Performance is a feature

How many orders did you book between 10am and 11am on November 4?

What percentage of “oe book” clicks have run in less than 1 second since August 17?

Morton's 1-line “oe book” ran in 1 second. Nancy's ran in 10 seconds. Why?

Is there a user or location that experiences particularly good or bad response times?

Which tasks have the most execution response time variance?

What is your best hour of “oe book” throughput since January 1? Your worst hour?

What tasks competed against “oe book” during its worst-throughput hour?

99% of “oe book” clicks must run in less than 1 second, and 99.99% must run in less than 5 seconds. Do they?

When will your 99th percentile “oe book” response time exceed your 1-sec tolerance?
You can **know.**

But it takes **work.**
Performance is a **feature**.

You have to...
- Design it.
- Code it.
- Test it.
- *Buy* hardware for it.
- Manage it.

If you have a **prepackaged** application,

You have to...
- **Trust** that your vendor did it right.
- Or **convince** your vendor to do it right.
- Or **retrofit** your vendor’s software.
- *Buy* hardware for it.
- Manage it.
Measuring performance

Your system CPU utilization is 96%.

Is that good performance? or bad performance?
Maybe?
System busy doing work.
Response times are fine.

Maybe not?
Lots of programs using PX.
Response times are horrible.

Your system CPU utilization is 96%.
THE FUNDAMENTAL THEOREM OF MEASURING THINGS

If obviously different experiences yield the same measurement, then you're measuring the wrong thing.
Performance Complaint Form X

I get only ________ clicks per ________ task name ________ task name ________ reports ________ reports per ________ number ________ number ________ hour ________ hour ________ minute, ________ minute, ________ second ________ second

☐ That’s not enough.
☐ Can we do better?

Performance

relationship of task executions to time

response time

time elapsed to execute a task

throughput
	task executions per unit of time
The key to measuring performance:

**measure task executions**

The machine can't know what your tasks are unless you tell it.

What is a **task**?

- a click
- a report
- a batch job
- part of a batch job
- ...
performance

relationship of task executions to time

time elapsed to execute a task

response time

throughput
task executions per unit of time

20
How many orders did you book between 10am and 11am on November 4?

What percentage of “oe book” clicks have run in less than 1 second since August 17?

Morton’s 1-line “oe book” ran in 1 second. Nancy’s ran in 10 seconds. Why?

Is there a user or location that experiences particularly good or bad response times?

Which tasks have the most execution response time variance?

What is your best hour of “oe book” throughput since January 1? Your worst hour?

What tasks competed against “oe book” during its worst-throughput hour?

99% of “oe book” clicks must run in less than 1 second, and 99.99% must run in less than 5 seconds. Do they?

When will your 99th percentile “oe book” response time exceed your 1-sec tolerance?

...
CREATE OR REPLACE PROCEDURE gl_posting AS
  l_sqlcode NUMBER;
  l_sqlerrm VARCHAR2(512);
  entries_posted NUMBER;
BEGIN
  ilo_task.begin_task(module => 'GL', action => 'Posting');
  -- Code to execute the business task goes here.
  ilo_task.end_task(widget_count => entries_posted);
EXCEPTION
  WHEN OTHERS THEN
    l_sqlcode := SQLCODE;
    l_sqlerrm := SQLERRM;
    ilo_task.end_task(error_num => l_sqlcode);
    -- Handle exceptions here.
    raise;
END;
/* Allocate a connection from the pool. */
Connection conn = this.datasource.getConnection();

/* Set task identification attributes: who is executing what? */
Socket socket = appconn.getSocket();
String ip = socket.getInetAddress().toString();
String uname = appconn.getUserName();
String uuid = UUID.randomUUID().toString();
String cid = String.format("%s %s %s", ip, uname, uuid);
this.metrics[OracleConnection.END_TO_END_CLIENTID_INDEX] = cid;
this.metrics[OracleConnection.END_TO_END_ACTION_INDEX] = appconn.getAct();
this.metrics[OracleConnection.END_TO_END_MODULE_INDEX] = appconn.getMod();
((OracleConnection)conn).setEndToEndMetrics(this.metrics, (short) 0);

/* Enable Oracle extended SQL tracing. */
if (appcon.getTracingIntention)
  Statement stmt = conn.prepareStatement("begin method_r_trace.enable; end;\n");
  stmt.execute();
}

/* Code to execute the business task goes here. */

/* Disable Oracle extended SQL trace. */
if (appcon.getTracingIntention)
  stmt = conn.prepareStatement("begin method_r_trace.disable; end;\n");
  stmt.execute();

/* Give the connection back to the pool. */
conn.close();
Tracing gives you magic.

