OP_AAR_2007_LFPIB	'LFPIB'	358936095	02-MAR-09
OP_AAR_2007_MVA	'MVA'	0	05-MAR-09
OP_AAR_2007_LEP	'LEP'	14413737	01-MAR-09

10 rows selected.

Statistikk

```
DVH@DEMO10G SQL> SELECT num_rows, last_analyzed
                   FROM user_tables
                   WHERE table_name = 'OPPGAVE_POST';
```

```
NUM_ROWS LAST_ANALY
-----
6222478714 10.03.2009
```

Tabell

```
DVH@DEMO10G SQL> SELECT num_rows, last_analyzed
                   FROM user_tab_partitions
                   WHERE partition_name = 'OP_2004';
```

```
NUM_ROWS LAST_ANALY
-----
948448860 10.03.2009
```

Partisjon

```
DVH@DEMO10G SQL> SELECT num_rows, last_analyzed
                   FROM user_tab_subpartitions
                   WHERE subpartition_name = 'OP_AAR_2007_SL';
```

```
NUM_ROWS LAST_ANALY
-----
110963067 10.03.2009
```

Subpartisjon

SQL på overtid...

```
DVH@DEMO10G SQL> SELECT /*+ PARALLEL(p,16) PARALLEL(o,16) */ *  
FROM oppgave o, oppgave_post p  
WHERE p.aar IN (2007)  
AND o.aar IN (2007)  
AND o.oppgave_type_kode = 'DSB'  
AND p.oppgave_type_kode = 'DSB'  
...  
AND o.id = p.opp_id;  
  
...  
578 rows selected
```

Elapsed: 01:42:09.14

```
DVH@DEMO10G SQL> SELECT /*+ PARALLEL(p,16) PARALLEL(o,16)  
USE_HASH(o p) ORDERED */ *  
  
...  
578 rows selected
```

Elapsed: 00:00:28.40

Explain plan - I

```
DVH@DEMO10G SQL> @?/rdbms/admin/utlxplp
```

Operation	Name	Cost
SELECT STATEMENT		7
TABLE ACCESS BY LOCAL INDEX ROWID	OPPGAVE_POST	3
NESTED LOOPS		7
PARTITION RANGE SINGLE		4
PARTITION LIST SINGLE		4
TABLE ACCESS BY LOCAL INDEX ROWID	OPPGAVE	4
INDEX RANGE SCAN	OPP_OPP_UK2	3
PARTITION RANGE SINGLE		2
PARTITION LIST SINGLE		2
INDEX RANGE SCAN	OP_OP_UK	2

Explain plan - II

```
DVH@DEMO10G SQL> @?/rdbms/admin/utlxplp
```

Operation	Name	Cost	Pstart	Pstop
SELECT STATEMENT		2978		
PX COORDINATOR				
PX SEND QC (RANDOM)	:TQ10002	2978		
HASH JOIN BUFFERED		2978		
PX JOIN FILTER CREATE	:BF0000	2767		
PX RECEIVE		2767		
PX SEND HASH	:TQ10000	2767		
PX BLOCK ITERATOR		2767	KEY	KEY
TABLE ACCESS FULL	OPPGAVE_POST	2767	80	80
PX RECEIVE		204		
PX SEND HASH	:TQ10001	204		
PX JOIN FILTER USE	:BF0000	204		
PX BLOCK ITERATOR		204	KEY	KEY
TABLE ACCESS FULL	OPPGAVE	204	78	78

Simulere i et testmiljø

```
DVH@DV1P SQL> EXECUTE dbms_stats.create_stat_table( -  
                OWNNAME=>'DVH', -  
                STATTAB=>'dvhstat');  
PL/SQL procedure successfully completed.
```

```
DVH@DV1P SQL> EXECUTE dbms_stats.export_schema_stats( -  
                OWNNAME=>'DVH', -  
                STATTAB=>'dvhstat');  
PL/SQL procedure successfully completed.
```

export/import...

```
DVH@DEMO10G SQL> EXECUTE dbms_stats.import_schema_stats( -  
                OWNNAME=>'DVH', -  
                STATTAB=>'dvhstat');  
PL/SQL procedure successfully completed.
```

Hva skjer?

Svaret finner du i «event 10053»

- I egen sesjon

```
ALTER SESSION SET EVENTS '10053 trace name context forever';  
...  
ALTER SESSION SET EVENTS '10053 trace name context off';
```

- Andre sesjoner

```
EXEC SYS.DBMS_SYSTEM.SET_EV (<sid>, <serial#>, 10053, {1|2}, '')  
...  
EXEC SYS.DBMS_SYSTEM.SET_EV (<sid>, <serial#>, 10053, 0, '')
```

Kjører spørringen igjen...

```
$ sqlplus dvh/#####  
  
SQL*Plus: Release 10.2.0.4.0 - Production  
  
DVH@DEMO10G SQL> ALTER SESSION SET EVENTS  
                    '10053 trace name context forever';  
  
DVH@DEMO10G SQL> EXPLAIN PLAN FOR  
                   SELECT /*+ PARALLEL(p,16) PARALLEL(o,16) */ *  
                   FROM oppgave o, oppgave_post p  
                   WHERE p.aar IN (2007)  
                   AND    o.aar IN (2007)  
                   AND    o.oppgave_type_kode = 'DSB'  
                   AND    p.oppgave_type_kode = 'DSB'  
                   ...  
                   AND    o.id = p.opp_id;
```

Explained.

