A practical approach to understanding and dealing with Application Response Time issues

By
Gary Piper
WWW.**PIPER-Rx**.com

Whenever someone mentions application performance monitoring, we are immediately steered towards "End to End" monitors. End to End monitoring, such as it is today, provides so much detail (usually presented in technical terms and format) and unfortunately, technical staff try and use it when explaining to business users why the application is slow. Often this achieves very little and just adds to user frustration. IMHO - End to End in most cases is whatever information a tool vendor can collect, which, while it has some intrinsic value for the IT group (root cause analysis etc.) it is usually far too technical to communicate anything useful to the end user. You can almost see a user's eyes fog over when you try and explain the application is slow because the database segment of "the transaction" was taking 300 milliseconds when it normally takes 20. Most of the terminology means nothing to your typical business user – they just get angry at what they see as a useless explanation.

Remember, this isn't saying this information is not useful to the IT department when trying to identify the root cause of a performance issue, but it should stay in the IT department...

Another issue with End to End is actually deciding what constitutes End to End: A connection to the application, a full transaction like creating a purchase order… you get the picture – it's very different depending on your viewpoint! In most cases each individual thinks they have the most appropriate view on this and much miscommunication then ensues. In today's environments End to End is in most cases is whatever information a vendor can collect and present back to you, regardless of the extent to which it is meaningful.

In my experience, in nearly all cases we tend to lose sight of the basic user's statement "it's slow" - a simple statement that deserves a simple response but is often instead followed up by IT with a frustrating, often meaningless, detailed technical answer.

The next most frustrating action I typically see is that when there is a performance issue, IT go looking for poor performing code – which in most cases, in OEBS you cannot use. "*Our monitoring tool shows you the bind variables*"....Even if your monitoring tool collects bind variables e.g. GL ***code_commination_id*** as part of a general ledger transaction, you still have to have enough functional knowledge to translate the value into a Chart Of Accounts code, and then determine if the Chart Of Accounts value was a large range (which drives slower performance than a small range).

My position has always been:

"Not all performance problems are caused by poor performing code"

In fact in my long experience of over 20 years and hundreds of sites it is my view that in a stabilised environment, the majority of performance issue are caused by inappropriate user behaviour, particularly at high processing times when users are stressed and under pressure.

The performance issue could just as easily be due to a user who wanted to run a trial balance for one area of one division and instead entered the wrong code segments and has run the trial balance for a large number of divisions. "Oops I did not mean to do that"… but it happens time and time again.

You need to remember that the essence of business processes is that they are very consistent, users generally do / perform the same thing / process the same time each business cycle. If statements were slow last month they will be slow this month etc...

There will always be higher activity and higher probability of performance issues during the month end close.

So the moral of the story is, before you go looking for poor performing SQL, check what your users are doing to you first.

What is a transaction?

This question is one that can be debated ad-nausium and everyone has an opinion. Some of the key issues with defining a transaction are covered in the paper "A business approach to Oracle E-Business Suite response time, E-to-E and SLAs" which is posted in the Papers section of the **PIPER-Rx** website (It's a bit tongue in cheek but I like to think it gets the point across!)

The user says "It's slow !…"

This raises one of the greatest sources of user / IT communication frustration that exists today. Let's get back to basics. What we need to do is start with the end user statements like:

- "It's slow!"
- "This stupid system doesn't work!"

What often happens next tends to frustrate business users, as the technical people now want to record database activity down to the lowest level and communicate all this great information back to the user. Not that there is anything wrong with this information per se, it is just that ***it doesn't help the user understand cause and effect in a way they can do something about*** so they get very frustrated and all too often by pass IT if they can.

Effectively users don't want a long and involved technical answer. All the techie stuff is for IT and should be kept there. In the first instance a non-technical statement like "its slow" requires a non-technical answer:

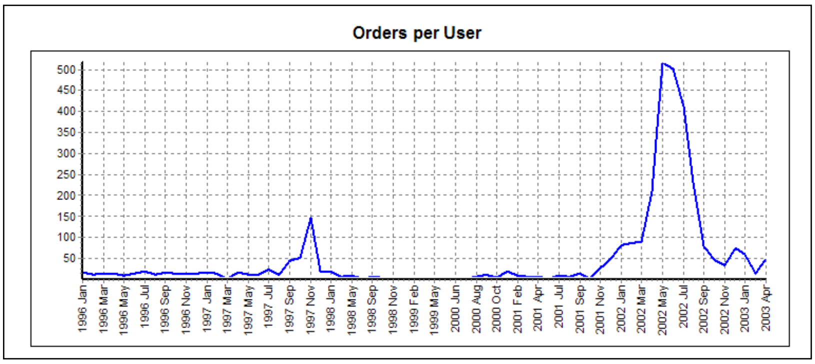
- "No it is not" (and I have the evidence)
- Yes it was and we are looking into it

You can tell the users later if needed that is was the GL department running 20 FSGs all at the same time. (If you have ever managed GL you will know what I mean. ☺)

A user under stress will complain

You should bear in mind performance complaints can be due to frustration by the end user. If they have too much work to do or a tight deadline, they will more often complain about application performance.

