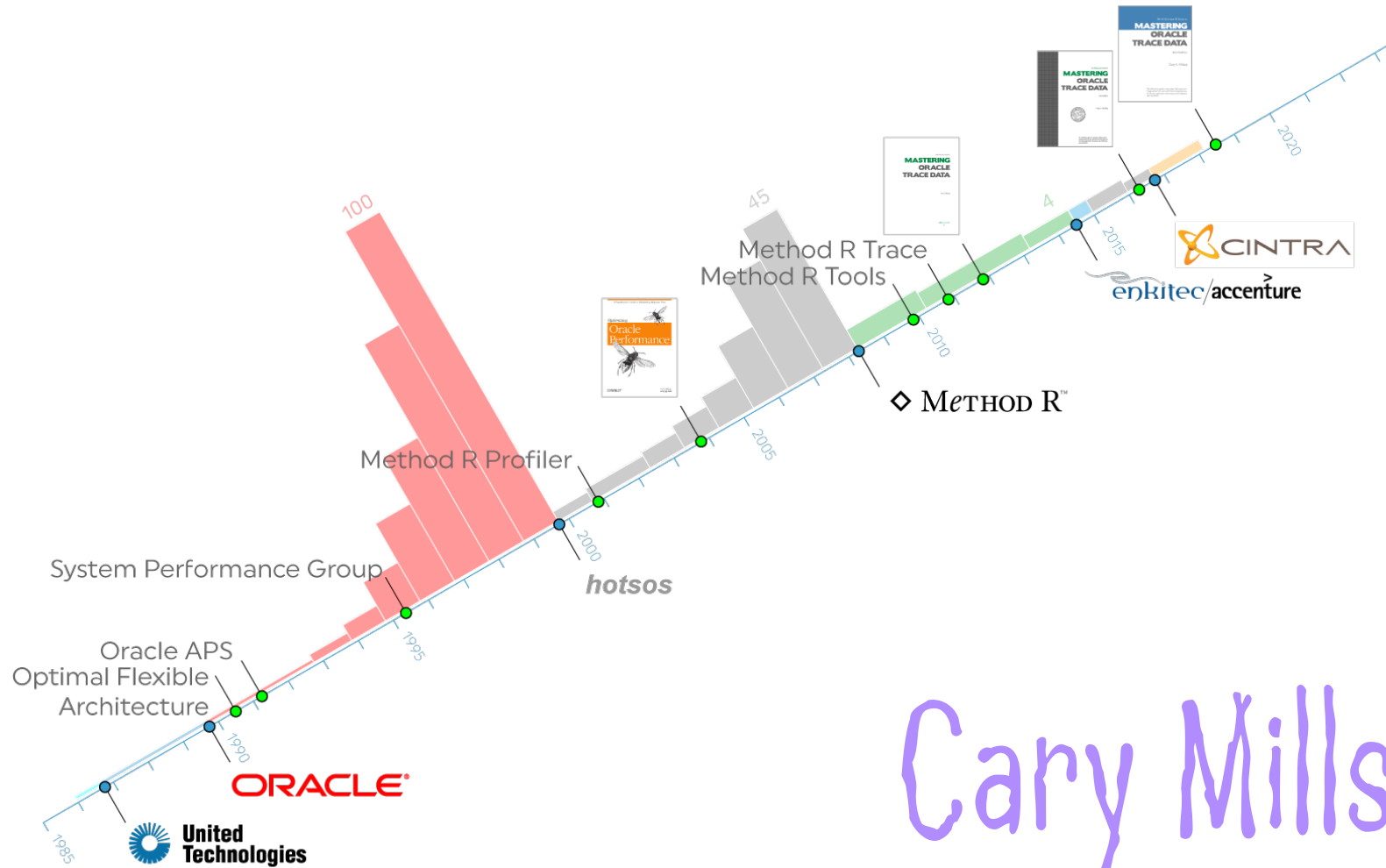


Next — Cary Millsap's

ADVENTURES IN TRACE DATA

Great Lakes Oracle Conference 2018 · Cleveland, Ohio
2:45–3:45p Wednesday 16 May 2018

© 2013, 2018 Cary Millsap



Cary Millsap

1 MOTIVES

2 WARM-UP EXERCISES

3 ACTUAL ADVENTURES

4 WRAP-UP

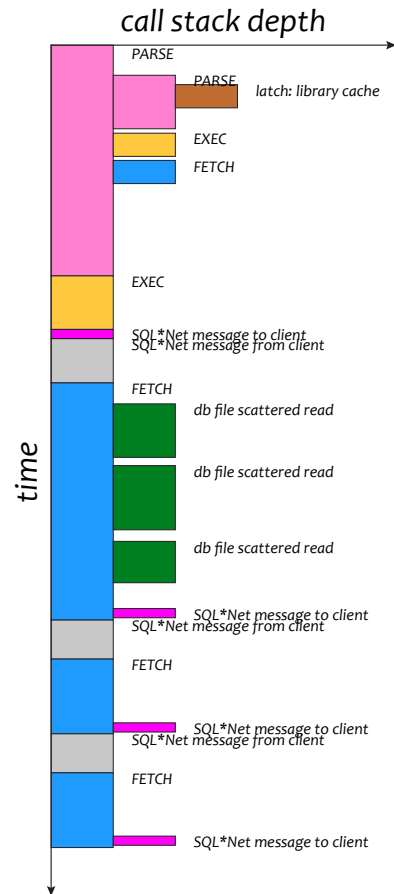
1 MOTIVES

What's in a trace file?

Who cares?

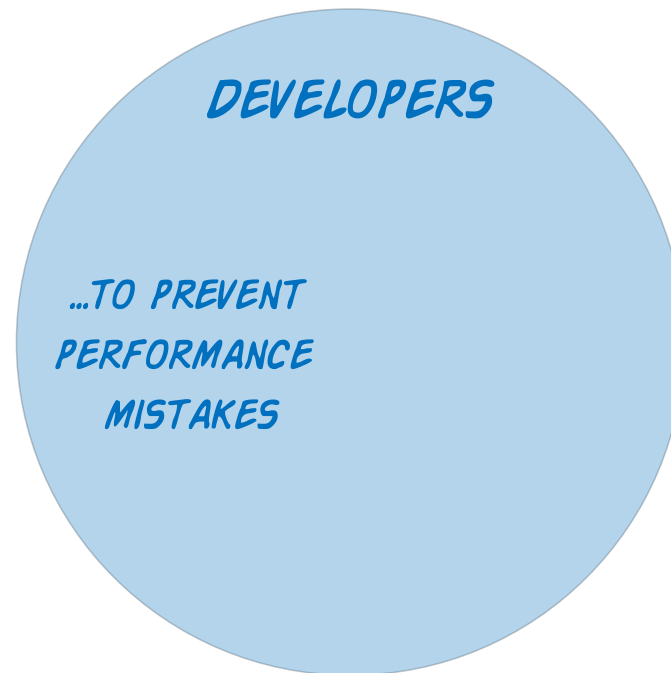
What's in a trace file?

