Microsoft®

tech·ed

中国 | 2011

While we are waiting….

- **This session will be delivered in English only**
  - But I have colleagues from the local office who speak Chinese and can help with questions.
- **Target audience is intermediate level SQL Server professionals**
  - SQL Server experts should feel free to attend other sessions

Microsoft® tech·ed 中国 2011

Performance tuning for non-beginners but not yet experts

Joe Yong

Senior Program Manager

Microsoft Corp

Before we begin

- **Goals**
  - Understand key performance influencers
  - Review common performance monitoring tools and techniques
  - How to interpret the data and determine what best practices are relevant to your environment
  - Review actual customer performance tuning mistakes
- **Non-goals**
  - Make you a SQL Server performance monitoring/tuning expert
  - Deep dive on features or tools
    - But feel free to ask questions
  - Provide performance troubleshooting magic bullets
  - Flashy demos

Intermediate level session on performance?

- **Plenty of content for beginners**
- **Plenty of content for expert level**
- **Not everybody can jump from A to Z in one step**
- **A lot of the content in this space is too cryptic and confusing for the average Joe**
- **This Joe does not believe you need to be an expert to find performance problems**
  - Though you might need help fixing them
    - But at least you know what needs to be fixed
      - And you'll learn to spot real experts vs. those who read a lot of blogs and repeat what they read

Agenda

- Understanding performance
- Measuring and tracking
- Interpreting the data
- Practical solutions

Agenda

- **Understanding performance**
  - **What is performance**
  - Performance influencers
- **Measuring and tracking**
- **Interpreting the data**
- **Practical solutions**

What is performance

- **Fulfilling workload demands within a period deemed acceptable or agreed upon by the user[1]**
  - Guaranteed < 2s business transaction response time
  - 130,000 database transactions per second
  - Loading >1 billion rows an hour
  - 18G**B**/s storage subsystem
  - 75seconds RTO for SQL Server in a Failover Cluster
  - Backup 2TB per hour over the network
  - Restore 1.3TB database in under 30 minutes
  - DBCC DBREINDEX on a 3 billion row table (ok, probably not a good idea)

[1] DBAs are just another class of user

Performance and scalability

- **Scalability is the level of performance maintained as workload influencers increase**
  - ✓ ERP application
    - 5 users: 2 seconds query response time
    - 700 users: 2.3 seconds query response time
  - × Generate year end report for
    - 300GB database: 85seconds
    - 1.5TB database: 30 minutes

Agenda

- **Understanding performance**
  - What is performance
  - **Performance influencers**
- **Measuring and tracking**
- **Interpreting the data**
- **Practical solutions**

Performance influencers

- **User workload**
  - DBAs are a special class of users
- **System & infrastructure resources**
  - Server, storage, interconnects, networks
- **Optimization**
  - Workload and resources sized and configured appropriately for your performance requirements at your require level (scale)
- **Contention**
  - A performance blocker that highlights a system's ability to scale (or not)
- **Environment**
  - Mostly stuff beyond your control like data center, ISP, workload source (e.g. end-user PC)

Agenda

- Understanding performance
- **Measuring and tracking**
- Interpreting the data
- Practical solutions

Measuring and tracking

- **Measures will vary depending on user workload**
  - Accounting will have very different performance metrics from an online trading – financial analyzer metrics are nothing like either of those
- **Different perspective of the same workload will have different measures**
  - Retail POS system
    - Cashier cares about items being scanned and registered quickly
    - Store manager cares about sales totals at any time
    - Loss prevention specialist wants to know quickly if items registered in POS equals to decrease in automated inventory manager

Measuring and tracking

- **Workload: measures will vary depending on the workload type**
  - Accounting will have very different performance metrics from online trading – financial analyzer metrics are nothing like either of those
- **User: different perspective of the same application will have different measures**
  - Library information system
    - Image recording/scanning clerk cares about fast BLOB inserts
    - Researcher needs fast searches on large data sets across multiple formats
  - Retail POS system
    - Cashier cares about items being scanned and registered quickly
    - Store manager cares about sales totals at any time
    - Loss prevention specialist wants to know quickly if items registered in POS equals to decrease in automated inventory manager

