SQLintersection

Session: Thursday, 11.30am – 12.45pm

Corruption Survival Techniques (Level 200)

Paul S. Randal

paul@SQLskills.com

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, Ireland, Australia), consulting (anything you need)**
- **Online training: pluralsight** hardcore developer training **http://pluralsight.com/**
- **Get our bi-weekly newsletter: http://www.sqlskills.com/Insider**

Why Is This Session Important?

- **Corruption does happen, mostly caused by I/O subsystem problems**
- **People don't realize they have corruption until too late**
  - Either they don't know how to check for corruption or they miss warning signs
- **People don't know what to do when they do have corruption, leading to:**
  - More data loss and downtime than necessary
  - Monetary and even job losses
  - Overall lowered perception of SQL Server
    - Makes it harder to convince management that SQL Server is Enterprise capable

What Can Happen to an Unprepared DBA Confronted by Corruption?

Image from http://commons.wikimedia.org/wiki/File:Panic_button.jpg

SQL
intersection

Make Sure That…

- **If you're recovering from a disaster, or helping a client recover from a disaster:**
  - You must know what you're doing and have practiced
  - You must NOT make things worse
- **You must perform root-cause analysis**
  - Maybe obvious, e.g. a known SAN problem or power failure
  - You may have no idea what happened, so what to do?
    - Google/Bing for the corruption message you saw
    - Run I/O subsystem and server memory diagnostics
    - Examine the SQL Server error log and Windows event logs for clues
    - Check that firmware is up-to-date, and no buggy NTFS filter drivers
    - Possibly contact Microsoft Customer Support for assistance

Overview

- **How to run DBCC CHECKDB?**
- **How do the checks and repairs work?**
- **FAQ and best-practices**
- **Interpreting DBCC CHECKDB output**
- **Recovering from simple corruptions**

How Does Corruption Occur?

- **It's the I/O subsystem, usually always**
  - 0.0005% bad memory, 0.0005% SQL Server bugs
- **Consider MTBF of disks, you're going to have a problem at some point**
- **Jim Gray likened a disk head in a 15000rpm disk to a 747 flying at 500mph about ¼ inch above the ground**
  - What happens in a crash?
- **Corruptions cannot be caused by:**
  - Anything an application can do
  - Interrupting a shrink, rebuild, or long-running batch
  - Propagation by log-shipping, replication, mirroring, AGs

I/O Errors

- **Three types**
  - 823 – a hard I/O error
  - 824 – a soft I/O error
  - 825 – a read-retry error

- **Msg 824, Level 24, State 2, Line 1**

- *SQL Server detected a logical consistency-based I/O error: incorrect checksum (expected: 0x7232c940; actual: 0x720e4940). It occurred during a read of page (1:143) in database ID 8 at offset 0x0000000011e000 in file 'c:\sqlskills\broken.mdf'. Additional messages in the SQL Server error log or system event log may provide more detail. This is a severe error condition that threatens database integrity and must be corrected immediately. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.*

- **Logged in msdb.dbo.suspect_pages**

DBCC CHECKDB

- **The only way to read all allocated pages in the database**
  - Use to force page checksums to be checked
- **Choose between full checks and WITH PHYSICAL_ONLY**
- **Many algorithms to minimize runtime and run ONLINE**
- **New features in and since 2005**
  - Progress reporting, data purity, last known good, no false failures, increased scalability, reduced memory requirements
- **Blog post series:**
  - http://bit.ly/TFvxmG
  - Detailed information (~90 pages) in the *SQL Server 2008 Internals* and *SQL Server 2012 Internals* books

What Exactly Does CHECKDB Do?

- **Primitive checks of critical system tables**
  - Problems? Game over.
- **Allocation checks (i.e. DBCC CHECKALLOC) and repairs**
- **Logical checks of critical system tables and repairs**
- **Logical checks of all other tables (i.e. DBCC CHECKTABLE) and repairs**
- **Service Broker data validation and repairs**
- **Metadata checks (i.e. DBCC CHECKCATALOG) (no repairs)**
- **Indexed view and XML index checks and repairs**
  - Only under the WITH EXTENDED_LOGICAL_CHECKS option in SQL Server 2008+
- **What about when PHYSICAL_ONLY is used?**

How To Run DBCC CHECKDB

- **By default, CHECKDB will:**
  - Only return the first 200 errors in earlier versions
  - Return lots of info that's distracting in a corruption situation
- **Use the following command with only these options:**
  - DBCC CHECKDB (yourdb) WITH ALL_ERRORMSGS, NO_INFOMSGS
- **If it's taking longer than usual, that should mean that it found some corruption**
  - Check the error log for message 5268 to see if it's rescanning some data
- **Most importantly, wait for it to complete!**

How Often Should I Run CHECKDB?

