**Dokumentation: Analysebericht**

**Autor: …**

**Version: 1.00**

**Date of documentation: 07.11.2016**

**Date of approval: 08.11.2016**

**Approved by: <Name> – <Unternehmen>**

**Copyright: <Ihr Unternehmen>**

Unless otherwise specified, the documents are for your personal and non-commercial use. You may not modify copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell any information, software, products or services.

# Einleitung

Schreiben Sie hier etwas über Ihr Unternehmen.

# Über den Autor des Berichts

Der Analysebericht wurde <Name>, <Rolle> der <Unternehmen> verfasst. <Name> arbeitet seit 1998 mit Microsoft SQL Server und verfügt über ein mehr als 15‑jähriges Know-How in der Programmierung und Verwaltung / Administration von Microsoft SQL Server.

Herr Ricken hat die folgenden Zertifizierungen für Microsoft SQL Server erworben:

* Microsoft Certified Solutions Expert: Data Platform

Inhaltsverzeichnis

[Einleitung 2](#_Toc466379500)

[Über den Autor des Berichts 2](#_Toc466379501)

[Problembeschreibung 5](#_Toc466379502)

[Systemanalyse (Kurzfassung) 5](#_Toc466379503)

[Technische Konfiguration 5](#_Toc466379504)

[Hardware 5](#_Toc466379505)

[Virtualisierung 5](#_Toc466379506)

[Microsoft Windows Betriebssystem 6](#_Toc466379507)

[Microsoft SQL Server 6](#_Toc466379508)

[Ergebnis der Untersuchung 6](#_Toc466379509)

[Betriebssystem 6](#_Toc466379510)

[Laufwerke 6](#_Toc466379511)

[Arbeitsspeicher 7](#_Toc466379512)

[NUMA-Aufteilung 7](#_Toc466379513)

[Microsoft SQL Server 8](#_Toc466379514)

[Wartungsaufträge 9](#_Toc466379515)

[Cost Threshold For Parallelism 9](#_Toc466379516)

[Max Degree of Parallelism 9](#_Toc466379517)

[Wait Stats 10](#_Toc466379518)

[Datenbanken 12](#_Toc466379519)

[TEMPDB 12](#_Toc466379520)

[Benutzerdatenbank(en) 13](#_Toc466379521)

[Indexpflege 13](#_Toc466379522)

[Statistiken 14](#_Toc466379523)

[Handlungsempfehlungen 15](#_Toc466379524)

[SQL Server OS (Operation System) 15](#_Toc466379525)

[Laufwerke 15](#_Toc466379526)

[Arbeitsspeicher 16](#_Toc466379527)

[Aktivierung von standardisierten Traceflags 16](#_Toc466379528)

[Security 16](#_Toc466379529)

[Trennung von Test- und Produktionssystem 17](#_Toc466379530)

[Wiederherstellungsmodell von Benutzerdatenbank SMP\_NG 17](#_Toc466379531)

[Sharepoint 17](#_Toc466379532)

[Parallelisierung 18](#_Toc466379533)

[Maintenance 18](#_Toc466379534)

[Indexierung 18](#_Toc466379535)

[Zusammenfassung 18](#_Toc466379536)

<Name> wurde von der **<Untersuchtes Unternehmen>** beauftragt, die Konfiguration eines Microsoft SQL Servers im Produktivumfeld zu begutachten. Die Untersuchung des Systems wurde am <Datum> in einer Team Viewer‑Sitzung und Telefonverbindung/Vor Ort durchgeführt. Die Untersuchung dauerte von 08:00 – 11:00. Die Untersuchung des Systems wurde im Beisein von folgenden Personen durchgeführt:

|  |  |  |
| --- | --- | --- |
| Person | Unternehmen | Rolle / Abteilung |
| <Name> | <Unternehmen> | IT-System- und Netzwerk-Administrator Organisation und Informationsverarbeitung |

Sofern die Lösungsansätze unmittelbar während der Begutachtung konfiguriert wurden, wird explizit darauf hingewiesen.

# Problembeschreibung

Der Auftraggeber meldet bei bestimmten Verfahren innerhalb der Software Verzögerungen in der Bearbeitung. Konkret benannte Probleme mit einzelnen Modulen der Software wurden während der vorliegenden Untersuchung nicht analysiert. Da das System primär mit den gespeicherten Daten in der Datenbank kommuniziert, soll durch eine Untersuchung des Microsoft SQL Servers sichergestellt werden, dass keine Konfigurationsprobleme zu den benannten Einschränkungen führen.

# Systemanalyse (Kurzfassung)

Während der Untersuchung konnten folgende Probleme gefunden werden

* Fehlende Servicepacks
* Überdimensioniertes System 128 GB RAM / 8 Cores
* Nicht optimale Aufteilung von System-, Benutzerdatenbanken Laufwerk C: und D:
* Nicht optimale Aufteilung der Protokolldateien Laufwerk D:
* Nicht optimale Aufteilung der Systemdatenbank TEMPDB Laufwerk C:
* Nicht optimal formatierte Blockgröße der Laufwerke für Datenbankdateien: 4.096 Bytes
* Unterschiedliche Größen der Datenbankdateien von TEMPDB
* Zu niedriger „Schwellwert für Parallelisierung“: 5
* Zu viele Cores für Parallelisierung: 0
* Nicht aktivierte Maintenance-Aufträge für Index- und Statistikaktualisierungen
* Datenbank-Kompatibilitätsmodus der betroffenen DB auf SQL Server 2008
* Keine Trennung von Test- und Produktionssystemen
* Betrieb einer einschränkenden Applikation SharePoint

# Technische Konfiguration

Die technische Konfiguration dient zum Betrieb der Plattform-Technologie von Microsoft SQL Server. Sofern die Untersuchung der Hardware Bestandteil der Analyse war, wird zu diesem Teil ein separates Dokument ausgehändigt.