What to measure and track

- **Different application admins/owners track different performance metrics**
  - Not always aligned with what users care about
- **DBAs have some basic metrics that are "almost" universally helpful**
  - Resource utilization
  - Workload response time
  - Contention and waits
- **Important: performance metrics taken at any point in time without a baseline reference is often not useful**

Baseline

- **Establishes "known behavior"**
  - Resource utilization levels for a given workload and scale
    - Order entry system with 500 concurrent users consumes x GB RAM, y% sustained average CPU utilization – certified by ISV to scale almost linearly up to 11,000 concurrent users
  - User workload query response times
    - Monday 8am-11am: 2-5 seconds
    - Friday 12pm-4pm: 5-12 seconds
    - Other times: 1-3 seconds
- **Documents acceptable performance – your performance SLA**

Without a baseline

- **Networks never have problems**
  - We use multiple redundant 10Gb/s switches and average reliability is around 99.98%, what network problem?
- **Storage is never the problem**
  - We have two 2Gb/s ports from your server to the SAN, you are only using 400MB/s, go back and tune your SQL queries
- **End users can “instinctively” tell if performance is bad**
  - Why is it very slow today? It is usually a lot faster. You must have done something wrong during the upgrade.
- **End users’ PCs/laptops are always working fine and nothing has changed (or ever changes)**

Without a baseline

- Databases are guilty until proven innocent
- Without documented performance metrics and baselines you just can't win
- Especially if you have just performed a major operation (e.g. just upgraded to SQL Server 2008 R2)
- Difficult to convince other parties to investigate their areas (see previous slide)

Case study 1 – blame the upgrade

- **Major medical health care provider**
- **3 months upgrade planning and testing**
- **Successful upgrade over weekend**
- **Normal operations Monday morning; 6 phone calls and multiple emails starting 130pm**
  - Complaints about very slow queries, timeouts, etc…
  - Blames DBAs for botching the upgrade
  - Managers and senior managers put on CC

Case study 1 – blame the upgrade

- **DBA team reviews database server load and resource utilization**
  - Server not breaking a sweat
  - Network operations center (NOC) says network is healthy
  - Storage monitor shows light loads
  - DBAs run tests from local clients and verified performance metrics very close to baseline
    - Blocker scripts, long running queries, etc… all show nothing suspicious
- **Eliminated database, server, storage – focus on network**
- **2.5 hours later, faulty switch at Mexico office discovered**
  - NOC still says network is ok (apparently local network equipment problems don't count)
  - Other local US user problems not related

Basic measures independent of applications

- Storage
- Memory
- CPU
- Network

Storage

- **This is the physical foundation of your database**
- **80% deployments (that I've seen or worked on) have sub-optimal or incorrectly configured storage**
  - Including those certified by a storage vendor
- **Always test (SQLIO tool) before deployment**
- **Many counters and waitstats data available to monitor**
- **Start with the basics**
  - IOPS
  - Throughput
  - Latency
  - Queuing

Storage

Metric	Data source	Considerations
IOPS	Performance Monitor • Average disk read/write per second	• Number of I/O requests per second • Type of disk and size can affect this greatly • E.g. 2.5” SAS can sustain ~180 IOPS, 3.5” 15k rpm SCSI sustains ~130 IOPS

Throughput	Performance Monitor
	• Disk read/write bytes per second
		• Bytes throughput per second
		• Beware of B vs b
		  • 1Gb/s ~ 125MB/s
		  • 3Gb/s ~ 300MB/s

Latency	Performance Monitor
	• Average disk sec per read/write
		• Latency of I/O operations
		• Typically[1]
		  • Log: 1-5ms (ideally <1ms)
		  • Data (OLTP): 5-20ms (ideally <10ms)
		  • Data (DSS): 20-30ms

Queuing	Performance Monitor • Logical/Physical disk: Average disk queue length	• Number of I/O requests that cannot be serviced immediately • Value is measured across all disks for direct attached storage

[1] SQL Server Pre-deployment IO Best Practices

Memory

- **Next in-line in the list of common performance bottlenecks**
- **Difficult to really determine how much is enough given SQL Server's memory management approach**
  - Measure in reverse; look for pressure
- **Further complicated with NUMA servers**
  - One or more nodes may have severe memory pressure while other nodes are idle
- **Often result in performance problem symptoms in other areas**
- **Start with the basics**
  - Available free memory
  - Memory pressure hints