- *EACH CALL (DB, OS) YOUR PROGRAM EXPERIENCED*
- *WHEN IT HAPPENED*
- *HOW LONG IT TOOK*
- *DETAILS ABOUT THE CALL*

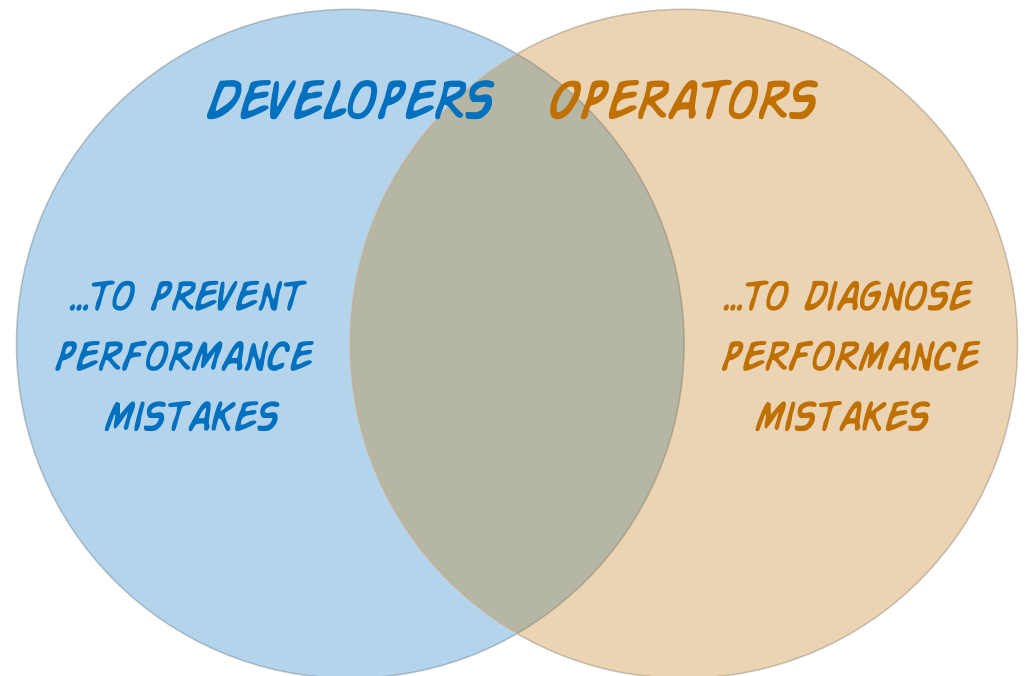


**TRACE FILES
EXPLAIN WHERE
YOUR PROGRAM'S
TIME GOES.**

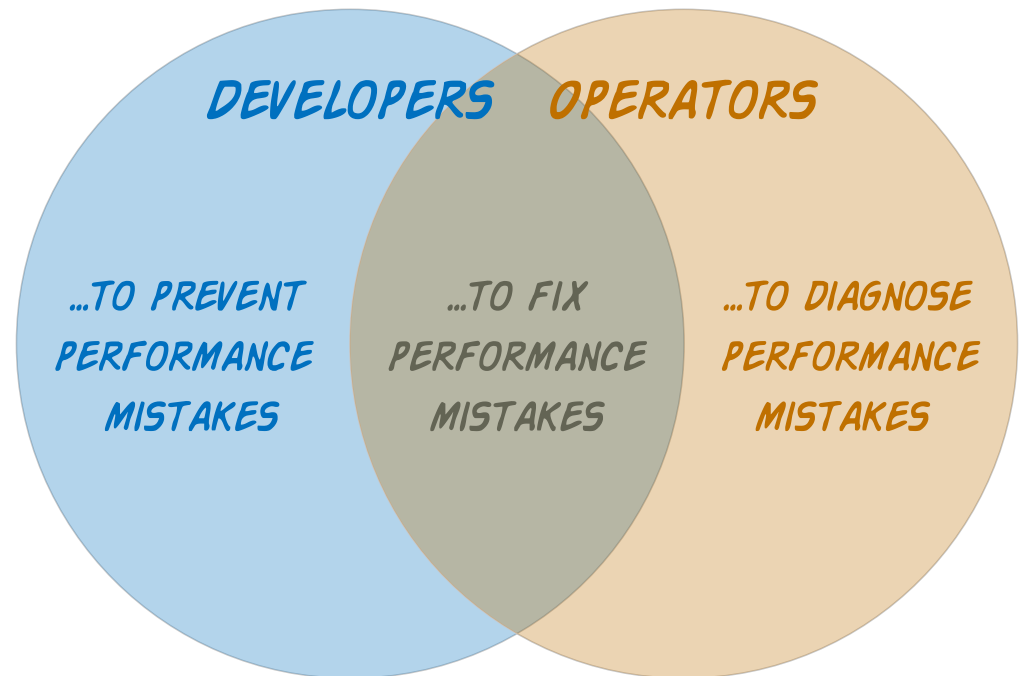
Who cares about trace files?



Who cares about trace files?



Who cares about trace files?



TRACE FILES

- EXPLAIN WHERE YOUR PROGRAM'S
TIME GOES.*
- HELP DEVELOPERS AND DBAS
PREVENT, FIND, AND FIX
PERFORMANCE MISTAKES.*

² WARM-UP EXERCISES

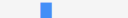
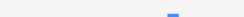
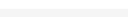
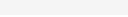
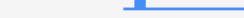
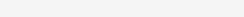
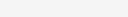
How can you process
a trace file?

- *TKPROF*
- *TRCA*
- *TVDXTAT*
- *PARSETRC*
- *TRCEXTPROF*
- *ORASRP*
- *METHOD R*
- *AND MORE...*

How can you process
a trace file?

I USE ♦ METHOD R™

- *HIGH-PRECISION TRACE FORMATTER*
- *DATA MINING TOOL*
- *VARIOUS UTILITIES*

	Subroutine	Duration		Calls	Duration per call (seconds)			
		seconds	% R		mean	min	skew	max
1	CPU: FETCH dbcalls ›	6.997937	84.9%	195,076	0.000036	.		5.038223
2	cell single block physical read ›	1.418016	17.2%	5,512	0.000257	0.000206		0.000534
3	gc cr disk read ›	0.161261	2.0%	2,747	0.000059	0.000018		0.000208
4	unaccounted-for between calls ›	0.006447	0.1%	12	0.000537	0.000024		0.005797
5	SQL*Net message from client ›	0.000762	.	3	0.000254	0.000130		0.000349
6	gc cr block 2-way ›	0.000364	.	4	0.000091	0.000069		0.000130
7	Disk file operations I/O ›	0.000033	.	2	0.000016	0.000016		0.000017
8	SQL*Net message to client ›	0.000005	.	3	0.000002	.		0.000003
9	CPU: CLOSE dbcalls ›	.	.	8	.	.		.
10	CPU: EXEC dbcalls ›	.	.	8	.	.		.
11	CPU: PARSE dbcalls ›	.	.	7	.	.		.
12	uncategorizable within synthetic dbcalls ›	.	.	1	.	.		.
13	CPU: COMMIT R-ONLY dbcalls ›	.	.	1	.	.		.
14	unaccounted-for within dbcalls ›	-0.340311	-4.1%	195,100	-0.000002	-0.363766		0.001370
15	Total (14)	8.244514	100.0%					



*AN APPROXIMATE PROFILE,
CREATED BY MRSKEW*

Response time by subroutine call name
mrskew "sample.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
cell single block physical read	4.644927	62.2%	6,899	0.000673	0.000528	0.028060
FETCH	2.127676	28.5%	1	2.127676	2.127676	2.127676
EXEC	0.672898	9.0%	2	0.336449	0.003999	0.668899
SQL*Net message from client	0.025852	0.3%	3	0.008617	0.001523	0.022614
PARSE	0.001000	0.0%	2	0.000500	0.000000	0.001000
6 others	0.000269	0.0%	24	0.000011	0.000000	0.000060
TOTAL (11)	7.472622	100.0%	6,931	0.001078	0.000000	2.127676

**MUST SUM TO THE
EXPERIENCE DURATION.**

**MUST MATCH THE DURATION OF THE
EXPERIENCE YOU'RE MEASURING.**

Response time by subroutine call name
mrskew "sample.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
-----	-----	-----	-----	-----	-----	-----
cell single block physical read	4.644927	62.2%	6,899	0.000673	0.000528	0.028060
FETCH	2.127676	28.5%	1	2.127676	2.127676	2.127676
EXEC	0.672898	9.0%	2	0.336449	0.003999	0.668899
SQL*Net message from client	0.025852	0.3%	3	0.008617	0.001523	0.022614
PARSE	0.001000	0.0%	2	0.000500	0.000000	0.001000
6 others	0.000269	0.0%	24	0.000011	0.000000	0.000060
-----	-----	-----	-----	-----	-----	-----
TOTAL (11)	7.472622	100.0%	6,931	0.001078	0.000000	2.127676

Response time histogram (disk) for a given call name pattern
mrskew --rc=disk.rc --name='cell single block physical read' "sample.trc"

RANGE {min ≤ e < max}	DURATION	%	CALLS	MEAN	MIN	MAX
-----	-----	-----	-----	-----	-----	-----
1. 0ms 5ms	4.465045	96.1%	6,883	0.000649	0.000528	0.001752
2. 5ms 10ms	0.062428	1.3%	7	0.008918	0.006324	0.009537
3. 10ms 15ms	0.089394	1.9%	8	0.011174	0.010217	0.012536
4. 15ms 20ms						
5. 20ms 25ms						
6. 25ms 50ms	0.028060	0.6%	1	0.028060	0.028060	0.028060
7. 50ms 100ms						
8. 100ms 1,000ms						
9. 1,000ms +∞						
-----	-----	-----	-----	-----	-----	-----
TOTAL (9)	4.644927	100.0%	6,899	0.000673	0.000528	0.028060

Response time histogram (disk) for a given call name pattern
mrskew --rc=disk.rc --name='cell single block physical read' "sample.trc"

RANGE {min ≤ e < max}			DURATION	%	CALLS	MEAN	MIN	MAX
1.	0ms	5ms	4.465045	96.1%	6,883	0.000649	0.000528	0.001752
2.	5ms	10ms	0.062428	1.3%	7	0.008918	0.006324	0.009537
3.	10ms	15ms	0.089394	1.9%	8	0.011174	0.010217	0.012536
4.	15ms	20ms						
5.	20ms	25ms						
6.	25ms	50ms	0.028060	0.6%	1	0.028060	0.028060	0.028060
7.	50ms	100ms						
8.	100ms	1,000ms						
9.	1,000ms	+∞						
TOTAL (9)			4.644927	100.0%	6,899	0.000673	0.000528	0.028060

