



Performance is a Feature

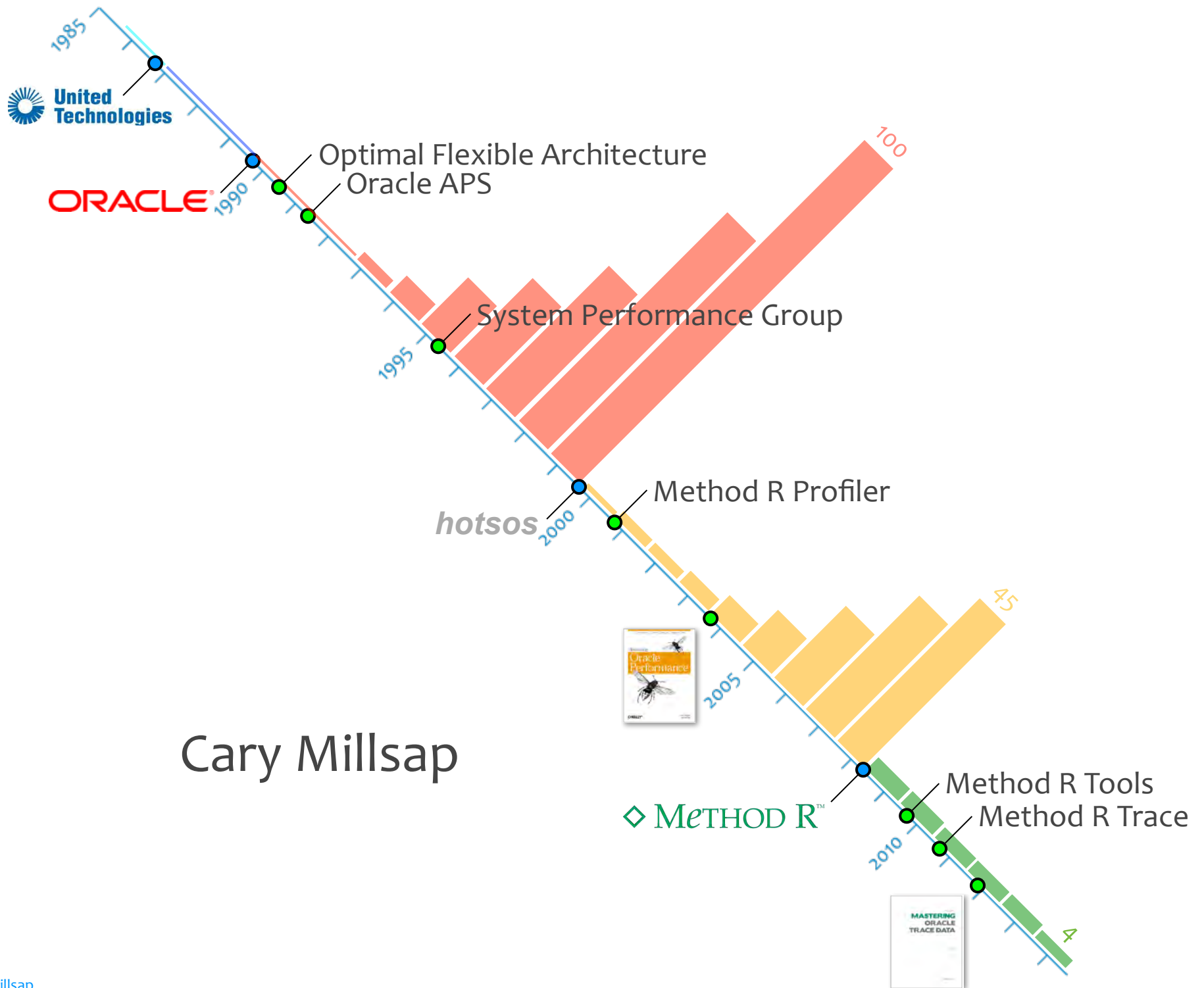
Here is the Specification

Cary Millsap
Method R Corporation

[@CaryMillsap](#) · cary.millsap@method-r.com

DOAG 2013 Konferenz + Ausstellung
Nürnberg NCC Ost, Nürnberg, Deutschland
09.00–09.45 CET Thursday 21 November 2013

© 2012, 2013 Method R Corporation



Cary Millsap

◇ MeTHOD R™

 method-r.com

 [@MethodR](https://twitter.com/MethodR)

create profiling tools

and other *performance software*

teach people

how to make *software run faster*

write applications

(primarily SQL and PL/SQL) for *high-performance projects*

fix problems

with *performance* on any Oracle-based system

1 What is performance?

PERFORMANCE IS NOT AN ATTRIBUTE OF A SYSTEM.

