4SYNERGY

Capacity Planning – without tooling

(Please keep IT simple)

Piet de Visser
The <u>Simple</u> Oracle DBA

After soo much discussions I wanted to just state a few simple rules of thumb.
\Too simple?? – yes.   But this is what I often do.

Agenda (45 minutes)   PdVBV

Capacity Planning   (= Everything)

Anecdotes   (I'm sure you have them too)
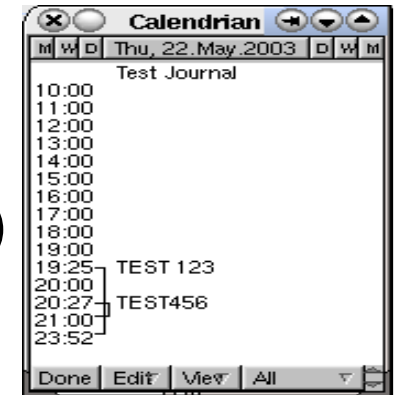
The quick fix-trick   (Q & D)

More elaborated tricks   (better …?)

Discussion   (Do Challenge!)

This will not solve your cap-plan-problem, but it may help you live with the shortcomings of the proces.
I want to pose a few stmnts, and then discuss with Peers.   Possibly Improve life at the **coalface** ?

- **Capacity = Performance (= hot, right?)**
  - **System Snappy; Users Happy … ?**
  - **Do batches run inside timeslot ?**
  - **Can we patch/upgrade inside 30min ?**

- **Capacity is Management-level guff. Budget..**
  - **Do we have enough kit ?**
  - **(SLAs, targets, requirements..)**
  - **Can we add more plants / trucks / customers ?**
  - **Will it still work in 6 – 12 – 18 – 24 months ?**

- **Capacity is rarely right - just live with that**
  - **%usr, %idle, %steal : at least one will be too high…**

(is it 1700 yet?) Capacity is Serious. Most problems have a capacity–angle. And if Capacity is never right, you want to be on the safe side: I prefer my systems Idle.

- **You want 17 or 1800 tx/sec ?**
  - **Java ???… (nuf said)**
- **Looking at 1TB (through bathroom window)**
  - **Processing 1M lines/hr…**
- **"our system is getting full"..**
  - **Heavy SQL in OLTP (we got lucky - fixable)**

- **KIWI – but not for (very)dynamic sql**
- **KIWI – but not for single-threaded processes.**
- **Check Jonathan's ppt about EXA, "The Answer"**
- **…There will always be "capacity" problems.**

If 1 tx = 300 sql-calls, if you have 32-bit machines, if you simply do too much single-calls, if you cannot use PQ, .. Or of your problem happens to just miss the exa-sweet spots… You have a Capacity problem

4 SYNERGY

)

Thank You !

')

Find out from the audience.
Include exa–plug...

Capacity : Demand and Supply (and you) PdVBV

- **Capacity : Demand - What do you need done**
  - **Get the Business numbers – Future Projections!**
  - **Nr Customers (and activity)**
  - **Nr orders / nr movements / amount of history…**

- **Capacity : supply - What kit do you have.**
  - **Hardware : CPU / Memory / Storage (Exa??)**
  - **Software: Which system/software will be used ?**

- **Capacity: can Supply match Demand ?**
  - **or your job is in danger,**
  - **Or… you can make a fortune touting Exa.. (or cloud)**

Demand: find what the system or business needs to to do (in the near future… 6 – 24 Months)
Supply: find how you can (or are forced to) provide for that. Then macth (simple eh?)

Capacity : Demand… Measure + Verify

PdVBV

- **Business says: 1 order per min … average…**
  - – **Verify in the system.**

- **Measurement + Testing is key ...**
  - – **(I know: sometimes, the measurement just isnt..)**
  - – **(I know: sometimes you cannot test)**

