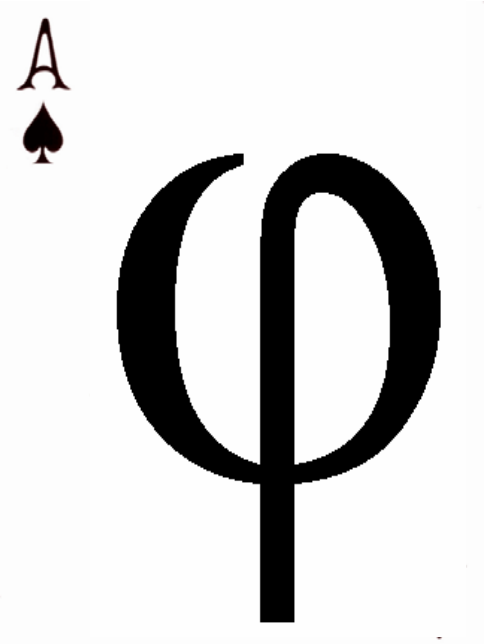
Still using Ratios

(But I think I know what I am doing)

Piet de Visser

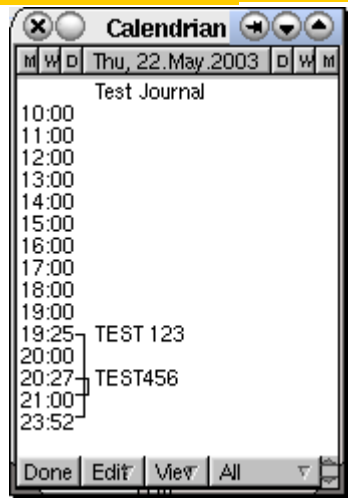
The Simple Oracle DBA



Keywords: Ratios, Capacity, Performance, mostly OLTP.
Also: Simplicity.

Agenda (45 minutes)

logica
SOLUTIONS THAT MATTER



It is about Efficiency, OLTP (Hah …)

Efficiency, Concurrency, Scalability…

- **High Result per unit of work (a ratio...)**
- **Little effort per task (a ratio…)**

Some Numbers (Do try this at home)

- **Will go from detail to general.**

Discussion (Do Challenge!)

Background - Requirements

logica

SOLUTIONS THAT MATTER

- 40M “documents“, 500+ users
  - Can we process inside a month ?
  - Every month ? With less resources ?
- 1800 msgs/sec
  - Can it run on given hardware ?
  - Can it run on “Any hardware” ?
- 25M transactions per hr, in <4 hrs.
  - only on 1st Mondays (or exceptional days) - scary
  - (7000 t/sec => 7 t/ms @10ms/t that is 70 CPUs)

Background – the jargon…

- **Keywords:**
  - **Performance** (speedy response)
  - **Capacity** (do we have enough …)
  - **Concurrency** (can we run all … at once)
  - **Scalability** (what if workload triples …)
- **Can limited tests predict these…?**
- **Rarely, but we can try…**

Scalability = Efficiency and Concurrency. Capacity and performance are strongly related.
(ps: I have colleagues who "predict" with much more confidence!)

What not…. (old ratios from 1995)

- **BCHR - logical vs physical IO**
  - **<90% => need more cache**
- **Scan Ration - Index vs table-scans**
  - **> 1% => need more indexes**
- **Parse ratio – (dynamic) sql**
  - **>1% => increase shared_pool**
- **Sort Ratio (memory vs disk)**
  - **>10% => increase sort_area_size…**

- **Number, counters, mostly Since startup.**
- **Aggregates !**
- **No direct relation with one particular task/problem.**

Some of the old ones – Millsap, Lewis and others have explained at length why they don't work.
Note: a good BCHR on a test will not guarantee good, consistent behaviour in production.

AWR has some ratios ….

logica

THAT MATTER

	Per Second	
Redo size:	629,039.51	663.37
Logical reads:	120,99	,474.09
Block changes:		43.47

Top 5 Timed Events

Event	Waits	Ti	Call Time	Wait Class
CPU time			58.9	
db file sequential read			18.7	User I/O
log file sync		6	18.7	Commit
log file para		3	17.3	System I/O
db file		12	5.8	System I/O

Tra	82.08

And … Is your Customer Happy ?

