SQLintersection

Scaling SQL Server

Glenn Berry
glenn@sqlskills.com

SQL
intersection

Glenn Berry

- **Consultant/Trainer/Speaker/Author**
- **Principal Consultant, SQLskills.com**
  - Blog: http://www.SQLskills.com/blogs/Glenn
  - Twitter: @GlennAlanBerry
  - Regular presenter at worldwide conferences on hardware, scalability, and DMV queries
  - Author of SQL Server Hardware
  - Chapter author of Professional SQL Server 2012 Internals and Troubleshooting
  - Chapter author of MVP Deep Dives Volumes 1 and 2
- **Instructor-led training: Immersion Events**
- **Online training: pluralsight** http://pluralsight.com/
- **Consulting: health checks, hardware, performance, upgrades**

Agenda

- Scalability issues with SQL Server 2014
- Postponing the scaling decision
- Hardware and scalability
- Scaling up SQL Server 2014

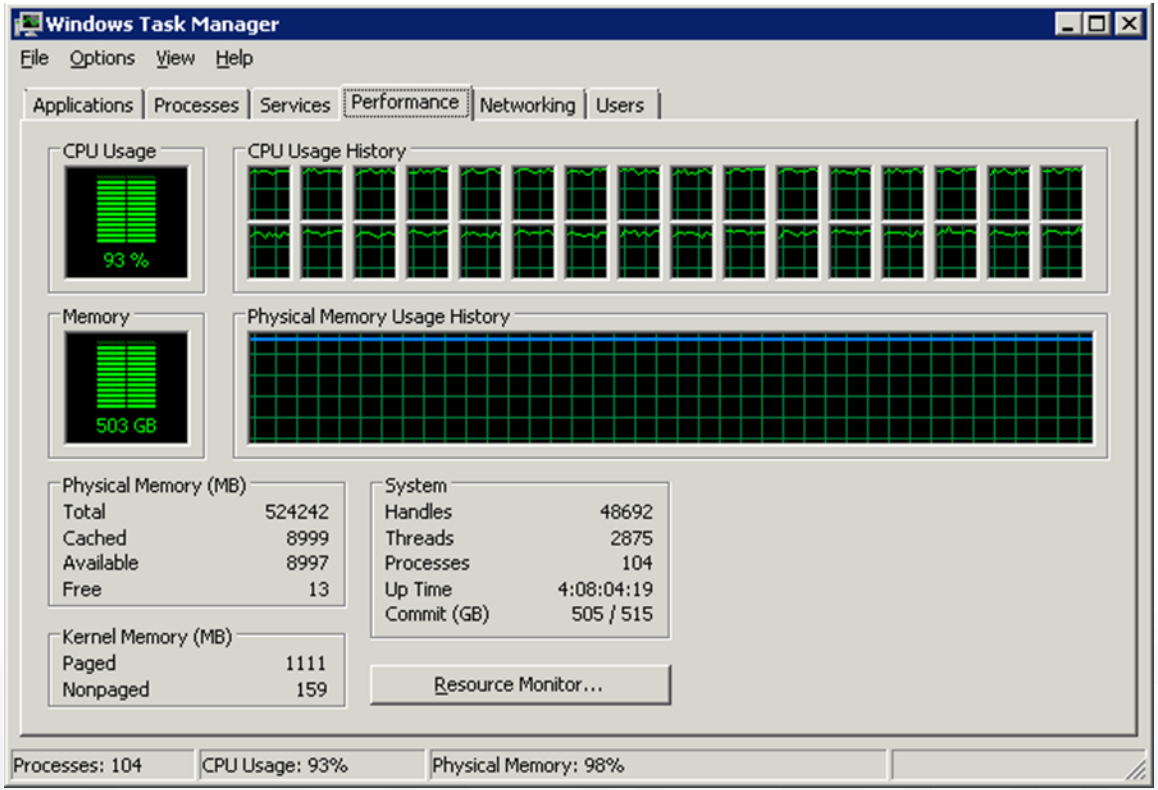
SQL intersection

Introduction

- **SQL Server 2014 scales very well**
  - Scaling issues are usually caused by non-SQL Server issues
- **SQL Server 2014 can handle nearly any workload**
  - By using the right features and techniques
  - By using the appropriate hardware for the workload
  - By scaling up
  - By scaling out
- **SQL Server 2014 will scale if you know what you are doing**
  - SQL Server 2014 is “enterprise-ready”

