The 5-minute SQL Server Health Check

- Christian Bolton – Technical Director, Coeo Ltd.
- Kevin Kline – Technical Strategy Manager, Quest Software
Agenda

- Introducing your speakers
- Hardware utilization and performance
- Where are the bottlenecks?
- Database and server configuration
- Database maintenance
- Health check summary
Kevin Kline

- SQL Server Expert for Quest Software
- Former developer, DBA, and enterprise architect since ‘86
- Former president of PASS (www.sqlpass.org)
- Microsoft MVP since ‘04
- Author of *SQL in a Nutshell* and 8 other books
- Twitter @kekline
- Blogs at http://sqlblog.com and http://KevinEKline.com
Christian Bolton

- Technical Director of Coeo Ltd
- Microsoft Certified Architect (MCA), Master (MCM), and MVP for SQL Server
- Lead Author of Professional SQL Server 2008 Internals and Troubleshooting
- Twitter: @christianbolton
- Blogs at: sqlblogcasts.com/blogs/christian
Hardware Utilization and Performance
Hardware utilization and performance

- Memory
- Storage
- CPU
Hardware utilization and performance

Memory

Check the following perfmon counters:

- Memory/Available Mbytes
  - > 100
- SQLServer: Memory Manager/Target Server Memory
- SQLServer: Memory Manager/Total Server Memory
  - Should be close to Target SQL Server Memory
- Buffer Manager: Page Life Expectancy
  - > 300 seconds
Hardware utilization and performance

Storage

- Check the following perfmon counters:
  - Logical Disk: Avg. Disk sec/Read
  - Logical Disk: Avg. Disk sec/Write
- We want these to be <5ms on a transaction log drive and <10ms on a data file drive
- 10-20ms is generally acceptable
- >20ms is considered to be unacceptable
Hardware utilization and performance

- **Check the file latency within SQL Server using**
  
  ```sql
  select 
    db_name(database_id), 
    io_stall_read_ms/num_of_reads AS 'Disk Read Transfer/ms', 
    io_stall_write_ms/num_of_writes AS 'Disk Write Transfer/ms'
  from 
    sys.dm_io_virtual_file_stats (2, 1)
  ```

- **Use this script to get the latency for each file:**

  ```sql
  select db_name(database_id), 
    io_stall_read_ms/num_of_reads AS 'Disk Read Transfer/ms', 
    io_stall_write_ms/num_of_writes AS 'Disk Write Transfer/ms'
  from 
    sys.dm_io_virtual_file_stats (2, 1)
  ```
Hardware utilization and performance

Storage

- RAID10 provides the best performance and availability
- Disk Sector Alignment can yield up to 40% improvement in some scenarios
  - Windows Server 2008 automatically implements sector alignment
Hardware utilization and performance

- Perfmon counters:
- %Privileged Time vs. %User Time
  - User Time should be > 70% of CPU usage
  - Low memory can increase Privilege Time
- Check Process:sqlservr %processor time
  - To see if SQL Server is the culprit

- However, I tend to use waits to diagnose SQL CPU issues....
Where are the Bottlenecks?
Where are the bottlenecks?

Using SQL Server waits

• Anytime a task in SQL Server is waiting for something
  – It is reported as a wait type
• SQL Server 2005 aggregates wait type information
• We can use it to see what the bottlenecks are on the server
SQL Server waits

Running Scheduler 1

Runnable Scheduler 1

Signal Wait

Suspended Scheduler 1

SQL Server “wait”
Common wait types

PAGEIOLATCH_*
  • Latch on a memory address while data is pulled from disk

OLEDB
  • Wait on the OLEDB provider
    – Full-Text Search
    – Lots of linked servers

CXPACKET
  • Query parallelism
Common wait types

SOS_SCHEDULER_YIELD
- Yielding processor time
- See signal wait

WRITELOG
- Writing transaction log to disk

LCK_M_*
- Waiting for a lock
Wait types to discount

WAITFOR
  • T-SQL WAITFOR command

SQLTRACE_BUFFER_FLUSH
  • Default trace

LAZYWRITER_SLEEPP
  • System process waiting to start
Other Interesting Wait Types

SLEEP_BPOOL_FLUSH
• Checkpoint I/O throttling

RESOURCE_SEMAPHORE_QUERY_COMPILE
• Throttling query compilations
  – Compilations, re-compilations, non-cacheable plans