Does it relate to relevant work ?

Commit Your ORACLE Knowledge

The numbers at the top will give some impression, but will generally not pinpoint the problems yet. Serious Performance nowadays is about measuring time and resources, the top-5 AWR and the MilliSeconds….!

Basics: what do we need to do….

logica

SOLUTIONS THAT MATTER

- Requirements (and not all system are “Xtreme”)
- Database (Tables & Indexes)
  - ACID
  - 12 + 1 rules of Codd and Date
- Application (someone will dream up code...)
- SQL: Store, Retrieve, Manipulate DATA ….
  - Fast enough to serve customer requirement.
  - Within reasonable hardware (and licen$$$)

My hobbyhorse: Fast - and Scaleable

logica
SOLUTIONS THAT MATTER

- **Individual actions; must be efficient**
  - **C reate / Insert (1x)**
  - **R ead / Queries (Nx, which fields, why?)**
  - **U pdate (Nx, which fields ?)**
  - **D elete (1x, bulk/del old data?)**
  - **…. Efficient ? … SQL and Indexes !**

- **Concurrent actions; must remain efficient**
  - **Limit locks (no blocking of others)**
  - **No unusable indexes (exhange part..!)**
  - **No hot-blocks (buffer busy waits).**

Start at detail: SQL – how much work…

logica
SOLUTIONS THAT MATTER

- What generates my (DB) workload ?
  - SQL, the statements that tell the DB what to do!
  - Do you recognize the qries in the awrrpt ?
  - (e.g. are you looking at the right report)

- From Statspack / AWR / V_$SQL / Traces
  - What Time (ela, cpu) does a stmnt take (aggregated!)
  - How much Work does a stmnt do (gets, rows-processed)
  - What job, what Unit of work was done ?
  - Is that Reasonable … ? (fast? scaleable?)

- (Expected) Frequency and “workload” for a qry ?

Most, if not all DB activity stems from SQL statements. So that is where I start.
And statspack, AWR, V_$SQL or 10046 are all excellent instruments.

Details: AWR, V$SQL or WRH$

logica

SOLUTIONS THAT MATTER

SQL ordered by Gets

- Resources reported for PL/SQL code includ
- Total Buffer Gets: 32,612,884
- Captured SQL account for 91.2% of Total

Buffer Gets	Executions	Gets per Exec	%
32,570,957	1,363	23,896.52	
1,851,387	1,765	1,048.94	
1,828,401	1,765	1,035.92	
1,764,845	1,765	999.91	
1,648,859	1,765	934.20	
1,352,149	1,765	766.09	
1,245,977	1,765	705.94	

SQL ordered by Executions

- Total Executions: 548,950
- Captured SQL account for 54.7% of Total

Executions	Rows Processed	Rows per Exec	Cl
77,294	77,294	1.00	
72,676	0	0.00	
60,498	60,498	1.00	
28,658	28,658	1.00	
10,330	10,330	1.00	
1,765	1,765	1.00	
1,765	1,765	1.00	
1,765	1,765	1.00	
1,765	1,765	1.00	
1,765	1,765	1.00	

```
SQL > Select buffer_gets, executions, rows_processed
From V$SQL where … order by … ;
```

Ratios per Statement (just mine…)

logica
SOLUTIONS THAT MATTER

- **Gets / Row** **<10**
  - **Realistically, any data is accessed via Index!**
- **Gets / Execute :** **<100**
  - **How much work (CPU, IO) will it take.**
  - **For a million rows, allow some more work…**
  - **But: be careful if executed at high-frequency.**
- **Gets / Transaction** **<1000**
  - **Why? Efficiency! And Limit the time of locking.**
  - **Problem: more difficult to measure in detail.**

Ratio to find Locking…

logica
SOLUTIONS THAT MATTER

- **Special case: TX-enqueue waits.**
  - **Table / Segment can be found from segment-stats**
  - **Which Stmnt ?… use a ratio!**