Response time histogram (ssd) for a given call name pattern
mrskew --rc=ssd.rc --name='cell single block physical read' "sample.trc"

RANGE {min ≤ e < max}			DURATION	%	CALLS	MEAN	MIN	MAX
1.	0μs	15μs						
2.	15μs	25μs						
3.	25μs	50μs						
4.	50μs	100μs						
5.	100μs	500μs						
6.	500μs	1,000μs	4.436500	95.5%	6,859	0.000647	0.000528	0.000999
7.	1,000μs	10,000μs	0.090973	2.0%	31	0.002935	0.001036	0.009537
8.	10,000μs	+∞	0.117454	2.5%	9	0.013050	0.010217	0.028060
TOTAL (8)			4.644927	100.0%	6,899	0.000673	0.000528	0.028060

Response time by execution id (modified)

mrskew --name='db file sequential read' --group='"\$p1:\$p2"' --gl=BLOCK-ID "sample.trc"

BLOCK-ID	DURATION	%	CALLS	MEAN	MTN	MAX
7:138	0.000033	32.0%	1	0.000033	0.000033	0.000033
8:2039938	0.000011	10.7%	1	0.000011	0.000011	0.000011
8:2318466	0.000011	10.7%	1	0.000011	0.000011	0.000011
9:127106	0.000011	10.7%	1	0.000011	0.000011	0.000011
8:102530	0.000010	9.7%	1	0.000010	0.000010	0.000010
8:1114248	0.000009	8.7%	1	0.000009	0.000009	0.000009
8:130	0.000009	8.7%	1	0.000009	0.000009	0.000009
9:130	0.000009	8.7%	1	0.000009	0.000009	0.000009
TOTAL (8)	0.000103	100.0%	8	0.000013	0.000009	0.000033

...FILE ID AND BLOCK ID FOR READ CALLS.

*A HISTOGRAM OF ROW COUNTS
GROUPED BY DATABASE CALL NAME*

```
Rows returned by dbcall name
mrskew --name=:dbcall --select='$row' --slabel=ROWS --precision=0 "sample.trc"
```

CALL-NAME	ROWS	%				
EXEC	1	100.0%				
CLOSE	0	0.0%	4	0	0	0
PARSE	0	0.0%	2	0	0	0
FETCH	0	0.0%	1	0	0	0
XCTEND	0	0.0%	1	0	0	0
BINDS						
TOTAL (6)	1	100.0%	10	0	0	1

SELECTING ROW COUNT
INSTEAD OF A DURATION.

*A LIST OF DATABASE CALLS
IN CHRONOLOGICAL ORDER*

Database calls in chronological order
mrskew **--rc=calls.rc** --top=0 "sample.trc"

END-TIME	LINE	PARSE-ID	EXEC-ID	CALL-NAME(BOUND-VALUES)	STATEMENT-TEXT
COMPLEX LIST OF OPTIONS SAVED IN A FILE.					
4330608.950084	28	0	0	CLOSE	#140441051716304:/Users/carymillsap/
4330608.951035	33	33	0	PARSE	BEGIN :file_id := 7; :blk_id := 142;
4330608.954476	34	33	44	BINDS(,)	BEGIN :file_id := 7; :blk_id := 142;
4330608.954477	44	33	44	EXEC(,)	BEGIN :file_id := 7; :blk_id := 142;
4330608.956375	46	33	44	CLOSE(,)	BEGIN :file_id := 7; :blk_id := 142;
4330608.956877	55	55	0	PARSE	select owner, segment_name from dba_
4330608.958309	60	60	0	• PARSE	• select text from view\$ where rowid
4330608.961047	61	60	67	• BINDS(00002D44.0004.0001)	• select text from view\$ where rowid
4330608.961048	67	60	67	• EXEC(00002D44.0004.0001)	• select text from view\$ where rowid
4330608.961156	68	60	67	• FETCH(00002D44.0004.0001)	• select text from view\$ where rowid
4330608.961314	70	60	67	• CLOSE(00002D44.0004.0001)	• select text from view\$ where rowid
4330608.962028	75	75	0	• PARSE	• select text from view\$ where rowid
4330608.962353	76	75	82	• BINDS(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.962354	82	75	82	• EXEC(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.962432	83	75	82	• FETCH(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.962535	85	75	82	• CLOSE(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.964338	90	90	0	• PARSE	• select text from view\$ where rowid
4330608.964617	91	90	97	• BINDS(00002D40.0003.0001)	• select text from view\$ where rowid
4330608.964618	97	90	97	• EXEC(00002D40.0003.0001)	• select text from view\$ where rowid
4330608.964692	98	90	97	• FETCH(00002D40.0003.0001)	• select text from view\$ where rowid
4330608.964797	100	90	97	• CLOSE(00002D40.0003.0001)	• select text from view\$ where rowid
4330608.971414	101	101	0	• PARSE	• select text from view\$ where rowid
4330608.971630	102	101	108	• BINDS(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.971631	108	101	108	• EXEC(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.971695	109	101	108	• FETCH(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.971757	110	101	108	• CLOSE(00002D42.0000.0001)	• select text from view\$ where rowid
4330608.973604	111	111	0	• PARSE	• select text from view\$ where rowid
4330608.973819	112	111	118	• BINDS(00002D40.0003.0001)	• select text from view\$ where rowid
4330608.973820	113	111	118	• EXEC(00002D40.0003.0001)	• select text from view\$ where rowid

```
$ more src/rc/calls.rc
--group='sprintf("%17.6f %10d %10d %10d %-40.40s %-.46s", $tim, $line, $parse_id, $exec_id, "· "x$dep.$name.
(scalar(@bind)?("".join(",","@bind)."):""), "· "x$dep.$sql)'
#          12345678901234567 1234567890 1234567890 1234567890 1234567890123456789012345678901234567890 123..
--glabel='          END-TIM          LINE  PARSE-ID    EXEC-ID  CALL-NAME(BOUND-VALUES)          STATEMENT-TEXT'
--alldepths --nohistogram --sort=1na --name=:dbcall --top=0
```

THE OPTIONS IN CALLS.RC.

SOME AMAZING STUFF YOU CAN FIND OUT...

Do I have EXEC calls with library cache misses? How much time do they cost?

Does one database file have worse latencies than the others?

Does the database pread() the same blocks over and over?

3 ACTUAL ADVENTURES

**WHY IS THIS
TAKING SO LONG?**

35-second experience.

Seems too long.

Anything we can do?

FIRST, ...A TRACE FILE.

Response time by subroutine call name
 mrskew "gdglqal_ora_68240.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	3,385.621256	99.7%	14,879	0.227544	0.001132	3,360.817216
EXEC	4.399228	0.1%	5,970	0.000737	0.000000	0.078001
gc current block 2-way	1.289916	0.0%	39	0.033075	0.000150	0.221993
gc cr block 2-way	1.111851	0.0%	62	0.017933	0.000141	0.218596
row cache lock	0.846364	0.0%	1,303	0.000650	0.000002	0.025897
FETCH	0.717607	0.0%	14,877	0.000048	0.000000	0.031201
PARSE	0.062400	0.0%	295	0.000212	0.000000	0.015600
db file sequential read	0.041004	0.0%	5	0.008201	0.005578	0.011897
rdbms ipc reply	0.036705	0.0%	34	0.001080	0.000013	0.006390
gc current block busy	0.032970	0.0%	1	0.032970	0.032970	0.032970
10 others	0.120624	0.0%	15,192	0.000008	0.000000	0.014807
TOTAL (20)	3,394.279925	100.0%	52,657	0.064460	0.000000	3,360.817216

...DOESN'T MATCH THE ~35-SECOND EXPERIENCE.

*WHY THE HUGE
"SQL*NET MESSAGE FROM CLIENT"?*

Response time by line number for a given call name pattern (modified)

mrskew --group='\$line' --glabel=LINE --name='SQL*Net message from client' "gdglqa1_ora_68240.trc"

LINE	DURATION	%	CALLS	MEAN	MIN	MAX
-----	-----	-----	-----	-----	-----	-----
489430	3,360.817216	99.3%	1	3,360.817216	3,360.817216	3,360.817216
80	0.116487	0.0%	1	0.116487	0.116487	0.116487
489422	0.086658	0.0%	1	0.086658	0.086658	0.086658
273210	0.083400	0.0%	1	0.083400	0.083400	0.083400
426949	0.043966	0.0%	1	0.043966	0.043966	0.043966
385102	0.032863	0.0%	1	0.032863	0.032863	0.032863
289347	0.026835	0.0%	1	0.026835	0.026835	0.026835
23623	0.019732	0.0%	1	0.019732	0.019732	0.019732
286947	0.019073	0.0%	1	0.019073	0.019073	0.019073
301786	0.017868	0.0%	1	0.017868	0.017868	0.017868
14,869 others	24.357158	0.7%	14,869	0.001638	0.001132	0.015516
-----	-----	-----	-----	-----	-----	-----
TOTAL (14,879)	3,385.621256	100.0%	14,879	0.227544	0.001132	3,360.817216

ONE CALL CONSUMED 99.3%
OF THE "SQL*NET MESSAGE
FROM CLIENT" DURATION.

