

Leseprobe

Ulrich B. Boddenberg

SQL Server 2014 für Professionals

Hochverfügbarkeit, Cloud-Szenarien, Backup/Restore, Monitoring &
Performance, Dimensionierung

ISBN (Buch): 978-3-446-44262-7

ISBN (E-Book): 978-3-446-44444-7

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-44262-7>

sowie im Buchhandel.

Inhalt

Vorwort	XIII
1 SQL Server im Business	1
1.1 Kosten	2
1.1.1 Anforderungen berücksichtigen – nicht mehr	2
1.1.2 Konsolidieren der SQL Server-Landschaft	4
1.2 Integration von Cloud-Ressourcen	5
1.3 Der SQL Server jenseits des Datenbankmoduls	5
2 Erweiterte Grundlagen	7
2.1 Instanzen	7
2.1.1 Installieren einer Instanz	8
2.1.2 „Inspektion“	19
2.2 Identitäten	22
2.2.1 Windows-Benutzer	23
2.2.2 SQL-Benutzer	28
2.3 Schema	31
2.4 Kerberos	36
2.4.1 SPN manuell registrieren	38
2.4.2 Automatische Registrierung des SPN ermöglichen	38
2.4.3 Service Principal Names der anderen SQL Server-Komponenten	41
2.5 Speicheroptimierte Tabellen	42
2.6 Datenbanken auf SQL Server 2014 bringen	43
2.6.1 Inplace-Upgrade des kompletten Servers	46
2.6.2 Einzelne Datenbanken	50
2.6.2.1 Sichern/Wiederherstellen	51
2.6.2.2 Oder: Trennen/Offline und anfügen	55
2.6.2.3 Nacharbeiten	57

3	Hardware und Lizenzen	59
3.1	Die optimale Umgebung für SQL Server	59
3.1.1	Elemente des SQL Server-Servers	59
3.1.2	Prozessoren	60
3.1.2.1	Bedarf an Prozessoren ermitteln	61
3.1.2.2	Geeignete Prozessoren/kleine Prozessorkunde	62
3.1.2.3	Hyperthreading – ja oder nein	70
3.1.2.4	MAXDOP – Maximum Degree of Parallelism	71
3.1.2.5	Anzahl der Kerne pro Edition	74
3.1.2.6	... und in virtualisierten Umgebungen?	75
3.1.2.7	Performance messen und überwachen	76
3.1.3	Hauptspeicher	76
3.1.4	Festplatten-System	81
3.1.4.1	Dateien, Protokolle, Seiten & Co.	81
3.1.4.2	Platten und I/Os	91
3.1.4.3	Planung und Einrichtung konkret	105
3.1.4.4	Überwachen und messen	109
3.1.5	Netzwerk	109
3.2	Lizenzierung	109
3.2.1	Lizenzmodelle und Richtpreise	110
3.2.2	Virtualisierte Umgebungen	113
3.2.2.1	Lizenzierung individueller virtueller Maschinen	113
3.2.2.2	High Density Virtualization	114
3.2.3	Lizenzierung für hochverfügbare Umgebungen	114
3.2.4	Datenquellen bei der Business Intelligence Edition	114
4	Verfügbarkeit	117
4.1	Virtualisierung	122
4.1.1	„Klassisches Modell“	122
4.1.2	Hyper V-Replikation	124
4.1.2.1	Hyper V-Replikation vorbereiten	125
4.1.2.2	Hyper V-Replikation für eine virtuelle Maschine einrichten	127
4.2	AlwaysOn-Verfügbarkeitsgruppen	134
4.2.1	Funktionsweise	134
4.2.2	Vorbereitung: Failovercluster-Feature installieren	136
4.2.3	Cluster einrichten	137
4.2.3.1	Der Konfigurationsüberprüfungs-Assistent	138
4.2.3.2	Der Clustererstellungs-Assistent	143
4.2.3.3	Zeugenserver konfigurieren	146
4.2.4	SQL Server installieren	152
4.2.4.1	Basisinstallation	152
4.2.4.2	AlwaysOn vorbereiten	152
4.2.4.3	AlwaysOn konfigurieren	154
4.2.5	Weitere Datenbank(en) hinzufügen	167

4.2.6	Zugriff auf AlwaysOn-Verfügbarkeitsgruppe	171
4.2.7	Kontrollieren	173
4.2.8	Failover	176
4.2.8.1	Geplantes Failover	176
4.2.8.2	Failover nach Absturz	177
4.3	Failoverclustering	181
4.3.1	Funktionsweise	181
4.3.2	iSCSI einrichten	183
4.3.2.1	Initiator einrichten	183
4.3.2.2	iSCSI-Target einrichten	185
4.3.2.3	iSCSI-Initiator mit Target verbinden	191
4.3.3	Failoverclustering (Windows) einrichten	194
4.3.3.1	Feature installieren	195
4.3.3.2	Cluster prüfen	195
4.3.3.3	Cluster erstellen	196
4.3.4	MSDTC installieren (optional)	197
4.3.5	SQL-Cluster installieren	200
4.3.5.1	Erster Knoten	201
4.3.5.2	Weitere Knoten	211
4.3.6	Zugriff auf die geclusterte SQL Server-Instanz	216
4.3.7	Weitere Instanzen installieren	216
4.4	Transaktionsprotokollversand	217
4.4.1	Funktionsweise	217
4.4.2	Einrichtung	219
4.4.3	Betrieb und Überwachung	226
4.4.4	Failover	229
4.5	Datenbankspiegelung	230
5	Backup und Restore	233
5.1	Einige Gedanken und Fakten vorab	233
5.1.1	Servicelevel	233
5.1.2	Wiederherstellungszeit	235
5.1.3	Datenverlustzeit	237
5.1.4	Logische Fehler	238
5.1.5	Katastrophenvorsorge	238
5.1.6	Genügt die Datensicherung (oder: Business Continuity vs. Disaster Recovery)?	239
5.2	Datensicherung mit Bordmitteln	239
5.2.1	Sicherungsstrategie	239
5.2.2	Sicherung durchführen/einfach	244
5.2.3	Sicherung mit Wartungsplan	248
5.2.4	In die Cloud sichern	264
5.2.5	Wiederherstellung	270
5.2.5.1	Rücksicherung der Vollsicherung	270

