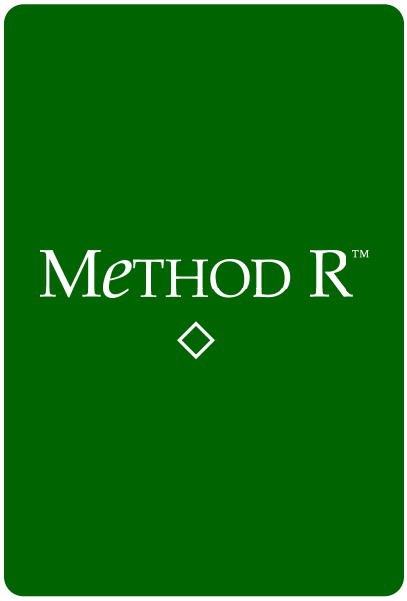


Millsap's Grand Unified Theory of "Tuning"

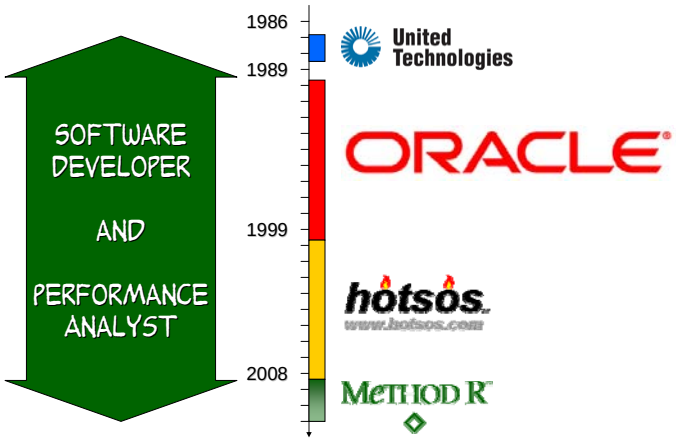
Cary Millsap
cary.millsap@method-r.com

Northern California Oracle Users Group
San Ramon, California
9:30a–10:30a Thursday 21 August 2008

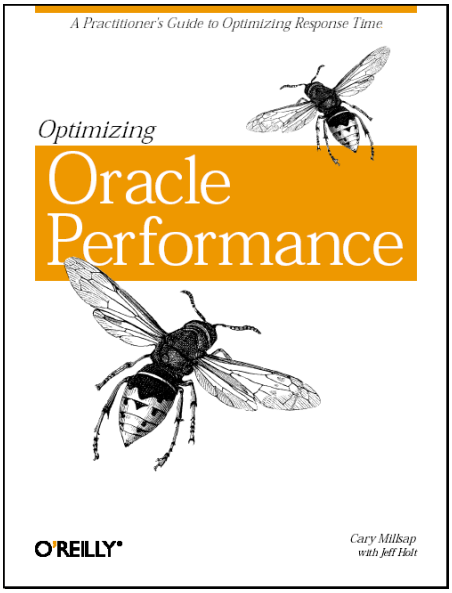
http://method-r.com

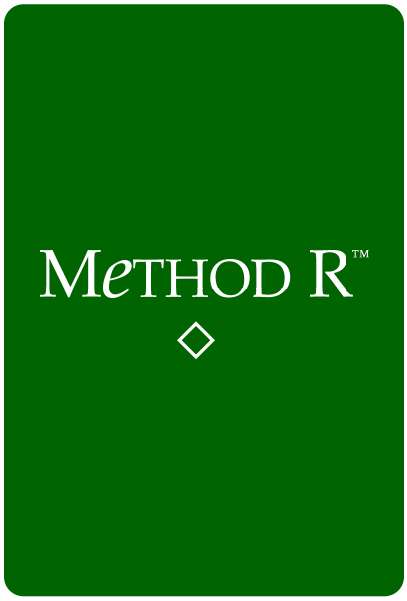
http://carymillsap.blogspot.com

cary.millsap@method-r.com



METHOD R™


A Practitioner's Guide to Optimizing Response Time
Optimizing
Oracle Performance
O'REILLY
Cary Millsap
with Jeff Holt

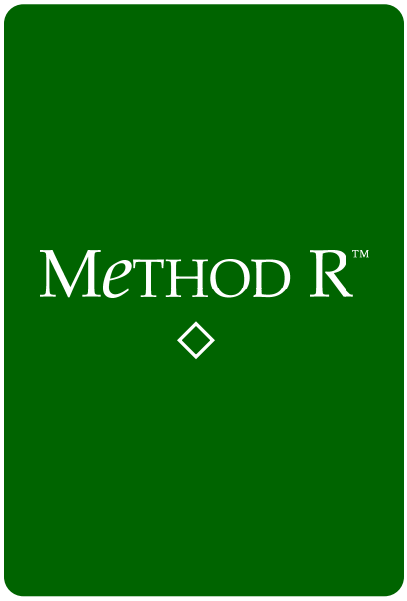


Millsap's Grand Unified Theory of "Tuning"

Cary Millsap
cary.millsap@method-r.com

Northern California Oracle Users Group
San Ramon, California
9:30a–10:30a Thursday 21 August 2008

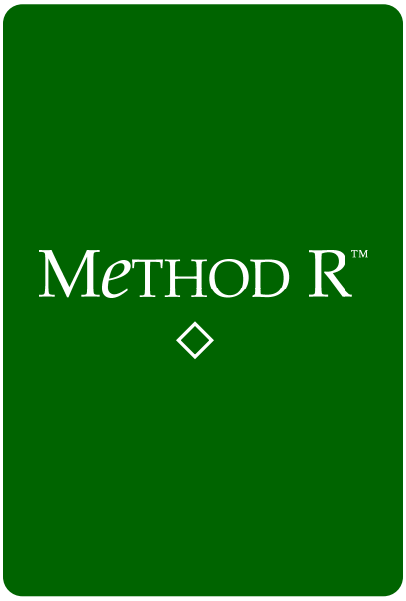
SO WHAT'S WITH THE QUOTES?