Memory

Metric	Data source	Considerations
Available free memory	Performance Monitor • Memory:Available Mbytes • SQLServer:BufferManager\Free Pages	• Tracks available system memory for use • Total number of pages on the free list
Memory pressure hints	Performance Monitor • SQLServer:BufferManager\Free list stalls /sec • SQLServer:Memory Manager\Target Server Memory (KB) and SQLServer:Memory Manager\Total Server Memory (KB) DMV • Sys.dm_os_wait_stats	• Tracks potential memory pressure by measuring number of free page requests that are not granted immediately • Track what SQL Server thinks it could use vs what it current is able to use • Track memory grants pending

Be careful with these commonly "recommended" counters

- **Often recommended in public forums, blogs, events as counters to determine if memory is adequate**
  - Cache hit ratio >90%
    - Highly workload dependent – if the queries do not touch the same data, hit ratio will be low but this is not a clear indicator of where the performance problem might be
  - Page life expectancy >300
    - Can be workload dependent and behavior is similar to cache hit ratios
  - Memory grants pending <2
    - Theoretically tracks memory requests pending but this can be caused by contention (e.g. blocking), not just lack of memory
- **Correlate multiple data point; don't just measure one**

CPU

- **Sustained high activity may or may not indicate bottlenecks or problems**
- **Hardware faults can sometimes result in high CPU utilization and/or context switches**
- **Start with the basics**
  - Overall utilization
  - CPU pressure hints

CPU

Metric	Data source	Considerations
Overall utilization	Performance Monitor • Processor:%Processor Time	• Amount of time the CPUs are actually doing work across the entire system • Value is for all CPUs in the system
CPU pressure hints	Performance Monitor • System:Processor Queue Length	• Total number of threads awaiting CPU resource • Value is for all CPUs in the system

Network

- **Usually one of the easiest resources to manage for performance**
- **Start with the basics**
  - Overall utilization
  - Device errors
- **Be careful with some commonly recommended counters**
  - Output queue length
    - Dependent on operating system implementation (E.g. always zero in Vista)

Network

Metric	Data source	Considerations
Overall utilization	Performance Monitor • Network Interface: Bytes Total /sec	• Measures volume of data sent and received by each network adapter or cumulative for all NICs in the system
Device errors	Windows Event Log • System Logs: network device errors	• Non-critical hardware faults are logged here • Critical hardware faults can result in blue-screen

Agenda

- Understanding performance
- Measuring and tracking
- **Interpreting the data**
- Practical solutions

Interpreting the data: storage*

Data	Considerations

• Average disk read/write per second
• Disk read/write bytes per second
	• Ensure system requirements are less than storage capacity
	• If using SAN, ensure SAN cache can flush to disk faster then workload requests fills up SAN cache
	• Beware of B vs b for bytes/second
	  • 1Gb/s ~ 125MB/s
	  • 3Gb/s ~ 300MB/s

• Average disk sec per read/write	• Hardware and configuration can affect values
	• Log: 1-5ms (ideally <1ms)
	  • Minimal tolerance for exceeding 5ms
	• Data (OLTP): 5-20ms (ideally <10ms)
	  • Moderate tolerance levels dependent on application
	• Data (DSS): 20-30ms
	  • Higher values are not uncommon and may not be a performance issue but try to keep <50ms

• Logical/Physical disk: Average disk queue length	• Use logical value for SAN devices and corroborate with SAN data

***Beware of polling intervals**

Interpreting the data: storage

- **Problems with storage can result in many different symptoms across the system**
- **General statements like**

  "…storage performance problem…"

  "…query timed out…"

  is not helpful in most modern systems – get the specifics
  - HBA
  - Switch
  - SAN
    - Ports
    - Cache
    - Disks

