(July 31st, 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 we're just starting IEPTO1, the first of two weeks of classes in Bellevue, WA. There are still a few seats available in IEPTO2 next week…

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

SQLskills News

We've published a new Pluralsight course! Glenn's latest course is *SQL Server: Upgrading and Migrating to SQL Server 2016*. Check it out here.

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.

And all of us are presenting sessions at SQLintersection or the PASS Summit (Erin and Glenn). You can read more about our SQLintersection lineup in my blog post 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 85 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

I haven't finished any books since the previous newsletter so here's a review of a book from 2013 that I haven't told you about so far: Abraham Verghese's *Cutting for Stone*, which was one of the contenders for my best book of 2013. Wonderful book! It's hard to describe the plot without giving away what happens in the book, but it's written by a surgeon, with lots of factually accurate descriptions of surgeries and medical issues. The book follows the life of a family in Addis Ababa in the '60s and '70s, with drama and betrayals. Masterfully written, became a two-day page-turner for me after dithering about buying it since it came out. Very 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.

I received a question in email last week from someone who was monitoring wait statistics for a database and saw a dramatic change in them after enabling Query Store.

From the pattern of waits, it looked like the database was pretty static, with only SELECT queries occurring. After Query Store was enabled, there was a huge increase in *HADR_SYNC_COMMIT* and various backup waits.

The reason for this is that when Query Store is enabled, it starts capturing metrics about the executing queries and storing those metrics in the database. That means there are inserts occurring so those inserts need to be logged, so logging waits start to occur. The *HADR_SYNC_COMMIT* waits are essentially logging waits occurring for synchronous availability group replicas, and the backup waits are because now there are log records that need to be backed up by regular log backups – so the backups have work to do

Bottom line: when you enable Query Store, there is some small overhead from doing so. Sometimes this may not be noticeable – it depends on your workload and how aggressive you configure Query Store to be about capturing query performance metrics.

Paul's Ponderings

The Curious Case topic this time gave me inspiration for the Ponderings. There are a few features that can have unexpected and/or detrimental side effects when they're enabled so below I present a list of those that spring to mind.

Versioning

When you enable versioning in a database (either read-committed isolation using row-versioning [*read_committed_snapshot*] or snapshot isolation [*allow_snapshot_isolation*]), any query that changes any table in that database causes record versions to be generated (the pre-change image of the record). These version records are stored in the version store in tempdb and are kept around as long a statement or batch is running that might need to see that version of the record.

This means that using versioning causes space to be used in tempdb, write activity in tempdb, and read activity in tempdb (from queries reading the pre-change versions). These side-effects are pretty well known.

The side-effect that isn't as well known is that when a record version is generated, the record being changed has to grow by 14 bytes to accommodate the versioning 'tag' that's added to the end of it (which has the versioning timestamp of the last change to the record and a pointer into the version store to the previous version of the record). If there isn't enough space on the page for the extra 14 bytes, and the page is from an index, the page has to split, causing index fragmentation (extra logging, latching, and the possible fragmentation effects themselves) which you'll need to address with fill factors and regular index maintenance. I go into a bit more depth on this here.

Availability Group Readable Secondaries

When a readable secondary is created for a database, queries against it use versioning under the covers. In a nutshell, this means that on the primary replica, any queries that change the database create empty versioning tags on the records being changed (14 bytes long, as described above), so that the 14 bytes of space is there on the readable secondary to accommodate the actual versioning information.

This means that having a readable secondary can cause index fragmentation on the primary replica, which you'll need to address with fill factors and regular index maintenance. I go into a bit more depth on this here.

Change Data Capture and Change Tracking

Change Data Capture, or CDC, uses the transactional replication Log Reader Agent job to regularly scan the transaction log of the database to harvest changes from committed transactions, and then writes them into a capture instance (i.e. table). This means it can cause log file contention and extra logging and space usage from the captured data being stored.

Change Tracking makes note of the changes in real time, so only incurs extra logging and space usage.

Transparent Data Encryption

Transparent Data Encryption, or TDE, encrypts the database data and log files on disk, and prevents instant file initialization from being used, which can lead to workload delays when a data file grows if the autogrowth setting it set very high.

TDE of any database also causes tempdb to be encrypted, which prevents instant file initialization for tempdb data files too, with consequent possible delays.

Extended Event Sessions and Traces

As with any trace or Extended Event session, including an event that is very frequent (e.g. tracking lock acquires and lock releases, or transaction log records being generated) can sometimes cause a noticeable degradation in performance.

Also, setting up a trace that goes to a table instead of a file, or using a very expensive event in an Extended Event session (like the *query_post_execution_showplan* event) can really cripple performance. Jonathan goes into depth on that problem here.

Trace Flags

This one's pretty obvious. Trace flags, by their very nature, exist to change the behavior of SQL Server in some way and so they all have an effect. Be very careful to only use trace flags for which you understand the consequence and why you're enabling it for your server.

3rd-Party Monitoring Tools

Although these aren't part of SQL Server per se, 3rd-party performance monitoring tools can have a detrimental effect on performance if they're not coded or configured correctly. We've seen plenty of cases over the years of tools creating tempdb problems, not showing their own high-cost queries, enabling costly trace flags, and other poorly-considered behaviors. Always make sure that you investigate the cost of enabling a tool. (I know that SentryOne – whose tools haven't done any of the things mentioned above – published a detailed analysis back in 2011 of the effect of their tools – see here – I don't know about other vendors.)

Another issue with any tool that uses DMVs extensively is that you'll see a huge number of *OLEDB* waits occurring, for a few tenths of a millisecond each. This isn't a problem, but will skew the results of wait statistics analysis.

