

<http://bit.ly/1TR9vg7>

Analyse eines Microsoft SQL Server

Uwe Ricken

Uwe Ricken

db Berater GmbH

I am working with IT-systems since early 1990's and with the main focus on *Microsoft SQL Server* since version 6.0. I started with development of database applications in 1998 with a professional CRM-System based on Microsoft products (Microsoft Office and *Microsoft SQL Server*).

Since 2008 I'm focused exclusively on *Microsoft SQL Server* and since 2008 I'm working in 3rd level support teams for banks, insurances and global industries.

Since May 2013 I'm a **Microsoft Certified Master: SQL Server 2008** which was an amazing way into the depth of *Microsoft SQL Server*.

In July 2013 I have been awarded with the MVP Award for *Microsoft SQL Server*.

www: <http://www.db-berater.de>
email: uwe.ricken@db-berater.de
blog: <http://www.sqlmaster.de>
twitter: [@dbberater](https://twitter.com/dbberater)
xing: http://www.xing.com/profile/Uwe_Ricken



Wann ist eine Untersuchung erforderlich?

- Berater / Consultant
 - Abnahme eines durch Drittanbieter installierten Systems
 - Problemanalyse
 - Optimierungsanalyse
- DBA
 - Problemanalyse
 - Optimierungsanalyse (na ja)

Workflow eines Datenbank-Requests

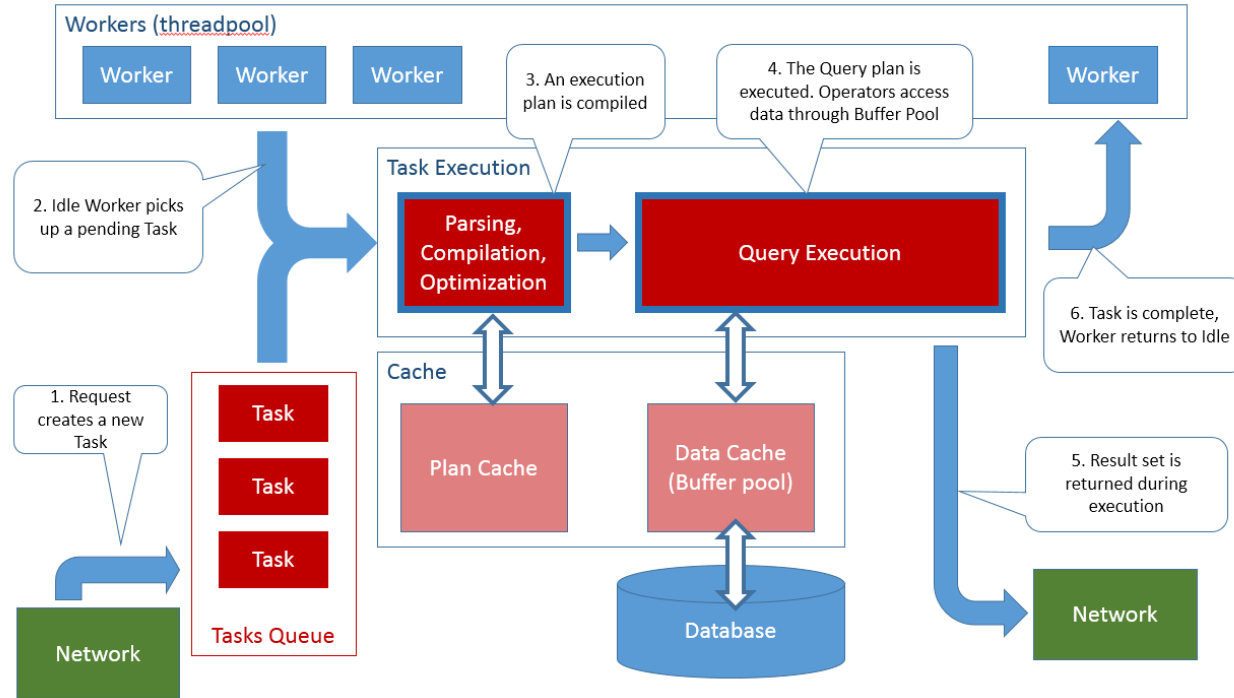


Abbildung: <http://rusanu.com/2013/08/01/understanding-how-sql-server-executes-a-query/>

Wichtigste Konfigurationseinstellungen

- Installation
 - Aktivierung von Instant File Initialization
 - Dediziertes Storage für Daten / Logs / TEMPDB
 - Formatierung der Disks mit 64 KBytes
 - Verwendung mehrerer Dateien für TEMPDB
 - ≤ 8 Cores = 1 File / Core
 - > 8 Cores = 8 Files
 - Limitierung des verfügbaren RAM (10% für Betriebssystem)

