



AXIS INTO
ICT

VERBETEREN OF VERMIJDEN

**Over de invloed van IO op
performance**



- Erik Swinkels, 40 jaar
- Meer dan 20 jaar ervaring als Oracle DBA (vanaf versie 5)
- Systeembeheer uitgevoerd op Unix, VMS, Novell en Windows
- Enkele jaren Oracle server support
- Gespecialiseerd in architectuur, backup/recovery, performance en beheer
- Groot voorstander van 'zo eenvoudig mogelijk', moeilijk is misschien leuk, maar vaak ook duur en onnodig lastig.



- Introductie
- Verbeteren
 - Wat bepaald de snelheid van IO
 - Effecten van disk IO op performance
- Voorkomen
 - Hoe zorg ik voor minder IO
 - Oplossingen uit de praktijk



Performance problemen met databases zijn meestal IO gerelateerd

Tijdens deze sessie:

- Hoe werkt een disk
- Wat is een database IO
- Waar (en hoe) moet ik zoeken naar een oplossing



- Storage als onderdeel van performance tuning is compleet nieuw voor mij
- Ik heb **niets** te zeggen over de storage die ik aangeboden krijg, het is zoals het is
- Ik probeer het wel uit te leggen maar ze zeggen dat ik er niets van begrijp/geen gelijk heb
- RAID 5 maakt tegenwoordig niets meer uit
- Ik weet hoeveel IOPS de database nodig heeft gedurende de piek
- Ik weet eigenlijk alles al maar ben benieuwd wat je te vertellen hebt ;-)



AXIS INTO
ICT

VERBETEREN





- Op 13 september 1956 introduceerde IBM de eerste harde schijf: **R**andom **A**ccess **M**ethod of **A**ccounting and **C**ontrol. De *RAMAC* bestond uit 50 gestapelde magnetische schijven met een diameter van 61 cm (24 inch). Er waren twee speelkoppen.

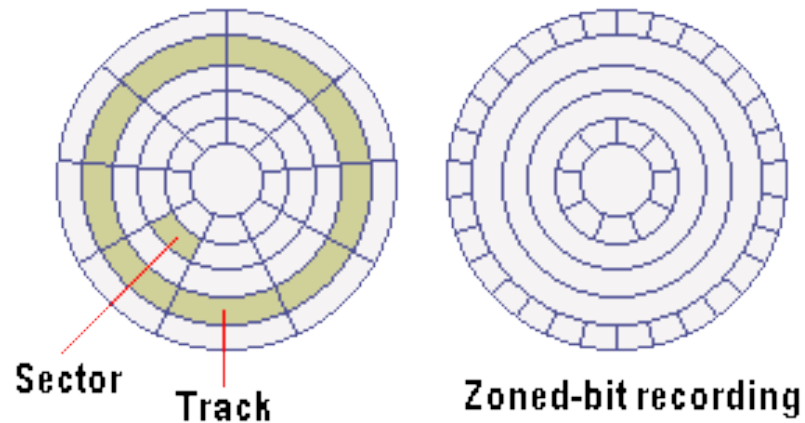
Voor die tijd werd alle opslag gedaan op tapes

- Sinds de introductie van de RAMAC groeide elk jaar de opslagcapaciteit van harde schijven, terwijl de omvang steeds kleiner werd.
- In 1980 brengt IBM de 3380, de eerste Gigabyte harddisk. Formaat koelkast. 250Kg en een prijsje van \$40,000
- Inmiddels is afgelopen maand de eerste 4TB disk op de markt verschenen.