To illustrate, the following is an example of a report I wrote back in 2006 showing the number of orders being entered per user:



Note the increase in activity for May and July. A three to five second delay in the invoices screen (which is normal) is acceptable when you are doing 200 invoices per day, but it becomes a problem if you have to do 500 per day.

As a company grows the transactional volume increases and organisations tend to save money by maintaining or reducing staffing levels…..ergo increased user work load.

This is just another example of not all performance issue are technical.

A Simple Performance Measure

Let's assume there is already some "all seeing all knowing" form of technical End to End monitoring tool in place in your organisation that measures all the technical stuff for the technical purist.

As for my part, I have always taken a very simple approach to application performance from an applications administrator perspective by using a performance measure that is ***understandable to the end user,*** reproducible, consistent and freely available

All I do is measure the run time of the concurrent program FNDOAMCOL. This is a great program, FNDOAMCOL is required to be run for the Oracle OAM product (default every 10 minutes) and it is so important it even has its own dedicated concurrent manager.

The great thing about this program and why it works for measuring overall application performance is that it performs the same tasks across the same objects each time it runs; it's runtime is totally dependent on application and database activity.The program generally has a minimum (baseline) runtime of 2-3 seconds and when its response time slows down, it will be as a result of other activity occurring in the application at the time. How simple is that...

If the application is slow and the response time is ok then it is not an application based issue, it's a network or other factor …☺ If the response time is slow then it is the application or database.

So by reviewing the FNDOAMCOL response time output you can provide the simple answer to the statement it's slow:

- No its not (and I have the evidence), or
- yes it is and we are looking into it

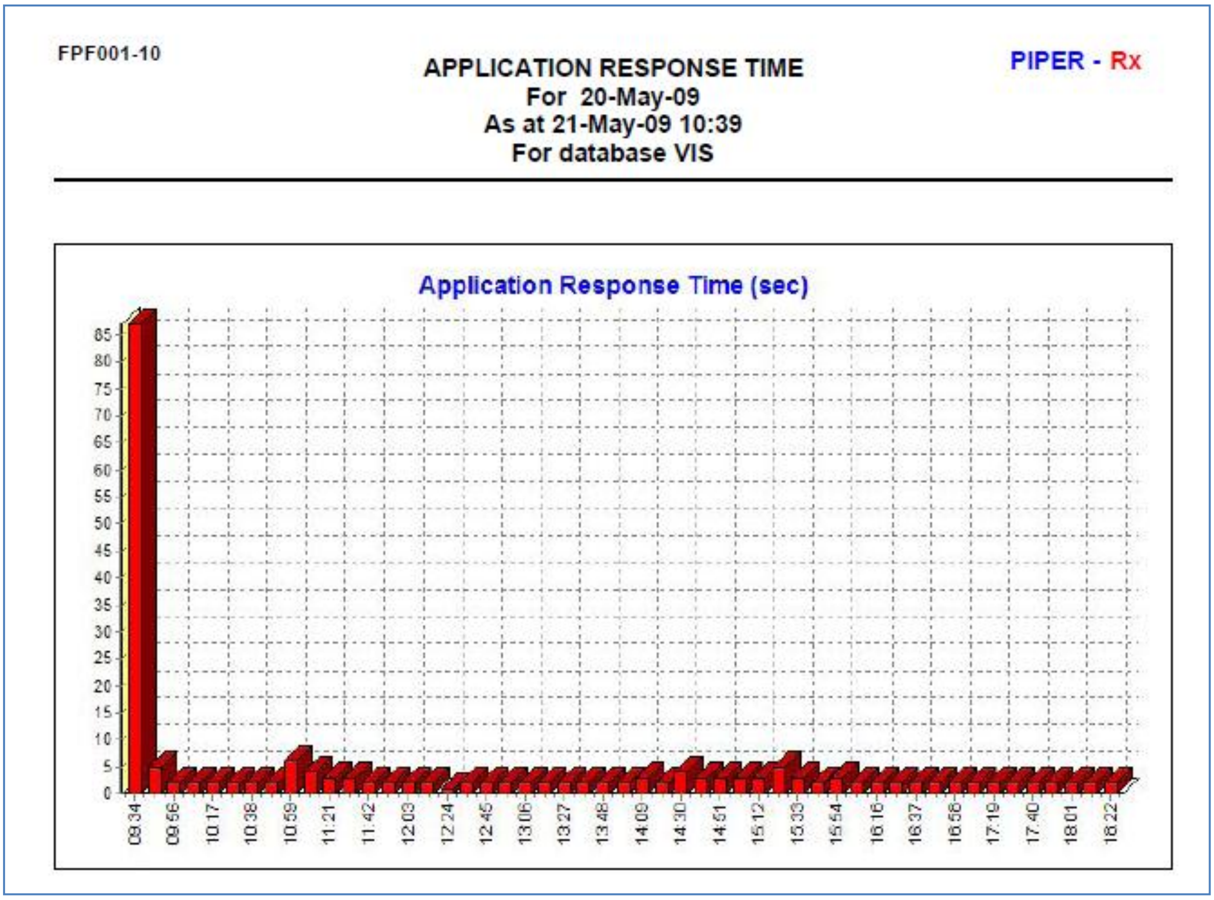
...and remember to answer with confidence, users can smell fear!

