



# European PASS Conference 2009

Microsoft SQL Server Users Conference & Expo

April 22 - 24, 2009 • Neuss, Germany • Swissôtel



## Users working together

Connect. Share. Learn.



# European PASS Conference 2009

Microsoft SQL Server Users Conference & Expo

April 22 – 24, 2009 • Neuss, Germany • Swissôtel

## **Sizing SQL Server for (unknown) Workload**

=tg= Thomas H. Groher, Senior Database Engineer,  
bwin Interactive Entertainment AG



# SELECT \* FROM =tg=

## SQL Server DBA since 1994

- First SQL Server Version ever used 4.21
- First Log Shipping with failover on SQL Server 6.0
- First SQL Server Cluster on SQL Server 6.5 (NT 4.0 + Wolfpack)
- First time > billion rows in 1 table on SQL Server 7.0
- First time 100% availability for more than 2 years in a row: SQL Server 2000 (936 days to be exact)
- First time OLTP long distance database mirroring SQL Server 2005
- First time to return more than a peta byte on query results to clients without failure SQL Server 2008  
(so far all SQL Server 2008 are at 100% availability (all went online RTM + 24h)
- Hundreds of possibilities for SQL 11 ... can't wait to raise the bar again

## Focus on SQL Server Infrastructure Architecture and Implementation

### Close Relationship with Microsoft

- SQLCAT (SQL Server Customer Advisory Team)
- SCAN (SQL Server Customer Advisory Network)
- TAP (Technology Adoption Program)



Close relationship with Hardware Vendors (Focus IA64)

Active **PASS** member and **PASS** Summit Speaker

Newest project: [www.sqlserver-hwguide.com](http://www.sqlserver-hwguide.com)

**WARNING:** I recently read a book about power point! (sorry ☺)



**European PASS Conference 2009**

# Agenda

- Workloads (definition)
- The Theory
- The Practical Part
  - How to capture
  - How to make sense of the captured data
  - How to scale
- Removing bottlenecks till the requirements are met



# Workload

What is a Workload?

Workload is the sum of all activity on a server caused by clients/users, administrators and the server itself.

Various types

- OLTP (Online Transaction Processing)
- DSS/DWH/BI (Decision Support System, Data Ware House)
- Read Cache
- Log Server
- Session State Server
- **and the nightmare: mixed or all of the above**



# OLTP Workload

- Many concurrent users
- Small queries 10-40% write, 60 to 90% read
- Constant read/write mix
- Lots of small short transactions on the same tables/rows



# DSS Workload

- (relatively short) ETL (Extract, Transform, Load) process usually using bulk inserts and updates (90 to 100% write)
- A few users with
- A “few” complex queries, covering large amount of data (99 to 100% read)
- Almost no concurrency



# Other Workloads

- Read Cache
  - OLTP Workload with < 10% write >90% read
  - Low concurrency (often only one source changing the data)
- Log Server
  - Write workload >95% mostly inserts
  - Store and forget
- Session State Server
  - OLTP Workload with 50% read and 50% write operations mostly on LOB's



# Maintenance & Disaster Recovery

- Daily Backups
- Log Backups
- Index Defrag
- Index Rebuild
- Archiving
- DWH Loads
- Rollouts
- ...

What does your SLA say about all the above

Don't be surprised if the maintenance workload is higher than the operational workload.



# The Theory

Two approaches

- a) Academically
- b) Practically

We go with a)

Just Kidding

Of course we go with b)



# The Theory – Mathematical Model

Remember Junior High School:

2 Apples cost	\$ 6
156 Apples cost	\$ ???
-----	
1 Apple cost	\$ 3
156 Apples cost	\$ 468

For the prices of apples this is a very exact science, when it comes to workloads its almost close enough



# The Theory – Mathematical Model

1 User produces a workload  $W$

$n$  Users produce a workload  $n \times W$

In a pure client/server environment this is close enough but in a three or more tier world with caching this might not be accurate enough. You might have to capture the workload of one user first ( $W_1$ ), and then the workload of two users ( $W_2$ ) working at the same time