*QUICK CHECK IN THE TRACE FILE
CONFIRMS THAT THIS CALL IS OUTSIDE
THE SCOPE OF THE USER EXPERIENCE.*

A large, stylized purple skull and crossbones graphic is centered on a black background. The skull has large, dark eye sockets and a wide, toothy grin. Two purple bones are crossed in an 'X' shape over the lower part of the skull. The word "DANGER!!" is written in a bold, orange, blocky font across the middle of the skull.

DANGER!!

THIS IS **ALWAYS** THE FIRST PROBLEM
PEOPLE HAVE WITH TRACE FILES:

YOU WANT **ONLY** THE DATA
THAT EXPLAINS THE **EXPERIENCE**
YOU'RE TRYING TO DIAGNOSE.

Response time by line number for a given call name pattern (modified)

mrskew --group='\$line' --glabel=LINE --name='SQL*Net message from client' "gdglqa1_ora_68240.trc"

LINE	DURATION	%	CALLS	MEAN	MIN	MAX
489430	3,360.817216	99.3%	1	3,360.817216	3,360.817216	3,360.817216
80	0.116487	0.0%	1	0.116487	0.116487	0.116487
489422	0.086658	0.0%	1	0.086658	0.086658	0.086658
273210	0.083400	0.0%	1	0.083400	0.083400	0.083400
426949	0.043966	0.0%	1	0.043966	0.043966	0.043966
385102	0.032863	0.0%	1	0.032863	0.032863	0.032863
289347	0.026835	0.0%	1	0.026835	0.026835	0.026835
23623	0.019732	0.0%	1	0.019732	0.019732	0.019732
286947	0.019073	0.0%	1	0.019073	0.019073	0.019073
301786	0.017868	0.0%	1	0.017868	0.017868	0.017868
14,869 others	24.357158	0.7%	14,869	0.001638	0.001132	0.015516
TOTAL (14,879)	3,385.621256	100.0%	14,879	0.227544	0.001132	3,360.817216

\$ mrcallrm --lines 489430 gdglqa1_ora_68240.trc > fixed.trc

THE LINE NUMBER OF
THE CALL TO "RM".

THIS IS A METHOD R WORKBENCH UTILITY. IT
MANIPULATES THE TIME VALUE ON EVERY LINE IN
THE TRACE FILE AFTER THE LINES YOU DELETE.

NOW, THE CORRECTED TRACE FILE...

Response time by subroutine call name
 mrskew "fixed.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	24.804040	74.1%	14,879	0.001667	0.000000	0.116487
EXEC	4.399228	13.1%	5,970	0.000737	0.000000	0.078001
gc current block 2-way	1.289916	3.9%	39	0.033075	0.000150	0.221993
gc cr block 2-way	1.111851	3.3%	62	0.017933	0.000141	0.218596
row cache lock	0.846364	2.5%	1,303	0.000650	0.000002	0.025897
FETCH	0.717607	2.1%	14,877	0.000048	0.000000	0.031201
PARSE	0.062400	0.2%	295	0.000212	0.000000	0.015600
db file sequential read	0.041004	0.1%	5	0.008201	0.005578	0.011897
rdbms ipc reply	0.036705	0.1%	34	0.001080	0.000013	0.006390
gc current block busy	0.032970	0.1%	1	0.032970	0.032970	0.032970
10 others	0.120624	0.4%	15,192	0.000008	0.000000	0.014807
TOTAL (20)	33.462709	100.0%	52,657	0.000635	0.000000	0.221993

THERE YOU GO.

Response time by subroutine call name
 mrskew "fixed.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	24.804040	74.1%	14,879	0.001667	0.000000	0.116487
EXEC	4.399228	13.1%	5,970	0.000737	0.000000	0.078001
gc current block 2-way	1.289916	3.9%	39	0.033075	0.000150	0.221993
gc cr block 2-way	1.11185		62	0.017933	0.000141	0.218596
row cache lock	0.846364	2.5%	1,303	0.000650	0.000002	0.025897
FETCH	0.717607	2.1%	14,877	0.000048	0.000000	0.031201
PARSE	0.062400	0.2%	295	0.000212	0.000000	0.015600
db file sequential read	0.041004	0.1%	5	0.008201	0.005578	0.011897
rdbms ipc reply	0.036705	0.1%	34	0.001080	0.000013	0.006390
gc current block busy	0.032970	0.1%	1	0.032970	0.032970	0.032970
10 others	0.120624	0.4%	15,192	0.000008	0.000000	0.014807
TOTAL (20)	33.462709	100.0%	52,657	0.000635	0.000000	0.221993

THE PROBLEM.

Response time by subroutine call name
 mrskew "fixed.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	24.804040	74.1%	14,879	0.001667	0.000000	0.116487
EXEC	4.399228	13.1%	5,970	0.000737	0.000000	0.078001
gc current block 2-way	1.289916	3.9%	39	0.033075	0.000150	0.221993
gc cr block 2-way	1.111851	3.3%	62	0.033075	0.000150	0.218596
row cache lock	0.846364	2.5%	1,303	0.000650	0.000002	0.025897
FETCH	0.717607	2.1%	14,877	0.000048	0.000000	0.031201
PARSE	0.062400	0.2%	295	0.000212	0.000000	0.015600
db file sequential read	0.041004	0.1%	5	0.008201	0.005578	0.011897
rdbms ipc reply	0.036705	0.1%	34	0.001080	0.000013	0.006390
gc current block busy	0.032970	0.1%	1	0.032970	0.032970	0.032970
10 others	0.120624	0.4%	15,192	0.000008	0.000000	0.014807
TOTAL (20)	33.462709	100.0%	52,657	0.000635	0.000000	0.221993

WHY SO MANY?

*HOW MANY ROWS DOES
EACH STATEMENT PROCESS?*

Rows returned by dbcall name (modified)

mrskew --name=:dbcall --select='\$row' --slabel=ROWS --precision=0 --group='substr(\$sql,0,20)."' --gl=SQL "fixed.trc"

SQL	ROWS	%	CALLS	MEAN	MIN	MAX
-----	-----	-----	-----	----	---	---
select TM_DIM_PR_TT_...	6,762	35.1%	5,934	1	0	2
select TM_TT01.KODE ...	5,018	26.0%	2,723	2	0	2
select TM_DATAFELT_B...	1,390	7.2%	2,784	0	0	1
select TM_TT_DETALJ0...	1,325	6.9%	1,392	1	0	2
select TM_HIST_KONT_...	1,254	6.5%	3,772	0	0	2
select TM_LONNSGRUPP...	939	4.9%	475	2	0	2
select TM_FAKTOR01.N...	743	3.9%	1,489	0	0	1
select TO_CHAR(TM_AV...	546	2.8%	775	1	0	2
select TM_INTERNAKK0...	331	1.7%	665	0	0	1
select distinct TM_T...	301	1.6%	157	2	0	2
34 others	671	3.5%	976	1	0	2
-----	-----	-----	-----	----	---	---
TOTAL (44)	19,280	100.0%	21,142	1	0	2

**NO SQL STATEMENT EVER MANIPULATES
MORE THAN 2 ROWS PER DATABASE CALL.**

*WHY NEVER MORE THAN
2 ROWS PER FETCH?*

value=0

Bind#6

oacdtty=02 mxl=22(22) mxlc=00 mal=00 scl=00 pre=00
oacflg=00 fl2=1000001 frm=00 csi=00 siz=0 off=152
kxsbbbf=2ad50880 bln=22 avl=01 flg=01
value=0

Bind#7

oacdtty=96 mxl=32(12) mxlc=00 mal=00 scl=00 pre=00
oacflg=00 fl2=1000001 frm=01 csi=178 siz=0 off=176
kxsbbbf=2ad50898 bln=32 avl=01 flg=01
value=" "

Bind#8

oacdtty=96 mxl=32(01) mxlc=00 mal=00 scl=00 pre=00
oacflg=00 fl2=1000001 frm=01 csi=178 siz=0 off=208
kxsbbbf=2ad508b8 bln=32 avl=01 flg=01
value="J"

EXEC #5:c=62400,e=67062,p=0,cr=905,cu=0,mis=1,r=0,dep=0,og=1,tim=820437697591

WAIT #5: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437697622

FETCH #5:c=0,e=62,p=0,cr=10,cu=0,mis=0,r=1,dep=0,og=1,tim=820437699246

WAIT #5: nam='SQL*Net message from client' ela= 1444 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437699246

WAIT #5: nam='SQL*Net message to client' ela= 0 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437699316

FETCH #5:c=0,e=87,p=0,cr=12,cu=0,mis=0,r=2,dep=0,og=1,tim=820437699356

WAIT #5: nam='SQL*Net message from client' ela= 1451 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437700840

WAIT #5: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437701137

FETCH #5:c=0,e=100,p=0,cr=14,cu=0,mis=0,r=2,dep=0,og=1,tim=820437701181

WAIT #5: nam='SQL*Net message from client' ela= 1218 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437702426

WAIT #5: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437702485

FETCH #5:c=0,e=58,p=0,cr=8,cu=0,mis=0,r=1,dep=0,og=1,tim=820437702505

WAIT #5: nam='SQL*Net message from client' ela= 1326 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437703864

FETCH #4:c=0,e=7,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=1,tim=820437703892

WAIT #4: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437703948

WAIT #4: nam='SQL*Net message from client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437705168

WAIT #1: nam='SQL*Net message to client' ela= 1 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437705194

FETCH #1:c=0,e=20,p=0,cr=0,cu=0,mis=0,r=2,dep=0,og=1,tim=820437705201

WAIT #1: nam='SQL*Net message from client' ela= 1264 driver id=1413697536 #bytes=1 p3=0 obj#=54819 tim=820437706491

**THREE OF THE 8 BINDS
UPON CURSOR #5.**

6 ROWS IN 4 NETWORK ROUNDTRIPS.

...COULD HAVE BEEN JUST ONE.