- **Short: Measure!**
  - – **How much work in 1 unit (1 day, 1 order, 1 report).**
  - – **Use _only_ available tools (don't fall for fancy stuff)**
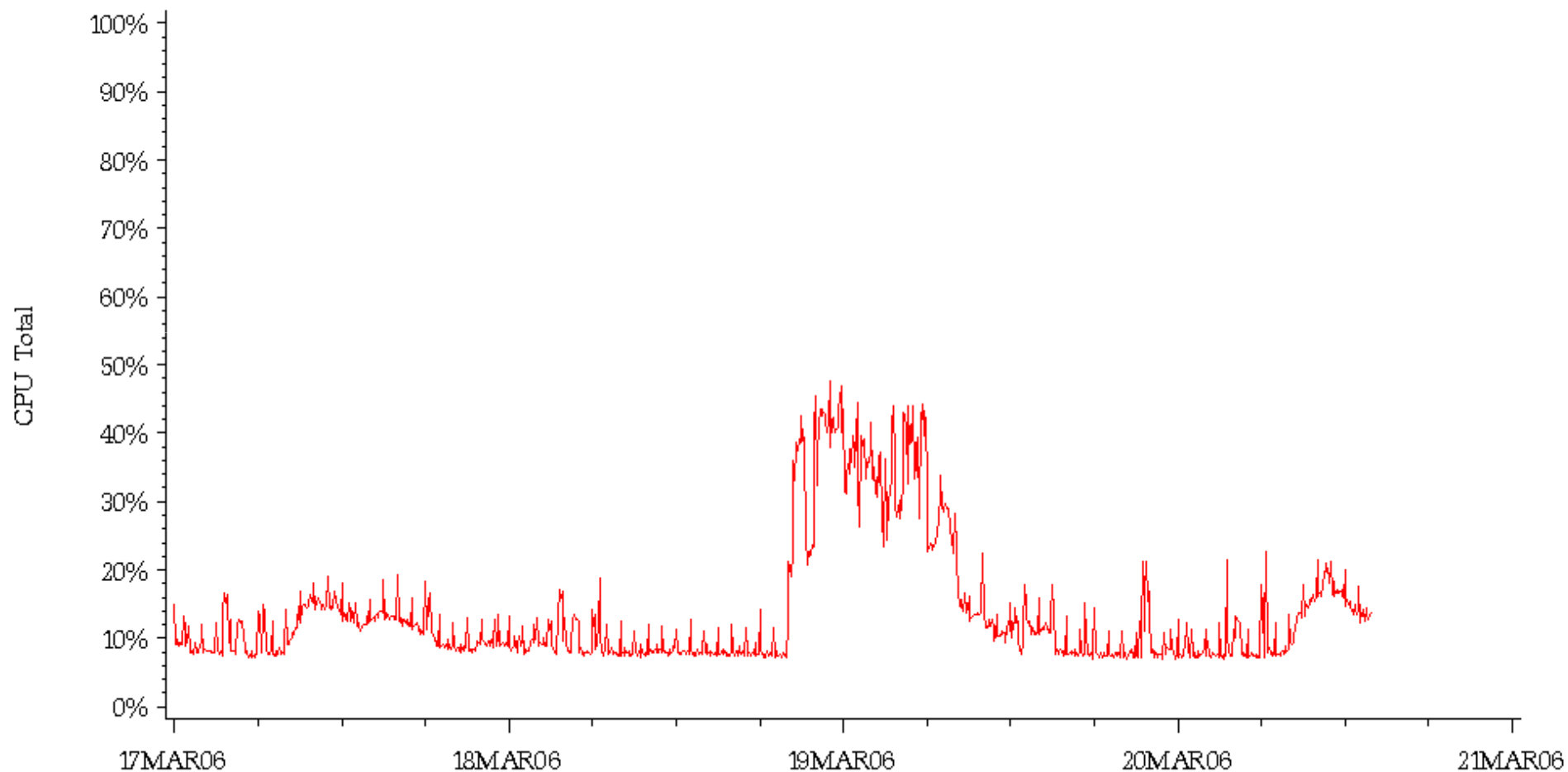 - – **SQL / Statspack / AWR**
  - – **vmstat / sar : how much is used…**

Demand: Measure and Verify. And if you must use expensive tools (RAT, load–Gunner..) : the more expensive + fancy the tool is : the less it will help you. (but it may look good)

Business As Usual… Typical CPU usage of database with some anomalies.