## Hardware

Die physikalische Hardware war nicht Bestandteil der durchgeführten Analyse

## Virtualisierung

Der zu untersuchende Microsoft SQL Server wird in einer virtuellen Umgebung des Herstellers <Microsoft/VMware> betrieben. Die Einstellungen/Konfiguration des Hyper-V Hosts wurde im Rahmen einer Voruntersuchung vom <Datum> durchgeführt. Die Untersuchung hat eine ideal konfigurierte Virtualisierung ergeben.

## Microsoft Windows Betriebssystem

Der zu untersuchende Microsoft SQL Server wird mit einem virtualisierten Windows Server 2012 betrieben. Insgesamt verwaltet das Betriebssystem 8 vCores sowie 128 GB RAM. Der Plattenspeicher wird über ein SAN-System[[1]](#footnote-1) zur Verfügung gestellt.

## Microsoft SQL Server

Bei der Begutachtung des Systems wurde ein Microsoft SQL Server 2014 Enterprise Edition (64bit) vorgefunden. Der Microsoft SQL Server wird als Standardinstanz[[2]](#footnote-2) betrieben und weist den Patch-Stand 12.0.2269.0 (**RTM**) auf. Für die begutachtete Instanz werden 8 vCores sowie 128 GB RAM zur Verfügung gestellt.

# Ergebnis der Untersuchung

Die Untersuchung wurde mittels einer „Top-Down-Methode“ durchgeführt. Hierbei wurde zunächst die – virtualisierte – Hardware (CPU, RAM, STORAGE) untersucht. Anschließend wurden die Konfigurationseinstellungen des Microsoft SQL Servers untersucht um abschließend die Datenbanken selbst zu untersuchen.

Hinweis

Aus datenschutzrechtlichen Gründen werden im Analysebericht nur Systemdatenbanken und die Datenbanken benannt, die im Rahmen der Beauftragung für den Applikationsbetrieb relevant sind. Welche Datenbanken davon betroffen waren, wurde vorgegeben.

Hinweis

Sofern sicherheitsrelevante Probleme analysiert wurden, werden diese nicht in diesem Dokument benannt, wenn sie nicht ursächlich für eine Performanceperspektive sind.

Basierend auf den nachfolgenden Ergebnissen der Untersuchung werden entsprechende Handlungsempfehlungen mit Hinweis auf die verantwortliche Partei benannt.

## Betriebssystem

Das System wir in einer virtuellen Umgebung betrieben. Wie die virtuellen Maschinen auf Seiten von Hyper-V konfiguriert wurden, wurde nicht untersucht. Eine Voruntersuchung hat jedoch ausreichende und optimal konfigurierte Ressourcen ermittelt. Das System verfügt über 8 vCores und 128 GB RAM. Die 8 vCores wurden mit 2 NUMA-Knoten konfiguriert.

## Laufwerke

Für das Betriebssystem sowie für die Datenbanken werden insgesamt 2 Laufwerke zur Verfügung gestellt.

|  |  |
| --- | --- |
| Laufwerk | Verwendung |
| C: | Betriebssystem und Systemdatenbanken |
| D: | Benutzerdatenbanken |

Die Schreib- und Leselatenzen (in Millisekunden) der einzelnen Laufwerke stellen sich wie folgt dar:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Drive | Read Latency | Write Latency | Overall Latency | Avg Bytes/Read | Avg Bytes/Write | Avg Bytes/Transfer |
| C: | 1 | 0 | 0 | 62.274 | 46.172 | 46.610 |
| D: | 2 | 0 | 1 | 579.829 | 30.190 | 129.247 |

Das zur Verfügung gestellte Storage weist eine **SEHR GUTE** Performance auf. Eine gute SAN-Konfiguration sollte Overall-Latenzen zwischen 1 und 10 aufweisen. Obwohl die Laufwerke sehr gute Latenzen aufweisen, wurde festgestellt, dass sie mit einer falschen Blockgröße formatiert sind (4.096 Bytes). Die „kritische“ Datenbank teilt sich mit allen anderen Benutzerdatenbanken das Laufwerk D.

Die Systemdatenbank TEMPDB verwendet das Systemlaufwerk C. TEMPDB[[3]](#footnote-3) ist eine Systemdatenbank mit einem nicht unerheblichen Einfluss auf die Performance aller Datenbanken, die in einer Instanz von Microsoft SQL Server betrieben werden. TEMPDB unterliegt einer starken Beanspruchung, da sie als zentrale Datenbank von Microsoft SQL Server für alle Aktionen verwendet wird, in der temporäre Objekte benötigt werden:

* Read Committed Snapshot Isolation[[4]](#footnote-4)
* HASH Join Operationen[[5]](#footnote-5)
* SORT Operationen[[6]](#footnote-6)
* Temporäre Tabellen
* Tabellenvariablen
* …

Durch die Zuordnung von TEMPDB auf Laufwerk C: ist im schlimmsten Fall mit einem Totalausfall des Systems zu rechnen, wenn die Ressourcen von TEMPDB den vollständigen Plattenplatz von Laufwerk C: belegen!