HOW FAST SHOULD THIS THING RUN?

Response time by subroutine call name
mrskew "fixed.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	24.804040	74.1%	14,879	0.001667	0.000000	0.116487
EXEC	4.399228	13.1%	5,970	0.000737	0.000000	0.078001
gc current block 2-way	1.289916	3.9%	39	0.033075	0.000150	0.221993
gc cr block 2-way	1.111851	3.3%	6			8596
row cache lock	0.846364	2.5%	1,306			5897
FETCH	0.717607	2.1%	14,879			1201
PARSE	0.062400	0.2%	29			5600
db file sequential read	0.041004	0.1%	5	0.008201	0.003378	0.011897
rdbms ipc reply	0.036705	0.1%	34	0.001080	0.000013	0.006390
gc current block busy	0.032970	0.1%	1	0.032970	0.032970	0.032970
10 others	0.120624	0.4%	15,192	0.000008	0.000000	0.014807
TOTAL (20)	33.462709	100.0%	52,657	0.000635	0.000000	0.221993

WITH ARRAYSIZE=20
INSTEAD OF ARRAYSIZE=2,
THIS WILL BECOME 1,488.
 $14,879 \times 2/20$

Response time by subroutine call name
 mrskew "fixed.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	24.804040	74.1%	14,879	0.001667	0.000000	0.116487
EXEC	4.399228	13.1%	5,970	0.000737	0.000000	0.078001
gc current block 2-way	1.289916	3.9%	39	0.033075	0.000150	0.221993
gc cr block 2-way	1.1118				0.000141	0.218596
row cache lock	0.8463				0.000002	0.025897
FETCH	0.7176				0.000000	0.031201
PARSE	0.062400	0.2%	295	0.000212	0.000000	0.015600
db file sequential read	0.041004	0.1%	5	0.008201	0.005578	0.011897
rdbms ipc reply	0.036705	0.1%	34	0.001080	0.000013	0.006390
gc current block busy	0.032970	0.1%	1	0.032970	0.032970	0.032970
10 others	0.120624	0.4%	15,192	0.000008	0.000000	0.014807
TOTAL (20)	33.462709	100.0%	52,657	0.000635	0.000000	0.221993

...THIS WILL BECOME ~2.480.

$24.804 \times 2/20$

Response time by subroutine call name
 mrskew "fixed.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	24.804040	74.1%	14,879	0.001667	0.000000	0.116487
EXEC	4.399228	13.1%	5,970	0.000737	0.000000	0.078001
gc current block 2-way	1.289916	3.9%	39	0.033075	0.000150	0.221993
gc cr block 2-way	1.111851	3.3%	62	0.017933	0.000141	0.218596
row cache lock	0.846364	2.5%	1,303	0.000650	0.000002	0.025897
FETCH	0.717607	2.1%	14,877	0.000048	0.000000	0.031201
PARSE	0.062400	0.2%	295	0.000212	0.000000	0.015600
db file sequential read	0.041004	0.1%	5	0.008201	0.005578	0.011897
rdbms ipc reply	0.036705	0.1%	34	0.001080	0.000013	0.006390
gc current block busy	0.032970	0.1%	1	0.032970	0.032970	0.032970
10 others	0.120624	0.4%	15,192	0.000008	0.000000	0.014807
TOTAL (20)	33.462709	100.0%	52,657	0.000635	0.000000	0.221993

...AND THIS WILL BECOME ~11.

33.462 - 24.804 + ~(24.804 × 2/20)

OPPORTUNITY

ARRAYSIZE IN (20, 100, ...).

SHOULD RUN 11 SECONDS INSTEAD OF 35.

REWRITING IN PL/SQL SHOULD SAVE EVEN MORE TIME.

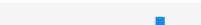
**AREN'T 5.9MS SAN
LATENCIES GOOD ENOUGH?**

21-hour experience.

Seems too long.

Anything we can do?

FIRST, ...A TRACE FILE.

Subroutine	Duration		Calls	Duration per call (seconds)			
	seconds	% R		mean	min	skew	max
1 db file sequential read ›	59,081.406104	76.8%	10,013,394	0.005900	0.000010		15.853019
2 log buffer space ›	6,308.758563	8.2%	9,476	0.665762	0.000004		1.010092
3 free buffer waits ›	4,688.730190	6.1%	200,198	0.023420	0.000004		1.021281
4 CPU: EXEC dbcalls ›	4,210.810000	5.5%	43,946	0.095818	0.000000		5.400000
5 24 other subroutines	2,652.639343	3.4%	705,741	0.003759	-1,004.215969		1,002.994167
6 Total (28)	76,942.344200	100.0%					

21 HOURS, 22 MINUTES.

Response time by subroutine call name (modified)
mrskev --top=5 "prod_ora_14920.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
db file sequential read	59,081.406102	76.6%	10,013,394	0.005900	0.000010	15.853019
log buffer space	6,308.758563	8.2%	9,476	0.665762	0.000004	1.010092
free buffer waits	4,688.730190	6.1%	200,198	0.023420	0.000004	1.021281
EXEC	4,214.190000	5.5%	36,987	0.113937	0.000000	5.400000
log file switch completion	1,552.471890	2.0%	1,853	0.837815	0.000006	1.013093
23 others	1,303.198855	1.7%	70,400	0.018511	0.000000	8.964706
TOTAL (28)	77,148.755600	100.0%	10,332,308	0.007467	0.000000	15.853019

CLOSE ENOUGH.

$(77,148.755600 - 76,942.344200) / 76,942.344200 = 0.00268$

Response time by subroutine call name (modified)
 mrskew --top=5 "prod_ora_14920.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
db file sequential read	59,081.406102	76.6%	10,013,394	0.005900	0.000010	15.853019
log buffer space	6,308.758563	8.2%	9,476	0.665762	0.000004	1.010092
free buffer waits	4,688.730190	6.1%	200,198	0.023420	0.000004	1.021281
EXEC	4,214.190000	5.5%	1,853	0.837815	0.000006	5.400000
log file switch completion	1,552.471890	2.0%	70,400	0.018511	0.000000	8.964706
23 others	1,303.198855	1.7%				
TOTAL (28)	77,148.755600	100.0%	10,332,308	0.007467	0.000000	15.853019

76.6% IS OS READ CALLS.

Response time by subroutine call name (modified)
 mrskew --top=5 "prod_ora_14920.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
db file sequential read	59,081.406102	76.6%	10,013,394	0.005900	0.000010	15.853019
log buffer space	6,308.758563	8.2%	9,476	0.665762	0.000004	1.010092
free buffer waits	4,688.730190	6.1%	200,198	0.023420	0.000004	1.021281
EXEC	4,214.190000	5.5%	36,987	0.113950	0.000000	1.400000
log file switch completion	1,552.471890	2.0%	1,853	0.837815	0.000000	1.013093
23 others	1,303.198855	1.7%	70,400	0.018511	0.000000	8.964706
TOTAL (28)	77,148.755600	100.0%	10,332,308	0.007467	0.000000	15.853019

...10 MILLION OF THEM.

Response time by subroutine call name (modified)
mrskew --top=5 "prod_ora_14920.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
db file sequential read	59,081.406102	76.6%	10,013,394	0.005900	0.000010	15.853019
log buffer space	6,308.758563	8.2%	9,476	0.665762	0.000004	1.010092
free buffer waits	4,688.730190	6.1%	200,198	0.023420	0.000004	1.021281
EXEC	4,214.190000	5.5%	36,987	0.113937	0.000000	5.400000
log file switch completion	1,552.471890	2.0%				1.013093
23 others	1,303.198855	1.7%				8.964706
TOTAL (28)	77,148.755600	100.0%	10,332,308	0.000000	0.000000	15.853019

BUT AVERAGE LATENCY IS
ONLY 5.9 MILLISECONDS

CONGRATULATIONS!

