

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

*Source:* <http://coding.derkeiler.com/Archive/General/comp.object/2004-05/1433.html>

---

*From:* Hugo Kornelis (*hugo\_at\_pe\_NO\_rFact.in\_SPAM\_fo*)

*Date:* 05/25/04

Date: Tue, 25 May 2004 23:48:42 +0200

On Tue, 25 May 2004 02:43:53 GMT, Nick Landsberg wrote:

Hi Nick,

(snip)

>> *The setup I used for these tests is:*

>> *\* SQL Server 2000 SP3a, running as a service on my desktop computer, with*

>> *only one active application (Query Analyzer).*

>> *\* 1.3 GHz Athlon processor / 256 MB RAM.*

>

>*This would be underpowered for a commercial setup*

>*from a CPU standpoint and from a memory standpoint.*

Yes, indeed. But this computer is sitting right under my desk, with no other users besides me. And my normal use of SQL Server does definitely not involve navigating trees or producing what can best be described as the cross join of a 10000 row table with itself, with extra information based on rather complex logic.

(The 10000 row number I used here was my first "bigger" test, after some very small tests to verify that my code worked. I tried to cancel that query after three hours of execution time, causing my computer to crash; after that, SQL Server needed an additional 2 hours of recovery time to repair the damaged test database – that is when I decided to reduce the size of my tests <g>)

(snip)

> *I have no idea*

>*whether or not SQL-server would write to both*

>*data and log files for tempdb at the same time,*

>*but this may have increased the latency with many*

>*(almost) random seeks.*

I don't know either. I did notice that both the data and log files for the actual test database and the data file for tempdb grew considerably when I ran my first tests (before I preallocated ample space); tempdb's log file

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

grew little or not at all.

I don't think moving tempdb's log to my other HD would have helped. With only one active log file on that HD, there's never any need to reposition the disk arm.

(snip)

> *I am still wondering tho, about how much of that*  
> *elapsed time was disk I/O latency vs. CPU time. It depends on the*  
> *size of the in-memory cache (which I said above was small for*  
> *commercial implementations) and the effectiveness of the caching*  
> *algorithm, usually LRU.*

So am I. I tried running my test script again (for 1000 things and 2883 hierarchies), but with the options SET STATISTICS TIME and SET STATISTICS IO set to ON. (This time I didn't quit my other applications, instead I happily continued reading & writing in Agent, testing some simple queries in another database on the same server and checking some web pages).

The elapsed time reported by SET STATISTICS TIME is close to the elapsed time I calculated by comparing start and end time, but there should have been no difference at all (reported elapsed: 1106144 ms; my calculation says 18:47:413 or 1127413 ms – a 21 sec difference!). The cpu time reported by SET STATISTICS TIME is 939797 ms, so it looks as if there's little time lost while waiting.

I suspect the output of SET STATISTICS IO to be incorrect. I won't reproduce it here (it's quite long, listing stats for each query executed in the stored proc and per table used). The part I don't believe is that quite a lot of logical reads (from cache) are reported, but not a single physical read (from disk). Yet, after execution of the query the amount of space used in my test DB is almost 230 MB and tempdb (which I had shrunk earlier today) has grown back to just over 2000 MB! With only 256 MB present in my computers and many other applications running at the same time, I can't believe that this was done without any physical I/O.

I would have liked to supply accurate measurements, but it seems I'll now have to ask about this behaviour in one of the SQL Server newsgroups instead...

(snip rest of message)

Best, Hugo

--

(Remove \_NO\_ and \_SPAM\_ to get my e-mail address)