(July 17th, 2017)

If you know someone who 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,

As this hits your inboxes I'm putting the finishing touches on Kimberly's next Pluralsight course, *SQL Server: Indexing for Performance* – should be available in early August. And of course, we're enjoying the excellent summer weather here in the Pacific Northwest!

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

SQLskills News

Only a couple of weeks to go before our classes in Bellevue, WA and we have a few seats remaining in the IEPTO1 and IEPTO2 classes.

We've released our October line-up of classes in Chicago, including IE0, IEPTO1, IEPTO2, IEPS, IESSIS1, plus THREE new courses. We're debuting a **new two-day class on Azure SQL Database and Azure VMs**, a **new three-day class on Upgrading SQL Server**, and a **new two-day class on Clustering and Availability Groups**. See here for the complete 2017 SQLskills Immersion Event class schedule.

Our team is presenting a number of workshops at Fall conferences, we hope you'll join us:

- Paul will be presenting a workshop at the Fall SQLintersection conference in Las Vegas on Sunday, October 29th, titled *Performance Troubleshooting using Waits and Latches*. Check out the details here.
- Erin will be presenting a workshop at the Fall PASS Summit in Seattle on Monday, October 30th, titled *Solving Common Performance Problems Using Query Store*. Check out the details here.
- Jonathan will be presenting a workshop at the Fall SQLintersection conference in Las Vegas on Monday, October 30th, titled *Extended Events: WTF or FTW!* Check out the details here.
- Kimberly will be presenting a workshop at the Fall SQLintersection conference in Las Vegas on Friday, November 3rd, titled *Very Large Tables: Optimizing Performance and Availability through Partitioning*. Check out the details here.
- Tim will be presenting a workshop at the Fall SQLintersection conference in Las Vegas on Friday, November 3rd, titled *Common SQL Server Mistakes and How to Correct Them*. Check out the details here.

Don’t forget to check out our SQL101 posts… hopefully they’ll help refresh or reinforce topics for even the more seasoned DBAs in the community. The blog posts are automatically collected here.

Finally, even if you can’t join us in person, we’ve renewed our call for remote user group sessions for the second half of this year. We have almost 80 scheduled and completed so far; if you’d like one of us to present for your user group, check out my blog post here.

Book Review

The latest book I’ve read is David Sax’s *The Revenge of Analog: Real Things and Why They Matter*. What a great book! As a confirmed Luddite, I was really looking forward to the various parts of the book. It covers the resurgence of vinyl records, paper and magazines (check out Stack and Delayed Gratification, both of which I subscribed too), photo film, board games, books, bricks-and-mortar retail, and analog ideas around work, education, and online companies. It's quite clear that going all-digital isn't what people want. Very interesting, not written in a preachy or manifesto way, and highly recommended!

The Curious Case of…

This section of the newsletter explains problems we’ve found on client systems; they might be something you’re experiencing too.

Jonathan was working with a long-term client recently where they wanted him to help reduce CPU usage and application page loading times. Most of the common queries had already been tuned to sub-second execution times and making new gains wasn’t going to be easy.

Jonathan then noticed several views that are used very frequently that each perform a union of three (different) sets of rarely-changing data and were each taking around 300ms of CPU time for every page load.

Instead of using these views, Jonathan persisted the results of these unions in properly-indexed tables, and the code was also changed to truncate and repopulate one of the view-replacement tables when the underlying data changed (at most once per month) to maintain the correctness of the persisted union.

By making these changes and eliminating the CPU usage for materializing the view results, the system was about to double the throughput of page load requests!

Bottom line: as with all queries, limiting the amount of data that has to be processed is a key to reducing CPU usage and execution time. Sometimes a non-obvious solution such as I described above can have a tremendous effect on a workload.

Paul's Ponderings

One of the concepts I find people misunderstand frequently is the *recovery interval*, either for the server as a whole or the per-database setting that was introduced in SQL Server 2012 for indirect checkpoints. This came up again last week so I thought it would make a good topic for the newsletter.

There are two misconceptions here:

1. The recovery interval equals how often a checkpoint will occur
2. SQL Server guarantees the recovery interval

A bit of background: crash recovery has two tasks to perform: replaying log records from committed transactions (called REDO) and removing the effect of log records from uncommitted transactions (called UNDO). REDO only has to occur if there have been committed transactions where the changed data pages have not been written to disk (which is done by periodic checkpoints or a lazy writer thread if there's memory pressure on the buffer pool).

The recovery interval specifies an ideal upper bound on how long the REDO portion of crash recovery will take. The length of time REDO takes depends on how many log records need to be replayed from committed transactions. To help REDO stay on track, the recovery interval setting forces a checkpoint to occur when a certain number of log records have been generated (a number calculated using the chosen recovery interval).

So although recovery interval *does* control how often a checkpoint occurs, the recovery interval time does not equal the time between checkpoints.

For instance, if the recovery interval is set to one minute (the default), and a checkpoint occurs *now*, when will the next checkpoint occur? If activity in the database is very infrequent, a checkpoint may not occur for a very long time because the calculated recovery time for that activity will be less than one minute. If activity in the database is extremely high, checkpoints may occur every few seconds. Checkpoint frequency entirely depends on how fast log records are being generated *and* the recovery interval setting.