RIGHT?

MMM...
NOT GOOD.

Response time by subroutine call name (modified)
 mrskew --top=5 "prod_ora_14920.trc"

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
db file sequential read	59,081.406102	76.6%	10,013,394	0.005900	0.000010	15.853019
log buffer space	6,308.758563	8.2%	9,476	0.665762	0.000004	1.010092
free buffer waits	4,688.730190	6.1%	200,198	0.023420	0.000004	1.021281
EXEC	4,214.190000	5.5%	36,987	0.113937	0.000000	5.400000
log file switch completion	1,552.471890	2.0%	1,853	0.837815	0.000006	1.013093
23 others	1,303.198855	1.7%	70,400	0.018511	0.000000	8.964706
TOTAL (28)	77,148.755600	100.0%	10,332,308	0.007467	0.000000	15.853019

*WHAT ARE THE INDIVIDUAL
READ CALL LATENCIES?*

Response time histogram (disk) for a given call name pattern (modified)
 mrskew --rc=disk.rc --name='sequential' "prod_ora_14920.trc"

RANGE {min ≤ e < max}			DURATION	%	CALLS	MEAN	MIN	MAX
1.	0ms	5ms	315.628400	0.5%	9,493,734	0.000033	0.000010	0.004999
2.	5ms	10ms	517.751254	0.9%	66,995	0.007728	0.005000	0.009999
3.	10ms	15ms	865.725214	1.5%	69,613	0.012436	0.010000	0.014999
4.	15ms	20ms	869.673298	1.5%	50,212	0.017320	0.015000	0.019999
5.	20ms	25ms	759.207695	1.3%	33,979	0.022343	0.020000	0.024999
6.	25ms	50ms	2,884.671848	4.9%	81,348	0.035461	0.025000	0.049999
7.	50ms	100ms	5,814.227556	9.8%	79,717	0.072936	0.050000	0.099999
8.	100ms	1,000ms	26,057.804096	44.1%	130,471	0.199721	0.100002	0.999717
9.	1,000ms	+∞	20,996.716743	35.5%	7,325	2.866446	1.000184	15.853019
TOTAL (9)			59,081.406102	100.0%	10,013,394	0.005900	0.000010	15.853019

AVERAGE LATENCY FOR THESE
 130,471 CALLS IS 199 MS PER CALL.
 NOT GREAT.

Response time histogram (disk) for a given call name pattern (modified)
 mrskew --rc=disk.rc --name='sequential' "prod_ora_14920.trc"

RANGE {min ≤ e < max}			DURATION	%	CALLS	MEAN	MIN	MAX
1.	0ms	5ms	315.628400	0.5%	9,493,734	0.000033	0.000010	0.004999
2.	5ms	10ms	517.751254	0.9%	66,995	0.007728	0.005000	0.009999
3.	10ms	15ms	865.725214	1.5%	69,613	0.012436	0.010000	0.014999
4.	15ms	20ms	869.673298	1.5%	50,212	0.017320	0.015000	0.019999
5.	20ms	25ms	759.207695	1.3%	33,979	0.022343	0.020000	0.024999
6.	25ms	50ms	2,884.671848	4.9%	81,348	0.035461	0.025000	0.049999
7.	50ms	100ms	5,814.227556	9.8%	79,717	0.072936	0.050000	0.099999
8.	100ms	1,000ms	26,057.804096	44.1%	130,471	0.199721	0.100002	0.999717
9.	1,000ms	+∞	20,996.716743	35.5%	7,325	2.866446	1.000184	15.853019
TOTAL (9)			59,081.406102	100.0%	10,013,394	0.005900	0.000010	15.853019

**AVERAGE LATENCY FOR THESE
7,325 CALLS IS 2.866 SECONDS PER CALL.
RIDICULOUS!**

Response time histogram (disk) for a given call name pattern (modified)
mrskew --rc=disk.rc --name='sequential' "prod_ora_14920.trc"

RANGE {min ≤ e < max}			DURATION	%	CALLS	MEAN	MIN	MAX
1.	0ms	5ms	315.628400	0.5%	9,493,734	0.000033	0.000010	0.004999
2.	5ms	10ms	517.751254	0.9%	66,995	0.007728	0.005000	0.009999
3.	10ms	15ms	865.725214	1.5%	69,613	0.012436	0.010000	0.014999
4.	15ms	20ms	869.673298	1.5%	50,212	0.017320	0.015000	0.019999
5.	20ms	25ms	759.207695	1.3%	33,979	0.022343	0.020000	0.024999
6.	25ms	50ms	2,884.671848	4.9%	81,348	0.035461	0.025000	0.049999
7.	50ms	100ms	5,814.227556	9.8%	79,717	0.072936	0.050000	0.099999
8.	100ms	1,000ms	26,057.804096	44.1%	130,471	0.199721	0.100002	0.999717
9.	1,000ms	+∞	20,996.716743	35.5%	7,325	2.866446	1.000184	15.853019
TOTAL (9)			59,081.406102	100.0%	10,013,394	0.005900	0.000010	15.853019

IF THESE CALLS WERE 50 MS APIECE,
IT WOULD SAVE 42,000 SECONDS.

$$(5,814 + 26,057 + 20,996) - ((79,717 + 130,471 + 7,325) \times .050) = 41,991$$

THAT'S 11.7 HOURS.

*SURELY ALL THOSE BIG-LATENCY READS
ARE BIG ORACLE MULTI-BLOCK READS.
RIGHT?*

Response time histogram (disk) for a given call name pattern (modified)

mrskew --name='sequential' --where='\$ela > .100' --group='\$p3' --gl='BLKS/CALL' "prod_ora_14920.trc"

BLKS/CALL	DURATION	%	CALLS	MEAN	MIN	MAX
1	47,054.520839	100.0%	137,796	0.341480	0.100002	15.853019
TOTAL (1)	47,054.520839	100.0%	137,796	0.341480	0.100002	15.853019

**NOPE. THE 100+ MS CALLS
ARE ALL 1-BLOCK READS.**

*IS IT JUST ONE FILE THAT'S
CAUSING THE LATENCY PROBLEM?*

Response time by SQL text with hints stripped (modified)

mrskew --name='sequential' --where='\$ela > .100' --group='\$p1' --gl='FILE-ID' "prod_ora_14920.trc"

FILE-ID	DURATION	%	CALLS	MEAN	MIN	MAX
128	2,926.935537	6.2%	13,312	0.219872	0.100003	12.382647
376	2,189.664203	4.7%	4,481	0.488655	0.100009	9.748775
373	1,781.236158	3.8%	3,713	0.479730	0.100050	8.541436
378	1,700.316929	3.6%	4,574	0.371735	0.100008	11.439730
381	1,688.533530	3.6%	4,415	0.382454	0.100003	8.556457
164	1,531.551619	3.3%	4,907	0.312116	0.100017	8.840377
347	1,459.798151	3.1%	3,606	0.404825	0.100007	7.912530
348	1,300.775468	2.8%	3,560	0.365386	0.100045	8.877727
89	1,179.555118	2.5%	2,556	0.461485	0.100003	10.905847
152	1,108.177465	2.4%	2,889	0.383585	0.100018	8.491293
139 others	30,187.976661	64.2%	89,783	0.336233	0.100002	15.853019
TOTAL (149)	47,054.520839	100.0%	137,796	0.341480	0.100002	15.853019

*IT'S NOT JUST ONE FILE.
IT'S ALL OF THEM.*

OPPORTUNITY

INVESTIGATE SAN CACHING, WORKLOAD, ETC.

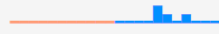
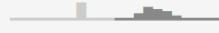
SHOULD RUN <10 HOURS INSTEAD OF 21.

**WHY DOES A SIMPLE SEARCH
TAKE 2 MINUTES ON A
MULTI-MILLION DOLLAR
ORACLE EXADATA
MACHINE?**

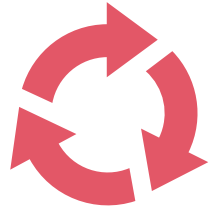
Why are our lists of
“30 things you should do”
not solving our problem?

***BECAUSE YOUR LISTS VIEW
THE WORLD THE WRONG WAY.***

WE BEGAN WITH A TRACE...

