# **SQLintersection**

Session: 11/20, 1:30am - 2:45pm

# Practical Guidance to Make Your Tier-1 SQL Server ROAR!

Pam Lahoud Pedro Lopes





# **Speakers: Pam Lahoud & Pedro Lopes**









@SQLPedro



# **Reminder:** Intersect with Speakers and Attendees

- Tweet tips and tricks that you learn and follow tweets posted by your peers!
  - □ Follow: #SQLintersection and/or #DEVintersection
- Join us Wednesday Evening for SQLafterDark
  - Doors open at 7:00 pm
  - Trivia game starts at 7:30 pm Winning team receives something fun!
  - Raffle at the end of the night Lots of great items to win including a seat in a five-day SQLskills Immersion Event!
  - The first round of drinks is sponsored by SentryOne and SQLskills

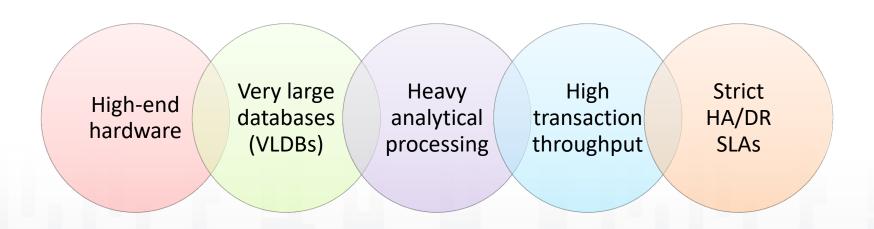




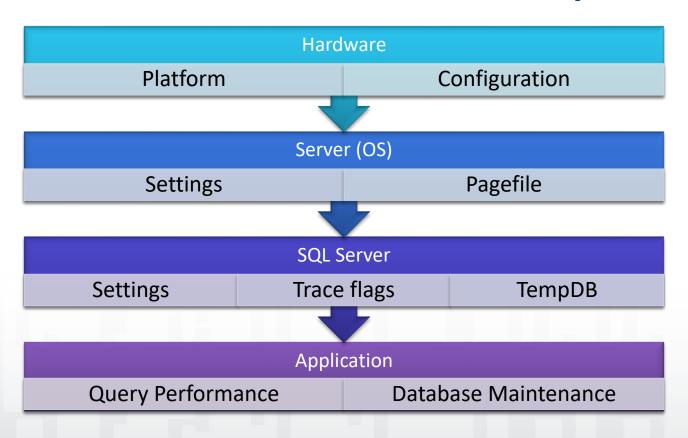


# What's a tier-1 SQL Server?

In short – a server where performance really counts



# Performance considerations at every level



# **Hardware**



# **Platform**

Physical

Virtual

Infrastructure as a Service (IaaS)

Platform as a Service (PaaS)

# **Physical Server**

# Know your workload

- OLTP: Large number of fast cores, large amount of memory, disk may not be as critical
- OLAP: Individual core clock speed not as critical, large amount of memory, high-end disk subsystem

#### Disk considerations

- Type of storage not as important as the specifications
- Size the storage subsystem based on number of IOPS and latency requirements, not just data size
- For very high-end systems, consider innovations in I/O such as persistent memory
- Keep in mind SQL Server does not yet support 8K sector size drives, support will be coming in a future release

# **Virtual Server Considerations**

Same recommendations as physical servers for resource allocation

Minimal performance penalty, if any

More flexibility for HA/DR solutions

# Configuration of host server is key

- Don't overcommit resources
- Ensure Power Management is set to High Performance
- Take care to configure vNUMA appropriately

# Infrastructure as a Service Considerations

### Know your workload, and tune before you go

- Throwing hardware at the problem no longer cost effective
- Streamline application before migration if possible

#### Sizing recommendations

- DS3\_v2 or higher for SQL Enterprise Edition
- DS2\_v2 or higher for SQL Standard Edition

# I/O will likely be the biggest challenge

- Use Premium Storage
- Know your IOPS and stripe across multiple disks to achieve higher throughput
- Implement DB I/O optimizations such as page compression and instant file initialization
- Other best practice recommendations can be found here: https://aka.ms/SQLlaaSPerf