Wichtigste Konfigurationseinstellungen

- Betrieb
 - Aktivierung von Traceflags
 - 1117: Gleichmäßige Vergrößerung aller Dateien einer Filegroup
 - 1118: Verwendung von UNIFORM Extents
 - 2371: Dynamische Anpassung des Schwellwerts für Statistikaktualisierungen
 - Konfiguration von Parallelitätseinstellungen
 - MAXDOP max Anz. Cores / 1 (!) Socket
 - Cost Threshold (OLTP): 25
 - Cost Threshold (DWH): 50

Wichtigste Datenbankeinstellungen

- Datendateien
 - Ausreichende Initialgröße der Datei(en)
 - Ausgewogene Vergrößerungsintervalle für Datei(en)
 - GLEICHE Größeneinstellungen bei Multifile-Systemen
- Protokolldateien
 - Ausreichende Initialgröße
 - Sehr gut durchdachte Vergrößerungsintervalle
<= 96 VLF (!!!)

Berechnung von VLF für Protokolldateien

- ≤ 64 MB 4 VLF
- >64 MB und < 1.024 MB 8 VLF
- >1.024 MB 16 VLF

- Grenze: 96 VLF
- Ist: 1000 VLF = 30 GB
- $4 * 8$ GB = 32 GB
- $4 * 16$ = 64

- 1. Schritt: SHRINKFILE: 1 MB \Rightarrow 2 VLF
- 2. Schritt: MODIFY: 8.192 MB \Rightarrow 18 VLF 8 GB
- 3. Schritt: MODIFY: 8.192 MB \Rightarrow 34 VLF 16 GB
- 3. Schritt: MODIFY: 8.192 MB \Rightarrow 50 VLF 24 GB
- 4. Schritt: MODIFY: 8.192 MB \Rightarrow 66 VLF 32 GB

Vielen Dank für Ihre Aufmerksamkeit

Uwe Ricken

<http://bit.ly/1TR9vg7>

Agenda

- Betriebssystem
 - Überprüfung der Umgebung des Dienstkontos
001 - A01 - System Environment - Security.sql
 - Formatierte Blockgröße der vorhandenen Speichermedien
001 - A01 - System Environment - Security.sql
 - Berechtigungen des Dienstkontos
001 - A01 - System Environment - Security.sql
- Systemkonfiguration
 - CPU / RAM / IO
001 - A02 - System Environment - Drive Latency.sql

Agenda

- Datenbankkonfiguration
 - System- / Benutzerdatenbanken
 - TEMPDB (!!!)
 - Sizing / Growth / Storage
 - Verwendung
- CPU / Memory / PLE / Storage / Serverprotokolle
- SQL Server Agent Jobs
- Backups

Agenda

- Indexierung
 - Wie viele Indexe sind in der Datenbank vorhanden?
 - Wie viele Indexe werden für einzelne Tabellen genutzt?
 - Wie werden Indexe genutzt?
 - Welche Indexe können gelöscht werden?
 - Wo können neue Indexe die Performance verbessern
- Statistiken

Agenda

- Wie wird der Plan Cache verwendet
 - Werden AdHoc-Abfragen verwendet?
 - Werden parametrisierte Abfragen verwendet?