**PERFORMANCE IS AN ATTRIBUTE OF
EACH INDIVIDUAL EXPERIENCE
WITH A SYSTEM.**

What I mean...

Is performance **always** either fast for everyone,
or slow for everyone?

No.

It can be fast for some **clicks**, slow for others.

...fast for some **people**, slow for others.

...fast for some **clicks** for some **people** **sometimes**, slow for others.

**PERFORMANCE IS AN ATTRIBUTE OF
INDIVIDUAL EXPERIENCES.**

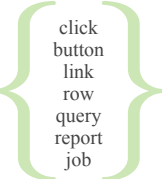
“My {
click
button
link
row
query
report
job} has to finish quickly.”

This is what performance is.

How do you know when your user's {
click
button
link
row
query
report
job
} is fast?

Do you have to feel it?

Experience it first-hand?

How do you know when your user's  is fast?

It would be better if you could look it up.

What have people been *experiencing*?

What have people been *experiencing*?

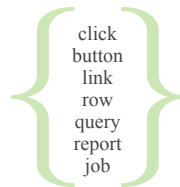
That's a *feature*.

2 How to measure performance?

People feel performance two ways.

“How long does it take?”

“How much stuff can we do?”

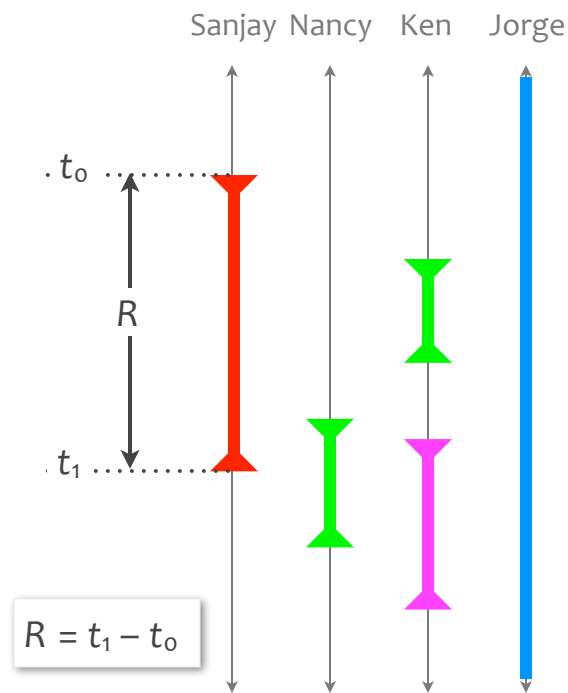


click
button
link
row
query
report
job

“How long does it take?”

Response time (R)

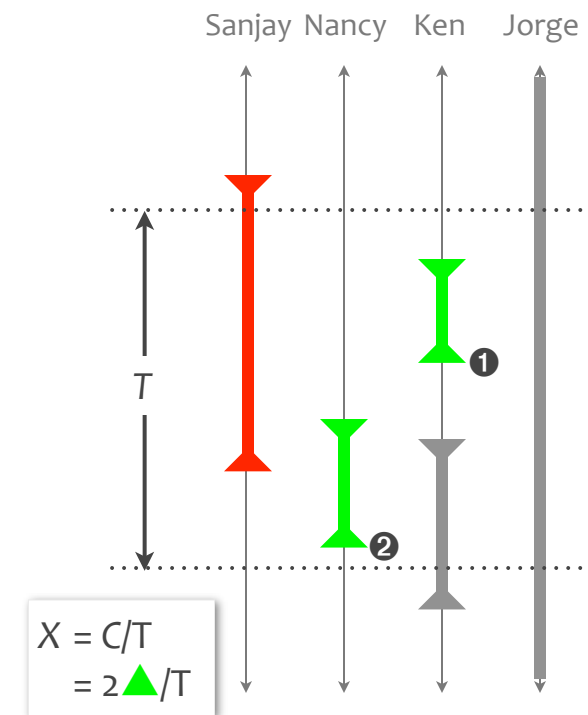
Duration from service request to service fulfillment.