Millsap's Grand Unified Theory of "Tuning"

Cary Millsap
cary.millsap@method-r.com

Northern California Oracle Users Group
San Ramon, California
9:30a–10:30a Thursday 21 August 2008



Part 1: G.U.T.

THERE ARE TWO TYPES OF PERFORMANCE PROBLEMS IN THIS WORLD...

1

Response time problems.

2

Inefficiencies that aren't response time problems.

Yet.

THEREFORE…

G.U.T. proposition I

You must be able to attack response time problems for specific tasks that the business cares about.

EXAMPLE...

Posting hurts.

Fix it.

WHY SHOULD YOU CARE?

G.U.T. proposition 2

You need to be able to attack inefficiencies that aren't yet noticeable as user response time problems.

EXAMPLE...

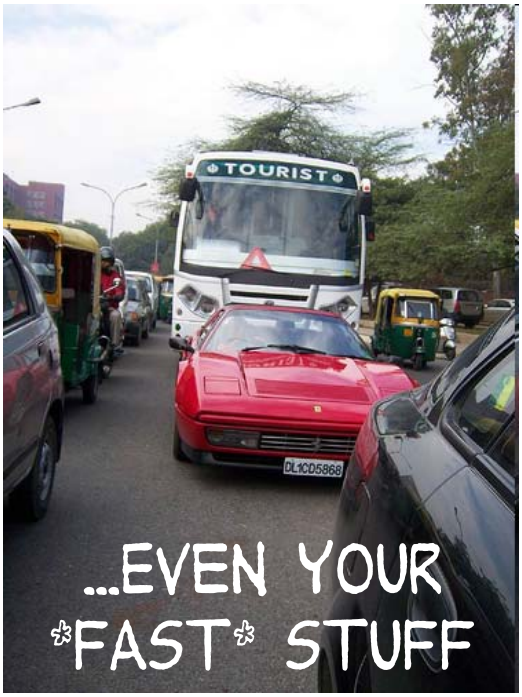
Posting takes 3 hours.
It should take 2,
but no user really cares.

WHY SHOULD YOU CARE?

BECAUSE *WASTE*
COSTS YOU MONEY

AND WASTE...

...MAKES
OTHER WORK
GO SLOWER

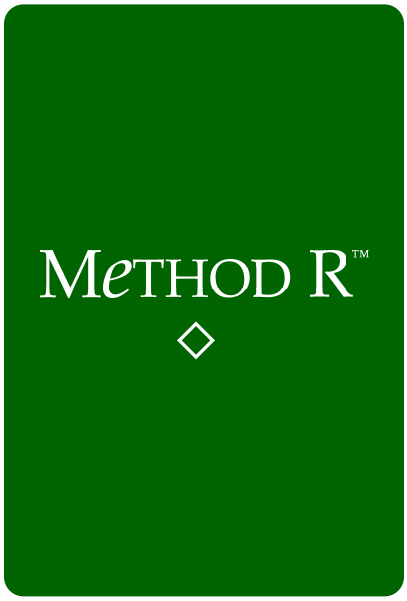


RECAP...

G.U.T. propositions

You **must** be able to attack

1. Response time problems
2. Efficiency problems



Part 2: Historical perspective

In the beginning...

1989

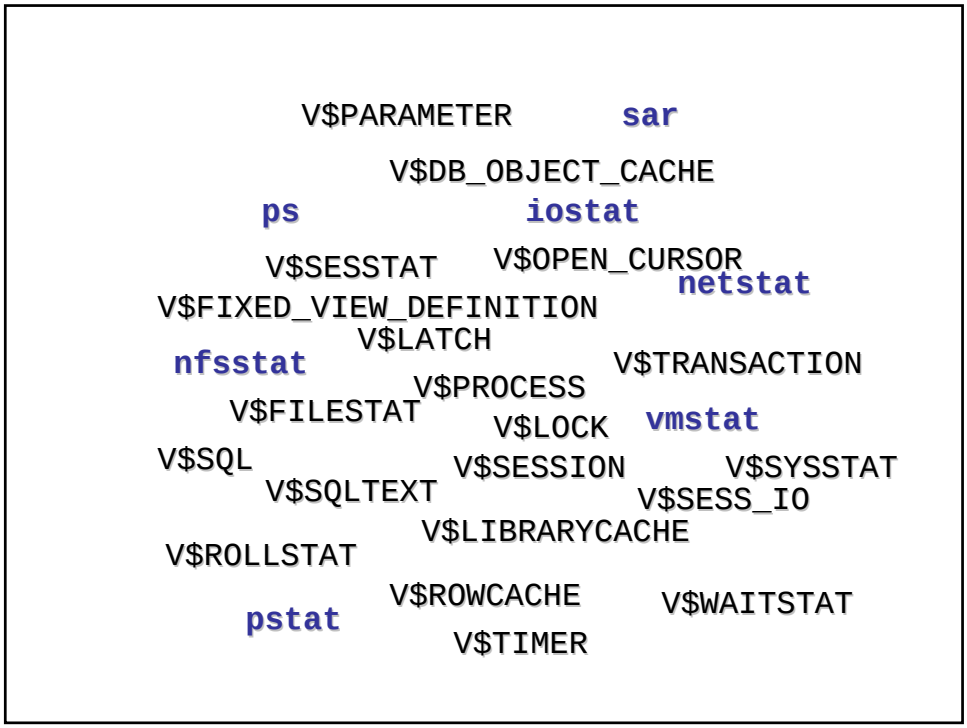
6.0.26

"TUNING" WAS...