- Trace fil i *USER_DUMP_DEST*-katalogen

Event 10053 trace

- Generell del
- Forklaring til akronymer
- SQL
- Parametere som er blitt benyttet
- Rettelser som er inkludert
- System statistikk
- Basis Statistikk
- Basis aksess kostnader
- Eksekveringsplaner
- Kostnader ved bruk av spesielle egenskaper

Forklaring til akronymer

CBQT - cost-based query transformation
 JPPD - join predicate push-down
 FPD - filter push-down
 PM - predicate move-around
 CVM - complex view merging
 SPJ - select-project-join
 SJC - set join conversion
 SU - subquery unnesting
 OBYE - order by elimination
 ST - star transformation
 qb - query block
 LB - leaf blocks
 DK - distinct keys
 LB/K - av. # of leaf blocks per key
 DB/K - av. # of data blocks per key
 CLUF - clustering factor
 NDV - number of distinct values
 Resp - response cost
 Card - cardinality
 Resc - resource cost
 NL - nested loops (join)
 SM - sort merge (join)
 HA - hash (join)

CPUCSPEED - CPU Speed
 IOTFRSPEED - I/O transfer speed
 IOSEEKTIM - I/O seek time
 SREADTIM - avr. single block read time
 MREADTIM - avr. multiblock read time
 MBRC - avr. multiblock read count
 MAXTHR - max I/O system throughput
 SLAVETHR - avr. slave I/O throughput
 dmeth - distribution method
 1: no partitioning required
 2: value partitioned
 4: right is random (round-robin)
 512: left is random (round-robin)
 8: broadcast right and part. left
 16: broadcast left and part. Right
 32: partition left part. of right
 64: partition right part. of left
 128: use hash part. dimension
 256: use range part. dimension
 2048: use list part. dimension
 1024: run the join in serial
 sel - selectivity
 ptn - partition

Parametere som er blitt benyttet

```

*****
PARAMETERS USED BY THE OPTIMIZER
*****

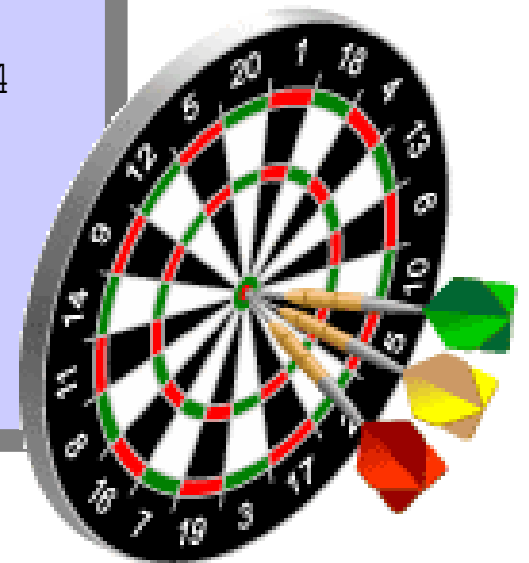
*****
PARAMETERS WITH ALTERED VALUES
*****

_pga_max_size                = 2097140 KB
query_rewrite_integrity      = trusted
*****

optimizer_mode_hinted        = false
optimizer_features_hinted    = 0.0.0
_query_rewrite_fudge         = 90
optimizer_features_enable    = 10.2.0.4
_optimizer_search_limit      = 5
cpu_count                     = 24
active_instance_count        = 1
parallel_threads_per_cpu     = 2
hash_area_size               = 131072

```

- Totalt 200 parametere



Systemstatistikk

```

*****
SYSTEM STATISTICS INFORMATION
*****

Using WORKLOAD Stats
CPUSPEED: 801 millions instructions/sec
SREADTIM: 7 milliseconds
MREADTIM: 49 millisecons
MBRC: 57.000000 blocks
MAXTHR: 771072 bytes/sec
SLAVETHR: 1430528 bytes/sec

```

- cpuspeed: millioner sykler per sekund
- sreadtim: vente tid for å lese et blokk, millisekunder
- mreadtim: vente tid for å lese flere blokker, millisekunder
- **Cost =**
 $(\#SRds * sreadtim + \#MRds * mreadtim + \#CPUCycles / cpuspeed) / sreadtim$

Kan kjøre pakken `dbms_stats` for å samle statistikk:

```

SYS@DEMO10G SQL> execute dbms_stats.gather_system_stats('START');
...
SYS@DEMO10G SQL> execute dbms_stats.gather_system_stats('STOP');

```

Enkel les og les av flere blokker...



When I looked at the stats that were collected, I was surprised to see that with most of the databases, mreadtim < sreadtim.

and we said...

I decided to outsource this one :) I asked Jonathan Lewis to comment and he wrote:

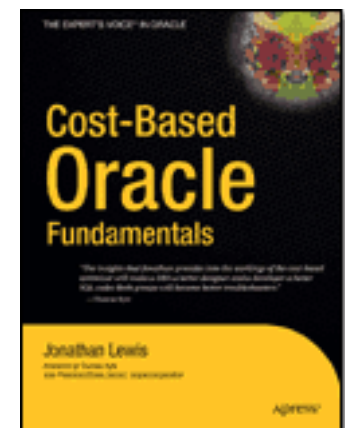
Tom,

...

I am still unsure of the best way to deal with the problem, but one thought is to use the gathered statistics to give Oracle a value for the MBRC value, and then use an external calibration tool to time the hardware on randomly scattered reads of the single and multiblock read size.