Trace data: dbcalls, syscalls

begin method_r_trace.enable; end;

select * from oe_pick where id = :1

begin method_r_trace.disable; end;

## Group duration by module/action...

<table>
<thead>
<tr>
<th>MOD/ACT</th>
<th>DURATION</th>
<th>%</th>
<th>CALLS</th>
<th>MEAN</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>oe / book</td>
<td>232.433195</td>
<td>31.3%</td>
<td>324,109</td>
<td>0.000717</td>
<td>0.000000</td>
<td>0.439000</td>
</tr>
<tr>
<td>pa / etch</td>
<td>159.85674</td>
<td>21.6%</td>
<td>190,551</td>
<td>0.000839</td>
<td>0.000000</td>
<td>0.293000</td>
</tr>
<tr>
<td>oe / ship</td>
<td>113.72608</td>
<td>15.3%</td>
<td>129,670</td>
<td>0.000877</td>
<td>0.000000</td>
<td>0.394000</td>
</tr>
<tr>
<td>pa / corr</td>
<td>94.278553</td>
<td>12.7%</td>
<td>130,234</td>
<td>0.000724</td>
<td>0.000000</td>
<td>0.301000</td>
</tr>
<tr>
<td>oe / pick</td>
<td>67.108856</td>
<td>9.0%</td>
<td>72,625</td>
<td>0.000924</td>
<td>0.000000</td>
<td>0.517000</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------</td>
<td>-------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>TOTAL (6)</td>
<td>741.549768</td>
<td>100.0%</td>
<td>930,195</td>
<td>0.000797</td>
<td>0.000000</td>
<td>0.517000</td>
</tr>
</tbody>
</table>

## Group task execution duration by client_id...

<table>
<thead>
<tr>
<th>CLIENT_ID</th>
<th>DURATION</th>
<th>%</th>
<th>CALLS</th>
<th>MEAN</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.17.12 KPHelps eec4c72f-b685-4b5b-8447-688b8aecb6df</td>
<td>49.699960</td>
<td>6.7%</td>
<td>74,351</td>
<td>0.000668</td>
<td>0.000000</td>
<td>0.220486</td>
</tr>
<tr>
<td>10.17.22.76 VSANDERS a69fdcc6-6c00-4fe2-b531-b2c1951d4671</td>
<td>48.977182</td>
<td>6.6%</td>
<td>108,760</td>
<td>0.000450</td>
<td>0.000000</td>
<td>0.154742</td>
</tr>
<tr>
<td>10.17.23.174 PRBAKE ab49d20c-6b61-4c61-be51-13a3c955a77c</td>
<td>47.520700</td>
<td>6.4%</td>
<td>49,059</td>
<td>0.000969</td>
<td>0.000000</td>
<td>0.439000</td>
</tr>
<tr>
<td>10.17.24.138 VICHALE cadd46de-bf8d-44dc-ad78-f0b3938820e0</td>
<td>43.59261</td>
<td>5.9%</td>
<td>43,983</td>
<td>0.000991</td>
<td>0.000000</td>
<td>0.189084</td>
</tr>
<tr>
<td>10.17.22.115 WSTANLEY 5c01b8e5-e0d3-44c2-ada8-c8b1ed17abee</td>
<td>23.714888</td>
<td>3.2%</td>
<td>21,701</td>
<td>0.001093</td>
<td>0.000000</td>
<td>0.151083</td>
</tr>
<tr>
<td>10.17.22.249 TIRWIN 2161468f-5504-419a-8226-f5eed0d30f63</td>
<td>22.946626</td>
<td>3.1%</td>
<td>23,882</td>
<td>0.000961</td>
<td>0.000000</td>
<td>0.132008</td>
</tr>
<tr>
<td>10.17.22.98 DHALEY 8a41c465-e00f-43b2-afcd-ba4edd740f99</td>
<td>21.967559</td>
<td>3.0%</td>
<td>22,522</td>
<td>0.000975</td>
<td>0.000000</td>
<td>0.159246</td>
</tr>
<tr>
<td>10.17.21.239 UDECKER 052b3d18-e992-46a2-8ae3-87427c7859e</td>
<td>20.312153</td>
<td>2.7%</td>
<td>19,922</td>
<td>0.001020</td>
<td>0.000000</td>
<td>0.138701</td>
</tr>
<tr>
<td>10.17.21.167 ESANDERS e463a90b-32cd-49fc-b1e8-8c599269e1e</td>
<td>22.607869</td>
<td>3.0%</td>
<td>24,175</td>
<td>0.000935</td>
<td>0.000000</td>
<td>0.128223</td>
</tr>
<tr>
<td>10.17.22.98 CHALEY 84c1c465-ee0d-40b2-afcd-bacadd747f5f99</td>
<td>21.967559</td>
<td>3.0%</td>
<td>22,522</td>
<td>0.000975</td>
<td>0.000000</td>
<td>0.159246</td>
</tr>
<tr>
<td>10.17.21.239 UDECKER 052b3d18-e992-46a2-afcd-beaedd740f99</td>
<td>20.312153</td>
<td>2.7%</td>
<td>19,922</td>
<td>0.001020</td>
<td>0.000000</td>
<td>0.138701</td>
</tr>
<tr>
<td>158 others</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL (168)</td>
<td>741.549768</td>
<td>100.0%</td>
<td>930,195</td>
<td>0.000797</td>
<td>0.000000</td>
<td>0.517000</td>
</tr>
</tbody>
</table>
Group task execution duration by call-name for a specific task execution (profile):

