Kapitel 2

Komponenten des Datenbankservers

In diesem Kapitel sehen wir uns die Bestandteile einer typischen Datenbankinfrastruktur an. Nach dieser Einführung erhalten Sie in den nachfolgenden Kapiteln genauere Einzelheiten darüber, wie Datenbanken gestaltet, implementiert und bereitgestellt werden.

Die Möglichkeit, SQL Server auf Linux auszuführen, ist zwar noch neu, aber Microsoft hat dafür gesorgt, dass er auch dort möglichst genauso läuft wie auf Windows. Wenn Unterschiede bestehen, werden wir Sie darauf hinweisen.

Unabhängig davon, welche Konfiguration Sie nutzen, besteht eine Datenbankinfrastruktur doch immer aus den folgenden vier grundlegenden Komponenten:

- Arbeitsspeicher
- Prozessor
- Permanentspeicher
- Netzwerk

Wir werden auch einige Möglichkeiten für Hochverfügbarkeit erwähnen, darunter auch Verbesserungen an Verfügbarkeitsgruppen, die in SQL Server 2017 vorgenommen wurden. Außerdem geben wir eine Einführung in Sicherheitsprinzipien, wobei wir uns unter anderem mit dem Zugriff auf lokale SQL Server-Instanzen mit Windows und Linux und auf Microsoft Azure SQL-Datenbanken beschäftigen. Abschließend werfen wir noch einen kurzen Blick auf die Virtualisierung.

Arbeitsspeicher

SQL Server ist so entworfen, dass er so viel Arbeitsspeicher nutzt, wie er braucht und wie Sie ihm geben. Standardmäßig ist der Arbeitsspeicher, auf den SQL Server zugreifen kann, nach oben nur durch das auf dem Servercomputer verfügbare physische RAM oder durch den von der verwendeten Edition nutzbaren Arbeitsspeicher begrenzt (wobei der jeweils niedrigere Wert gilt).

Arbeitssätze

Der physische Arbeitsspeicher, den das Betriebssystem SQL Server zur Verfügung stellt, wird als *Arbeitssatz* bezeichnet und vom SQL Server-Speicher-Manager wiederum in mehrere Teile zerlegt, wobei die beiden größten und wichtigsten der *Pufferpool* und der *Prozedurcache* (oder *Plancache*) sind.

In strengem Sinne beschreibt der Begriff Arbeitssatz nur physischen Speicher. Wie Sie aber in Kürze sehen werden, wird diese Definition bei der Pufferpoolerweiterung etwas unscharf.

Im Abschnitt »Konfigurationseinstellungen« von Kapitel 3 werfen wir einen genaueren Blick auf die Standardspeichereinstellungen.

Zwischenspeichern von Daten im Pufferpool

Zur Leistungssteigerung können Sie Daten im Arbeitsspeicher zwischenspeichern, da der Zugriff auf Daten im Arbeitsspeicher erheblich schneller abläuft als der Zugriff auf Daten im Permanentspeicher.

Der Pufferpool ist ein Cache im Arbeitsspeicher und besteht aus 8-KB-Datenseiten, bei denen es sich um Kopien von Seiten aus der Datenbankdatei handelt. Zu Anfang ist die Kopie im Pufferpool mit dem Original identisch, aber Änderungen an den Daten werden zunächst auf die Kopie angewendet (und im Transaktionsprotokoll vermerkt) und dann asynchron in die Datendatei geschrieben.

Wenn Sie eine Abfrage ausführen, fordert das Datenbankmodul die benötigte Datenseite vom Puffer-Manager an (siehe Abb. 2.1). Sind die Daten noch nicht im Pufferpool vorhanden, so tritt ein Seitenfehler auf. (Dies ist eine Vorkehrung des Betriebssystems, mit der die Anwendung darüber informiert wird, dass sich die Seite nicht im Arbeitsspeicher befindet.) Der Puffer-Manager ruft die Daten vom Speichersubsystem (für den Permanent-, nicht den Arbeitsspeicher) ab und schreibt sie in den Pufferpool. Sobald sich die Daten im Pufferpool befinden, wird die Abfrage fortgesetzt.

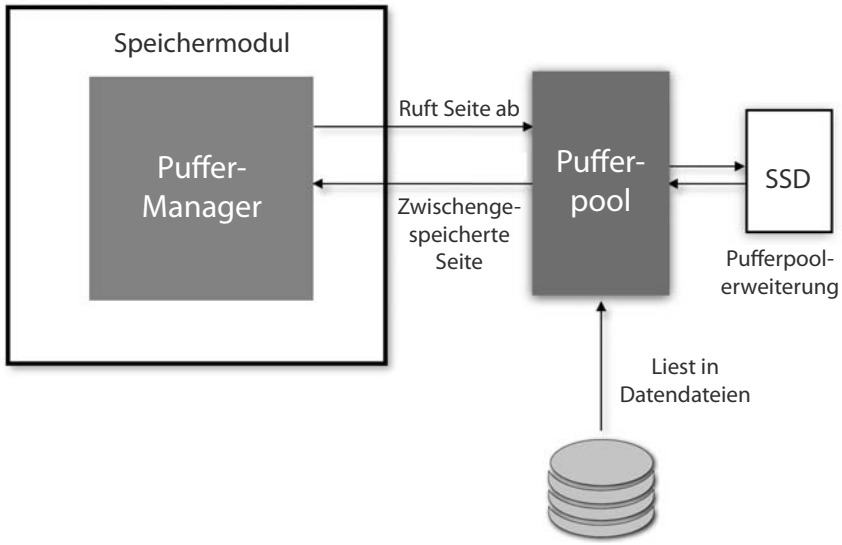

Abbildung 2.1 Pufferpool und Pufferpoolerweiterung

Der Pufferpool ist gewöhnlich der stärkste Verbraucher des Arbeitssatzes, da sich hier die Daten befinden. Übersteigt die Menge der für eine Abfrage angeforderten Daten die Kapazität des Pools, werden Datendateien auf die Festplatte ausgelagert. Dazu wird entweder die Pufferpoolerweiterung oder `TempDB` verwendet.

Die Pufferpoolerweiterung nutzt den nicht flüchtigen Speicher, um die Größe des Pufferpools zu erhöhen. Dadurch wird im Grunde genommen der Arbeitssatz der Datenbank vergrößert und eine Brücke zwischen dem Permanentspeicher mit den Datendateien und dem Pufferpool im physischen Arbeitsspeicher geschlagen.

Aus Leistungsgründen sollte es sich dabei um SSD-Speicher handeln, der direkt an den Servercomputer angeschlossen ist.

- **Wie Sie die Pufferpoolerweiterung einschalten, erfahren Sie im Abschnitt »Konfigurationseinstellungen« in Kapitel 3. Weitere Informationen über `TempDB` erhalten Sie im Abschnitt »Physische Datenbankarchitektur«, ebenfalls in Kapitel 3.**

Zwischenspeichern von Plänen im Prozedurcache

Im Allgemeinen ist der Prozedurcache kleiner als der Pufferpool. Wenn Sie eine Abfrage ausführen, kompiliert der Abfrageoptimierer einen Abfrageplan, um dem Datenbankmodul genau zu erklären, wie es die Abfrage ausführen soll. Um Zeit zu sparen, wird eine Kopie dieses Plans aufgehoben, damit er nicht bei jeder Ausführung der Abfrage neu kompiliert werden muss. In Wirklichkeit ist der Vorgang natürlich nicht ganz so einfach (so können Pläne auch entfernt werden, und triviale Pläne werden gar nicht erst zwischengespeichert), aber diese einfache Beschreibung reicht aus, um Ihnen eine grundlegende Vorstellung zu geben.

Der Prozedurcache wird vom Speicher-Manager in mehrere Cachespeicher aufgeteilt. Dieser Cache ist auch der Ort, an dem Sie nachsehen müssen, ob nur einmal verwendete Abfragepläne den Arbeitsspeicher belasten.

- **Weitere Informationen über zwischengespeicherte Ausführungspläne erhalten Sie in Kapitel 9 und auf *https://blogs.msdn.microsoft.com/blogdoezequiel/2014/07/30/too-many-single-use-plans-now-what*.**

Sperren von Seiten im Arbeitsspeicher

Wenn Sie die Richtlinie *Sperren von Seiten im Speicher* einschalten, ist Windows nicht mehr in der Lage, den Arbeitssatz zu »trimmen« (zu verkleinern).