Workflow eines Datenbank-Requests

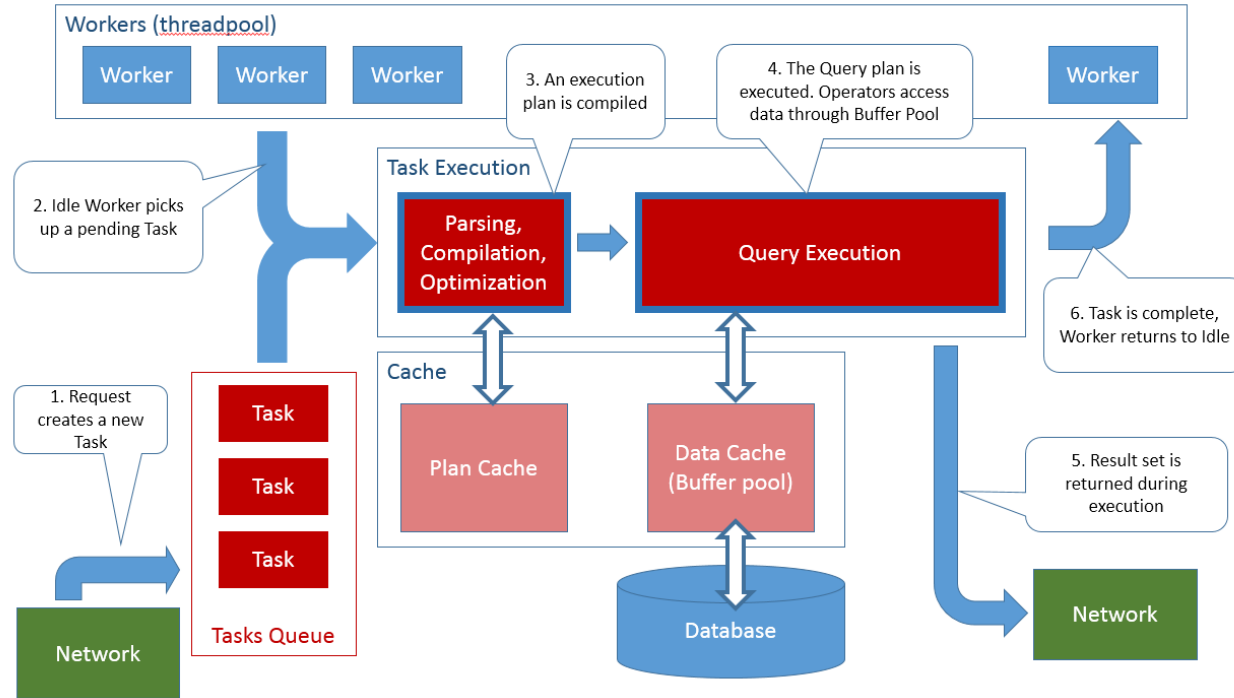


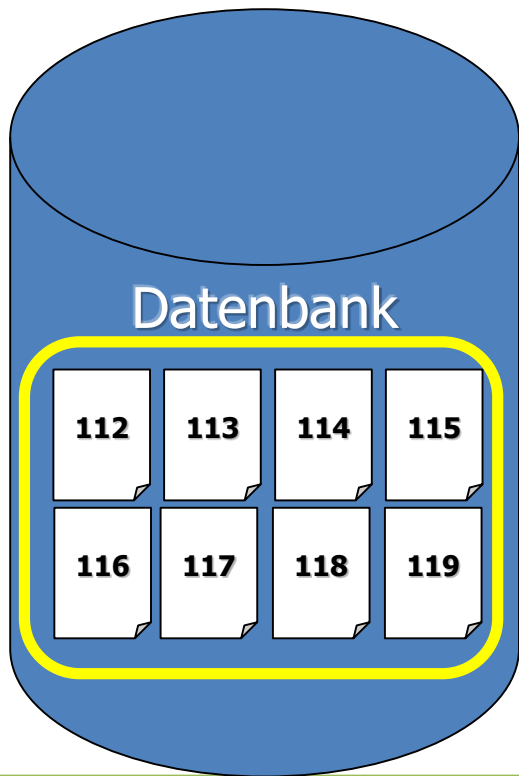
Abbildung: <http://rusanu.com/2013/08/01/understanding-how-sql-server-executes-a-query/>

Untersuchung des Betriebssystems

- Physikalische / Virtuelle Umgebung
 - Wie viele CORES / RAM?
 - Gibt es mehrere HDD im System?
 - Wie sind die HDD formatiert?