## Arbeitsspeicher

Der Arbeitsspeicher ist mit 128 GB Gesamtkapazität (virtuell) konfiguriert worden. Von diesen 128 GB RAM wurden exklusiv bis zu 102 GB für Microsoft SQL Server reserviert.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CPU | RAM (physically) | RAM (granted) | RAM (committed) | max\_workers\_count |
| 8 | 128.00 | 102.40 | 102.40 | 576 |

Weitere Untersuchungen (siehe „Wait Stats“ auf Seite 7) haben ergeben, dass der zur Verfügung gestellte Arbeitsspeicher nicht ausreichend für das System dimensioniert ist wie auch die Auswertung der PLE (Page Life Expectancy) zeigt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| object\_name | NUMA | PLE | Measure\_Date\_Time | Server\_Start\_Date\_Time |
| SQLServer:Buffer Node | 001 | 1.286.676 | 2016-11-07 08:29:49.390 | 2016-10-23 12:05:19.820 |
| SQLServer:Buffer Node | 000 | 1.286.676 | 2016-11-07 08:29:49.390 | 2016-10-23 12:05:19.820 |

Die obige Auswertung zeigt die durchschnittliche Aufenthaltsdauer einer Datenseite im Buffer Pool pro NUMA‑Node. Alle Knoten sollten einen Wert von >= 1.000 Sekunden aufweisen. Da diese Werte nur einen Snapshot repräsentieren, kann mit Hilfe von PerfMon[[7]](#footnote-7) eine Aufzeichnung über einen ganzen Tag durchgeführt werden, um den täglichen Verlauf zu analysieren.

## NUMA-Aufteilung

Die 8 vCores werden auf 2 NUMA (Non Uniform Memory Access) verteilt. Somit stehen pro 4 vCores 128/2 GB RAM zur Verfügung. Unter NUMA versteht man eine Computer-Speicher-Architektur für Multiprozessorsysteme, bei denen jeder Prozessor einen eigenen, **lokalen** Speicher hat.

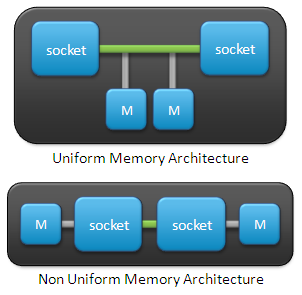


Abbildung 1: NUMA

Jeder Prozessor kann anderen Prozessoren über einen gemeinsamen Adressraum direkten Zugriff darauf gewähren (**Distributed Shared Memory**). Die Speicherzugriffszeiten in einem solchen Verbund hängen daher davon ab, ob sich eine Speicheradresse im lokalen oder im fremden Speicher befindet.

## Microsoft SQL Server

Microsoft SQL Server 2014 wird auf dem oben beschriebenen Windows-System mit dem Patchlevel 12.0.2269.0 (RTM) betrieben. Die für die Performance wesentlichen Konfigurationseinstellungen wurden wie folgt vorgefunden:

| name | Description | value\_in\_use |
| --- | --- | --- |
| recovery interval (min) | Maximum recovery interval in minutes | 0 |
| locks | Number of locks for all users | 0 |
| fill factor (%) | Default fill factor percentage | 0 |
| cross db ownership chaining | Allow cross db ownership chaining | 0 |
| max worker threads | Maximum worker threads | 0 |
| cost threshold for parallelism | cost threshold for parallelism | 5 |
| max degree of parallelism | maximum degree of parallelism | 1 |
| min server memory (MB) | Minimum size of server memory (MB) | 12.288 |
| max server memory (MB) | Maximum size of server memory (MB) | 104.858 |
| clr enabled | CLR user code execution enabled in the server | 0 |
| optimize for ad hoc workloads | When this option is set, plan cache size is further reduced for single-use adhoc OLTP workload. | 0 |
| Database Mail XPs | Enable or disable Database Mail XPs | 0 |
| xp\_cmdshell | Enable or disable command shell | 0 |

Aus den Konfigurationswerten geht hervor, dass die Einstellungen für die Parallelisierung nicht optimal sind.

### Wartungsaufträge

Für einen optimierten Betrieb von Microsoft SQL Server empfehlen sich regelmäßige Wartungs- und Sicherungsaufgaben mit Hilfe des SQL Server Agenten auszuführen. Mit Hilfe des Microsoft SQL Server Agenten können sich wiederholende Routineaufgaben terminiert werden und in einem Regelbetrieb aufgenommen werden.

Bei der Untersuchung wurden die folgenden Aufträge in Microsoft SQL Server vorgefunden:

|  |  |  |
| --- | --- | --- |
| Auftrag | Eigentümer | Erstellungsdatum |
| CommandLog Cleanup | sa | 02.04.2015 14:52 |
| DatabaseBackup - SYSTEM\_DATABASES - FULL | sa | 02.04.2015 14:52 |
| DatabaseBackup - USER\_DATABASES - DIFF | sa | 02.04.2015 14:52 |
| DatabaseBackup - USER\_DATABASES - FULL | sa | 02.04.2015 14:52 |
| DatabaseBackup - USER\_DATABASES - LOG | sa | 02.04.2015 14:52 |
| DatabaseIntegrityCheck - SYSTEM\_DATABASES | sa | 02.04.2015 14:52 |
| DatabaseIntegrityCheck - USER\_DATABASES | sa | 02.04.2015 14:52 |
| IndexOptimize - USER\_DATABASES | sa | 02.04.2015 14:52 |
| Output File Cleanup | sa | 02.04.2015 14:52 |
| sp\_delete\_backuphistory | sa | 02.04.2015 14:52 |
| sp\_purge\_jobhistory | sa | 02.04.2015 14:52 |

Alle oben genannten Aufträge stammen aus den Maintenance-Ressourcen von Ola Hallengren[[8]](#footnote-8) und sind ein in der Community von Microsoft anerkannter und akzeptierter Standard für die täglichen Routineaufgaben zur Wartung von Datenbanken. Bei der Untersuchung der Aufträge wurde festgestellt, dass die Aufträge zwar korrekt implementiert wurden; sie wurden jedoch nie terminiert und somit auch nie ausgeführt!