	Subroutine	Duration		Calls	Duration per call (seconds)			
		seconds	% R		mean	min	skew	max
1	CPU: FETCH dbcalls ›	97.722000	88.7%	5,673	0.017226	0.000000		0.366000
2	CPU: PARSE dbcalls ›	8.224000	7.5%	3,330	0.002470	0.000000		1.873000
3	SQL*Net message from client	1.869114	1.7%	1,782	0.001049	0.000661		0.031459
4	unaccounted-for within dbcalls	1.043190	0.9%	16,702	0.000062	-0.002407		0.135799
5	11 other subroutines	1.341603	1.2%	16,619	0.000081	0.000000		0.017000
6	Total (15)	110.199907	100.0%					

	Statement	Duration		Rows	LIOs (cr+cu)	PIOs (r)	Distinct plans	Distinct cursors	Distinct texts
		seconds	% R						
1	[frjd8zfy2jfdq] SELECT executions, end_of_fetch_count, elapsed_t... ›	97.676370	88.6%	882	.	.	1	10	1
2	[8hkasvxxk1u5x] select gsmFormulaOutputTypeMML.Name from Materia... ›	4.862325	4.4%	45	3,152	.	2	1	437
3	[5453sdb0bkxh] select gsmMaterialTypeMML.Name from MaterialSpec... ›	4.593040	4.2%	75	3,936	.	1	1	437
4	[2s0d5qpr3qu3s] SELECT Thumbnail, SpecNumber, SpecName, ShortNam... ›	2.089259	1.9%	437	760	.	1	1	1
5	59 other statements	0.978913	0.9%	3,711	12,972	.	48	166	52
6	Total (63)	110.199907	100.0%	5,150	20,820	.	53	179	928



WHAT?

Agile PLM for Process “search for ingredient” feature

HOW LONG?

110s

WHY?

SQL ID frjd8zfy2jfdq takes 97s (Bug 20413540).

WHAT IF?

If install patch, then $110s - 97s = 13s$?



HOW LONG?

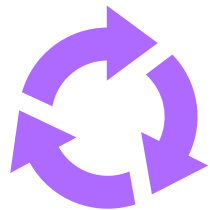
40s

WHY?

Two different optimizer adaptive feature queries burn 30s.

WHAT IF?

If disable optimizer adaptive feature, then 10s?



HOW LONG?

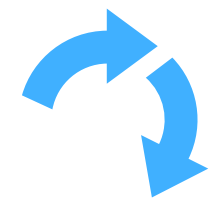
10s

WHY?

1,072 network round trips, 532 parse calls.

WHAT IF?

If combine two SQL statements into one natural join, then .5s?



HOW LONG?

0.5s

WHY?

All remaining work is legitimate; no waste.

WHAT ELSE?

Oracle E-Business “invoicing” feature

...

*HOW USEFUL WERE
THESE LISTS OF
"30 THINGS..."?!*



*TRACE FILES LET YOU VIEW
THE WORLD THE WAY YOU
WANT TO SEE IT:*

1. WHAT?

2. HOW LONG?

3. WHY?

4. WHAT IF?

**WHAT DOES THE SYSTEM
ACTUALLY FEEL LIKE?**

What are people's true
response times?

Are they as good as they
should be?

FIRST, ...A TRACE FILE.

*...ACTUALLY, A WHOLE **BUNCH** OF TRACE FILES,
IN THIS CONNECTION POOLING SYSTEM.*

Response time by user experience (connection pool) (modified)
mrskew --group='\$client_id' --gl=CLIENT-ID ...

CLIENT-ID	DURATION	%	CALLS	MEAN	MIN	MAX
10.17.23.41 AGREEN c9a65ad7-c62a-4de9-bb4a-d118d9a5eda3	3.162362	0.4%	604	0.005236	0.000000	0.135835
10.17.22.14 GLAMB b4748d88-eea4-4d81-a406-e341ea36548f	3.115821	0.4%	604	0.005159	0.000000	0.280440
10.17.22.242 GRUSSELL e7d19f79-4aba-4c32-b75e-7a43248177b3	2.971476	0.3%	604	0.004920	0.000000	0.141190
10.17.22.192 VBARRY edd36692-e78f-46d5-8b4d-803b3f1a91c6	2.627919	0.3%	604	0.004351	0.000000	0.135124
10.17.22.72 IGRIMES 08e41fbe-2303-4575-a135-edf75e262414	2.387713	0.3%	604	0.003953	0.000000	0.163660
10.17.22.233 AWIGGINS 5cadfc51-4f33-479c-9529-6a979914c00e	1.701808	0.2%	604	0.002818	0.000000	0.174480
10.17.24.75 FSIMS cea2fb20-3bde-4ca8-a67c-4061b38a70dc	1.693133	0.2%	604	0.002803	0.000000	0.138768
10.17.22.251 UWYNN b3196c98-906d-4394-bc55-0339518a63b2	1.658180	0.2%	604	0.002745	0.000000	0.165795
10.17.24.45 JSNYDER 33af037c-d63a-4ad0-bac8-6e8d9842a98d	1.608262	0.2%	604	0.002663	0.000000	0.134551
10.17.22.46 BFERRELL 6ba45937-24d3-44be-aad9-81c1a0cff295	1.542965	0.2%	604	0.002555	0.000000	0.125474
22,092 others	859.872753	97.5%	176,196	0.004880	0.000000	0.454674
TOTAL (22,102)	882.342392	100.0%	182,236	0.004842	0.000000	0.454674

THIS APPLICATION ASSIGNS A
UNIQUE CLIENT-ID FOR EACH
USER EXPERIENCE.

THREE CHEERS FOR
WHOEVER DESIGNED
THAT!!!

THESE ARE ACTUAL DURATIONS
PEOPLE FEEL WHEN THEY RUN
STUFF.

HOW 'BOUT A FANCIER REPORT?

Response time by execution id, SQL id, and placeholder value set (modified)
mrskew --group='\$client_id=~/.* .* .*/?sprintf("%-8s %s", (split(/ /,\$client_id))[1..2]):"' --gl=CLIENT-ID ...

CLIENT-ID		DURATION	%	CALLS	MEAN	MIN	MAX
AGREEN	c9a65ad7-c62a-4de9-bb4a-d118d9a5eda3	3.162362					0.135835
GLAMB	b4748d88-eea4-4d81-a406-e341ea36548f	3.115821					0.280449
GRUSSELL	e7d19f79-4aba-4c32-b75e-7a43248177b3	2.971476					
VBARRY	edd36692-e78f-46d5-8b4d-803b3f1a91c6	2.627919					
IGRIMES	08e41fbe-2303-4575-a135-edf75e262414	2.387713					
AWIGGINS	5cacfc51-4f33-479c-9529-6a979914c00e	1.701808	0.2%	604	0.002818		
FSIMS	cea2fb20-3bde-4ca8-a67c-4061b38a70dc	1.693133	0.2%	604	0.002803		0.138768
UWYNN	b3196c98-906d-4394-bc55-0339518a63b2	1.658180	0.2%	604	0.002745	0.000000	0.165795
JSNYDER	33af037c-d63a-4ad0-bac8-6e8d9842a98d	1.608262	0.2%	604	0.002663	0.000000	0.134551
BFERRELL	6ba45937-24d3-44be-aad9-81c1a0cff295	1.542965	0.2%	604	0.002555	0.000000	0.125474
22,092 others		859.872753	97.5%	176,196	0.004880	0.000000	0.454674
TOTAL (22,102)		882.342392	100.0%	182,236	0.004842	0.000000	0.454674

...PRINTED NEATLY WITH EXTRA
CARE IN THE --GROUP EXPRESSION,
SO NOW THE CLIENT-ID
INFORMATION LINES UP.

Fancy!

Response time by execution id, SQL id, and placeholder value set (modified)
 mrskew --group='\$client_id=~/.* .* */?sprintf("%-8s %s", (split(/ /,\$client_id))[1..2]):"' --gl=CLIENT-ID ...

CLIENT-ID		DURATION	%	CALLS	MEAN	MIN	MAX
AGREEN	c9a65ad7-c62a-4de9-bb4a-d118d9a5eda3	3.162362	0.4%	604	0.005236	0.000000	0.135835
GLAMB	b4748d88-eea4-4d81-a406-e341ea36548f	3.115821	0.4%	604	0.005159	0.000000	0.280440
GRUSSELL	e7d19f79-4aba-4c32-b75e-7a43248177b3	2.971476			4920	0.000000	0.141190
VBARRY	edd36692-e78f-46d5-8b4d-803b3f1a91c6	2.627919			4351	0.000000	0.135124
IGRIMES	08e41fbe-2303-4575-a135-edf75e262414	2.387713			3953	0.000000	0.163660
AWIGGINS	5cacfc51-4f33-479c-9529-6a979914c00e	1.701808			2818	0.000000	0.174480
FSIMS	cea2fb20-3bde-4ca8-a67c-4061b38a70dc	1.693133	0.2%	604	0.002803	0.000000	0.138768
UWYNN	b3196c98-906d-4394-bc55-0339518a63b2	1.658180	0.2%	604	0.002745	0.000000	0.165795
JSNYDER	33af037c-d63a-4ad0-bac8-6e8d9842a98d	1.608262	0.2%	604	0.002663	0.000000	0.134551
BFERRELL	6ba45937-24d3-44be-aad9-81c1a0cff295	1.542965	0.2%	604	0.002555	0.000000	0.125474
22,092 others		859.872753	97.5%	176,196	0.004880	0.000000	0.454674
TOTAL (22,102)		882.342392	100.0%	182,236	0.004842	0.000000	0.454674

**WHY DID AGREEN'S
EXPERIENCE TAKE
3.162362 SECONDS?**

*LET'S DRILL INTO JUST AGREEN'S
3.162362-SECOND EXPERIENCE.*

Response time by execution id (modified)

mrskew --where '\$client_id=~/'c9a65ad7-c62a-4de9-bb4a-d118d9a5eda3/' ...