SQL *intersection*

This is Your Nightmare on 32 cores…



SQL intersection

Top Database Scalability Issues

- Poor application architecture and design
- Poor database architecture and design
- Poor indexing strategy and maintenance
- Incorrect OS and configuration settings
- Incorrect database configuration settings
- Inadequate storage subsystem performance
- Old or inappropriate hardware

Common Application Design Issues

- **Overuse of ORM code/query generators**
- **Using ad-hoc SQL from the application**
- **Overuse of XML data**
- **Using SQL Server UDFs (especially scalar UDFs)**
  - Often evaluated once per row, not once per query
- **Including too much business logic in the database**
  - Especially if it is complicated business logic
- **Overuse of DML triggers for application logic**
- **Overuse of CLR assemblies**
- **Using built-in T-SQL functions in JOIN and WHERE clauses**
  - Can negate an index seek

Application Refactoring/Optimization

- **This is extremely important!**
  - Defending SQL Server from bad applications…
  - Huge improvements possible in this area when you have available resources
- **You can use DMV queries, SQL Server Profiler and Extended Events**
  - Finding the most costly queries and stored procedures
  - Finding the the most frequently executed queries and stored procedures
  - Finding application logic issues that cause extra stored procedure calls
- **Unfortunately, you may have no control over the application(s)**
  - 3rd party applications
    - Many ISV applications have serious database usage issues
  - Microsoft applications that use SQL Server as a data store
    - SharePoint, Lync, System Center Team Foundation Server, etc.
  - Internal applications where the development team has other priorities

SQL Server is a Database Server!

- **Its job is hard enough already**
- **It can be the single most expensive scalability bottleneck**
  - Let the application and web servers do as much work as possible
  - Just because you can do something in SQL Server does not mean you should
  - Avoid the temptation to use new SQL Server features "just because"
- **Don't use SQL Server as an application server**
  - Don't incorporate too much complicated business logic in stored procedures
  - Avoid multi-page T-SQL stored procedures
  - Avoid multi-purpose T-SQL stored procedures
  - Avoid CLR assemblies (unless there is no other way to do something)
  - Try to use SQL Server mainly for simple CRUD operations

Common Database Design Issues

- Poor normalization
- Lack of check constraints, foreign keys
- Heap tables
- Inappropriate data types
- Wide data types
- Poor query design
- Using ad-hoc SQL instead of stored procedures
- Using SQL Server default database properties
- Data and log file physical layout

Data and Log File Physical Layout

- **High percentage of databases have one data file in PRIMARY file group**
  - This complicates management and can hurt performance
- **Best practice to create a MAIN file group and make it the default file group**
  - Create two-four data files in the MAIN file group
    - This gives you more flexibility about where to deploy the data files
  - Create all user objects in the MAIN file group (so system objects are in PRIMARY)
  - Make sure to enable instant file initialization (only works for data files)
- **No reason (outside of an emergency) to have more than one log file**
  - Make sure to manually grow the log file to desired size in 4000MB or 8000MB increments to minimize your VLF counts
  - Set log file autogrow size to 4000MB or 8000MB (unless you have a very slow storage subsystem)
  - If your storage subsystem is very slow, consider setting autogrow size to 1000MB or 2000MB

Common Indexing Issues 1

- **Too many indexes**
  - Don’t go wild with the “missing index” DMV
  - Be very careful with the Database Tuning Advisor
  - Consider your workload type and how volatile the data is
  - OLTP and DW workloads have different index requirements
- **Too few indexes**
  - Can cause CPU, memory, and I/O pressure
  - Can cause locking, blocking and deadlocks
  - Missing index DMV can help find candidate indexes
  - Missing index warnings in execution plan can also be useful
  - You can query the plan cache to find missing index warnings

Missing Index Warning Example

Microsoft SQL Server Management Studio

File Edit View Query Project Debug Tools Window Community Help

New Query | NoCompressionTest | Execute

Object Explorer

Connect

av-m4600-0784 (SQL Server 10.50.27

Databases
System Databases
Database Snapshots
InsertLoadTest
NoCompressionTest
Database Diagrams
Tables
System Tables
dbo.OnlineSearchHistory
dbo.OnlineSearchHistory
Views
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
Management
SQL Server Agent