BAU - Batch
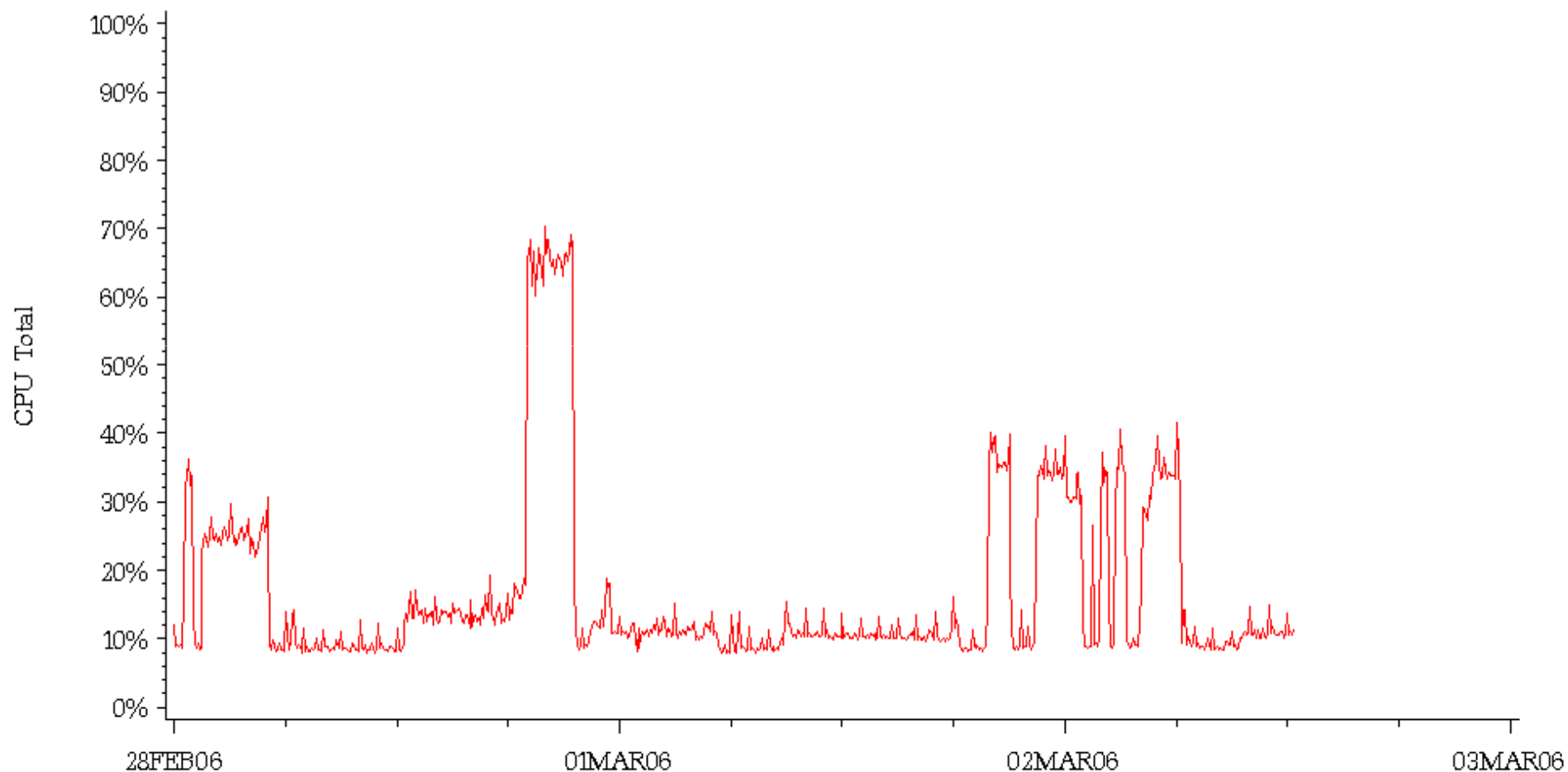
PdVBV

Typical batch – spot the impact of the eways and the index.

Capacity : Demand; Measure + Verify ….   PdVBV

- ## Measure, Trust + verify
  - Go verify with SQL*plus, sar and .. xls

- ## Activity per Day / Week / Month
  - 100K movements = 1 per second… (average!)

- ## MB growth per day / week / month
  - 20M /day in SJO, or 1G/day in SGP..