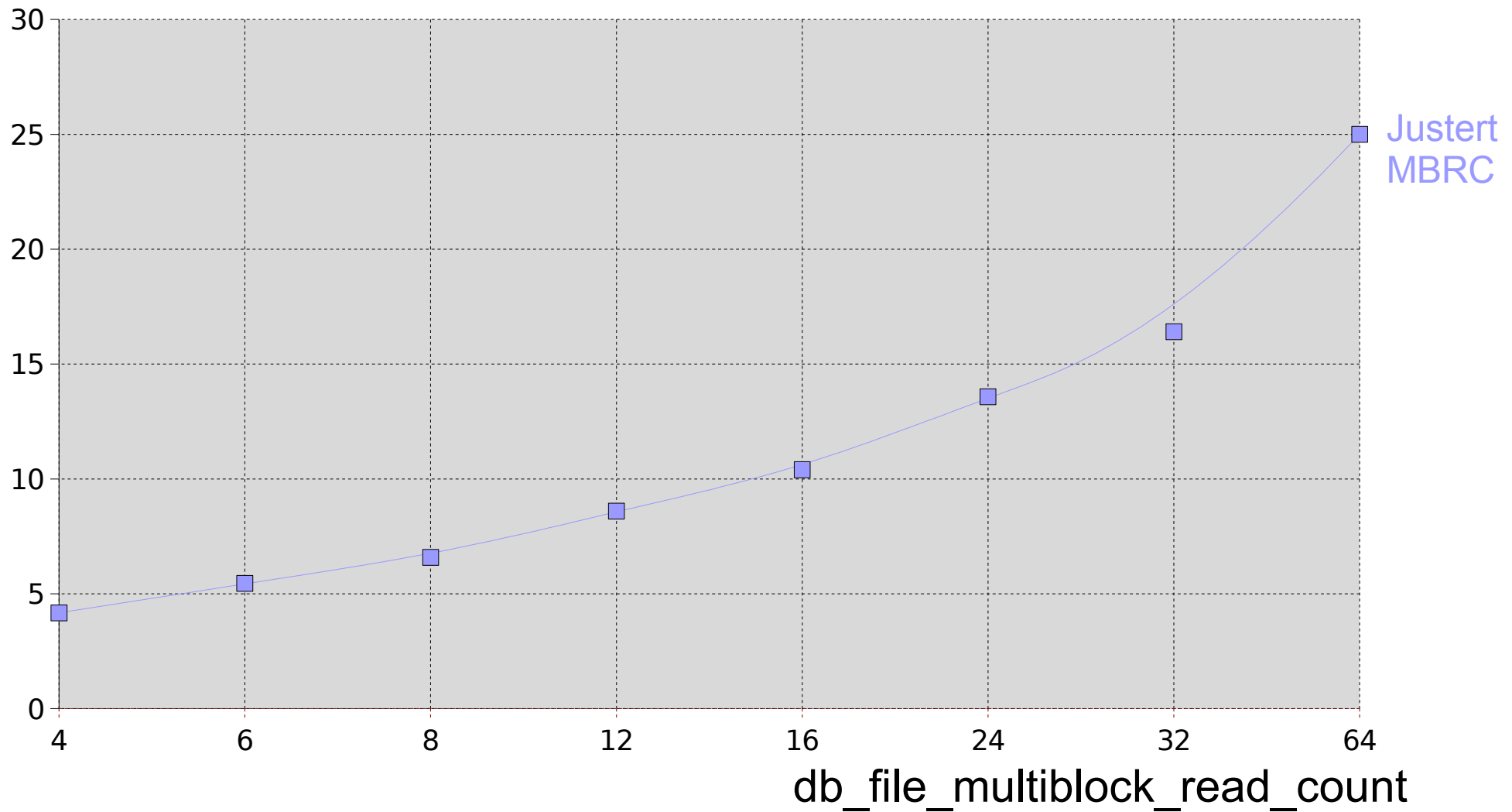
Regards

Jonathan Lewis
Author of **Cost Based Oracle: Fundamentals**



Full tabell les

db_file_multiblock_read_count



Basis statistikk - I

```
*****
```

BASE STATISTICAL INFORMATION

```
*****
```

Table Stats::

```
Table: OPPGAVE_POST  Alias:  P  (making adjustments for
partition skews)
```

```
ORIGINAL VALUES::      #Rows: 29124657  #Blks:  57352
```

```
AvgRowLen:  52.00
```

SUBPARTITIONS::

```
PRUNED: 1
```

```
ANALYZED: 1  UNANALYZED: 0
```

```
Partition [74]  (NOT ANALYZED)
```

```
#Rows: 0  #Blks:  143  AvgRowLen:  1300234247.00
```

```
#Rows: 29124657  #Blks:  312240  AvgRowLen:  52.00
```

Index Stats::

```
Index: OP_OP_UK  Col#: 2 3 4 5 7  PARTITION [74]
```

```
LVLS: 2  #LB: 131296  #DK: 30468369  LB/K: 1.00  DB/K: 1.00
```

```
CLUF: 30358917.00
```

```
LVLS: 2  #LB: 131296  #DK: 30468369  LB/K: 1.00  DB/K: 1.00
```

```
CLUF: 30358917.00
```

Basis statistikk - II

```
*****
```

BASE STATISTICAL INFORMATION

```
*****
```

Table Stats::

```
Table: OPPGAVE_POST  Alias:  P  (making adjustments for
partition skews)
```

```
ORIGINAL VALUES::      #Rows:  948448860  #Blks:  2227708
```

```
AvgRowLen:  58.00
```

SUBPARTITIONS::

```
PRUNED: 1
```

```
ANALYZED: 1  UNANALYZED: 0
```

```
Partition [8]
```

```
#Rows:  948448860  #Blks:  2227708  AvgRowLen:  58.00
```

```
#Rows:  948448860  #Blks:  312240  AvgRowLen:  58.00
```

Index Stats::

```
Index: OP_OP_UK  Col#:  2  3  4  5  7  PARTITION [8]
```

```
LVLS:  3  #LB:  2997536  #DK:  925637449  LB/K:  1.00  DB/K:
```

```
1.00  CLUF:  922467766.00
```

```
LVLS:  3  #LB:  2997536  #DK:  925637449  LB/K:  1.00  DB/K:
```

```
1.00  CLUF:  922467766.00
```

Partisjon OP_2007 med **948448860** rader

Basis statistikk - I

BASE STATISTICAL INFORMATION

Table Stats::