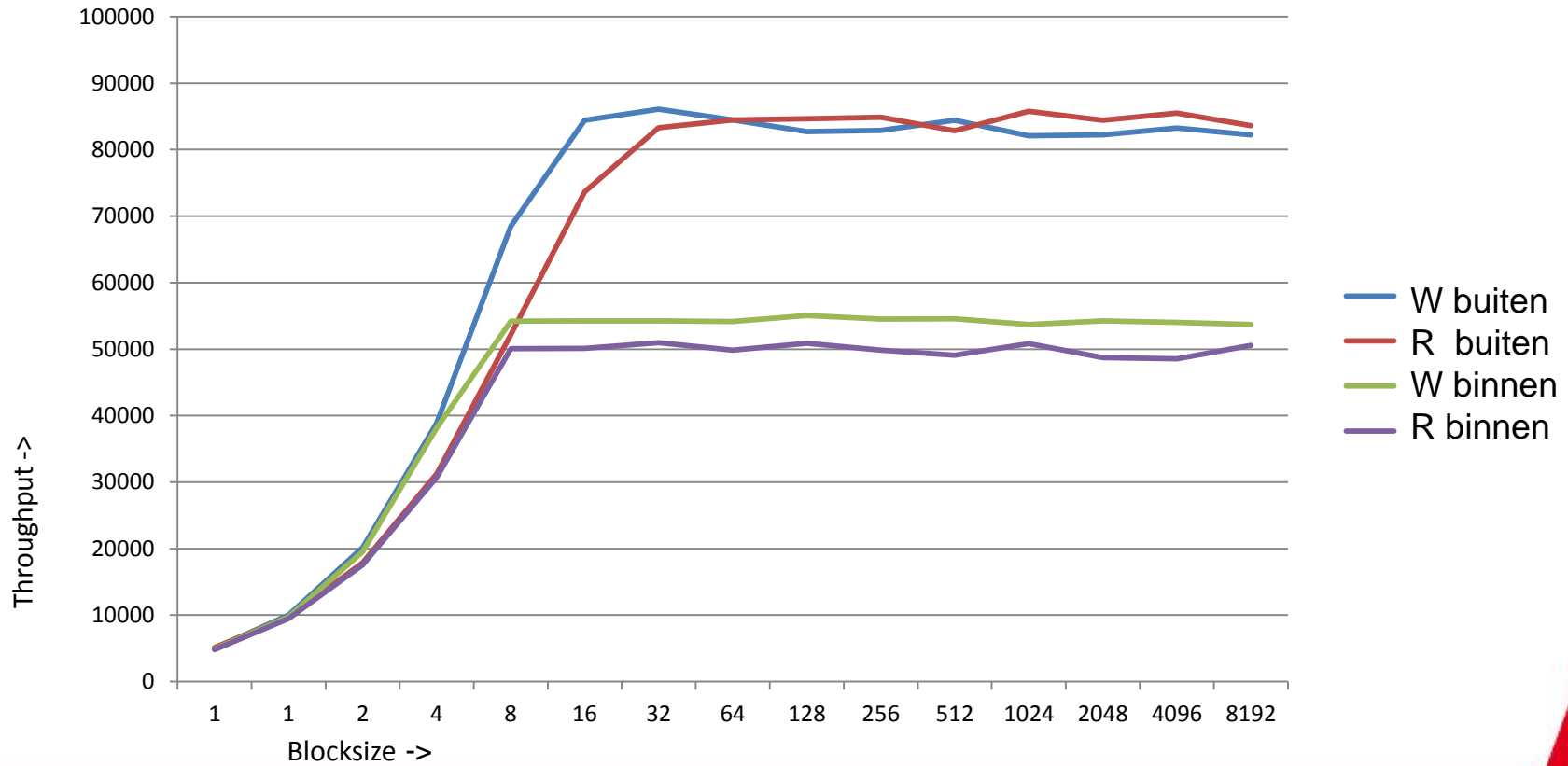


- De feitelijke magnetische drager
- Meer platters-> meer gewicht-> meer stroom
- De opslagdichtheid voor platters wordt uitgedrukt in “bits per vierkante inch”
- Huidige techniek schaal tot 1Tb (nu op ong. 700Gb), “nieuwste” techniek tot tientallen Tb’s

- Hoe ?



- Binnenkant of buitenkant ?





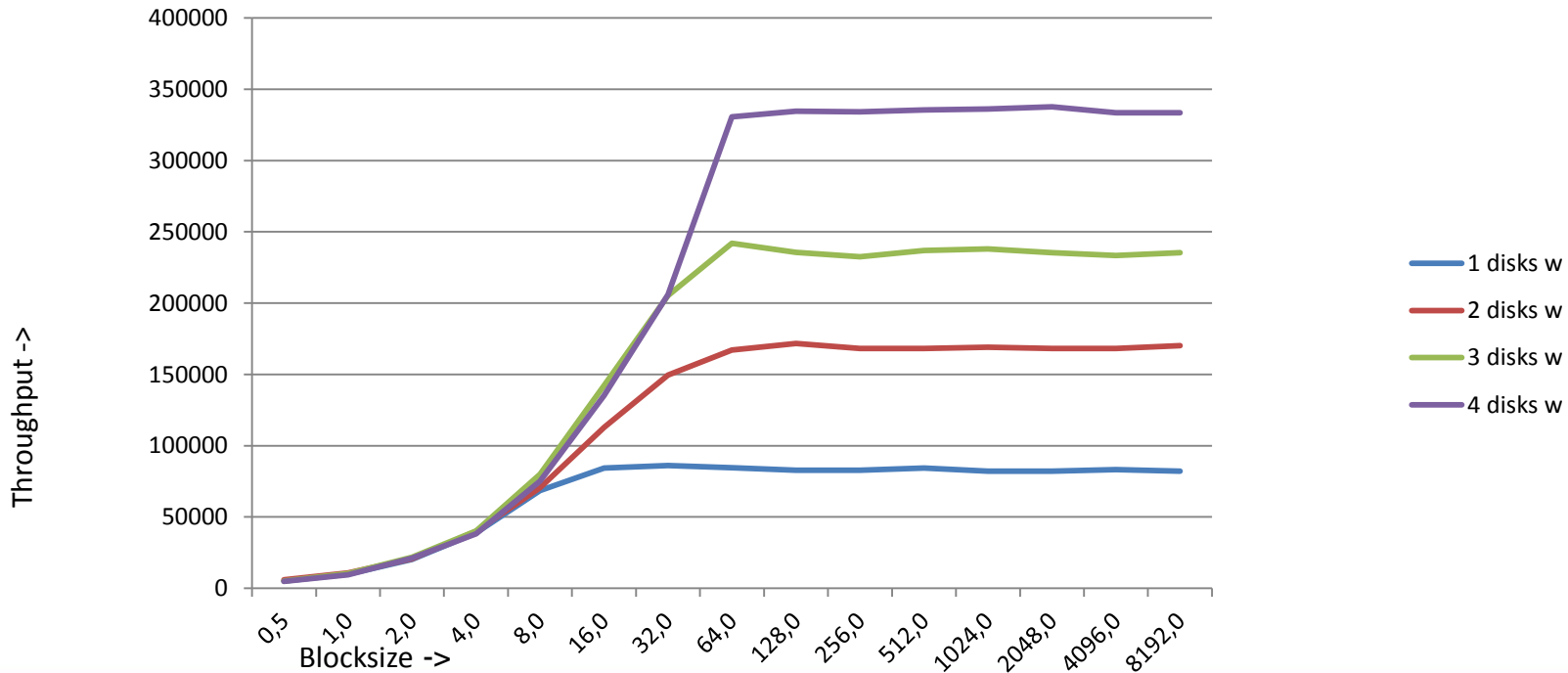
- Disken worden groter maar niet gelijkwaardig sneller. (een groeiend probleem)
- Random reads worden een steeds groter probleem
- IOPS zijn beperkt, throughput is beperkt
- Een steeds grotere vraag naar capaciteit (met name DWH omgevingen)