5.2.5.2	Rücksicherung mit (mehreren) inkrementellen und Transaktionsprotokollsicherungen	275
5.2.5.3	Das Protokollfragment	279
5.2.5.4	Wiederherstellen der Master-Datenbank	283
5.3	Microsoft Data Protection Manager 2012 R2	286
5.3.1	Hardware-Voraussetzungen	287
5.3.1.1	Festplattenbereich	287
5.3.1.2	Bandgerät	288
5.3.2	Installation	290
5.3.2.1	Voraussetzungen	291
5.3.2.2	DPM	294
5.3.3	Basiskonfiguration	300
5.3.3.1	Plattenspeicher	301
5.3.3.2	Bandroboter (Bibliothek)	302
5.3.3.3	Agenten	306
5.3.3.4	Backup auf Azure	309
5.3.4	Schutzgruppen einrichten	327
5.3.5	Überwachung	336
5.3.6	Berichte/Bandwechsel	338
6	SQL Server und die Cloud – die Cloud und SQL Server	341
6.1	Azure SQL-Datenbank	342
6.1.1	Abgrenzung zu SQL Server auf Azure-VM	343
6.1.2	Einschränkungen der Azure SQL-Datenbanken	346
6.1.3	Azure SQL-Datenbank anlegen	349
6.1.4	Administration des Datenbankservers	354
6.1.5	Datenbank in Azure SQL bereitstellen	360
6.1.6	Nutzung der Azure SQL-Datenbanken durch Azure-Dienste	364
6.2	Virtuelle Maschinen in Azure mit SQL Server	366
6.2.1	Netzwerk erstellen und Site-to-Site-VPN einrichten	367
6.2.2	Virtuelle Maschine im eigenen Azure-Netz erstellen	377
6.2.3	Virtuellen SQL Server erstellen	380
6.2.3.1	Erstellen des virtuellen Servers	381
6.2.3.2	Administrativen Zugriff ermöglichen	387
6.2.3.3	Windows-Firewall und Cloud-Adapter	390
6.2.3.4	Datenbank in Azure SQL-VM bereitstellen	394
6.3	Datenbanken mit Azure-Storage	402
6.3.1	Anlegen des Speichers in Azure	404
6.3.2	Datenbank anlegen	411
6.3.3	Monitoring der Performance	414
6.4	Backup in die Cloud	415
6.5	SQL Server auf Azure-VMs als Notfallrechenzentrum	416
6.5.1	AlwaysOn-Verfügbarkeitsgruppen	417
6.5.2	Transaktionsprotokollversand	420
6.5.3	Datenbankspiegelung	421

7	Überwachung und Monitoring	423
7.1	SQL Server-Protokolle	424
7.2	Datenbank-E-Mail	426
7.2.1	Basiseinrichtung	427
7.2.2	Datenbank-E-Mail verwenden	431
7.2.2.1	Operatoren anlegen	431
7.2.2.2	SQL Server-Agent vorbereiten	432
7.2.2.3	Benachrichtigung und Warnungen aktivieren	433
7.3	Datensammlung	440
7.3.1	Datensammlung einrichten	441
7.3.1.1	Verwaltungs-Data Warehouse konfigurieren	441
7.3.2	Daten abrufen	446
7.3.2.1	Datenträgerverwendung	446
7.3.2.2	Serveraktivität	447
7.3.2.3	Abfragestatistik	449
7.4	Dynamic Management Views, DMVs	451
7.5	Erweiterte Ereignisse/Extended Events	454
7.5.1	Einrichten und konfigurieren	455
7.5.2	Live-Ansicht und Datenauswertung	464
7.6	Audit	469
7.6.1	Überwachung einrichten	469
7.6.2	Protokoll anzeigen	473
7.7	Ressourcenkontrolle – Resource Governor	475
7.7.1	Einrichten	476
7.7.2	Überwachung	481
7.7.2.1	Performance-Monitor	482
7.7.2.2	DMVs	483
7.8	Performance-Monitor und SQL Server Profiler	485
7.9	Weitere Überwachungswerkzeuge – SCOM	485
8	Troubleshooting	489
8.1	Einige Basisaspekte	489
8.1.1	Ereignisanzeige	489
8.1.2	Hardware/VM-Konfiguration	490
8.1.3	Installation von Patches	490
8.2	Logfiles werden unendlich groß	491
8.3	„Hilfe, ich muss Datenbankdateien verschieben“	494
8.3.1	Dateien verschieben mit möglichst viel Komfort	494
8.3.2	Mit SQL-Befehlen	498
8.3.3	Sonderfall: „Mir ist ein Logfile versehentlich so groß geworden, dass ich es nirgendwo mehr hinschieben kann“	499
8.3.4	Systemdatenbanken verschieben	501
8.4	Datenbank zwischen SQL Servern kopieren/verschieben	501