### Cost Threshold For Parallelism

Die Option „cost threshold for parallelism“[[9]](#footnote-9) dient zur Angabe des Schwellwerts, ab dem Microsoft SQL Server parallele Pläne für Abfragen erstellt und ausführt. SQL Server erstellt nur dann einen parallelen Plan für eine Abfrage, wenn die geschätzten Kosten für die Ausführung eines seriellen Plans für die gleiche Abfrage höher sind als der in „cost threshold for parallelism“ festgelegte Wert. Die Kosten beziehen sich auf eine **geschätzte Zeit in Sekunden**, die für das Ausführen des seriellen Plans bei einer bestimmten Hardwarekonfiguration benötigt wird.

### Max Degree of Parallelism

Wenn SQL Server auf einem Computer mit mehreren Cores ausgeführt wird, wird der am besten geeignete Grad an Parallelität von Microsoft SQL Server erkannt. Hierbei handelt es sich um die Anzahl an Prozessoren, die für die Ausführung einer einzigen Anweisung eingesetzt werden. Dies gilt für die einzelnen Ausführungen paralleler Pläne. Sie können mithilfe von „max degree of parallelism“[[10]](#footnote-10) die Anzahl der Prozessoren beschränken, die für die Ausführung paralleler Pläne verwendet werden.

Die Einstellung „1“ wird verwendet, da neben der zu untersuchenden Datenbank noch eine SHAREPOINT[[11]](#footnote-11)-Umgebung auf der Instanz gehostet wird.

### Wait Stats

Jede Aktion in Microsoft SQL Server wird protokolliert. Diese Protokollierung zeichnet auf, wieviel Zeit aufgewendet wurde, um auf eine Ressource zu warten. Alle folgenden Wait Stats-Auswertungen sind Aufzeichnungen SEIT dem Neustart.

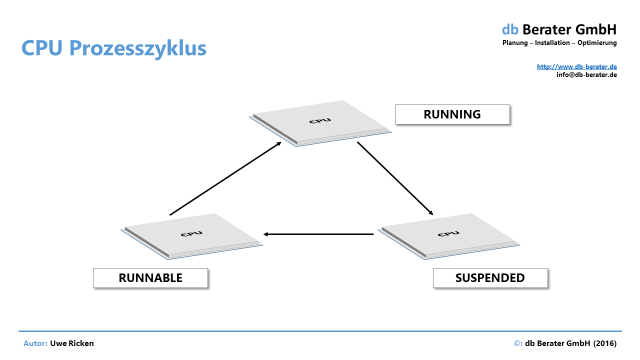


Abbildung 2: Arbeitszyklus eines Prozesses in Microsoft SQL Server

Der LifeCycle eines Prozesses / Threads in Microsoft SQL Server beginnt mit der Zuteilung von Rechenzeit. In diesem Moment ist der Prozess im Status „RUNNING“. In zwei Situationen wird der Prozess gezwungen, die Rechenzeit an einen anderen Prozess zu übergeben:

* Es werden weitere Ressourcen benötigt (Daten / Sperren / …)
* Das Quantum von 4ms ist abgelaufen

Wenn Ressourcen für die Bearbeitung des Prozesses nicht unmittelbar zur Verfügung stehen, muss der Prozess auf die angeforderten Ressourcen warten und wechselt in den SUSPENDED Moduls. Sobald alle Ressourcen vorhanden sind, kann der Prozess wieder CPU-Zeit anfordern und wechselt in den [RUNNABLE]-Status. Dieser Status ist wie eine Warteschlange nach dem FIFO[[12]](#footnote-12)-Prinzip aufgebaut.

Das Quantum von 4ms wird erreicht, wenn einem Prozess Rechenzeit zugeteilt wurde und dieser Zeitraum (4ms) nicht ausreicht, um den Prozess abzuschließen. In diesem Fall muss der Prozess die Rechenzeit an den nachfolgenden Prozess abgeben und sich erneut in die Warteschlange einreihen.

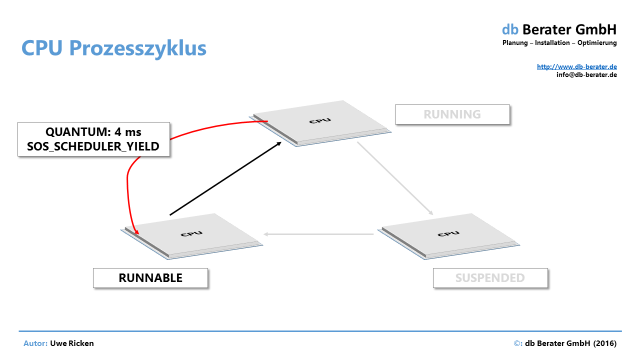


Abbildung 3: Protokollieren eines SOS\_SCHEDULER\_YIELD

Sind in dieser Warteschlange zu viele Prozesse, müssen sie auf die Zuteilung der CPU-Zeit warten. Diese Zeit wird als Signaltime bezeichnet. In einem gut ausgestatteten System sollte diese Zeit bei <=0.0001 Sekunden liegen.

