(August 17th, 2015)

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 newsletter is coming to you from Redmond, where we're packing to leave on Wednesday (arriving Thursday) for the U.K. to get adjusted to the time zone before our London IEPTO1 class next week.

We've got a busy Fall with our classes running in London, Dublin, and Chicago and another great week at SQLintersection (in Las Vegas this October) as well as PASS – that same week. We might further tweak our line-up for SQLintersection but we've already posted a lot of the great sessions that will be presented; it's going to be a fantastic week for database architects and developers! If you're going to PASS, consider sending some of your team (primarily developers) to SQLintersection – then, everyone can return and cross train each other on what you've learned and you all win! Check out more information online at www.SQLintersection.com and don't forget to register with the discount code of SQLskills to save $50 off your registration.

The latest book I've read is Daniel Silva's *The Kill Artist*. Last week I decided to try some contemporary thrillers by authors I hadn't read so ordered this book and *The Lions of Lucerne* by Brad Thor – picked because they're both long series dealing with the same protagonist. Silva's book was excellent (haven't read Thor's book yet)! It deals with an Israeli secret service agent tracking down a Palestinian terrorist, with plenty of action, intrigue, and good characters. Reminds me of old Craig Thomas spy thrillers like *Jade Tiger* and *Firefox* I used to read when I was a teenager (from just looking on Wikipedia I see he died back in 2011 – how sad). Highly recommended and I'm glad I took the chance on him - the next six books in the series already arrived ☺!

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

Paul's Ponderings

One of the email questions I received last week was on something I've seen a few times before and I thought it would make a good editorial for this week. In a nutshell, the question is: *When our log shipping secondary is applying log backups, sometimes it takes a lot longer than usual. Any ideas why this might be the case?*

Log shipping has been around forever, and it's still a hugely applicable and useful feature for very simply maintaining one or more secondary copies of a database. You can also use a secondary copy for reporting, where the restore of the log backup uses the WITH STANDBY option, leaving the database in an accessible, but read-only state (when the logs aren't being applied).

This works as follows:

1. Make sure all users are disconnected from the database
2. Write all the log records from the backup into the database's log file
3. Perform the REDO part of recovery (ensuring that all operations from committed transactions are present in the database)
4. Perform the UNDO part of recovery (ensuring that all operations from uncommitted transactions are not present in the database)

Step 4 writes all the log records generated by the UNDO operations into a special file called the undo file. This means that the database is in read-only mode and is transactionally-consistent so that users can access it. The reason the log records are written into the undo file is so that the transaction log of the database is not altered in any way, allowing subsequent log backups to be restored. If this weren't the case, the UNDO log records would advance the database's LSN, meaning that subsequent log backup restore operations would fail.

When the restore process begins on the secondary database, if an undo file exists, there is another step that is performed before steps 2-4 above. This additional step needs to take all the log records in the undo file and undo the effects of them – essentially putting the database back into the state as of the end of step 3. This database state is the same as if the previous log backup had been restored using *WITH NORECOVERY* instead of *WITH STANDBY*.

The occasional long-running restore problem happens when a log backup is restored that contains a long-running transaction that does not commit before the end of the log backup. This means that it must be completely undone as part of restoring the log backup (step 4), resulting in a very large undo file. This in itself can make restoring a log backup take a lot longer than usual.

When the next log backup is restored, the additional step that undoes all the log records in the undo file has a very large undo file to process and takes much, much longer than usual. And if the log backup being restored *also* has an uncommitted, long-running transaction then it's the perfect storm as the step 4 will also take a long time.

The situation where I've seen this is when the primary database is undergoing index maintenance and a log backup finishes near the end of a very long-running index rebuild operation of a large clustered index. The initial restore of that log backup on the secondary database takes much longer than usual to complete because of step 4 in the restore process. The next log backup on the primary also completes just before an index rebuild completes. When it is restored on the secondary, the whole of the large undo file has to be undone again, then the log restore occurs, and then another large undo file is generated to undo the second uncommitted index rebuild.

This is a possibility you have to be aware of if the secondary database must be available 24x7 for reporting, with only minimal downtime when each log backup is restored. In that case I would carefully augment the index maintenance operations on the primary with log backups to ensure that only complete, committed index rebuilds are present in the log backups being restored on the secondary database.

An alternative would be to move from log shipping to database mirroring or availability groups, where the log records are continually being sent from the principal to the mirror database (or primary to secondary replica databases, in availability group terms) and there are no extra steps involving undoing log operations multiple times.

With database mirroring, the drawback of this is that reporting would have to use database snapshots, so there's a complexity trade-off involved. With availability groups, the reporting would have to use a readable secondary, which can lead to index fragmentation on the primary replica (which I talked about in the newsletter last October), but that can be surmounted.

Call to Action

Not really a call to action this time, but this is yet another example where understanding how SQL Server performs common operations can make it much easier to diagnose performance problems.

Video Demo

In the demo this time, Tim demonstrates the new SQL Server 2016 feature called Stretch Database. This gives you the ability to store portions of a database in the Microsoft Azure Cloud, specifically into an Azure SQL Database. This is pretty cool since Stretch Database leverages the processing power in Azure to run queries against remote data by rewriting the query. Currently in SQL Server 2016 CTP2 you can only use Stretch Database on entire tables and there are a number of limitations, you can read about those here.

The video is about 7.5 minutes long and you can get it:

- In WMV format here.
- In MOV format here.

And the demo code is here.

Enjoy!

SQLskills Offerings

We've released all of our classes for 2015, listed below. We'll release the first portion of our 2016 schedule in September.

To help your boss understand the importance of focused, technical training, we've also added a few items to help you justify spending your training dollars with us:

- Letter to your boss explaining why SQLskills training is worthwhile
- Community blog posts about our classes
- Immersion Event FAQ

Upcoming Immersion Events

London, UK

- August 24-28, 2015: **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1 (formerly IE1) **SOLD OUT!**

Dublin, Ireland

- October 12-16, 2015: **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2 (formerly IE2) **SOLD OUT!**

Chicago, IL

- November 16-18, 2015: **IE0**: Immersion Event for the Accidental/Junior DBA
- November 16-19, 2015: **IEPDS**: Immersion Event on Practical Data Science
- November 16-20, 2015: **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1 (formerly IE1) **Over 70% booked already; register early to avoid disappointment!**

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

SQLintersection

We're looking forward to another great SQLintersection event! This Fall SQLintersection is running the week of October 26, 2015 back in Las Vegas. Yes, it's the same week as PASS. No, that's not really a problem! We don't have a lot of folks that attend both PASS and SQLintersection so the overlapping week isn't a big problem. And, it's a great way to get different perspectives on the tech and have everyone come back and do knowledge-transfer.

Check it out online at www.SQLintersection.com. When you register, don't forget to use the discount code "SQLskills" (without the quotes and it isn't case-sensitive) and you can save $50 off registration. We hope to see you there!

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