“How much stuff can we do?”

Throughput (X)

Count of service fulfillments in a given time interval.



If you know t_0 and t_1
for each experience,

...you can compute
both R and X .

The key is to

record t_0 and t_1

...and just a few other things.

3 What do you get?

What have people been *experiencing*?

That's a *feature*.

1

How long do people wait for their stuff?

Show response time for any user experience executed within the past 5 years.

1

How long do people wait for their stuff?

Show response time for any user experience executed within the past 5 years.

User name	IP address	Task name	Time requested	R (seconds)
nancy	192.164.1.103	Ship Order	2012-09-04T14:22:08	1.927
ron	192.164.1.138	Book Order	2012-09-04T14:22:12	0.223
ken	148.27.1.42	Pick Order	2012-09-04T14:22:18	0.891
alex	192.164.1.142	Book Order	2012-09-04T14:22:19	0.887
ron	192.164.1.138	Book Order	2012-09-04T14:23:11	0.270
harold	148.113.22.32	Ship Order	2012-09-04T14:23:27	2.084

Where do you get these things?

From code you write like this:

```
● user_name      = sys_context('userenv', 'session_user');  
● ip_address    = sys_context('userenv', 'ip_address');  
● task_name     = "Book Order";  
● experience_id = randomUUID();  
● t0            = gettimeofday();  
  
● /* Your Book Order code path goes here. */  
  
● t1            = gettimeofday();  
● status       = log_task(user_name, ip_address, ..., t0, t1);
```

● Oracle function

● OS function

● Your function

2

How long do tasks take?

Show response time grouped by task.

2

How long do tasks take?

Show response time grouped by task.

Task name	R_{mean}	$R_{99\%}$
Ship Order	2.084	9.597
Pick Order	1.260	4.803
Book Order	0.353	1.626

99% of Book Order experiences are better than **1.626s**.

1% of Book Order experiences are worse than 1.626s.

3

Which response times exceed our tolerance?

Store a performance tolerance for each task and compare each R to its tolerance.

3

Which response times exceed our tolerance?

Store a response time tolerance for each task and compare each R to its tolerance.

R experiences must not exceed 2.0s more often than 1 time in 100.

if and only if

$$R_{(1 - 1/100)} \leq 2.0s$$

$$R_{99\%} \leq 2.0s$$

3

Which response times exceed our tolerance?

Store a response time tolerance for each task and compare each R to its tolerance.

Task name	$T_{99\%}$	$R_{99\%}$	% experiences that exceed $T_{99\%}$ (mustn't exceed 1.0%)
Ship Order	5.000	9.597	8.21%
Pick Order	5.000	4.803	0.94%
Book Order	2.000	1.626	0.70%

Ship Order is not meeting our service level objective.

3

Which response times exceed our tolerance?

Store a response time tolerance for each task and compare each R to its tolerance.

Response Time Expectation Violations by Task



4

Which task will next violate its R tolerance?