Dadurch wird sichergestellt, dass Windows SQL Server bei Speicherdruck nicht um Ressourcen bringen oder SQL Server-Arbeitsspeicher in die Systemauslagerungsdatei von Windows Server verlegen kann, was die Leistung dramatisch verringern würde. Windows stiehlt nicht dreist Arbeitsspeicher von SQL Server, sondern tut dies nur als Reaktion auf Speicherdruck. In einem solchen Fall wird der Arbeitsspeicher aller Anwendungen beeinträchtigt.

Besteht jedoch keine Möglichkeit, den Druck durch die Speicheranforderungen anderer Anwendungen oder des virtuellen Hosts zu verringern, führt die Richtlinie *Sperren von Seiten im Speicher* dazu, dass Windows nicht mehr genug Arbeitsspeicher bereitstellen kann, um stabil zu laufen. Aus diesem Grunde darf *Sperren von Seiten im Speicher* nicht die einzige Methode sein, um die Speicherzuweisung von SQL Server zu schützen.

Bei diesem Problem muss also zwischen Stabilität und Leistung abgewogen werden, wobei Letzteres auf Systemen mit beschränkten Speicherressourcen und älteren Betriebssystemen im Vordergrund stand. Auf großen Servern mit Betriebssystemen ab Windows Server 2008 und insbesondere auf virtualisierten Systemen ist der Bedarf dafür, SQL Server mit dieser Richtlinie gegen Speicherdruck abzuschirmen, geringer, aber immer noch vorhanden.

Im Allgemeinen wird dazu geraten, die Richtlinie *Sperren von Seiten im Speicher* in SQL Server 2017 standardmäßig einzuschalten, wenn Folgendes gilt:

- Es handelt sich um einen physischen und nicht um einen virtuellen Servercomputer (siehe den Abschnitt »Freigabe von mehr als dem vorhandenen Arbeitsspeicher (übermäßige Zusicherung)« weiter hinten in diesem Kapitel).
- Das physische RAM ist größer als 16 GB (das Betriebssystem braucht selbst einen Arbeitssatz).
- Der Wert für den maximalen Arbeitsspeicher wurde entsprechend eingestellt (SQL Server kann nicht alles verwenden, was er sieht).
- Der Leistungsindikator *Arbeitsspeicher\Verfügbare MB* wird regelmäßig überwacht (um etwas Arbeitsspeicher frei zu halten).

In seinem Simple-Talk-Artikel auf *https://www.simple-talk.com/sql/database-administration/great-sql-server-debates-lock-pages-in-memory* erklärt Jonathan Kehayias diese Vorgehensweise genauer.

Speichereinschränkungen in den Editionen

Seit SQL Server 2016 SP1 haben viele Features der Enterprise Edition auch ihren Weg in die einfacheren Ausgaben gefunden. Das wurde offensichtlich getan, damit Softwareentwickler viel mehr Code schreiben können, der in allen Editionen des Produkts läuft.

Manche Merkmale sind zwar immer noch auf bestimmte Editionen beschränkt (etwa die Hochverfügbarkeit), doch andere sind in allen Editionen eingeschaltet, auch in Express, darunter Columnstore und In-Memory-OLTP. Allerdings kann nur in der Enterprise Edition das gesamte physische RAM für diese Features genutzt werden. Bei anderen Editionen bestehen Einschränkungen.

Expertentipp

In-Memory-OLTP

Für In-Memory-OLTP ist ein Mehraufwand von mindestens der doppelten Datenmenge für ein speicheroptimiertes Objekt erforderlich. Beispielsweise müssen für die exklusive Nutzung einer speicheroptimierten Tabelle von 5 GB mindestens 10 GB RAM zur Verfügung stehen. Bedenken Sie dies, bevor Sie dieses Feature in der Standard Edition einschalten.

Auch bei speicheroptimierten tabellenwertigen Funktionen müssen Sie in der Standard Edition aufpassen, da jedes neue Objekt Ressourcen erfordert. Zu viele davon können den Arbeitssatz aushungern und einen Absturz von SQL Server verursachen.

Mehr darüber erfahren Sie auf *https://docs.microsoft.com/sql/relationaldatabases/in-memory-oltp/requirements-for-using-memory-optimized-tables*.

CPU

Der Prozessor, auch CPU (Central Processing Unit) genannt, gilt als das »Gehirn« des Computers und ist der wichtigste Bestandteil des Systems. CPU-Geschwindigkeiten werden in Hertz (Hz) oder Zyklen pro Sekunde gemessen. Die Prozessoren von heute weisen Geschwindigkeiten in der Größenordnung von Gigahertz (GHz) oder Milliarden Zyklen pro Sekunde auf.

Moderne Systeme können mehrere Prozessoren aufweisen, die selbst mehrere Kerne haben können (wobei diese Kerne wiederum in mehrere virtuelle Kerne aufgeteilt sein können).

Für typische SQL Server-Arbeitslasten spielt die Geschwindigkeit eines einzelnen Kerns die größte Rolle. Weniger Kerne mit höheren Taktraten sind besser als mehr Kerne mit geringerer Geschwindigkeit, insbesondere bei den Editionen außer Enterprise.

Bei Systemen mit mehr als einer CPU kann jeder CPU ihr eigener Arbeitsspeicher zugewiesen werden. Das hängt von der physischen Architektur des Motherboards ab.

Simultanes Multithreading

Einige CPU-Hersteller haben die physischen Kerne ihrer Prozessoren in virtuelle Kerne aufgeteilt, um noch ein bisschen mehr Leistung aus ihnen herauszukitzeln. Dazu verwenden sie das sogenannte simultane Multithreading (SMT). Intel nennt diese Technik Hyper-Threading. Bei einem Intel Xeon mit 20 physischen Kernen sieht das Betriebssystem aufgrund der Verwendung von SMT 40 virtuelle Kerne.

Besonders undurchsichtig wird die Lage bei der Verwendung von SMT in virtuellen Maschinen, da das Gastbetriebssystem nicht mehr zwischen physischen und virtuellen Kernen unterscheiden kann.

Auf physischen Datenbankservern sollte SMT eingeschaltet werden. In virtuellen Umgebungen dagegen müssen Sie dafür sorgen, dass die virtuellen CPUs korrekt zugewiesen werden. Mehr darüber erfahren Sie im Abschnitt »Hardwareabstraktion durch Virtualisierung« weiter hinten in diesem Kapitel.

NUMA

Die Prozessoren sind die schnellsten Komponenten eines Systems und verbringen viel Zeit damit, auf Daten zu warten. Früher teilten sich alle CPUs eine RAM-Bank auf der Hauptplatine über einen gemeinsamen Bus. Als mehr Prozessoren hinzukamen, führte das zu Leistungsproblemen, da immer nur eine CPU auf einmal Zugriff auf das RAM hatte.

Eine Lösung dafür ist eine Mehrkanal-Speicherarchitektur, bei der es mehrere Kanäle zwischen den CPUs und dem RAM gibt, wodurch die Konkurrenz bei gleichzeitigem Zugriff verringert wird.

Eine praktikablere Lösung besteht jedoch darin, jedem Prozessor sein eigenes lokales physisches RAM zu geben, das nah beim jeweiligen CPU-Sockel platziert wird. Diese Vorgehensweise wird als NUMA (Non-Uniform Memory Access) bezeichnet. Der Vorteil besteht darin, dass jede CPU auf ihr eigenes RAM zugreifen kann, was die Verarbeitung erheblich beschleunigt. Wenn eine CPU allerdings mehr RAM benötigt, als ihr lokal zur Verfügung steht, muss sie Speicher von einem der anderen Prozessoren im System anfordern (*Fremdspeicherzugriff*), was die Leistung beeinträchtigt.

SQL Server ist NUMA-fähig. Wenn das Betriebssystem eine NUMA-Konfiguration auf Hardwareebene erkennt, bei der mehr als eine CPU angeschlossen ist, und jede CPU über ihr eigenes physisches RAM verfügt (siehe Abb. 2.2), dann verteilt SQL Server seine internen Strukturen und Dienstthreads über die NUMA-Knoten.

Seit SQL Server 2014 SP 2 richtet das Datenbankmodul automatisch NUMA-Knoten auf Instanzebene ein. Die dazu verwendete Technik wird als *Soft-NUMA* bezeichnet. Werden mehr als acht CPU-Kerne erkannt (einschließlich SMT-Kernen), werden automatisch Soft-NUMA-Knoten im Arbeitsspeicher angelegt.