bstat + estat

=

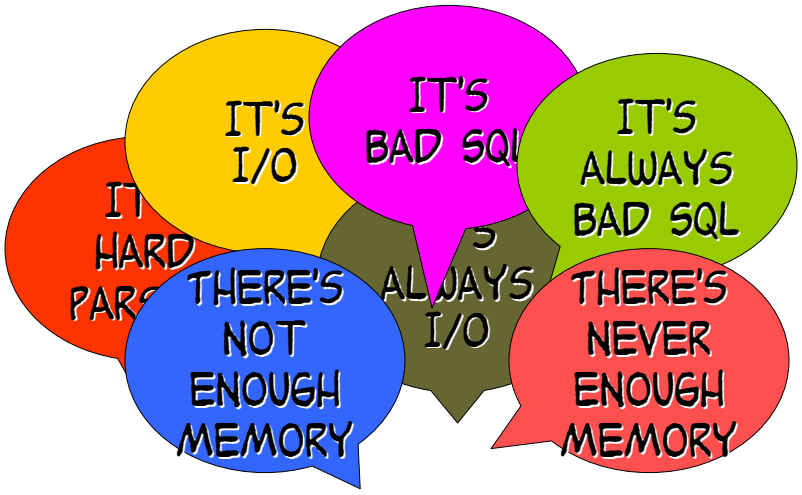
report.txt



PEOPLE LOOKED FOR "BAD NUMBERS"

INEFFICIENCIES

BUT HOW CAN YOU KNOW WHAT CAUSES A SPECIFIC TASK TO BE SLOW?


IT'S I/O
IT'S BAD SQL
IT'S ALWAYS BAD SQL
IT HARD PARS
S ALWAYS I/O
THERE'S NOT ENOUGH MEMORY
THERE'S NEVER ENOUGH MEMORY

MY PROBLEM...

How can you possibly

KNOW

that?

REMINDED ME OF...

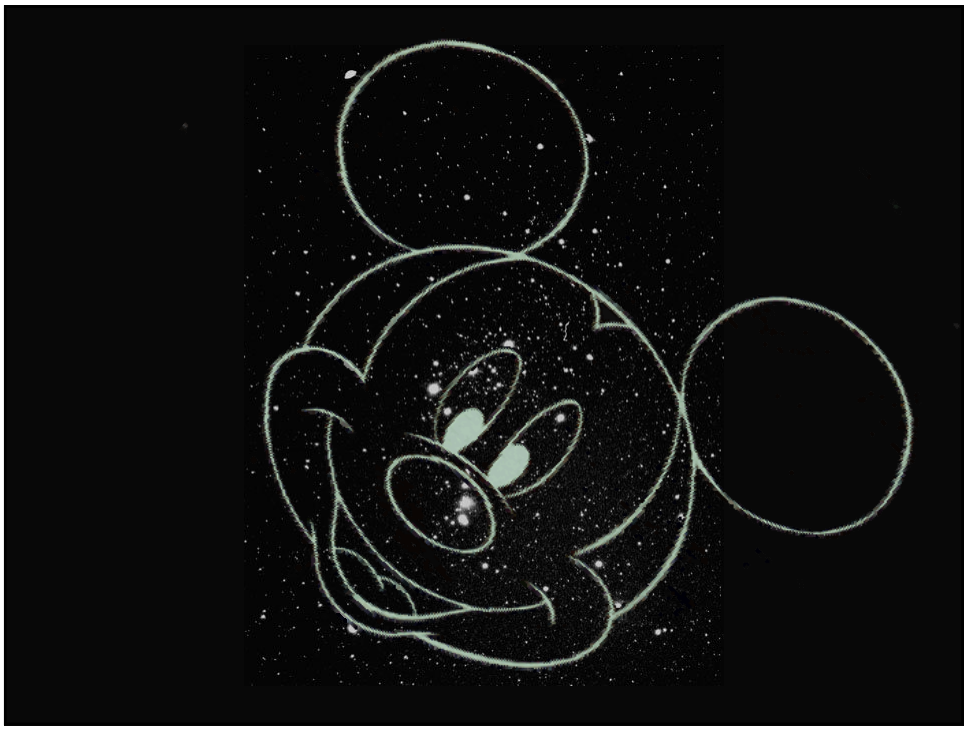
YOU *DO* SEE IT...

RIGHT?

BUT WHO SAYS

THAT

IS WHAT YOU
HAVE TO SEE?