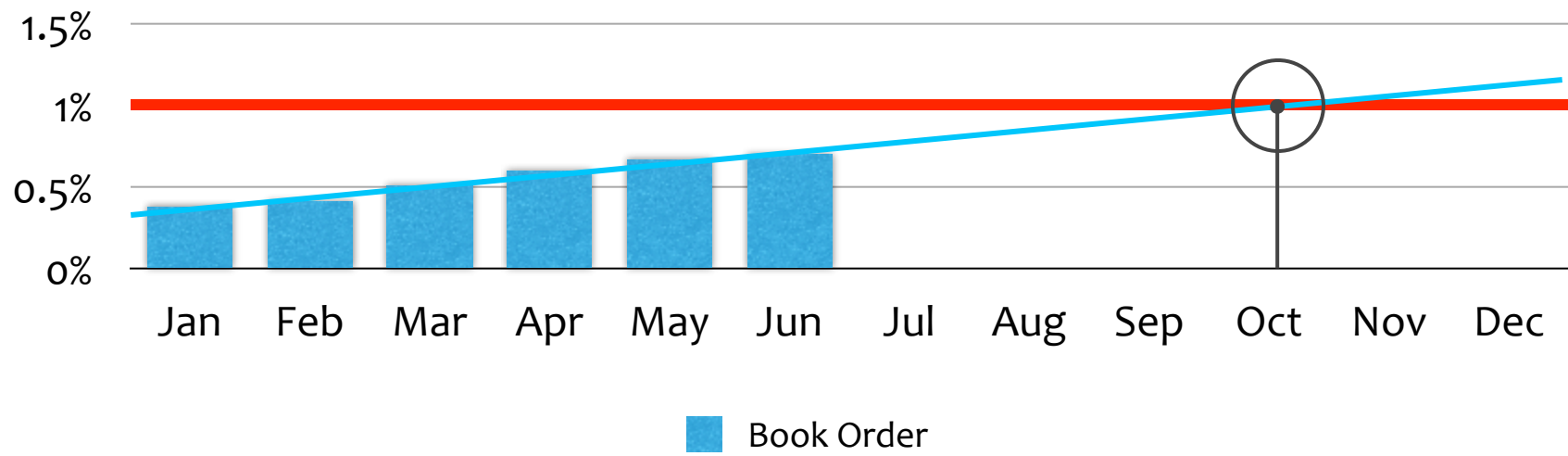
Compute when a daily $R_{99\%}$ for a task is expected to violate its tolerance.

4

Which task will next violate its tolerance?

Compute when a daily $R_{99\%}$ for a task is expected to violate its tolerance.

Response Time Expectation Violations by Task



4

Which task will next violate its tolerance?

Compute when a daily $R_{99\%}$ for a task is expected to violate its tolerance.

Task name	$T_{99\%}$	$R_{99\%}$	% experiences that exceed $T_{99\%}$ (mustn't exceed 1.0%)	Est. days until $R > T$ (i.e., until % > 1.0)
Ship Order	5.000	9.597	8.21%	—
Pick Order	5.000	4.803	0.94%	18
Book Order	2.000	1.626	0.70%	117

“Pick Order is on an 18-day collision course.”

5

Does Nancy's stuff really take longer?

Show response time grouped by task and user.

5

Does Nancy's stuff really take longer?

Show response time grouped by task and user.

Task name	User name	$T_{99\%}$	$R_{99\%}$	% experiences that exceed $T_{99\%}$ (mustn't exceed 1.0%)
Ship Order	nancy	5.000	24.168	18.21%
Ship Order	ron	5.000	4.966	0.94%
Ship Order	ken	5.000	4.208	0.70%
Ship Order	harold	5.000	3.174	0.67%

“Nancy's *Ship Orders* really do take longer.”

6

Is it just Nancy? Or a whole subnetwork?

Show response time grouped by task and subnetwork.

6

Is it just Nancy? Or a whole subnetwork?

Show response time grouped by task and subnetwork.

Task name	IP subnetwork	$T_{99\%}$	$R_{99\%}$	% experiences that exceed $T_{99\%}$ (mustn't exceed 1.0%)
Ship Order	192.164.1	5.000	18.211	12.89%
Ship Order	148.27.1	5.000	4.669	0.90%
Ship Order	148.13.22	5.000	4.812	0.72%

“Everybody in Atlanta is suffering.”

What have people been *experiencing*?

That's a *feature*.

The important thing is what you measure.

Performance is about measuring experiences.

...Not resources.

4 Bonus: the *Traceable Experience* feature

7

Why did Nancy's task take so long?

Show response time call-by-call, grouped by end-user experience.

How to distinguish
end-user experiences
from each other?

In Oracle,
module, action, clientid.

module = application name

action = task name

clientid = UUID + user + IP

In PL/SQL, use `dbms_application_info.set_module`
`dbms_session.set_identifier`

In Java, use `setEndToEndMetrics()`

What do you get?