8.4.1	Variante 1: Mit grafischer Unterstützung durch Assistenten	501
8.4.1.1	Auftrag vorbereiten	502
8.4.1.2	Fehler suchen und beheben	506
8.4.1.3	Methode „Trennen/Anfügen“	509
8.4.2	Ohne Assistenten die Aufgabe erledigen	510
8.5	Das wirklich ernsthafte Problem mit den SQL-Anmeldungen	511
8.6	SQL Server umbenennen	516
8.7	„Hilfe, ich kann keine Verbindung zum SQL Server aufbauen“	516
8.7.1	Kontrolle der Dienste im Konfigurations-Manager	516
8.7.2	Netzwerkprotokolle prüfen	517
8.7.3	Windows-Firewall konfigurieren	519
8.7.3.1	Konfiguration der Windows-Firewall für benannte Instanzen	521
8.7.3.2	Windows-Firewall für SQL Server-Browser-Dienst anpassen	525
8.8	Performance messen und bewerten	526
8.8.1	Performance-Monitor „live“ bedienen	527
8.8.2	Aufzeichnen von Daten	528
8.8.3	Einige Werte	534
8.8.3.1	Memory	535
8.8.3.2	Disk	536
8.8.3.3	Prozessor	537
8.8.3.4	Allgemein	537
8.9	Abfragen analysieren	539
8.9.1	SQL Profiler	539
8.9.1.1	Der Testfall	539
8.9.1.2	Kleiner Exkurs: Datenbankzugriff mit ORM-Werkzeugen	540
8.9.1.3	Ein wenig Analyse	544
8.9.2	Ausführungsplan	548
8.9.2.1	Anzeigen des Ausführungsplans	548
8.9.2.2	UNION vs. UNION ALL	550
9	Replikation	553
9.1	Technische Vorbereitung	555
9.2	Einige Grundbegriffe	557
9.3	Replikationstypen	558
9.4	Momentaufnahmeveröffentlichung/Snapshot-Replikation	559
9.5	Transaktionsreplikation	560
9.6	Merge-Replikation	561
9.7	Zusammenfassung	561
9.8	Editionsvergleich	562
9.9	Snapshot-Replikation	563
9.9.1	Funktionsweise	563

9.9.2	Veröffentlichung einrichten	564
9.9.2.1	Vorbereitungen durchführen	564
9.9.2.2	Veröffentlichung einrichten	565
9.9.3	Verleger- und Verteilereigenschaften	570
9.10	Abonnements hinzufügen	572
9.10.1	Push-Abonnenten hinzufügen	573
9.10.1.1	Vorbereitung	573
9.10.1.2	Anlegen des Abonnements	574
9.10.2	Pull-Abonnenten hinzufügen	581
9.10.3	Häufige Fehler	583
9.11	Transaktionsreplikation	584
9.11.1	Funktionsweise	585
9.11.2	Einrichtung und Test	587
9.11.2.1	Vorbereitungen	587
9.11.2.2	Veröffentlichung einrichten	587
9.11.2.3	Abonnement anlegen	592
9.11.3	Fazit	597
9.12	Peer-zu-Peer-Veröffentlichung	598
9.13	Merge-Replikation	598
9.13.1	Veröffentlichung vorbereiten	599
9.13.2	Abonnement einrichten	606
9.13.3	Replikation initiieren	609
9.13.4	Konfliktbehandlung	610
9.13.5	Synchronisation über das Web	612
Index	613

Vorwort

Liebe Leserin, lieber Leser,

ich freue mich, dass Sie mein Buch zu SQL Server in der Hand halten. Bevor man ein Buch schreibt, hat man eine gewisse Vision, wozu das Buch dienen soll. Man denkt als Autor darüber nach, welche Schwerpunkte für die Zielgruppe spannend sind, was unbedingt im Buch stehen muss und was vielleicht nicht so wichtig ist.

Ich bin jemand, der als Berater viele Szenarien in Unternehmen, Behörden und Organisationen sieht, also mit den Produkten und Technologien im „wirklichen Leben“ umgeht. Dieses Buch ist sozusagen die Zusammenfassung meiner Praxiserfahrung mit dem SQL Server. Plant man ein Buch über eine so vielschichtige Technologie wie SQL Server, muss man notwendigerweise Schwerpunkte setzen, weil die Menge an Seiten, die die Druckerei zwischen zwei Buchdeckel bekommt, begrenzt ist (und außerdem soll das Buch irgendwann fertig werden). Die Auswahl der Themen folgt diesen Prämissen:

- Die Zielgruppe dieses Buchs sind IT-Architekten und Administratoren. Auch wenn Entwicklerthemen häufig sehr spannend sind, spielen sie insgesamt in diesem Buch nur eine relativ geringe Rolle.
- Ich habe überlegt, auf welche Fragestellungen ich bei meinen Kunden gestoßen bin. Dieses dürften die Fragestellungen sein, die auch im wirklichen Leben der Leser dieses Buchs eine Rolle spielen.
- Das Produkt SQL Server enthält bekanntlich neben dem Datenbankmodul drei weitere spannende Komponenten, nämlich die Analysis Services, die Reporting Services und Integration Services. Dieses Buch handelt vom Datenbankmodul – und zwar nicht, weil mich die anderen Themen nicht interessieren würden, sondern weil diese drei Themen im Grunde genommen ein eigenes Buch mit etwas anderem Schwerpunkt verdienen würden (schreiben Sie doch mal dem Verlag, wenn Sie ein Titel „SQL Server BI-Komponenten für Administratoren und Architekten“ interessieren würde).
- Ich habe mich stets bemüht, möglichst viel Hintergrundwissen zu den einzelnen Themen zu vermitteln. Um herauszufinden, wie man das SQL Server Management Studio startet oder das Windows-Ereignisprotokoll öffnet, brauchen Sie mit Sicherheit dieses Buch nicht. Das Wertvollste, das ich Ihnen liefern kann, sind Praxiserfahrung, konzeptionelle Vorgehensweisen und Hintergrundwissen.

Ich hoffe, dass Sie mit der Zusammenstellung der Themen zufrieden sind und Sie dieses Buch in Ihrer täglichen Arbeit mit dem SQL Server weiterbringt. Es ist auf Basis der derzeit aktuellen Version SQL Server 2014 erstellt worden, die meisten Aspekte passen aber sogar noch auf SQL Server 2005: Im Datenbankmodul gibt es zwar auch viele Innovationen, aber nur relativ wenige grundlegende Änderungen, die das Produkt komplett verändert hätten.