RESOURCE_SEMAPHORE
• Waiting for a query memory grant
Dynamic management views

**sys.dm_os_exec_requests**

- Wait information
- Session level only
- Limited use for waits

<table>
<thead>
<tr>
<th>sessi...</th>
<th>start_time</th>
<th>status</th>
<th>command</th>
<th>sql_handle</th>
<th>stat...</th>
<th>stat...</th>
<th>plan_handle</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>17</td>
<td>sleeping</td>
<td>TASK M...</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>background</td>
<td>BRKR T...</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>16</td>
<td>19</td>
<td>background</td>
<td>BRKR T...</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
<td>sleeping</td>
<td>TASK M...</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td>sleeping</td>
<td>TASK M...</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>19</td>
<td>55</td>
<td>running</td>
<td>SELECT</td>
<td>0x02000...</td>
<td>0</td>
<td>-1</td>
<td>0x060001...</td>
</tr>
<tr>
<td>20</td>
<td>61</td>
<td>running</td>
<td>SELECT</td>
<td>0x03000...</td>
<td>2462</td>
<td>2546</td>
<td>0x050005...</td>
</tr>
</tbody>
</table>
Dynamic management views

sys.dm_os_waiting_tasks

- Wait information
- Task level
- Very accurate
- Transient data
sys.dm_os_waiting_tasks example

```sql
SELECT wt.waiting_task_address, wt.session_id, wt.wait_duration_ms, wt.wait_type, wt.blocking_session_id, wt.resource_description, [text], query_plan
FROM sys.dm_os_waiting_tasks wt
INNER JOIN sys.dm_exec_requests er ON wt.session_id = er.session_id
CROSS APPLY sys.dm_exec_sql_text(sql_handle)
CROSS APPLY sys.dm_exec_query_plan(plan_handle)
WHERE wt.session_id > 50
```

<table>
<thead>
<tr>
<th>address</th>
<th>session_id</th>
<th>wait_duration_ms</th>
<th>wait_type</th>
<th>blocking_session_id</th>
<th>resource_description</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>058274C8</td>
<td>78</td>
<td>91</td>
<td>WAITFOR</td>
<td>NULL</td>
<td>NULL</td>
<td>create procedure sys.sp_replmonitorrefreshjob { @ite...</td>
</tr>
<tr>
<td>0581D708</td>
<td>91</td>
<td>137</td>
<td>TRACEWRITE</td>
<td>NULL</td>
<td>NULL</td>
<td>create procedure sys.sp_trace_getdata {@traceid int, ...</td>
</tr>
<tr>
<td>1A573948</td>
<td>233</td>
<td>2</td>
<td>PAGEIOLATCH_SH</td>
<td>NULL</td>
<td>10:1:5074477</td>
<td>CREATE FUNCTION [dbo].[udf_CalculateOrderValueGranularity] (...</td>
</tr>
<tr>
<td>05809708</td>
<td>257</td>
<td>0</td>
<td>CPACKET</td>
<td>257</td>
<td>exchangeEvent id=Port8...</td>
<td>CREATE PROCEDURE [dbo].[usp_MainMatrixStyleColByRe...</td>
</tr>
<tr>
<td>07782088</td>
<td>257</td>
<td>1</td>
<td>CPACKET</td>
<td>257</td>
<td>exchangeEvent id=Port8...</td>
<td>CREATE PROCEDURE [dbo].[usp_MainMatrixStyleColByRe...</td>
</tr>
<tr>
<td>07782748</td>
<td>257</td>
<td>1</td>
<td>CPACKET</td>
<td>NULL</td>
<td>exchangeEvent id=Port8...</td>
<td>CREATE PROCEDURE [dbo].[usp_MainMatrixStyleColByRe...</td>
</tr>
<tr>
<td>E0D5C508</td>
<td>257</td>
<td>2</td>
<td>CPACKET</td>
<td>257</td>
<td>exchangeEvent id=Port8...</td>
<td>CREATE PROCEDURE [dbo].[usp_MainMatrixStyleColByRe...</td>
</tr>
<tr>
<td>0581DDC8</td>
<td>257</td>
<td>1</td>
<td>CPACKET</td>
<td>257</td>
<td>exchangeEvent id=Port8...</td>
<td>CREATE PROCEDURE [dbo].[usp_MainMatrixStyleColByRe...</td>
</tr>
</tbody>
</table>
Dynamic management views

**sys.dm_os_wait_stats**

- Wait information
- Cumulative by wait type
- Persistent data
- Transient data

<table>
<thead>
<tr>
<th>wait_type</th>
<th>waiting_tasks_count</th>
<th>wait_time_ms</th>
<th>max_wait_time_ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAZYWRITER_SLEEP</td>
<td>1467</td>
<td>1412932</td>
<td>1232</td>
</tr>
<tr>
<td>SQLTRACE_BUFFER_FLUSH</td>
<td>352</td>
<td>1406005</td>
<td>4056</td>
</tr>
<tr>
<td>CXPACKET</td>
<td>17884</td>
<td>388052</td>
<td>4492</td>
</tr>
<tr>
<td>WRITELOG</td>
<td>11577</td>
<td>180352</td>
<td>2106</td>
</tr>
<tr>
<td>PAGEIOLATCH_UP</td>
<td>5357</td>
<td>30981</td>
<td>764</td>
</tr>
<tr>
<td>SOS_SCHEDULER_YIELD</td>
<td>542099</td>
<td>11466</td>
<td>312</td>
</tr>
<tr>
<td>PAGEIOLATCH_SH</td>
<td>267</td>
<td>10374</td>
<td>93</td>
</tr>
</tbody>
</table>
SQL Server 2008