Bei der Auswertung der Wartevorgänge des SQL Servers sind folgende Werte ermittelt worden:

| WaitType | Wait\_S | Resource\_S | Signal\_S | WaitCount | Percentage |
| --- | --- | --- | --- | --- | --- |
| ASYNC\_NETWORK\_IO[[13]](#footnote-13) | 43.630,44 | 43.543,62 | 86,82 | 3.090.482 | 50,58 |
| BACKUPIO[[14]](#footnote-14) | 20.721,99 | 20.711,94 | 10,05 | 2.328.490 | 24,02 |
| OLEDB[[15]](#footnote-15) | 5.440,65 | 5.440,65 | 0,00 | 1.038.739.747 | 6,31 |
| BACKUPTHREAD[[16]](#footnote-16) | 3.786,70 | 3785,70 | 1,00 | 13086 | 4,39 |
| PREEMPTIVE\_OS\_AUTHENTICATIONOPS[[17]](#footnote-17) | 3.110,99 | 3110,99 | 0,00 | 8797310 | 3,61 |
| WRITELOG[[18]](#footnote-18) | 2.555,18 | 2216,20 | 338,99 | 9851276 | 2,96 |
| ASYNC\_IO\_COMPLETION[[19]](#footnote-19) | 1.610,75 | 1610,68 | 0,07 | 1043 | 1,87 |
| SOS\_SCHEDULER\_YIELD[[20]](#footnote-20) | 979,80 | 29,60 | 950,20 | 37626400 | 1,14 |

Das Ergebnis der Auswertung zeigt, dass das System primär auf Clientaktionen warten muss, da entweder:

* Eine sehr große Datenmenge angefordert wird
* Der Client die Datenmenge nicht effektiv genug annimmt/weiterverarbeitet.

| WaitType | Avg Wait\_S | Suspended (s) | Runnable (s) |
| --- | --- | --- | --- |
| ASYNC\_NETWORK\_IO | 0,0141 | 0,0141 | 0,0000 |
| BACKUPIO | 0,0089 | 0,0089 | 0,0000 |
| OLEDB | 0,0000 | 0,0000 | 0,0000 |
| BACKUPTHREAD | 0,2894 | 0,2893 | 0,0001 |
| PREEMPTIVE\_OS\_AUTHENTICATIONOPS | 0,0004 | 0,0004 | 0,0000 |
| WRITELOG | 0,0003 | 0,0002 | 0,0000 |
| ASYNC\_IO\_COMPLETION | 1,5443 | 1,5443 | 0,0001 |
| SOS\_SCHEDULER\_YIELD | 0,0000 | 0,0000 | 0,0000 |

Das System verarbeitet applikationsrelevant ASYNC\_NETWORK\_IO sowie OLDEB Wartevorgänge. Da die obigen Messungen für die vollständige Instanz gelten, können die Wartevorgänge nicht einzelnen Applikationen zugeordnet werden.

## Datenbanken

### TEMPDB

Die Größe und die physische Platzierung der TEMPDB-Datenbank können sich auf die Leistung eines Systems negativ auswirken. Wird für die TEMPDB eine zu kleine Größe definiert, muss bei jedem Neustart der SQL Server-Instanz ein Teil der Verarbeitungslast des Systems dazu aufgewendet werden, die TEMPDB -Datenbank automatisch auf den Umfang zu vergrößern, der zum Unterstützen der anfallenden Arbeitsauslastung erforderlich ist. Dieser zusätzliche Aufwand kann vermieden werden, indem die TEMPDB -Daten- und Protokolldatei vergrößert wird.

Allen TEMPDB -Dateien sollte im Voraus Speicherplatz zugewiesen werden, indem die Dateigröße auf einen Wert festlegt wird, der hoch genug ist, um der typischen Arbeitsauslastung in der Umgebung gerecht zu werden. Dadurch wird verhindert, dass die TEMPDB -Datenbank zu häufig vergrößert wird, was zu Leistungseinbußen führen kann. Durch Verwenden mehrerer Dateien werden TEMPDB -Speicherkonflikte reduziert, und die Skalierbarkeit wird deutlich verbessert.

Die Systemdatenbank TEMPDB ist aktuell wie folgt konfiguriert:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | physical\_name | type\_desc | size\_mb | used\_mb |
| tempdev | **C:\Progr…\tempdb.mdf** | **ROWS** | **475.000000** | **102.375000** |
| templog | C:\Progr…\templog.ldf | LOG | 128.000000 | 59.500000 |
| tempdev\_02 | C:\Progr…\tempdev\_02.ndf | ROWS | 375.000000 | 97.250000 |
| tempdev\_03 | C:\Progr…\tempdev\_03.ndf | ROWS | 375.000000 | 126.687500 |
| tempdev\_04 | C:\Progr…\tempdev\_04.ndf | ROWS | 375.000000 | 93.000000 |
| tempdev\_05 | C:\Progr…\tempdev\_05.ndf | ROWS | 375.000000 | 104.187500 |
| tempdev\_06 | C:\Progr…\tempdev\_06.ndf | ROWS | 375.000000 | 103.000000 |
| tempdev\_07 | C:\Progr…\tempdev\_07.ndf | ROWS | 375.000000 | 98.937500 |
| tempdev\_08 | **C:\Progr…\tempdev\_08.ndf** | **ROWS** | **475.000000** | **112.250000** |

Zwei Dateien in TEMPDB besitzen unterschiedliche Größen. Durch die unterschiedlichen Größen in den Datenbankdateien werden Hotspots generiert, die ein häufigeres Schreiben in die größeren Dateien erzwingen (Round Robin Verfahren). Diese „Hotspots“ kommen dadurch zustande, dass Microsoft SQL Server immer versucht, alle Dateien einer Dateigruppe gleichmäßig (relativ) zu füllen. Wird versucht, alle Dateien zu 50% zu befüllen, müssen in Datei 1 und Datei 8 50MB mehr geschrieben werden.

Es wurde ebenfalls festgestellt, dass TEMPDB auf dem Systemlaufwerk abgelegt ist (Standardverzeichnis). Durch die Speicherung von TEMPDB auf dem Systemlaufwerk besteht die Gefahr, dass

Die Datenbank TEMPDB liegt vollständig auf dem Systemlaufwerk und sollte möglichst auf ein dediziertes Laufwerk verschoben werden.

