Aphorisms

By David Kurtz, Go-Faster Consultancy Ltd

Several years ago I realised that I had a collection of catchphrases and quotes that I was using when discussing performance issues. I even put them all into a rather frivolous presentation. However, they do express valid points. To successfully address performance issues, you have to look at the problem in the correct light.

Performance is exactly what the user perceives it to be. No more, no less

The availability and response time that a system produces are a sum of all the components in the technology chain: disk sub-system, operating system, database, application server, network and PC.

It takes time for requests to be transmitted over a network, through an application server, be processed by the database, which will also have to access disks, and then to be returned. There is even processing on the client side just to render a screen. It all takes time.

If any component fails or performs poorly then the whole system is perceived to fail or perform poorly. The users can only observe one thing: the hourglass icon on their own PC.

If you go to a cash point and you can't draw out money, you do not care whether the central computer is down, or the network, or the cash point itself is down, or even if it has run out of money. The bottom line is that the system is not available, and you do not get your money.

Look at it from the user's point of view.

'Poor performance' occurs when the user's perception does not match their expectation

One of the problems with performance is that it can all be a bit touchy feely. Users usually say that 'the system is slow'.

You need to be able to measure performance as perceived by the user, and you need some sort of metric, perhaps part of an SLA that defines 'acceptable' performance. This cuts both ways. It can be used to tell the users that the performance is actually reasonable and not to take action, or it can be used to justify the time and effort of a performance improvement exercise.

An implication of this is that you can tune the users as well as the system. However, advice on that topic is outside the scope of this article.

Performance Tuning is a search for lost time

Ultimately, time is the only metric that matters. Everyone knows the phrase 'Time is money', I sometimes say that 'Time is the only currency that the user understands'.

Usually, concentrate on time, not simply I/O or CPU utilisation. It normally all goes together, but sometimes the other measures can be misleading. The statement with the highest I/O statement may not always be the one that takes the longest.

So, identify the actual transaction that is causing the most pain (this often approximates to the longest cumulative execution time) and address that first.

Be careful when using system-wide or aggregated measurements. The long-running statement that isn't part of the user's problem transaction is probably not part of the users problem. In fact it may not be a problem at all.

Concentrate on the Big Picture

Keep in mind what you are trying to achieve and why. Most systems support business activities. Good performance helps to deliver competitive business advantage and/or reduces costs. Ask yourself whether a perceived problem impacts on those criteria.

The 'Top 10 Whinge List'

Sometimes, it can actually help to ask the users what they think their problems are. Get them to classify what is most important to them.

If necessary get a user to demonstrate while you look over their shoulder (this can also be good for relations between the users and the IT department, a.k.a. 'performance tuning by walking about').

When you have eliminated the impossible, whatever remains, however improbable, must be the truth

'A Study in Scarlet', Arthur Conan-Doyle

In multi-tier systems it can be difficult to directly measure some components. When trying to identify a bottleneck it can useful to prove that a tier is working properly and so determine that the problem lies elsewhere.

It is a capital mistake to theorise before one has data

'The Memoirs of Sherlock Holmes', Arthur Conan-Doyle

It is very easy to jump to a conclusion. Take time to do the analysis properly otherwise sometimes you get caught.

When you do the analysis properly you may get an unexpected answer. The answer might be counter-intuitive; it may be so peculiar that you might have rejected it out of hand. However, it might also be the truth.

Detection is, or ought to be, an exact science. It should be be treated in the same cold and unemotional manner

'The Sign of Four', Arthur Conan-Doyle

The methodology of tuning is that of the laboratory, it is experimental science.

If you want to test or prototype something then you must conduct a realistic test while being careful to make realistic assumptions.

1. Observe & measure
2. Analyse
3. Formulate change
4. Predict effect from theory
5. Make change
6. Rerun test
7. Measure effect
8. Confirm (not Prove) or Disprove theory. If not satisfied go back to the beginning.

Singularity is invariably a clue. The more featureless and commonplace a crime is, the more difficult it is to bring it home

'The Adventures of Sherlock Homes', Arthur Conan-Doyle

And some of the SQL I see is nothing short of criminal!

I do not believe in coincidence. If you find something odd, don't immediately discount it as a fluke.

If you don't like the answer, change the question

A change in performance is due to a change somewhere in the system

- Infrastructure (disk subsystem)
- Data volume (cbo statistics)
- Index
- SQL code
- Application.

Sometimes an approach does not work for whatever reason.

Sometimes a programming technique does not scale, or because data volumes, or relative data volumes are different.

There are no rules here. No golden nuggets of advice. You are going to have to change something, often the question is what are you actually able to change.

Keep it simple

Generally, the more simple a thing is, the easier it is to understand, change, test, maintain, upgrade.

The more simple a SQL statement the less likelihood there is that the optimiser will take an inappropriate path.

It has long been an axiom of mine that the little things are infinitely the most important

'The Return of Sherlock Holmes', Arthur Conan-Doyle

You can find yourself wading through acres of trace files. And often you are looking for a very small fragment of information.

So make sure you are monitoring the right things.

To the curious instance of the dog in the night-time. The dog did nothing in the night-time. That was the curious instance

'The Return of Sherlock Holmes', Arthur Conan-Doyle

Sometimes it is more important that something did not happen.

The problems change from release to release, but the methods by which they are investigated remain the same

Jonathan Lewis

The march of so-called progress is inevitable. Software versions change, behaviour changes, bugs appear and disappear.

What matters is that your methods of analysis are rigorous, and that you follow the chain of evidence to its logical conclusion.

There are only two reasons for poor performance. You are either working too hard or you are being prevented from working

Jonathan Lewis

I'll leave you with a final piece of Sherlock Holmes.

You know my methods, apply them

'A Study in Scarlet', Arthur Conan-Doyle

About the Author

David Kurtz specialises in the performance tuning of PeopleSoft applications mostly on Oracle databases, and is currently the chairman of the UKOUG Unix SIG. He can be contacted on 07771-760660, or e-mailed at david.kurtz@go-faster.co.uk