HINWEIS: Noch ein Hinweis in eigener Sache: Ich bekomme häufig Anfragen von Lesern, ob ich deren Unternehmen und Organisationen auch vor Ort beraten würde, ein Projekt begleite oder eine Individualschulung oder Coaching durchführe.

Die Antwort lautet ganz klar: Ja. Die Beratung und Begleitung von Unternehmen und Organisationen ist mein Kerngeschäft und ich bin ein „Mann der Praxis“.

Insofern sind Sie herzlich eingeladen, mit mir Kontakt aufzunehmen. Sie können mir per E-Mail schreiben (Ulrich@Boddenberg.de) oder meine Website besuchen (<https://www.boddenberg.de>).

Bleibt mir noch, Ihnen nun viel Spaß mit diesem Buch und viel Erfolg beim Einsatz von SQL Server zu wünschen.

Ich möchte mich bei meiner Frau Ilona und unserer Amy für viel liebevolle mentale Unterstützung bei der Erstellung dieses Buchs bedanken!

Ulrich B. Boddenberg, Januar 2015

mal die Daten auf Platte und somit muss dieser Bereich des Servers besonders berücksichtigt werden. Solange Sie nicht SSD-Platten für die Datenbanken verwenden, sprechen wir über „Mechanik“, allein schon deshalb ist es kritisch.

- *Hauptspeicher:* SQL Server macht weiterhin umfangreichen Gebrauch vom Hauptspeicher, insofern muss dieser natürlich auch richtig dimensioniert sein. Hier hängt es beispielsweise auch von der Anzahl der Instanzen ab, wie viel Speicher benötigt wird. Es gilt natürlich die bewährte IT-Weisheit, dass man nie genug Speicher haben kann – es gibt aber irgendwo eine Grenze, ab der zusätzliches RAM richtig viel Geld kostet und verhältnismäßig wenig Performancezuwachs bringt.
- *Netzwerk:* Die modernen Netzwerke sind so breitbandig, dass von dieser Seite vermutlich die geringsten Probleme entstehen, nichtsdestotrotz muss das Thema „Netzwerk“ natürlich berücksichtigt werden.

Neben den Teilkomponenten muss natürlich auch das große Ganze gesehen werden:

- Wie sieht es mit der Virtualisierung von SQL Server aus? Gemeinhin virtualisiert man ja heute fast alles, bei Datenbankservern schrecken die meisten Administratoren allerdings nach wie vor zurück.
- Wenn Sie SQL Server direkt auf Hardware betreiben, stellt sich die Frage, welche Anforderungen an die Server-Architektur bestellt werden müssen.
- ... und unabhängig von der Hardware (bzw. emulierten Hardware) gibt es einige Konfigurationseinstellungen, die sich durchaus lohnen.

3.1.2 Prozessoren

Wenn Sie sich einmal gedanklich vorstellen, wie der SQL Server arbeitet, kommen Sie etwa zu diesem Ergebnis – grob vereinfacht:

- 300 Nutzer haben eine Anwendungssoftware, die mit SQL Server arbeitet. Bei der Arbeit mit der Software werden ständig Datenbankabfragen erzeugt, die entweder ganz simpel Daten aus der Datenbank abrufen und Datensätze verändern, erzeugen oder löschen. Natürlich gibt es auch umfangreiche Transaktionen, in deren Verlauf beispielsweise Sperren gesetzt werden, das spielt für diese Betrachtung aber zunächst keine Rolle.
- Da all diese Benutzer nichts anderes tun, als Aufträge zu erfassen, kann man davon ausgehen, dass pro Sekunde 50 bis 100 Abfragen an die Datenbank gerichtet werden. Diese enorme Zahl kommt dadurch zustande, dass viele ERP-Systeme ständig Daten nachladen, um beispielsweise Auswahlboxen zu füllen, Plausibilitätsprüfung durchzuführen und Diverses andere mehr, wie natürlich auch Daten zu speichern, Warenbestände zu aktualisieren usw.
- Wenn Sie einen Prozessor für diese Datenbankinstanz hätten, müsste der alle 100 Abfragen bedienen – da jede Sekunde 100 Abfragen kommen, hat er schon viel zu tun, weil er vermutlich gar nicht so schnell arbeiten kann. Die Geschwindigkeit, mit der der Prozessor die Abfragen bedienen kann, hängt natürlich nicht allein von der möglichen Prozessorleistung ab, sondern auch von anderen Komponenten des Servers wie beispielsweise dem Festplattensystem. Liefern die Platten zu langsam Daten, kann der Prozessor die Abfragen notwendigerweise nicht schneller bearbeiten.

- Wenn man von einem idealen System ausgeht, bei dem die Daten von den Platten ohne Verzögerung kommen (was natürlich nicht wirklich realistisch ist), müssen wir also lediglich für mehr Prozessoren sorgen, wenn ein Prozessor die Arbeit nicht allein erledigen kann, weil zu viele Datenbankabfragen pro Sekunde durchgeführt werden müssen. In einer idealen Welt ist die Rechnung einfach: Mit vier Prozessoren haben Sie viermal mehr Rechenkapazität, sodass jeder Prozessor nur 25 Datenbankabfragen pro Sekunde bedienen muss. Die Welt ist zwar nun nicht optimal, sodass viermal mehr Prozessoren nicht automatisch zu einer vierfachen Leistung führen, trotzdem kann man festhalten, dass SQL Server in Hinblick auf die Prozessoren recht gut skaliert, da die Anfragen an die Datenbank gleichmäßig über die Prozessoren verteilt werden können.

Aus diesem Grund eignen sich SQL Server sehr gut für die Verwendung in Multiprozessor-Systemen oder anders gesagt: Systeme mit vielen Prozessoren bzw. vielen Kernen kommen der Arbeitsweise von SQL Server sehr entgegen.

