Apologies
French speaker

@conseilit
/christophelaporte
conseilit@outlook.com

~ since 1997: SQL 6.5 / WinNT4

MVP
Microsoft Certified Master
Microsoft Certified Trainer

MS SQL Server
Toulouse – France
Our sponsors

[Logos of various sponsors]
Agenda today

Quick recap on hardware
Host Configuration
SQL Server and virtualization
Windows Server configuration
SQL Server installation
SQL Server configuration
SQL Server maintenance
Quick recap on hardware - CPU

Frequency vs # cores

SQL Server editions

- Impact on Licensing
  Enterprise: OS Max
  Standard: 24 cores / 4 socket
  Web: 16 cores / 4 sockets

12 cores?
- Xeon Gold 5118 @2.3 Ghz
- Xeon Gold 6126 @2.6 Ghz
- Xeon Gold 6136 @3.0 Ghz
- Xeon Gold 6146 @3.2 Ghz
- Xeon Platinum 8158 @3.0Ghz
Quick recap on hardware - memory

Huge impact on performance
  low cost performance improvement!
  No impact on licensing fees 😊

But some limitations based on sku 😞
  Express (<2016) : 1GB
  Express (>=2016) : 1,410 GB +
    352MB CSI +
    352MB per DB using Hekaton
  Standard (2012) : 64 GB
  Standard (2014) : 128 GB
  Standard (>=2016SP1) : 128 GB +
    32GB CSI +
    32GB per DB using Hekaton
  Enterprise (OS Limit) : 24 TB

No “On Size fits all” configuration
  Based on the instance workload
  And on the working set

But, please ...
  At least 6 to 8 GB per core
Quick recap on hardware - Disks

Time to say goodbye to spinning disks
Flash drives are your best friends now!

(Very) low latency
Less CPU needed (19% vs 37% for 100 000 IOPS 100% 4K reads)
NVMe is even faster than SAS/ SATA SSD

Designed for flash drives
AHCI: 1 queue & 32 commands per queue
NVMe: 64K queues & 64K commands per queue

Think different

$ per IOPS instead of $ per GB

<table>
<thead>
<tr>
<th>Operation</th>
<th>Duration</th>
<th>IOSize</th>
<th>IOType</th>
<th>PendingIO</th>
<th>FileSize</th>
<th>IOPS RAID SSD</th>
<th>MBs/Sec</th>
<th>Min_Lat(ms)</th>
<th>Avg_Lat(ms)</th>
<th>Max_Lat(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>10</td>
<td>8</td>
<td>Random</td>
<td>8</td>
<td>20000</td>
<td>161 955,10</td>
<td>1 265.27</td>
<td>0</td>
<td>0</td>
<td>47</td>
</tr>
<tr>
<td>Write</td>
<td>10</td>
<td>8</td>
<td>Random</td>
<td>8</td>
<td>20000</td>
<td>61 224,10</td>
<td>478,31</td>
<td>0</td>
<td>1</td>
<td>36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation</th>
<th>Duration</th>
<th>IOSize</th>
<th>IOType</th>
<th>PendingIO</th>
<th>FileSize</th>
<th>IOPS Intel P3700</th>
<th>MBs/Sec</th>
<th>Min_Lat(ms)</th>
<th>Avg_Lat(ms)</th>
<th>Max_Lat(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>30</td>
<td>8</td>
<td>Random</td>
<td>8</td>
<td>5000</td>
<td>293 653,32</td>
<td>2 294,16</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Write</td>
<td>30</td>
<td>8</td>
<td>Random</td>
<td>8</td>
<td>5000</td>
<td>131 564,20</td>
<td>1 027,84</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>
Host configuration

Install latest version for
Bios
Firmware
Drivers

BIOS configuration
Profile setting: max performance
Enable Hyper-threading
Enable Intel turbo boost

Host OS
Power setting high performance
Virtualizing SQL Server

Yes .. Of course
Dev / Test / Production
But things should be done the right way

CPU
- Disable Hot Add CPU
  #sockets vs #cores (Next slide)

Memory
- NO dynamic memory
  Set reservation = allocated memory

Network
- VMXNet3 mandatory (latency and performance)

IO subsystem for best performance
- Add more paravirtual SCSI controllers (pvSCSI)
- Attach multiple VMDKs to each controller
  Thick provisioning eagerly zeroed

Disable unused devices
- Floppy, DVD, COM, USB, LPT
NUMA Architecture

NUMA considerations

« Remote » or « foreign » memory access 2 to 5 times slower than local memory

# sockets vs # cores

/!
\ licensing SQL Web / Standard edition
4 sockets

vNUMA

Disabled by CPU Hot Add option
By default enabled only if > 8 cores

Recommendation

Try to fit the VM on a NUMA node
```sql
SELECT * FROM sys.dm_os_schedulers
```
Windows Server configuration

Security
  Windows admins
    Very strong passwords!
  Limit RDP access to OS
  Windows Serveur Core?
    Change default RDP tcp port

Network
  Enabling RSS?
  Increase max port number
  Increase SMB Timeout

Page file
  Why large file?
  4GB max!

