(December 19th, 2016)

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,

This newsletter is coming to you from Bali, where we've just finished a great 10-day dive trip on the Damai liveaboard dive boat around Komodo.

Please note that the next newsletter will be published on Monday, January 16th. We hope you all enjoy the holiday season with your family and friends! (And watch out for my usual end-of-year blog posts about books and numbers…)

We're looking forward to seeing a bunch of you next May in Orlando, FL for our Spring SQLintersection (check out www.SQLintersection.com for more details) and remember that all of our 2017 classes have discounts of up to $200 for registering before the end of this year (see here for the schedule).

And even if you can't join us in person, I've put out our annual call to user groups to get remote sessions scheduled for 2017. This year we'll end up having done 94 user groups and PASS Virtual Chapter presentations across the team, up from 86 in 2015, which I think is a tremendous achievement. If you'd like one of us to present for your user group, check out my blog post here.

The latest book I've read is Charles Stross' *The Atrocity Archives*. I love Stross's work (and not just because he's from Edinburgh, where I lived for 8 years). This is the first in a 7-part (so far) series about an agent from a secret UK government department called The Laundry, that's responsible for fending off occult threats to the country. The premise of the universe is that complex mathematical proofs are actually formulas for summoning demons and other nasty things, so have to be prevented and guarded appropriately – i.e. math is magic. It's great fun, tongue-in-cheek, and has snappy writing, as I'd expect from Stross. Highly recommended.

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

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 had a random email from someone last week who couldn't get DBCC CHECKDB to run on a database after a storage problem in a VM, with the following results:

Msg 1823, Level 16, State 2, Line 2
A database snapshot cannot be created because it failed to start.

Msg 7928, Level 16, State 1, Line 2
The database snapshot for online checks could not be created. Either the reason is given in a previous error or one of the underlying volumes does not support sparse files or alternate streams. Attempting to get exclusive access to run checks offline.

Msg 5030, Level 16, State 12, Line 2
The database could not be exclusively locked to perform the operation.

Msg 7926, Level 16, State 1, Line 2
Check statement aborted. The database could not be checked as a database snapshot could not be created and the database or table could not be locked. See Books Online for details of when this behavior is expected and what workarounds exist. Also see previous errors for more details.

Msg 845, Level 17, State 1, Line 2
Time-out occurred while waiting for buffer latch type 3 for page (1:1145633), database ID 12.

This means that the database snapshot creation failed, because a page could not be read while performing crash recovery of the database into the snapshot (see section 1 of this post for more background).

The person explained there weren't any backups and was asking about the feasibility of detaching the database, deleting the log file, and then trying to attach the database again. I sent back a big "NOOOO!!!" and explained that in cases like that, the best bet is to extract as much data as possible into a new database rather than trying to hack around the problem.

Bottom line: when you have corruption of some kind and no backups are available, the safest and easiest course of action is to extract data into a new database.

Paul's Ponderings

Following on from the previous two newsletters that discussed the SQL Server 2016 SP1 changes, and how to pick processors and memory for SQL Server 2016, this week's ponderings are from Jonathan as he discusses memory usage and the ramifications of SQL Server 2016 SP1. Enjoy!

One of the most misunderstood configuration options for SQL Server is the 'max server memory' *sp_configure* option, and what exactly it does, as the answer is different depending on which version of SQL Server you are running.

Prior to SQL Server 2012, 'max server memory' only limited single 8KB page allocations for SQL Server, usually only for the Buffer Pool (for the cache of data file pages, the plan cache, execution memory grants, and so on). Starting in SQL Server 2012, the memory manager was rewritten to eliminate the earlier single-page and multi-page allocators and introduce a new any-size page allocator that unified the memory manager, bringing much more of the memory used by SQL Server under the configured value of 'max server memory'.

However, even with the new memory manager, 'max server memory' is not the actual maximum limit of memory that SQL Server can use. Additional memory allocations can come from VAS (Virtual Address Space) through SQL Server's calls to *VirtualAlloc()* internally (or through CLR assembly code that uses the API under the covers), and for the thread call stacks (2MB per thread for x64 systems). This means that SQL Server as a process can allocate and consume more memory than allowed by 'max server memory'.