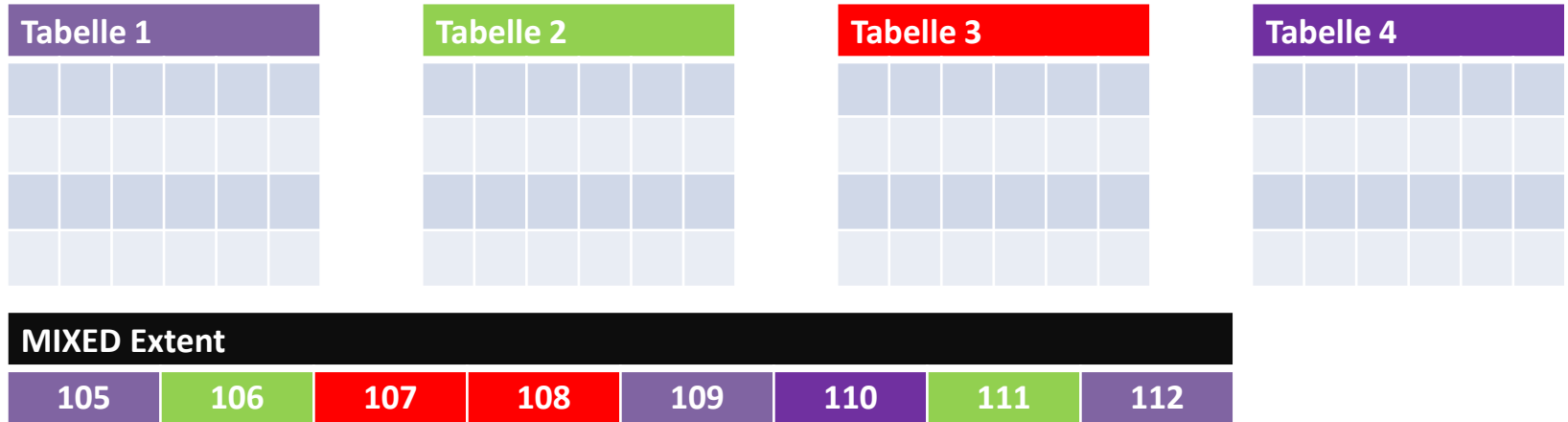
Datenbankstrukturen

- System- und Benutzerdaten werden in Datenseiten gespeichert.
- Jede Datenseite besitzt eine **feste** Größe von 8.192 Bytes:
 - 96 Bytes für den Page Header
 - 8.060 Bytes für Daten
 - 36 Bytes für Slot-Array
- 8 zusammenhängende Datenseiten werden in Extents gespeichert:
 - Mixed Extents speichern Datenseiten von verschiedenen Objekten
 - Uniform Extents speichern Datenseiten von einem Objekt
- Systemrelevante Data Pages:
 - PFS (Page Free Space):
 - SGAM (Shared Global Allocation Map):
 - GAM (Global Allocation Map):
 - IAM (Index Allocation Map):



MIXED Extents

- Mixed Extents speichern Informationen verschiedener Objekte



UNIFORM Extents

- Uniform Extents speichern Informationen eines einzelnen

Tabelle 1					

Tabelle 2					

Tabelle 3					

Tabelle 4					

MIXED Extent							
105	106	107	108	109	110	111	112

UNIFORM Extent							
113	114	115	116	117	118	119	120

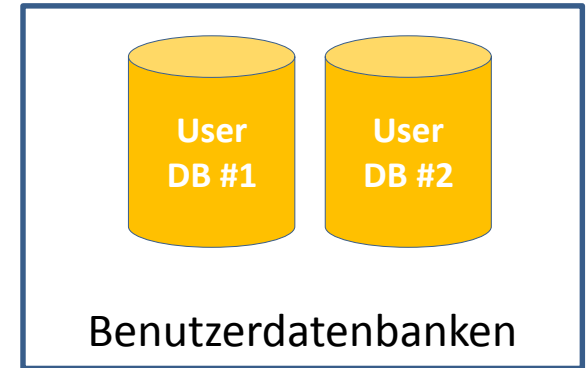
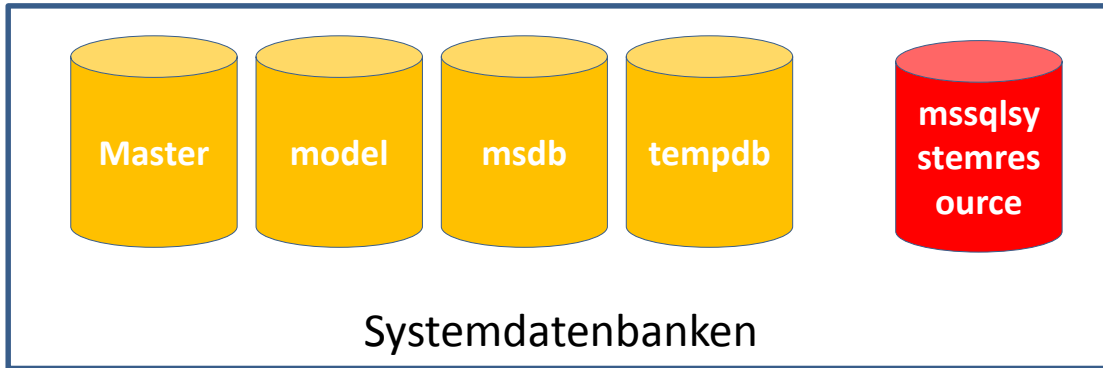
Betriebsumfeld des SQL Server

- Wie viel RAM wird dem SQL Server zugeteilt?
- Kann SQL Server alle Prozessoren verwenden?
- Welche Sicherheitseinstellungen sind konfiguriert worden?
 - Windows Authentifizierung
 - Mixed Authentifizierung
- Welche Berechtigungen hat das Dienstkonto?
 - Ist Instant File Initialization aktiviert?
 - Ist „Lock Pages in Memory“ aktiviert?

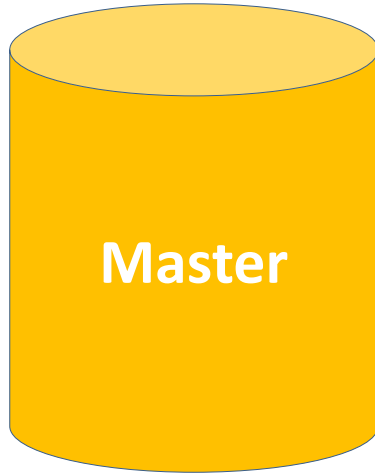
Betriebsumfeld des SQL Server

- Datenbankeinstellungen
 - FILLFACTOR
 - Standardverzeichnisse für Datenbanken
- Erweiterte Konfigurationseinstellungen
 - Max Degree of Parallelism
 - AdHoc Workloads
 - Gibt es konfigurierte Ressource-Governour-Einstellungen