Power option
  High performance

Antivirus exclusions
  Mandatory *.MDF *.NDF *.LDF
  Potentially *.BAK *.TRN
Windows Server configuration

Volumes
- Naming rules
- Mount points
- Text file in the root folder
- Quickly Identify the volume

Formatting
- GPT / MBR
- NTFS 64K
- Disable Indexation
- Disable 8.3
- Disable Last Access
- LargeFRS

```powershell
function FormatVolumes () {
    # Online disks
    Get-Disk | where-Object IsOffline -Eq $True | Set-Disk -IsOffline $False
    $diskList = Get-Disk | where-Object partitionstyle -eq "raw"
    foreach ($currentDisk in $diskList) {
        # affect volume label
        switch ($currentDisk.Number) {
        ...
        Get-Disk $currentDisk.Number | Initialize-Disk -PartitionStyle GPT
        $part = Get-Disk $currentDisk.Number | new-Partition -UseMaximumSize -AssignDriveLetter
        $part | Format-volume -FileSystem NTFS -AllocationUnitSize 65536 -ShortFileNameSupport:$false
        -Confirm:$false -NewFileSystemLabel $diskLabel -UseLargeFRS | Out-Null
    }
}

# for each drive, disable indexing
$driveList = Get-wmiobject -class Win32_Volume | where-object label -like '*SQL2'
foreach ($currentDrive in $driveList) {
    $indexing = $currentDrive.IndexingEnabled
    if ($indexing -eq $true) {
        $currentDrive | set-wmiinstance -arguments @{IndexingEnabled=$false} | Out-Null
    }
}

Get-wmiobject -class Win32_Volume | select-object name,label,indexingEnabled,blocksize,filesystem | where-object label -like '*SQL*' | format-table -autosize
```
SQL Server installation

Next Next Next ???

Better now than previous versions

Default or named Instance

No matters

Collation

Should satisfy business goals

Sysadmin accounts

Windows groups instead of users

Service account

For each service

Lock Page In Memory

Perform Maintenance Volume Tasks (IFI)

Kerberos

Allow read / write SPN : DSACLs

Fine for double hop authentication
SQL Server Configuration

Min / Max server Memory = Total OS memory
- Minus 1GB for OS
- Minus 1GB for each 4 GB block from 4 to 16 GB
- Minus 1GB for each 8 GB block beyond 16 GB

Optimize for adhoc workload
Default backup compression
Default backup checksum
  - Configuration since 2014 +
  - TF3023 for older versions of SQL Server
Remote Admin Connection

Network packet size
  - For large data movement through linked servers
Min memory per query
  - Default 1MB
  - Can be reduced to 512KB according to the workload
**SQL Server Configuration**

By default no CPU affinity
- But if you do so
  - Add TF8002

Cost threshold for parallelism
- 5 is really ... bad
- 25, 35 or even 50 is better
- Can adjust the value live

MaxDop
- Depends on NUMA architecture
  - Basic rule
    - MaxDop = #core in NUMA node
  - Or adjust
    - Accordingly to software editor requirements
    - Accordingly to the workload (BI vs OLTP)
Post-installation

Adjust TempDB (SQL2014-)
   # of files
   Same size and auto growth
Increase the size of MSDB Database
Eventually alter Model Database
Increase the # of Errorlog files
Adjust System_Health xEvent retention
   increase # of files and / or file size
   for better troubleshooting experience

Configure Database Mail
   Profile
   Accounts

SQL Agent
   Configure jobs history
   Configure mail profile
   Create operators
   Create basic alerts
Post-installation – Trace Flags

Trace Flags
Some basic ones

834 : large page allocations (do not use if columnstore indexes)
1117 : auto growth all files simultaneously
1118 : remove single page allocations
2371 : update statistics threshold
3226 : Remove every successful backup message
7806 : Enable DAC on SQL Server Express
7412 : lightweight query execution statistics profiling
   Might involve 2% CPU overhead
   Valid for SQL Serve 2017 and SQL Server 2016 SP1
   Enabled by default on SQL Server 2019

And also

272 : SQL2012+ : no gap for identity after restart or failover
PowerShell is your friend

Scripting all the configuration steps
  Time saving
  Ease to standardize configuration

PowerShell
  SQL Server specific cmdlets
  SMO library available

dbatools.io
  DBA must-have toolbox
  Hundreds PowerShell modules
  Migration, configuration, administration
Post-installation - Security

Hide instance?
Enable SQL Browser in case of named instance?
Change TCP port for default instance?
Always keep Windows Firewall enabled
   And add required rules
Disable SA account?
Remove all files in the Setup Bootstrap folder?
SQL Server maintenance

Maintenance routine
- Backup (database, differential, transaction log)
- Integrity checks, on primary and secondary for AGs
- Index maintenance according to fragmentation level
- Index statistics updates

Archive LOG folder
- xEvents files
- Errorlog files
- Default trace

Test your backups!
- Daily / Weekly restore critical databases
- A DBA is ranked on restore, not backup

SQL Server maintenance plans
- performs well
- But are difficult to migrate
- And could be more customizable

Ola Hallengren scripts
- Definitively a standard
- Highly customizable

NEVER collocate data and backups on the same disk array
Quick database settings considerations

Recovery model
  Depends on RPO
Auto Shrink
  Always False, ALWAYS
Auto Close
  Always False
Read Committed Snapshot Isolation
  seriously consider
Delayed durability
  might give it a try
Containment

Database scoped configuration
  MAXDop
  Legacy Cardinality Estimator
  Parameter sniffing
  Query optimizer fixes
Conclusion

Do not trust power savings
As a baseline

No Windows deep configuration changes
  Difficult to maintain / might change with OS upgrades

No fancy Trace Flag or SQL Server configuration option
  SQL Server is fast
    Good HW choices, well understanding of SQL Server features (CI, Hekaton, Delayed Durability ...)
  Highlight your DBA skills by right indexing your DBs/improving T-SQL statements
    Much more efficient than changing an improbable parameter somewhere in SQL Server

KISS
  Keep It Simple (Stupid)

The future
  SQL Server in containers / K8s cluster : same configuration options ?
  No configuration available on Azure SQL Database at instance level !
Thank you for attending

Q&A