3.1.2.1 Bedarf an Prozessoren ermitteln

Die spannende Frage wird nun sein, wie viele Prozessoren Sie für Ihr konkretes System benötigen. Diese Frage werde ich leider nicht pauschal beantworten können. Auch eine Faustformel im Stil „pro n Benutzer ein Prozessor“ ist in jedem Fall unseriös, denn zu viel hängt von der Komplexität und Qualität der Abfragen ab, von der Anzahl der Abfragen pro Benutzer und Zeiteinheit und diversen anderem mehr. Um zu einer einigermaßen verlässlichen Schätzung des Prozessorbedarfs zukommen, sehe ich zwei grundsätzliche Wege:

- Vermutlich benötigen Sie den SQL Server, weil Sie eine Anwendung einsetzen möchten, die eben einen Datenbankserver benötigt. Die Menschen, die diese Anwendung entwickelt haben, sollten anhand ihres konkreten Nutzungsszenarios und der Anzahl der Benutzer eine einigermaßen verbindliche Aussage treffen können, wie viele Prozessoren benötigt werden. Falls Sie selber ein Softwarehersteller sind, sollten Sie aus Erfahrung wissen, wie viele Prozessoren bei welchem Anwendungsszenario und welcher Benutzeranzahl benötigt werden. Im Zweifelsfall bietet es sich an, einen Lastsimulator zu programmieren, der unterschiedliche Szenarien und Benutzerzahlen durchsetzen kann.
- Wenn Sie ein bestehendes SQL Server-System haben, können Sie an diesem Messungen durchführen und den Bedarf des neuen Systems daraus ableiten. Ich arbeite regelmäßig bei Konsolidierungsprojekten mit, in denen aus der gewachsenen Landschaft mit vielen Dutzend SQL Servern möglichst wenige gut verwaltete und leistungsfähige Systeme gemacht werden sollen. Es ist nun allerdings keine triviale Aufgabe, anhand der Leistungsdaten von 30 Servern ein neues konsolidiertes Szenario zu planen.



HINWEIS: Wenn Sie denken, dass Sie ein leistungsfähiges Hardware-System mit vielen Prozessoren und jeweils vielen Kernen beschaffen und sich dann auf der sicheren Seite fühlen, ist das aus technischer Sicht durchaus nachvollziehbar. Etwas mehr Geld für Hardware zu investieren, kann durchaus eine gute Idee sein, um spätere Probleme auszuschließen oder zumindest zu minimieren.

Sie werden aber relativ schnell feststellen, dass dieser Plan (also Hardware zu überdimensionieren, um spätere Probleme auszuschließen) eventuell aus Gründen des Budgets scheitert. Die Lizenzierung von SQL Server kann mit Prozessorlizenzen oder CALs erfolgen, wobei die Enterprise Edition nur prozessor-

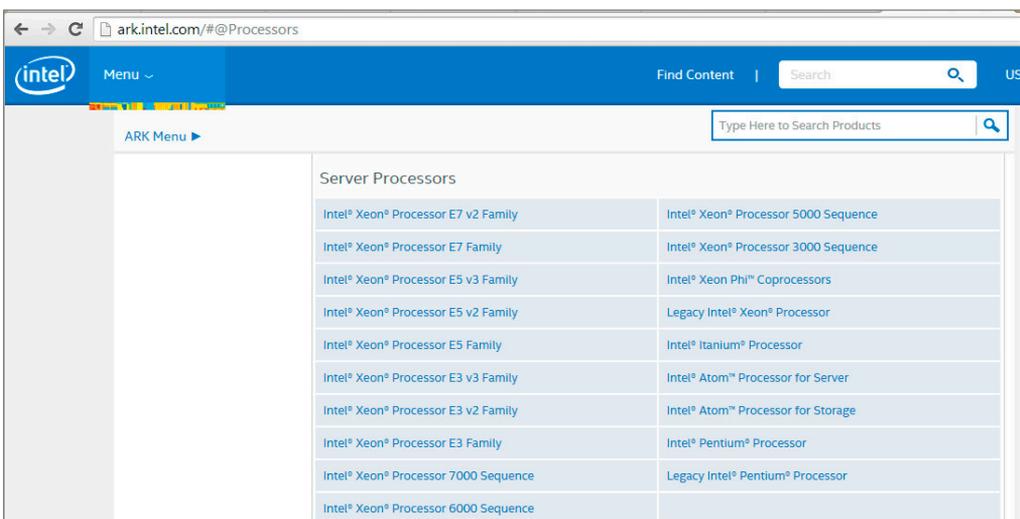
basiert lizenziert werden kann. Bei der prozessorbasierten Lizenzierung kommt es seit SQL Server 2012 auf die Anzahl der Kerne an. Mit anderen Worten könnte es sein, dass eine recht großzügig ausgelegte Hardware immense Lizenzkosten verursacht. Insofern führt eine geplante Überdimensionierung unter Umständen nicht nur zu erhöhten Hardware-Kosten, sondern vor allem zu einem massiven Anstieg der zu kaufenden Lizenzen.

3.1.2.2 Geeignete Prozessoren/kleine Prozessorkunde

Intel kennt vier Kategorien von Prozessoren:

- Desktop-Prozessoren
- Prozessoren für mobile Systeme
- Prozessoren für Embedded-Geräte
- Prozessoren für Server

Es versteht sich von selbst, dass für einen SQL Server auch entsprechende Server-Hardware zum Einsatz kommen sollte – bzw. muss! Ich erwähne das deshalb, weil die Serverhersteller im Einstiegsbereich gern Systeme als Server deklarieren, die eigentlich etwas aufgebohrte Arbeitsplatz-PCs sind. Diese kleinen Server werden mit Desktop-Prozessoren betrieben und bewegen sich demzufolge auch von der Leistung auf Desktop-Niveau. Zur „echten“ Server-Prozessoren gehören auch entsprechende Chipsätze, Speicher und diverse andere Komponenten, die eben auf den Serverbetrieb optimiert sind. Auch wenn Sie nicht der absolute Hardware-Experte sind, können Sie anhand des verwendeten Prozessors sehr einfach erkennen, aus welcher Leistungsklasse das System stammt. Intel bietet unter der URL <http://ark.intel.com/#@Processors> eine recht übersichtliche Zusammenfassung der derzeit verfügbaren und historischen Prozessorfamilien an. Bild 3.1 zeigt die Übersicht der Server-Prozessoren, auf der man erkennt, dass es eine ganze Menge an Prozessorfamilien, beginnend mit dem Intel Pentium, gibt.