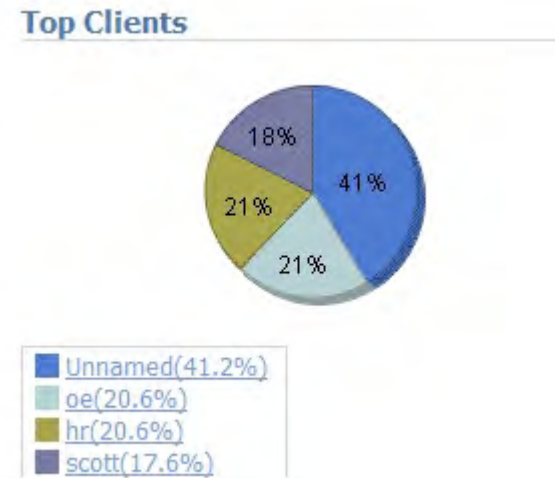
What do you get?

You get hooks in your v\$ data to group by.

```
SQL> desc v$session
...
module
...
action
...
client_identifier
...
```

What do you get?

OEM (...ASH, etc.) can help you more.



What do you get?

You can control tracing.

```
dbms_monitor.serv_mod_act_trace_enable(:serv, :mod, :act, ...)
```

```
dbms_monitor.client_id_trace_enable(:client_id)
```

What do you get?

You get hooks in your trace data to group by.

```
*** CLIENT ID:(798164fb-ad9a-43e1-b543-0e1cfa94195c PDRAKE 172.27.13.1
*** SERVICE NAME:(SYS$USERS) 2012-05-09 11:42:55.278
*** MODULE NAME:(oe) 2012-05-09 11:42:55.278
*** ACTION NAME:(book) 2012-05-09 11:42:55.278
```

What can your trace files tell you?

What have people been *experiencing*?

EXPERIENCE-ID	USER	IP-ADDRESS	MOD/ACT	DURATION	
eec4c72f-b685-4b5b-8447-688b8aecbc6f	NPHELPS	10.17.22.12	oe/book	49.699960	6
ab9fdc58-6c09-4fe2-b253-8b2c191e4671	VSAUNDERS	10.17.22.76	oe/book	48.977182	6
ab8e9d2c-8ba6-4c5f-8673-13a3c958af7a	PDRAKE	10.17.23.174	oe/book	47.520700	6
ced4bdee-f0d8-44cd-a878-f0b938802bc0	VMICHAEL	10.17.24.138	oe/book	43.597261	5
5c01b8e5-e0d3-44c2-ada8-c8b1ed17abee	MSTANLEY	10.17.22.115	pa/mtch	23.714088	3
2161468f-5504-419a-8226-f5ee0ad39b73	TIRWIN	10.17.22.249	pa/reco	22.946626	3
e463a90b-32a8-49fc-81e8-0c2559269e1e	ESANDERS	10.17.21.167	pa/mtch	22.607869	3
8a41c465-e00f-43b2-afcd-ba4edd7df899	DHALEY	10.17.22.98	oe/pick	21.967559	3
052b3d18-e992-46a2-8ae3-87427c78598e	UDECKER	10.17.21.239	oe/ship	20.312153	2
2270f4b4-1c46-4ad7-ac6a-2ed1fc13c283	LDILLARD	10.17.21.14	oe/ship	20.024742	2
158 others				420.181628	56
TOTAL (168)				741.549768	100

Why did Nancy's program take so long?

CALL-NAME	DURATION	%	CALLS	MEAN	MIN	MAX
SQL*Net message from client	40.507440	81.5%	18,585	0.002180	0.000000	0.220486
EXEC	6.684420	13.4%	18,585	0.000360	0.000000	0.128008
FETCH	2.388145	4.8%	18,583	0.000129	0.000000	0.120008
SQL*Net message to client	0.094487	0.2%	18,585	0.000005	0.000000	0.018525
cursor: pin S wait on X	0.025463	0.1%	4	0.006366	0.000000	0.011166
cursor: pin S	0.000005	0.0%	5	0.000001	0.000000	0.000005
PARSE	0.000000	0.0%	2	0.000000	0.000000	0.000000
pooled connection free	0.000000	0.0%	1	0.000000	0.000000	0.000000
XCTEND	0.000000	0.0%	1	0.000000	0.000000	0.000000
TOTAL (9)	49.699960	100.0%	74,351	0.000668	0.000000	0.220486

Why 40 sec of 'SQL*Net message from client' calls?