SQLQuery5.sql - av...\glenn.berry (53))* | SQLQuery4.sql - av...\glenn.berry (52))*

```
SET STATISTICS IO ON;

-- Original query against non-compressed clustered index (40 seconds)
SELECT SearchTerm, NumItemsReturned, SearchElapsedTime
FROM dbo.OnlineSearchHistoryNonCompressed
WHERE SearchTerm = 'Jessica Alba'; -- SearchTerm column is nvarchar, so this is bad!

--Table 'OnlineSearchHistoryNonCompressed'. Scan count 9, logical reads 2334388, physical reads 34710,
--read-ahead reads 2331799, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

-- Original query against page compressed clustered index (9 seconds)
SELECT SearchTerm, NumItemsReturned, SearchElapsedTime
FROM dbo.OnlineSearchHistoryPageCompressed
WHERE SearchTerm = 'Jessica Alba'; -- SearchTerm column is nvarchar, so this is bad!

--Table 'OnlineSearchHistoryPageCompressed'. Scan count 9, logical reads 515733, physical reads 1098,
--read-ahead reads 514827, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.


-- Modified query
SELECT SearchTerm, NumItemsReturned, SearchElapsedTime
FROM dbo.OnlineSearchHistoryNonCompressed
WHERE SearchTerm = N'Jessica Alba'; -- literal value converted to unicode
```

Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT [SearchTerm],[NumItemsReturned],[SearchElapsedTime] FROM [dbo].[OnlineSearchHistoryPageCompressed] WHERE [SearchTerm]=@1
Missing Index (Impact 99.9906): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [dbo].[OnlineSearchHistoryPageCompressed] ([SearchTerm])…
```

SELECT Cost: 0 % ← Parallelism (Gather Streams) Cost: 4 % ← Clustered Index Scan (Cluste… [OnlineSearchHistoryPageComp… Cost: 96 %

Registered Servers | Object Explorer

Query executed successfully. | AV-M4600-0784 (10.50 SP1) | SANJUAN\glenn.berry (52) | NoCompressionTest | 00:00:09 | 0 rows

Ready

Common Indexing Issues 2

- **Not using SQL Server data compression where appropriate**
  - Enterprise Edition-only feature added in SQL Server 2008
  - Unicode compression added in SQL Server 2008 R2
  - No changes for SQL Server 2012/2014/2016
  - Consider your workload type and how volatile the data is
- **Trade less I/O utilization for some extra CPU utilization**
  - Often a huge net win if you are under I/O or memory pressure
  - Can reduce both I/O pressure and memory pressure
  - Table/index must be a good compression candidate
    - Large, relatively static, with a good compression ratio
- **Provides some protection from expensive range scan queries**
  - Clustered index scan is not as expensive if it is compressed
  - Much less data has to be read from the storage subsystem
  - Compressed index takes up less space in the buffer pool

Common Indexing Issues 3

- **Poorly maintained indexes**
  - Fragmentation causes extra I/O for range scans
  - Also affects the costing of index operations (many pages inflate operator cost)
  - Does not affect singleton select performance very much
  - Wastes space in data files
- **Overly maintained indexes**
  - Don't rebuild/reorganize indexes unless they need it
    - This adds a lot of extra resource pressure to your system
  - Don't use Maintenance Plan Wizard to maintain indexes!
    - It makes it too easy to select bad combinations of operations
  - Many great scripts available to maintain indexes
    - Ola Hallengren's SQL Maintenance Solution
      - http://ola.hallengren.com/

Common Indexing Issues 4

- **Using Spatial or XML indexes**
  - Cannot be built or rebuilt in online mode
  - XML shredding can be costly and CPU-centric
  - Both XML and Spatial indexes often perform poorly
    - Troubleshooting Spatial Query Performance: http://bit.ly/15CbEli
  - Spatial indexes are easily fragmented
- **Spatial index bug in older SQL Server 2008 R2 builds**
  - Spatial index often not used without using an index hint
  - Fixed in SQL Server 2008R2 RTM CU9
    - http://support.microsoft.com/kb/2570501
  - Spatial index performance issues in older SQL Server 2012 builds
    - Performance fixes in SQL Server 2012 SP1 CU7
      - http://bit.ly/19ojl0o

Operating System Issues 1

- **Not using Windows Server 2012 or Windows Server 2012 R2**
  - More secure, with better performance than older OS versions
    - SMB 3.0/3.02 gives much better network performance
  - Longer Microsoft support lifetime
  - Windows Server 2012 Standard Edition has full functionality for SQL Server
    - Supports 4TB of RAM
    - Supports Windows clustering feature (required for AlwaysOn AGs)
  - Windows Task Manager is much more useful in Windows Server 2012
- **Windows Server 2012 and SQL Server 2012 work better together**
  - Required for supporting more than 256 logical cores or more than 2TB of RAM
  - Low level optimizations for scalability and performance help all size machines
- **Windows Server 2012 R2 and SQL Server 2014 work better together**

Operating System Issues 2

- **Using older versions of Windows Server**
  - Windows Server 2003 and 2003 R2 are out of mainstream support
    - SQL Server 2005 is also out of mainstream and extended support
  - Windows Server 2008 and 2008 R2 are out of mainstream support
    - Mainstream support for both ended on January 13, 2015
      - http://bit.ly/1fOXFzT
    - Mainstream support for SQL Server 2008 and 2008 R2 ended on July 8, 2014
      - http://bit.ly/16Had4d
- **Older versions of Windows Server have lower license limits**
    - Windows Server 2008 was limited to 64 logical cores
    - Windows Server 2008 R2 Standard Edition was limited to 32GB of RAM
- **SQL Server 2012 requires at least Windows Server 2008 SP2**