System- und Benutzerdatenbanken

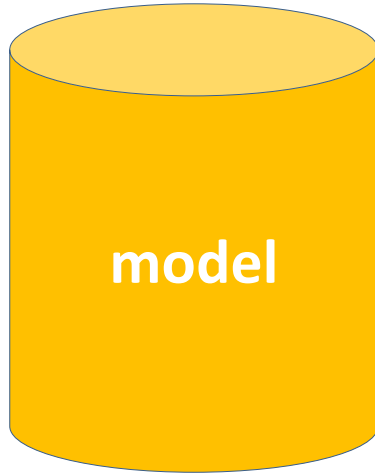


master-Datenbank



- Die Master-Datenbank erfasst alle Informationen auf Systemebene für ein SQL Server-System. Dazu gehören z. B. Metadaten wie Anmeldekonto, Endpunkte, verknüpfte Server und Systemkonfigurationseinstellungen.
- In SQL Server werden Systemobjekte nicht mehr in der Master-Datenbank gespeichert. Stattdessen werden sie in der Ressourcen-Datenbank gespeichert.
- Die master-Datenbank verwaltet ausserdem die Existenz und den Speicherort aller Benutzer-Datenbanken.
- SQL Server kann nicht starten, wenn die Master-Datenbank nicht verfügbar ist.

model-Datenbank



- Die Model-Datenbank wird als Vorlage für alle Benutzerdatenbanken verwendet, die auf einer Instanz von SQL Server erstellt werden.
- Der gesamte Inhalt der Modelldatenbank inklusive Datenbankoptionen wird in die neue Datenbank kopiert.
- Einige der Einstellungen von model werden auch für die Erstellung von tempdb beim Start verwendet, so dass die Model-Datenbank immer auf einem SQL Server-System vorhanden sein muss.
- Neu erstellte Benutzerdatenbanken verwenden das gleiche Wiederherstellungsmodell wie die Model-Datenbank.
- Die Voreinstellung ist vom Benutzer konfigurierbar.

msdb-Datenbank



- Die msdb-Datenbank wird vom SQL Server Agent für die Terminierung von Warnungen und Aufträgen sowie von anderen Funktionen wie SQL Server Management Studio, Service Broker und Database Mail verwendet.
- Standardmäßig verwendet msdb das einfache Wiederherstellungsmodell.
- Wenn Backup- und Wiederherstellungshistorien verwendet werden, sollte das vollständige Wiederherstellungsmodell für msdb verwendet werden.
- Wann immer Setup.exe verwendet wird, um die Systemdatenbanken neu zu erstellen, wird das Wiederherstellungsmodell von msdb automatisch auf einfach gesetzt!

tempdb-Datenbank



- Die tempdb-Systemdatenbank ist eine globale Ressource, die allen Benutzern zur für folgende Operationen zur Verfügung steht
 - Temporäre Benutzerobjekte, die explizit erstellt werden, wie zB: globale oder lokale temporäre Tabellen, temporäre gespeicherte Prozeduren
 - Tabellenvariablen oder Cursor.
 - Interne Objekte, die von der SQL Server Database Engine erstellt werden, z. B. Arbeitstabellen, um Zwischenergebnisse für Spools oder Sortierungen zu speichern.
 - Rowversionen, die von DML-Operationen in einer Datenbank generiert werden, die mit Hilfe von RCSI oder Snapshot-Isolation verwendet wird.
 - Multiple Active Results Sets (MARS) und AFTER-Trigger.
- Operationen innerhalb von tempdb sind minimal protokolliert.
- Tempdb wird jedes Mal neu erstellt, wenn SQL Server gestartet wird, so dass das System immer mit einer Kopie der Datenbank beginnt.

mssqlsystemresource-Datenbank



- Die Ressourcen-Datenbank ist eine **schreibgeschützte** Datenbank, die alle Systemobjekte von SQL Server enthält.
- SQL Server-Systemobjekte wie sys.objects werden physisch in der Ressourcen-Datenbank gespeichert, erscheint jedoch logisch im Sys-Schema jeder Datenbank.
- Die Ressourcen-Datenbank enthält keine Benutzerdaten oder Benutzer-Metadaten.

