Gold partner:

Generální partner:

Skryté poklady SQL TempDB

Marek Chmel

Microsoft MVP: SQL Server Architecture
Microsoft Certified Trainer, Regional Lead
Certified Ethical Hacker

18. – 21. května 2015

Tech·Ed DevCon

Session Agenda

- TempDB overview
- TempDB usage
- TempDB scaling and optimizations
- TempDB monitoring
- Demos

What is TempDB?

- TempDB is a system database (ID = 2)
- Runs with simple recovery model
- Cannot be backed up
- Is recreated after every service start up

TempDB Usage

- TempDB is a system DB, created during the installation
- TempDB is recreated after every restart
- Typical usage of TempDB
  - Temporary tables
  - Table variables
  - RCSI version store
  - Index sort
  - Query memory spills
  - AlwaysOn AG ReadOnly Replicas

Usage continued

- Internal objects which are using TempDB include
  - INSTEAD OF triggers to store data for internal processing
  - Service Broker to store messages
  - Cursors to store a key and the query results
  - Storing XML variables and other LOB variables
  - Storing intermediate results for hash joins and hash aggregates

DEMO

TempDB content

Temporary Tables

- Temporary tables are similar to permanent tables, except temporary tables are stored in **tempdb** and are deleted automatically when they are no longer used
- Can be configured as local or global
- Cannot be referenced in FOREIGN KEY constraints

- **Myth #1**: Temporary tables are physical only

Table variables

- Special data type used to store a result set
- Automatically cleared at the end of that batch
- Can result in fewer recompilations of a stored procedure as compared to temporary tables
- Transactions against table variables require less locking and logging resources

- **Myth #2**: Table variables reside in memory only

DEMO

Temporary tables and table variables

DEMO

Triggers

Sort in TempDB for Index Operations

- use **tempdb** to store the intermediate sort results that are used to build the index
- this option increases the amount of temporary disk space that is used to create an index, but could reduce the time that is required to create or rebuild an index when **tempdb** is on a set of disks different from that of the user database
- The Database Engine first scans the data pages of the base table to retrieve key values and builds an index leaf row for each data row. When the internal sort buffers have been filled with leaf index entries, the entries are sorted and written to disk as an intermediate sort run.

Online Index Operations

- If a clustered index is created, rebuilt, or dropped online, a temporary nonclustered index is created to map old bookmarks to new bookmarks
- If the SORT_IN_TEMPDB option is set to ON, this temporary index is created in **tempdb**

TempDB and Eager Writes

- Eager writes occur when a transaction is writing a large amount of data to the buffer pool.
- SQL Server 2014 introduced a new enhancement to eliminate eager writes for minimally logged transactions in tempdb
  - BULK IMPORT
  - SELECT INTO
  - SORT_IN_TEMPDB = ON for indexes
- The change in SQL Server 2014 is to relax the need to flush these pages, as quickly, to the TEMPDB data files. When doing a select into … #tmp … or create index WITH SORT IN TEMPDB the SQL Server now recognizes this may be a short lived operation.
- This was ported also into SQL 2012
  - SQL 2012 SP1 CU10
  - SQL 2012 SP2 CU1

RCSI

- SQL Server uses by default “pessimistic” locking in user databases
- Two modes allow optimistic locking
  - Snapshot Isolation Mode
  - Read Committed Snapshot Isolation Mode (RCSI)
- Once snapshot isolation is enabled, updated row versions for each transaction are maintained in **tempdb**

DEMO

How does RCSI work?

TempDB Scaling

Creating More DB Files

- You can create more DB files for TempDB
  - If you have less than 8 cores then #files = #cores
  - For more than 8 cores #files = 8 and if needed increase in blocks of 4
- What is critical
  - Files need to be the same size and same autogrowth settings
  - You can still use the traceflag #1118
  - You can carefully use the traceflag #1117
- Ideally use dedicated storage for TempDB

- **Myth #3**: Data files should be 1:1 with processor cores

TempDB Sizing

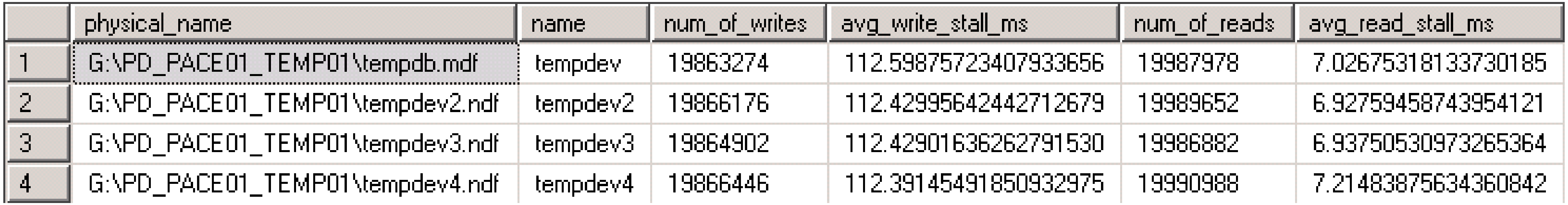
- There is no formula to calculate the right TempDB size
- TempDB is recreated after startup
  - Log has to be zeroed
  - Data file might not be if you have IFI configured correctly