    - Will not install on any older version of Windows Server

Power Management Effects

- **Power management has a significant negative effect on performance**
  - Can be in the 20-25% range, depending on the workload
- **Two types of power management**
  - Windows Power Plan and hardware-level (controlled in the main BIOS)
  - Hardware power management will override Windows Power Plan setting
- **Processor clock speed is reduced to save on electrical power usage**
  - Clock speed is increased in reaction to an increase in processor load (takes about 100ms)
  - The increase does not happen fast enough to avoid a performance decrease
- **PCIe slot voltage is also affected by power management**
  - Storage vendors warn about this issue (example: SanDisk Fusion-io)
- **Each newer generation of processor is less affected by this issue**
  - Nehalem and Westmere are the most serious. Sandy Bridge and Ivy Bridge are somewhat better
  - Haswell and Broadwell are better about power management

CPU-Z

CPU-Z

CPU | Caches | Mainboard | Memory | SPD | Graphics | Bench | About

Processor

Name	Intel Xeon E5 2690		
Code Name	Sandy Bridge-EP/EX	Max TDP	135.0 W
Package	Socket 2011 LGA		
Technology	32 nm	Core VID	1.196 V
Specification	Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz		
Family	6	Model	D	Stepping	7
Ext. Family	6	Ext. Model	2D	Revision	C2
Instructions	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x, AES, AVX		

Clocks (Core #0)

Core Speed	3299.23 MHz
Multiplier	x 33.0 (12 - 38)
Bus Speed	99.98 MHz
QPI Link	3999.07 MHz

Cache

L1 Data	8 x 32 KBytes	8-way
L1 Inst.	8 x 32 KBytes	8-way
Level 2	8 x 256 KBytes	8-way
Level 3	20 MBytes	20-way

Selection: Processor #1 | Cores: 8 | Threads: 16

CPU-Z Ver. 1.75.0.x64 | Tools | Validate | Close

SQL intersection

Configuring Your Operating System

- **Make sure to use Windows Server 2012 R2 or Windows Server 2016 when it is available**
  - They have higher license limits and a better support story
  - SMB 3.0/3.02 gives better network performance than 2.0 or 2.1
  - They also perform better than older versions of Windows Server
- **Grant the appropriate rights to the SQL Server Service account**
  - Perform Volume Maintenance Tasks
  - Lock Pages in Memory (LPIM)
- **Windows Page File**
  - Set it to a fixed, relatively small size (4096MB)
    - This allows SQL Server mini-dumps to occur
  - No need to have the page file be 1.5 times the physical RAM size
    - This is an outdated recommendation
  - If your database server is ever paging, you have severe external memory pressure!

Instance Configuration Settings 1

- **Windows User's Rights for the SQL Server Service Account**
  - Perform volume maintenance tasks right (SQL Server 2016 lets you do this in the setup program)
  - Lock pages in memory (LPIM) right
- **Optimize for ad-hoc workloads**
  - Should be enabled
- **Max degree of parallelism (MAXDOP)**
  - Set to the number of physical cores in a NUMA node
- **Cost threshold for parallelism**
  - Usually should be raised to a higher value than the default of 5
- **Max server memory setting**
  - Should be set to an appropriate, non-default value
- **Backup compression default**
  - Should be enabled unless you are under sustained CPU pressure

Instance Configuration Settings 2

- **Tempdb data files**
- **Number, size and placement**
  - Start with four to eight data files, look for allocation contention issues
  - More details: http://bit.ly/16la3n8
- **All tempdb data files should be the same size**
  - Otherwise you will not get equal usage of all the files
- **Use a dedicated, fast LUN to host tempdb data and log files**
  - Strongly consider using RAID 10 for tempdb
- **May be a good candidate for flash-based storage**
  - This depends on your workload!
  - Don't just assume that you need flash-based storage for tempdb

Database Property Settings

- **Autoclose**
  - Always disable!
- **Autoshrink**
  - Always disable!
  - Shrinking data files is very resource intensive
    - Causes severe index fragmentation. Paul Randal will hunt you down and make you eat haggis!
- **Auto update statistics**
    - Usually should be enabled, unless you have a very good reason not to
- **Auto update statistics asynchronously**
  - Usually should be enabled, especially for OLTP workloads