Een harddisk is eigenlijk maar een heel simpel ding.



STRIPING

één disk heeft een maximum, wat als ik meerdere disken een stukje van het werk laat opknappen.



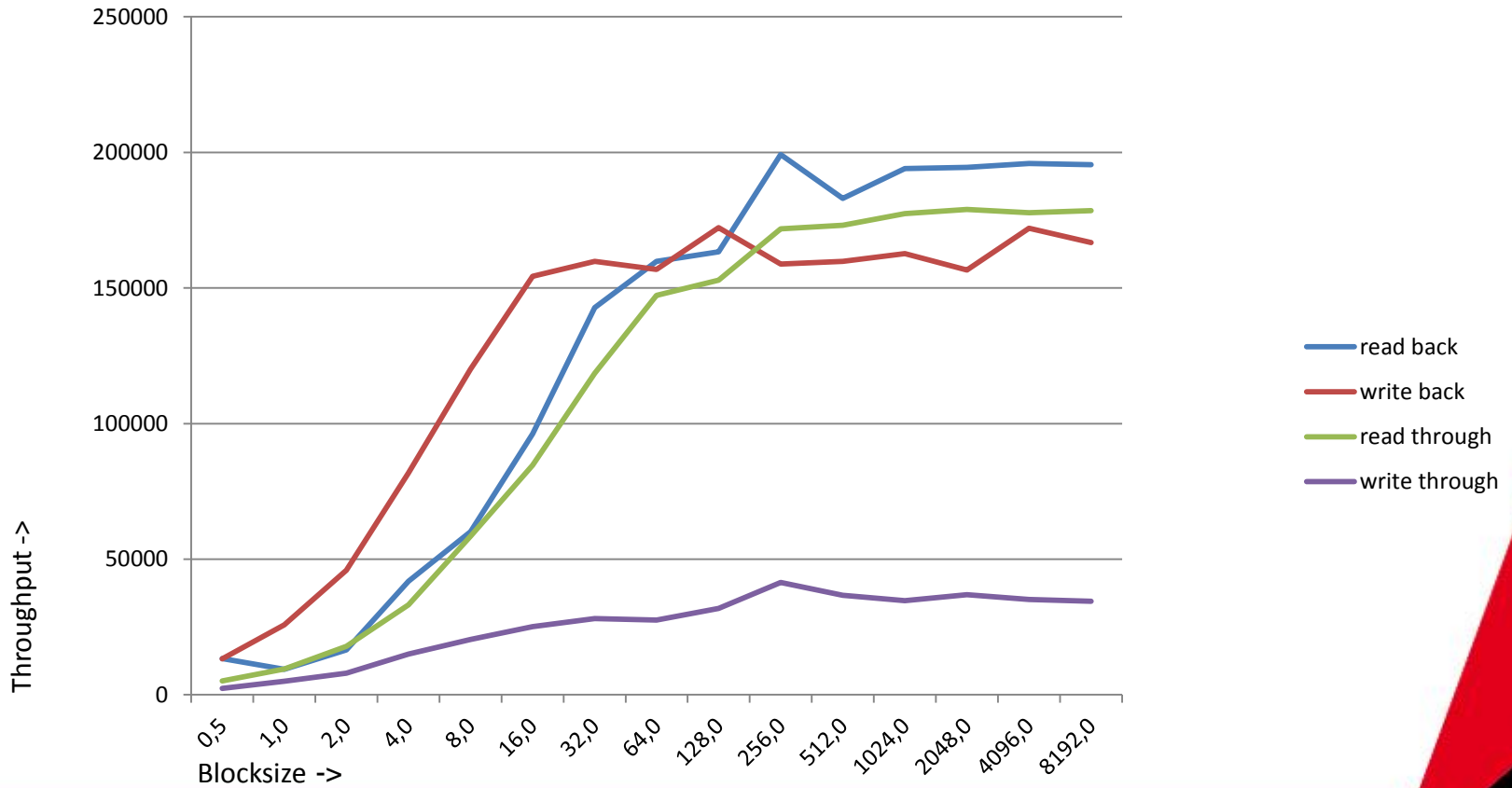


Redundant Array of Independent Disks

- Ook wel Inexpensive en Drives
- Methode voor verbeteren prestaties en/of beveiliging van de data
- Belangrijkste versies:
 - RAID-0 ->striping
 - RAID-1 ->mirroring
 - RAID-5 ->parity 1 disk
 - RAID-6 ->parity 2 disken
 - JBOD -> Just a Bunch Of Disks
- Combinaties zijn mogelijk en vaak gewenst vanwege performance eisen



- Wie gebruikt RAID 5/6 ?