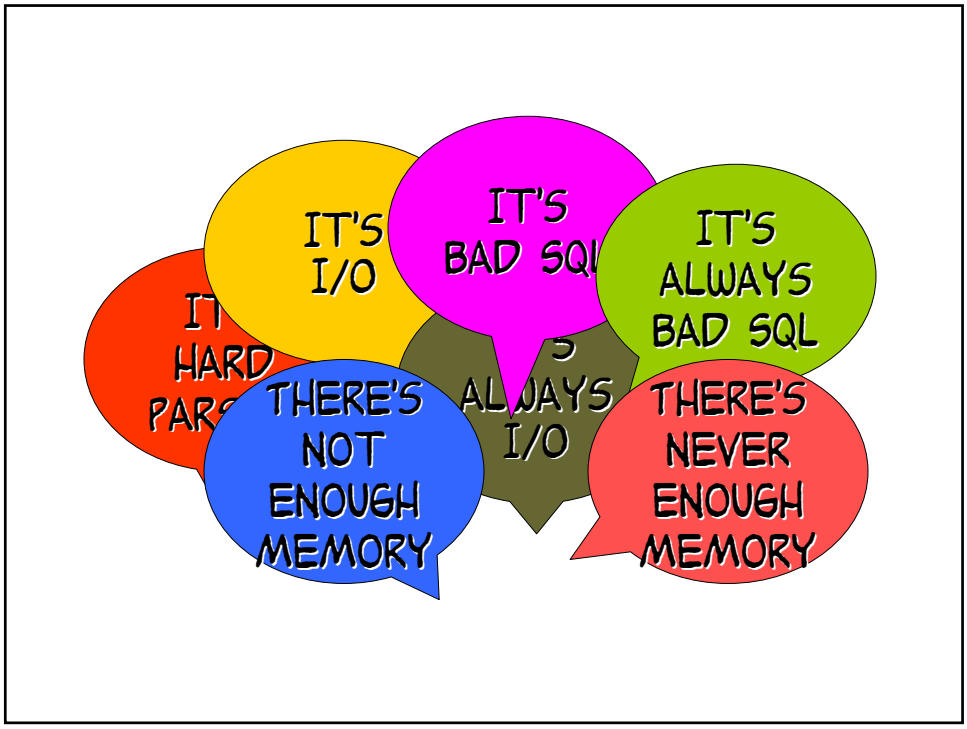
WHY NOT?

THE PROBLEM WITH
bstat/estat, ...

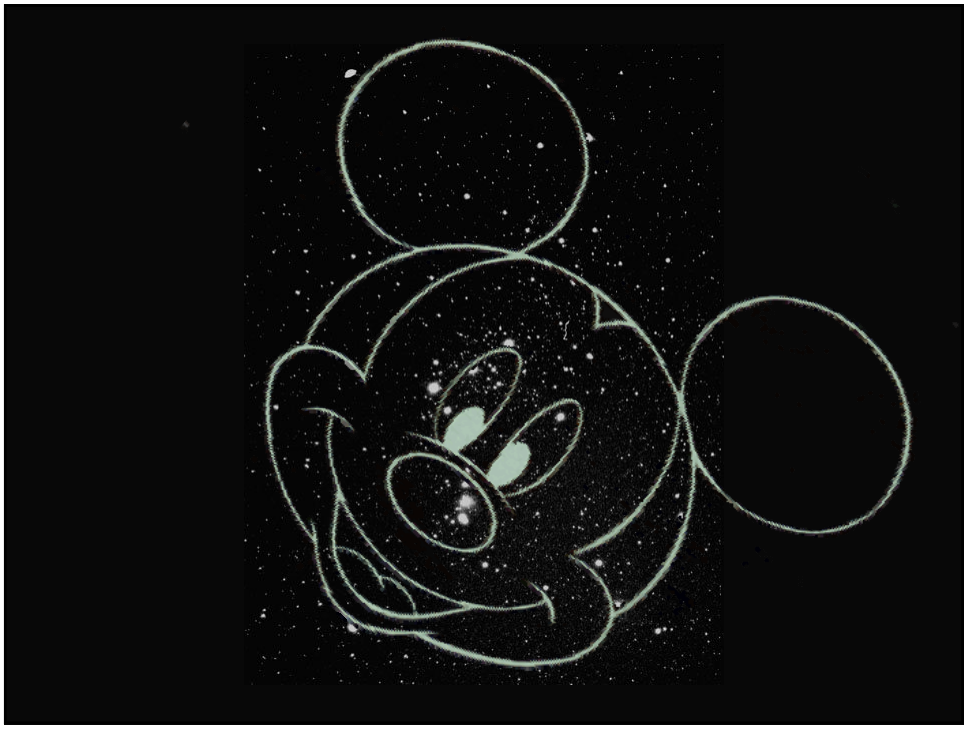
AND
utlbstat/utlestat, ...

AND STATSPACK, ...

AND EVEN AWR, ...


IT'S I/O
IT'S BAD SQL
IT'S ALWAYS BAD SQL
IT HARD PARS
ALWAYS I/O
THERE'S NOT ENOUGH MEMORY
THERE'S NEVER ENOUGH MEMORY


North
West
vailroger.googlepages.com/orionconstellation



SOMETIMES YOU
CAN'T KNOW

EXAMPLE...

WHAT IS YOUR TASK'S PROBLEM?