# Server (OS)



# **Server (OS) Configuration**

#### Apply the latest version of the OS if possible

#### Keep up to date with OS updates

#### Power plan must be set to High Performance

- Sets CPU clock at full-speed (if not restricted in BIOS)
- On Intel CPUs enables temporary overclock if needed (a.k.a. Turbo Boost)

#### Windows Paging File

- Don't waste disk space, huge page file is not required
- If SQL Server memory is being paged out (Error 17890: A significant part of SQL server process memory has been paged out), add more memory

#### NTFS configuration

- Format data and log drives in NTFS 64-KB blocks
- Alignment is automatic starting with Windows 2008, review if partitions were upgraded
- Turn off NTFS file encryption and compression

# **SQL Server**



# **Core Licensing vs. CAL Licensing**

Make sure you install the correct version of the binaries

For core based licensing, install Enterprise Core to use all the cores on the server

Watch for the following message in the error log:

SQL Server detected 2 sockets with 12 cores per socket and 24 logical processors per socket, 48 total logical processors; using 20 logical processors based on SQL Server licensing.

# **Core vs. CAL Licensing in Setup :: SQL Server 2019**

Product Key MICROSOFT PRE-RELEASE SOFTWARE LICENSE TERMS License Terms MICROSOFT SQL SERVER 2019 COMMUNITY PREVIEW 3 (CTP3) Global Rules These license terms are an agreement between Microsoft Corporation (or based on where you Product Updates live, one of its affiliates) and you. Please read them. They apply to the pre-release software Install Setup Files named above, which includes the media on which you received it, if any. The terms also apply Install Rules to any Microsoft Feature Selection Feature Rules Feature Configuration Rules Print Ready to Install I accept the license terms and Privacy Statement Installation Progress Notice: A paid SQL Server edition product key has been provided for the current action - Enterprise. Complete Please ensure you are entitled to this SQL Server edition with proper licensing in place for the product key (edition) supplied. Warning: Enterprise Server/CAL license Product Key supplied. This edition limits SQL Engine CPU utilization to 20 physical cores, or 40 logical cores with Hyper-threading enabled. SQL Server detected 2 sockets with 12 cores per socket and 24 logical processors per socket. 48 total logical processors are available Check this box to acknowledge this limitation to continue or click Back/Cancel to enter an Enterprise Core product license that supports up to the operating system maximum.

## **NUMA**

Important for SQL Server memory object management and thread scheduling

#### If running in virtualized environment, emulate host NUMA settings

Avoid uneven and/or partially overlapping NUMA nodes

#### **Auto Soft-NUMA**

- Enabled by default on SQL Server 2016 and later, for servers with more than 8 processors in a single NUMA node
- SQL Server 2014 SP2 needs trace flag 8079
- Important to consider Soft-NUMA when calculating MAXDOP

### **Parallelism**

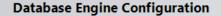
#### **MAXDOP**

- Default of 0 is rarely the correct setting
- Workload dependent testing is the only way to know the right setting
- Look for recommendations from vendor if running a third-party application
- No testing?
  - If not NUMA, set to number of logical processors, no higher than 8
  - If NUMA, set to number of logical processors in a single NUMA node, no higher than 16
- Starting with SQL Server 2016, can set this at the database level

## Cost threshold for parallelism

- Workload dependent testing is the only way to know the right setting
- No magic number so default value of 5 is fine. If you detect problems, test no blind changes
- Consider making a larger number if you want to favor OLTP over OLAP workloads on the same server

# Parallelism in Setup :: SQL Server 2019



Specify Database Engine authentication security mode, administrators, data directories and TempDB settings.

Product Key

License Terms

Global Rules

**Product Updates** 

Install Setup Files

Install Rules

Feature Selection

Feature Rules

Instance Configuration

PolyBase Configuration

Server Configuration

#### **Database Engine Configuration**

Consent to install Microsoft R ...

Consent to install Python

Feature Configuration Rules

Ready to Install

Installation Progress

Complete

Server Configuration | Data Directories | TempDB | MaxDOP | FILESTREAM

When an instance of SQL Server runs on a computer that has more than one CPU logical core, it detects the best degree of parallelism, that is, the number of processors employed to run a single statement, for each parallel plan execution. MAXDOP specifies the maximum number of cores to utilize for this instance.

Detected logical CPU cores on this computer: 24

Maximum degree of parallelism (MaxDOP)\*: 12

\* The displayed default value was either calculated by Setup, or was explicitly specified on the Setup command line with the /SQLMAXDOP parameter.