- Een manier om de ‘traagheid’ van disken op te vangen.
- ZEER VEEL Storage managers denken **onterecht** dat het DE oplossing is
- RAM geheugen is vele malen sneller dan een disk
- “met genoeg cache maakt RAID-5 niet uit”
- CacheCramming -> Orion

Een vergelijking:

- de Quadcore XEON W5590 heeft 3 levels cache:
- Level 1 : 256KB -> 8x32KB (4x 32Kb voor Instructies en 4x 32Kb voor Data)
- Level 2 : 1MB -> 4x256KB
- Level 3 : 8MB -> 4x2MB
- Waarom in deze verhouding? Hoe zit dat dan met Oracle’s eigen buffercache, filesystemcache, disk cache, san cache ?



AXIS INTO
ICT

DISK VOL





Gegevens :

- Grote tabel (524032 blokken)
- Slechte vulling
- kleine rijen (1 kolom)
- set timing on
- geen indexen
- naam afhankelijk van locatie op disk



```
Select count(*) from achter;
```

```
TIJD: 00:01:17.98
```

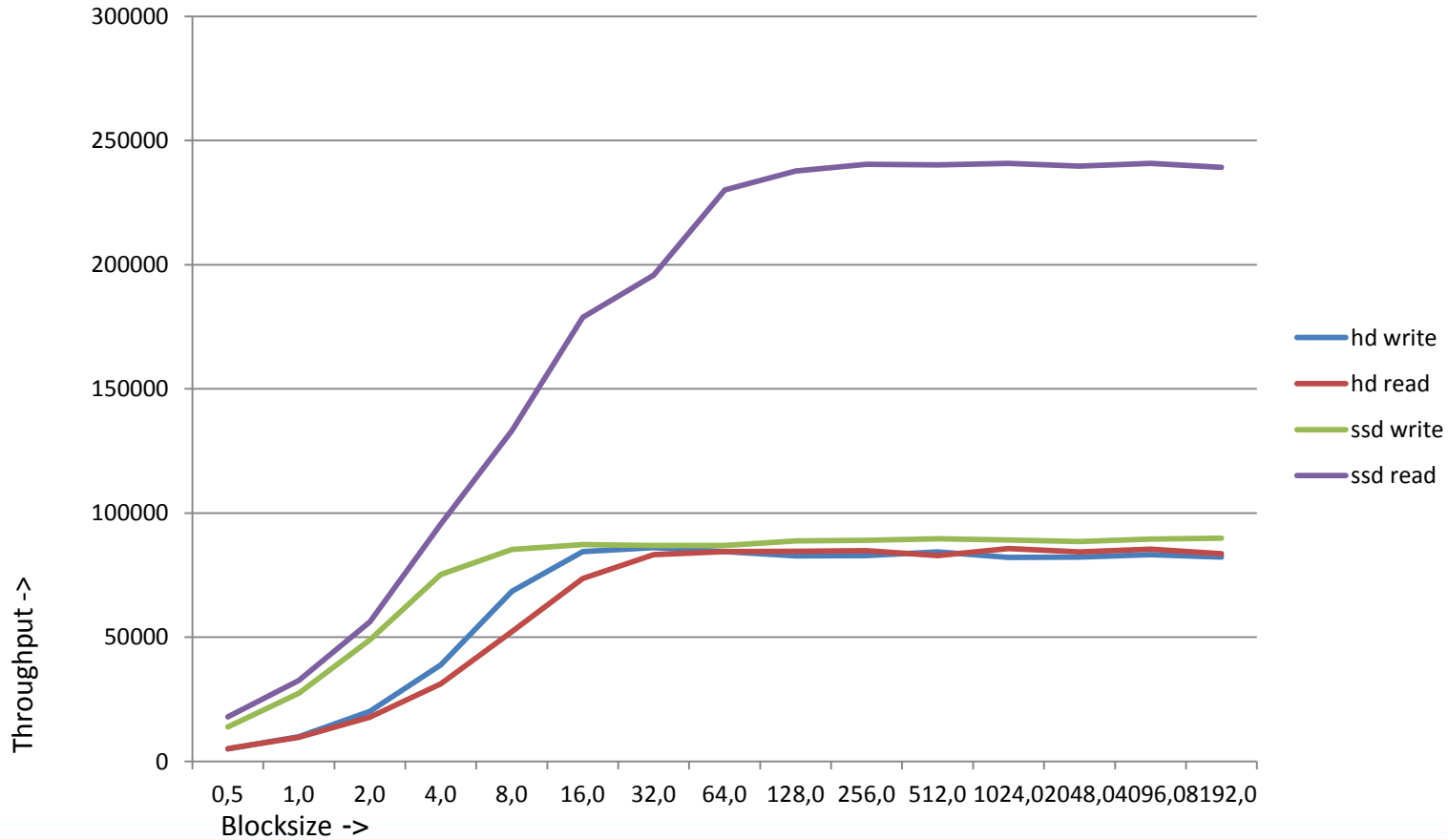
```
Select count(*) from voor;
```

```
TIJD: 00:00:39.53
```

```
Vershil: ruim 39 seconden !
```