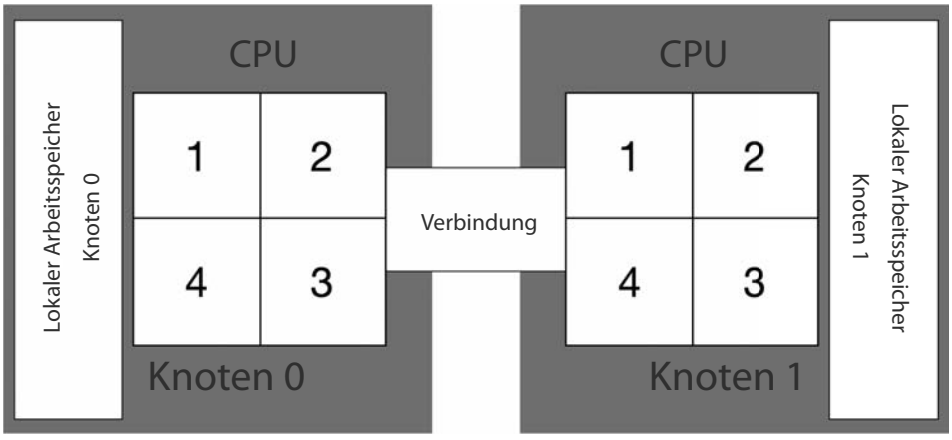

Abbildung 2.2 NUMA-Konfiguration mit zwei Prozessorsockeln

Expertentipp

Maximale Kernanzahl in den einzelnen Editionen

Die SQL Server Standard Edition ist künstlich beschränkt und kann nur maximal 24 physische Kerne nutzen. In einem System mit zwei 16-Kern-Prozessoren muss die Standard Edition für alle 32 Kerne lizenziert werden, obwohl sie acht davon gar nicht verwendet.

Außerdem erfolgt eine ungleichmäßige NUMA-Verteilung, da SQL Server die 16 Kerne der ersten CPU und acht Kerne der zweiten nutzt, es sei denn, dass Sie Affinitätseinstellungen einrichten (siehe den Abschnitt »Konfigurationseinstellungen« in Kapitel 3).

Wählen Sie die Hardware und die Edition für Ihre SQL Server-Installation daher sorgfältig aus. Wenn Sie mehrere VMS auf einem System installieren wollen, ist die Enterprise Edition die bessere Wahl, da sie für alle Kerne der Hardware lizenziert ist. Damit sind automatisch alle SQL Server-VMs abgedeckt, die Sie auf dieser Hardware einrichten.

Deaktivieren der Energiespareinstellungen

Moderne Systeme können Energiespareinstellungen nutzen, um den Stromverbrauch zu verringern. Das ist zwar gut für die Umwelt, aber schlecht für die Abfrageleistung, da die CPU-Geschwindigkeit herabgesetzt werden kann, um Strom zu sparen.

Schalten Sie in allen Betriebssystemen, auf denen Sie SQL Server ausführen, die Energiespareinstellung *Höchstleistung* ein, und überprüfen Sie, dass auch das BIOS entsprechend eingerichtet ist. Dedizierte VM-Hosts müssen vorübergehend heruntergefahren werden, um diese Änderung vorzunehmen.

Speichern von Daten

Wenn sich die Daten nicht im Arbeitsspeicher befinden, müssen sie an anderer Stelle abgelegt werden. Die Speichertechnologie hat sich in den letzten Jahren rapide weiterentwickelt. Als Permanentspeicher dienen heutzutage keine mechanischen Festplatten, in denen Scheiben mit magnetischen Oberflächen rotieren. Aus Macht der Gewohnheit werden die alten Bezeichnungen »Festplatte« und »Laufwerk« jedoch auch auf moderne, nicht flüchtige Speichersysteme angewendet.

Für SQL Server sollte das Speichersubsystem eine geringe Latenz aufweisen, sodass Lese- und Schreibvorgänge des Datenbankmoduls auf dem Laufwerk so schnell wie möglich ablaufen. Die folgende Liste nennt einige häufige Begriffe im Zusammenhang mit Speichergeräten.

- **Laufwerk** Das physische Speichergerät. Dabei kann es sich um ein mechanisches Laufwerk, ein Halbleiterlaufwerk (Solid-State Drive) der gleichen Bauform oder um Karten handeln, die direkt in das Motherboard eingesteckt werden.
- **Volume** Eine logische Darstellung des Permanentspeichers aus der Sicht des Betriebssystems. Bei einem Volume kann es sich um ein ganzes Laufwerk, einen Teil eines Laufwerks oder einen logischen Abschnitt eines Speicherarrays handeln. In Windows erhält ein Volume gewöhnlich seinen eigenen Laufwerksbuchstaben oder Bereitstellungspunkt.
- **Latenz** Die Latenz gibt an, wie lange es dauert, um Daten von einem Laufwerk zu lesen oder zu schreiben (Sekunden pro Lese- bzw. Schreibvorgang).

- **IOPS** IOPS (Input/Output Operations Per Second, also E/A-Vorgänge pro Sekunde) gibt die Anzahl der Lese- und Schreibvorgänge pro Sekunde an. Ein Speichergerät kann je nachdem, ob die E/A-Vorgänge sequenziell oder wahlfrei sind, unterschiedliche Leistungswerte zeigen. Der IOPS-Wert ist über die Warteschlangentiefe mit der Latenz verknüpft.
- **Warteschlangentiefe** Die Anzahl der ausstehenden Lese- und Schreibanforderungen in der Anforderungswarteschlange eines Speichergeräts. Je tiefer die Warteschlange, umso schneller das Laufwerk.

Die Leistung von SQL Server hängt direkt von der Speicherleistung ab. Durch den Übergang zu Virtualisierung und gemeinsam genutzten Speicherarrays hat sich der Schwerpunkt zu wahlfreien Datenzugriffsmustern verlagert. Geringe Latenzen und hohe IOPS-Werte für wahlfreien Zugriff sind daher für durchschnittliche SQL Server-Arbeitslasten von Vorteil.

In den folgenden beiden Abschnitten werden wir uns näher mit der bevorzugten Speicherkonfiguration für SQL Server beschäftigen.

Arten von Speicher

Es gibt zwei Hauptarten von nicht flüchtigem Speicher, nämlich den *mechanischen* und den *Halbleiterspeicher*.

Mechanische Festplatten

Herkömmliche rotierende Scheiben weisen aufgrund ihrer Form und ihrer Funktionsweise von Natur aus eine gewisse Latenz auf, nämlich die *Suchzeit*. Der Schreib/Lese-Kopf ist an einem Arm montiert, der die Oberfläche der Scheibe absuchen muss, während diese sich dreht, um die Bereiche zu finden, in denen die E/A-Operation vorgenommen werden soll. Sind die Daten auf der Scheibe fragmentiert, dauert der Zugriff länger, da der Kopf hin und her springen muss, um Daten oder freien Platz zu finden.

Die Standardschnittstellen für mechanische Laufwerke sind SATA (Serial ATA) und SAS (Serial Attached SCSI).

Je größer die Kapazität der Scheiben, umso schmaler werden die Spuren, was zu einer Verschlechterung der Leistung führt und die Wahrscheinlichkeit für mechanische Störungen und Datenbeschädigungen erhöht. Die Rotationsenergie der Scheibe selbst setzt eine physische Grenze für die Geschwindigkeit des Motors.

Kurz: Je höher die Kapazität einer mechanischen Festplatte ist, umso langsamer und fehleranfälliger wird sie.

Solid-State-Laufwerke

Bei der Solid-State- oder Halbleitertechnologie wird Flashspeicher genutzt. Dabei fallen keine Suchzeiten an, da jede Zelle, in der Daten gespeichert sind, fast unmittelbar erreicht werden kann. Dadurch ist Solid-State-Speicher sehr viel schneller als mechanischer Speicher.

Solid-State-Speichergeräte können verschiedene Formen annehmen. Bei Geräten für Endverbraucher ist ein 2,5"-Gehäuse mit SATA-Schnittstelle am weitesten verbreitet, wie es auch schon bei mechanischen Laptoplaufwerken verwendet wird. Dadurch ist es möglich, mechanische Speichergeräte einfach auszutauschen.

Bei Servern gibt es jedoch Flashspeicher verschiedener Formen. Für die lokale Speicherung kann eine PCIe-Schnittstelle (Peripheral Component Interconnect Express) verwendet werden, die direkt am Motherboard angeschlossen wird. Ein Beispiel dafür ist NVMe (Non-Volate Memory Express).

Die Solid-State-Technologie ist jedoch nicht perfekt. So ist in einer Zelle nur eine bestimmte Anzahl von Schreibvorgängen möglich, bevor sie ausfällt. Vielleicht haben Sie das schon selbst bei einem USB-Stick erlebt. Diese Geräte neigen dazu, nach intensivem Gebrauch zu versagen. Es gibt Algorithmen für den Verschleißausgleich, die Schreibvorgänge über die Zellen verteilen, um die Lebensdauer von Solid-State-Geräten zu verlängern.