You can modify the MaxDOP here to be used as the default in all query executions for this instance, unless overridden at the query level. To suppress parallel query plan generation, set MaxDOP to 1.

See Configure the max degree of parallelism Server Configuration Option for more information.

< Back

Next >

Cancel

# **Memory**

#### Max server memory

- Common rule of thumb is 85%, but this is not appropriate for all workloads
- Not all SQL Server memory falls within max server memory limit
- If running things like SSIS or SSAS on the same server, adjust accordingly to prevent SQL
   Server from stepping on other processes and vice-versa

For an x64 server with 32GB RAM and 32 logical CPUs:

2 MB stack size \* 960 default worker threads on 32 CPU machine
+ 4 GB for OS
= roughly 6 GB
Max server memory = 26 GB

# **Memory**

#### Max server memory

- Common rule of thumb is 85%, but this is not appropriate for all workloads
- Not all SQL Server memory falls within max server memory limit
- If running things like SSIS or SSAS on the same server, adjust accordingly to prevent SQL
   Server from stepping on other processes and vice-versa

#### Min server memory

- Default is fine for most workloads
- Consider setting higher for the following conditions
  - Running on a virtual server with an overcommitted host
  - Running with other workloads on the same server
  - Running multiple instances of SQL Server on the same server (FCI for example)

# **Memory in Setup :: SQL Server 2019**

#### **Database Engine Configuration**

Specify Database Engine authentication security mode, administrators, data directories, TempDB, Max degree of parallelism, Memory limits, and Filestream settings.

Product Key

License Terms

Global Rules

**Product Updates** 

Install Setup Files

Install Rules

Feature Selection

Feature Rules

Instance Configuration

Server Configuration

**Database Engine Configuration** 

Feature Configuration Rules

Ready to Install

Installation Progress

Complete

the SQL Server Memor server memory.	y Manager for this instand	e, by specifying mir	n server memory a <mark>nd</mark> /or max
	○ <u>R</u> ecommended	<ul> <li>Default</li> </ul>	
Min Server Memory (N	IB): 0	0	
Max Server Memory (N	1B): 124544	2147483647	
	re explicitly specified in the	Setup command line	your system configuration and using the /SQLMINMEMORY

# **Memory – Large Pages**

# Lock pages in memory (LPIM)

- Good idea to set on most servers (and another reason to set Max Server Memory)
- Needed for large pages

# Large page allocations

- Enable with TF 834 to set large-page allocations for the buffer pool, columnstore and in-memory OLTP
- Good for large memory machines with analytical workloads
- Not supported with columnstore indexes in SQL Server 2019, use TF 876 instead (preview) to set large-page allocations for columnstore only, without SQL permanently hanging on to the memory

# **I/O Considerations**

#### Turn on instant file initialization

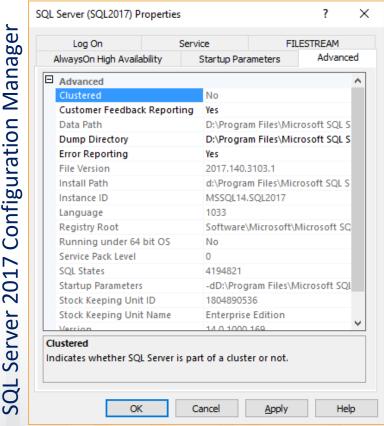
- Starting with SQL Server 2016, leverage SQL Server Setup to enable
- Starting with SQL Server 2019, leverage SQL Server Configuration Manager to enable (for all versions)

# File placement

- Still recommend separate logical drives for data, transaction log and tempdb, even on a SAN
- No need for multiple data files in the same filegroup unless spreading across multiple VHDs, or hitting allocation contention on a user database (similar to TempDB)
- If using multiple files per filegroup, ensure <u>all</u> are of equal size proportional fill algorithm
- For IaaS, use Storage Spaces to create logical partitions across multiple disks rather than database files