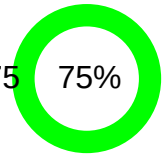
Resource consumption (seconds) by task

	Total system	
Latches	7,502	74%
I/O	275	3%
Other	2,323	23%
Total	**10,100**	**100%**

ACTUALLY NOT

Resource consumption (seconds) by task

	Total system		Your task	
Latches	7,502	74%	2	2%
I/O	275	3%	75	75%
Other	2,323	23%	23	23%
Total	**10,100**	**100%**	**100**	**100%**



HERE'S HOW IT HAPPENS

Resource consumption (seconds) by task

	Total system		Your task		Other tasks	
Latches	7,502	74%	2	2%	7,500	75%
I/O	275	3%	75	75%	200	2%
Other	2,323	23%	23	23%	2,300	23%
Total	**10,100**	**100%**	**100**	**100%**	**10,000**	**100%**

SOMETIMES YOU
CAN'T KNOW

...UNTIL YOU
LOOK AT THE RIGHT
IT

Resource consumption (seconds) by task

	Total system		Your task		Other tasks	
Latches	7,502	74%	2	2%	7,500	75%
I/O	275	3%	75	75%	200	2%
Other	2,323	23%	23	23%	2,300	23%
Total	**10,100**	**100%**	**100**	**100%**	**10,000**	**100%**

WHEN YOU DON'T...

An epiphany...

1991

(Oracle release doesn't matter)

"A higher database buffer cache hit ratio is a bad thing"

—Willis Ranney

HUH?

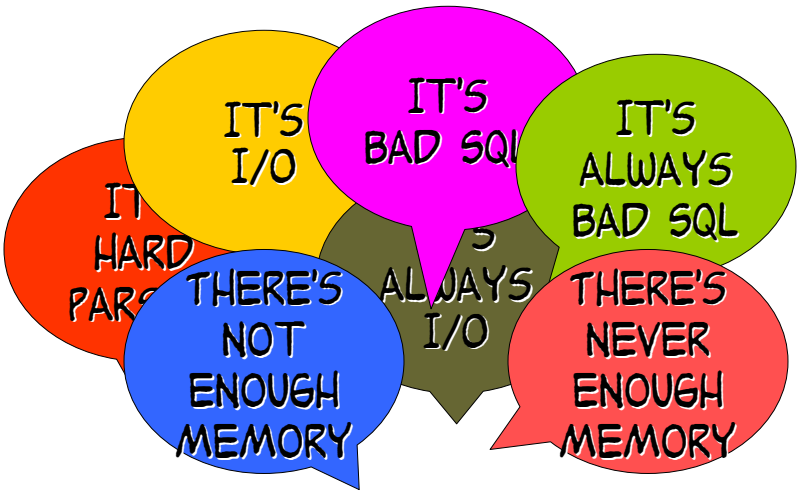
IMAGINE...
Q1 AND Q2 RETURN
IDENTICAL RESULTS

WHICH IS BETTER?

Query	BCHR
Q1	99%
Q2	90%

Query	BCHR	Memory accesses	Disk accesses
Q1	99%	100	1
Q2	90%	10	1

YOU CAN'T COUNT
ON THE RATIOS!



THE RULES IN A LOT OF PEOPLE'S HEADS...

WERE WRONG.

ANOTHER MAN WHO
KNEW THIS...

"To 'tune,' watch the user.
Not the DBA."

—Dave Ensor

BUT HOW?

tkprof

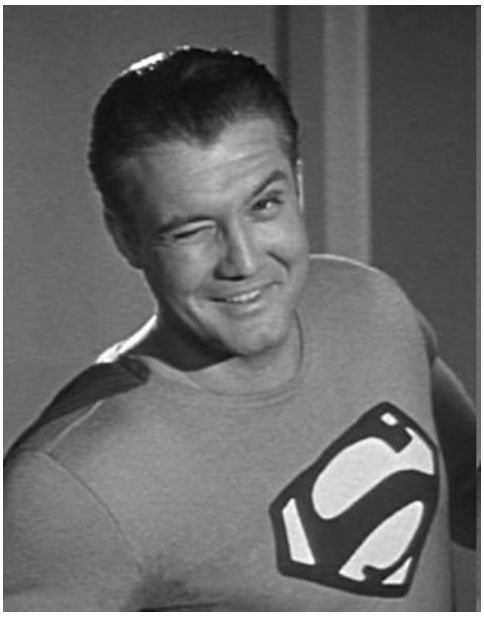
AN ORACLE DEVELOPER TOOL SINCE V6

```
select count(*)
from
 dba_objects


call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.07       0.15          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch        2      0.21       0.26          4       6695          0           1
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        4      0.29       0.41          4       6695          0           1

Misses in library cache during parse: 1
Optimizer mode: ALL_ROWS
Parsing user id: 5

Rows     Row Source Operation
-------  ---------------------------------------------------
      1  SORT AGGREGATE (cr=6695 pr=4 pw=0 time=264659 us)
  52319   VIEW  DBA_OBJECTS (cr=6695 pr=4 pw=0 time=1256721 us)
  52319    UNION-ALL  (cr=6695 pr=4 pw=0 time=942801 us)
  52319     FILTER  (cr=6694 pr=3 pw=0 time=367280 us)
  53574      HASH JOIN  (cr=655 pr=0 pw=0 time=804635 us)
     67       TABLE ACCESS FULL USER$ (cr=6 pr=0 pw=0 time=242 us)
  53574       TABLE ACCESS FULL OBJ$ (cr=649 pr=0 pw=0 time=216062 us)
   2379      TABLE ACCESS BY INDEX ROWID IND$ (cr=6039 pr=3 pw=0 time=145102 us)
   3013       INDEX UNIQUE SCAN I_IND1 (cr=3015 pr=3 pw=0 time=88196 us)(object id 39)
      0     NESTED LOOPS  (cr=1 pr=1 pw=0 time=2590 us)
      0      INDEX FULL SCAN I_LINK1 (cr=1 pr=1 pw=0 time=2578 us)(object id 107)
      0      TABLE ACCESS CLUSTER USER$ (cr=0 pr=0 pw=0 time=0 us)
      0       INDEX UNIQUE SCAN I_USER# (cr=0 pr=0 pw=0 time=0 us)(object id 11)
```

IT'S WHAT YOU NEED...