- ## System usage (SAR or OEM)
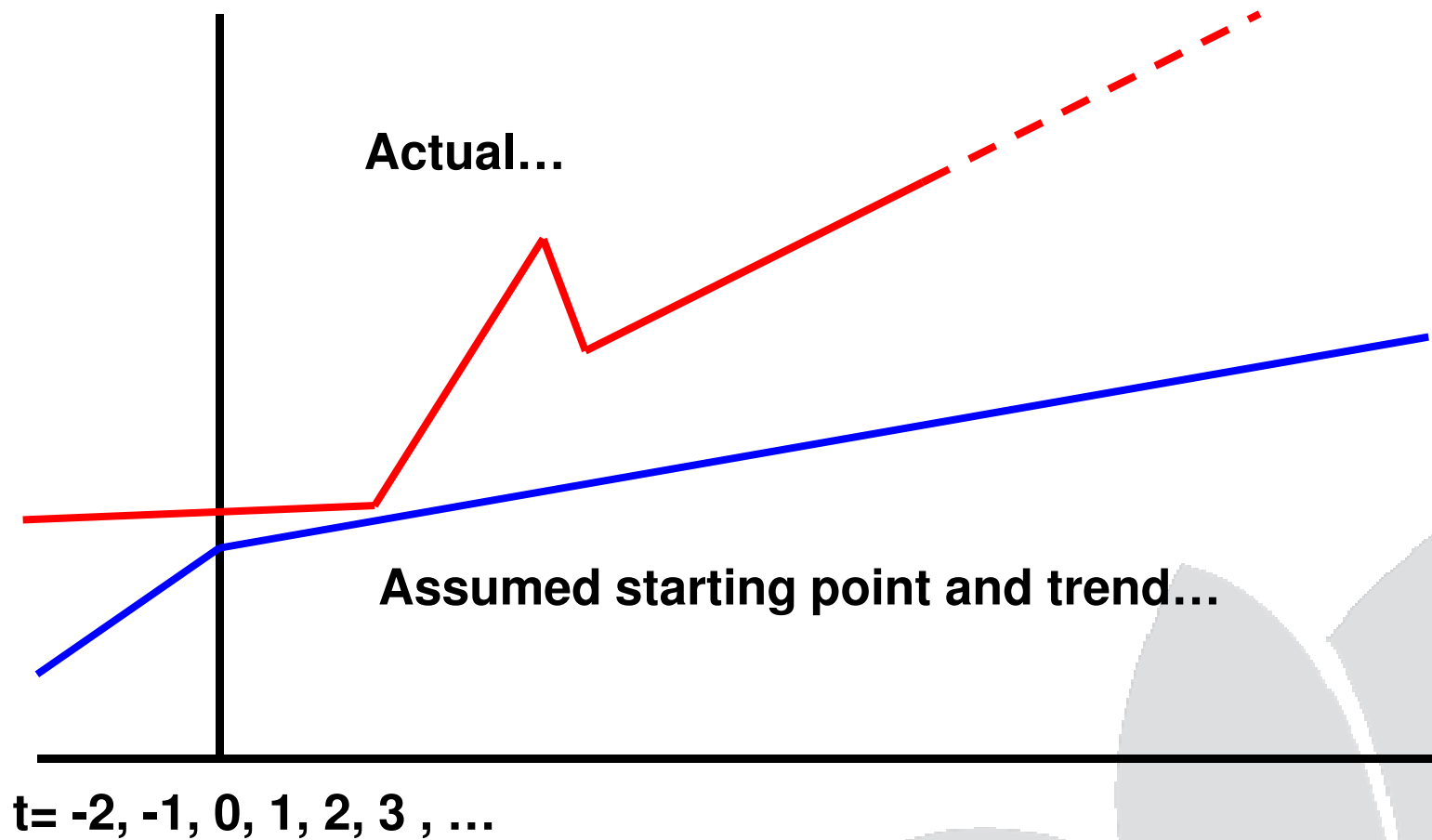 - And XLS (make some simple plots)
  - Beware of %Steal

First thing I do : run sar, save some AWRs, save + plot some NUMBERS.
Repeat Daily. After 1 week, I get a feeling for system…

4 SYNERGY

Actual…

Assumed starting point and trend…

t= -2, -1, 0, 1, 2, 3 , …

Just let this speak for itself….
(are YOU measuring your usage?)

The Ceiling – Iron or Brains ?