SQL Server 2005
- 230 wait types

SQL Server 2008
- 475 wait types
- PREEMPTIVE_*
- FT_*
Database and Server Configuration
Database and Server Configuration

- Can impact performance and introduce fragmentation
- Set fixed MB growth amounts
  - SQL Server can use Windows 2003 instant file initialization so large MB values are OK
    - If the service account isn’t a local admin you need to assign it the ‘Performance Volume Maintenance Tasks’ security policy
  - However, it doesn’t work for Transaction logs so be a bit more conservative
- Autogrowth is a last resort, size your database appropriately
Database and Server Configuration

**Tempdb**

- Size appropriately
  - Autogrown tempdb will reset after restart
- Move onto its own disk array
  - Set the size to fill 90% of the drive
    - Maybe overkill
    - Never need to autogrow
    - No performance disadvantage
  - Microsoft IT
    - 200GB drive
    - 180GB TempDB
Max Server Memory

• By default SQL engine will *not* acquire all requested memory upon startup
• Configure Max Server Memory
  – Leave some headroom for the OS and for memory requests outside the buffer pool

Option 1

• Run without max server mem
• Monitor at peak load
  – Memory Manager: Total Server memory (KB)
• Configure max server mem to peak figures
MAX Server Memory

Option 2

- Calculate worse case
- 2GB for the OS
- xGB for worker threads based on table below (2MB each on x64)
- 1GB for Multi-page allocations, linked servers etc
- 1-3GB for other applications

<table>
<thead>
<tr>
<th>Number of CPU’s</th>
<th>32-bit computer</th>
<th>64-bit computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;= 4 processors</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>8 processors</td>
<td>288</td>
<td>576</td>
</tr>
<tr>
<td>16 processors</td>
<td>352</td>
<td>704</td>
</tr>
<tr>
<td>32 processors</td>
<td>480</td>
<td>960</td>
</tr>
</tbody>
</table>
Max Server Memory

Scenario

- 8 cores, 32GB RAM
- 2GB for the OS
- 1GB for worker threads
- 1GB for Multi-page allocations, linked servers etc
- 1GB for backup program
- = 5GB

- Max Server Memory to 27GB
Database Maintenance
Database Maintenance

What should I be running?

- CHECKDB
- Index maintenance
  - Rebuild or Reorganize
- Statistics Maintenance
  - Auto-update statistics
  - Index rebuilds
Where to go next…
Health Check Summary
SUMMARY

- Check Memory, Disk, and then CPU
- Check waits to find bottlenecks
- Check database settings
- Check Max Server Memory setting
- Check database maintenance in place
- Deploy tools like PerfStats or the Quest tool set for further analysis
<table>
<thead>
<tr>
<th>Solution Area</th>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup and Recovery</td>
<td>LiteSpeed® for SQL Server</td>
<td>Fast, flexible backup and recovery with industry-leading compression technology</td>
</tr>
<tr>
<td>Performance Management</td>
<td>Foglight® Performance Analysis for SQL Server</td>
<td>Discover and resolve performance issues in production before they impact end users and service levels</td>
</tr>
<tr>
<td>Capacity Management</td>
<td>Quest® Capacity Manager</td>
<td>Gain control of disk space, growth rates and index maintenance</td>
</tr>
<tr>
<td>Development</td>
<td>Toad® for SQL Server</td>
<td>Plan and develop applications that deliver both functionality and optimal performance</td>
</tr>
<tr>
<td>Administration</td>
<td>Toad® for SQL Server</td>
<td>Comprehensive schema, object, security and change management</td>
</tr>
<tr>
<td>Community, Knowledge, Training</td>
<td>SQL ServerPedia</td>
<td>Be part of the community and learn about SQL Server with Quest’s free online resources</td>
</tr>
</tbody>
</table>
Online Resources

**SQLServerPedia.com** has over 25 top industry bloggers, wiki articles, book reviews, and free SQL Server training videos all accessible without registration.

**SQLServer.Quest.com** is our dedicated SQL Server community site with forums, downloadable add-ons and access to product managers & developers.

**Quest.com/backstage** is our “Backstage” area with all of our brochures, posters, and t-shirts, plus upcoming events like our Pain of the Week webcasts.