The screenshot shows the Intel ARK website interface. The browser address bar displays 'ark.intel.com/#@Processors'. The page features the Intel logo, a navigation menu, and a search bar. Below the navigation, there is a section titled 'Server Processors' which contains a table listing various processor families and models.

Server Processors	
Intel® Xeon® Processor E7 v2 Family	Intel® Xeon® Processor 5000 Sequence
Intel® Xeon® Processor E7 Family	Intel® Xeon® Processor 3000 Sequence
Intel® Xeon® Processor E5 v3 Family	Intel® Xeon Phi™ Coprocessors
Intel® Xeon® Processor E5 v2 Family	Legacy Intel® Xeon® Processor
Intel® Xeon® Processor E5 Family	Intel® Itanium® Processor
Intel® Xeon® Processor E3 v3 Family	Intel® Atom™ Processor for Server
Intel® Xeon® Processor E3 v2 Family	Intel® Atom™ Processor for Storage
Intel® Xeon® Processor E3 Family	Intel® Pentium® Processor
Intel® Xeon® Processor 7000 Sequence	Legacy Intel® Pentium® Processor
Intel® Xeon® Processor 6000 Sequence	

Bild 3.1 Auf der Intel-Website gibt es eine Übersicht mit allen Prozessoren nebst umfangreichen technischen Daten.

3.1.2.2.1 Prozessoren für kleine Server (ein Socket)

Bild 3.2 zeigt einen kleinen Überblick über die Xeon E3-Familie. Hierbei handelt es sich um Server, die für den Einsatz in kleinen Servern mit nur einem Prozessor (anders gesagt: für einen Socket) gedacht sind. Sie sehen in der Auflistung, dass es sich hierbei überwiegend um Prozessoren mit vier Kernen handelt. Wie ich weiter vorn bereits ausgeführt habe, profitiert der SQL Server sehr von Mehrkern-Architekturen, allerdings müssen Sie bedenken, dass Sie unter Umständen (also je nach Lizenzierungsmodell) diese ganzen Kerne auch lizenzieren müssen.

Compare	Product Name	Status	Launch Date	# of Cores	TDP	Recommended Customer Price	Processor Graphics †
Compare +	Intel® Xeon® Processor E3-1220 v3 (8M Cache, 3.10 GHz)	Launched	Q2'13	4	80 W	BOX : \$203.00 TRAY : \$193.00	None
Compare +	Intel® Xeon® Processor E3-1220L v3 (4M Cache, 1.10 GHz)	Launched	Q3'13	2	13 W	TRAY : \$193.00	None
Compare +	Intel® Xeon® Processor E3-1225 v3 (8M Cache, 3.20 GHz)	Launched	Q2'13	4	84 W	TRAY : \$213.00 BOX : \$224.00	Intel® HD Graphics P4600
Compare +	Intel® Xeon® Processor E3-1226 v3 (8M Cache, 3.30 GHz)	Launched	Q2'14	4	84 W	BOX : \$224.00 TRAY : \$213.00	Intel® HD Graphics P4600
Compare +	Intel® Xeon® Processor E3-1230 v3 (8M Cache, 3.30 GHz)	Launched	Q2'13	4	80 W	TRAY : \$240.00 BOX : \$250.00	None
Compare +	Intel® Xeon® Processor E3-1230L v3						

Bild 3.2 Kleiner Überblick über die Xeon E3-Familie

Wenn Sie einen Server mit einem Prozessor aus der E3-Familie, der als SQL Server dienen soll, auswählen müssen, sollten Sie zu dem Modell mit der höchsten Taktfrequenz greifen. Beim SQL Server spielen zwei Parameter eine Rolle:

- Die Parallelität der Verarbeitung, also die Anzahl der Abfragen, die gleichzeitig verarbeitet werden können, wird bestimmt durch die Anzahl der Kerne. In der E3-Familie können Sie hier nicht variieren, weil so gut wie alle Prozessoren als Vier-Kern-Prozessoren ausgelegt sind.
- Die Performance, mit der die einzelne Abfrage verarbeitet wird, wird notwendigerweise durch die Taktfrequenz bestimmt.

In der E3-Familie müssen Sie nur nicht die Entscheidung treffen, ob Sie lieber mehr Kerne oder eine höhere Taktfrequenz wünschen, daher greifen Sie notwendigerweise zur höchsten Taktfrequenz. Natürlich bestimmen auch weitere Parameter als die Taktfrequenz über die Leistung des Systems, wie beispielsweise die Geschwindigkeit des Zugriffs auf den Hauptspeicher. Diese Parameter variieren aber innerhalb der E3-Familie nicht.

Bei der Auswahl von Prozessoren tendieren viele Leute dazu, nicht die teuerste Variante mit der höchsten Taktfrequenz zu wählen. Das ist durchaus sinnvoll, wenn der Server als Domänencontroller oder Dateiserver dienen soll, wo es auf die Prozessorgeschwindigkeit nicht unbedingt ankommt. Beim SQL Server ist diese Vorgehensweise jedoch falsch –jedenfalls dann, wenn Sie die beste Performance wünschen.