Een SSD is gewoon een disk, maar zonder draaiende en bewegende delen !





```
Select count(*) from solid;
```

```
TIJD: 00:00:16.85
```

```
Vershil: 61 en 23 seconden !
```



- Flash chips worden gebruikt in SSD's
- Direct Flash oplossingen zitten dichterbij de CPU om de 'traagheid' van diskcontrollers te omzeilen.
- Mogelijk geworden door de PCI-e interface
- PCI-e heeft met versie 2.x inmiddels een snelheid van 500 MB/s per lane
Een PCI-e flashkaart met 8x interface kan dus 4GB/s opleveren.
- PCI-e versie 3 zal dit verdubbelen



AXIS INTO
ICT

VERMINDEREN





- Aan de techniek zit een grens
- Techniek is KOSTBAAR !
- Oplossing moeten we dus **VOORAL** zoeken in de software
- Wat voor oplossingen zijn er mogelijk..

Soms is de oplossing eenvoudiger dan je denkt...

Als er een limiet zit op wat de storage kan leveren
beperk dan de te leveren hoeveelheid



- Minimaliseer physical reads
 - Bijvoorbeeld door vergroten buffer cache
 - > Maar dat is niet genoeg, want meer gets is ook meer cpu
- Minimaliseer dus zoveel mogelijk ook de logical reads
- Stel jezelf de vraag: Wat zijn we eigenlijk aan het doen?
 - En waarom?
- Verbeter prestaties hardware (betere hardware, meer ijzer kopen)



- FULL TABLE SCAN
- SLECHTE EXECUTIEPLANNEN
- SORTERINGEN OP DISK
- OVERMATIGE REDO (veel log switches)



- Oracle Enterprise Manager
 - Diagnostic pack (licentie)
 - AWR en ASH
 - Tuning pack (licentie)
- SQL Developer
- StatsPack

- 3rd party tools, waarvan TOAD het meest bekende voorbeeld is
- Gezond verstand



- Veel ervaring is wel handig (dus regelmatig ..)
- Logisch nadenken over elementaire zaken
- Stel jezelf de goede vragen: “Waarom is er zoveel gelezen ?



- KUNNEN WE DE HELE DAG OVER PRATEN

- Beperken full table scans

- FAT Indexing

- Alle relevante data in de index

- Select naam,adres van tabel_met_veel_kolommen

- Index op naam,adres maakt table access overbodig

- En in het verlengde.... Haal alleen op wat je moet hebben.



AXIS INTO
ICT

Praktijk voorbeeld

Traag laden pagina's op website





```
SELECT length(Dat) Len, LstModDtm LstModDtm
FROM AsstAb
WHERE LOWER(Pad) = '/publish/library/1181/banner.jpg'
```

Snapshot Comment	Snap Id	Snap Time	Sessions	Curs/Sess
~~~~~	-----	-----	-----	-----
Begin Snap:	861	20-Jun-11 07:00:03	60	4.1
End Snap:	961	20-Jun-11 17:00:00	88	6.2
Elapsed:	599.95 (mins)	Av Act Sess:	1.1	
DB time:	641.34 (mins)	DB CPU:	167.10 (mins)	



Top 5 Timed Events

~~~~~

| Event | Waits | Time (s) | Avg wait (ms) | %Total Call Time |
|-----------------------------|------------|----------|---------------|------------------|
| CPU time | | 10,021 | | 41.5 |
| direct path read | 11,274,830 | 8,398 | 1 | 34.8 |
| log file parallel write | 123,094 | 1,347 | 11 | 5.6 |
| control file parallel write | 39,160 | 1,101 | 28 | 4.6 |
| log file sync | 89,266 | 906 | 10 | 3.7 |



| Physical Rds | Executions | Rds per Exec | %Total | CPU Time (s) | Elapsd Time (s) | Hash Value |
|----------------------------------------------------------------------------------------------------------------|------------|--------------|--------|--------------|-----------------|------------|
| 5,909,046 | 2,769 | 2,134.0 | 4.5 | 80.61 | 337.31 | 857815251 |
| Module: w3wp.exe | | | | | | |
| SELECT length(Dat) Len,LstModDtm LstModDtm FROM AsstTab WHERE LOWER(Pad) = '/publish/library/1181/banner.jpg' | | | | | | |
| 4,438,720 | 2,080 | 2,134.0 | 3.4 | 59.55 | 342.18 | 2327435175 |
| Module: w3wp.exe | | | | | | |
| SELECT length(Dat) Len,LstModDtm LstModDtm FROM AsstTab WHERE LOWER(Pad) = '/publish/library/1641/twitter.gif' | | | | | | |



```
SELECT STATEMENT ALL_ROWS Cost: 584  
  1 TABLE ACCESS FULL TABLE TRT.ASSTAB  
    Cost:584 Bytes: 33.390 Cardinality: 210
```

1. Redenen voor full tablescan
2. Performance daalt met groei data



1. Literals in plaats van binds
2. Totaal van ongeveer 80 miljoen I/O's gedurende statspack periode
3. Enkele duizenden executies per query

Tabeldefinitie: 2 kolommen PAD en NAAM
Met een Primary Key op Pad

Vraag 1: Hebben we een probleem?