IF
YOUR PROBLEM
IS SQL

AND OF COURSE,
"THE PROBLEM IS
ALWAYS SQL"



Next...

Inspiration

1995

Oracle7

"I can reduce response time by X if you let me do Y"

—Virag Saksena

MY GOAL...

PUT THIS IN A BOTTLE.

WHICH BRINGS ME BACK TO THOSE "QUOTES"

METHOD R™

◇

Millsap's Grand Unified Theory of "Tuning"

Cary Millsap
cary.millsap@method-r.com

Northern California Oracle Users Group
San Ramon, California
9:30a–10:30a Thursday 21 August 2008

THE PROBLEM...

"TUNING" ISN'T GOOD ENOUGH

TUNING	WHAT WE WANT
MAKE SOME COMPONENT FASTER	MAXIMIZE ECONOMIC VALUE

TUNING	WHAT WE WANT
MAKE SOME COMPONENT FASTER	MAXIMIZE ECONOMIC VALUE
TRIAL AND ERROR	INFORMED ACTION

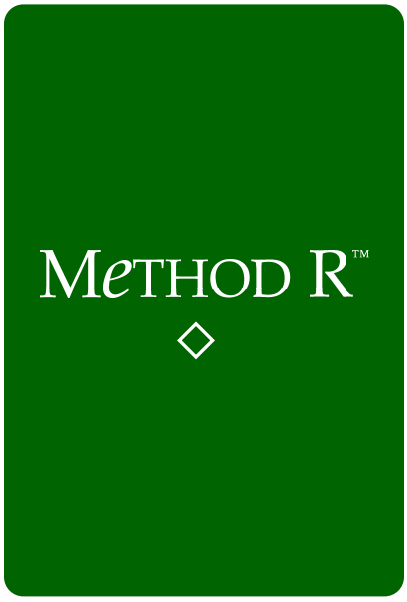
TUNING	WHAT WE WANT
MAKE SOME COMPONENT FASTER	MAXIMIZE ECONOMIC VALUE
TRIAL AND ERROR	INFORMED ACTION
WEAK ACCOUNTABILITY	ACCOUNTABILITY TO FORECAST

TUNING	WHAT WE WANT
MAKE SOME COMPONENT FASTER	MAXIMIZE ECONOMIC VALUE
TRIAL AND ERROR	INFORMED ACTION
WEAK ACCOUNTABILITY	ACCOUNTABILITY TO FORECAST
SUCCESS MEASURED BY EFFORT	SUCCESS MEASURED BY RESULTS

TUNING	?
MAKE SOME COMPONENT FASTER	MAXIMIZE ECONOMIC VALUE
TRIAL AND ERROR	INFORMED ACTION
WEAK ACCOUNTABILITY	ACCOUNTABILITY TO FORECAST
SUCCESS MEASURED BY EFFORT	SUCCESS MEASURED BY RESULTS

TUNING	*** OPTIMIZATION ***
MAKE SOME COMPONENT FASTER	MAXIMIZE ECONOMIC VALUE
TRIAL AND ERROR	INFORMED ACTION
WEAK ACCOUNTABILITY	ACCOUNTABILITY TO FORECAST
SUCCESS MEASURED BY EFFORT	SUCCESS MEASURED BY RESULTS

TUNING	OPTIMIZATION
MAKE SOME COMPONENT FASTER	MAXIMIZE ECONOMIC VALUE
TRIAL AND ERROR	INFORMED ACTION
WEAK ACCOUNTABILITY	ACCOUNTABILITY TO FORECAST
SUCCESS MEASURED BY EFFORT	SUCCESS MEASURED BY RESULTS



Part 3: How

The goal...

Optimization! ...in a bottle

2000

8i, 9i

WANTED...

REPEATABLE

TEACHABLE

method

=

deterministic
sequence of steps

THERE *WERE*
METHODS...

ORACLE PERFORMANCE IMPROVEMENT METHOD

1. Get candid feedback from users
2. Get full set of OS, db, app statistics—good and bad
3. Sanity check the OS
4. Review Top Ten list*
5. Build conceptual model
6. Propose remedies
7. Validate changes
8. Repeat until goals are met or become impossible

http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/technique.htm

ORACLE PERFORMANCE IMPROVEMENT METHOD

1. Get candid feedback from users
2. Get full set of OS, db, app statistics—good and bad
3. Sanity check the OS
4. **Review Top Ten list***
5. Build conceptual model
6. Propose remedies
7. Validate changes
8. Repeat until goals are met or become impossible

http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/technique.htm

*THE OPIM TOP TEN LIST

1. Bad connection management
2. Bad use of cursors and the shared pool
3. Bad SQL
4. Use of nonstandard initialization parameters
5. Getting database I/O wrong
6. Redo log setup problems
7. Serialization of data blocks
8. Long full table scans
9. High amounts of recursive SQL
10. Deployment and migration errors

http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/technique.htm

BUT THERE WERE TWO BIG PROBLEMS...