Table: OPPGAVE_POST Alias: P (making adjustments for
partition skews)

ORIGINAL VALUES:: #Rows: **29124657** #Blks: 57352

AvgRowLen: 52.00

SUBPARTITIONS::

PRUNED: 1

ANALYZED: 1 UNANALYZED: 0

Partition [74] (NOT ANALYZED)

#Rows: 0 #Blks: 143 AvgRowLen: 1300234247.00

#Rows: 29124657 #Blks: 312240 AvgRowLen: 52.00

Index Stats::

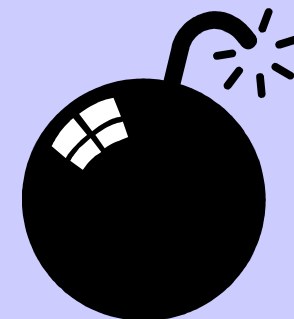
Index: OP_OP_UK Col#: 2 3 4 5 7 PARTITION [74]

LVLS: 2 #LB: 131296 #DK: 30468369 LB/K: 1.00 DB/K: 1.00

CLUF: 30358917.00

LVLS: 2 #LB: 131296 #DK: 30468369 LB/K: 1.00 DB/K: 1.00

CLUF: 30358917.00



Subpartisjon OP_AAR_2007_GA 'GA' med **29124657** rader

OPPGAVE_POST - list

```
DVH@DEMO10G SQL> SELECT subpartition_name,
                        high_value,
                        num_rows,
                        last_analyzed
FROM user_tab_subpartitions
WHERE partition_name = 'OP_2007';
```

SUBPARTITION_NAME	HIGH_VALUE	NUM_ROWS	LAST_ANALYZED
OP_AAR_2007_GA	'GA'	29124657	01-MAR-09
OP_AAR_2007_DEFAULT	DEFAULT	84771628	01-MAR-09
OP_AAR_2007_LFPUB	'LFPUB'	0	05-MAR-09
OP_AAR_2007_SL	'SL'	110963067	01-MAR-09
OP_AAR_2007_SLN	'SLN'	122369653	01-MAR-09
OP_AAR_2007_GB	'GB'	0	05-MAR-09
OP_AAR_2007_DSB	'DSB'	118014974	01-MAR-09
OP_AAR_2007_LFPIB	'LFPIB'	358936095	02-MAR-09
OP_AAR_2007_MVA	'MVA'	0	05-MAR-09
OP_AAR_2007_LEP	'LEP'	14413737	01-MAR-09

10 rows selected.

OPPGAVE_POST - range

```
DVH@DEMO10G SQL> SELECT partition_name, num_rows, last_analyzed
FROM user_tab_partitions
WHERE table_name = 'OPPGAVE_POST';
```

PARTITION_NAME	NUM_ROWS	LAST_ANALYZED
OP_1999	0	05-MAR-09
OP_2000	356966370	01-MAR-09
OP_2001	463351240	01-MAR-09
OP_2002	633242498	01-MAR-09
OP_2003	711124142	01-MAR-09
OP_2004	816936253	01-MAR-09
OP_2005	873603550	01-MAR-09
OP_2006	920591868	01-MAR-09
OP_2007	948448860	01-MAR-09
OP_2008	14840741	01-MAR-09
OP_2009	12768	01-MAR-09
OP_2010	0	05-MAR-09
OP_DEFAULT	0	05-MAR-09

13 rows selected.

Er statistikken OK?

```
DVH@DV1P SQL> EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS( -  
    'DVH', -  
    CASCADE=>TRUE, -  
    GRANULARITY=>'ALL', -  
    METHOD_OPT=>'FOR COLUMNS', -  
    DEGREE=>10);
```

Skjema

```
DVH@DV1P SQL> EXECUTE dbms_stats.gather_table_stats( -  
    OWNNAME=>'DVH', -  
    TABNAME=>'OPPGAVE_POST', -  
    PARTNAME=>'OP_AAR_2004_DEFAULT', -  
    DEGREE=>16, -  
    CASCADE=>TRUE, -  
    GRANULARITY=>'ALL', -  
    METHOD_OPT=>'FOR COLUMNS');
```

Subpartisjon

Bug 7210921

Søkeord i *My Oracle Support*: [statistics subpartitions](#)

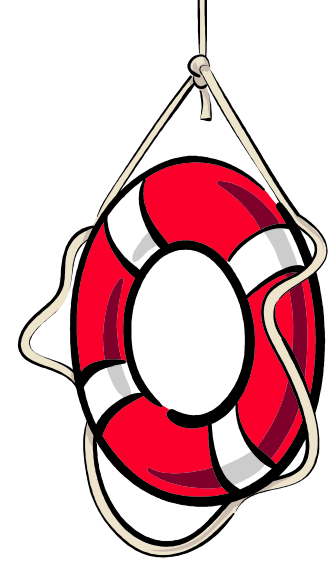
Bug No.	7210921	Updated	18-NOV-2008
Filed	26-JUN-2008	Product Version	10.2.0.4
Product	Oracle Server	Platform Version	No Data
Platform	Microsoft Windows XP	Affects Platforms	Generic
Database Version	10.2.0.4	Status	Dev. to Q/A
Severity	Severe Loss of Service	Fixed in Product Version	11.1
Base Bug	N/A		

Problem statement:

STATISTICS WITH SUBPARTITIONS ARE NOT CORRECTLY INTERPRETED IN EXPLAIN PLANS



Alternativene!