### Benutzerdatenbank(en)

Hinweis

Bezüglich der Datenbanken werden ausschließlich die Datenbanken berücksichtigt, die im Rahmen der Applikation verwendet werden. Andere Datenbanken waren nicht Ziel der Untersuchung.

| Name | level | type | Physical location | size\_MB | growth\_MB | used\_MB |
| --- | --- | --- | --- | --- | --- | --- |
| <Name> | 100 | ROWS | <Path> | 72.795.75 | 100 | 67.358.38 |
| <Name> | 100 | LOG | <Path> | 26.00 | 000 | 8.26 |

Die Datenbank <Name> läuft im Kompatibilitätsmodus von Microsoft SQL Server 2008 und ist vom Vergrößerungsintervall eher unzureichend dimensioniert. Auf Grund der aktuellen Größe sowie eines Speicher-/Nutzungsverhältnis von 92,50% ist eine unmittelbare Aktion nicht erforderlich.

Die Datenbank wird im Wiederherstellungsmodus SIMPLE betrieben! Dieses Wiederherstellungsmodell keine zeitliche Wiederherstellung zu; es kann nur eine Wiederherstellung von der letzten Vollsicherung / Differenzialsicherung gewährleistet werden.

## Indexpflege

Die Indexe innerhalb der untersuchten Datenbanken befinden sich in einem nicht optimalen Zustand. Einige Indexe sind hochgradig fragmentiert. Heaps weisen einen sehr hohen Anteil an „Forwarded Records“ auf.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object\_Name | Index\_name | Fragm. % | page\_count | Forw recs |
| [dbo].<Name1> | \*\*\* HEAP \*\*\* | 97,14 % | 80.557 | **223.598** |
| [dbo],<Name1> | <IndexName> | 26,30% | 45.257 |  |
| [dbo],<Name1> | <IndexName> | 27,73 | 30860 |  |

Die Abbildung zeigt repräsentativ die Situation für die Indexe der Tabelle [dbo].<Name>. Bei der Tabelle handelt es sich um einen Heap, der 2 weitere nichtgruppierte Indexe besitzt. Die Tabelle selbst besitzt über 200.000 sogenannte „Forwarded Records[[21]](#footnote-21)“.

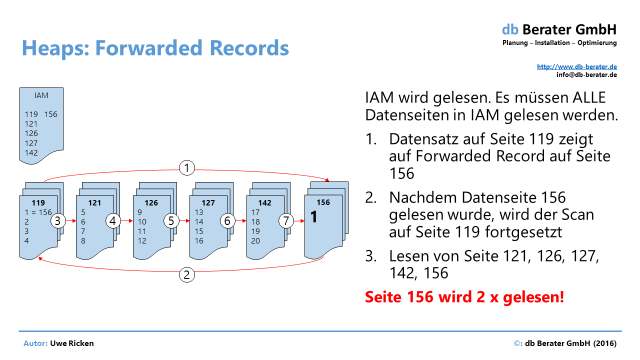


Abbildung 4: Nachteile beim Lesen von Forwarded Records

Einige der analysierten Tabellen sind als Heaps implementiert und weisen eine hohe Zahl von Forwarded Records auf.

## Statistiken

Der Abfrageoptimierer verwendet Statistiken zum Erstellen von Abfrageplänen, die die Abfrageleistung verbessern. Bei den meisten Abfragen generiert der Abfrageoptimierer automatisch die notwendigen Statistiken für einen hochwertigen Abfrageplan. In einigen Fällen müssen Sie weitere Statistiken erstellen oder den Abfrageentwurf ändern, um optimale Ergebnisse zu erzielen[[22]](#footnote-22).

Statistiken werden automatisch (<= SQL Server 2014) aktualisiert, wenn eine Relation mehr als 500 Datensätze besitzt und 20% + 500 der Datensätze geändert wurde.

Für sehr große Tabellensysteme sind diese Intervalle zu hoch; Indexe auf fortlaufenden Datenwerten unterliegen einem Problem, das man „Ascending Key Problem“ nennt. Hierbei sind in der Statistik die neuen – fortlaufenden – Werte nicht bekannt und der Abfrageoptimierer schätzt in diesem Fall, dass 1 Datensatz verwendet wird.

Für die existierenden Statistiken werden keine manuellen Aktualisierungen angestoßen. Die nachfolgende Tabelle zeigt einige untersuchte Statistiken sowie das Datum der letzten Aktualisierung.

| name | last\_updated | rows | Mod\_counter |
| --- | --- | --- | --- |
| … | 2016-11-03 11:22:26 | 5.572.564 | 20.059 |
| … | 2016-09-08 10:51:59 | 5.441.376 | 375.078 |
| … | 2016-09-08 10:50:06 | 5.441.376 | 375.108 |
| … | 2016-08-31 13:02:59 | 1.907.456 | 339.624 |
| … | 2016-08-30 08:05:36 | 5.418.637 | 439.725 |
| … | 2016-08-29 14:09:37 | 1.905.933 | 349.531 |
| … | 2016-08-29 14:09:36 | 1.905.933 | 349.531 |
| … | 2016-08-26 13:39:29 | 5.413.314 | 460.844 |

Die Tabelle zeigt für die Tabelle [dbo].<Name> die indexierten Statistiken. Es ist erkennbar, dass bis auf zwei Statistiken alle anderen Statistiken mehrere Monate alt sind.

# Handlungsempfehlungen

Im dritten Kapitel werden für die gefundenen Schwachpunkte Handlungsempfehlungen gegeben.

## SQL Server OS (Operation System)

Der untersuchte Microsoft SQL Server 2014 EE arbeitet mit dem PATCH-Level „12.0.2269.0“. Es wird empfohlen, das aktuelle Servicepack (12.00.5000) oder aktuelle CU (12.00.5522) einzuspielen.