*THE OPIM TOP TEN LIST

1. Bad connection management
2. Bad use of cursors and the shared pool
3. Bad SQL
4. Use of nonstandard initialization parameters
5. Getting database I/O wrong
6. Redo log setup problems
7. Serialization of data blocks
8. Long full table scans
9. High amounts of recursive SQL
10. Deployment and migration errors

WHAT IF YOUR PROBLEM IS NUMBER NINE?

http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/technique.htm

*THE OPIM TOP TEN LIST

1. Bad connection management
2. Bad use of cursors and the shared pool
3. Bad SQL
4. Use of nonstandard initialization parameters
5. Getting database I/O wrong
6. Redo log setup problems
7. Serialization of data blocks
8. Long full table scans
9. High amounts of recursive SQL
10. Deployment and migration errors

DO YOU REALLY WANT TO SPEND TIME CHECKING THE FIRST EIGHT?

http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/technique.htm

*THE OPIM TOP TEN LIST

1. Bad connection management
2. Bad use of cursors and the shared pool
3. Bad SQL
4. Use of nonstandard initialization parameters
5. Getting database I/O wrong
6. Redo log setup problems
7. Serialization of data blocks
8. Long full table scans
9. High amounts of recursive SQL
10. Deployment and migration errors

...AND WHAT IF YOUR PROBLEM IS NUMBER ELEVEN?

http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/technique.htm

*THE OPIM TOP TEN LIST

1. Bad connection management
2. Bad use of cursors and the shared pool
3. Bad SQL
4. Use of nonstandard initialization parameters
5. Getting database I/O wrong
6. Redo log setup problems
7. Serialization of data blocks
8. Long full table scans
9. High amounts of recursive SQL
10. Deployment and migration errors

http://download.oracle.com/docs/cd/B28359_01/server.111/b28274/technique.htm

IT'S LIKE A
FULL TABLE SCAN

...OF A TABLE THAT'S
MISSING SOME ROWS

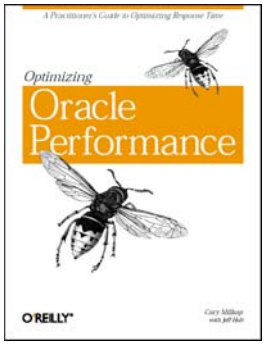
WHAT WE NEED...

AN *INDEX* INTO THE PROBLEM YOU'RE HAVING RIGHT NOW...

...WHOSE *KEY* IS THE NAME OF YOUR BUSINESS TASK

+

=



Method R

1. Target the right task
2. Collect its R details
3. Forecast, act
4. Repeat until optimized

Resource consumption (seconds) by task

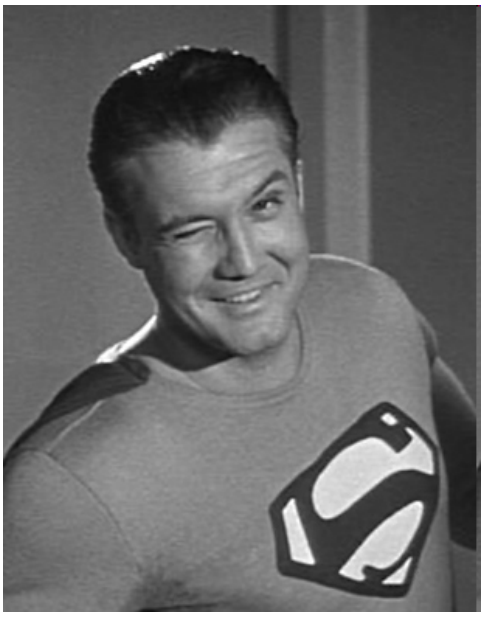
	Total system		Your task		Other tasks	
Latches	7,502	74%	2	2%		
I/O	275	3%	75	75%	200	2%
Other	2,323	23%	23	23%	2,300	23%
Total	**10,100**	**100%**	**100**	**100%**	**10,000**	**100%**

ORACLE EXTENDED SQL
TRACE DATA

"CASE STUDIES" SESSION
FOR MORE INFO

"By the way... The problem is not always your SQL."