- Oppgrader til *Oracle11g*
- Nedgrader til *Oracle10g* versjon 10.2.0.3
- Benytt parameteren *OPTIMIZER_FEATURES_ENABLE*

```
SYSTEM@DEMO10G SQL> ALTER SYSTEM SET
                        optimizer_features_enable = '10.2.0.3';
```

- Anvend hints:

```
DVH@DEMO10G SQL> SELECT /*+ PARALLEL(p,16) PARALLEL(o,16)
                        USE_HASH(o p) ORDERED */ *
FROM oppgave o, oppgave_post p
...;
```

- Spesifiser subpartisjon

```
DVH@DEMO10G SQL> SELECT /*+ PARALLEL(p,16) PARALLEL(o,16) */ *
FROM oppgave subpartition(opp_aar2007_dsb) o,
     oppgave_post subpartition(op_aar_2007_dsb) p
...;
```

Tabell statistikk

SINGLE TABLE ACCESS PATH

Access Path: TableScan

Cost: 2935.67 Resp: 203.87 Degree: 0

Cost_io: 2724.00 Cost_cpu: 1182062467

Resp_io: 189.17 Resp_cpu: 82087671

Access Path: index (RangeScan)

Index: OPP_OPP_UK

resc_io: 5116002.00 resc_cpu: 39279344872

ix_sel: 0.99989 ix_sel_with_filters: 0.075571

Cost: 5123041.95 Resp: 5123041.95 Degree: 1

Access Path: index (RangeScan)

Index: OPP_OPP_UK2

resc_io: 558577.00 resc_cpu: 5524731778

ix_sel: 0.075571 ix_sel_with_filters: 0.075571

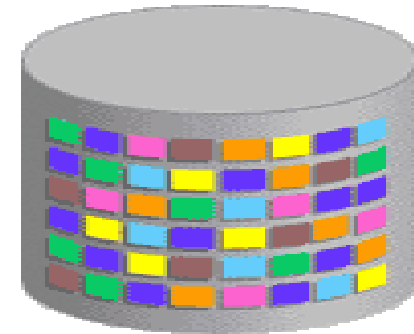
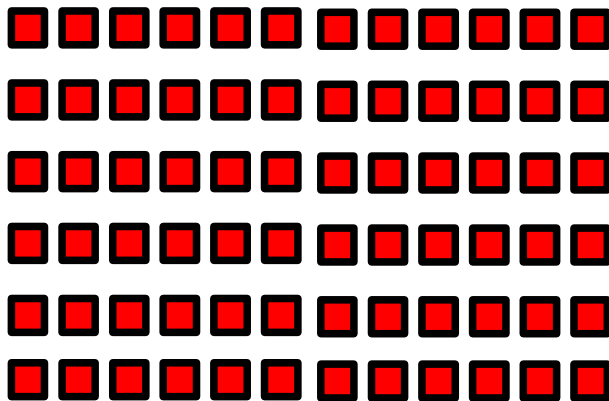
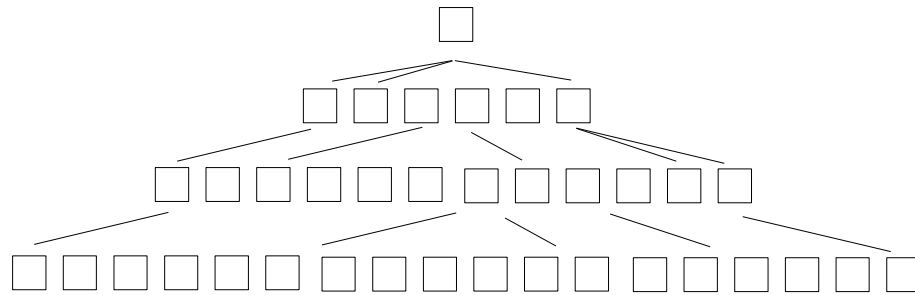
Cost: 559566.28 Resp: 559566.28 Degree: 1

Best:: AccessPath: TableScan

Cost: 203.87 Degree: 16

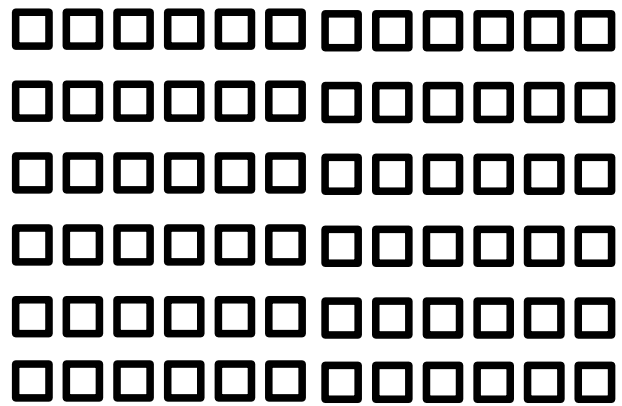
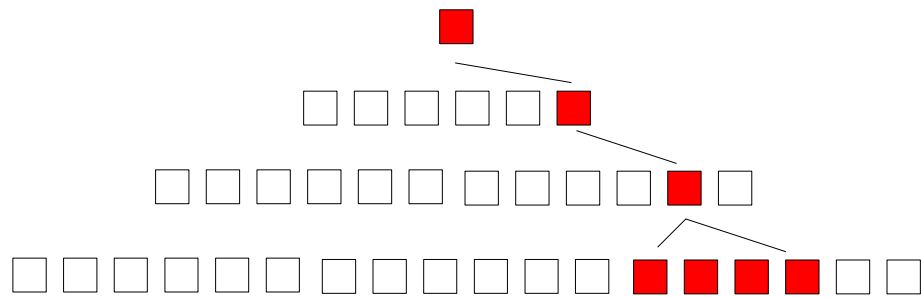
Resp: 203.87 Card: 4967535.68 Bytes: 0