TOAD Report Manager Reports

To assist I have provided a simple response time report in the TOAD Report Manager format (TRD). This reports the response time activity (FNDOAMCOL runtime) for a selected day:

http://www.piper-rx.com/pages/reports_free.html#_performance

Example Report – Page 1



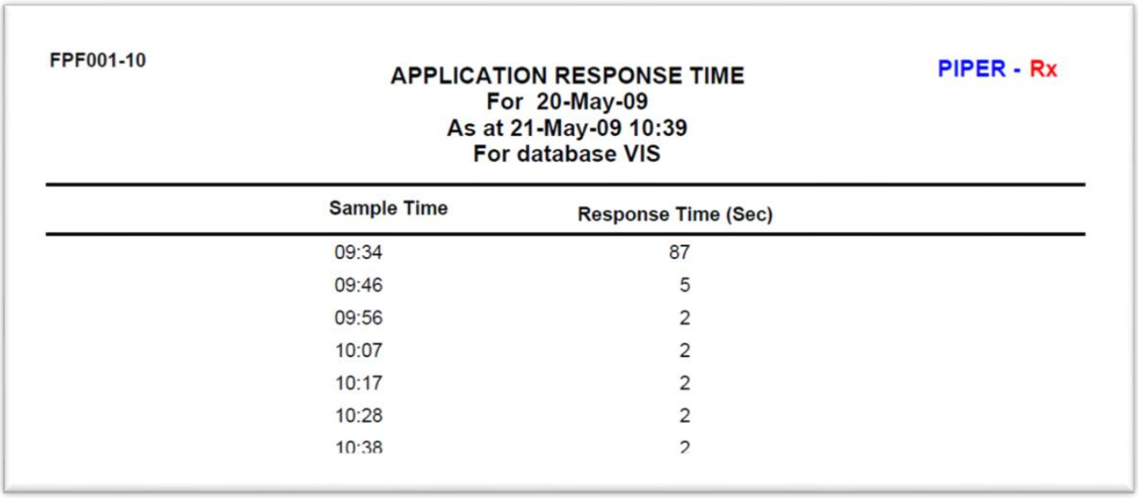
FPF001-10

APPLICATION RESPONSE TIME
For 20-May-09
As at 21-May-09 10:39
For database VIS

PIPER - Rx

Application Response Time (sec)

Example Report – Page 2



FPF001-10

APPLICATION RESPONSE TIME
For 20-May-09
As at 21-May-09 10:39
For database VIS

PIPER - Rx

Sample Time	Response Time (Sec)
09:34	87
09:46	5
09:56	2
10:07	2
10:17	2
10:28	2
10:38	2

This report will report on all FNDOAMCOL runs that exist in the ***fnd_concuurent_requests*** table.

So now you have a simple, FREE easily understandable application response time measure you can report on and discuss with your users.

Good Luck with using it! It has worked for me for many years........

Want to know more?

There is loads more ***FREE*** information on this topic and all aspects of OEBS Application Administration at the **PIPER-Rx** website. After over 20+ years working with Oracle (the product, not the Company) and Oracle E-Business Suite (since Release 5) I have visited countless sites and pretty much seen it all when it comes to Applications Administration. Since the late 1990's I have spent more time sharing these learnings and the most popular papers and case studies I have presented are available at the **PIPER-Rx**.com website as well as a whole host of Tips and Reports I have used throughout my career.

All information at the **PIPER-Rx**.com website is ***FREE*** so why not check it out….I hope you find it useful! **– 30,000+ downloaders to date can't be wrong!**

Disclaimer

The material contained in this document is provided by the author "as is" and any express or implied warranties, including, but not limited to, any implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the author be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of any content or information, even if advised of the possibility of such damage. It is always recommended that you seek independent, professional advice before implementing any ideas or changes to ensure that they are appropriate.