- **It all depends on a combination of:**
    - Stability of I/O subsystem
    - Backup strategy
    - Acceptable downtime if corruption occurs
    - Acceptable data-loss if corruption occurs
    - Window available to take the extra I/O and CPU load
    - What kind of system it is (e.g., production, test, backup)
    - How the data is partitioned
- **Examples:**
    - No backups and persistent corruptions
    - VLDB with very low downtime/data-loss tolerance

SQL intersection

Consistency Checking Survey

How often do you run consistency checks? (regardless of *how* you run them)

What are consistency checks?		9%	25
Never		5%	14
Only when corruption is detected some other way		8%	22
Only during an event like an upgrade or migration		1%	2
Regularly, but less than monthly		3%	8
Monthly		5%	14
Weekly		37%	103
Daily		25%	69
More frequently than daily		1%	3
Only after performing a restore, or after a failover		1%	3
Other? Enter here...		5%	13
		Total: 276 responses	

SQL
intersection

How Long Will CHECKDB Take to Run?

- **Depends on many factors:**
  - Size of the database and complexity of the database schema
  - Concurrent I/O and CPU load on the server
  - Concurrent update activity on the database
  - Throughput capabilities of the I/O subsystem, especially tempdb
  - Number of CPUs on the box
  - Which options were specified
  - Number and type of corruptions that exist
- **See blog post for more details:**
  - http://bit.ly/RRL570

How To Consistency-Check VLDBs?

- **DBCC CHECKDB of a multi-TB database can take hours!**
- **Five options for reducing the run time:**
    - Don't run any checks
    - Run WITH PHYSICAL_ONLY
    - Break up the checks
    - Use partitioning
    - Use another system
- **Don't give up—with a little thought it's possible**

Consistency Checking Using Another Server (1)

- **Removes the consistency checking workload from the production server**
- **Methodology:**
  - Perform full database backup on the production server
  - Restore database on another server
  - Run consistency checks on the restored database
- **What to do if a corruption is found?**
  - Is it the I/O subsystem on this server?
  - Is it the backup file?
  - Is it the database on the production server?
- **If corruption is found, which should be rare, it forces you to run consistency checks on the production server**

Consistency Checking Using Another Server (2)

- **Some people advocate consistency checking a mirror database or an availability group secondary replica as a way of consistency checking the principal database or primary replica**

- **This is not valid**

- **The various servers are stored on different I/O subsystems and corruptions do not propagate between servers**
  - E.g. ensuring that a mirror database has no corruption does not imply anything about the state of the principal database

- **Consistency checking should be performed on all copies of a database**
  - Especially true for availability groups if a secondary replica is being used to offload full backups from the primary replica

Consistency Checking Survey

What method do you use to run consistency checks on your production database(s)?

Run DBCC CHECKDB with no options on the production database		62%	261
Run DBCC CHECKDB WITH PHYSICAL_ONLY on the production database		10%	42
Run DBCC CHECKDB on a restored backup on another server		11%	47
Run DBCC CHECKDB on a database snapshot on a mirror database		1%	4
Spread consistency checking over several days using DBCC CHECKTABLE		0%	1
Spread consistency checking over several days using DBCC CHECKFILEGROUP		1%	4
Use BACKUP WITH CHECKSUM to validate page checksums, no DBCCs		1%	4
Run DBCC CHECKDB on a log shipping secondary, or 2012 AG secondary		1%	3
Run DBCC CHECKDB on a SAN copy/mirror		2%	7
Create your own database snapshot of the production database and run DBCC CHECKDB on it		0%	2
Other? Enter here...		11%	47
		Total: 422 responses	

SQL
intersection

How To Tell Something Went Wrong?

