(May 14th, 2018)

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,

We've just arrived back in Redmond after running our Spring classes in Chicago (10 classes over 3 weeks!). Now we're concentrating on building more online content until our next in-person class in Bellevue, WA in June.

And talking of online content, next week we're debuting a new, live, online class! Erin will be presenting our new ***Online IEQS: Immersion Event on Solving Common Performance Problems with Query Store***. This will be delivered live via WebEx on May 22-24 (roughly 12-13 hours of content including open Q&As; similar to two full workshop days without leaving the comfort of your home/office!). It's priced at only US$795, but if you **email me before you register, I'll give you a discount**. **Move fast to claim your seat!** See here for all the details and here for some cool customer problems we've solved recently using Query Store.

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

SQLskills News

In-person US classes: Our single remaining in-person class for 2018 is IEPTO1 in Bellevue, WA in June. **We don't have any additional classes planned in the US in 2018;** see here for our 2018 Immersion Event class schedule.

In-person London classes: We're bringing four of our Immersion Events to London in September: IEPTO1 and IEPTO2, plus our new classes: **IEAzure** (on Azure, Azure VMs, and azure Managed Instance) and **IECAG** (on clustering and availability groups). See here for details.

Finally, even if you can't join us in person, I've put out a call for 2018 remote user group sessions. In 2017, we did more than 100 of these around the world and we have set up more than 60 for 2018 already! If you'd like one of us to present for your user group, check out my blog post here. **Note: Tim has a new user group session on Azure Managed Instance** that he's happy to present to your group – see here for details.

Book Review

I haven't finished any books since the last newsletter, so here's a review of a book I read back in January: Kazuo Ishiguro's *The Remains of the Day*. Wow - what a wonderful book! I picked this

up as Ishiguro won the 2017 Nobel Prize for Literature and this particular book of his won the Booker Prize in 1989. The book is a series of reminiscences by Mr. Stevens, the perfect butler, of his time working for Lord Darlington between the two World Wars as he takes a week-long trip into the English countryside in the late 1950s. There are really three stories: Stevens himself and his misguided, blinkered trust in his master; Lord Darlington's involvement and machinations with Germany; and Stevens' relationship with the housekeeper, Miss Kenton. It's beautifully written, and Ishiguro captures the essence of the anachronistic, stiff-upper lip butler to perfection. *Highly* recommended. I have no doubt this will be in my top-10 for 2018.

The Curious Case of…

This section of the newsletter explains recent problems we've helped with on client systems; they might be something you're experiencing too.

Jonathan was working with a new client recently who was having problems understanding why their queries could have hugely variable performance. Due to the architecture of their database, nearly all queries resulted in a query plan that used a clustered index seek.

They were under the impression that a clustered index seek is always low cost, as it will only return a single row (as it's not a scan).

That is incorrect. A seek can easily return multiple rows. In their case, the clustered index was on an ID that was the filter predicate for most of their queries, and the number of rows per ID could vary wildly. This meant that for some IDs the clustered index seek was only returning a few rows but for others was returning hundreds of thousands of rows.

The overall fix for this kind of variability depends on the situation and is beyond the scope of this section, but the *Curious Case* this time is in recognizing that "it's not a scan, so it must be efficient" is wrong.

Bottom line: When you encounter seek operations, don't assume that they're only returning a small number of rows. You always need to look at two things: the query plan AND the output from *SET STATISTICS IO ON* so you can get better insight into what that seek is *really* doing behind the scenes.

Paul's Ponderings

When performance is poor, a common reaction is to blame "the database;" it's got to 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 indexing strategies, poor coding practices, statistics and/or cardinality estimation issues, index fragmentation, parameter sniffing, improper configurations, poor database design – and, the list goes on and on and on.

Even when starting the troubleshooting process, it's common to use a step-wise methodology that begins with wait statistics analysis; the focus is to find out where (and ultimately, why) SQL Server is waiting. And, while this is usually a good strategy, it's not always SQL Server that's the problem.

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

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 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. The application would be better to cache the incoming data locally, reply back to SQL Server as soon as possible that it has the data, and then they both continue processing.
- **Selections:** 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 up to SQL Server (called a ***selection***) and have the minimum possible data returned to the application. SQL Server is very, very good at filtering data, given the right indexes to support the filter predicates. It's generally a very good idea to reduce the amount of data sent to the client!
- **Projections:** 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 (called: a ***projection***) rather than just *SELECT* *, this cuts down on the data being processed and returned. And, with fewer columns requested, SQL Server often has more optimal ways of getting to this data – also improving performance.
- **OSFA:** Is that dialog one where every possibly option is given and then only one "generalized" stored procedure sits behind it? This is very common and often results in horrible performance and terrible parameter sensitivity problems. Kimberly wrote a really good blog post about this here.
- **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 processing. It may be 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 (or mid-tier) 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.
- **ORMs and statement builders:** Be careful here… potential for plan cache bloat and lots of compile time (if the statements are truly dynamic) or potential for parameter sniffing problems if the statements use sp_executesql behind the scenes. Sadly, there's isn't a simple solution to the ORM problem. Kimberly talks a lot about this in her Pluralsight course: *SQL Server: Optimizing Ad Hoc Statement Performance*.

