SQLskills.com
immerse yourself in sql server

(November 26th, 2012)

If you know someone who you think would benefit from being an Insider, feel free to forward this PDF to them so they can sign up here.

Quick Tips for our Insider friends!

Hey Insiders!

This bi-weekly Quick Tips is coming to you from Redmond but I wrote it while in Galesburg, IL where we spent Thanksgiving with Kimberly's Mom and our kids. It was good to get away from the abysmal Seattle weather for a few days and relax with some hearty mid-West cooking!

My latest Pluralsight online course is now available – 7.5 hours on SQL Server: Logging, Recovery, and the Transaction log – see here for details. Joe's latest course on SQL Server: Common Performance Issue Patterns has also been published – see here.

And, for those of you who have already taken one of our Immersion Events, make sure you reply to the email I sent to you on October 5th for your free 30-day Pluralsight code – there's no time limit for redeeming it.

Our super-early-bird pricing for the US and UK Immersion Events ends December 31st, so if you want to save $300 on your registration and use up that training budget in 2012, now's the time to talk to you manager!

The most recent book I've read is Neal Stephenson's (et al) *The Mongoliad: Book One*. This is the start of the Foreworld Saga that deals with Europe in the mid-Thirteenth Century. Genghis Khan is dead but his Mongol hordes continue to ravage Russia and Eastern Europe. The book follows two storylines – one set in the new Khagan's court in Karakorum and the other following a group of Knights on a quest to bring down the Mongol Empire. It's not particularly deep or complicated, but a satisfying historical romp nonetheless and I'm already half way through the second one.

Please let us know if you liked what you read/saw here and/or have any suggestions for future Quick Tips.

Note: you can get all the prior Insider newsletters here.

Paul's Ponderings

When performance is poor, the common reaction is that must be SQL Server itself that's causing the problem. And, of course, there could be a myriad of things that could be wrong – be it incorrect nonclustered indexes, plan cache bloat, fragmentation, parameter sniffing, and the list goes on.

Even when starting the troubleshooting process using a step-wise methodology like waits-and-queues, the focus is generally to find out why SQL Server is waiting.

Unfortunately, it's not typical to focus on the application and ask whether the application is the one at fault. The problem could be that you're asking for too much data and driving a very expensive workload to SQL Server. SQL Server is only reacting, and it can't keep up with the poorly designed load.

To help you think outside the box, here are some things to consider about the way your application queries the data in SQL Server. These could be adding unnecessary load and negatively stressing CPU, memory, and/or I/O:

- **Processing:** For the data that's being pulled from SQL Server, is the application processing the data one row at a time (commonly called RBAR, or row-by-agonizing-row, processing)? If so, this means that SQL Server has a thread waiting for the application to acknowledge the data sent to it, and can lead to high ASYNC_NETWORK_IO waits (yes, this wait typically means an application RBAR issue, not a network problem). The application would be better to cache the incoming data locally, and reply back to SQL Server as soon as possible that it has the data.
- **Filtering:** Is the application going to filter the data locally before using it or displaying it? In that case, it should be waaaay more efficient to push the predicate down to SQL Server and have the minimum possible data returned to the application. SQL Server is very, very good at filtering data, given the right nonclustered indexes to support the filter predicates.
- **OSFA:** Are all the table columns being returned absolutely necessary? Are you trying to build a "one size fits all" dialog? By using a targeted SELECT list rather than just SELECT *, this cuts down on the data being processed and returned. And, with fewer columns requested, SQL Server might have more optimal ways of getting to this data – also improving performance.
- **Ordering:** Does the data being returned really need to be sorted with an ORDER BY? If not, this might cut out a sort operation. Often sort operations can be expensive as they may end up requiring a costly sort-spill to tempdb.
- **Just in case:** Can the SELECT be postponed until it's really required? If an application is issuing a SELECT *just in case* the user clicks an application button, then it might be wasted processed. It's better to wait until the button is actually pushed before issuing the SELECT, removing all of the processing when the button is not pushed.
- **Consider caching:** If the same data is being queried again and again, cache it locally and only issue a new SELECT when the data changes. This is ideal when data does not change frequently or if up-to-the-minute data is not required.