Call to action: Any time you're going to enable a new feature, enable some performance monitoring code or tool, or turn on a trace flag, it really pays to make sure you understand whether what you're about to do could lead to a performance problem or other unexpected consequence. The last thing you want to do is cause a problem inadvertently.

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.

AMD Ryzen Threadripper Processors

AMD has announced that their new HEDT Ryzen Threadripper processors will launch on August 10, 2017. Initially, there will be two models in this line. The top of the line model is the Ryzen Threadripper 1950X, which has 16 cores and 32 threads, a base clock speed of 3.4GHz, and will cost $999.00. The other model is the Ryzen Threadripper 1920X, which has 12 cores and 24 threads, a base clock speed of 3.5GHz and will cost $799.00.

Both models will have 32MB of L3 cache, and will have 64 PCIe 3.0 lanes available on the processor, with 60 of those lanes available for things like discrete graphics cards or PCIe NVMe storage devices. Both models will also have a boost speed of 4.0 GHz.

These processors require X399-based motherboards, which will have eight DDR4 memory slots, so you will be able to have 128GB of RAM with commodity 16GB DDR4 DIMMs. The processor itself will support up to 1TB of RAM when 128GB LR-DIMMs (Load Reduced DIMMs) are used. The first round of X399 motherboards are due to be announced on July 25, 2017.

This processor family is relevant for a Data Professional who might want to have a relatively affordable (especially compared to a competing Intel HEDT processor, which costs about twice as much) workstation to run large SQL Server workloads or multiple concurrent decent sized VMs on a desktop development and testing machine without being as quickly constrained by I/O, memory or processor core count limitations.

AMD EPYC Memory Latency and Bandwidth Tests

ServeTheHome.com has done some interesting testing of memory latency and memory bandwidth performance on a two-socket, Supermicro Ultra EPYC Server (SYS-2023US) with two, AMD EPYC 7601 32-core processors and 256GB of either DDR4-2400 or DDR4-2666 RAM. They did their testing with the Intel Memory Latency Checker tool.

Their testing showed an 8% decrease in latency and a 9% increase in memory bandwidth when using DDR4-2666 compared to DDR4-2400 memory. They also saw a 5% increase in application performance using a Linux kernel compile benchmark. There were also large performance improvements with the latest firmware for the system compared to the initial firmware.

This means that you should be careful to make sure that you have the latest BIOS and other firmware updates for an AMD EPYC system, and you should also make sure to get DDR4-2666 memory rather than DDR4-2400 memory for an AMD EPYC system, since the AMD Infinity Fabric is so sensitive to memory performance.

NVM Express Releases 1.3 Specification

On June 27, 2017 NVM Express, Inc. released its NVMe 1.3 specification, which has three major new features. These include virtualization enhancements, a feature called Sanitize (for secure erase), and a directive called Streams, which reduces write amplification (to improve endurance for NAND-based SSDs). New products that use this latest NVMe specification should start to show up over the next six to twelve months.

Not All Flash Storage is Created Equal

A common misconception I still see is the general idea that any type of NAND-based flash storage is pretty much the same in terms of performance, consistency, and endurance. This is actually pretty far from the truth. There are several broad categories of NAND-based flash storage with different price points and different performance and endurance characteristics.

Consumer-grade flash storage often has higher peak performance than Enterprise flash storage, but usually performs worse under a heavy workload, has less performance consistency, has less write endurance, and lacks power-loss protection.

Within Enterprise flash storage products, there are also several broad categories of products for different types of workloads. One category are SATA interface, read-intensive products, which generally have less overprovisioning and different firmware to offer good read performance and much lower costs per gigabyte of storage.

Storage Review has a good round-up of Enterprise read-intensive SATA SSDs, which shows some pretty dramatic differences in performance between five different SATA SSDs. Part of their testing used two 64GB VMs running SQL Server 2014 using Dell’s Benchmark Factory for Databases to run a TPC-C workload to measure transactions per second and average latency. The best performer was the 1.92TB Toshiba HK4R, while the worst performer (by far) was the 1.6TB Intel DC S3520.

#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.

We've collectively been doing a bunch of client work recently around stored procedure performance, so that's what I'm highlighting this time:

- Kimberly's course: *SQL Server: Optimizing Stored Procedure Performance*
- Kimberly's course: *SQL Server: Optimizing Stored Procedure Performance – Part 2*
- Kimberly's recent post: *SQLskills SQL101: Stored Procedures*
- Kimberly's *Optimizing Procedural Code* blog post category

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

- Paul: *SQLskills SQL101: Why DBCC CHECKDB can miss memory corruption*
- Paul: *New course: Upgrading and Migrating to SQL Server 2016*
- Glenn: *Upgrading to SQL Server 2016*
- Glenn: *CPU-Z 1.80 is Available*
- Glenn: *SQLskills SQL101: Creating SQL Server Databases*
- Tim: *SQLskills SQL101: Validating SQL Server Backups*
- Erin: *SQLskills 101: The Other Bad Thing About Clearing Procedure Cache*
- Jonathan: *Downgrading SQL Server Editions*

I hope you find these useful and interesting!

Video Demo

In the demo video this time, Glenn demonstrates the new SQL Server Diagnostics add-in for SSMS that you can use to upload and analyze SQL Server dump files. This tool is also useful for checking what hotfixes are included in each SQL Server Cumulative Update. The links shown in the video are as follows:

- SQL Server Diagnostics (Preview)
- SQL Server Management Studio extension for Diagnostics

The video is around 5 minutes long and you can get it:

- In WMV format here
- In MOV format 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

- **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