SQLintersection

Thursday, 11.30 – 12.45

DBA Mythbusters (Level 2-300)

Paul S. Randal
paul@SQLskills.com

SQL intersection

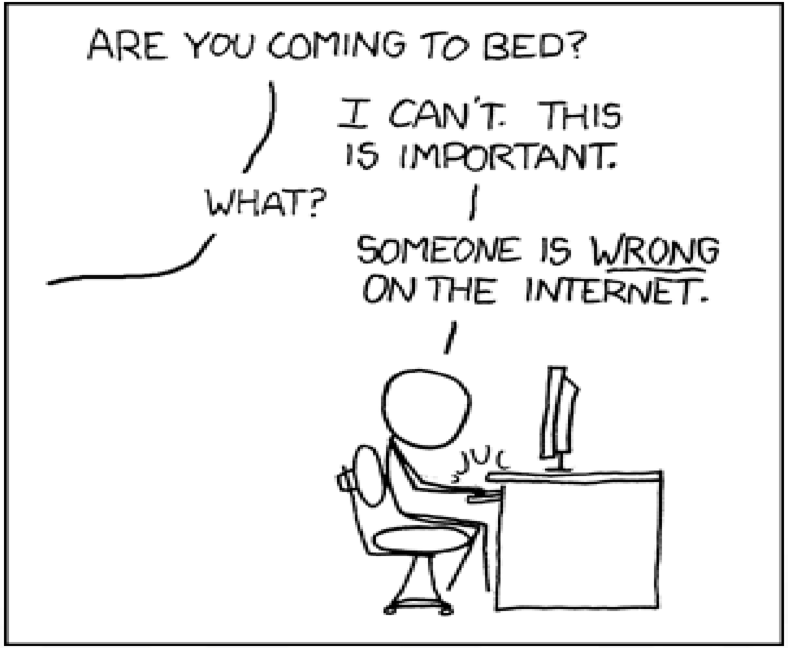
SQLskills
immerse yourself in sql server

This is me: Paul S. Randal

- Consultant/Trainer/Speaker/Author
- CEO, SQLskills.com
  - Email: Paul@SQLskills.com
  - Blog: http://www.SQLskills.com/blogs/Paul
  - Twitter: @PaulRandal
  - 5 years at DEC responsible for the VMS file-system and chkdsk
  - Almost 9 years as developer/manager in the SQL Storage Engine team through August 2007, ultimately responsible for Core Storage Engine
- Instructor-led training (US, UK, Australia), consulting (anything you need)
- Online training: pluralsight hardcore developer training http://pluralsight.com/
- Become a SQLskills Insider: http://www.sqlskills.com/Insider

MVP Microsoft Most Valuable Professional

And This is My Life…


ARE YOU COMING TO BED?
I CAN'T. THIS IS IMPORTANT.
WHAT?
SOMEONE IS WRONG ON THE INTERNET.

Source: http://xkcd.com/386/

SQL intersection

Why Is This Important?

- **Lots of myths and misconceptions have grown and persisted over the years about how SQL Server behaves**
- **Adherence to these misconceptions can lead to:**
  - Bad practices
  - Wasted time and resources
  - Confusion
  - Arguments ☺
- **Five years since I last did Mythbusters sessions and posts…**
  - See http://www.sqlskills.com/blogs/paul/CommonSQLServerMyths.pdf
- **Let's debunk some myths!**

Myth #0

BUSTED!!

- Oracle is much better than SQL Server
- This one's obviously untrue! ☺
- On to the real ones…

Myth #1

BUSTED!!

- Page life expectancy should be around 300

- Page life expectancy measures (in seconds) how long a new page is expected to stay in the buffer pool
  - Also can be thought of as measure of memory pressure on the buffer pool
- Do you think that flushing your 100GB buffer pool every 5 minutes is a sign of a healthy SQL Server?
- That guidance is from 10 years ago!
- Correct PLE is whatever is normal for *your* system
  - If it drops and stays dropped, there’s a problem

SQL *intersection*

Myth #2

IT DEPENDS !!

- Buffer Manager: Page life expectancy is the best counter to use

- If on a NUMA system, Buffer Manager PLE is the harmonic mean of PLEs from all Buffer Nodes
  - Buffer pool is split into partitions on NUMA system
- Monitor all Buffer Node: Page life expectancy counters
- Example harmonic mean calculation
  - 4 buffer nodes with PLE = 4000, 4000, 4000, 2200
  - Buffer Manager PLE = 4 / (1/(1000x4000) + 1/(1000x4000) + 1/(1000x4000) + 1/(1000x3300)) / 1000 = 3321

SQL intersection

Myth #3

BUSTED!!

- You can offload consistency checks to an availability group secondary

- The AG secondary is on a different I/O subsystem from the primary
- Running consistency checks on the secondary says nothing about the state of the primary
- You need to run consistency checks on the primary and ALL secondaries!