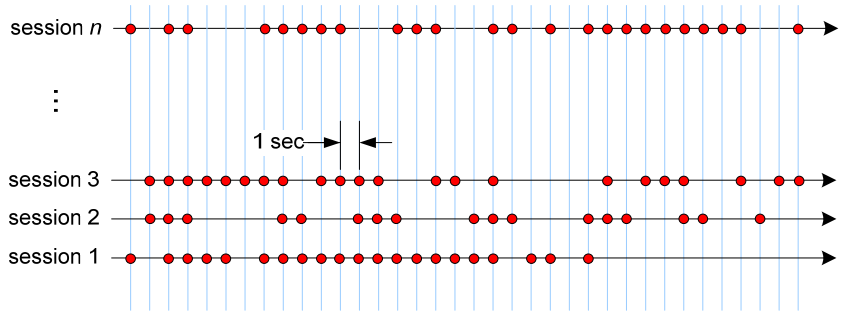
—Cary Millsap



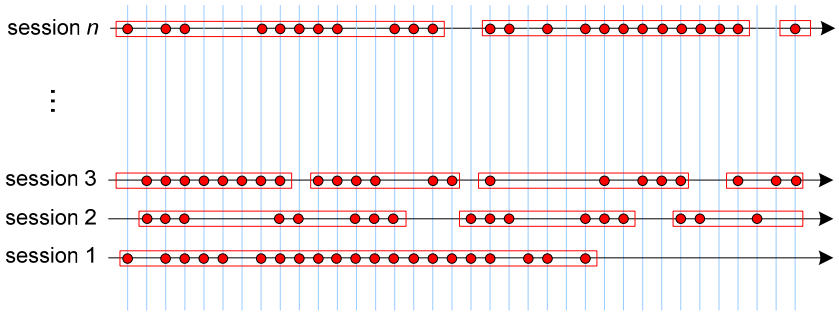
Meanwhile, back at
Oracle Corporation...

EASIER, BETTER ACCESS TO SYSTEM WORKLOAD STATISTICS

ACTIVE SESSION HISTORY ...ASH



THE TASK PERSPECTIVE…



Resource consumption (seconds) by task

	Total system		Your task		Other tasks	
Latches	7,502	74%	2	2%	7,500	75%
I/O	275	3%	75	75%	200	2%
Other	2,323	23%	23	23%	2,300	23%
Total	**10,100**	**100%**	**100**	**100%**	**10,000**	**100%**

Resource consumption (seconds) by task

	Total system		Your task		Other tasks	
Latches	7,502	74%	2	2%	7,500	75%
I/O	275	3%	75	75%	200	2%
Other	2,323	23%	23	23%	2,300	23%
Total	**10,100**	**100%**	**100**	**100%**	**10,000**	**100%**

HOW IT ALL FITS...

THE GRAND

UNIFIED

THEORY

G.U.T. proposition I

You must be able to attack response time problems for specific tasks that the business cares about.

TRACE DATA

Where does this task's time go?

ASH ANSWERS QUESTIONS ABOUT WORKLOAD CONTEXT

Who had my lock?

Who had my latch?

Who was eating my CPU?

Who was fighting me for disk?

G.U.T. proposition 2

You need to be able to attack inefficiencies that aren't yet noticeable as user response time problems.

ASH

Who's using the most stuff?

TRACE DATA ANSWERS QUESTIONS ABOUT TASK EFFICIENCY

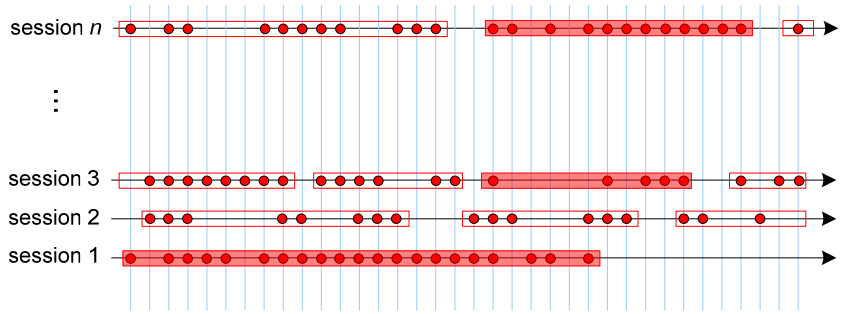
Can this task be performed more efficiently?

ALMOST IMPOSSIBLE TO PROVE THIS WITHOUT TASK DATA

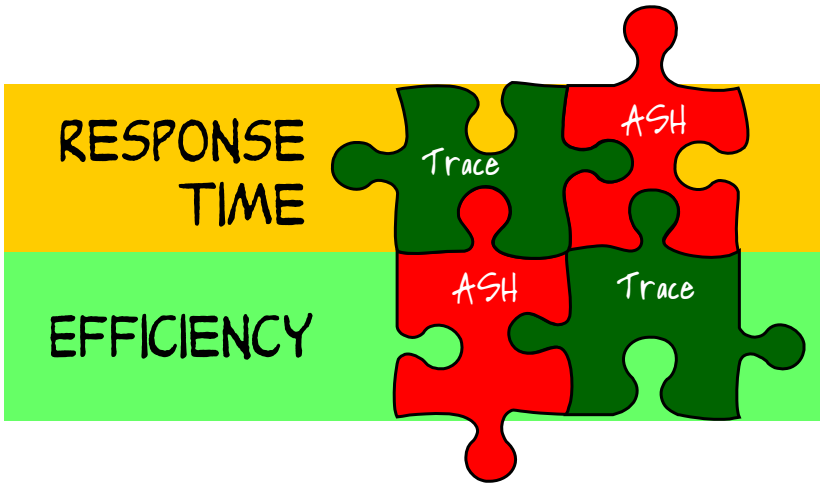
YOU DON'T
HAVE TO TRACE
EVERYTHING

...YOU CAN *SAMPLE*
WITH TRACES, TOO

TRACE *SOME* TASKS



HOW THE TOOLS FIT TOGETHER...



ENOUGH?

Thank you


problems
perspective
inefficiencies
key
attack
grand
indexed
also
noticeable
Method R
consumption
able
name
theory
response time
workload
resource
business
ASH
task
Sometimes
mustparts
context
necessary
user
findunifiedyet
need
know
three
way
http://wordle.net

http://method-r.com

http://carymillsap.blogspot.com

cary.millsap@method-r.com