Case study 2 – blame SQL Server scalability

- **Top 10 largest banks in the US**
- **Commercial customers system upgrade SQL Server & hardware**
- **Phase 1 (SQL Server) stable – tens of millions transacted daily**
- **Phase 2 (new hardware) nearing rollout; expecting 200% increase in customer base within 6-12 months then stable**
- **Second largest possible server in the market (at the time) – sized to handle ~10x growth**
- **Top of the line SAN – sized to handle at least 12x growth**
- **POC stress test consistently collapses at 3x ~ 4x load**
  - Blamed SQL Server scalability; suggestions to use ***O***ther database
  - All other parties claim no-fault

Case study 2 – blame SQL Server scalability

- **POC background**
  - Realistic workload capture/replay
  - Server burn-in tests onsite (memory, CPU, peripherals)
  - Theoretical maximum throughput & capacity for SAN calculated
    - Alerted sponsor and vendor that SAN is significantly under configured – dismissed by vendor citing POC team lacks knowledge on brand new SAN
  - Actual SAN capacity tests onsite (IOPS, MB/s, latency)
    - Basic tests support POC team's calculations
- **Full POC tests fails with multiple indicators pointing to SAN overload**
  - 1.5 days before SAN team finally concede they had erred in their data collection and interpretation (neglected to mention original design error)
  - This was with detailed documentation and extensive data
- **Re-designed SAN with POC team guidance supported new stress tests up to 7x workload – load generators exhausted**

Case study 2 – blame SQL Server scalability

- **Chasing symptoms**
  - Query timeouts
  - CXPacket waits – someone read on a blog that this was bad
  - Context switching
  - Compilation & re-compilations – apparently this is bad too
  - Load generators' CPUs constantly ~90%
  - Application's history of performance issues – send developers for ~~flogging~~ training
- **Root cause**
  - Storage unable to keep up
    - Overlapping stripes – same HDDs striped twice to create LUN
    - Over estimated SAN cache capacities – not enough HDDs to flush cache quickly

Interpreting the data: memory

Data	Considerations

• Memory:Available Mbytes
• SQLServer:BufferManager\Free Pages
• SQLServer:BufferManager\Free list stalls /sec
• SQLServer:Memory Manager\Target Server Memory (KB) and SQLServer:Memory Manager\Total Server Memory (KB)
	• Appropriate values dependent on type of application and volatility of query activity
	• Reasonable starting points
	  • Available Mbytes: >50
	  • Free pages: >640
	  • Free list stalls: <2
	  • Target/total memory: dependent on total physical memory on server and number/type of workloads active

• SQLServer:Buffer Node\Free pages • SQLServer:Buffer Node\Foreign pages • SQLServer:Buffer Node\Remote node page lookups /sec	• Additional counters to monitor for NUMA based servers • Local node memory usage and cross-node usage dependent on application behavior • SQL Server tries to work efficiently within local node but application design and/or behavior can prevent this

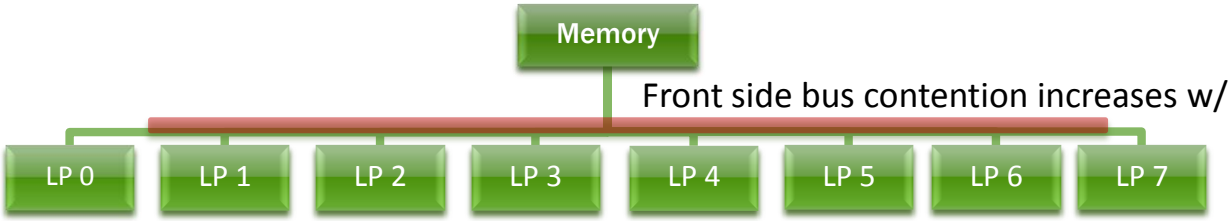
Case study 3 – blame SQL Server scalability

- **Major retail chain running sales monitoring and inventory tracking application**
- **Regular performance degradation but non-discernable pattern or root causes**
- **System review indicates storage IO bottlenecks**
  - CPU utilization low to moderate
  - Overall memory utilization low to moderate
- **Upgraded SAN; storage IO bottlenecks removed**
- **Performance issues persist**