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	3.161922	100.0%	201	0.015731	0.001619	0.135835
SQL*Net message to client	0.000440	0.0%	201	0.000002	0.000000	0.000027
FETCH	0.000000	0.0%	201	0.000000	0.000000	0.000000
EXEC	0.000000	0.0%	1	0.000000	0.000000	0.000000
BINDS						
TOTAL (5)	3.162362	100.0%	604	0.005236	0.000000	0.135835

...ONLY THE SINGLE 3.162362-
SECOND EXPERIENCE.

*WHY IS THERE SO MUCH
"SQL*NET MESSAGE FROM CLIENT"
TIME HERE?*

Response time by execution id (modified)
mrskew --where='\$client_id=~ /c9a65ad7-c62a-4de9-bb4a-d118d9a5eda3/' --rc=p10.rc --name='SQL*Net message from client' ...

	RANGE {min ≤ e < max}		DURATION	%	CALLS	MEAN	MIN	MAX
1.	0.000000	0.000001						
2.	0.000001	0.000010						
3.	0.000010	0.000100						
4.	0.000100	0.001000						
5.	0.001000	0.010000	0.650374	20.6%	175	0.003716	0.001619	0.009987
6.	0.010000	0.100000	0.120466	3.8%	7	0.017209	0.010447	0.035213
7.	0.100000	1.000000	2.391082	75.6%	19	0.125846	0.119146	0.135835
8.	1.000000	10.000000						
9.	10.000000	100.000000						
10.	100.000000	1,000.000000						
11.	1,000.000000	+∞						
TOTAL (11)			3.161922	100.0%	201	0.01573		

JUST THE 3.161992 SECONDS
OF "SQL NET MESSAGE
FROM CLIENT" DURATION.

SOME OF THEM ARE
PRETTY SLOW.

ARE THESE CALLS
ALL NECESSARY?

*WHY ARE THERE SO MANY (201)
NETWORK ROUND TRIPS?*

Response time by execution id (modified)

mrskew --where='\$client_id=~ /c9a65ad7-c62a-4de9-bb4a-d118d9a5eda3/' --select='\$r' --sl=ROWS --name=:dbcall --pre=1 ...

CALL-NAME	ROWS	%	CALLS	MEAN	MIN	MAX
-----	-----	-----	-----	-----	---	-----
FETCH	2,001.0	100.0%	201	10.0	1.0	10.0
EXEC	0.0	0.0%	1	0.0	0.0	0.0
BINDS						
-----	-----	-----	-----	-----	---	-----
TOTAL (3)	2,001.0	100.0%	202	9.9	0.0	10.0

REMIND YOU
OF ANYTHING?

YOU'LL FIND THIS ONE OVER AND OVER
AGAIN, BECAUSE IT'S HARD TO SEE
WITHOUT TRACE DATA.

OPPORTUNITY

TEST ARRAYSIZE=100.

SHOULD RUN 0.3 SECONDS, NOT 3.0 SECONDS.

**THE REAL POINT
OF THIS STORY**

*YOU CAN MEASURE WHAT
PEOPLE ARE **EXPERIENCING!***

AMAZING!

4 WRAP-UP

MEASURING PERFORMANCE
=
MEASURING EXPERIENCES

*TRACE FILES SIMPLIFY YOUR
OPTIMIZATION PROCESS TO
ANSWERING FOUR SIMPLE QUESTIONS:*

1. WHAT?

2. HOW LONG?

3. WHY?

4. WHAT IF?

*TRACE FILES HAVE THINGS IN THEM
YOU CAN'T FIND ANYWHERE ELSE.*

*- WHERE ALL YOUR **TIME** WENT*

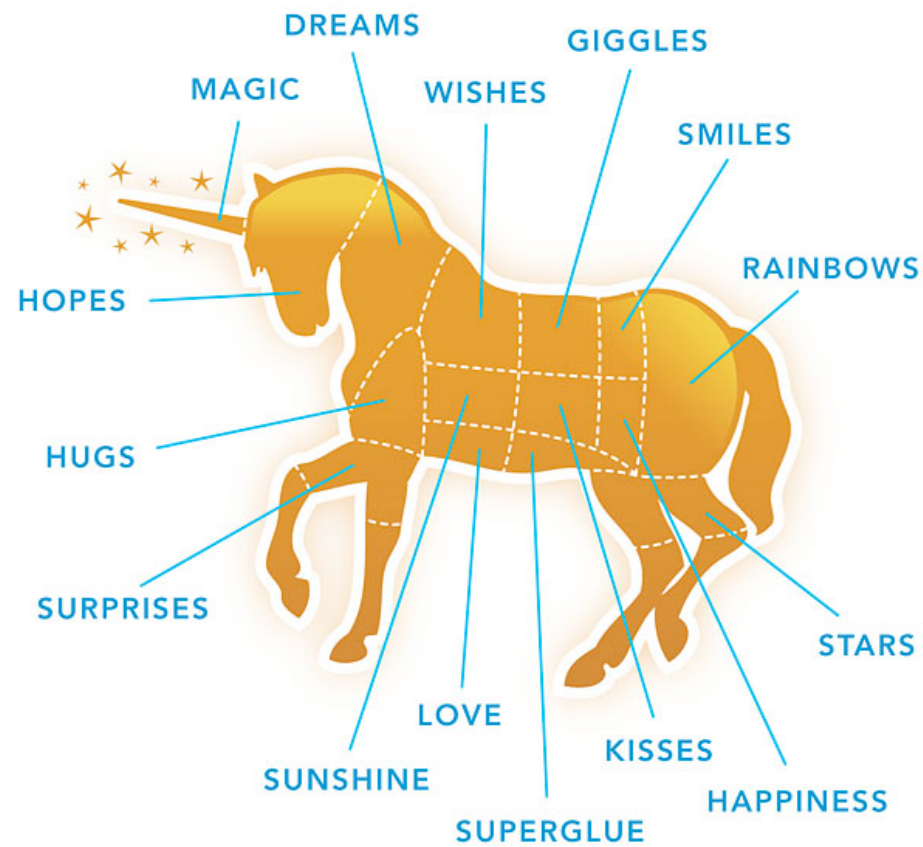
*- 1:1 MAPPING TO PEOPLE'S ACTUAL
EXPERIENCES*

THE KEY TO TRACE FILES:

- GENERATING THE TRACE*
- KNOWING WHAT'S IN THERE*
- KNOWING HOW TO GET IT OUT*

NOW YOU'VE SEEN IT.
NOW YOU UNDERSTAND.

TRACE FILES ARE MAGIC.

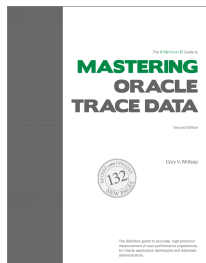


Thank you!

ADVENTURES IN TRACE DATA

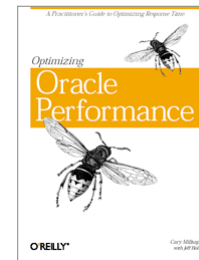
© 2013, 2018 Cary Millsap

REFERENCES



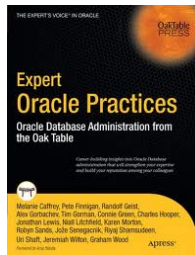
CARY MILLSAP. 2016.
MASTERING ORACLE TRACE DATA.
 METHOD R CORPORATION

DETAILED INFORMATION ABOUT
 ORACLE TRACE DATA AND WHAT
 TO DO WITH IT.



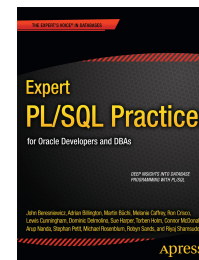
CARY MILLSAP, JEFF HOLT. 2003.
OPTIMIZING ORACLE PERFORMANCE.
 O'REILLY

THE BOOK THAT CHANGED HOW
 PROFESSIONALS THINK ABOUT
 SOFTWARE PERFORMANCE.



ROBYN SANDS, ET AL. 2010.
EXPERT ORACLE PRACTICES.
 APRESS

DETAILED INFORMATION ABOUT
 INSTRUMENTING YOUR ORACLE
 APPLICATION CODE.



RON CRISCO, ET AL. 2011.
EXPERT PL/SQL PRACTICES.
 APRESS

MORE DETAILED INFORMATION ABOUT
 INSTRUMENTING YOUR ORACLE
 APPLICATION CODE.