These are just a few things to think about when analyzing how an application uses SQL Server. Making some of these changes can have a profound effect on the amount of work SQL Server has to do, especially if a single change in the application query logic is multiplied by hundreds or thousands of instances of the application running simultaneously.

Call to action: If your SQL Server is getting stressed from excessive application queries pulling large amounts of data, instead of first trying to tune those queries, go to your developers and ask them whether each query is really necessary, and whether it has to pull back as much data as it is. And, if you're the developer; ask yourself what you can do to streamline your application's requests to SQL Server. You'll be amazed at the difference you can make!

I'm curious to hear your thoughts about limiting the data an application pulls from SQL server, so please feel free to drop me a line, treated confidentially of course.

Video Demo

In this video, Glenn describes the differences in Task Manager between Windows Server 2008 R2 and Windows Server 2012, focusing on the extra information that is available in Windows Server 2012. There is a lot of useful hardware information available in Windows Server Task Manager 2012 that was previously only available using other utilities such as CPU-Z or msinfo32. This makes it much easier to quickly understand your hardware capabilities and current configuration.

The video is just under 5 minutes long and I produced the video in WMV and MOV formats so everyone can watch. You can get the videos:

- For WMV: here
- For MOV: here

No demo code this time.

Enjoy!

SQLskills Offerings

All of our 2013 public classes are now open for registration! Based on requests from people, attendee ratings of the hotels we used this year, and the ease of using hotels we know, we're using the same locations again. This means we cover both sides of the US, central US, and Europe.

Please know that these classes are final as the hotel contracts are signed, and the classes will not be cancelled or moved for any reason, nor will the dates change.

- February 4-8, 2013: Internals and Performance (**IE1**) in Tampa, FL – USA
- February 11-15, 2013: Performance Tuning (**IE2**) in Tampa, FL – USA
- April 29-May 3, 2013: Internals and Performance (**IE1**) in Chicago, IL – USA

- April 29-May 3, 2013: Immersion Event for Business Intelligence (**IEBI**) in Chicago, IL – USA (co-located but in a different training room. Attendance is for one event or the other; these cannot be combined for one attendee where they move back/forth.)
- May 6-10, 2013: Performance Tuning (**IE2**) in Chicago, IL – USA
- May 13-17, 2013: High Availability & Disaster Recovery (**IE3**) in Chicago, IL – USA
- May 13-17, 2013: Immersion Event for Developers (**IEDev**) in Chicago, IL – USA (co-located but in a different training room. Attendance is for one event or the other; these cannot be combined for one attendee where they move back/forth.)
- May 20-24, 2013: Development Support (**IE4**) in Chicago, IL – USA
- June 3-7, 2013: Internals and Performance (**IE1**) in London – UK
- June 10-14, 2013: Performance Tuning (**IE2**) in London – UK
- June 17-21, 2013: High Availability & Disaster Recovery (**IE3**) in London – UK
- June 24-28, 2013: Development Support (**IE4**) in London – UK
- September 16-20, 2013: Internals and Performance (**IE1**) in Bellevue, WA – USA
- September 23-27, 2013: Performance Tuning (**IE2**) in Bellevue, WA – USA

One thing to note is that the course prices have increased slightly for 2013, reflecting increasing food, logistics, travel, and accommodation costs. We kept our prices the same for the last three years but now we have to raise them a little.

For US classes, the new early-bird price is US$3,295 and the full-price is US$3,795. However, for all registrations received before January 1, 2013, and for all past attendees in the 12 months prior to registration, we will only charge the 2012 early bird price of US$2,995 (the super-early-bird price). **Be sure to get your registrations in early!**

For UK classes, the new early-bird price is US$3,795 and the full-price is US$4,295. There is a similar super-early-bird and past-attendee price equal to the 2012 UK early bird price of US$3,495. **Again, be sure to get your registrations in early!**

See here for the main Immersion Event Calendar page that allows you to drill through to each class for more details and registration links.

So, that's it for now. We hope to see you soon!

Summary

We hope you've enjoyed this issue - we really enjoy putting these together.

If there is anything else you're interested in, we'd love to hear from you - drop us a line.

Thanks,
Paul and Kimberly

Paul@SQLskills.com and Kimberly@SQLskills.com