Ein weiteres Problem von Flashspeichern ist die sogenannte *Write Amplification*. Wird auf einem mechanischen Laufwerk eine Datei überschrieben, so wird die ursprüngliche Datei zum Löschen markiert, aber noch nicht entfernt. Muss ein Laufwerk erneut in einen Bereich schreiben, so überschreibt es ihn, ohne zu entfernen, was vorher dort stand.

Solid-State-Laufwerke dagegen müssen den fraglichen Bereich löschen, bevor sie dort neue Daten schreiben, was auf Kosten der Leistung geht. Aufgrund der Zellengröße kann es bei kleineren Dateien auch erforderlich sein, einen Bereich zu löschen, der größer ist als die Datei selbst, was den Leistungsverlust noch verstärkt. Es gibt verschiedene Techniken zur Milderung dieses Problems, allerdings verringern sie wiederum die Lebensdauer des Flashspeichers.

Die Leistungsprobleme von mechanischen und die Lebensdauerprobleme von mechanischen und Solid-State-Laufwerken lassen sich dadurch verringern, dass beide Arten in Laufwerksarrays kombiniert werden. Durch die Verteilung der Last wird das Ausfallrisiko gesenkt und die Leistung gesteigert.

Einrichten der Speicherebene

Nicht flüchtiger Speicher kann in Form eigenständiger Geräte als lokaler oder direkt angeschlossener Speicher (Direct-Attached Storage, DAS) vorliegen. Speichergeräte können aber auch auf verschiedene Weise kombiniert werden, um für Redundanz oder Konsolidierung zu sorgen, um eine bessere Kostenkontrolle zu haben oder um verschiedene Leistungsstufen anzubieten. Beispielsweise müssen Archivdaten, auf die gewöhnlich nicht so häufig zugegriffen wird, nicht auf dem schnellstmöglichen Laufwerk untergebracht werden.

Lokaler Speicher

Lokaler Speicher, auch Direct-Attached Storage (DAS) genannt, wird direkt an das System angeschlossen, das darauf zugreift. Er kann aus unabhängigen mechanischen Festplatten, Solid-State-Laufwerken, Bandlaufwerken für Sicherungen, CD- und DVD-ROM-Laufwerken und aus Schaltschränken mit Speicherarrays bestehen.

Lokaler Speicher hat eine geringere Latenz als Speichernetzwerke (Storage-Area Networks, SAN) und Netzwerkspeicher (Network-Attached Storage, NAS), da kein Netzwerk zwischen dem System und dem Speicher liegt. (Mehr über diese beiden Arten von Speicher erfahren Sie weiter hinten in diesem Kapitel.) Allerdings kann er nicht gemeinsam mit anderen Systemen genutzt werden, es sei denn, dass das lokale Dateisystem mit einem Protokoll wie SMB 3.0 (Server Message Blocks) im Netzwerk freigegeben ist.

In SQL Server ist lokaler Flashspeicher (Solid State) für `TempDB` zu bevorzugen und wird auch in Failoverclusterinstanzen unterstützt (und empfohlen). Lokalen Speicher können Sie auch für die Pufferpoolerweiterung nutzen.

▶ **Wie Sie `TempDB` am besten einrichten, erfahren Sie im Abschnitt »Konfigurationseinstellungen« von Kapitel 3.**

Speicherarrays und RAID

Werden mehrere Laufwerke zusammen mit einem Controller für den Zugriff auf die einzelnen Geräte in einem Schaltschrank untergebracht, ohne dabei an Redundanz oder Leistung zu denken, so spricht man von *JBOD* (»just a bunch of disks«, also »bloß eine Ansammlung von Laufwerken«). Diese Festplatten könnten auch einzeln verwendet oder zu einem einzigen Volume zusammengefasst werden.

Eine ordnungsgemäße Kombination von Laufwerken in einem Array kann dagegen die Gesamtleistung verbessern und die Gefahr von Datenverlusten beim Ausfall einzelner Laufwerke verringern. Diese Vorgehensweise wird *RAID* genannt (Redundant Array of Independent Disks).

Es gibt verschiedene RAID-Konfigurationen, die als *Ebenen* bezeichnet werden. Bei einigen davon liegt das Gewicht auf Redundanz, bei den anderen auf Leistung. Mehr Redundanz bedeutet zwar eine geringere Gefahr von Datenverlusten, aber dafür steht weniger Kapazität zur Verfügung. Dagegen kann eine höhere Leistung das Risiko von Datenverlusten mit sich bringen.

Bei einem *Stripeset ohne Parität* (RAID 0) werden mehrere Laufwerke verwendet, um die rohe Lese/Schreib-Leistung zu verbessern, allerdings gibt es dabei keine Redundanz. Wenn ein Laufwerk ausfällt, besteht eine erhebliche Wahrscheinlichkeit für einen katastrophalen Datenverlust im gesamten Array. JBOD-Einrichtungen, die mehrere Laufwerke überspannen, werden ebenfalls zu dieser RAID-Ebene gerechnet.

Bei der *Spiegelung* (RAID 1) wird gleichzeitig auf zwei Laufwerke geschrieben. Es gibt zwar eine kleine Leistungseinbuße beim Schreiben, da beide Laufwerke die Daten zur selben Zeit speichern müssen und das eine dafür etwas länger brauchen mag als das andere, aber dafür ist die Leseleistung fast doppelt so hoch wie bei einer einzelnen Festplatte, da der Vorgang parallel auf beiden Laufwerken ausgeführt werden kann (mit einem geringen Overhead durch den RAID-Controller, der das Laufwerk auswählt und die Daten abruft). Der nutzbare Platz beträgt nur 50 % der Rohkapazität. Es darf nur ein Laufwerk im Array ausfallen, um noch alle Daten wiederherstellen zu können.

Für ein *Stripeset mit Parität* (RAID 5) ist eine ungerade Anzahl von Laufwerken erforderlich (mindestens drei). Bei jedem Schreibvorgang wird eines der Laufwerke zufällig für die Speicherung der Parität (eine Art Prüfsumme) ausgewählt. Die Leistungseinbuße beim Schreiben ist hoch, da alle Laufwerke ihre Daten speichern und zusätzlich die Parität berechnet und festgehalten werden muss. Fällt eines der Laufwerke im Array aus, können die anderen die auf ihm abgelegten Inhalte aufgrund der Parität rekonstruieren, allerdings kann es einige Zeit dauern, um das Array wiederherzustellen. Der verfügbare Platz entspricht der Anzahl der Laufwerke minus eins. Bei drei Laufwerken im Array ist so viel Platz zur Verfügung, wie zwei der Laufwerke bieten; der verbliebene Platz wird für die Parität genutzt (aber gleichmäßig im Array verteilt). Es darf nur ein Laufwerk im Array ausfallen, um noch alle Daten wiederherstellen zu können.

Um mehr Redundanz und bessere Leistung zu bieten, können die einzelnen RAID-Ebenen auch kombiniert werden. Daraus ergeben sich z. B. die Ebenen RAID 1+0 (oder RAID 10), RAID 0+1 und RAID 5+0 (RAID 50).

Bei RAID 1+0 werden je zwei Laufwerke in einer Spiegelkonfiguration (RAID 1) verwendet, um für Redundanz zu sorgen, und die einzelnen Spiegel dann zu einem Stripeset (RAID 0) kombiniert.

Umgekehrt findet bei RAID 0+1 erst das Striping statt (RAID 0), woraufhin dann das gesamte Stripeset gespiegelt wird (RAID 1). Sowohl bei RAID 0+1 als auch bei RAID 1+0 beläuft sich der nutzbare Platz auf 50 % der Rohkapazität.

In RAID 1+0 oder 0+1 ist eine vollständige Wiederherstellung möglich, wenn eine gesamte Seite des Spiegels bzw. höchstens ein Laufwerk auf jeder Seite des Spiegels ausfällt.

Bei RAID 5+0 werden mehrere Laufwerke (mindestens drei) als RAID-5-Satz eingerichtet, das dann mit mindestens einem weiteren RAID-5-Satz der gleichen Konfiguration zu einem Stripeset (ohne Parität) kombiniert wird. Der nutzbare Platz beträgt $(x - 1)/y$, wobei x die Anzahl der Laufwerke in jedem der RAID-5-Sätze ist und y die Anzahl der RAID-5-Sätze im Gesamtarray. Bei neun Laufwerken steht also so viel Platz zur Verfügung, wie sechs von ihnen bieten. Damit noch eine vollständige Wiederherstellung möglich ist, darf höchstens ein Laufwerk in jedem RAID-5-Satz ausfallen. Versagt in irgendeinem der Sätze mehr als ein Laufwerk, so geht das gesamte RAID-5+0-Array verloren.

