Donuts for DBAs V – Redmond talks about Grid Computing
By Redmond Bim.

Those that did not attend Oracle World in September this year must consider themselves the luck ones. They are lucky because they were not subjected to grid computing.

In a move worthy of commending[*], Oracle allowed internet viewing for the listening of key note speakers and database guru's. There was no charge for this and it allowed for the general population to be virtually there.

Being virtually there has a lot of advantages to actually being there. You can chat and make glib comments about the speakers, you can munch on donuts and pause the show to get a fresh coffee, and you can very quickly stop the show if it gets to hard to watch.

I knew before I started watching that this year's theme was "Grid" computing. The grid isn't a new concept but it is a powerful concept. I didn't realise that I would be indoctrinated by the message of the grid. Indoctrinated isn't probably strong enough, it was more like mind - towel thrashed. Image grabbing a slightly wet towel, twirling it around to form a towel whip, attach the word "*Grid"* to it in flaming molten metal, and then repeatedly flick it at your head.

After three days listening and being subjected to this, I then coined the term "Bored of the Grid", mainly because the message coming out from Oracle was very confusing and repetitive. Also because it seemed similar to a book I once read called "Bored of the Rings", and I am in the camp where I found the second movie in the trilogy effective at curing insomnia. I strongly suspect the management at Oracle have latched onto this concept because it is trendy and in attempting to brow beat the general public into believing that it is the next big thing (after Java off course, which is now old hat), they have muddled the *griddy*[†] waters.

Being Redmond and being sceptical I decided to do some background checking on what the grid is. So naturally enough I went to Google and typed in "what is grid computing?", and received no response. My ADSL connection was down - so much for high availability. Interestingly it highlights a key flaw in the basic concept of grid computing, which assumes you have a solid internet connection.

Not to worry, in true Redmond fashion I will give my opinion on what grid computing is and make out that this document has been well researched and is based on true facts [‡]. I will use occasional references to obscure documents that have not been published yet. Actually, I will come up with new terms and definitions and lead the industry in what I think grid computing really is.

From watching the grid video's at Oracle's home page, the message that come through was that the grid for computing is like the electricity grid. From other high profile industry sites I found that the grid is a way of utilising computer resources that would otherwise not be used.

Redmond sees the grid as the natural evolution of the internet. As we use it

[*] Ra,ra. Ra ra ra.

[†] copyright pending

[‡] As opposed to the other facts which are not so true, or commonly referred to as marketing

more and more, we become more dependent on it. Services like banking, listening to music, buying items and reading the newspaper, can now be done on-line. Applications and functions are being moved from local (being my home or work office) to the internet.

The grid is different things to different companies, and this is where Oracle's confused message comes in. In an attempt to be seen as an industry leader they jumped in and emphasised that the Oracle database was the grid database. The next release of Oracle 10g, is *g* for grid, just to emphasise that point.

The grid though can be seen as three different models of computing usage, and Oracle only satisfies one of these models properly. I understand why Oracle had to jump early, marketing and selling is different to using, and like any good company if they want to survive in this competitive world they need an edge, especially an edge that Microsoft has trouble pushing (though I do suspect that Microsoft's marketing engine is slightly more formidable than Oracle's, and within 18 months they will tout .asp and .net as the grid architecture of the future).

Oracle did a similar thing with Oracle8. They pushed Objects as the next big thing without having the true database architecture to adequately support it. It wasn't until Oracle 9iR2, that the version was mature enough to deal with objects. I suspect a similar thing will happen with 10g. It will not be until Oracle Release 11 that the definition of what grid computing actually is will be properly understood and Oracle will be able to handle it.

The electricity grid is based on plugging an appliance into the wall socket and taking electricity (and getting charged for it[§]). This is the big difference with Grid computing. With grid computing you first have to push your data and the processing request to the grid and then take back the results. In some of the models I will go through, the data is already out there, but in all cases you have to push a processing request for how to handle that data out, and you get data back.

For example, with an on-line bank, you push out the request to transfer funds and the resulting data coming back is a HTML page.

Model 1: Centralised Services

This is the current mainstay of the internet and is what Oracle eludes to (but doesn't really admit to) when it talks about the grid. In this situation you push your data out to a site and then later you push out requests to retrieve that data. On-line banking and internet email are two examples that come to mind. More services are appearing daily covering different lines of business. At Oracle World, Oracle stressed how by using 10g you can efficiently scale and centrally manage these applications. No arguments with this one - Oracle pushes RAC, high availability and a one-stop simplified HTML based enterprise manager. Auto tuning and recovery were talked about by Ken Jacobs. This model is one that Oracle does very well at and the new features in 10g do make it easier to work with. It is though only one of three models.

Model 2: Distributed Processing

The real power of the grid came in with the concept of making use of idle computer resources to perform complex mathematical calculations.

[§] Hey that's a pun, and Redmond does not like puns, so please treat it with the utter disdain it deserves.

The Search for Extraterrestrial Intelligence (SETI) pioneered this concept with a screen saver that could make use of your internet connected computer to perform analysis of radio signals. A lot of academic institutions talk about this concept as the grid. In this model, you push out your data and processing requirements and any grid computer, when idle, processes the request.

You can't do this with a centralised service. You cannot send out a SQL statement and have it run distributed across multiple computers, mainly because the data is in one spot, and the SQL statement can only run on one computer with the same database on it (or RAC one).

The distributed processing model is aimed at problems that are intrinsically shared nothing in their architecture. Not many computing problems in the industry are well suited for this, though most in the academic world are.