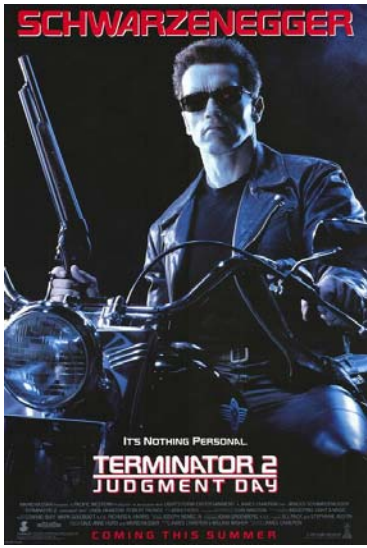
- **CPU-time / Elapsed time (percentage) >50 %**
  - **Notably: looking for Concurrency or IO problems**
  - **Also : if it waits for anything but CPU**
  - **(buff-busy, log-file-sync, log-file-par-wrt)**

- **To fix locks: often means talking to Arch and Dev…**
  - **Start transaction at latest possible moment.**
  - **Avoid running totals and similar constructs if possible.**

Fixes, if needed…

logica
SOLUTIONS THAT MATTER

- "Elimination": don't run the stmnt.
  - Best option!
- "Optimization": make it faster.
  - Realistic option (hopefully)
- "Containment" : run the stmnt less frequent.
  - (= Worst option; It Will Be Back!
- Do-Nothing:
  - If you are confident about workload and hardware….



If and How to fix will depend on your situation, but you basically have those options.
And nowadays, a lot of problems get "killed by Iron". Capacity is becoming cheaper all the time.

Before we Continue…

logica
SOLUTIONS THAT MATTER

- Warning : Ratios are Still BAD
- In All Cases, PLEASE understand the LIMITATIONS.
- Most ratios are Aggregates – not Individuals
- Skew
- What if 1 execute cause the average to go Off the Scale ?
  - "…where status=:1" using c_archived;

- Mitigation:
  - Know your system and your Data!
  - Small intervals (avg over small N).
  - Careful (re)testing.

Zoom out… : why all that SQL again ?

logica
SOLUTIONS THAT MATTER

- System processes “units” - find the unit.
  - Documents, Messages, Services, SOA…,
  - Orders, Clearings, Bookings
  - whatever (there will be an app for that…)

- Run “unit tests” of 1, 10, 100, … n “units”
- How much does 1 “unit” take ?
- How many units we need (per second? per hr? per day?)

- Anecdote:
  - Required: 1800 msg/sec, 9 very basic msgs, <2 sec/msg.
  - Tested: 17 msg/sec, 2 node RAC, 8 cores each – CPUs @ 85+%

Zoom out… : does my app scale ?

logica
SOLUTIONS THAT MATTER

- Dig in: how much “Work” is done per unit:
  - Executes / unit (what SQL, at what frequency)
  - Gets / unit (how much LIO, PIO)
  - Round-trips / unit (network and conn-pool sizing!)

- SNAP$, WRH$, 10046, even V_$SQLwill tell you
  - “snap” the stmnts on a unit-test or volume-test.
  - (and verify they are “from the unit”)
- Determine if workload will “realistically scale”
- Proposed Fix: don’t do 300 SQLs (round-trips!) per msg.

Zoom out… : my numbers (YMMV)

logica
SOLUTIONS THAT MATTER

- **Round-trips / unit (how to count ?) <1**
  - **RT: Expensive & risky, too many components**
- **Executes / unit <100**
  - **Minimize stmnts per "unit"**
- **Gets / unit <1000**
  - **LIO(and PIO) take up CPU- and Cache-capacity**

- **Yours will be Higher, but "knowing" will help you Think!**
- **(and on some system you can just add Iron…)**

Sumary, Advice.. (hardest slides) 1/3

logica
SOLUTIONS THAT MATTER

- Know the limitations of “ratios” !

- When diving into details:
  - Ratios ~ load per statement, per transacti

- Gets / Row <10
- Gets / Exe <100
- Gets / Transaction <1000

Sumary, Advice… 2/3

logica
SOLUTIONS THAT MATTER

- At higher level:
  - Ratio ~ Load per Unit (per msg, per screen, …)

- Round Trips / Unit <=1
- Executes / Unit <100
- Gets / Unit <1000

- (did I mention the Limitations…?)