- **Myth #4:** TempDB size should be X % of the largest DB

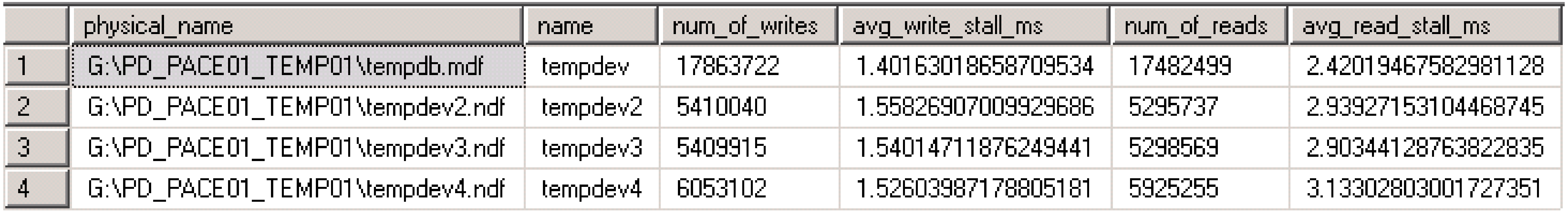
TempDB Monitoring

TempDB Monitoring

- Monitoring the read and write stalls
  - – Writes should finish within 20ms or less
  - – All files should be utilized equally



	physical_name	name	num_of_writes	avg_write_stall_ms	num_of_reads	avg_read_stall_ms
1	G:\PD_PACE01_TEMP01\tempdb.mdf	tempdev	19863274	112.59875723407933656	19987978	7.02675318133730185
2	G:\PD_PACE01_TEMP01\tempdev2.ndf	tempdev2	19866176	112.42995642442712679	19989652	6.92759458743954121
3	G:\PD_PACE01_TEMP01\tempdev3.ndf	tempdev3	19864902	112.42901636262791530	19986882	6.93750530973265364
4	G:\PD_PACE01_TEMP01\tempdev4.ndf	tempdev4	19866446	112.39145491850932975	19990988	7.21483875634360842



	physical_name	name	num_of_writes	avg_write_stall_ms	num_of_reads	avg_read_stall_ms
1	G:\PD_PACE01_TEMP01\tempdb.mdf	tempdev	17863722	1.40163018658709534	17482499	2.42019467582981128
2	G:\PD_PACE01_TEMP01\tempdev2.ndf	tempdev2	5410040	1.55826907009929686	5295737	2.93927153104468745
3	G:\PD_PACE01_TEMP01\tempdev3.ndf	tempdev3	5409915	1.54014711876249441	5298569	2.90344128763822835
4	G:\PD_PACE01_TEMP01\tempdev4.ndf	tempdev4	6053102	1.52603987178805181	5925255	3.13302803001727351

TempDB Contention

- **Global Allocation Map (GAM):** Tracks which extents have been allocated. There is 1 GAM page for every 4 GB of data file. It is always page 2 in the data file and then repeats every 511,232 pages.
- **Shared Global Allocation Map (SGAM):** Tracks which extents are being used as mixed (shared) extents. There is 1 SGAM page for every 4 GB of data file. It is always page 3 in the data file and then repeats every 511,232 pages.
- **Page Free Space (PFS):** Tracks the allocation status of each page and approximately how much free space it has. There is 1 PFS page for every 1/2 GB of data file. It is always page 1 in the data file and then repeats every 8,088 pages.

DEMO

TempDB Scaling and Contention Monitoring

Session End

Ing. Marek Chmel, MSc

Sr. Technical Team Lead, SQL Server Team
mc654x@att.com | marek.chmel@technet.ms

Aktuální a navazující kurzy sledujte na www.gopas.cz

DÁREK PRO VÁS!

Vyplňte dotazníkové hodnocení a…

…získejte tričko **TechEd-DevCon 2015!**

SOUTĚŽ! SOUTĚŽ! SOUTĚŽ!

TechEd party!
Billiard klub Harlequin Praha, 20.5.2015

Buďte The Best IT Pro
nebo The Best Developer

GOPAS DAQUAS Microsoft Tech-Ed DevCon