Coldfusion and Oracle 8i Performance

Posted At : March 9, 2008 12:07 AM | Posted By : Mark Kruger
Related Categories: Coldfusion & Databases

Recently I had the opportunity to work with a server setup that included two load balanced servers running Coldfusion 7 enterprise backstopped by a hefty Enterprise Oracle 8i server. The CF 7 servers were going down repeatedly. The database people blamed the web people who blamed the networking people who blamed the government. No one seemed to make any headway on the issue. I was engaged to get the CF 7 (and 2 CF 8 servers using a separate configuration) stable. I arrived on site with my change document in hand. I learned, or I was reminded of, a number of things during the experience that I will try to boil down into two or three posts - including:

- **Don't Accept the Default Settings** - It's surprising how many enterprise servers I run into that have been installed but left with the "default settings". Naturally the default settings are simply inadequate to support any usage beyond the bare minimum.
- **Most Code Evolves Without Planning**
- **Team Troubleshooting is Key** - When systems are highly complex, no one person will ever have all the necessary information to explore all the possible solutions. It takes bringing together folks from the various disciplines (networking, database, Coldfusion, Linux etc.) to really make it work.
- **Big Oracle Tip** - This will be the topic of this post

Here's the skinny on my Oracle tip. After many optimizations we were still seeing a failure of *all* systems connected to the Oracle 8i server that used the same Oracle user to connect. The user was set up with a limit of 60 connections. Naturally we thought that increasing this upper limit would help. The theory was that with 6 servers connecting (1 CF 6.01, 2 CF7, 2 CF8 and 1 PHP) we just needed more headroom. Simply increasing the number of "allowed sessions" on the user profile did not have any affect however. The connections just kept climbing until they crossed some new threshold and the systems bombed again. Google was very little help. There are surprising few concise pages about Oracle anywhere out there - let alone concise pages about Oracle and Coldfusion. Finally, I stumbled onto **this gem** from the Adobe knowledge base.

The article references the process whereby CF calls the Oracle service regarding the status of a connection in the pool. This process is called the "skimmer" by the article. It is supposed to "clean up" connections in the pool that are specifically dropped by Oracle. That was not really *precisely* our problem, but it was close enough to get us looking in the direction of the "idle timeout" setting. What we found was that there was no idle timeout set on the Oracle server for the web user - meaning the creation and destruction of connections was left entirely at the discretion of the driver or system connecting to the DB.

Because Coldfusion doesn't terminate these connections in a way that Oracle understands, the end result is that Oracle *Oracle sees inactive sessions that Coldfusion knows nothing about*. Coldfusion never picks them up and uses them as if they were a part of the pool. Instead, Coldfusion ends up creating new connections to replace the old ones that it thought it had deleted. On the server we were working with there were 60 connections that Oracle "knew" about, but 50 of them where "inactive" and

not being used at all. This left 10 connections for all of our systems to work with. More importantly it meant that any new request for an additional connection would be denied.

The Fix

Here's the fix as we devised it.

- **Set the Idle timeout in the Oracle User Profile** - This will guarantee that at a minimum these "idle connections" (seen as "inactive sessions" in TOAD) will be terminated by Oracle when they reach that threshold.
- **SQLNET Probe** - There is a setting for idle network connections in the SQLNET service that will probe the connections at specific intervals and try to ferret out these orphaned sessions and kill them.
- **Pool Connection Timeout** - Under the advanced settings tab for the datasource in the CF Administrator, underneath the "max pooled statements" there is a setting that just says "timeout". This is not the timeout for the pool. It is the timeout for connections within the pool (in minutes). The second text box says interval and as I understand it, it is the frequency that the "skimmer" should check the pool for connections that have lived beyond the "timeout" value. We set the timeout to 10 minutes for a quicker rotation and we set the interval to 5. This makes the JDBC service busier - but it also guarantees (along with the settings on the Oracle side) that we will have regular processes attempting to clean up the connections and keep them fine tuned to a reasonable level.

After these changes the results were quite decisive. The pool climbed to a reasonable level and then settled into a pattern of climbing for a few connections then dropping a few connections, climbing, dropping, climbing dropping - as the 2 sides both kept the connection pool in check.

Please note, during the process of this troubleshooting (like the most troubleshooting) I was googling furiously and looking for solutions. Coldfusion folks are vociferous bloggers but I found a decided dearth of blogs about Coldfusion and Oracle. I'm sure one of the muse readers is going to jump in and point me to the exact blog entry or article I needed to see this right away. In the meantime, I hope this blog is of some use to folks connecting to Oracle 8i using Coldfusion.