Für SQL Server muss die Speicherebene eine möglichst hohe Leistung zeigen. RAID 1+0 bietet die beste Leistung und Redundanz.

Hinweis

Manche Datenbankadministratoren glauben, dass RAID Datensicherungen überflüssig mache, doch in Wirklichkeit kann RAID nicht zu 100 % gegen Datenverluste schützen. Aufgrund ihrer geringen Kosten und der hohen Kapazität werden häufig digitale Bänder als Sicherungsmedien verwendet, allerdings nutzen mehr und mehr Organisationen Cloudangebote wie Microsoft Azure-Archivspeicher und Amazon Glacier als langfristige, kostengünstige Speicherlösungen für Sicherungen. Führen Sie regelmäßige reguläre Sicherungen Ihrer SQL Server-Installation durch und lagern Sie Kopien davon sicher außerhalb Ihres Standorts.

Zentrale Speicherung in Speichernetzwerken

Ein Speichernetzwerk (Storage-Area Network, SAN) ist ein Netzwerk aus Speicherarrays, die Dutzende, Hunderte oder gar Tausende von Laufwerken (mechanisch oder mit Solid-State-Technologie) an einem zentralen Ort umfassen können, die in einer oder mehreren RAID-Konfigurationen angeordnet sind und Zugriff auf Blockebene bieten. Dadurch wird die Platzverschwendung verringert und die systemübergreifende Verwaltung erleichtert, insbesondere in virtualisierten Umgebungen.

Zugriff auf *Blockebene* bedeutet, dass das Betriebssystem Blöcke beliebiger Größe und Ausrichtung lesen und schreiben kann. Dadurch ist es in der Lage, den vorhandenen Speicherplatz flexibler zu nutzen.

Die Gesamtspeicherkapazität eines SANs kann auf mehrere LUNs (Logical Unit Numbers) aufgeteilt werden, wobei sich jede LUN einem physischen oder virtuellen Server zuteilen

lässt. Diese LUNs können Sie nach Bedarf verlagern und in der Größe ändern, was die Verwaltung viel einfacher macht als beim Anschluss von physischem Speicher an einen Server.

Die Nachteile eines SANs bestehen darin, dass Sie dabei unter Umständen mit Fehlkonfigurationen oder einem langsamen Netzwerk leben müssen. So kann es beispielsweise sein, dass eine RAID-Ebene mit schlechter Schreibleistung eingestellt ist oder dann die Speicherblöcke nicht auf geeignete Weise ausgerichtet sind.

Speicheradministratoren kennen sich unter Umständen nicht mit besonderen Arbeitslasten wie denen von SQL Server aus, weshalb sie ein Leistungsmodell wählen, das zwar für den Rest der Organisation gut ist, um den Verwaltungsaufwand zu verringern, die Ausführung von SQL Server aber beeinträchtigt.

Expertentipp

Fibre Channel und iSCSI im Vergleich

Zur Verbindung zwischen den Systemen und dem Speicher können in Speicherarrys Fibre Channel (FC) und iSCSI (Internet Small Computer Systems Interface) zum Einsatz kommen.

Fibre Channel ermöglicht höhere Datenübertragungsraten als iSCSI, weshalb es sich besser für Systeme eignet, die eine geringe Latenz benötigen, allerdings wird dieser Vorteil durch höhere Kosten für besondere Ausrüstung erkauft.

iSCSI nutzt standardmäßige TCP/IP-Verbindungen und kann daher auf bereits vorhandener Netzwerkausrüstung laufen, was den Einsatz oft billiger macht. Den iSCSI-Durchsatz können Sie dadurch steigern, dass Sie den Speicher in seinem eigenen dedizierten, isolierten Netzwerk unterbringen.

Netzwerkspeicher

Bei Netzwerkspeicher (Network-Attached Storage, NAS) handelt es sich um spezialisierte Hardwaregeräte, die an das Netzwerk angeschlossen werden und gewöhnlich ein Array aus mehreren Laufwerken enthalten. Der Zugriff auf den Speicher erfolgt dabei gewöhnlich auf Dateiebene statt auf Blockebene wie bei einem SAN.

Der NAS-Speicher wird auf dem Gerät selbst konfiguriert. Zur gemeinsamen Nutzung des Speichers im Netzwerk werden Dateifreigabeprotokolle (wie SMB, CIFS [Common Internet File System] und NFS [Network File System]) verwendet.

NAS-Geräte sind relativ weit verbreitet, da sie Zugriff auf gemeinsamen Speicher zu weit geringeren Kosten bieten als ein SAN. Beachten Sie aber die Sicherheit bei der Verwendung von Dateifreigabeprotokollen.

Speicherplätze (Storage Spaces)

Windows Server 2012 und höhere Versionen unterstützen das Feature Speicherplätze (Storage Spaces), mit der lokaler Speicher auf eine flexiblere und besser skalierbare Weise verwaltet werden kann als mit einem RAID.

Statt eines RAID-Sets auf der Speicherebene kann Windows Server ein virtuelles Laufwerk auf Betriebssystemebene anlegen und dabei Kombinationen verschiedener RAID-Ebenen

nutzen. Des Weiteren haben Sie die Möglichkeit, verschiedene physische Laufwerke zu kombinieren, um Bereiche mit unterschiedlicher Leistung zu bilden.

Stellen Sie sich beispielsweise einen Server mit 16 Laufwerken vor, davon acht mechanische und acht SSDs. Mit Speicherplätzen können Sie ein einziges Volume aus allen 16 Laufwerken erstellen und dabei die aktiven Dateien im Solid-State-Bereich unterbringen, was die Leistung drastisch verbessert.

SMB-3.0-Dateifreigaben

SQL Server kann jetzt auch Speicher auf einer Netzwerkfreigabe mit SMB 3.0 oder höher verwenden, da das Protokoll mittlerweile ausreichend schnell und stabil für die Speicheranforderungen des Datenbankmoduls (Leistung und Ausfallsicherheit) geworden ist. Das macht es möglich, eine Failoverclusterinstanz (mehr dazu in dem entsprechenden Abschnitt weiter hinten in diesem Kapitel) ohne gemeinsam genutzten Speicher wie ein SAN zu erstellen.

Die Leistung des Netzwerks ist jedoch von entscheidender Bedeutung, weshalb wir empfehlen, ein dediziertes, isoliertes Netzwerk für die SMB-Freigabe anzulegen und Netzwerkkarten mit RDMA-Unterstützung (Remote Direct Memory Access) zu verwenden. Dadurch kann das Windows Server-Feature SMB Direct eine SMB-Verbindung mit geringer Latenz und hohem Durchsatz aufbauen.

Für kleinere Netzwerke mit begrenzter Speicherkapazität und einem NAS sowie für eine Failoverclusterinstanz ohne gemeinsam genutzten Speicher kann SMB 3.0 eine praktikable Lösung darstellen. Weitere Informationen erhalten Sie in Kapitel 12.

Verbindung mit SQL Server über ein Netzwerk

Wir haben das Netzwerk schon bei der Beschreibung der Speicherebene häufig erwähnt, aber es gibt noch viel mehr dazu. In diesem Abschnitt sehen wir uns an, was beim Zugriff auf das Datenbankmodul über ein Netzwerk geschieht und streifen auch kurz virtuelle LANs (VLANs).

Sofern SQL Server und die Anwendung, die darauf zugreift, kein komplett abgeschlossenes System bilden, erfolgt der Datenbankzugriff über eine oder mehrere Netzwerkschnittstellen. Da böswillige Personen die Netzwerkpakete bei der Übertragung untersuchen und manipulieren können, ist eine Authentifizierung erforderlich, was das System insgesamt komplizierter macht.

Achtung

Sorgen Sie dafür, dass der gesamte TCP/IP-Verkehr zum und vom SQL Server verschlüsselt ist. Für Anwendungen, die sich auf demselben Servercomputer befinden wie die SQL Server-Instanz, ist das nicht erforderlich, sofern Sie das Shared-Memory-Protokoll verwenden.

SQL Server 2017 verlangt strenge Regeln für die Netzwerksicherheit. Dadurch ist es möglich, dass ältere Versionen von Konnektoren und Protokollen nicht mehr wie erwartet funktionieren.