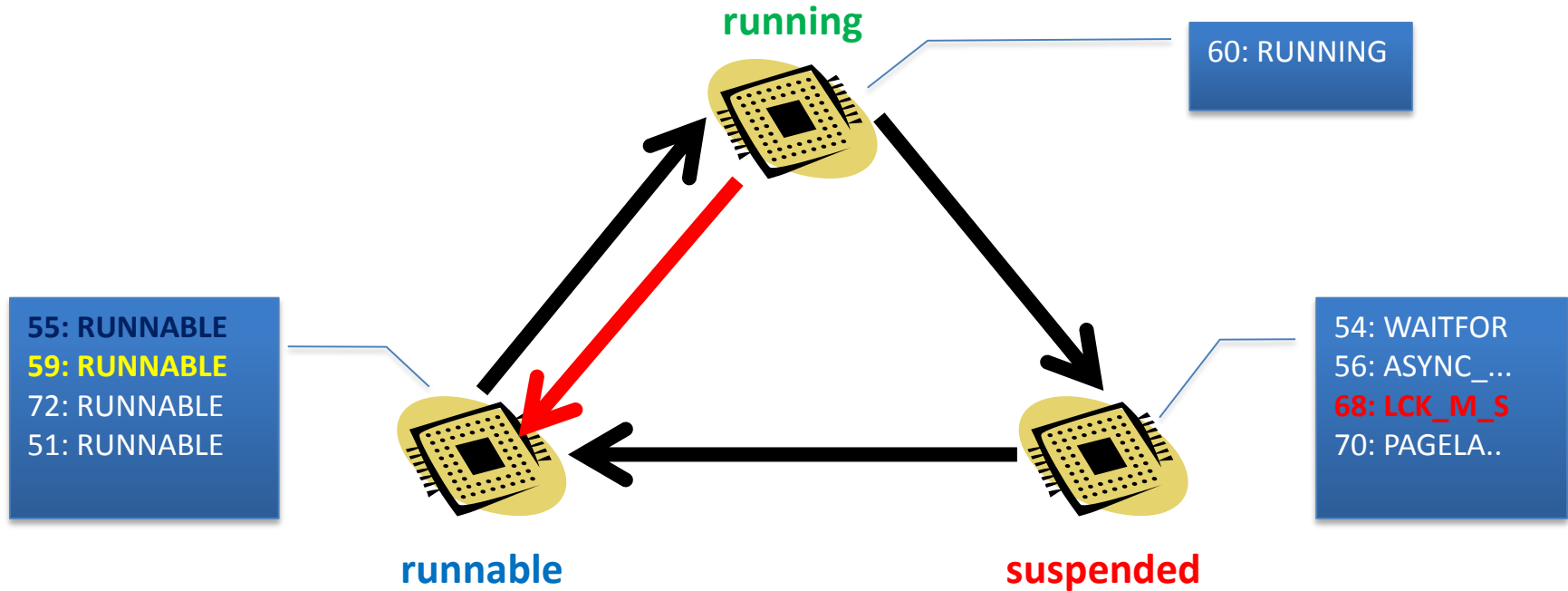
Konfiguration der Datenbanken

- Wie sind Datenbanken auf Storage verteilt?
- Gibt es eine Trennung zwischen DATA und LOG?
- Welchen Wachstumswert haben Datenbanken als Standard eingestellt?
- Wie groß sind die Datenbanken?
- Wie viel Platz ist in den Datenbanken noch frei?