RANGE {min ≤ e < max}	DURATION	%	CALLS	MEAN	MIN	M
1. 0.000000 0.000001	0.000000	0.0%	3	0.000000	0.000000	0.0000
2. 0.000001 0.000010						
3. 0.000010 0.000100	0.000358	0.0%	5	0.000072	0.000041	0.0000
4. 0.000100 0.001000	8.391000	20.7%	13,947	0.000602	0.000104	0.0009
5. 0.001000 0.010000	9.307749	23.0%	3,822	0.002435	0.001000	0.0099
6. 0.010000 0.100000	20.877546	51.5%	793	0.026327	0.010001	0.0994
7. 0.100000 1.000000	1.930787	4.8%	15	0.128719	0.103157	0.2204
8. 1.000000 10.000000						
9. 10.000000 100.000000						
10. 100.000000 1,000.000000						
11. 1,000.000000 +∞						
TOTAL (11)	40.507440	100.0%	18,585	0.002180	0.000000	0.2204

Were the calls all made by the same SQL?

SQLID	DURATION	%	CALLS	MEAN	MIN	MAX
-----	-----	-----	-----	-----	-----	-----
6m3a3v3yjbvabh	40.506884	100.0%	18,584	0.002180	0.000000	0.220486
40kwnn4q98wqn	0.000556	0.0%	1	0.000556	0.000556	0.000556
-----	-----	-----	-----	-----	-----	-----
TOTAL (2)	40.507440	100.0%	18,585	0.002180	0.000000	0.220486

How many rows per database call?

CALL-NAME	ROWS	%	CALLS	MEAN	MIN	MAX
-----	-----	-----	-----	-----	-----	-----
FETCH	18,378	100.0%	18,583	1	0	1
EXEC	2	0.0%	18,585	0	0	1
PARSE	0	0.0%	2	0	0	0
XCTEND	0	0.0%	1	0	0	0
-----	-----	-----	-----	-----	-----	-----
TOTAL (4)	18,380	100.0%	37,171	0	0	1

ARRAY FETCH SIZE PROBLEM

5 Conclusion

Performance is about
measuring *experiences*.

You can **write apps** that make it easy
to know what performance
people have been ***experiencing***.

And you can retrofit **existing** apps.

EXPERIENCE

- CLIENT-ID
 - EXPERIENCE-ID
 - USERNAME
 - IP-ADDRESS
- T₀
- T₁
- ...

N
1

TASK

- MODULE
- ACTION
- ...

1
N

SQL

- SQL-ID
- ...

Performance is a *feature*.

Everything you've seen today, we've already built.

 cary.millsap@method-r.com

 [@CaryMillsap](https://twitter.com/CaryMillsap)

 **MeTHOD R™**

 method-r.com

 [@MethodR](https://twitter.com/MethodR)

create profiling tools

and other performance software

teach people

how to make software run faster

write applications

(primarily SQL and PL/SQL) for high-performance projects

fix problems

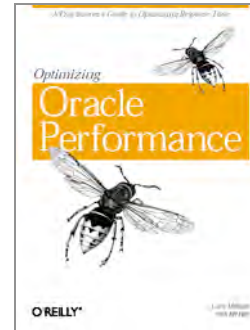
with performance on any Oracle-based system

6 References



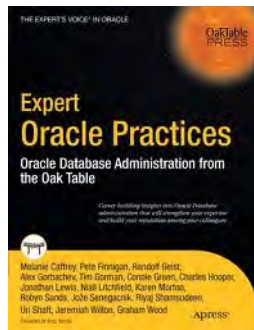
Ron Crisco, et al. 2011
[Expert PL/SQL Practices](#)
Apress

Detailed information about instrumenting your Oracle application code.



Cary Millsap, Jeff Holt. 2003
[Optimizing Oracle Performance](#)
O'Reilly

Detailed information about Oracle trace data and what to do with it.



Robyn Sands, et al. 2010
[Expert Oracle Practices](#)
Apress

Detailed information about instrumenting your Oracle application code.



Cary Millsap. 2011
[Mastering Oracle Trace Data](#)
Method R Corporation

Follow-up to *Optimizing Oracle Performance* teaches you how to master Oracle trace data.