TLS (Transport Layer Security) und der Vorläufer SSL (Secure Sockets Layer) ermöglichen die Verschlüsselung des Netzwerkverkehrs zwischen zwei Punkten. (Mehr darüber erfahren Sie in Kapitel 7.) Nach Möglichkeit sollten Sie neuere Bibliotheken verwenden, die die TLS-Verschlüsselung unterstützen. Falls Sie den Anwendungsdatenverkehr nicht mit TLS verschlüsseln können, sollten Sie IPSec nehmen, das auf der Ebene des Betriebssystems konfiguriert wird.

Protokolle und Ports

Verbindungen zu SQL Server erfolgen über TCP (Transport Control Protocol), wobei als Standardport für die Standardinstanz 1433 verwendet wird. Einiges davon wurde bereits in Kapitel 1 beschrieben, und in Kapitel 7 werden wir uns erneut damit beschäftigen. Benannten Instanzen weist der SQL Server-Konfigurations-Manager zufällig ausgewählte Ports zu, wobei der SQL Server-Browser die Verbindungen zu solchen Instanzen koordiniert. Mit dem Konfigurations-Manager ist es jedoch auch möglich, benannten Instanzen statische TCP-Ports zuzuweisen.

Es ist möglich, den Standardport nach der Installation von SQL Server im Konfigurations-Manager zu ändern. Das bietet zwar keine Sicherheit gegen Portscanner, kann aber aufgrund von Netzwerkrichtlinien erforderlich sein.

Netzwerke bilden auch die Grundlage der Cloud. Abgesehen von der offensichtlichen Tatsache, dass die Azure-Cloud über das Internet zugänglich ist (bei dem es sich um ein Netzwerk aus Netzwerken handelt), handelt es sich bei der gesamten Azure-Infrastruktur, die sowohl der »Infrastruktur als Dienst« (virtuelle Maschinen mit Windows oder Linux, auf denen SQL Server läuft) als auch der »Plattform als Dienst« (Azure SQL-Datenbank) zugrunde liegt, um ein virtuelles Gefüge unzähliger Komponenten, die über Netzwerke verbunden sind.

Virtuelle LANs

Mithilfe eines virtuellen LANs (Virtual Local Area Network, VLAN) können Netzwerkadministratoren Computer auch dann logisch gruppieren, wenn diese nicht physisch an denselben Netzwerkswitch angeschlossen sind. Dadurch können Server ihre Ressourcen in einem physischen LAN gemeinsam nutzen, ohne mit anderen Geräten im selben LAN zu kommunizieren.

VLANs arbeiten auf einer sehr maschinennahen Ebene (OSI-Schicht 2, also der Sicherungsschicht) und werden auf einem Netzwerkswitch eingerichtet. Auf dem Switch kann ein Port für ein einzelnes VLAN festgelegt sein, wobei der gesamte Datenverkehr zu und von diesem Port von dem Switch dem VLAN zugeordnet wird.

Hohe Verfügbarkeit

Mit jeder neuen Version von Windows Server scheinen sich Terminologie und Definitionen zu ändern oder an neu hinzugekommene Features angepasst zu werden. Da SQL Server jetzt auch auf Linux läuft, ist es besonders wichtig, genau zu verstehen, was mit »hoher Verfügbarkeit« gemeint ist.

Grundlegend bedeutet hohe Verfügbarkeit, dass ein Dienstangebot irgendeiner Art (z. B. SQL Server, ein Webserver, eine Anwendung oder eine Dateifreigabe) einen Ausfall irgend-

einer Art übersteht oder zumindest vorhersehbar in einen Standbyzustand übergeht und dass Datenverlust und Ausfallzeit dabei auf ein Minimum reduziert sind.

Alles kann ausfallen. Eine Unterbrechung kann beispielsweise durch das Versagen einer Festplatte hervorgerufen werden, das wiederum auf übermäßige Hitze, übermäßige Kälte, übermäßige Feuchtigkeit oder einen Alarm im Rechenzentrum zurückgehen kann, der so laut ist, dass die dadurch hervorgerufenen Vibrationen die internen Komponenten beschädigen und zu einem Aufsetzen des Kopfes führen.

Seien Sie sich auch all der anderen möglichen Störungen bewusst, die auftreten können. Die folgende Liste ist mit Sicherheit nicht erschöpfend. Es ist sehr wichtig zu verstehen, dass es vergebliche Liebesmüh ist, Annahmen über die Stabilität der Hardware, der Software und des Netzwerks zu machen.

- Ausfall einer Netzwerkkarte
- Ausfall eines RAID-Controllers
- Spannungsspitze oder Spannungsabfall und dadurch Versagen der Stromversorgung
- Beschädigung eines Netzwerkkabels
- Beschädigung eines Stromkabels
- Feuchtigkeit auf dem Motherboard
- Staub auf dem Motherboard
- Überhitzung aufgrund eines ausgefallenen Lüfters
- Fehlerhafte Tastatur, die Tasteneingaben falsch interpretiert
- Ausfall aufgrund von »Bit Rot«
- Ausfall aufgrund eines Bugs in SQL Server
- Ausfall aufgrund von schlecht geschriebenem Code in einem Dateisystemtreiber, der zu einer Beschädigung des Laufwerks führt
- Ausfall von Kondensatoren auf dem Motherboard
- Insekten oder Nagetiere, die an Komponenten nagen und dabei tödlichen Stromschlägen erliegen (das riecht wirklich übel!)
- Ausfälle aufgrund einer Löschanlage mit Wasser statt Gas
- Fehlkonfiguration eines Netzwerkrouters, durch die eine ganze Region unerreichbar wird
- Ausfall aufgrund eines abgelaufenen SSL- oder TLS-Zertifikats
- Ausführung einer `DELETE`- oder `UPDATE`-Anweisung ohne `WHERE`-Klausel (menschliches Versagen)

Die Wichtigkeit der Redundanz

Angesichts der Tatsache, dass alles ausfallen kann, sollte nach Möglichkeit für Redundanz gesorgt werden. Die traurige Realität ist jedoch, dass diese Entscheidung oft durch das Budget beschränkt wird. Das verfügbare Geld ist umgekehrt proportional zum Umfang der

akzeptablen Datenverluste und Ausfallzeiten. In geschäftskritischen Systemen ist die verfügbare Betriebszeit jedoch von höchster Bedeutung. Eine Hochverfügbarkeitslösung ist günstiger, als Ausfallzeiten hinzunehmen, die das Unternehmen in jeder Minute viel Geld kosten.

Eine völlige Freiheit von Ausfallzeiten und Datenverlusten kann praktisch nicht garantiert werden. Es ist immer ein Kompromiss nötig. Wie er aussieht, entscheidet das Unternehmen aufgrund der vorhandenen Ressourcen (Ausrüstung, Mitarbeiter, Geld). Die technische Lösung wird dann aufgrund dieses Kompromisses aufgebaut. Die Strategie wird durch die Zielvorgaben für den Wiederherstellungspunkt und die Wiederherstellungszeit bestimmt, die in der Servicevereinbarung vorgegeben sind.

Zielvorgabe für den Wiederherstellungspunkt

Die Zielvorgabe für den Wiederherstellungspunkt (Recovery Point Objective) bedeutet im Grunde genommen: »Einen wie großen Datenverlust können Sie akzeptieren?« Wie viele Daten zwischen der letzten Sicherung des Transaktionsprotokolls und dem Ausfall gehen verloren? Dieser Wert wird gewöhnlich in Sekunden oder Minuten gemessen. Je kürzer die Vorgabe für den Wiederherstellungspunkt, umso mehr kostet die Hochverfügbarkeitslösung.

Zielvorgabe für die Wiederherstellungszeit

Dieses Ziel (Recovery Time Objective) gibt an, wie viel Zeit zur Verfügung steht, um das System nach einem Ausfall wieder in einen bekannten und nutzbaren Zustand zu versetzen. Dabei können für Hochverfügbarkeitszwecke und für Notfallwiederherstellung verschiedene Vorgaben gemacht werden. Der Wert wird gewöhnlich in Stunden gemessen.

Notfallwiederherstellung

Hochverfügbarkeit und Notfallwiederherstellung sind zwei verschiedene Dinge. Sie werden oft unter derselben Überschrift abgehandelt, da es gemeinsame technische Lösungen für beides gibt. Allerdings geht es bei Hochverfügbarkeit darum, den Dienst am Laufen zu halten, während Notfallwiederherstellung das ist, was geschieht, wenn die Infrastruktur komplett ausfällt. Die Notfallwiederherstellung ist wie eine Versicherung: Man glaubt immer, man würde sie nie benötigen – bis es zu spät ist.

