SQL Server: Detecting and Correcting Database Corruption

Module 2: Causes of Database Corruption

Paul S. Randal
http://www.SQLskills.com/blogs/paul/
Paul@SQLskills.com

pluralsight
hardcore developer training

Introduction

- **It’s very useful to understand the causes of corruption and why it’s almost inevitable that you will see database corruption at some point in your SQL Server career**

- **In this module we’ll cover:**
  - What causes corruption
  - What does not cause corruption
  - Corruption propagation to remote servers
  - “Disappearing corruption” phenomenon

How Does Corruption Occur? (1)

- **If a data file page is “good” when written out of SQL Server’s memory, but “bad” when read back into memory, that’s corruption**
  - Log files can be corrupt too, which we discuss in the advanced course
- **It’s almost always the I/O subsystem**
- **The I/O subsystem means anything “underneath” SQL Server**
  - Windows operating system
  - File system filter drivers
    - E.g. antivirus, defraggers, encryption
  - Network cards, switches, cables
  - SAN controllers
  - RAID controllers
  - Disks
- **Lots of moving parts, lots of code**
  - Where these is code, there is the potential for bugs

How Does Corruption Occur? (2)

- **Consider the Mean Time Between Failure (MTBF) of disks**
  - Various sources online give figures around 1.5 million hours as MTBF as long as the operating environment is ideal
    - Dust, high/low temperature, vibration, power cycling are not ideal
  - It's a *mean*, which means there's a bell-curve of failure times
    - Someone's going to get the low-end of the curve
- **Jim Gray likened a disk head in a 15,000rpm disk to a Boeing 747 flying at 500mph about ¼ inch above the ground**
  - What are the tolerances involved?
  - What happens in a crash?

How Does Corruption Occur? (3)

- **What if it’s not the I/O subsystem?**
- **Memory corruption**
  - Bad memory chips
  - Scribblers
- **SQL Server bugs**
- **Human error**
  - Manually trying to edit a database
  - Manually deleting a transaction log

Increase Protection Using RAID

- **RAID**
  - Redundant Array of Independent/Inexpensive Disks
- **Various configurations give degrees of redundancy**
  - RAID-0 (striping)
  - RAID-5 (rotating parity)
  - RAID-1 (mirroring)
  - RAID-10 (striping over mirrors)
- **RAID levels: http://en.wikipedia.org/wiki/RAID**
- **This applies to SSDs (Solid State Drives) as well**
  - A single SSD card/drive is RAID-0
    - Even an SSD which presents two “drives” is still a single point of failure
  - Use multiple SSDs in at least RAID-1 configuration

What Does NOT Cause Corruption

- **Many misconceptions about what can cause corruption**
- **Corruptions are not caused by:**
  - Anything an application can do
  - Anything you can do in SQL Server with supported, documented commands
  - Interrupting a database shrink, index rebuild, or long-running batch
  - Shutting down SQL Server
- **In a very rare case, you may hit a bug in SQL Server**

Corruption Propagation To Remote Servers

- **SQL Server redundancy technologies do not propagate data file corruptions caused by the I/O subsystem**
  - Database mirroring, log shipping, replication, availability groups
- **These technologies send log records (or actions based on log records), <u>not</u> data file pages from disk**
- **Possible to have a "second-order" corruption propagated**
  - Corrupt page is not detected on the local server and used to calculate a result which is stored locally, and sent to the remote server
- **Backing up a database should detect corruption, preventing propagation to a remote server during restore**
  - Covered in Module 4
- **SAN replication should also not propagate corruption**
  - Replicating changed disk blocks as they're written, not read from disk

Disappearing Corruption

- **Typical description of the phenomenon:**
  - Consistency-checking job fails during the night
  - DBA runs a consistency check again in the morning, but no corruption!
  - DBA questions where the initial report of corruption was correct
- **What's really happening:**
  - Corruption is reported on some pages allocated to an index
  - After the consistency-checking job runs, an index maintenance job runs
  - The index containing the corrupt pages is rebuilt, which builds a new index with new pages, and then de-allocates the old index and pages
  - Another consistency check runs but doesn't find any corruption
  - The second consistency check is reading different pages from the first one, so doesn't see the corrupt pages
- **Consistency checks validate those data file pages that are allocated to tables and indexes in the database**

Summary

- Most corruptions are caused by software bugs or hardware problems in the I/O subsystem
- Corruptions rarely propagate to remote servers
- Resiliency can be vastly increased using RAID

- In the next module, we'll discuss:
  - I/O errors
  - How SQL Server detects corruptions while reading pages