# Φ $\vdash$ 0 N rve ū

# **Setting LPIM and IFI in Config Manager**



#### SQL Server (SQL2017) Properties × Φ Log On Service FILESTREAM Manage Always On Availability Groups Advanced Startup Parameters ☐ Advanced Clustered No Customer Feedback Reporting onfiguration Data Path D:\Program Files\Microsoft SQL S Dump Directory D:\Program Files\Microsoft SQL S Error Reporting Yes File Version 2017.140.3103.1 d:\Program Files\Microsoft SQL S Install Path MSSOL14.SOL2017 Instance ID Instant File Initialization Yes Language 1033 Lock Pages In Memory Yes Registry Root Software\Microsoft\Microsoft SQ 9 Running under 64 bit OS No Service Pack Level 0 2 SQL States 4194821 Startup Parameters -dD:\Program Files\Microsoft SOI rve Stock Keening Unit ID 180//800536 Instant File Initialization Sel Grants Perform Volume Maintenance Task privilege to SQL Server Database Engine Service Account. SQL Cancel Apply Help

# **TempDB**

#### Fast storage

- No need for redundancy, optimize for performance (avoid RAID 5 and 6)
- Consider flash, SSD or even PMEM

#### Multiple data files

- Start with one data file per logical processor up to 8
- Add more in multiples of 4 files until you no longer see contention
- Files must be equally sized
- If adding new files, be sure to restart SQL Server to rebalance across all the files
- Starting with SQL Server 2016, leverage SQL Server Setup to configure

#### Upgrade to the latest version to get all the new fixes and enhancements

- Latest CU for SQL Server 2016 and 2017
- SQL Server 2019 CTP 3.0 to try memory-optimized metadata and cache improvements

# **Trace Flags**

#### For SQL Server 2016, 2017 and 2019

- 174: SOS\_CACHESTORE spinlock contention, KB 3026083
- 3468: Disable indirect checkpoints on tempdb, KB 4040276
- 7471: Parallel statistics update jobs, KB 3156157
- 2566: Faster CHECKDB command, KB 945770
- 7412: Enabled lightweight profiling, KB 3170113 (not needed in SQL Server 2019)
- 4199: Optimizer fixes (Optional, test workload first, can also be set via a database-level configuration)

#### For SQL Server 2012 and 2014, all of the above plus

- 1117: Autogrow all files in the same filegroup at the same time (tempdb, KB 2154845)
- 1118: Turn off mixed page allocations (tempdb, KB 2154845)
- 2371: Enable linear recompilation threshold for large tables, KB 2754171
- 6498: Avoid compilation waits for concurrent large queries, KB 3024815
- 6532,6533,6534: Spatial query performance, KB 3107399
- 8048,8079: CMEMTHREAD waits for systems with more than 8 cores per NUMA node
- 8075: VAS fragmentation, KB 3074434

# **Application**



# **Partitioning?**

#### A perfect fit for VLDBs

- Sliding windows scenarios
- Fast archiving
- Allows partition switching and optimal filegroup backup/restore (read-only data)

#### Find an optimal partitioning scheme based on the existing business requirements

- Examine the partitioning function, the data in the table, and the business uses of the data
- When partitioning any index (including clustered), the index key must contain the partitioning column
- If a natural partitioning key is available, use it (date, incremental ID)

#### Leverage partition-aligned indexes

- Partitioned indexes, functions, and schemes should have the same partitioning key and ranges
- To enable partition switching, all indexes on the table must be aligned
- Use the partitioning key in your query predicates (WHERE / JOIN) to encourage partition elimination
- Beware of queries with TOP, MIN/MAX as these must scan all partitions

# **Collations**

#### Beware of user databases with collation different from system databases

 Can lead to comparison issues such as "Cannot resolve the collation conflict between SQL\_Latin1\_General\_CP1\_CI\_AS and Latin1\_General\_CI\_AI in the equal to operation"

#### In SQL Server 2019, leverage the new UTF-8 collations

- UTF-8 is enabled when creating or changing an object's collation to a collation with the "\_UTF8" suffix, such as LATIN1\_GENERAL\_100\_CI\_AS\_SC to LATIN1\_GENERAL\_100\_CI\_AS\_SC\_UTF8
- UTF-8 is only available to Windows collations that support supplementary characters (\_SC), as introduced in SQL Server 2012
- CHAR and VARCHAR datatypes store UTF-8, NCHAR and NVARCHAR continues to use UCS-2 or UTF-16 encoding only and remain unchanged
- If using ASCII, significant storage savings may also be achieved.