> Hinweis
>
> **»Notfälle« sind jegliche Ausfälle und Ereignisse, die zu einer ungeplanten Dienstunterbrechung führen.**

Cluster

In einem Cluster werden mehrere Computer (Knoten) zu einer Gruppe verbunden. Die Mitglieder dieser Gruppe arbeiten zusammen und erscheinen im Netzwerk als ein einziger Computer.

In den meisten Clustern kann immer nur ein Computer auf einmal aktiv sein. Dazu teilt ein Quorum dem Cluster mit, welcher Knoten der aktive sein soll. Dieses Quorum schreitet auch ein, wenn es zu einem Ausfall der Kommunikation zwischen Knoten kommt.

Jeder Knoten hat eine Stimme in dem Quorum. Damit bei einer geraden Anzahl von Knoten eine einfache Mehrheit möglich ist, muss ein zusätzlicher Zeugencomputer in das Quorum aufgenommen werden.

Expertentipp

Was ist Always On?

Always On ist eine Bezeichnung für eine Reihe von Features. Es handelt sich dabei nicht um den Namen für eine bestimmte Technologie, sondern eher um einen Marketingbegriff. Unter »Always On« fallen tatsächlich zwei verschiedene Technologien, mit denen wir uns weiter hinten in diesem Kapitel noch beschäftigen wollen. Wichtig ist, sich zu merken, dass »Always On« nicht »Verfügbarkeitsgruppen« bedeutet.

Windows Server-Failoverclustering

Microsoft beschreibt dieses Feature wie folgt:

Failovercluster bieten hohe Verfügbarkeit und Skalierbarkeit für viele Serverarbeitsauslastungen. Dazu zählen Serveranwendungen wie Microsoft Exchange Server, Hyper-V, Microsoft SQL Server und Dateiserver. Die Serveranwendungen können auf physischen Servern oder virtuellen Computern ausgeführt werden. In diesem Thema wird das Failoverclusteringfeature beschrieben. [Failovercluster können] auf 64 physische Knoten und 8.000 virtuelle Computer skaliert werden. (https://technet.microsoft.com/library/hh831579(v=ws.11).aspx)

Beachten Sie die Terminologie. Windows Server-Failoverclustering ist der Name der Technologie, die einer Failoverclusterinstanz zugrunde liegt. Darin wiederum sind zwei oder mehr Knoten (Computer) zu einer Ressourcengruppe zusammengeschlossen und erscheinen hinter einem als virtueller Netzwerkname (VNN) bezeichneten Netzwerkendpunkt als ein einziger Computer. Ein SQL Server-Dienst auf einer Failoverclusterinstanz ist clusterfähig.

Linux-Failoverclustering mit Pacemaker

Anstatt das Windows Server-Failoverclustering zu nutzen, kann SQL Server in einem Linux-Cluster beliebige Clusterressourcen-Manager nutzen. Microsoft empfiehlt Pacemaker, da er im Lieferumfang vieler Linux-Distributionen enthalten ist, darunter auch in Red Hat und Ubuntu.

Expertentipp

Node Fencing und STONITH auf Linux

Wenn in einem Cluster etwas schiefgeht und sich ein Knoten nach einem festgelegten Timeout in einem unbekannten Zustand befindet, muss dieser Knoten vom Cluster getrennt und neu gestartet oder zurückgesetzt werden. In Linux-Clustern wird dies als *Node Fencing* bezeichnet und erfolgt nach dem Prinzip STONITH (Shoot The Other Node In The Head). Wenn ein Knoten ausfällt, ist STONITH eine wirkungsvolle, wenngleich drastische Vorgehensweise, um einen fehlgeschlagenen Linux-Knoten zurückzusetzen oder auszuschalten.

Auflösen der Clusterpartitionierung durch ein Quorum

Die meisten Clustertechnologien nutzen ein Quorummodell, um eine Erscheinung zu verhindern, die als *Partitionierung* oder auch *Split Brain* bezeichnet wird. Wenn ein Cluster aus einer geraden Anzahl von Knoten besteht und die Hälfte der Knoten aus der Sichtweise der anderen Hälfte offline geht (und umgekehrt), ergeben sich zwei Hälften, die beide glauben, dass der Cluster immer noch läuft, und beide einen Primärknoten haben.

Je nach Anbindung an die beiden Hälften des Clusters kann es sein, dass eine Anwendung in die eine Hälfte schreibt und eine andere Anwendung in die zweite Hälfte. Die bestmögliche Lösung in einem solchen Fall würde darin bestehen, den Cluster auf den Zustand zu einem Zeitpunkt vor dem Eintreten dieser Spaltung zurückzusetzen, wobei jedoch alle nach dem Ereignis geschriebenen Daten verloren gehen.

Um das zu verhindern, gibt jeder Knoten im Cluster seine Funktionsfähigkeit durch einen regelmäßigen »Herzschlag« zu erkennen. Wenn mehr als die Hälfte der Knoten nicht mehr zeitnah reagieren, wird der Cluster als ausgefallen angesehen. In einem Quorum wird durch eine Abstimmung mit einfacher Mehrheit bestimmt, wie viele Knoten eine »ausreichende Anzahl« ausmachen.

Beim Windows Server-Failoverclustering gibt es vier Arten von Mehrheiten: Knotenmehrheit, Knoten- und Dateifreigabemehrheit, Knoten- und Datenträgermehrheit sowie »nur Datenträger«. Bei den drei letztgenannten Typen wird ein zusätzlicher Zeuge verwendet, der nicht direkt am Cluster teilnimmt. Er erhält ein Stimmrecht, wenn es eine gerade Anzahl von Knoten im Cluster gibt und es daher zu einem Unentschieden bei der Abstimmung kommen könnte.

Failoverclusterinstanzen mit Always On

Eine Failoverclusterinstanz von SQL Server können Sie sich als ein oder zwei Knoten mit gemeinsamem Speicher vorstellen (gewöhnlich einem SAN, da der Zugriff ohnehin höchstwahrscheinlich über das Netzwerk erfolgt).

Auf Windows Server kann SQL Server das Windows Server-Failoverclustering nutzen, um für Hochverfügbarkeit auf der Ebene der Serverinstanzen zu sorgen (also um Ausfallzeiten zu minimieren). Dazu erstellt er eine Failoverclusterinstanz aus zwei oder mehr Knoten. Aus der Sicht des Netzwerks (und damit auch aus der Sicht der Anwendung, der Endbenutzer usw.) erscheint die Failoverclusterinstanz als eine einzige SQL Server-Instanz, die auf einem einzigen Computer läuft. Alle Verbindungen zeigen auf den VNN.

Wenn die Failoverclusterinstanz startet, übernimmt einer der Knoten den Besitz und bringt die SQL Server-Instanz online. Tritt auf dem ersten Knoten ein Fehler auf (oder wird ein Failover geplant, um Wartungsarbeiten auszuführen), gibt es mindestens einige Sekunden Ausfallzeit, in denen der erste Knoten so gut aufräumt, wie er kann, und der zweite Knoten seine SQL Server-Instanz online schaltet. Wenn die Dienste auf dem neuen Knoten laufen, werden die Clientverbindungen dorthin umgeleitet.

Expertentipp

Wie viel Zeit nimmt ein Failover in Anspruch?

Bei einem geplanten Failover müssen jegliche modifizierten Seiten im Pufferpool auf das Laufwerk geschrieben werden. Auf einem Server mit umfangreichem Pufferpool kann die Ausfallzeit daher länger sein als geplant. Mehr über Prüfpunkte erfahren Sie in Kapitel 3 und 4.

Auf Linux ist das Prinzip sehr ähnlich. Ein Clusterressourcen-Manager wie Pacemaker verwaltet den Cluster, und wenn ein Failover auftritt, erfolgt aus der Sicht von SQL Server der gleiche Vorgang: Der erste Knoten wird heruntergefahren und der zweite hochgefahren, um den Platz des ersten als Besitzer einzunehmen. Der Cluster hat auch wie unter Windows eine virtuelle IP-Adresse. Den virtuellen Netzwerknamen müssen Sie manuell zum DNS-Server hinzufügen.

- **Mehr über die Einrichtung eines Linux-Clusters erfahren Sie in Kapitel 11.**

Failoverclusterinstanzen können auch in SQL Server Standard Edition angelegt werden, allerdings sind sie auf zwei Knoten beschränkt.

Protokollversand

Der Transaktionsprotokollversand in SQL Server ist eine äußerst flexible Technologie, die eine relativ kostengünstige und leicht zu verwaltende Lösung für Hochverfügbarkeit und Notfallwiederherstellung bietet.