If time, my wishlist: 1) Redo per stmnt, redo-per-unit, would be nice to get redo from statspack and 2) roundtrips. Some easy way to count round-trips (from statspack).

Sumary, Advice… 3/3

logica
SOLUTIONS THAT MATTER

- **Troubleshooting:**
  - **Find the heavy units (and fix)**
- **Testing:**
  - **Guesstimate workload**
  - **Decide if “accaptable”, or how to fix**
- **Planning:**
  - **Assess capacity (and keep some slack)**
- **Try it! - and Tell us about it.**

Commit Your ORACLE Knowledge

Remember the fixes: “Eliminate”, “Optimize”, “Contain” or “Ignore”
… or get more hardware (and did I mention the limitation of “ratios” ?)

Don’t Take my word for it…

logica
SOLUTIONS THAT MATTER

Tahiti.oracle.com: start with concept-guides

Technet (but be critical)

Oracle-L : real world stuff

www . Bloggingaboutoracle . org (company ramblings)

SimpleOracleDba . Blogspot . com (my ramblings)

Do some testing yourself …

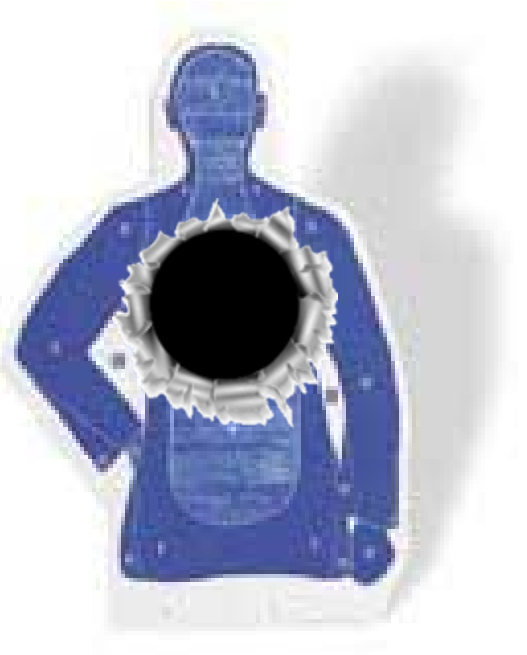
Goethe (simplicity, limitations…)

Commit Your ORACLE Knowledge

Quick Q & A (3 min ;-) 3 .. 2 .. 1 .. Zero

logica
SOLUTIONS THAT MATTER

- **Questions ?**
- **Reactions ?**
- **Experiences from the audience ?**



Commit Your ORACLE Knowledge

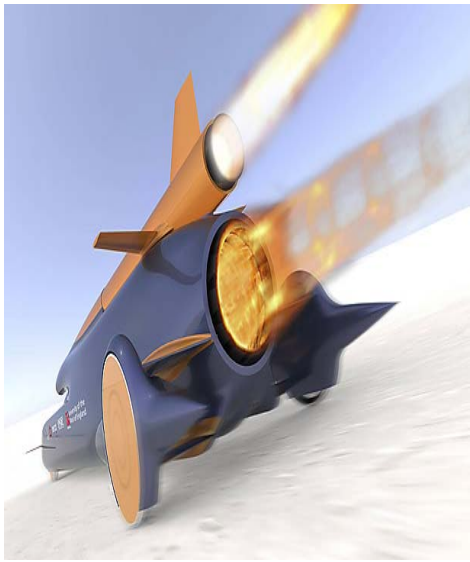
Here…

- **Placeholder slide to indicate ….**

Clusters often seem promising, but the “size” estimate requires predictable data.
Willing to learn on this one! Someone do a ppt on this next Conference ? (demo_sthc.sql if time)

But… As soon as the where-clause gets out of the “comfort zone”,
The STHC reverts to conventional behaviour, or worse.

Storage-wise, and STHC resembles a HEAP, but it is not supposed to “grow”
And why would I need the PK…



Inconclusive. Modern hardware has moved the bottleneck from CPU to Disk.
Check the system stats on old versus new systems: CPUs are sooo much faster, IO can hardly keep up.