PdVBV

t= -2, -1, 0, 1, 2, 3 , …

Just let this speak for itself….
(if You hit the ceiling: can you plot graph and indicate problem?)
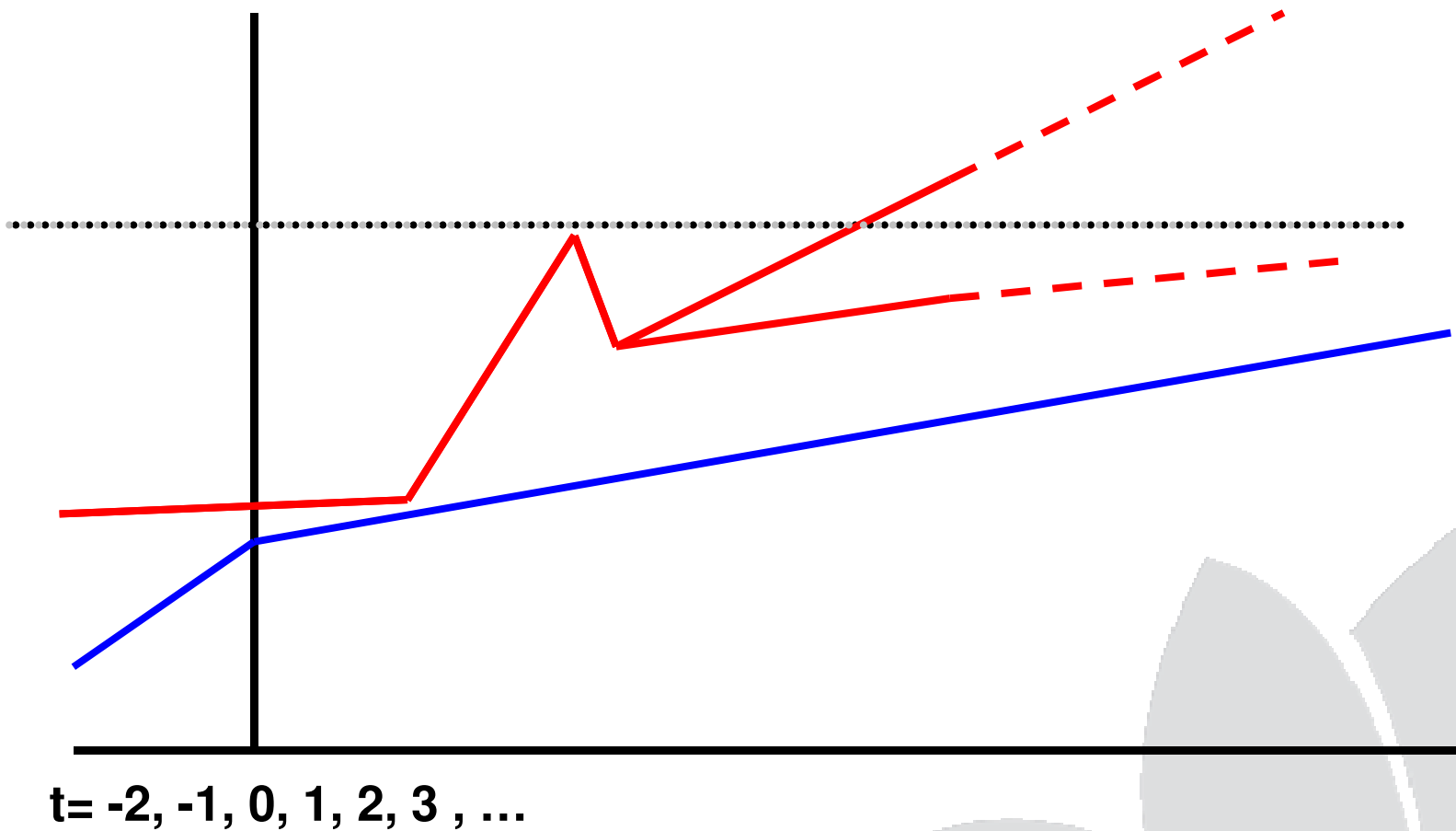
- **How big is your system**
  - **(very often, nowadays just big…)**
  - **Cat /proc/cpuinfo, meminfo**
  - **Statspack (will tell you !)**
  - **Surprise: you use 2G out of a 24G server**
  - **Surprise: your db does 10min of work in 60min…**

- **Pitfalls:**
  - **Virtual**
  - **Slowness in storage**
  - **Peak-loads (10am and 10pm)**

Many system over–supplied. But some are over–taxed… Go verify yours…

AWR has most of the data you need….

4SYNERGY

	Per Second	
Redo size:	629,039.51	663.37
Logical reads:	120,98...	1,474.09
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 paral...	3			17.3	System I/O
db fil...	12			5.8	System I/O

| Tra... | | 82.08 | |

And … it tells you CPU_COUNT, system-memory and usage…

AWR is a very good starting point for "supply" data, but not the total picture yet…

4 SYNERGY

- **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 <u>Time</u> (ela, cpu) does a stmnt take (aggregated!)**
  - **How much <u>Work</u> does a stmnt do (gets, rows-processed)**
  - **What job, what <u>Unit</u> 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.