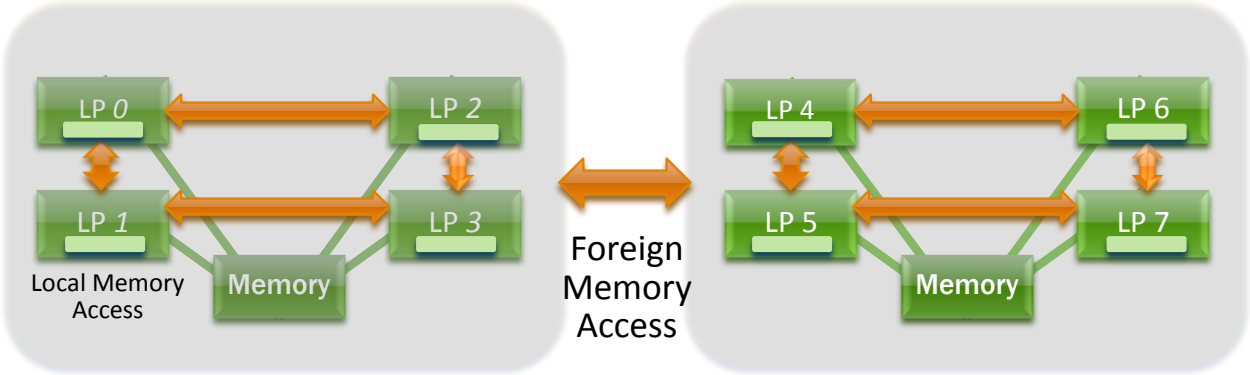
Case study 3 – blame SQL Server scalability

- **Engaged specialists for detailed system review**
  - Occasional storage bottlenecks still exist
  - 1-2 out of 4 NUMA nodes experiences memory pressure routinely
    - Induced by application design and data patterns
      - Manually managed workload/worker threads
      - Long running transactions
      - Vastly different data volume from hundreds of sources – from few MB to few GB

NUMA



Symmetric Multiprocessing Architecture (SMP)



Non-Uniform Memory Access (NUMA)

Foreign memory access >> local memory access

Interpreting the data: CPU

Data	Considerations
• Processor:%Processor Time	• Reasonable starting point • <80% but highly dependent on nature of application and spike values
• System:Processor Queue Length	• Reasonable starting point • <10 per logical CPU

Interpreting the data: network

Data	Considerations
• Network Interface: Bytes Total /sec	• Compare values to NIC capacity • Again, beware of B vs b
• System Logs: network device errors	• Device errors are bad news

What about other counters/waitstats?

- **There are many more you can monitoring and analyze**
  - Queries and query plans
  - Filestats
  - Waitstats
  - Index usage
  - Statistics
  - Extended events
- **These will help you identify the root cause of the problems**
  - Always use both resource utilization data with waitstats data in the context of the specific application

Agenda

- Understanding performance
- Measuring and tracking
- Interpreting the data
- **Practical solutions**

Non-expert solutions

- **Default reports**
  - Build your own once you're familiar with it
  - Don't run this too frequently
- **Profiler (or SQL Trace) + Performance Monitor**
  - Correlate data from these tools
- **Performance data warehouse or scheduled DMV snapshots**

Dynamic Management Views (DMVs)

- Really useful once you get past the basics; for example:
- Top 50 statements by IO

Top 50 queries by IO

```
SELECT TOP 50
        (qs.total_logical_reads + qs.total_logical_writes)
   /qs.execution_count
   as [Avg IO]
      , substring (qt.text,qs.statement_start_offset/2
      , (case when qs.statement_end_offset = -1
            then len(convert(nvarchar(max), qt.text)) * 2
       else qs.statement_end_offset end -
   qs.statement_start_offset)/2) as query_text
      , qt.dbid
      , qt.objectid
FROM sys.dm_exec_query_stats qs
CROSS APPLY sys.dm_exec_sql_text (qs.sql_handle) as qt
ORDER BY [Avg IO] DESC
```

IO waits per file

```
select database_id
       , file_id
       , io_stall,io_pending_ms_ticks
       , scheduler_address
from sys.dm_io_virtual_file_stats (NULL, NULL) as t1,
     sys.dm_io_pending_io_requests as t2
where t1.file_handle = t2.io_handle
```