## Laufwerke

Die Datenbanken sind nicht optimal im System verteilt. Aktuell steht ausschließlich ein Laufwerk zur Verfügung. Obwohl die gemessenen Latenzzeiten eine außergewöhnliche Performance gezeigt haben, ist es sinnvoll, die Datenbanken gemäß „Best Practice“ auf dem Storage zu verteilen.

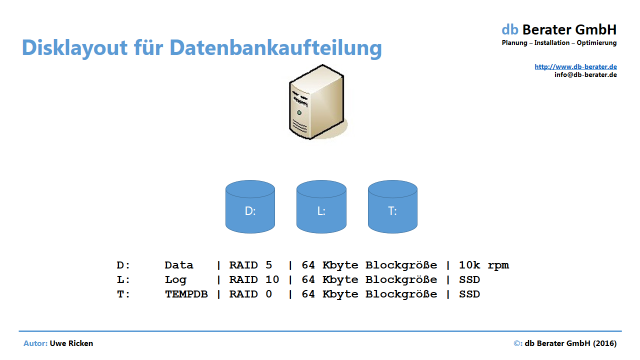


Abbildung 5: Aufteilung von Datenbankdateien

Alle Laufwerke, die von Datenbanken des Microsoft SQL Server verwendet werden, müssen neu formatiert werden. Eine ideale Blockgröße der formatierten Platten sollte 64 KBytes aufweisen (aktuell 4 KBytes).

<https://technet.microsoft.com/en-us/library/dd758814.aspx>

## Arbeitsspeicher

Der Arbeitsspeicher ist mehr als ausreichend dimensioniert. Aus Sicht des Autors reicht eine maximale Zuordnung von 64 GB vollkommen aus. Es wird empfohlen, im ersten Schritt die Option „max server memory“ auf den Wert 57.344 MB (56 GB) festzulegen.

## Aktivierung von standardisierten Traceflags

Folgende Traceflags sollten als Startparameter für Microsoft SQL Server festgelegt werden:

| Traceflag | Beschreibung |
| --- | --- |
| 1117 | Bewirkt bei Datenbanken mit mehreren Datenbankdateien ein gemeinsames Wachsen aller Dateien bei Platzbedarf (immer gleiche Größe für alle Dateien)  <https://msdn.microsoft.com/en-us/library/ms188396.aspx> |
| 1118 | Verhindert die Verwendung von Mixed Extents und verwendet ausschließlich Uniform Extents.  <https://msdn.microsoft.com/en-us/library/ms188396.aspx> |
| 2371 | Der Schwellwert für die Aktualisierung von Statistiken wird bei großen Tabellen reduziert  <https://support.microsoft.com/en-us/kb/2754171> |

## Security

Das Systemkonto [sa] sollte deaktiviert werden und ausschließlich ein administrativer Zugang über speziell berechtigte Domänen-Benutzer/Gruppen möglich sein.

Sowohl der lokale Administrator als auch der Domänen-Administrator sind SYSADMIN auf der untersuchten Instanz; es wird empfohlen, den lokalen Administrator aus der Instanz zu löschen!

Grundsätzlich wird empfohlen, mit Hilfe einer Domänengruppe (z. B. SQLAdmins) zu arbeiten. Diese Gruppe ist SYSADMIN auf dem Microsoft SQL Server.

## Trennung von Test- und Produktionssystem

Auf der Instanz wurden verschiedene Testdatenbanken (Name der Datenbanken) gefunden. Es wird dringend empfohlen, alle Testdatenbanken vom Produktivsystem zu entfernen und auf eine dedizierte SQL Server Instanz zu portieren um negative Ressourcenbelegungen durch Tests zu unterbinden.

## Wiederherstellungsmodell von Benutzerdatenbank SMP\_NG

Die Benutzerdatenbank [SMP\_NG] wird im Wiederherstellungsmodell SIMPLE betrieben. Dadurch ist eine „Just in Time“ Wiederherstellung ausgeschlossen und ein maximaler Datenverlust für ein Vollsicherungsintervall möglich.

Für die Datenbank <Name> soll das Wiederherstellungsmodus FULL konfiguriert werden, damit periodisch – im Rahmen eines definierten SLA – Protokollsicherungen durchgeführt werden können. Mit Hilfe dieser Protokollsicherungen kann ein Datenverlust bis auf wenige Minuten (Intervall der Protokollsicherungen) reduziert werden.

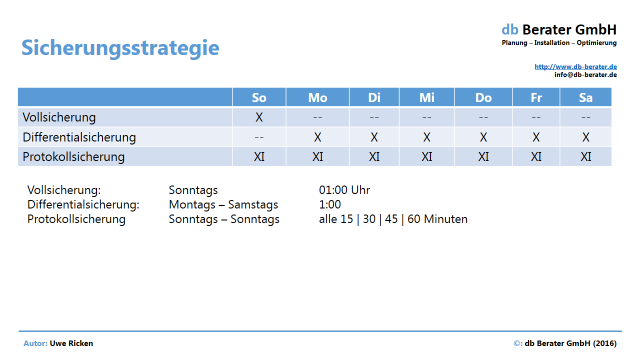


Abbildung 6: Mögliche Sicherungsstrategie

Durch ein Sicherungsmodell wie oben dargestellt, wird einmal pro Woche eine vollständige Sicherung durchgeführt. Anschließend wird täglich eine Differentialsicherung durchgeführt, um alle Änderungen seit der letzten Vollsicherung zu sichern. Zusätzlich wird jeden Tag in einem festen Intervall (gem. SLA) eine Protokollsicherung durchgeführt. Das Intervall bestimmt den maximalen Datenverlust.

