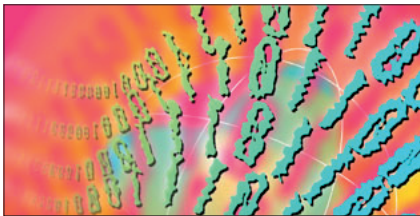


## Optimising Oracle performance

Published: 26th April 2006  
Channel: Data Management

**I**t is rare that I get a chance to use what I learned in my degree course (and even rarer that I remember what I was taught), so it was a pleasure recently to be briefed on BEZProphet, a database performance management tool for Oracle databases.



Now, database performance management tools, while not exactly two a penny, are not uncommon beasts, but BEZProphet has something special going for it, which I will come to in a moment.

BEZProphet basically does three things: it analyses performance, makes predictions and notifies the administrator of those predictions. It can also collect relevant performance data if necessary or it can use the figures collected by Symantec InDepth for Oracle or (in the next release) Oracle Enterprise Manager or similar products.

In terms of analysis, at a high level the product works from the perspective of a business view. Thus you might look at the performance load that different sales divisions were placing on the system, how much work the finance department was doing and so on and you can see how large these are with respect to one another and which is growing, shrinking or standing still. You can then drill down into any group to as much detail as you like, ultimately down to the level of individual SQL statements, tables or joins.

Based on these analyses, BEZProphet can recognise trends (as it stores historical performance information) and project when a problem may occur. For example, it might forecast that on current growth rates the response time

to the French sales division will exceed service level agreements in, say, July.

Suppose that this is the case, then BEZProphet will send an alert to the DBA, it will isolate the main cause of the problem (CPU, disk, memory and so forth) and recommend a potential solution such as to create a new index, create a materialised view, or more memory. Further, BEZProphet can predict the result of making these changes and re-calculate trend lines based on such changes.

This is the exciting part: BEZProphet can perform these predictions and simulations in a matter of, literally, seconds. The time to run such simulations using conventional techniques is more usually measured in days and, given that you may have to run multiple simulations before arriving at an optimal solution, this means that BEZProphet can help you resolve database performance issues in a few hours that would otherwise take weeks or even months.

And now here's where I get to use the fact that my dissertation was on Graph Theory. The reason why BEZProphet is so fast is because it uses an iterative calculated closed queuing network model. Now, a graph is a series of nodes that is connected by arcs. A network is a graph in which the arcs are directed (that is, information flows along the arc in one direction). A queuing network is one in which there are tasks, or jobs, that flow through the network. In other words, standard computer networks are queuing networks and a queuing network model is exactly that: a representation of a queuing network.

Now, for mathematical purposes (that is, modelling what goes on in a real network) there are two types of queuing network: an open network and a closed network. In an open network tasks are loaded onto the network, flow through it and then leave it, exactly as in real life. However,

the problem with using this approach for simulating a real environment is that you have to continuously create input streams to feed into the network. And, of course, this has to mirror the real world. This requires a lot of processing power, which is why it is typically done off-line, as a batch exercise, and is why it takes a long time (actually, this also applies to non-iterative closed networks).

In a closed network, on the other hand, nothing comes into or leaves the network: the duration of each task is infinite. The problem with this approach (which is mathematically richer) is that it doesn't reflect the real world, because things don't last forever and new tasks do enter the network. What BEZ Systems (the developers of BEZProphet) have done is to build algorithms that allow a closed network to be iteratively refreshed. Thus BEZProphet, which runs on its own server, is continually active, with its model of the environment being incrementally updated on a regular (typically hourly) basis.

So, bottom line: if you are into proactively managing your Oracle database then you should give BEZProphet serious consideration, because it is doing things that no other offering (that I know of) can do. The product is available in the States from BEZ Systems and in the UK (and Europe) from Application Performance. BEZProphet is limited to Oracle database environments at present but the company has announced that it will be extending its capabilities to application and web servers towards the end of this year. No doubt the company will consider the needs of other database users in due course.

Philip Howard  
*Director of Research, Bloor Research*

  
...optimise your IT investments  
© Bloor Research 2006