If a problem can be broken down into small chunks, and each chunk is separate and independent of the others, then this is one that the grid using this model can solve. In the early days of computing, shared nothing seemed like a good architecture, mainly because it was the university academics that had problems that could be solved by the architecture. As a computer outgrew its resources, you just tacked on a new one and the resources were shared equally between the two. As we all now know, this architecture is doomed for failure with any data based query problem. Try running a financial system transparently on a shared nothing architecture and watch it slow down. Its because a financial system does not have parts that can be broken down and solved separately. A table join between two tables cannot be broken down into independent units solved on separate computers as all the data in each table is joined with the other data - it is shared.

Good examples of problems solved by this model are simulations like weather, economic, nuclear reactor modelling, or the search for new proteins.

Which leads to an interesting aside, the next wave of parasitic viruses yet to emerge are ones that use this Model to spread. They are simple and deceptive. What happens is that you visit a web site and suddenly notice for about 20 seconds a rapid increase in CPU usage and then it stops[**]. Most users would think nothing of it as the time it takes to format a page can take this long.

The parasite though has done its work. Cleverly built into Javascript (even in a pop up window that is hidden), it runs and does its work. It will processing part of a shared nothing problem. The download of HTML page downloads the javascript code (processing) and the data. It does its work and then sends the results back. Its not a virus as its not damaging or even accessing your hard drive, all its using is your CPU. It only runs for a short period of time (no need to alert anyone to its presence) and then stops. It will be hard to detect and even harder to stop, and initially most people will think nothing of it (just like Spam), until nearly every site you go to appears sluggish as it consumes CPU resources. It will happen, especially when companies needing these computational problems solved start paying heavily accessed sites to embed the parasitic javascript code. Its just

[**] if you are observant, but as far as the parasite is concerned you have no idea how to monitor the CPU

like advertising except you don't see the ad.

And its legal - well I think it is. Actually I am not a lawyer so in reality I wouldn't have a clue whether its legal or not, but based on extensive research with legal minded people the conclusion is that it is likely to be legal.

So the Oracle database, or any database for that matter that relies on SQL, does not readily handle shared nothing problems. It will take a long time to solve it as the inherent nature of relational structures prevent it from being solved. [††]

Model 3: Network Model

This is the one that excites me the most. Not as exciting as a caramel custard yeast donut freshly baked and hot just out of the oven (mmmm), but exciting none the less.

In this model, different centralised applications start transparently sharing data and problems on the grid. At the moment the only way of achieving this is through XML, which is an open standard.

A good example of this is through exchange rates. A business builds an e-commerce site, but rather than building an application to manage exchange rates, the application simply makes an XML call to a site that already has the exchange rate data. It passes with it an XML query asking for the current rate and a list of countries with that rate changed. The business receives the result back in as XML data, and then displays it to the customer. This could be displayed as if it was part of the business application.

Other concepts for this include a thesaurus, dictionary, language translation, video and audio. All of these can be used to enhance a centralised application and simplify the development of them.

Now the Oracle database, ever since 8i has had tools built into the database to allow this to happen. What they do not have is the inherent architecture to truly support it. So while they might be able to put a tick in the box to say they do it, the database is not yet mature enough to deal with it. Application developers will need to build the smarts to interface with the sites and pull the data in. I am not aware of any database vendor that is close to supporting this model. There are a number of organisations though actively pursuing this path, but they go down the path of allowing all data to be shared. For this model to be correctly used, it will mean it will have to follow a set of universal standards. It will also have to be query language, database and operating system independent.

At the moment the standard used for transporting the data is XML. I imagine the XML Query Language will be used instead of SQL, but as anyone has found out when they use it, it is difficult, painful and non-intuitive to use. In fact its use should be actively discouraged as its continuous existence will likely hamper the use of this model. Redmond calls for the banning of the XML query language. (Redmond also calls for Oracle to ban RMAN which is another topic)

Though XML is a good and easy to use standard for transporting data between internet sites on the grid, the

[††] Talk about Object structures and there might be a possible solution, but that is for another paper. Actually its going to be the next one I write.

interpretation of that data is not defined within the language (yes, some will say it is, I cheerfully say its not, and as I am writing this article I get the final word). For example, there is no standard defined for interfacing to a bank. Each bank, though using HTTP (none are yet advanced enough to use XML), uses a different method for passing merchant and purchase order data to and from the bank. Not only does each bank do it differently, they do it very differently. Try building an interface to them and see what I mean - endure the pain, experience the suffering. One would have thought with the huge amounts of money we pay in interest rates on credit cards some bright spark in the banking industry would have been able to come up with a standard (and I tell you this, its not rocket science to do, I could come up with one in ten minutes, and have).

So there it is, the Grid clearly and concisely explained and how Oracle partially addresses it. No doubt in the future, Oracle will get there, but in the meantime, be wary of the advertising and hype around Grid computing and just see it for what it is, a natural extension and evolution of the internet.

Redmond misses the old Oracle Open World days in Australia and would like to see them come back. In addition to being a great place for networking, it allowed Redmond to taste the culinary delights of donuts in a different capital city every year. It was also a great excuse for drinking beer and seeing groups of highly intoxicated techno-nerds attempt to reinvent dance moves at the Monday night BMC party.

If you want to contact Redmond then please do, but there are no clues as to how to do it. If you would like to see Oracle Open World start up again then talk, no scream at your local Oracle User Group reps and tell them to get their act together. They do listen but they are hard of hearing so hammer the point home with a healthy pint of Guinness and a pack of fresh donuts.