Full Table Scan



- Beregnad kostnad:
 nblks / k

Range Scan

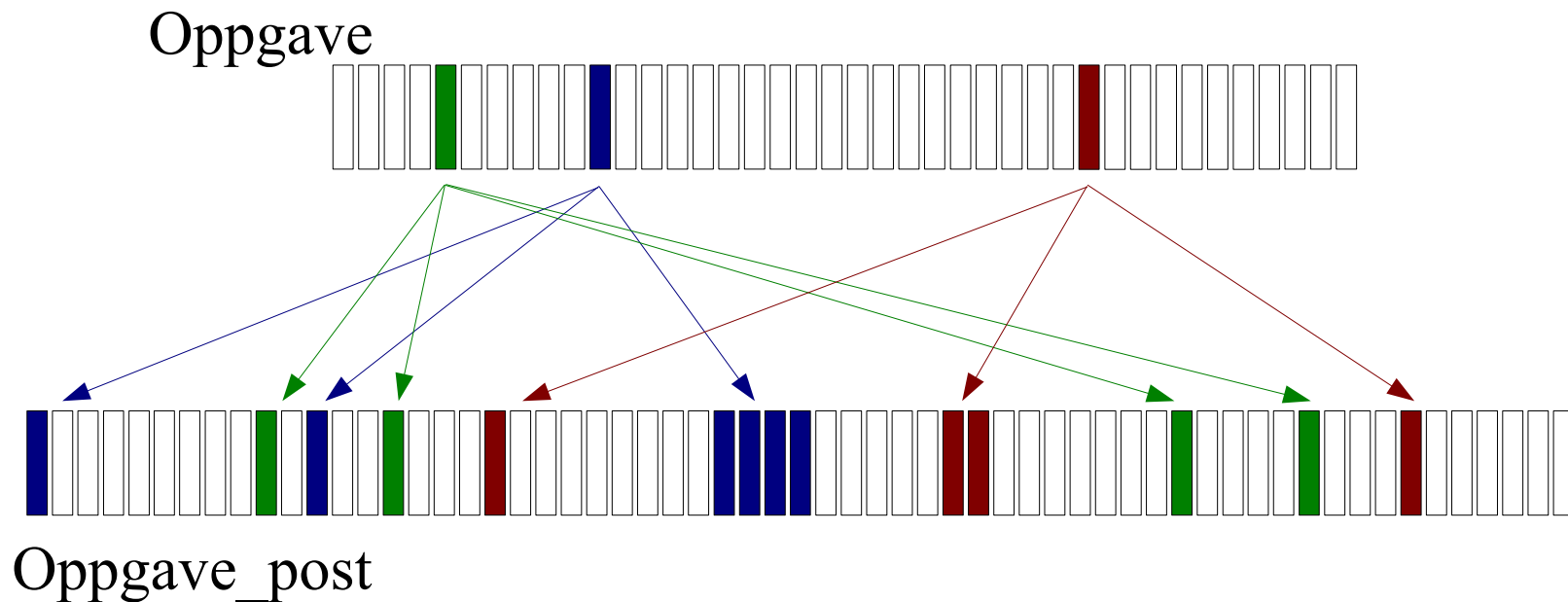


- **Beregnet kostnad:**

$$\text{blevel} + \text{FF} * \text{leaf_blocks} * \text{clustering_factor}$$

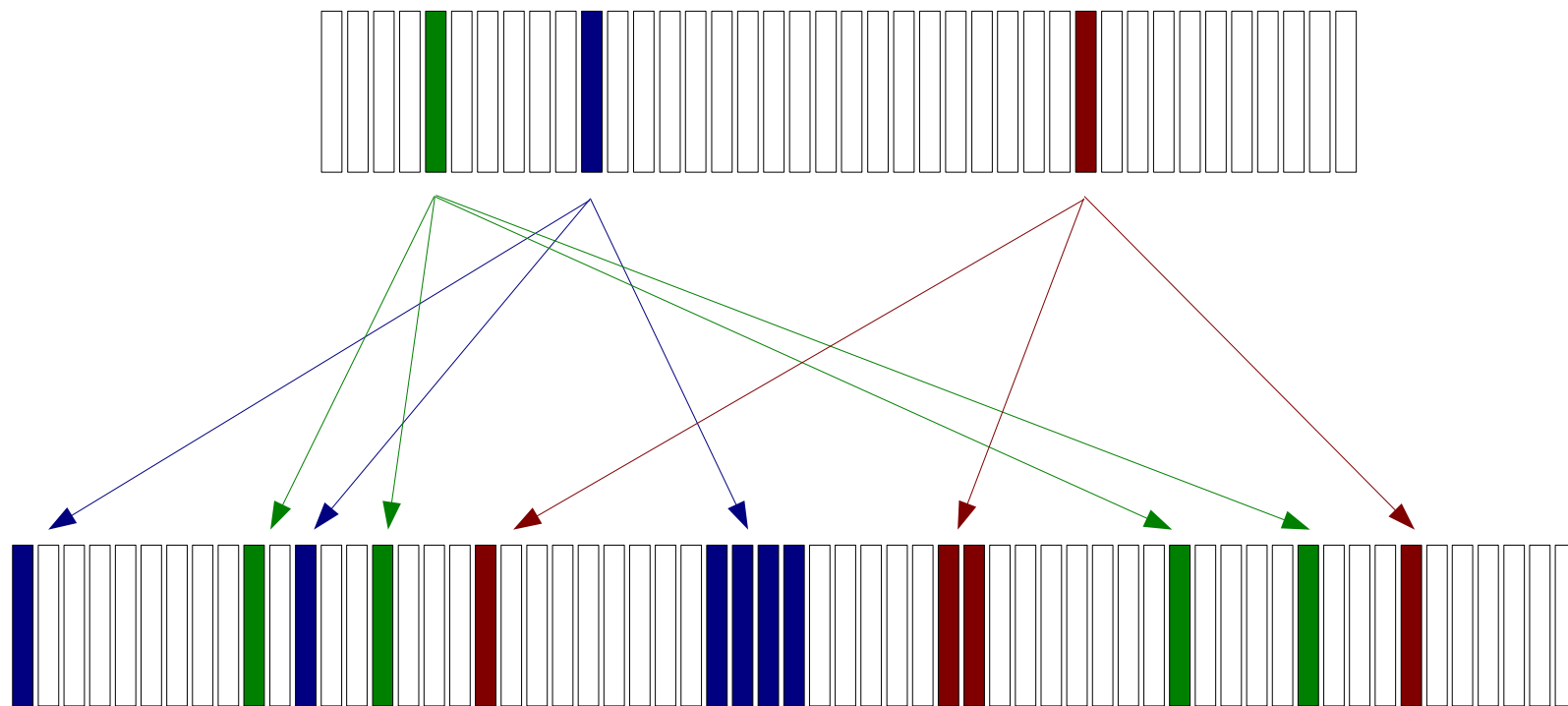
«Nested Loop», «Sort Merge join» eller «Hash join»?