<table>
<thead>
<tr>
<th>CALL-NAME</th>
<th>DURATION</th>
<th>%</th>
<th>CALLS</th>
<th>MEAN</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL*Net message from client</td>
<td>40.507440</td>
<td>81.5%</td>
<td>18,585</td>
<td>0.002180</td>
<td>0.00000</td>
<td>0.220486</td>
</tr>
<tr>
<td>EXEC</td>
<td>6.684420</td>
<td>13.4%</td>
<td>18,585</td>
<td>0.000360</td>
<td>0.00000</td>
<td>0.128008</td>
</tr>
<tr>
<td>FETCH</td>
<td>2.384145</td>
<td>4.8%</td>
<td>18,583</td>
<td>0.000129</td>
<td>0.00000</td>
<td>0.120008</td>
</tr>
<tr>
<td>SQL*Net message to client</td>
<td>0.094487</td>
<td>0.2%</td>
<td>18,585</td>
<td>0.000005</td>
<td>0.00000</td>
<td>0.018525</td>
</tr>
<tr>
<td>cursor: pin S wait on X</td>
<td>0.025463</td>
<td>0.1%</td>
<td>4</td>
<td>0.000001</td>
<td>0.00000</td>
<td>0.011166</td>
</tr>
<tr>
<td>cursor: pin S</td>
<td>0.000005</td>
<td>0.0%</td>
<td>5</td>
<td>0.000001</td>
<td>0.00000</td>
<td>0.000005</td>
</tr>
<tr>
<td>PARSE</td>
<td>0.000000</td>
<td>0.0%</td>
<td>1</td>
<td>0.000000</td>
<td>0.00000</td>
<td>0.000000</td>
</tr>
<tr>
<td>pooled connection free</td>
<td>0.000000</td>
<td>0.0%</td>
<td>1</td>
<td>0.000000</td>
<td>0.00000</td>
<td>0.000000</td>
</tr>
<tr>
<td>XCTEND</td>
<td>0.000000</td>
<td>0.0%</td>
<td>1</td>
<td>0.000000</td>
<td>0.00000</td>
<td>0.000000</td>
</tr>
<tr>
<td>TOTAL (9)</td>
<td>49.699960</td>
<td>100.0%</td>
<td>74,351</td>
<td>0.000668</td>
<td>0.00000</td>
<td>0.220486</td>
</tr>
</tbody>
</table>

Group call duration by duration for a specific call type for a specific task execution:

<table>
<thead>
<tr>
<th>RANGE (min ≤ e &lt; max)</th>
<th>DURATION</th>
<th>%</th>
<th>CALLS</th>
<th>MEAN</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000001</td>
<td>0.0%</td>
<td>3</td>
<td>0.000000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000010</td>
<td>0.0%</td>
<td>5</td>
<td>0.000007</td>
<td>0.000041</td>
<td>0.000052</td>
</tr>
<tr>
<td>0.000100</td>
<td>0.000050</td>
<td>0.0%</td>
<td>5</td>
<td>0.000008</td>
<td>0.000104</td>
<td>0.000099</td>
</tr>
<tr>
<td>0.000100</td>
<td>0.000000</td>
<td>0.0%</td>
<td>1</td>
<td>0.000001</td>
<td>0.000104</td>
<td>0.000099</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.010000</td>
<td>23.0%</td>
<td>13,947</td>
<td>0.000602</td>
<td>0.000104</td>
<td>0.000999</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.0%</td>
<td>15</td>
<td>0.128719</td>
<td>0.103157</td>
<td>0.220486</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.000000</td>
<td>0.0%</td>
<td>1</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.000000</td>
<td>1.000000</td>
<td>100.0%</td>
<td>18,585</td>
<td>0.002180</td>
<td>0.000000</td>
<td>0.220486</td>
</tr>
</tbody>
</table>

TOTAL (5) 40.507440 100.0% 18,585 0.002180 0.000000 0.220486
Group call duration by call execution id for a specific call type for a specific execution...

<table>
<thead>
<tr>
<th>LINE#</th>
<th>DURATION</th>
<th>%</th>
<th>CALLS</th>
<th>MEAN</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>146783</td>
<td>0.220486</td>
<td>0.5</td>
<td>1</td>
<td>0.220486</td>
<td>0.220486</td>
<td>0.220486</td>
</tr>
<tr>
<td>83778</td>
<td>0.181826</td>
<td>0.4</td>
<td>1</td>
<td>0.181826</td>
<td>0.181826</td>
<td>0.181826</td>
</tr>
<tr>
<td>121473</td>
<td>0.164216</td>
<td>0.4</td>
<td>1</td>
<td>0.164216</td>
<td>0.164216</td>
<td>0.164216</td>
</tr>
<tr>
<td>182868</td>
<td>0.128376</td>
<td>0.3</td>
<td>1</td>
<td>0.128376</td>
<td>0.128376</td>
<td>0.128376</td>
</tr>
<tr>
<td>120701</td>
<td>0.126663</td>
<td>0.3</td>
<td>1</td>
<td>0.126663</td>
<td>0.126663</td>
<td>0.126663</td>
</tr>
<tr>
<td>246050</td>
<td>0.109707</td>
<td>0.3</td>
<td>1</td>
<td>0.109707</td>
<td>0.109707</td>
<td>0.109707</td>
</tr>
<tr>
<td>74221</td>
<td>0.109724</td>
<td>0.3</td>
<td>1</td>
<td>0.109724</td>
<td>0.109724</td>
<td>0.109724</td>
</tr>
<tr>
<td>120661</td>
<td>0.109231</td>
<td>0.3</td>
<td>1</td>
<td>0.109231</td>
<td>0.109231</td>
<td>0.109231</td>
</tr>
<tr>
<td>181952</td>
<td>0.109787</td>
<td>0.3</td>
<td>1</td>
<td>0.109787</td>
<td>0.109787</td>
<td>0.109787</td>
</tr>
<tr>
<td>192950</td>
<td>0.109635</td>
<td>0.3</td>
<td>1</td>
<td>0.109635</td>
<td>0.109635</td>
<td>0.109635</td>
</tr>
<tr>
<td>18,575 others</td>
<td>39.106820</td>
<td>96.5</td>
<td>18,575</td>
<td>0.002105</td>
<td>0.000000</td>
<td>0.108649</td>
</tr>
<tr>
<td>TOTAL (18,585)</td>
<td>40.507440</td>
<td>100.0</td>
<td>18,585</td>
<td>0.002180</td>
<td>0.000000</td>
<td>0.220486</td>
</tr>
</tbody>
</table>

Group call duration by call execution id and sql id for a specific call type for a specific execution...