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 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!

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 Delays 10mm Mass Production to 2019

Intel has been experiencing persistent yield problems and delays as they try to move from a 14nm manufacturing process to a 10nm manufacturing process. Intel was originally supposed to transition from 14nm to 10nm in 2H 2016, but now they have admitted that the move won't be complete until at least 1H 2019.

This multi-year problem is the main reason why Intel finally abandoned their "Tick-Tock" product release cycle in favor of what they call "Process, Architecture, and Optimization" (PAO). It is pretty amazing that Intel has blown what used to be a multi-year advantage in manufacturing process technology compared to their competitors.

AMD is already selling 2nd generation Ryzen desktop processors that use a 12nm manufacturing process, and AMD's CEO Dr. Lisa Su has recently discussed AMD's plans to sample new 7nm server processors later this year.

Microsoft Releases Windows 10 April 2018 Update

On April 30, 2018, Microsoft released the latest semi-annual update for Windows 10, which is called the "April 2018 Update" (rather than another Creator's Update or a Spring Update). This will be Windows 10, version 1803. Some of the new features and top improvements include Timeline, Focus Assist, Dictation, and improvements in the Edge browser.

CNet covers the more interesting new features in this update here. The Verge has a more comprehensive list here. The Microsoft Windows Blog covers what's new here.

Microsoft Releases SQL Server 2016 SP2

On April 24, 2018, Microsoft released SQL Server 2016 Service Pack 2, which is Build 13.0.5026. This Service Pack includes many features and manageability improvements that were originally introduced in SQL Server 2017. In fact, there are enough new changes that I decided to create a separate DMV Diagnostic script just for SQL Server 2016 SP2.

Samsung 970 PRO and EVO M.2 NVMe SSDs Released

Samsung has released a new generation of M.2 NVMe SSDs that use a new Phoenix controller for better performance and better longevity than the previous generation drives. Their quoted speeds for the 970 PRO are 3.5GB/s for sequential reads, 2.7GB/s for sequential writes and up to 500K IOPS for both reads and writes (for QD 32 4K transfers). This level of performance is hitting the limit of a PCIe 3.0 x4 M.2 slot.

StorageReview has a detailed review of the Samsung 970 PRO here. Samsung actually lowered the prices of these drives substantially on launch day. This makes it an easier decision to use one of these drives for an existing or new system with M.2 NVMe slots.

#TBT

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

Producing a performance baseline and other monitoring is the theme this time, so here are some resources for you:

- Erin's Pluralsight course on *SQL Server: Benchmarking and Baselining* that has 3.5 hours covering what metrics to capture and how to capture them.
- If you're still using Trace/Profiler, Erin's Pluralsight course on *SQL Server: Replacing Profiler with Extended Events* will be really beneficial.
- And a bunch of blog posts:
    - Erin: *SQL Server Baselines Series on SQLServerCentral.com*

    - Erin (from our Accidental DBA series): *(Day 18 of 30): Baselines*
    - Erin: *(Day 19 of 30): Tools for On-Going Monitoring*
    - Jon: *(Day 21 of 30): Essential PerfMon counters*
    - Paul: *Capturing IO latencies for a period of time*
    - Paul: *Capturing wait statistics for a period of time*

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

- Erin: *Do you need to update statistics after an upgrade?*
- Erin: *Query Store Examples: Stories from customers*
- Glenn: *Windows Disk Cleanup*
- Glenn: *SQL Server Diagnostic Information Queries for May 2018*

I hope you find these useful and interesting!

Video Demo

From Erin: Over two years ago I recorded an Insider Video talking about new tempdb spill information added to Hash and Sort warnings in the SQL Server 2016 release – check out the newsletter from March 28, 2016 if you're interested. In this week's video I'm covering tempdb spill info again, but now we're running SQL Server 2017 so there's more info in the plan and Extended Events and we're also digging into the DMVs and Query Store too!

The video is about 7 minutes long and you can get it:

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

And the demo code is here.

Enjoy!

Upcoming SQLskills Events

We have lots of events coming up in 2018 – from our ***online*** **IEQS** course to our own LIVE, in-person Immersion Events in both the U.S. and London; all events are open for registration. Every event has a different focus and different benefits – from deep-technical training in our online courses and in-person IEs to wide-ranging topics at SQLintersection where you can learn more effectively how to keep moving forward in both your database and your career! And, of course, one benefit all our in-person events provide is networking.

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
- So why do you want to come to our training? And the winners are…
- Community blog posts about our classes
- Immersion Event FAQ

Online, May 2018

- **IEQS**: Immersion Event on Query Store **** NEW ****
  - May 22-24

Bellevue, WA, June 2018

- **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1
  - June 18-22 **(** Buy 2, get 1 free!, *only 8* seats remaining **)**

London, UK, September 2018

- **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1
  - September 10-14
- **IEAzure**: Immersion Event on Azure SQL Database and Azure VMs
  - September 10-11
- **IECAG**: Immersion Event on Clustering and Availability Groups
  - September 12-13
- **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2
  - September 17-21

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