- Nested loop sikker men tregest.
- Hash Join hurtigst.
- Sort Merge midt i mellom.
- Hvilken tabell skal være den drivende?



Nested Loop – NL Join

Oppgave

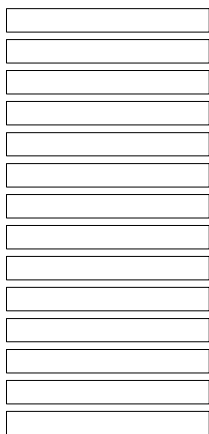


Oppgave post

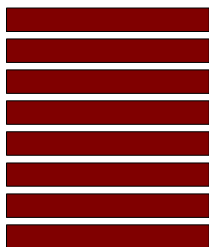
Join cost = cost of accessing outer table +
 (cardinality of outer table * cost of accessing inner table)

Sort Merge – SM Join

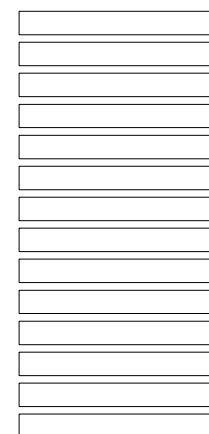
Oppgave



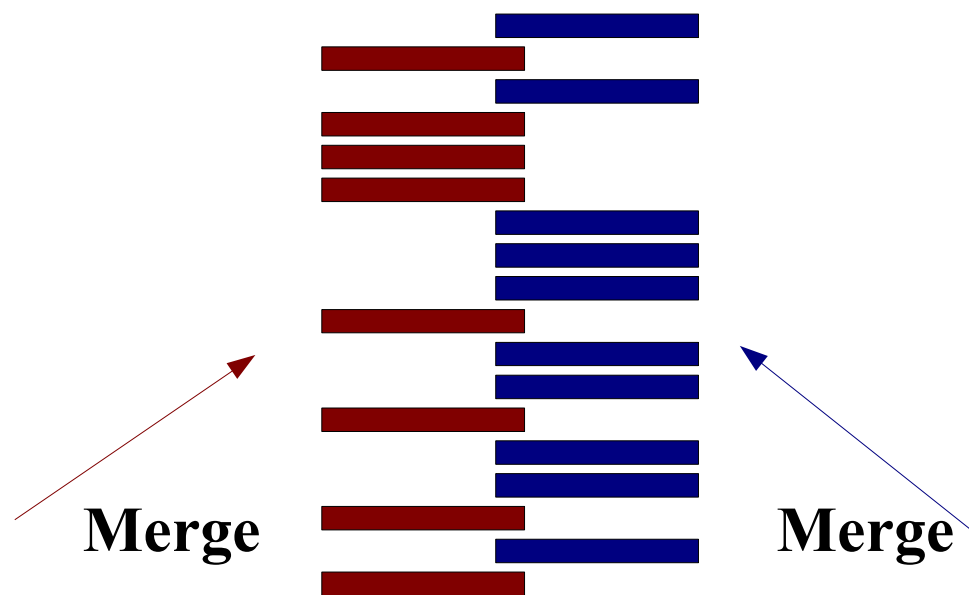
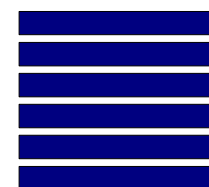
Sort