Vraag 2: Wat moet er gebeuren ?



```
CREATE UNIQUE INDEX IX_LOWER_PAD ON ASSTAB  
(LOWER("PAD"))
```

```
SELECT STATEMENT ALL_ROWS Cost: 2  
  2 TABLE ACCESS BY INDEX ROWID TABLE  
    ASSTAB Cost: 2 Bytes: 157 Cardinality: 1  
  1 INDEX UNIQUE SCAN INDEX (UNIQUE)  
    IX_LOWER_PAD Cost: 1 Cardinality: 1
```



| Tablespace | | | | | CPU | Elapsd | | |
|------------|------------|---------|------------|---------|--------|----------|-------|--------|
| | Av | Av | Av | | Av | Buffer | Av | Buf |
| | Reads | Reads/s | Rd(ms) | Blks/Rd | Writes | Writes/s | Waits | Wt(ms) |
| IN_DATA | 11,139,572 | 309 | 0.0 | 11.6 | 7,382 | 0 | 521 | 73.1 |



AXIS INTO
ICT

**Nog meer over
indexen**





Zeer grote tabel (200.000+ rijen)

Query:

....

AND SAD\_SL\_TRANS\_DTTM IS NULL

Plan

SELECT STATEMENT ALL\_ROWS Cost: 1,887 Bytes: 131 Cardinality: 1

1 TABLE ACCESS FULL TABLE SAD\_SL\_TRNS\_INF Cost: 1,887 Bytes: 131 Cardinality: 1

CREATE INDEX SAD\_SL\_TRNS\_INF\_XYZ
ON SAD\_SL\_TRNS\_INF (SAD\_SL\_TRANS\_DTTM, 0)

Plan

SELECT STATEMENT ALL\_ROWS Cost: 2

2 TABLE ACCESS BY INDEX ROWID TABLE SAD\_SL\_TRNS\_INF Cost: 2 Bytes: 131 Cardinality: 1

1 INDEX RANGE SCAN INDEX SAD\_SL\_TRNS\_INF\_XYZ Cost: 2 Cardinality: 1



- Optimizer is heel slim....?
- Maar we moeten 'm wel helpen

| Plan | |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| SELECT STATEMENT ALL_ROWS Cost: 2 Bytes: 161 Cardinality: 1 | |
| 2 | TABLE ACCESS BY INDEX ROWID TABLE SYSADM.PS_CLASS_TBL Cost: 6 Bytes: 161 Cardinality: 1 |
| 1 | INDEX RANGE SCAN INDEX SYSADM.PSACCLASS_TBL Cost: 2 Cardinality: 1 |

| Plan | |
|-------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| SELECT STATEMENT ALL_ROWS Cost: 2 Bytes: 160 Cardinality: 1 | |
| 2 | TABLE ACCESS BY INDEX ROWID TABLE SYSADM.PS_CLASS_TBL Cost: 2 Bytes: 160 Cardinality: 1 |
| 1 | INDEX UNIQUE SCAN INDEX (UNIQUE) SYSADM.PS_CLASS_TBL_PERF_ERIK1_TMP Cost: 1 Cardinality: 1 |

- System statistics
- Single block vs Multi block IO



- ALTER TABLE MOVE
- ALTER INDEX REBUILD
- Datapump (of old school IMP/EXP)

Na een export -> import van de testtabel van 524032 blokken blijven er nog 7936 over

```
Select count(*) from ....;
```

```
TIJD ACHTER: 00:00:17.68
```

```
TIJD VOOR : 00:00:05.06
```

```
TIJD SSD : 00:00:00.86
```



CREATE TABLE XXX
COMPRESS FOR ALL OPERATIONS

Select count (\*) from ...;

TIJD ACHTER: 00:00:02.00

TIJD VOOR : 00:00:01.06

TIJD SSD : 00:00:00.76

2<sup>e</sup> maal: 00:00:00.10



- Techniek om tabel in stukjes op te delen
- Beperken gevolgen full table scan
- Oplossing voor contentie problemen
- Met elke nieuwe Oracle versie meer mogelijkheden
 - Partition pruning
 - Beperken van data
 - Life cycle management



Maak gebruik van de techniek...