TempDB usage tasks and SQL

```
SELECT t1.session_id,
(t1.internal_objects_alloc_page_count + task_alloc) as allocated,
(t1.internal_objects_dealloc_page_count + task_dealloc) as
 deallocated, t3.sql_handle
from sys.dm_db_session_space_usage as t1,
        sys.dm_exec_requests t3,
(select session_id,
   sum(internal_objects_alloc_page_count)
                       as task_alloc,
   sum (internal_objects_dealloc_page_count) as
        task_dealloc
      from sys.dm_db_task_space_usage group by session_id) as t2
where t1.session_id = t2.session_id and t1.session_id >50
and t1.database_id = 2    --- tempdb is database_id=2
and t1.session_id = t3.session_id
order by allocated DESC
```

Indexes that are not used

```
SELECT object_name(i.object_id), i.name
FROM sys.indexes i, sys.objects o
WHERE  i.index_id NOT IN
       (select s.index_id
        FROM sys.dm_db_index_usage_stats s
        WHERE s.object_id=i.object_id
        AND i.index_id=s.index_id
        AND database_id = <dbid> )
AND o.type = 'U'
AND o.object_id = i.object_id
ORDER BY object_name(i.object_id) ASC
```

Hardware setup

- Hardware vendors have built-in tests for most environments
- More thorough testing can be performed by vendor or certified partners

Case study 4 – blame unproven technologies

- **Global bank with extensive SQL Server and other database deployments**
- **Major SQL Server infrastructure overhaul and consolidation preparation**
  - Largest x64 server in the market – fully loaded
  - Latest enterprise SAN – almost fully loaded
  - Windows Server 2008 R2 and SQL Server 2008
- **HA and DR by design**
  - Large multi-node failover cluster
  - SAN replication for DR

Case study 4 – blame unproven technologies

- **System design reviewed and approved by specialists from MS**
  - Secondary review and approval by independent SQL Server experts
- **Servers & storage setup and configured by specialists from hardware principal and SI partner**
  - All the latest & greatest from server DIMM to physical HDD
- **Never ending problems with performance (right from the start)**
  - Unable to complete even one full cycle of performance testing
  - Intermittent cluster failures, occasional system dumps, several bluescreens
  - Some failures require full system rebuild – 2 days each time
- **Unable to even hand-over to QA team after 2 months**

Case study 4 – blame unproven technologies

- **Chasing symptoms**
  - Software setup taking a really long time – blamed storage performance
  - Bluescreens – blamed hardware, Windows
  - Intermittent cluster failures – blamed use of latest and greatest of everything
  - Never before seen such high level of instability/problems – blamed everybody (including performance test team for system overload)
- **Root cause**
  - Poorly seated components in server discovered with hardware diagnostic tools ~ 2 days work
    - Strip, rebuild, burn-in, full-test – handover to QA in < 2 weeks

Summary

- **Performance monitoring and troubleshooting can be done by the average Joe**
- **Baseline data is critical**
- **Start with the basics**
  - Capture everything including the kitchen sink approach is very intrusive and usually not helpful
    - If you must capture everything (sometimes it's right), analyze in layers
- **Do not draw conclusions based on singular data points or using raw data by itself**
- **Leverage what already exists before creating**
- **What and how are easy questions, why and why not are the tough questions**

Resources

- **Troubleshooting performance problems in**
  - SQL Server 2008
    http://msdn.microsoft.com/en-us/library/dd672789.aspx
  - SQL Server 2005
    http://technet.microsoft.com/en-us/library/cc966540.aspx
- **SQL Server Pre-deployment IO Best Practices**
  - http://sqlcat.com/whitepapers/archive/2008/10/03/running-sql-server-2008-in-a-hyper-v-environment-best-practices-and-performance-recommendations.aspx
- **SQL Server waits and queues**
  - http://sqlcat.com/whitepapers/archive/2007/11/19/sql-server-2005-waits-and-queues.aspx
- **SQLCAT best practices and high watermark customer experiences**
  - http://sqlcat.com/Default.aspx

Microsoft® tech·ed 中国 2011

Questions?

joeyong@microsoft.com

Microsoft® tech·ed 中国 2011

Microsoft®

您的潜力，我们的动力

© 2008 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.