<table>
<thead>
<tr>
<th>SQLID</th>
<th>DURATION</th>
<th>%</th>
<th>CALLS</th>
<th>MEAN</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>146783: 6e63c6yjvubh</td>
<td>0.220486</td>
<td>0.5</td>
<td>1</td>
<td>0.220486</td>
<td>0.220486</td>
<td>0.220486</td>
</tr>
<tr>
<td>83778: 6e63c6yjvubh</td>
<td>0.181826</td>
<td>0.4</td>
<td>1</td>
<td>0.181826</td>
<td>0.181826</td>
<td>0.181826</td>
</tr>
<tr>
<td>121473: 6e63c6yjvubh</td>
<td>0.164216</td>
<td>0.4</td>
<td>1</td>
<td>0.164216</td>
<td>0.164216</td>
<td>0.164216</td>
</tr>
<tr>
<td>182868: 6e63c6yjvubh</td>
<td>0.128376</td>
<td>0.3</td>
<td>1</td>
<td>0.128376</td>
<td>0.128376</td>
<td>0.128376</td>
</tr>
<tr>
<td>120701: 6e63c6yjvubh</td>
<td>0.126663</td>
<td>0.3</td>
<td>1</td>
<td>0.126663</td>
<td>0.126663</td>
<td>0.126663</td>
</tr>
<tr>
<td>246050: 6e63c6yjvubh</td>
<td>0.109707</td>
<td>0.3</td>
<td>1</td>
<td>0.109707</td>
<td>0.109707</td>
<td>0.109707</td>
</tr>
<tr>
<td>74221: 6e63c6yjvubh</td>
<td>0.109724</td>
<td>0.3</td>
<td>1</td>
<td>0.109724</td>
<td>0.109724</td>
<td>0.109724</td>
</tr>
<tr>
<td>120661: 6e63c6yjvubh</td>
<td>0.109231</td>
<td>0.3</td>
<td>1</td>
<td>0.109231</td>
<td>0.109231</td>
<td>0.109231</td>
</tr>
<tr>
<td>181952: 6e63c6yjvubh</td>
<td>0.109787</td>
<td>0.3</td>
<td>1</td>
<td>0.109787</td>
<td>0.109787</td>
<td>0.109787</td>
</tr>
<tr>
<td>192950: 6e63c6yjvubh</td>
<td>0.109635</td>
<td>0.3</td>
<td>1</td>
<td>0.109635</td>
<td>0.109635</td>
<td>0.109635</td>
</tr>
<tr>
<td>18,575 others</td>
<td>39.106820</td>
<td>96.5</td>
<td>18,575</td>
<td>0.002105</td>
<td>0.000000</td>
<td>0.108649</td>
</tr>
<tr>
<td>TOTAL (18,585)</td>
<td>40.507440</td>
<td>100.0</td>
<td>18,585</td>
<td>0.002180</td>
<td>0.000000</td>
<td>0.220486</td>
</tr>
</tbody>
</table>
Diff profile (group duration by call type for specific execution) across task executions...

Why is Morton’s execution faster than Nancy’s?

<table>
<thead>
<tr>
<th>CALL-_NAME</th>
<th>DURATION</th>
<th>%</th>
<th>CALLS</th>
<th>MEAN</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL (8)</td>
<td>48.977182</td>
<td>100.0%</td>
<td>108,760</td>
<td>0.000450</td>
<td>0.000000</td>
<td>0.154742</td>
</tr>
<tr>
<td>PARSE</td>
<td>0.000000</td>
<td>0.0%</td>
<td>2</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>cursor: pin S</td>
<td>0.002366</td>
<td>0.0%</td>
<td>2</td>
<td>0.001183</td>
<td>0.000000</td>
<td>0.002366</td>
</tr>
<tr>
<td>SQL*Net message to client</td>
<td>0.153011</td>
<td>0.3%</td>
<td>27,189</td>
<td>0.000006</td>
<td>0.000000</td>
<td>0.014986</td>
</tr>
<tr>
<td>FETCH</td>
<td>2.128134</td>
<td>4.3%</td>
<td>27,187</td>
<td>0.000078</td>
<td>0.000000</td>
<td>0.056003</td>
</tr>
<tr>
<td>EXEC</td>
<td>5.444346</td>
<td>11.1%</td>
<td>27,189</td>
<td>0.000200</td>
<td>0.000000</td>
<td>0.124007</td>
</tr>
<tr>
<td>SQL*Net message from client</td>
<td>41.249325</td>
<td>84.2%</td>
<td>27,189</td>
<td>0.001517</td>
<td>0.000000</td>
<td>0.154742</td>
</tr>
</tbody>
</table>