- db file sequential read hoog
- Gedrag te beïnvloeden

Opties: KEEP, NONE, DEFAULT

- create table TEST storage(**flash\_cache keep**)



- Result set caching
- Adaptive cursor sharing
- System statistics, optimizer
- ASM
- Parallel Query

- Hints
- Underscore parameters



AXIS INTO
ICT

Praktijk voorbeeld

Grote database voor web applicatie





- Het lukt niet meer om voor 07:00 het systeem geladen te krijgen, met ernstige gevolgen voor de performance.
- Helder probleem en goed punt om te starten is een AWR geduren de laadperiode.

| Event | Waits | Time(s) | Avg Wait(ms) | % Total Call Time | Wait Class |
|--------------------------|-----------|---------|--------------|-------------------|------------|
| db file sequential read | 3,196,007 | 68,237 | 21 | 34.0 | User I/O |
| PX Deq Credit: send blkd | 7,962,015 | 41,155 | 5 | 20.5 | Other |
| db file scattered read | 858,887 | 24,590 | 29 | 12.3 | User I/O |
| direct path read | 243,146 | 24,478 | 101 | 12.2 | User I/O |
| direct path read temp | 868,256 | 13,199 | 15 | 6.6 | User I/O |



| Tablespace | Reads | Av Reads/s | Av Rd(ms) | Av Blks/Rd | Writes | Av Writes/s | Buffer Waits | Av Buf Wt(ms) |
|------------|-----------|------------|-----------|------------|---------|-------------|--------------|---------------|
| LLL_DATA | 2,876,117 | 67 | 77.44 | 17.43 | 228,219 | 5 | 44,811 | 79.51 |
| TEMP | 1,061,894 | 25 | 57.75 | 9.42 | 394,133 | 9 | 10 | 3.00 |
| LLL_INDEX | 914,207 | 21 | 18.42 | 1.35 | 41,426 | 1 | 1,559 | 107.58 |
| REC_INDEX | 224,577 | 5 | 44.48 | 1.00 | 2,334 | 0 | 994 | 53.77 |
| REC_DATA | 156,432 | 4 | 38.85 | 1.00 | 12,009 | 0 | 1,104 | 47.74 |

- Simpel probleem, IO is traag dus MEER ijzer
- Iets eleganter is om te kijken waar het vandaan komt
- Ook uit AWR

| Physical Reads | Executions | Reads per Exec | %Total | CPU Time (s) | Elapsed Time (s) | SQL Id | SQL Module | SQL Text |
|----------------|------------|----------------|--------|--------------|------------------|---------------------------------------------------|----------------------------------|---------------------------------------------|
| 21,680,004 | 1 | 21,680,004.00 | 34.86 | 1380.98 | 16758.48 | 6jw5q9wf5
anqm | Redwood
job agent
13149430 | declare
CMDFILE
varchar2(2
000)... |
| 17,174,755 | 1 | 17,174,755.00 | 27.62 | 573.19 | 10676.69 | b1mf1fkq4
9fgp | Redwood
job agent
13149430 | begin
product_co
mpare... |
| 10,176,986 | 6 | 1,696,164.33 | 16.37 | 405.70 | 7923.89 | 373bbkdct
d1dm | Redwood
job agent
13153471 | WITH
TYPES AS
(SELECT
/*+ mate... |



AXIS INTO
ICT

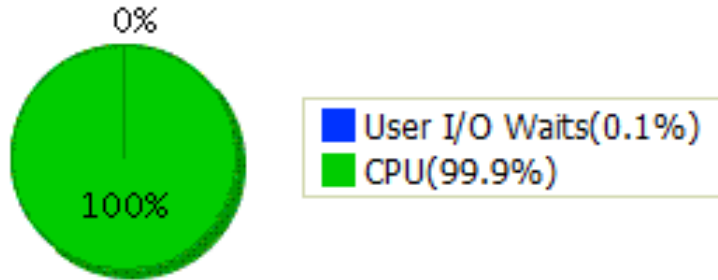
Praktijk voorbeeld

Echt slechte query





Activity By Waits



Execution Statistics

| | Total | Per Execution | Per Row |
|--------------------|---------------|------------------|---------|
| Executions | 1 | 1 | n/a |
| Elapsed Time (sec) | 162,920.43 | 162,920.43 | n/a |
| CPU Time (sec) | 162,754.22 | 162,754.22 | n/a |
| Buffer Gets | 3,690,569,635 | 3,690,569,635.00 | n/a |
| Disk Reads | 72,904 | 72,904.00 | n/a |
| Direct Writes | 0 | 0.00 | n/a |
| Rows | 0 | 0.00 | n/a |
| Fetches | 0 | 0.00 | n/a |



AXIS<sup>INTO</sup>ICT

Even tellen..

3.690.569.635 gets..

Blocksize 8k

$(3.690.569.635 * 8) / 1024 / 1024 = 28175 \text{ Gb...}$

And counting..