- Same argument holds for trying to offload consistency checks to a mirror, log shipping secondary, SAN mirror, etc. – you can't!

SQL intersection

Myth #4

BUSTED!!

- AG readable secondaries don't cause performance problems

- All queries on a readable secondary are converted to snapshot isolation
- This means 14-byte versioning tags must be present on the secondary
- This means the tags have to be added on the primary
  - As the primary and secondaries are exact physical copies of each other
  - But the tags don't have to be filled in on the primary, just the space needs to be accounted for
- All changes to the primary database will incur a versioning tag which will start to cause page splits and index fragmentation!

SQL intersection

Myth #5

MAYBE !!

- Automatic page repair can instantly fix broken pages

- Applies to database mirroring and availability groups
- The secondary/mirror cannot send the requested page back to the primary unless the page is known to be at the right point in time
  - □ What is the LSN in the primary when it requested the page image?
  - □ Secondary redo queue must be replayed to that LSN
  - □ Otherwise the secondary may be sending back the wrong version of the page

SQL intersection

Myth #6

BUSTED!!

- Using temp tables for intermediate query results is always a good idea

- Creating a temp table to hold intermediate results forces SQL Server to interrupt the data pipeline through a query to persist the results to disk
- Sometimes just doing one query rather than pre-aggregating or pre-sorting can be way more efficient and lead to far lower run time and tempdb usage
- Always compare the methods before production
- And if using temp tables, use minimum amount of data and correct indexes

Myth #7

BUSTED!!

- "Fully logged" means you'll always see one log record for each part of an operation

- Consider a rebuild of a 100,000 row index
  - You would expect to see 100 thousand LOP_INSERT_ROW log records, right?
  - Wrong – it will log about LOP_FORMAT_PAGE log records instead with full page images with the net effect of all the inserts on
- "Fully logged" simply means the transaction log contains enough information to reconstitute the transaction after a crash or restore
- What about TRUNCATE TABLE?

SQL
intersection

Myth #8

BUSTED!!

- **NOLOCK / READ UNCOMMITTED means no locks**

- **First off, they're the same thing**
- **And they do have to acquire some locks:**
  - Schema-stability locks (Sch-S) to prevent the structure of the table/index changing
  - BULK_OPERATION locks on heaps to prevent reading of unformatted pages
- **And they still have to take latches to access the physical page images in memory, so there's still some potential for blocking at the latch level**

SQL *intersection*

Myth #9

BUSTED!!

- You should always plan a backup strategy

- Always plan a *restore* strategy
- Then plan what backups you need to take

- The other way can result in disaster

- Let me tell you a story…

SQL
intersection

Myth #10

BUSTED!!

- The best thing to put on SSDs are always tempdb and transaction logs

- Don’t fall into the trap of listening to other people
- Investigate where your biggest I/O subsystem bottleneck is
  - Try to solve it within SQL Server
  - If not, put that on your SSD
- Or design a new I/O subsystem layout to take advantage of the SSD
- What about the RAID level to use?

Myth #11

BUSTED!!

- Using SSDs means you don’t have to care about index fragmentation

- Index fragmentation has two forms:
  - Logical fragmentation that stops efficient readahead
  - Low page density that wastes space
- SSDs make reads faster, but still a trip down/up I/O stack for each one
- SSDs don’t stop page splits from happening
  - Lots of extra transaction log
- SSDs don’t stop low page density from happening
  - Wasted disk space, wasted buffer pool memory

Myth #12

BUSTED!!

- Adding more memory is always a good idea

- Consider some of the potential problems
  - □ Shutting down the instance will take longer
  - □ P.O.S.T. of the server will take longer
  - □ Allocating buffer pool memory may take longer (see KB article 2819662)
  - □ Warming up the buffer pool will take longer
  - □ Could lead to complacency

SQL intersection

Myth #13

BUSTED!!

- Shrinking tempdb can cause corruption

- KB article 307487 was updated in 2014
- No problems with shrinking tempdb since SQL Server 2000
- However, just because you can, doesn't mean you should…
- And be aware of what happens to tempdb size on a server restart

SQL intersection

Myth #14

BUSTED!!

- **DBCC CHECKDB runs when SQL Server starts up**

- **The messages in the error log are confusing:**
  - 2015-05-15 13:16:20.07 spid7s   CHECKDB for database 'master' finished without errors on 2015-05-01 09:59:42.447 (local time). This is an informational message only; no user action is required.
- **This is just reporting the time that DBCC CHECKDB last completed without finding any errors**
  - Stored in the boot page of the database (file 1, page 9)
  - Check with DBCC TRACEON (3604); DBCC DBINFO;

Myth #15

BUSTED!!