  For example, changing an existing column data type from NCHAR(10) to CHAR(10) using an UTF-8 enabled collation, translates into nearly 50% reduction in storage requirements. This is because NCHAR(10) requires 22 bytes for storage, whereas CHAR(10) requires 12 bytes for the same Unicode string

## **Database Maintenance**

#### Maintain your indexes

- This may mean Rebuild or Reorganize depending on current level of fragmentation
- Workloads that scan are more vulnerable to performance issues with fragmented indexes
- Leverage intelligent management. Ex: Adaptive Index Defrag (<a href="http://aka.ms/AID">http://aka.ms/AID</a>)

#### **Update statistics**

- As needed, so leverage intelligent management. Ex: Adaptive Index Defrag (<a href="http://aka.ms/AID">http://aka.ms/AID</a>)
- Even more relevant if not using AUTO\_UPDATE\_STATISTICS

#### Implement Integrity checking

- DBCC CHECKDB runs depends on the individual business needs and the importance of the information in the database
- At minimum, run against all production databases at least once a week and review results
- Not running means not knowing of early signs of corruption, and can lead to data loss

# **Query Optimizer / Cardinality Estimation**

# Every new version introduces functional changes to the QO under the <u>latest</u> Database Compatibility Level

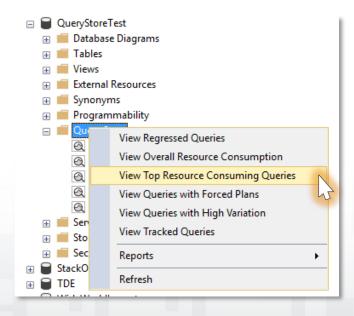
- Upgrading and staying in source Database Compatibility level is fully supported
- Still plan to certify ASAP on latest; New work should start on newer Database Compatibility
- Evaluate use of TF 4199 / DB scoped config "Query Optimizer Fixes"

#### A new version introduces a new CE version

- Mechanism for estimating required number of rows to satisfy query, using statistical models and heuristics
- No more "Legacy CE" vs "New CE" only
  - "Legacy CE" = CE 70, default for SQL Server 7.0 to 2012
  - "New CE" = CE 120 (SQL Server 2014) through CE 150 (SQL Server 2019)
- It's <u>recommended</u> to follow documented upgrade process to avoid increased risk of regression if upgrading Database Compatibility! Use Query Tuning Assistant in this case

# **Query Store**

Comprehensive query-performance information when you need it most!

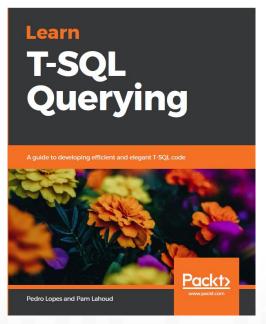


## **Learn more**

https://aka.ms/sqlworkshops

Download and try SQL	https://aka.ms/ss19	
Server 2019	https://aka.ms/SQL2019WhatsNew	
Check out these great	https://aka.ms/DataDemos	
data-related demos	https://aka.ms/IQPDemos	
	https://aka.ms/SQL2019Notebooks	
Continue learning with	https://aka.ms/LearnTSQLQuerying	
our new book	https://aka.ms/LearnTSQLQuerying_errata	
One shortcut to rule them all!	https://aka.ms/SQLShortcuts	

Use our free training



# Thank you!





# **Questions?**

Don't forget to complete an online evaluation!

Practical Guidance to Make Your Tier-1 SQL Server ROAR!

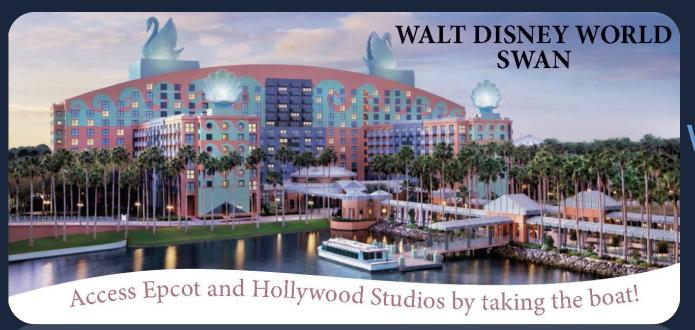
Your evaluation helps organizers build better conferences and helps speakers improve their sessions.



Thank you!

# Save the Date!

# www.SQLintersection.com



2020

Week of April 6

We're back in Orlando!



Access Epochana Honywood Studios by taking the boat!

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!