- **So you setup a regular job to run consistency checks, and turned on page checksums – how can you tell if it goes wrong?**
- **There has to be some kind of monitoring otherwise you'll never know!**
- **Solution: Agent alerts, or other monitoring (e.g. SCOM)**
- **Create alerts for:**
    - Severity 19 errors and above
    - Any user-defined errors (e.g. flagging that CHECKDB job failed)
    - Anything else you're interested in
- **See http://bit.ly/hrBiTK and http://bit.ly/TFwKdK**

SQL
intersection

As Soon As Corruption Is Suspected

- **No need to panic!**
  - Panic leads to mistakes and downtime/data-loss
- **Go to the DR run-book**
- **Determine the extent of the corruption**
  - Run DBCC CHECKDB
  - Look in the SQL Server error log, Windows event log
  - Check maintenance job history
- **Check what backups are available**
- **Wait for CHECKDB to finish before doing anything else**
  - You may not NEED to do anything intrusive/destructive

Interpreting DBCC CHECKDB Output (1)

- **There are over 100 errors that DBCC CHECKDB can output, some with over 200 message states**
  - Effectively there are roughly a thousand different corruption conditions that can be reported
- **Figuring out what one corruption means isn't too bad**
  - MSDN has some of them published
- **Figuring out multiple corruptions can become very hard and usually isn't worth the time**
- **There are some tips and tricks you can use to determine the course of action to take**

SQL *intersection*

Interpreting DBCC CHECKDB Output (2)

- **Did DBCC CHECKDB fail?**
    - If it stops before completing successfully, something bad has happened that is preventing it from running correctly
    - This means there is no choice but to restore from a backup or try exporting the data, as DBCC CHECKDB cannot be forced to run (and hence repair)
- **Examples of fatal (to DBCC CHECKDB) errors:**
    - 7984 – 7988: corruption in critical system tables
    - 8930: corrupt metadata such that DBCC CHECKDB could not run
- **The SQL Server error log message will list an error state**
    - See the "Understanding DBCC Error Messages" portion of the Books Online for DBCC CHECKDB for details at http://bit.ly/179p6At

Demo

Example fatal errors to DBCC CHECKDB

Interpreting DBCC CHECKDB Output (3)

- **Are the corruptions only in nonclustered indexes?**
  - If recommended repair level is REPAIR_REBUILD, then yes
  - Otherwise, check all index IDs in errors and if all greater than 1, then yes
  - If yes, you don't need to restore or run repair
- **Was there an un-repairable error found?**
  - Examples:
    - 2570 error: invalid data for the column type (data purity error)
    - 8992 error: CHECKCATALOG (metadata mismatch) error
    - 8909, 8938, 8939 (page header corruption) errors where type is 'PFS'
  - None of these can be automatically repaired so your options are to restore or to attempt manual repairs

SQL
intersection

Manually Fixing Nonclustered Indexes

- **It doesn't make sense to put the database offline and run DBCC CHECKDB or DBCC CHECKTABLE with REPAIR_REBUILD to fix corrupt nonclustered indexes**
- **All it will do is disable and rebuild the index, so why not do it yourself?**
- **You cannot just rebuild the index**
  - Index rebuilds read the old index to build the new index
- **Steps to use, inside a transaction:**
  - ALTER INDEX name ON tablename DISABLE
  - ALTER INDEX name ON tablename REBUILD
- **Using a transaction is necessary to prevent any index-enforced constraints from being violated while the index is disabled**

Demo

Nonclustered index corruption only

Manually Fixing Data Purity Errors

- **If a 2570 data purity error was found, SQL Server cannot repair it**
  - The error is that a column value is out-of-bounds
  - Which value should SQL Server pick to repair it?
- **You must manually repair data purity errors**
  - Work out which row is invalid using SELECT or DBCC PAGE
  - Update the column to a valid value
  - http://support.microsoft.com/kb/923247 explains the options
- **Make sure you choose a value that makes sense for your business logic and the data the column value represents**

Demo

Manually repairing an invalid data value (2570)

Recovering Using Backups

- **Best way to avoid data loss**
- **Not necessarily the best way to avoid downtime**
  - Depends what kind of backups are available
  - Although data and backup compression help...
- **Plethora of options available**
  - Full database backup is a good starting point
  - Series of transaction log backups as well is much better
- **Remember:**
  - Backups have to exist to be useful
  - Backups have to be valid to avoid data loss