AXIS INTO
ICT

Praktijk voorbeeld

Tuning van Peoplesoft omgeving





| Buffer Gets | Execution
s | Gets per
Exec | %Total | CPU Time
(s) | Elapsed
Time (s) | SQL Id | SQL
Module | SQL Text |
|-------------|----------------|------------------|--------|-----------------|---------------------|------------------------------------|-----------------------|----------------------------------------------------|
| 14,811,855 | 77,816 | 190.34 | 8.04 | 246.69 | 262.92 | brvugtx
6k318g | PSDSTS
RV@acc
- | SELECT
DISTINC
T(RC.CL
ASSID)
FR... |
| 14,727,083 | 1,149,460 | 12.81 | 8.00 | 251.46 | 264.90 | 2ft57tgt
a7429 | PSAPPS
RV@acc | SELECT
CRSE_G
RADE_O
FF,
GRADIN
... |
| 7,773,351 | 971,722 | 8.00 | 4.22 | 50.84 | 55.01 | 6qcfvtyb
ns467 | PSAPPS
RV@acc | SELECT
INSTITU
TION,
DESCR,
COU... |



Segments by Logical Reads

- Total Logical Reads: 184,170,745
- Captured Segments account for 74.6% of Total

| Owner | Tablespace Name | Object Name | Subobject Name | Obj. Type | Logical Reads | %Total |
|--------|-----------------|-------------|----------------|-----------|---------------|--------|
| SYSADM | PSINDEX | PS_PSROL | ECLASS | INDEX | 19,100,032 | 10.37 |
| SYSADM | PTTBL | PSROLEDE | FN | TABLE | 10,102,576 | 5.49 |
| SYSADM | PSINDEX | PS_SNS_C | RSE_OFFE
R | INDEX | 9,824,672 | 5.33 |
| SYSADM | PSINDEX | PS2TERM_ | TBL | INDEX | 6,926,208 | 3.76 |
| SYSADM | SAAPP | PS_SNS_C | RSE_DTL | TABLE | 6,782,656 | 3.68 |



```
/* Formatted on 9/12/2011 6:08:06 PM (QP5 v5.126.903.23003) */
SELECT DISTINCT (RC.CLASSID)
  FROM PSROLEUSER RU, PSROLECLASS RC
 WHERE RU.ROLEUSER = :1 AND RU.ROLENAME = RC.ROLENAME
    AND NOT EXISTS
      (SELECT 'X'
        FROM PSROLEUSER RU2,
             PSROLEDEFN R2,
             PSROLECLASS RC2,
             PSCCLASSDEFN C2
       WHERE RU2.ROLEUSER = RU.ROLEUSER
            AND RU2.ROLENAME = RU.ROLENAME
            AND RU2.ROLENAME = R2.ROLENAME
            AND R2.ROLESTATUS = 'A'
            AND R2.ROLENAME = RC2.ROLENAME
            AND RC2.CLASSID = C2.CLASSID
            AND C2.CLASSID = RC.CLASSID)
```



Plan

SELECT STATEMENT ALL\_ROWS Cost: 19

13 SORT UNIQUE Cost: 19 Bytes: 1,696 Cardinality: 16

12 HASH JOIN ANTI Cost: 18 Bytes: 1,696 Cardinality: 16

3 HASH JOIN Cost: 7 Bytes: 976 Cardinality: 16

1 INDEX RANGE SCAN INDEX (UNIQUE) SYSADM.PS\_PSRROLEUSER Cost: 3 Bytes: 189 Cardinality: 7

2 INDEX FAST FULL SCAN INDEX (UNIQUE) SYSADM.PS\_PSRROLECLASS Cost: 3 Bytes: 53,176 Cardinality: 1,564

11 VIEW VIEW VW\_SQ\_1 Cost: 11 Bytes: 810 Cardinality: 18

10 NESTED LOOPS Cost: 11 Bytes: 1,836 Cardinality: 18

8 HASH JOIN Cost: 11 Bytes: 1,512 Cardinality: 18

6 HASH JOIN Cost: 8 Bytes: 350 Cardinality: 7

4 INDEX RANGE SCAN INDEX (UNIQUE) SYSADM.PS\_PSRROLEUSER Cost: 3 Bytes: 189 Cardinality: 7

5 TABLE ACCESS FULL TABLE SYSADM.PSRROLEDEFN Cost: 4 Bytes: 11,661 Cardinality: 507

7 INDEX FAST FULL SCAN INDEX (UNIQUE) SYSADM.PS\_PSRROLECLASS Cost: 3 Bytes: 53,176 Cardinality: 1,564

9 INDEX UNIQUE SCAN INDEX (UNIQUE) SYSADM.PS\_PSRCLASSDEFN Cost: 0 Bytes: 18 Cardinality: 1



- Query goed bestuderen
- Zeer lastig probleem (o.a. vanwege de subselect)
- Groei van de data gaat zeker grotere problemen geven
- Support.oracle.com -> zoeken op query of probleem
 - Feature



- NOOIT RAID-5 / RAID-6 gebruiken (tenzij performance niet belangrijk is 😊)
- Hoe meer spindels (disken) hoe beter (tien 4Gb disken van 15 jaar geleden in RAID-0 zijn echt veel sneller dan een ultra moderne disk van 500Gb)
- Gebruik nooit meer dan 80% van een disk !!
- Cache is leuk, maar nooit DE oplossing
- Denk **goed** na over het gedrag van je systeem. Een OLTP omgeving heeft heel andere eisen dan een DWH. Zorg dat je de storage omgeving daar op aanpast
- Bedenk goed dat een storage manager mogelijk veel weet van storage maar niet van Oracle. Neem dus niet altijd alles zomaar aan 😊



AXIS <sup>INTO</sup> ICT

Vragen

?



AXIS<sup>INTO</sup>
ICT

Hartelijk dank

Contactinfo:

ict@axisinto.nl