Der grundlegende Vorgang sieht wie folgt aus: Für die primäre Datenbank wird entweder das vollständige oder das massenprotokollierte Wiederherstellungsmodell verwendet, wobei das Transaktionsprotokoll regelmäßig alle paar Minuten gesichert wird. Die Sicherungsdateien werden auf einen freigegebenen Netzwerkspeicherort übertragen, wo ein oder mehrere sekundäre Server sie zu einer Standby-Datenbank wiederherstellen.

Wenn Sie den Assistenten für den Protokollversand in SQL Server Management Studio verwenden, klicken Sie auf der Registerkarte *Wiederherstellung* auf *Datenbankstatus beim Wiederherstellen von Sicherungen* und wählen Sie dann *Kein Wiederherstellungsmodus* oder *Standbymodus* (*https://docs.microsoft.com/sql/database-engine/log-shipping/configure-log-shipping-sql-server*).

Sollten Sie Ihre eigene Lösung für den Protokollversand erstellen, müssen Sie `RESTORE` mit `NORECOVERY` oder `STANDBY` verwenden.

Bei einem Failover wird das Protokollfragment auf dem primären Server ebenfalls gesichert (falls verfügbar; dies garantiert die völlige Freiheit von Datenverlusten für Transaktionen, die bereits mit Commit bestätigt wurden), an den freigegebenen Speicherort übertragen und nach den letzten regulären Transaktionsprotokollen wiederhergestellt. Die Datenbank wird dann in den Modus `RECOVERY` versetzt (wobei unvollständige Transaktionen zurückgenommen und vollständige Transaktionen ausgeführt werden).

Sobald die Anwendung auf den neuen Server verwiesen wird, ist die Umgebung wieder zurück – entweder ohne jeglichen Datenverlust (falls das Protokollfragment herüberkopiert wurde) oder mit minimalem Datenverlust (wenn nur das letzte übertragene Transaktionsprotokoll wiederhergestellt wurde).

Der Protokollversand funktioniert in allen Editionen von SQL Server, und zwar sowohl auf Windows als auch auf Linux. Da die Express Edition keinen SQL Server-Agent enthält, kann sie jedoch nur als Zeuge fungieren, weshalb Sie den Vorgang durch einen separaten Zeitplanungsmechanismus verwalten müssen. Sie können für sämtliche Editionen von SQL Server auch Ihre eigenen Lösungen erstellen, z. B. mit Azure-Blobspeicher und *AzCopy.exe*.

AlwaysOn-Verfügbarkeitsgruppen

Wie bereits angedeutet, ist es das, was die meisten Personen meinen, wenn sie »Always On« sagen. Die offizielle Bezeichnung dieses Merkmals aber lautet AlwaysOn-Verfügbarkeitsgruppen oder schlicht Verfügbarkeitsgruppen.

Was aber ist eine Verfügbarkeitsgruppe? Früher wurden in SQL Server Datenbankspiegelung und Failoverclustering als zwei getrennte Einrichtungen für Hochverfügbarkeit angeboten. Da die Datenbankspiegelung seit SQL Server 2012 offiziell als veraltet gilt, was zeitlich mit der Einführung von Verfügbarkeitsgruppen zusammenfiel, kann man sich Verfügbarkeitsgruppen als eine Kombination der beiden früheren Merkmale mit Protokollversand als Zugabe vorstellen.

Expertentipp

Worum handelte es sich bei der Datenbankspiegelung?

Die Datenbankspiegelung erfolgte auf Datenbankebene. Dabei wurden zwei Kopien einer Datenbank auf zwei getrennten SQL Server-Instanzen unterhalten und durch die kontinuierliche Übertragung von aktiven Transaktionsprotokolleinträgen synchron gehalten.

Verfügbarkeitsgruppen geben uns die Möglichkeit, eine diskrete Menge von Datenbanken auf einem oder mehreren Knoten eines Clusters hochverfügbar zu halten. Im Gegensatz zu Failoverclusterinstanzen, die eine gesamte Serverinstanz betreffen, funktionieren sie auf Datenbankebene.

Anders als bei Failoverclusterinstanzen wird SQL Server in Verfügbarkeitsgruppen nicht als clusterfähige Version, sondern als eigenständige Instanz installiert.

Verfügbarkeitsgruppen werden in Windows Server durch Windows Server-Failovercluster-ing und in Linux durch einen Clusterressourcen-Manager wie Pacemaker zur Verfügung gestellt. Sie agieren aber *ausschließlich auf Datenbankebene*. Wie in Abb. 2.3 gezeigt, handelt es sich dabei um eine Menge von einer oder mehreren Datenbanken in einer Gruppe (einem *Verfügbarkeitsreplikat*), die (mithilfe des Protokollversands) *repliziert* werden. Vom Original, dem *primären Replikat*, können bis zu acht *sekundäre Replikate* erstellt werden. Dazu wird die synchrone oder asynchrone *Datensynchronisierung* angewendet, die wir uns etwas genauer ansehen wollen:

- **Synchrone Datensynchronisierung** Das Protokoll wird in allen sekundären Replikaten festgeschrieben (die Transaktionen werden mit Commit in das Transaktionsprotokoll übernommen), *bevor* die Transaktion im primären Replikat mit Commit bestätigt wird. Dadurch ist ein Datenverlust ausgeschlossen, allerdings kann die Leistung erheblich leiden. Es kann sehr kostspielig sein, die Netzwerklatenz so weit zu reduzieren, dass diese Vorgehensweise auch bei hohen Transaktionslasten funktioniert.
- **Asynchrone Datensynchronisierung** Die Transaktion wird als bestätigt angesehen, sobald sie im Transaktionsprotokoll des primären Replikats festgeschrieben ist. Tritt irgendeine Störung auf, bevor die Protokolle in allen sekundären Replikaten festgeschrieben sind, so besteht die Gefahr eines Datenverlusts. Der Wiederherstellungspunkt in diesem Fall ist die jüngste mit Commit bestätigte Transaktion, die es erfolgreich in alle sekundären Replikate geschafft hat. Die Funktion der »zeitverzögerten Dauerhaftigkeit« kann die Leistung steigern, allerdings besteht dann ein höheres Risiko von Datenverlusten.

Expertentipp

Was ist zeitverzögerte Dauerhaftigkeit?

Beginnend mit SQL Server 2014 gibt es das Speicheroptimierungsfeature der zeitverzögerten Dauerhaftigkeit (auch als »Lazy Commit« bezeichnet), das einen erfolgreichen Commit meldet, bevor die Transaktionsprotokolle tatsächlich auf einem Laufwerk gespeichert werden. Das kann zwar die Leistung steigern, erhöht aber auch das Risiko von Datenverlusten, da die Transaktionsprotokolle erst bei der asynchronen Übertragung auf das Laufwerk gespeichert werden. Mehr darüber erfahren Sie auf *https://docs.microsoft.com/sql/relational-databases/logs/control-transaction-durability*.

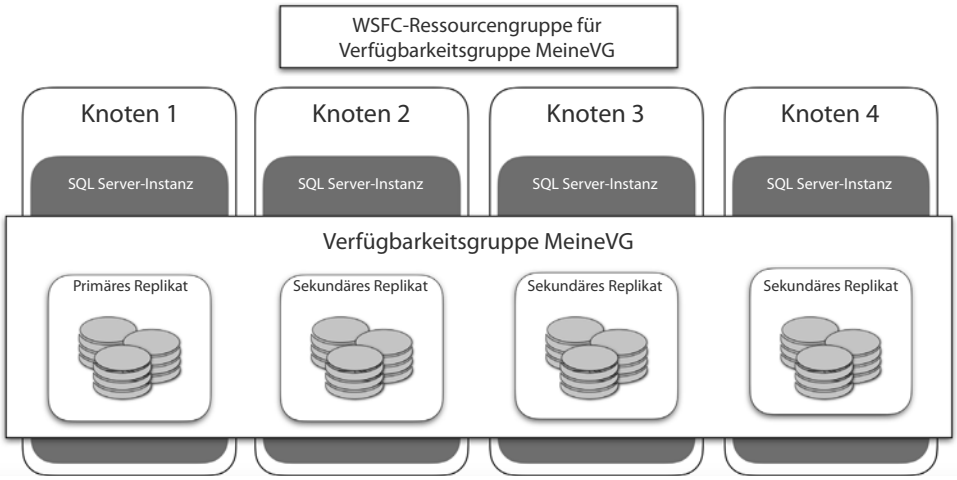

Abbildung 2.3 Ein Windows Server-Failovercluster mit vier Knoten