And of course SQL Server cannot *guarantee* the recovery interval. Imagine the recovery interval is set to one minute, and I start a transaction that updates millions of records over several hours. If SQL Server crashes just before the transaction commits, how long do you think it will take to roll back the millions of updates? Certainly it will take much longer than one minute – the configured recovery interval.

So I can update my definition to be: the recovery interval is therefore the ideal upper bound on how long the REDO portion of crash recovery will take, assuming no long-running transactions.

Hope this helps clear up any misconceptions.

Glenn's Tech Insights

This section of the newsletter highlights recent news and views from the hardware and Windows worlds that we think will be interesting to SQL Server community members.

Intel Xeon Scalable Processor Family

On July 11, Intel formally launched the Intel Xeon Scalable Processor Family (Skylake-SP), for two, four, and eight-socket servers. These new 14nm processors replace the previous 14nm Intel Xeon E5 v4 and E7 v4 (Broadwell-EP and Broadwell-EX) families.

Intel has also changed their naming convention for these new processors. Instead of the familiar Xeon E5 and Xeon E7 lines, we now have Xeon Platinum, Xeon Gold, Xeon Silver, and Xeon Bronze lines. Within those new lines we have a pretty confusing mix of different SKUs and product naming, with new model suffixes, such as “M” (for 1.5TB of RAM per socket), “T” (for extra thermal limit/high reliability), and “F” (for Fabric).

This new family of processors is not electrically or physically compatible with the previous Broadwell processors, so they require new model servers from the server vendors. One place where there is less confusion is that you can use ***any*** of these processors in ***any*** new two, four, or eight-socket server. All you have to do is choose the right processor for your needs and budget.

For SQL Server usage, you should only be looking at the Xeon Platinum and Xeon Gold lines. The Xeon Silver and Bronze lines are artificially crippled for product differentiation purposes, and Microsoft charges the same per core license cost regardless of the underlying performance of the processor.

As with the previous Intel Xeon E5 v4 and Xeon E7 v4 families, Intel offers several different SKUs for each physical core count, starting at four physical cores and going up to twenty-eight physical cores. At each physical core count, there are one (or sometimes more than one) preferred SKU that gives the best performance for different SQL Server workloads. I will soon have a list of “preferred” processor SKUs for SQL Server.

If you are going to be using SQL Server 2016 Standard Edition, you need to keep in mind the artificially low, per-instance license limits for physical cores. SQL Server 2016 Standard Edition is limited to 24 physical cores per instance.

If you buy a new machine with more than 24 physical cores, SQL Server Standard Edition will only use the first 24 physical cores that it finds, regardless of whether they are balanced across the NUMA nodes.

To make matters worse, Microsoft will still expect you to pay for **all** of the physical cores in the machine, even though a single instance of SQL Server 2016 Standard Edition won’t let you use all of them (if you have more than 24).

So far, Microsoft has not announced any plans to raise this per-instance core limit for SQL Server 2017 Standard Edition. This is something you need to keep in mind if you are going to be buying a new server!

#TBT

(Turn Back Time…) This section of the newsletter highlights some older resources we've referred to recently that you may find useful, plus select blog posts we've published since the previous newsletter.

As I was writing up the Curious Case section, it reminded me of four Pluralsight courses that Joe Sack recorded that are all about identifying and fixing esoteric query plan performance issues, so these are what I'm highlighting this time:

- Joe's course: *SQL Server: Common Performance Issue Patterns*
- Joe's course: *SQL Server: Common Query Tuning Problems and Solutions – Part 1*
- Joe's course: *SQL Server: Common Query Tuning Problems and Solutions – Part 2*
- Joe's course: *SQL Server: Troubleshooting Query Plan Quality Issues*

Here are a few of the blog posts we've published since the last newsletter:

- Paul: *SQLintersection Fall 2017*
- Paul: *SQLskills SQL101: Practicing disaster recovery*
- Glenn: *SQL101: AMD EYPC 7000 Series Processors*
- Glenn: *SQL Server Diagnostic Information Queries for July 2017*
- Erin: *Migrating to In-Memory OLTP…what's stopping you?*

I hope you find these useful and interesting!

Video Demo

With Memory Optimized tables you have two options for indexes: hash and nonclustered. It's easy to think that because all the data is in memory, the index choice doesn't matter, but it does! In this video, Erin walks through a simple example of why you need to think about the queries that access your data in-memory in order to choose the proper index.

The video is around 16 minutes long and you can get it:

- In WMV format here
- In MOV format here

And the demo code is here.

Enjoy!

Upcoming Immersion Events

All 2017 classes are available for registration!

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

Bellevue, WA, July/August 2017

- **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1
  - July 31-August 4
- **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2
  - August 7-11

Chicago, IL, October 2017

- **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1
  - October 2-6
- **IE0**: Immersion Event for the Junior/Accidental DBA
  - October 2-4
- **IECAG**: Immersion Event on Clustering and Availability Groups
  - October 5-6 **** New class! ****
- **IESSIS1**: Immersion Event on Learning SQL Server Integration Services
  - October 2-6
- **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2
  - October 9-13
- **IEPS**: Immersion Event on PowerShell for SQL Server DBAs
  - October 9-11
- **IEAzure**: Immersion Event on Azure SQL Database and Azure VMs
  - October 9-10 **** New class! ****
- **IEUpgrade**: Immersion Event on Upgrading SQL Server
  - October 11-13 **** New class! ****

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

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