Oppgave post

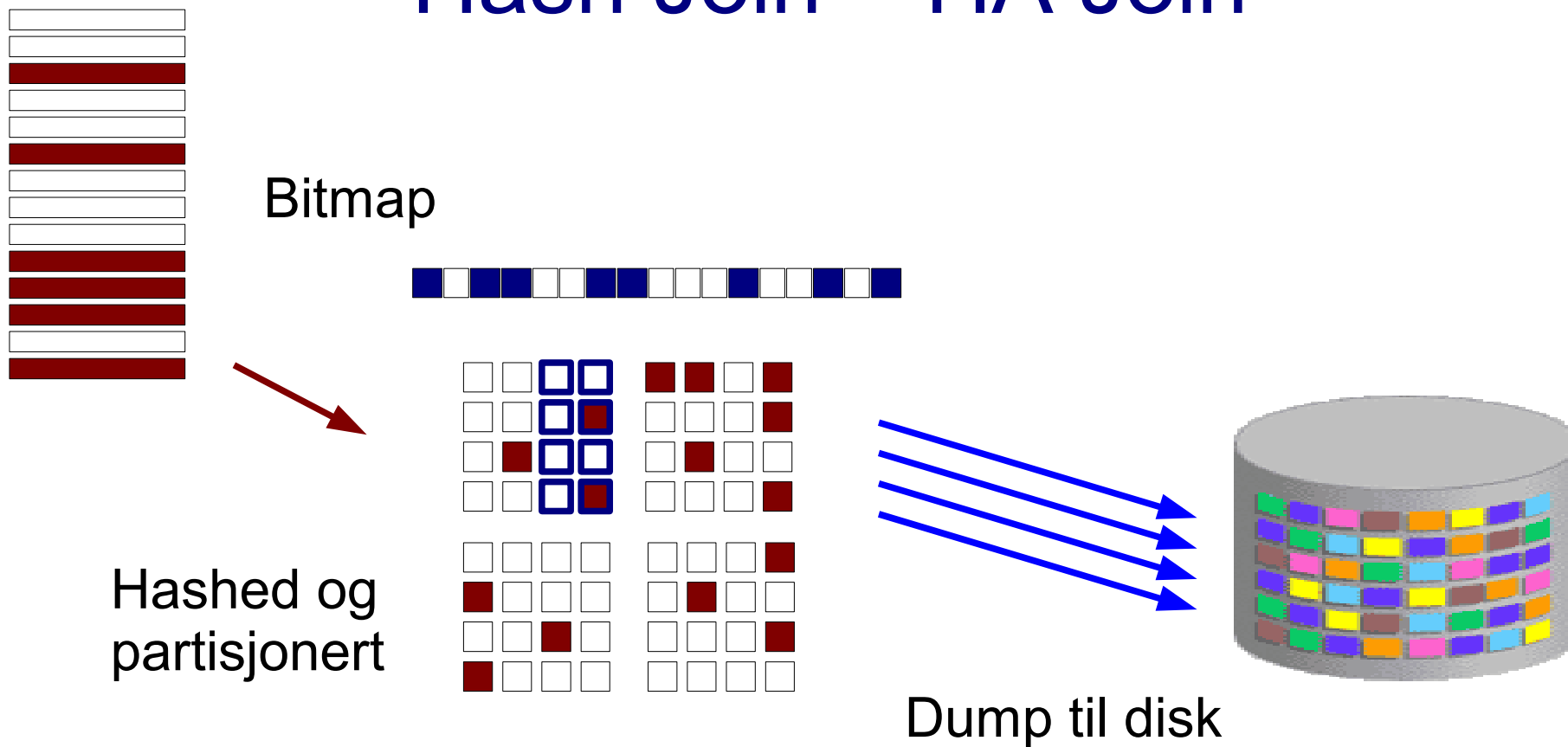


Sort



Join cost = (cost of accessing outer table + outer sort cost)
 (cost of accessing inner table * inner sort cost)

Hash Join – HA Join



Join cost = cost of accessing outer table +
 cost of building hash table +
 cost of accessing inner table

Nested Loop Join

NL Join

```

Outer table: Card: 770452.70  Cost: 2766.57  Resp: 2766.57
Degree: 16  Bytes: 58
Inner table: OPPGAVE  Alias: 0
Access Path: TableScan
Access Path: index (UniqueScan)
Index: OPP_OPP_UK2
Access Path: index (RangeScan)
Index: OPP_OPP_UK
Access Path: index (AllEqUnique)
Index: OPP_OPP_UK2
resc_io: 2.00  resc_cpu: 16553
ix_sel: 1.1497e-09  ix_sel_with_filters: 1.1497e-09
NL Join: Cost: 1583028.22  Resp: 109932.52  Degree: 16
Cost_io: 1579210.00  Cost_cpu: 21323120916
Resp_io: 109667.36  Resp_cpu: 1480772286

```



Best NL cost: 109932.52

```

resc: 1583028.22  resc_io: 1579210.00  resc_cpu: 21323120916
resp: 109932.52  resp_io: 109667.36  resp_cpu: 1480772286

```

Join Card: 770452.65 =

```

outer (770452.70) * inner (4967535.68) * sel (2.0131e-07)

```

Sort Merge Join

SM Join

Outer table:

resc: 39838.57 card 770452.70 bytes: 58 deg: 16 resp:
2766.57

Inner table: OPPGAVE Alias: 0

resc: 2935.67 card: 4967535.68 bytes: 72 deg: 16 resp:
203.87

using dmeth: 2 #groups: 1

...

SM join: Resc: 95230.00 Resp: 3030.14 [multiMatchCost=0.00]

SM cost: 3030.14

resc: 95230.00 resc_io: 92197.00 resc_cpu: 16938007749
resp: 3030.14 resp_io: 2849.17 resp_cpu: 1010684793

...

SM join: Resc: 95230.00 Resp: 3030.14 [multiMatchCost=0.00]

SM cost: 3030.14

resc: 95230.00 resc_io: 92197.00 resc_cpu: 16938007749
resp: 3030.14 resp_io: 2849.17 resp_cpu: 1010684793



Hash Join

HA Join

Outer table:

resc: 39838.57 card 770452.70 bytes: 58 deg: 16 resp:
2766.57

Inner table: OPPGAVE Alias: 0

resc: 2935.67 card: 4967535.68 bytes: 72 deg: 16 resp:
203.87

using dmeth: 2 #groups: 1

Cost per ptn: 7.35 #ptns: 16

hash_area: 0 (max=8192) buildfrag: 206 probefrag:
1592 passes: 1

Hash join: Resc: 42891.88 Resp: 2977.79

[multiMatchCost=0.00]

HA cost: 2977.79

resc: 42891.88 resc_io: 41028.00 resc_cpu: 10408967104

resp: 2977.79 resp_io: 2849.17 resp_cpu: 718282439

Best: JoinMethod: Hash

Cost: 2977.79 Degree: 16 Resp: 2977.79

Card: 770452.65 Bytes: 130



Sammenfatning

- Noen få «store» kritiske tabeller
- Forsiktighet ved oppgradering
- Ha en formening om svarstidene
- Sjekk statistikken
- Sjekk resultatet
- Explain plan / autotrace
- «Event 10053» siste utvei