Ich kann natürlich jetzt aus diesem Buch heraus nicht erkennen, ob Sie die beste Performance benötigen. Bei der Beschaffung sollte es aber keine Rolle spielen, ob Sie jetzt ein paar 100 Euro mehr ausgeben oder nicht. Zumindest dann nicht, wenn Sie diese Mehrkosten im Vergleich mit SQL Server-Lizenzen betrachten. Und erst recht nicht, wenn Sie später jeden Tag bereuen, dass Sie nicht den schnellsten verfügbaren Prozessor gewählt haben.

Intel® Xeon® Processor E3-1280 v3 (8M Cache, 3.60 GHz)	
<ul style="list-style-type: none"> Applications Essentials Performance Memory Specifications Graphics Specifications Expansion Options Package Specifications Advanced Technologies Intel® Data Protection Technology Intel® Platform Protection Technology 	
Specifications	
Essentials	
Performance	
# of Cores	4
# of Threads	8
Processor Base Frequency	3.6 GHz
Max Turbo Frequency	4 GHz
TDP	82 W
Memory Specifications	
Max Memory Size (dependent on memory type)	32 GB
Memory Types	DDR3-1333/1600
# of Memory Channels	2
Max Memory Bandwidth	25.6 GB/s
ECC Memory Supported ‡	<input checked="" type="checkbox"/> Yes
Graphics Specifications	
Processor Graphics ‡	None
Expansion Options	
PCI Express Revision	3.0
PCI Express Configurations ‡	1x16, 2x8, 1x8/2x4
Max # of PCI Express Lanes	16
Package Specifications	
Max CPU Configuration	1
Package Size	37.5mm x 37.5mm
Graphics and IMC Lithography	22nm

Bild 3.3 Auszug aus den technischen Daten eines E3-Prozessors

In Bild 3.3 sehen Sie einen Auszug aus den technischen Daten eines Xeon E3-Prozessors. Ich möchte jetzt nicht auf jede Einzelheit eingehen, aber ein paar Details thematisieren:

- Der hier gezeigte Prozessor verfügt über vier Kerne und Hyperthreading. Letzteres sehen Sie daran, dass der Parameter *# of Threads* doppelt so hoch ist wie die Anzahl der Kerne.

Hyperthreading wird in Verbindung mit dem SQL Server durchaus kontrovers diskutiert. Ich greife das weiter hinten in diesem Abschnitt nochmals auf.

- Die Taktfrequenz habe ich zuvor bereits als wichtiges Merkmal im Zusammenhang mit SQL Server genannt: Je schneller der Prozessor arbeiten kann, desto schneller wird die Abfrage ausgeführt: Taktfrequenz bringt in diesem Fall also etwas.
- Die E3-Prozessoren unterstützen maximal 32 Gigabyte Hauptspeicher. Das ist nach heutigen Maßstäben nicht viel und könnte durchaus dazu führen, dass Sie sich für eine andere Prozessorarchitektur entscheiden (müssen).
- Die Parameter wie die Anzahl der Kanäle für den Zugriff auf Speicher oder die Bandbreite für den Speicherzugriff sind insbesondere im Vergleich mit anderen Prozessorfamilien interessant, die hier deutlich mehr Leistung bieten (jedenfalls die größeren Server-Prozessoren).
- ECC-Speicher, also Fehler erkennender Speicher, sollte in einem Server Pflicht sein. Wie man sieht, wird dies von der E3-Familie unterstützt.
- Weiterhin ist zu erkennen, dass diese Familie nur Server mit einem Sockel (also einem Prozessor) unterstützt.

3.1.2.2 Prozessoren für mittlere Systeme

Die Xeon E5-Prozessoren stellen die mittlere Familie für Serversysteme dar. Diese Prozessoren sind für Server mit zwei Sockeln gedacht. Server-Systeme dieser Kategorie dürften zu den meistverkauften Servern zählen. Es scheint mir daher, dass die Entwicklung daher in dieser Generation auch besonders zügig vorangeht.

Bild 3.4 zeigt einen Überblick aus der E5-Prozessor-Liste – die Menge an Prozessoren erschlägt einen zunächst. Ich habe zwei Prozessoren ausgesucht, anhand derer ich einige Aspekte diskutieren möchte:

- Zunächst habe ich den E5-2643 ausgesucht, weil es der Prozessor aus der E5-Familie mit der höchsten Taktfrequenz ist. Er hat zwar mit sechs Kernen verhältnismäßig wenig Kerne, wenn Sie maximale Performance für die einzelnen Abfragen und nicht allzu hohe Parallelität benötigen, ist der Prozessor die beste Wahl.
- Das genaue Gegenteil ist der E5-2697, der derzeit die höchste Anzahl an Kernen innerhalb der E5-Plattform bietet (die Varianten mit 16 und 18 Kernen sind im Moment nur angekündigt) und dabei die höchste Taktfrequenz aufweist – die Familie hat noch weitere 14-Kern-Prozessoren, aber deutlich niedriger getaktet.