- **After Quick + Dirty measurements (guesstimates)**
  - **You know "demand"**
  - **You know "supply"**

- **Does it match ?**
  - **Only you can answer (and the users!)**

- **If not…**
  - **Performance-tweaking.. (op-tuh-my-sation)**
  - **KIWI…**
  - **Mostly a combination.**
  - **(example: eliminating the report-DB Totally fixed it all…)**

IF not match, short–term fix will be "tuning", long term fixe may by hardware, or … optimization may just carry you over for longer period..

4SYNERGY

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 … ;
```

AWR and V$ tables offer lots of data!
Notice the nr of 1765 executes of a qry. How many rows? How many buffers ?

My hobbyhorse: Fast - and Scaleable

Individual actions; must be efficient
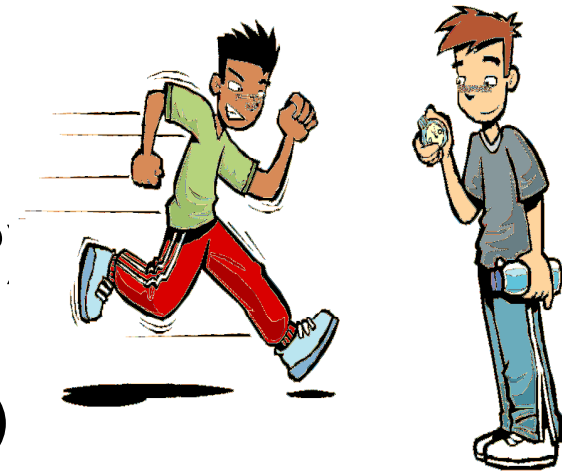
    C reate / Insert (1x)

    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).

Measure individual activity (of Critical components) – can be a bit of work, but once you "get it", its not hard)
Then think & measure your concurrency issues : does it remain efficient, doest it scale…   © JL

- **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.**

Focus on Effort-per-action, and let each action be efficient.
The numbers are.. Just easy to remember. Some system function happily at 10x these numbers.

- **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 <u>Concurrency</u> 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.**
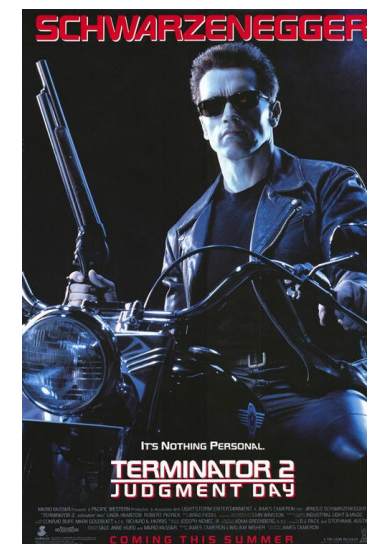 - **Avoid running totals and similar constructs if possible.**

Locking problems: nobody home… to find the exact stmnt: check for its CPU to Elapsed ratio.

Fixes, if needed…   PdVBV

- **"Elimination": don't run the component.**
  - **Best option!**

- **"Optimization": make it faster.**
  - **Realistic option (hopefully)**

- **"Containment" : run the item less frequent.**
  - **(= Worst option; It Will Be Back!**

- **Do-Nothing (KIWI) :**
  - **IF… you are confident about workload and hardware.**
  - **Dynanic-SQL…? Hmm; Single Threaded work…? Never!**

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.

- **Capacity : Demand - What do you need done**
  - **Get simple numbers**
  - **Project into future**

- **Capacity : supply - What kit do you have.**
  - **Find Simple numbers**
  - **Verify!**

- **Capacity: Match ???**
  - **Tip: Plan simply, and carry some big kit..**

You know where to get demand (and verify), you know how to find supply… Match,
And plan largely. Any system with less then 50% idle is in danger… (IMHO)

Don't Take my word for it…   PdVBV

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 Shows the Master

Majority of times, I have been WRONG.
So go see for yourself – but don't complicate life.   "In der Beschrankung zeight sich der Meister"

Quick Q & A   3 .. 2 .. 1 .. Zero   PdVBV

4SYNERGY

- **Questions ?**

- **Reactions ?**

- **Experiences from the audience ?**

Question and Answer time. Discussion welcome   (what about that Razor?)
Teach me something: Tell me where you do NOT AGREE.

- **Placeholder slide to indicate**

- **END**

- **of this ppt,**

- **Remainder is just notes** in stuf ….

You are Here…

- You do APEX !

- You have skills..