Funny Coding Errors - Take 2

Posted At : June 10, 2005 4:31 PM | Posted By : Mark Kruger
Related Categories: Follies and Foibles, Coldfusion & Databases

I was reminded of this very humorous incident recently while chatting with some friends. A very good friend of mine who was a novice web programmer was just beginning to stretch his wings using Coldfuion and SQL. The site he was working on (which shall remain nameless) used an Interbase DB server. My friend Bob (let's call him Bob) was interested in tracking page views for some news stories and articles he was writing. He created a new table with a few columns and wrote to it with each request - logging the page id, news story and IP address. It seemed to work splendidly... at least for a while....

A few weeks later Bob called me frantically. His news pages were slowing to a crawl. it could take as long as 200 seconds to display a story! What was wrong? I told him I would look at it. I had the sys admin turn on debugging briefly and I examined the debug code. It took 2 seconds to see that there was a single query taking up most of the execution time all by itself . Upon further examination I found this code:

```
<cfquery name="getLog" datasource="myDsn">
   SELECT    *
   FROM   logTable
   ORDER BY log_id asc
</cfquery>

<Cfset logIdList = valueList(getLog.log_id)>

<Cfset newLogId = listlast(logIdList) + 1>

<cfquery name="writeLog" datasource="myDsn">
   INSERT INTO logTable
      (log_id, page_id, story_id, Ip)
   VALUES
      (#newLogid#,#page_id#,#story_id#,#ip#)
</cfquery>
```

Those of you who see whats wrong may already be chortling a bit. Bob was using an int for a primary key - "log_id". Since there was presumably no identity field or autonumber field in Interbase, Bob was forced to "figure out" what the next ID in the sequence should be. He needed to get the last "log_id" that was added to the table, add one to it and use that for the log_id on his insert. His solution utilized what he knew - Coldfusion list functions. He selected *all* the data from the log table and ordered it by the ID from lowest to highest. He made a list of the ids using the valuelist function. He knew from experience that the last item on that list would be the maximum number in the table. He added 1 to it and Viola! - he had his new key.

Now this actually worked for Bob. The site performed pretty well for at least a week. But over time it got progressively slower and slower. Why? Because there were a lot of rows building up in that table. By the time he called me there were 214,000 rows being selected every time the news page was called. In fact, I was surprised that the page ever completed at all. It seems that Interbase is capable of passing large datasets to CF and CF is capable of building a list of 214,000 elements - who knew! The fix? Something like this (I don't remember if the interbase syntax was different).

```
<cfquery name="getLog" datasource="myDsn">
   SELECT    max(log_id) + 1 AS newId
   FROM   logTable
   ORDER BY log_id asc
</cfquery>


<cfquery name="writeLog" datasource="myDsn">
   INSERT INTO logTable
      (log_id, page_id, story_id, Ip)
   VALUES
      (#getLog.newId#,#page_id#,#story_id#,#ip#)
</cfquery>
```

Bob was just happy to have his page go from 190 seconds down to 200 milliseconds, But it brings up a point. If you are looking to expand your skills as a developer, you will be well-served to learn all you can about SQL. What's the number 1 performance problem on CF sites (or on *any* web site)? It's interaction with the database. It's shocking how often I run into code from otherwise good developers where they do things like - select all the rows in a table to get the "recordcount" attribute, or loop through rows calculating an average or adding things together. Let the database work for you! Learn about aggregate functions, grouping, unions, subqueries, stored procedures, defaults, functions, UDFs etc. If you are using a sophisticated DB system like MS SQL server, Oracle, DB2, (and many others), take the time to learn how to leverage those platforms. Such products are among the most finely tuned and optimized pieces of business software ever created. Why not let that work for you!