    - Memory leak bug in older builds is corrected in a recent CU
  - Gives you more predictable query performance for OLTP workloads
  - Enable trace flag 2371 to change autostats threshold
    - More details: http://bit.ly/qOAIqs (only in SQL Server 2008 R2 SP1 or greater)

Database Property Default Settings

- **Database initial size (4 MB data file, 1MB log file)**
  - Much better to start out with larger file sizes
  - You should manually, explicitly grow your data and log files to an appropriate size
- **Autogrow settings (1MB data file, 10% log file)**
  - Data files should be set to autogrow by a larger amount (make sure IFI is enabled)
  - Log files should grow by 4000MB or 8000MB to reduce VLF counts
    - This depends on your I/O subsystem performance, since they must be zeroed out
- **Recovery model (depends on SQL Server edition)**
  - Make sure full and transaction log backups are being run as needed
  - Runaway transaction logs are an extremely common novice DBA mistake
- **Make sure your default database file locations are appropriate**
  - Not on the C: drive!
  - This can be set during SQL Server installation or changed afterwards
  - Make sure the default database file locations actually exist
    - Otherwise, Service Pack and Cumulative Update installs will fail!

Postponing the Scaling Decision

- **Refactor/optimize the application if possible**
  - This may not be possible in many situations
- **Improve the database architecture and design**
  - The DBA often has more control and influence in this area
- **Improve the index situation**
  - Careful index tuning and improved index maintenance can make a huge impact!
- **Make sure the OS and instance settings are correct**
  - This is easy to do, and usually under your control
- **Make sure the databases are configured correctly**
  - This is easy to do and definitely under your control
- **Make sure the storage subsystem is configured correctly**
  - Example: cache settings, RAID level
- **Make sure the server hardware is configured correctly**
  - Example: Power Management, Turbo Boost, hyper-threading, virtualization support

Finding and Eliminating Bottlenecks

- **Find and correct instance and database-level bottlenecks**
  - Use my SQL Server Diagnostic Information queries to start
    - http://bit.ly/vL9vXA
  - Use standard best practices for query and application tuning
- **Use modern, properly sized and configured hardware**
  - Hardware must be appropriate for the workload
    - OLTP vs. DW vs. OLAP workloads
  - Storage subsystem must be adequate for the workload
    - One of the most common bottlenecks with many SQL Server workloads
    - Need to consider latency, IOPS capacity and sequential throughput capacity
- **Effort expended here can eliminate the need to scale up or out!**
  - Nearly every system has many available opportunities for improvement
  - This is where you earn your paycheck!

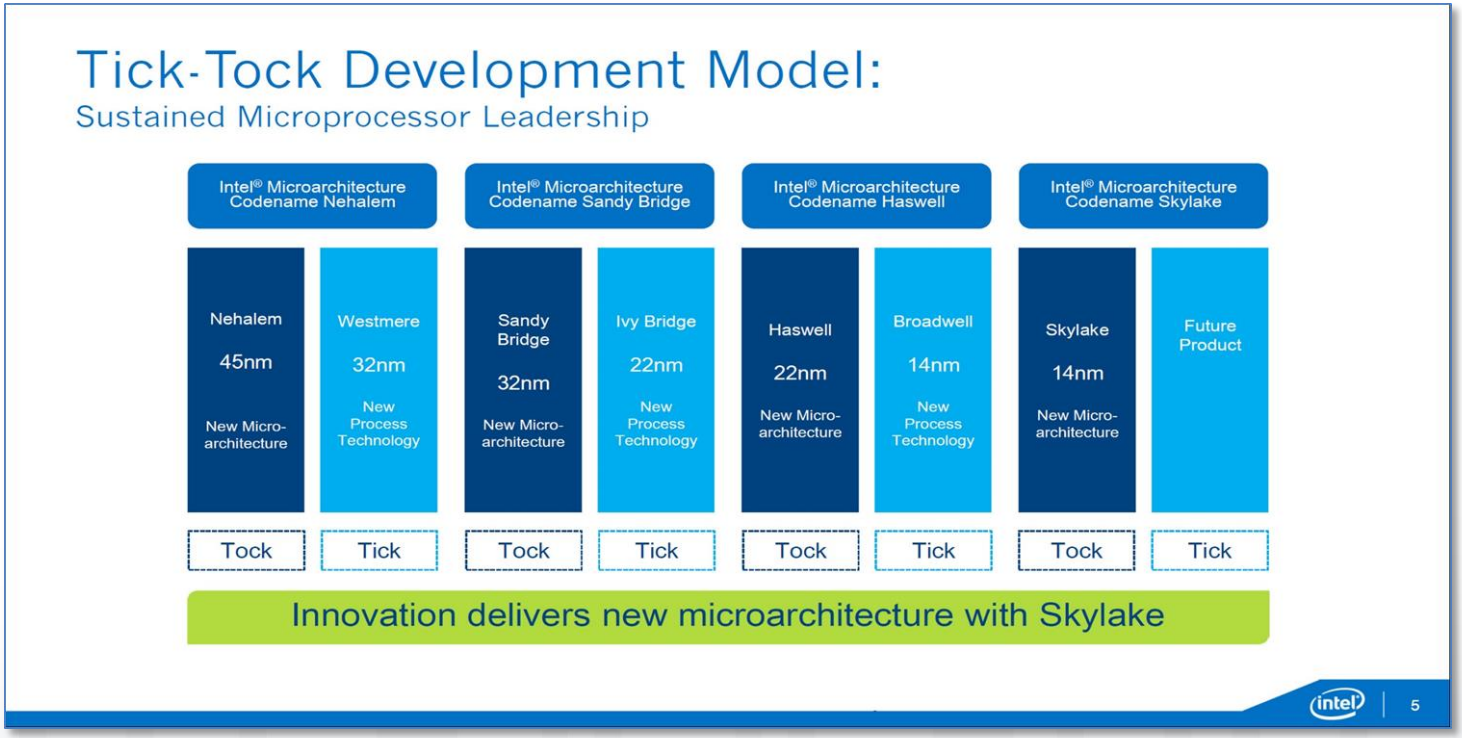
The Importance of Database Hardware

- **Database servers are mission critical assets that affect scalability**
  - Performance and scalability problems are immediately noticeable
  - Multiple applications typically depend on a single database server
- **Very difficult to compensate for poor hardware choices**
  - Inadequate I/O performance and capacity can cripple the system
  - Insufficient memory capacity can cause extra I/O pressure
  - Insufficient capacity hurts performance and scalability
- **Wise hardware selection can save on SQL Server license costs**
  - Physical core counts are cost driver for SQL Server 2014 core licensing
  - New two-socket servers can often replace older four-socket servers
  - It is possible to save so much on SQL Server license costs that your new hardware is free!

Hardware and Scalability

- **Use modern Intel hardware!**
- **Modern means Intel Xeon 5500, 7500 series or newer**
  - Nehalem-EP or Nehalem-EX or newer
  - Anything older will be SMP instead of NUMA
    - Out of warranty and severely underpowered by modern standards
    - Very possible that your laptop may have more CPU capacity than an older server
- **New two-socket servers can usually replace old four-socket servers**
  - Especially the latest Intel Xeon E5-2600 v4 series based servers
  - This saves on hardware costs and SQL Server 2014 license costs
  - Two-socket servers have much better single-threaded processor performance