- Rebuilding indexes solves performance problems even when there's no index fragmentation

- It's the query plan recompilation that 'fixes' the performance problem
- Rebuilding an index causes plan recompilation for plans on that table
- If a poor query plan had resulted (e.g. from parameter sniffing), the next plan to be compiled might be better
- Also, updating statistics in 2012+ doesn't invalidate plans if no table rows changed (this is a good thing!)

SQL intersection

Myth #16

MAYBE !!

- Adding an extra file to tempdb will help solve contention issues

- Adding an extra file means SQL Server can alternate between the files
- But allocation also takes into account proportional fill
  - It will allocate proportionally more from files with more free space
- If the existing file is quite full, the new file becomes allocation hot spot
  - No alleviation of contention issues!
- Make sure to take that into account when working with tempdb

SQL intersection

Myth #17

BUSTED !!

- Lots of OLEDB waits always means linked-server problems

- OLEDB waits mean that the OLE-DB protocol is being used
  - OLE-DB is not just used by linked servers
- How long are the waits?
  - 0 – 1-2ms = not linked servers
  - 10s or 100s of ms are likely to be linked servers

Myth #18

BUSTED !!

- ASYNC_NETWORK_IO waits always means network problems

- Rare for it to be the network
- The word NETWORK is horribly misleading in the name
- More likely the application doing RBAR processing

SQL intersection

Myth #19

BUSTED !!

- CXPACKET waits mean disable parallelism

- Check you expect parallelism for that query
- Check you don't have skewed parallelism
- Consider altering MAXDOP for query
- Consider setting server MAXDOP (8 or number of cores in a NUMA node)
- Consider using Resource Governor
- Better: increase 'cost threshold for parallelism'

Myth #20

BUSTED !!

- Checkpoints only write committed changes to disk
- Checkpoint writes all pages marked dirty regardless of whether the change was made by a committed or uncommitted transaction
  - Crash recovery takes care of fixing things up if there’s a crash
- Use sys.dm_os_buffer_descriptors to examine the relative proportion of dirty vs. clean pages in the buffer pool

SQL intersection

Myth #21

BUSTED !!

- DBCC DROPCLEANBUFFERS flushes the buffer pool

- Look at the name carefully – drop clean buffers
- It doesn't flush dirty pages
- You have to do a checkpoint for that, and then DROPCLEANBUFFERS

SQL intersection

Myth #22

BUSTED !!

- Tempdb data files should be 1:1 with processor cores

- SQL Server 2000: rule was #files = #logical processor cores, and TF 1118
  - E.g. my laptop CPU has 4 physical cores plus hyperthreading = 8 logical cores
- SQL Server 2005 onwards: Microsoft guidance was same until 2011
  - Everyone else said to start with ¼ to ½ the number of logical processor cores
- Universal guidance now in KB article 2154845
  - < 8 cores, start with #files = #cores
  - > 8 cores, start with #files = 8
  - Increase in blocks of 4 if still seeing contention

Myth #23

BUSTED !!

- Multiple log files will help performance

- SQL Server will always use log files sequentially
- You may see them all having I/Os, but that’s just updating the file header pages
- The only time another log file is needed is if the first one fills up and cannot grow, you cannot take a log backup, and you do not want to break the log backup chain
- Remove additional log files once you don’t need them

Myth #24

BUSTED !!

- The log should always be as small as possible

- The log needs to be as big as it needs to be
- Do not regularly shrink the transaction log
  - It'll just have to grow again, and can't use instant initialization
- How big should the log be?
  - Single largest transaction (ETL, large index rebuild, large update)
  - Asynchronous database mirroring/AG SEND queue
  - How long is the longest data backup?
  - Transactional replication (beware of CDC too)

Myth #25

- It depends

YES, IT REALLY DOES!!

- The answer to all questions about SQL Server that do not have obvious yes/no answers always starts with ‘it depends’
- The trick is then to explain *why* it depends, *what* it depends on, and *when* it depends
- Don't accept ‘it depends’ as an answer with no follow-on explanation

NOOOOOOOO!!

- One exception: should auto-shrink be enabled?

Plenty More Myths Around…

- **Repair**
  - It can fix everything
  - Safe to repair system databases
  - SQL Server causes corruptions
  - Corruptions can disappear
- **Performance**
  - You can't override MAXDOP
  - Always use data compression
  - Nested transactions exist
- **The transaction log**
  - Log records can move
  - The log is zeroed when cleared
  - BULK_LOGGED lowers backup size
- **High availability**
  - Just use a cluster
  - Replication isn't an HA solution

ALL BUSTED !!

SQL *intersection*

Summary and Resources

- Make sure you corroborate what you read online
- If something sounds fishy, try it yourself!

- Blog:
  - http://www.sqlskills.com/blogs/paul/category/misconceptions/
- Pluralsight
  - SQL Server: Myths and Misconceptions

Questions?

Don’t forget to complete an online evaluation on EventBoard!

DBA Mythbusters

Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

SQL intersection

Thank you!