1 User produces workload  $W_1$

2 Users produce workload  $W_2$

$n$  Users produce a workload of  $= W_1 + (W_2 - W_1) \times (n - 1)$



**Demo**

---

# Capture Workload Demo



**European PASS Conference 2009**

Session Code - Session Title

# Capture and Analyze the Workload

- Setup a trace
- Run the workload
- Read the trace results into tables
- Query how often each type of statement is executed and weight the count by the number of logical I/O's created
- Use the numbers gathered to configure SQL Stress



# SQL stress

Is a tool to stress SQL Server in a very unique way.

It simulates as many concurrent users you like (and your client hardware can handle).

Free download at: <http://www.sqlstress.com>



# Demo

Bottleneck Elimination Demo

---

## **SQL Stress Demo**



**European PASS Conference 2009**

Session Code - Session Title

# CPU & Memory performance

CPU is rarely the real bottleneck in today's servers but its possible.

In most cases it's the performance of the system memory that's the actual bottleneck.

Main cause for 100% CPU utilization are in memory table scans where the CPU spent most of the time waiting for the data from the memory.

Side Note: In this cases optimizing the queries actually helps more than adding additional CPU's



# CPU & Memory Performance

Selecting a CPU: Xeon x86/x64 or Itanium IA64  
1 or 2 sockets: Xeon Nehalem  
4 sockets Xeon Quad or Six Core  
>4 sockets \*) Itanium IA64

\*) or extreme I/O requirements

- MHz or MB, Cache vs Clock speed



# Too few Memory

SQL Server loves memory (especial the 64 bit versions)

If SQL Server has not enough memory it has to go to disk very often and that's slowing down memory a lot

The absolutely minimum of memory a system should have is the size of all index root and intermediate pages plus about 8 MB per user. If its less almost every query will end up touching the disks



# Too few Memory

To few memory is shown by a low cache hit ratio and high disk read activity on the data disks

- 32 / 64 bit Question
- NUMA (Local/Remote Memory)
  - CPU Affinity
  - IO Affinity
  - MaxDegreeOfParalellism 1 ... OLTP n .... DSS



# Too much memory

Is there something like “too much memory”?

Unfortunately Yes

Stopping SQL Service can take very, very long

Especially a problem on planned cluster failovers

Xeon Nehalem Memory (2 Socket):

- 6 Modules á 1266MHz or
- 12 Modules á 1066MHz or
- 18 Modules á 800MHz



# Disks

- Data disks
  - Log disk
- } Always separate
- IO Subsystems
    - Internal: Parallel SCSI, SAS, FC SCSI, SATA ,
    - External: iSCSI, SAN
  - Alignment
    - Windows 2003 server and before always manually align disks
    - Windows 2008 aligns automatically
  - Cache Settings:
    - SQL Server never benefits from read cache
    - 100% write cache for all data and log disks



# Data disks OLTP Workload

8k random reads define actual performance

8k random writes (done async by lazywriter → less critical)

IO Subsystems: Internal/External: almost no difference

Spreading the data across multiple spindles increases the performance almost linear if the data is distributed evenly

RAID Level 1/0, 1, 5, 6 and 0



# Data disks DSS Workload

8k random and 64/256/1024k sequential reads

(read ahead)

SQL Server can handle 250 to 350 MB/sec per CPU core for DSS type queries (table and range scans) (current record 20GB per second table scan on 64 core system)

→ Typical server disk are the bottleneck

→ e.g.: 2 socket = 8 core

→ ~2 GB/sec disk performance required

single spindle can read 20 to 50 MB/s

→ 20 to 50 spindles at least



# Data disks DSS Workload

IO Subsystems: Internal/External: almost no difference

Spreading the data across multiple spindles increases the performance almost linear if the data is distributed evenly

RAID Level 1/0, 1, 5, 6 and 0



# Log disk

Sector aligned up to 60k sequential writes and

Sector aligned up to 120k sequential reads

Multiple files don't increase performance therefore the most important performance indicator is latency