  - Current model servers have PCIe 3.0 support
    - Double the bandwidth of PCIe 2.0 standard

Current Intel Tick Tock Release Strategy



SQL intersection

Server Hardware Evaluation and Selection

- **Very important to have relatively modern server hardware**
  - Intel-based servers have many performance and scalability advantages
- **Try to have Xeon E5 or newer (Sandy Bridge-EP or newer)**
  - PCIe 3.0 support, good memory capacity and performance
  - Up to 768GB of RAM with 32GB DIMMs in a two-socket server
  - Up to 9.6 GT/sec QPI speed
- **Try to have Xeon E7 v2 or newer (Ivy Bridge-EX or newer) for large servers**
  - PCIe 3.0 support, good memory capacity and performance
  - Up to 3TB of RAM with 32GB DIMMs in a four-socket server
  - Up to 9.6 GT/sec QPI speed

SQL intersection

Preferred Broadwell-EP Processors – High Core Count

Model	Cores/L3 Cache	Base Speed	Turbo Speed	Price
E5-2699 v4	22/55 MB	2.2 GHz	3.6 GHz	$4,115.00
E5-2698 v4	20/50 MB	2.2 GHz	3.6 GHz	$3,226.00
E5-2697 v4	18/45 MB	2.3 GHz	3.6 GHz	$2,702.00
E5-2697A v4	16/40 MB	2.6 GHz	3.6 GHz	$2,891.00
E5-2690 v4	14/35 MB	2.6 GHz	3.6 GHz	$2,090.00

SQL intersection

Preferred Broadwell-EP Processors – Low Core Count

Model	Cores/L3 Cache	Base Speed	Turbo Speed	Price
E5-2687W v4	12/30 MB	3.0 GHz	3.5 GHz	$2,141.00
E5-2640 v4	10/25 MB	2.4 GHz	3.4 GHz	$939.00
E5-2667 v4	8/25 MB	3.2 GHz	3.6 GHz	$2,057.00
E5-2643 v4	6/20 MB	3.4 GHz	3.7 GHz	$1,552.00
E5-2637 v4	4/15 MB	3.5 GHz	3.7 GHz	$996.00

SQL intersection

Scaling Up SQL Server 2014

- **Scaling up is easy compared to scaling out**
- **Scaling up is expensive from a capital cost perspective**
  - Hardware costs increase exponentially
  - SQL Server 2014 core-license costs can add up very quickly
- **Scaling up is easy from an engineering perspective**
  - Very little development or testing effort is required
- **Higher hardware license limits with Windows 2012 R2/SQL Server 2014**
  - 640 logical processors and 4TB of RAM
- **You can currently get 6TB of RAM in a four-socket machine**
- **You can currently get 144 logical processors in a four-socket machine**
  - Intel Xeon E7-4800 v3 series (Haswell-EX)

Scaling Up Limitations

- **Scaling up is limited by hardware and license limits**
- **Scaling up is not as effective as you may assume**
  - Even with NUMA hardware, you don't get 1:1 scaling
  - A four-socket server does not have twice the capacity of a two-socket server
  - It also has lower single-threaded CPU performance
- **Two-socket servers have significantly better single-threaded performance**
  - Newer-generation Intel processors show up in two-socket space first
- **Commodity server hardware is limited to four or eight sockets**
  - Some vendors offer sixteen-socket x64 servers
- **Microsoft currently limits OS RAM to 4TB**
  - This license limit will go up to 12TB with Windows Server 2016
  - So far, Microsoft has not announced any change to the core license limits

References

- **Two Pluralsight courses on Scaling SQL Server 2012/2014**
  - Scaling SQL Server 2012 – Part 1
    - http://app.pluralsight.com/courses/scaling-sqlserver2012-part1
  - SQL Server: Scaling SQL Server 2012 and 2014: Part 2
    - http://app.pluralsight.com/courses/scaling-sqlserver2012-2014-part2
- **Microsoft Visual Studio Dev Essentials**
  - Free access to SQL Server 2014 Developer Edition
  - Free six month Pluralsight subscription
    - http://bit.ly/1q6xbDL
- **Microsoft IT Pro Cloud Essentials**
  - Lots of free Azure usage credits, MCP exam voucher, three month Pluralsight subscription
    - http://bit.ly/2443SAd

SQL intersection

Summary

- **There are many easy steps you can take to improve scalability**
  - These include storage configuration, hardware configuration, instance-level settings, database properties, index tuning and maintenance, etc.
- **Take the time to go though each layer of the system to optimize it**
  - Use my DMV Diagnostic Queries to find configuration issues and performance bottlenecks
- **Make sure to have modern hardware if possible**
  - This gives you extra capacity and better single-threaded performance

SQL intersection

Questions?

Don’t forget to complete an online evaluation on EventBoard!

Scaling SQL Server

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

SQL *intersection*

Thank you!