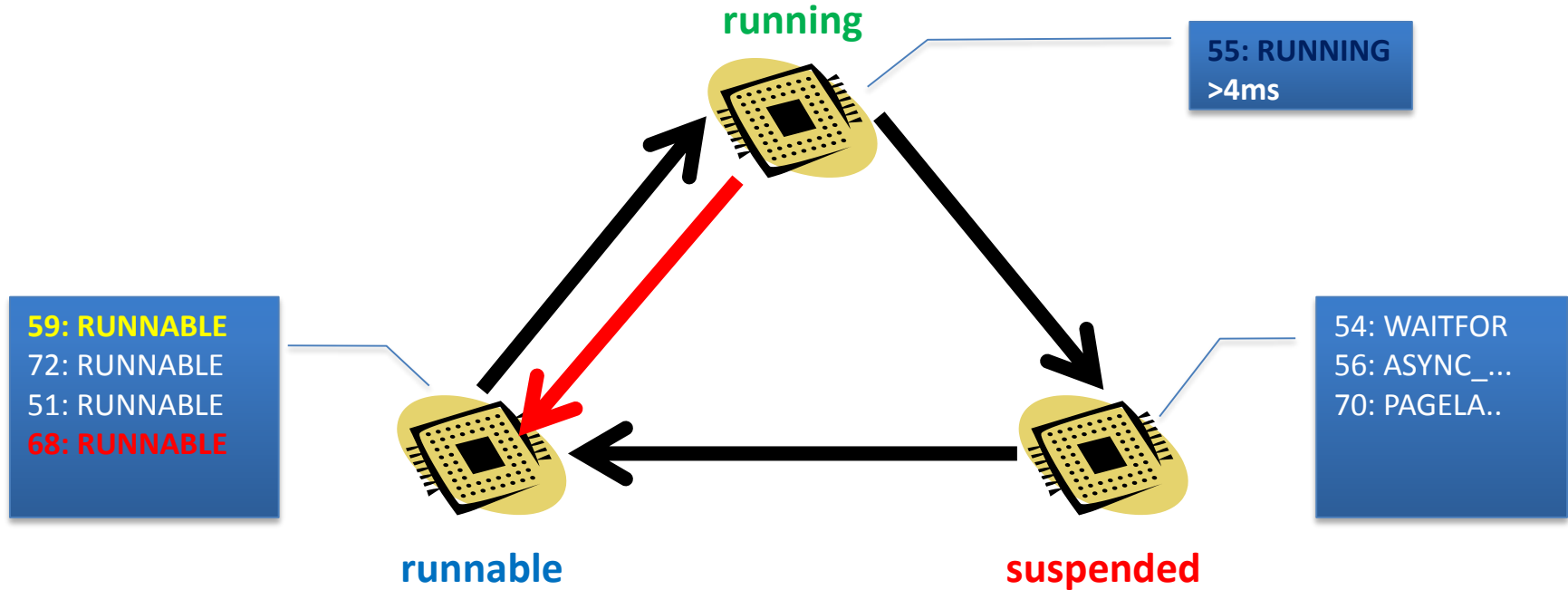
Schwachstelle TEMPDB

- Verwendungsarten
 - READ COMMITTED SNAPSHOT ISOLATION
 - HASH- /SORT-Operationen
 - Durch Benutzer angelegte temporäre Tabellen
 - Tabellenvariablen
 - ...
- Alle Operationen wollen Platz in TEMPDB reservieren
- Reservierung erfolgt über PFS / GAM / SGAM

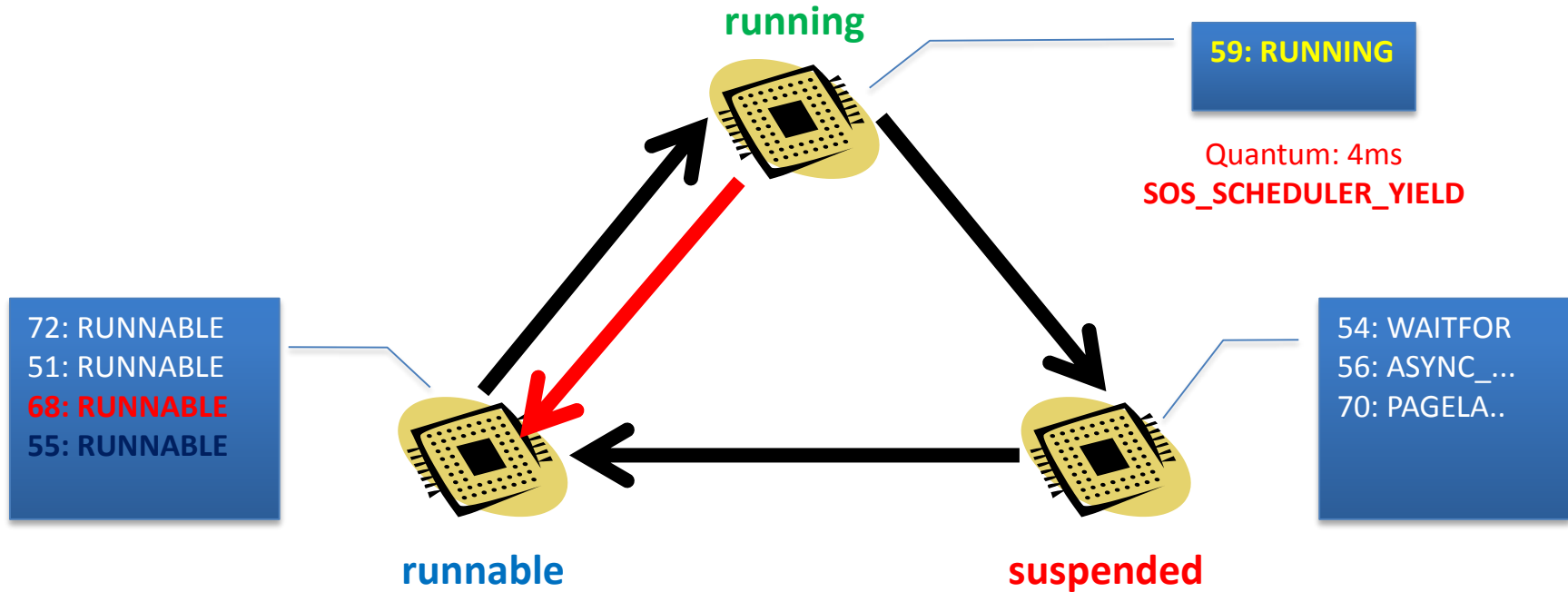
Wann muss Microsoft SQL Server warten?



Wann muss Microsoft SQL Server warten?



Wann muss Microsoft SQL Server warten?