## Sharepoint

Der SQL Server verwaltet neben Datenbanken für Benutzerapplikationen eine Sharepoint-Farm. Es wird empfohlen, die Sharepoint-Datenbanken auf einem dedizierten SQL Server zu betreiben, um mit einer höheren – möglichen – Parallelität zu arbeiten.

## Parallelisierung

Die Konfiguration für die eine Parallelisierung kann nicht justiert werden, da sich auf dem System neben der untersuchten Benutzerdatenbank ebenfalls eine SHAREPOINT-Installation. Für SHAREPOINT-Installationen ist zwingend vorgegeben, dass die Option „max degree of parallelism“ auf 1 gesetzt werden muss[[23]](#footnote-23). Damit wird den anderen Applikationen die Option genommen, mögliche Prozesse parallel zu verarbeiten.

## Maintenance

Auf dem untersuchten Microsoft SQL Server ist die vollständige Solution von Ola Hallengren (<http://ola.hallengren.com>) implementiert. Jedoch wurden die durch die Installation implementierten Aufträge nie terminiert und wurden somit noch nie ausgeführt.

Aktuell werden empfohlen, die Index-Maintenance sowie die Statistikaktualisierungen täglich auszuführen. Ein Zeitplan sollte danach bestimmt sein, welches Maintenance-Fenster zur Verfügung steht.

## Indexierung

In der untersuchten Datenbank werden einige Heaps verwendet. Heaps können nur skalieren, wenn mehrere Datenbankfiles verwendet werden. Das ist in der vorliegenden Untersuchung nicht gegeben. Es ist zu empfehlen, Tabellen mit hohem Änderungspotential (INSERT, UPDATE, DELETE) mit einem gruppierten Index zu versehen. Dadurch wird das Volumen der Datenbank reduziert (4 Bytes statt 8 Bytes bei einem Surrogatschlüssel vom Typen INT).

Durch die Verwendung von gruppierten Indexen werden Forwarded Records verhindert und die oben erwähnte Maintenance kann optimaler arbeiten (der Neuaufbau eines Heap bedeutet immer eine Aktualisierung ALLER nicht gruppierten Indexe).

# Zusammenfassung

Das System befindet sich in einem außerordentlich guten und leistungsfähigen Zustand. Insgesamt ist die zur Verfügung gestellte Hardware und Software für die Anforderungen mehr als ausreichend dimensioniert. Trotz teilweiser nachteiliger Konfiguration werden die Schwächen durch die sehr gute Hardware wieder wettgemacht.

Sofern Applikationsengpässe zu verzeichnen sind, kann mit sehr hoher Sicherheit ausgeschlossen werden, dass die Konfiguration des Microsoft SQL Server dafür die Ursache ist.

Um mögliche Ursachen bezüglich Applikationsengpässe zu analysieren, wurde angeregt, die Applikation für einen Tag serverseitig zu beobachten. Hierzu wurde dem Auftraggeber ein entsprechendes Skript zur Verfügung gestellt. Eine Auswertung der Ergebnisse steht noch aus. Daten aus dem Monitoring waren zum Zeitpunkt der Erstellung dieses Dokuments noch nicht vorhanden.

Ort, Datum

Autor

1. <https://de.wikipedia.org/wiki/Storage_Area_Network> [↑](#footnote-ref-1)
2. <https://msdn.microsoft.com/de-de/library/ms143694.aspx> [↑](#footnote-ref-2)
3. <https://msdn.microsoft.com/de-de/library/ms190768> [↑](#footnote-ref-3)
4. <https://technet.microsoft.com/de-de/library/ms189050.aspx> [↑](#footnote-ref-4)
5. <https://technet.microsoft.com/de-de/library/ms189313.aspx> [↑](#footnote-ref-5)
6. <https://technet.microsoft.com/de-de/library/ms191158.aspx> [↑](#footnote-ref-6)
7. <https://technet.microsoft.com/de-de/library/bb490957.aspx> [↑](#footnote-ref-7)
8. <http://ola.hallengren.com> [↑](#footnote-ref-8)
9. <https://technet.microsoft.com/de-de/library/ms188603(v=sql.105).aspx> [↑](#footnote-ref-9)
10. <https://technet.microsoft.com/de-de/library/ms181007(v=sql.105).aspx> [↑](#footnote-ref-10)
11. <https://products.office.com/de-de/sharepoint/collaboration> [↑](#footnote-ref-11)
12. First-In-First-Out [↑](#footnote-ref-12)
13. <http://documentation.red-gate.com/display/SM4/ASYNC_NETWORK_IO> [↑](#footnote-ref-13)
14. <http://documentation.red-gate.com/display/SM4/BACKUPIO> [↑](#footnote-ref-14)
15. <http://documentation.red-gate.com/display/SM4/OLEDB> [↑](#footnote-ref-15)
16. <http://documentation.red-gate.com/display/SM4/BACKUPTHREAD> [↑](#footnote-ref-16)
17. <https://www.sqlskills.com/help/waits/preemptive_os_authenticationops/> [↑](#footnote-ref-17)
18. <http://documentation.red-gate.com/display/SM4/WRITELOG> [↑](#footnote-ref-18)
19. <http://documentation.red-gate.com/display/SM4/ASYNC_IO_COMPLETION> [↑](#footnote-ref-19)
20. <http://documentation.red-gate.com/display/SM4/SOS_SCHEDULER_YIELD> [↑](#footnote-ref-20)
21. <http://www.db-berater.de/2015/08/forwarded-records-intern/> [↑](#footnote-ref-21)
22. <https://msdn.microsoft.com/de-de/library/ms190397.aspx> [↑](#footnote-ref-22)
23. <https://blogs.msdn.microsoft.com/rcormier/2012/10/24/you-shall-configure-your-maxdop-when-using-sharepoint-2013/> [↑](#footnote-ref-23)