For this reason, we always recommend setting 'max server memory' based on the installed memory in the server, and calculating a reserved portion of memory to cover the thread stack size, memory demands outside of SQL Server, and OS demands as well. In my Accidental DBA book the recommendation that I make is to reserve:

- 1 GB of RAM for the OS, plus…
- 1 GB for each 4 GB of RAM installed from 4–16 GB, plus…
- 1 GB for every 8 GB RAM installed above 16 GB RAM

And then monitor the *Memory\Available MBytes* performance counter in Windows to determine if you can increase the memory available to SQL Server above that calculated starting value. (Note: This counter should remain above 150-300MB at a bare minimum. Windows signals the *LowMemoryResourceNotification* at 96MB available memory, so you want a buffer, but I typically like it to be above 1GB on servers with 256GB or more RAM).

SQL Server 2016 Service Pack 1 introduces a number of big changes that we've discussed in the last two newsletters. Standard Edition is now allowed to use both columnstore indexes (32GB per instance) and In-Memory-OLTP (32GB per database) which is really awesome.

An interesting side effect of these changes is that 'max server memory' doesn't limit the memory usage for either of these two features, so a Standard Edition server with 128GB set for 'max server memory' and two databases using 32GB of In-Memory-OLTP storage along with columnstore indexes, could have memory allocation requirements of at least 224GB (128GB for the 'max server memory', plus 32GB for columnstore, plus 2 x 32GB for In-Memory OLTP).

Considering that it takes 7% kernel space commit overhead for page table entries to track the memory, that's an additional 16GB of memory required (covered by the reservation formula discussed earlier) before thread stacks are allocated or other memory demands get factored in. If the server only has 256GB RAM installed, that's something to pay attention to because the

server could very easily get into a memory pressure condition where reducing 'max server memory' or adding additional RAM may be required. Note that SQLOS will adjust memory usage under pressure, but there are many undesirable side effects from that happening automatically, and SQL Server consistently being in a memory shrink/grow/shrink/grow cycle.

Monitoring memory usage for SQL Server after applying SQL Server 2016 Service Pack 1 really isn't much different than for other versions of SQL Server – as long as you were doing it correctly to begin with!

Many people look at Task Manager to see the memory used by SQL Server, and if they use *Lock Pages in Memory*, which we recommend, often ask why SQL Server is only using a few hundred MBs of RAM. The value Task Manager shows does not include locked pages in use by SQL Server, it is only pageable commit space allocated under *VirtualAlloc()* by the process and not through *AllocateUserPhysicalPages()*, which is used under *Lock Pages in Memory*.

The correct way to monitor all of the memory used by SQL Server is through the PerfMon counter *SQLServer: Memory Manager\Total Server Memory (KB*). This counter includes all the memory committed by the memory manager for SQL Server and will accurately reflect the full memory usage by the process.

Call to action: make sure you're monitoring SQL Server's memory usage correctly, to avoid undue memory pressure, and if you're expanding SQL Server's memory usage (e.g. on Standard Edition of SQL Server 2016 SP1) using columnstore and/or In-Memory OLTP, be especially careful to take the extra memory usage into account.

Video Demo

In this demo, Jonathan looks at how to appropriately track memory usage by SQL Server using Performance Monitor, and why Task Manager may not accurately report the total memory usage by SQL Server.

The video is about 3 minutes long and you can get it:

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

No demo code this time.

Enjoy!

SQLskills Offerings

We've announced the first half of our 2017 class lineup, with discounts available on all classes for registrations/payments received before January 1st, 2017.

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

Chicago, IL, April/May 2017

- **IE0**: Immersion Event for Junior/Accidental DBAs
  - April 24-26 **US$120 discount for registering in 2016!**
- **IEPTO1**: Immersion Event on Performance Tuning and Optimization – Part 1
  - April 24-28 **US$200 discount for registering in 2016!**
- **IESSIS1**: Immersion Event on Learning SQL Server Integration Services
  - April 24-28 **US$200 discount for registering in 2016!**
- **IEBI**: Immersion Event on Business Intelligence
  - May 1-5 **US$200 discount for registering in 2016!**
- **IEPTO2**: Immersion Event on Performance Tuning and Optimization – Part 2
  - May 1-5 **US$200 discount for registering in 2016!**
- **IESSIS2**: Immersion Event on Advanced SQL Server Integration Services
  - May 1-5 **US$200 discount for registering in 2016!**
- **IEPS**: Immersion Event on Powershell
  - May 8-10 ****NEW** class, US$120 discount for registering in 2016!**
- **IEPDS**: Immersion Event on Practical Data Science
  - May 8-12 **US$120 discount for registering in 2016!**
- **IEHADR**: Immersion Event on High Availability and Disaster Recovery
  - May 8-12 **US$200 discount for registering in 2016!**

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