

## Re: Nearest Common Ancestor Report (XDb1's \$1000 Challenge)

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2004-05/1338.html>

---

**From:** Nick Landsberg (*hukolau\_at\_NOSPAM.att.net*)

**Date:** 05/24/04

Date: Mon, 24 May 2004 02:33:10 GMT

Hugo Kornelis wrote:

> *On Sun, 23 May 2004 00:13:06 GMT, Nick Landsberg wrote:*  
>  
>  
>>[ *Everything Snipped* ]  
>>  
>>*OK folks, I have found this discussion fascinating*  
>>*and educational. However I would like to add*  
>>*one more item into the equation, and it is not*  
>>*theory (although I notice that c.d.theory is*  
>>*in the distribution list) ---*  
>>  
>>*Given the possible implementations, how long*  
>>*would this query run on a 1 million record*  
>>*database? (That's actually small by today's standards)*  
>>*And then for a 10 million record database.*  
>>  
>>*Pick your own hardware, e.g. raid arrays,*  
>>*2.5 GHZ CPU's etc., but tell me how long*  
>>*this query would run, please.*  
>>  
>>*Theory is nice, but, in my experience,*  
>>*performance is what sells systems to*  
>>*customers.*  
>  
>  
> *Hi Nick,*  
>  
> *Fair question. In between answering Neo's latest messages, I was*  
> *constantly switching to my Query Analyzer window. I've been busy writing*  
> *out a script that will generate millions of rows in the things and add*  
> *random dependencies in the "leader" hierarchy, so I could test some*  
> *things.*  
>  
> *Some disclaimers to start with:*  
> *1. I think I'm pretty good at designing tables and writing SQL. But I have*

comp.object: Re: Nearest Common Ancestor Report (XDb1's \$1000 Challenge)

- > *second to none experience in tuning heavy-duty systems. I can (and*
- > *probably will) tweak some things here and there and gain some performance,*
- > *but it's not what I do best.*
- > *2. I don't have the money to spare on the kind of hardware that would be*
- > *present in a shop that runs this kind of queries on 1 million record*
- > *databases. I'll do some tests with large numbers, but all I can offer is a*
- > *Desktop PC, powered with one 1.3GHz AMD Athlon, 256 MB of RAM and equipped*
- > *with two 40GB 7200rpm IDE harddisks (I have the log on one of the HDs and*
- > *the data on the other – for this kind of queries, I'd sure like to plant*
- > *tempdb on a third HD).*
- >
- > *With only this computer to my disposal, I'll have to trim down the number*
- > *of rows considerably. It's not that my hardware can't handle a table with*
- > *1 million (or even 10 million) rows – but while waiting for the results of*
- > *my preliminary tests (using "only" roughly 0.5 million things), it*
- > *suddenly occurred to me that the report would be based on the Cartesian*
- > *product of the things table with itself – almost 250,000,000,000 rows for*
- > *my preliminary test, more than 121,000,000,000,000 rows if I would run the*
- > *complete test set I prepared (sporting over 11 million things). Just*
- > *sending the report to the screen would probably take days!! And I'd need*
- > *to add some really beefy hardware to store it.*
- >
- > *I just saw that it's past 2AM already, so I'll have to get back to you*
- > *later with test results from a "slightly" smaller test set.*
- >
- > *Two final notes:*
- > *1. I did already catch Neo's reply to you. I'll still try to run the tests*
- > *I intended, even though Neo probably won't. Just out of curiosity.*
- > *2. I expect my implementation to be quite slow when dealing with a large*
- > *numbers of rows. In fact, I was quite surprised when my first attempt*
- > *already managed to beat XDb1's execution time. Some time ago, I saw some*
- > *messages by Joe Celko in another newsgroup about an alternate way to store*
- > *trees. I don't recall the details, but from what I still know, I expect it*
- > *to be lots (and lots and lots) faster.*
- > *After finishing the tests on my own implementation, I might go try and*
- > *find his method, see what performance that one will yield. Just out of*
- > *curiosity.*
- >
- >
- > *Best, Hugo*

Thanks Hugo,

The reason why I posed the query is not so much to make work for you, but to get a flavor of how the two technologies compare /as they are now/.

If Neo's techniques prove out, they may be interesting for commercial work in several years. One never knows. As I alluded to upthread, these discussion remind me a lot about the

comp.object: Re: Nearest Common Ancestor Report (XDb1's \$1000 Challenge)

CODASYL vs. RDBMS discussions of roughly 20–25 years ago. Assuming that Neo's ideas pan out, I would not presume that they would supplant traditional RDBMS's, just as RDBMS's have not completely supplanted hierarchical and network models. There is more than one way to skin a cat, and you use the best tools available depending on the size of the cat :)

You and Neo have had an interesting discussion. And, what is rare on usenet, haven't let the discussion degenerate into a flame war. I congratulate you both!

--

"It is impossible to make anything foolproof because fools are so ingenious"

- A. Bloch