SQL
intersection

Restore vs. Repair

- **Did DBCC CHECKDB fail?**
  - Yes – you must restore or export, as you cannot run repair
- **Is it just nonclustered indexes that are damaged?**
  - Yes – neither restore or repair, manually rebuild them
- **Are there any un-repairable errors?**
  - Yes – you must restore or export, or potentially manually repair them
- **If you're still able to make a repair vs. restore choice:**
  - Consider your down time and data loss Service Level Agreements
  - Use option that limits down time and data loss while adhering to SLAs
- **There is a comprehensive flow chart of the decision making process available at http://bit.ly/VGFQH5**

Running Repair

- **Not always a last resort compared to restore**
  - Depends what backups are available and which SLA is more important
- **All repair options require SINGLE_USER mode, i.e. they're offline**
  - REPAIR_REBUILD – only rebuilds indexes
  - REPAIR_ALLOW_DATA_LOSS – repairs that will likely lose data
- **Specifying a repair option makes DBCC CHECKDB run single-threaded**
- **Try to repair smallest thing**
- **You can put a transaction around your repair operation and roll it back if you don't like what repair did**
- **Note: running repair on system tables usually causes more corruption**

Beware of REPAIR_ALLOW_DATA_LOSS

- **It was very deliberately named**
- **It usually fixes structural inconsistencies by de-allocating**
- **It doesn't take into account:**
  - Foreign-key constraints, replication, business logic and data relationships
- **Before running repair, protect yourself**
  - Take a backup and quiesce replication topologies involved
- **After running repair, check the data**
  - Run DBCC CHECKDB again to make sure all corruptions were repaired
  - Run DBCC CHECKCONSTRAINTS if necessary
  - Reinitialize any replication topologies involved

If You Are Forced To Use Repair...

- **That implies that your backup strategy does not allow you to meet your downtime and data loss Service Level Agreements**
- **Update your backup strategy!**
  - Figure out what restores you need to be able to perform
  - Change backup strategy to perform the backups that allows those restores
  - Implement regular backup validation
- **Also make sure that:**
  - You check constraints affected based on which tables were repaired
  - You check to see what data was lost
  - You reinitialize any affected replication topologies
  - Perform root-cause analysis

Demo

Using repair

Further Considerations

- **You might need to perform a tail-of-the-log backup**
  - And maybe a hack-attach log backup on another server
- **You might be able to use a single-page restore**
- **You may need to use EMERGENCY-mode repair if the log is damaged**
- **You may need to manually fix metadata problems**
- **You may need to hack the boot page to attach the database**
- **You may need to resort to manual data extraction or manipulation**
- **Lots more on these and other topics in the two Pluralsight courses on basic and advanced corruption recovery**

Resources

- **Blog post series**
  - http://www.sqlskills.com/blogs/paul/category/corruption/
  - http://www.sqlskills.com/blogs/paul/category/checkdb-from-every-angle/
  - http://www.sqlskills.com/blogs/paul/category/disaster-recovery/
  - http://www.sqlskills.com/blogs/paul/dbcc-writepage/
- **Pluralsight online courses**
  - SQL Server: Detecting and Recovering from Database Corruption
  - SQL Server: Advanced Corruption Recovery Techniques

Review

- **How to run DBCC CHECKDB?**
- **How do the checks and repairs work?**
- **FAQ and best-practices**
- **Interpreting DBCC CHECKDB output**
- **Recovering from simple corruptions**

SQL
intersection

Questions?

Don't forget to complete an online evaluation!

Corruption Survival Techniques

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

Thank you!

Save the Date!

www.SQLintersection.com

WALT DISNEY WORLD
SWAN

2018
Mar 25-28

We're back in Orlando!

Access Epcot and Hollywood Studios by taking the boat!

SQL
intersection

Leave the every day behind and enter a world of wonder and enchantment at the Walt Disney World® Resort. Located in the heart of the most magical place on earth, the Walt Disney World Swan and Dolphin Resort provides a truly extraordinary backdrop for our event! Beautiful tropical landscaping, tranquil waterways, and classic art and architecture work together to create a stunning landmark!