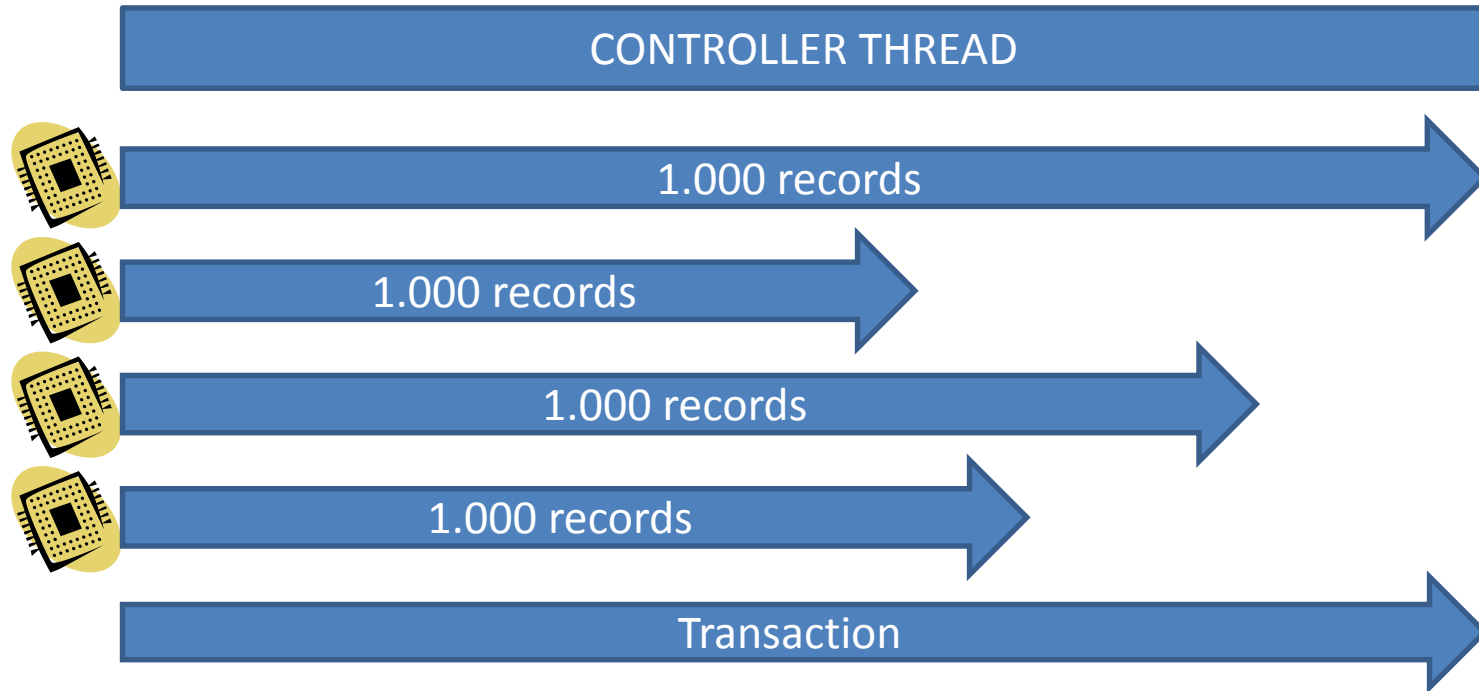
Populäre Wartevorgänge...

- [CXPACKET](#)
- [SOS_SCHEDULER_YIELD](#)
- [ASYNC_IO_COMPLETION](#)
- [ASYNC_NETWORK_IO](#)
- [THREADPOOL](#)
- [WRITELOG](#)
- PAGELATCH_XX
- PAGEIOLATCH_XX

CXPACKET

- Tritt auf, sobald eine Abfrage parallelisiert
 - Abfrage sollte untersucht werden und ggfls. durch Indexierung optimiert werden.
 - Um “effiziente” Parallelisierung sicherzustellen, sollte sichergestellt sein, dass Statistiken immer aktuell sind.

CXPACKET



PAGEIOLATCH_??

- Buffer Latch for eine Datenseite auf der physikalischen Disk
 - DT (destroy)
 - EX (exclusive)
 - KP (keep)
 - SH (shared)
 - UP (update)

PAGELATCH_??

- PAGELATCH_* latches sind interne Sperren, die von der SQL Server Engine verwendet werden, um konkurrierenden Zugriff auf Datenseiten im BUFFER POOL zu verhindern/verwalten.

Dokumentation

- Beginnen Sie mit Informationen zu Ihrer Person
- Beschreiben Sie den Grund für die Erstellung des Berichts
 - Gab es Probleme, die – plötzlich – aufgetreten sind?
 - Ist es eine initiale Untersuchung vor einem Go Live?
- Weisen Sie darauf hin, dass eine Hardware- Untersuchung nicht Bestandteil der Dokumentation ist, wenn Sie keinen Zugriff auf die einzelnen Hardwarekomponenten haben
- Legen Sie einen Struktur für Ihr Dokument fest, dass der Untersuchungsreihenfolge entspricht!

Dokumentation

- Begründungen durch Fußnoten mit Verweisen zur Dokumentation begleiten
 - BITTE KEINE FORENBEITRÄGE!
 - Möglichst nur Links zum Hersteller