IO Subsystems: Internal/External: huge difference

RAID Level :           1   (recommend)  
                          careful with 1/0 and 0  
                          avoid 5 and 6



# Log disk

## Storage Systems for Log files

- Internal direct connected SSD solutions 50  $\mu$ S
- Internal disk systems with battery backed cache < 0.25 ms
- External disk systems with battery backed cache < 2 ms
- Internal disk systems without cache < 3 ms
- Everything else > 3 ms

High volume: one log file per disk better than RAID 1/0

Spreading the data across multiple spindles increases the performance **only** if done correct

SAN based storage replication is the killer for log volumes



**Demo**

---

# **Storage Performance Counter**



**European PASS Conference 2009**

Session Code - Session Title

# Storage Selection

- Separate LOG and DATA
- Do three calculations
  - Space estimate
  - Performance estimate for workload
  - Maintenance workload estimate

→ Each will give you a number of disks/spindles  
Unfortunately the largest result is the one you actually need
- Start making trade offs based on your budget.
- Don't let marketing confuse you



# Network

Network Interface Cards (NIC) rarely the bottleneck today

LAN should not be a bottleneck today

WAN can be a bottleneck, careful with mirroring

NIC drivers, complete different story:

Tip: use the drivers the manufacturers use with benchmarks, low overhead, well tested

Use Multiple NIC's in a server to offload traffic

- Backup
- Log backup
- Mirroring

NUMA: Multiple NIC's one per node to partition workload and keep queries local



# Small Systems

1 to 2 CPU sockets

1 to 8 GB memory

1 to 4 disk spindles

- Try to separate LOG and DATA files on two different spindles, adding spindles will always help.
  - Dependent on Workload, try to do the best
- Increasing memory will always help
  - Try to keep the working set in memory
- Configure disk caches correctly



# Medium Systems

2 to 4 CPU sockets

4 to 32 GB memory

8 to 16 Disk spindles

- Separating DATA and LOG is a must!
  - With multiple databases on the system think about giving the ones with more workload private LOG drive
- Start using multiple disk controllers (gives extra cache), make sure if the system is NUMA based (AMD, newest Intel Xeon) to spread controllers and NIC and memory equally across the NUMA nodes
- SAN features and easier management and the ability to share many spindles can improve things



# Large Systems

4 to 8 CPU sockets

16 to 64 GB memory

16 to 50 Disk spindles

- NUMA is a given fact on this systems, make it your friend by supporting it.
- Definitely use multiple disk controllers
- Choose memory careful sometimes adding memory can slow down the system.
- Think about using different types storage systems for your log and data
- Start question if of shared storage is not slowing you down



# Very Large Systems

16 to 64 CPU sockets

128 to 2048 GB memory

100 to 5000 Disk spindles

- All bets are off, multiple NUMA nodes are a given, sometimes with even 2 different remote memory performances,
- you need to know any detail of your workload to use servers like this efficiently
- Different storage systems for log and data and backup are a given.
- See what bwin does in my other session

**Failure is not an option 24x7 VLDB administration**

Tomorrow after the lunch break



# Summary / Takeaways

Don't forget to scale all components, its not CPU's alone that define a servers performance.

Don't forget to consider maintenance workload and disaster recovery workload (both should be specified in a SLA) when dimensioning a server.





**Questions?**



[tg@grohser.com](mailto:tg@grohser.com)

# ***Next sessions:***

***13:30***



## **Working with Storage**

*Christian Bolton (Coeo Ltd.)*

*Track 1*

## **Serious SQL tuning & Optimization fun with 96 cores**

*Henk van der Valk (Unisys), Thomas Kejser (SQLCAT)*

*Track 2*

## **Passing Information to SQL Server - Parameters**

*Andras Belokosztolszki (Red Gate Software)*

*Track 3*

## **DBA Change Data Capture in Detail**

*Steffen Krause (Microsoft Deutschland GmbH)*



**Thank you!**

[tg@grohser.com](mailto:tg@grohser.com)