---------------------------

CALL-NAME DURATION % CALLS MEAN MIN MAX

TOTAL (7) 20.024742 100.0% 19,818 0.001010 0.000000 0.138701

---------------------------

XCTEND 0.000000 0.0% 1 0.000000 0.000000 0.000000
pooled connection free 0.000000 0.0% 1 0.000000 0.000000 0.000000
PARSE 0.000000 0.0% 2 0.000000 0.000000 0.000000
SQL*Net message to client 0.042101 0.2% 4,954 0.000008 0.000000 0.020164
FETCH 0.952060 4.8% 4,952 0.000192 0.000000 0.128008
EXEC 2.680166 13.4% 4,954 0.000541 0.000000 0.128008
SQL*Net message from client 16.350415 81.7% 4,954 0.003300 0.000000 0.138701

---------------------------

CALL-NAME DURATION % CALLS MEAN MIN MAX

TOTAL (9) 49.699960 100.0% 74,351 0.000668 0.000000 0.220486

---------------------------

pooled connection free 0.000000 0.0% 1 0.000000 0.000000 0.000000
PARSE 0.000000 0.0% 2 0.000000 0.000000 0.000000
cursor: pin S 0.000005 0.0% 5 0.000001 0.000000 0.000005
cursor: pin S wait on X 0.025463 0.1% 4 0.006366 0.000000 0.011166
SQL*Net message to client 0.094487 0.2% 18,585 0.000005 0.000000 0.018525
FETCH 2.388145 4.8% 18,583 0.000129 0.000000 0.120008
EXEC 6.684420 13.4% 18,585 0.000360 0.000000 0.128008
SQL*Net message from client 40.507440 81.5% 18,585 0.002180 0.000000 0.220486

---------------------------

CALL-NAME DURATION % CALLS MEAN MIN MAX

TOTAL (7) 20.024742 100.0% 19,818 0.001010 0.000000 0.138701

---------------------------

XCTEND 0.000000 0.0% 1 0.000000 0.000000 0.000000
pooled connection free 0.000000 0.0% 1 0.000000 0.000000 0.000000
PARSE 0.000000 0.0% 2 0.000000 0.000000 0.000000
SQL*Net message to client 0.042101 0.2% 4,954 0.000008 0.000000 0.020164
FETCH 0.952060 4.8% 4,952 0.000192 0.000000 0.128008
EXEC 2.680166 13.4% 4,954 0.000541 0.000000 0.128008
SQL*Net message from client 16.350415 81.7% 4,954 0.003300 0.000000 0.138701

---------------------------

CALL-NAME DURATION % CALLS MEAN MIN MAX

TOTAL (32) 1,637.118681 100.0% 74,351 0.000220 0.000000 0.220486

---------------------------

pooled connection free 0.000000 0.0% 1 0.000000 0.000000 0.000000
PARSE 0.000000 0.0% 2 0.000000 0.000000 0.000000
cursor: pin S 0.000005 0.0% 5 0.000001 0.000000 0.000005
cursor: pin S wait on X 0.025463 0.1% 4 0.006366 0.000000 0.011166
SQL*Net message to client 0.094487 0.2% 18,585 0.000005 0.000000 0.018525
FETCH 2.388145 4.8% 18,583 0.000129 0.000000 0.120008
EXEC 6.684420 13.4% 18,585 0.000360 0.000000 0.128008
SQL*Net message from client 40.507440 81.5% 18,585 0.002180 0.000000 0.220486
Performance feature wish list

Make it easy for developers to get trace data
Trace p\% of executions of a given task
Create profiles from trace files
Aggregate trace files by arbitrary dimension
Show profile differences side-by-side
Predict when pth percentile R exceeds threshold
Report tasks with excessive R variance-to-mean ratio

Let me know, at info@method-r.com.
References

Detailed information about instrumenting your Oracle application code.

One-day course including software that teaches you how to master Oracle trace data.

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

http://method-r.com
http://carymillsap.blogspot.com
@CaryMillsap
@MethodR

Method R