Compare +	Intel® Xeon® Processor E5-2640 v3 (20M Cache, 2.60 GHz)	Launched	Q3'14	8	90 W	BOX: \$944.00 TRAY: \$939.00
Remove -	Intel® Xeon® Processor E5-2643 v3 (20M Cache, 3.40 GHz)	Launched	Q3'14	6	135 W	TRAY: \$1552.00
Compare +	Intel® Xeon® Processor E5-2650 v3 (25M Cache, 2.30 GHz)	Launched	Q3'14	10	105 W	BOX: \$1171.00 TRAY: \$1166.00
Compare +	Intel® Xeon® Processor E5-2650L v3 (30M Cache, 1.80 GHz)	Launched	Q3'14	12	65 W	TRAY: \$1329.00
Compare +	Intel® Xeon® Processor E5-2660 v3 (25M Cache, 2.60 GHz)	Launched	Q3'14	10	105 W	BOX: \$1445.00 TRAY: \$1445.00
Compare +	Intel® Xeon® Processor E5-2667 v3 (20M Cache, 3.20 GHz)	Launched	Q3'14	8	135 W	TRAY: \$2057.00
Compare +	Intel® Xeon® Processor E5-2670 v3 (30M Cache, 2.30 GHz)	Launched	Q3'14	12	120 W	BOX: \$1593.00 TRAY: \$1589.00
Compare +	Intel® Xeon® Processor E5-2680 v3 (30M Cache, 2.50 GHz)	Launched	Q3'14	12	120 W	TRAY: \$1745.00 BOX: \$1749.00
Compare +	Intel® Xeon® Processor E5-2683 v3 (35M Cache, 2.00 GHz)	Launched	Q3'14	14	120 W	TRAY: \$1846.00
Compare +	Intel® Xeon® Processor E5-2687W v3 (25M Cache, 3.10 GHz)	Launched	Q3'14	10	160 W	BOX: \$2145.00 TRAY: \$2141.00
Compare +	Intel® Xeon® Processor E5-2690 v3 (30M Cache, 2.60 GHz)	Launched	Q3'14	12	135 W	TRAY: \$2090.00 BOX: \$2094.00
Compare +	Intel® Xeon® Processor E5-2695 v3 (35M Cache, 2.30 GHz)	Launched	Q3'14	14	120 W	TRAY: \$2424.00 BOX: \$2428.00
Remove -	Intel® Xeon® Processor E5-2697 v3 (35M Cache, 2.60 GHz)	Launched	Q3'14	14	145 W	BOX: \$2706.00 TRAY: \$2702.00
Compare +	Intel® Xeon® Processor E5-2698 v3 (40M Cache, 2.30 GHz)	Announced	Q3'14	16	135 W	N/A
Compare +	Intel® Xeon® Processor E5-2699 v3 (45M Cache, 2.30 GHz)	Announced	Q3'14	18	145 W	N/A

Bild 3.4 Die E5-Familie hat viele Mitglieder.

Bild 3.5 zeigt die Gegenüberstellung der technischen Parameter der beiden Prozessoren, die ich zuvor genannt habe: Die für unsere Betrachtung relevanten Parameter sind insbesondere Anzahl der Kerne und Taktfrequenz. Sie sehen weiterhin, dass die meisten anderen Parameter, wie beispielsweise die Übertragungsrate auf den Systembus, identisch sind. Wenn man auf die Preise des Prozessors guckt, sieht man, dass die Variante mit mehr Kernen etwas weniger als doppelt so teuer ist, allerdings würde ich diesen Parameter nicht überbewerten. Auf den Gesamtpreis des Servers wirkt sich der Preis des Prozessors natürlich aus, da Sie aber auch beispielsweise viel Speicher und viele Festplatten kaufen müssen, ist der prozentuale Anteil nicht so entscheidend – obwohl Sie immer zwei Prozessoren berücksichtigen müssen.

	Intel® Xeon® Processor E5-2697 v3 (35M Cache, 2.60 GHz)	Intel® Xeon® Processor E5-2643 v3 (20M Cache, 3.40 GHz)
▶ Product Name	Intel® Xeon® Processor E5-2697 v3 (35M Cache, 2.60 GHz)	Intel® Xeon® Processor E5-2643 v3 (20M Cache, 3.40 GHz)
▶ Code Name	Haswell	Haswell
Essentials		
▶ Status	Launched	Launched
▶ Launch Date	Q3'14	Q3'14
▶ Processor Number	E5-2697V3	E5-2643V3
▶ Cache	35 MB Intel® Smart Cache	20 MB Intel® Smart Cache
▶ Bus Type	QPI	QPI
▶ System Bus	9.6 GT/s	9.6 GT/s
▶ # of QPI Links	2	2
▶ Instruction Set	64-bit	64-bit
▶ Instruction Set Extensions	AVX 2.0	AVX 2.0
▶ Embedded Options Available	No	No
▶ Lithography	22 nm	22 nm
▶ Scalability	2S	2S
▶ VID Voltage Range	0.65V–1.30V	0.65V–1.30V
▶ Recommended Customer Price	TRAY: \$2702.00 BOX : \$2706.00	TRAY: \$1552.00
▶ Datasheet	Link	Link
▶ Product Brief	Link	Link
▶ Additional Information URL	Link	Link
Performance		
▶ # of Cores	14	6
▶ # of Threads	28	12
▶ Processor Base Frequency	2.6 GHz	3.4 GHz
▶ Max Turbo Frequency	3.6 GHz	3.7 GHz
▶ TDP	145 W	135 W

Bild 3.5 Zwei Prozessoren der E5-Familie im Vergleich

Was bedeutet dies nun für die Auswahl des Prozessors? Unsere englischen Freunde würden sagen: „It depends.“ Ich kann Ihnen hier keine Antwort geben, die pauschal immer richtig ist, sondern möchte folgende Punkte nennen:

- Es gilt die Faustregel: Viele Kerne bringen eine hohe Verarbeitungskapazität, also eine hohe Parallelität von gleichzeitig verarbeitbaren Abfragen. Eine hohe Taktfrequenz bringt Performance für jede einzelne Abfrage. Welches Szenario in Ihrem Fall das günstigere ist, kann nur von Fall zu Fall entschieden werden. Haben Sie sehr viele Benutzer, deren Abfragen auf die Datenbank relativ simpel sind, könnte es richtig sein, die Variante mit sehr vielen Kernen zu wählen. Müssen Sie eher eine moderate Anzahl von Benutzern versorgen, aber die Abfragen jeweils recht komplex sind, bringt es mehr, wenige Kerne und dafür die deutlich höhere Taktfrequenz zu haben.

- Wenn Sie die Ausführung des vorherigen Aufzählungspunkts lesen und sich fragen, was zu tun ist, wenn Sie sehr viele Benutzer haben und gleichzeitig die Abfragen recht performanceintensiv sind, gibt es hier noch einen Aspekt, den man diskutieren könnte: Wenn Sie die Datenbanklast über zwei Server verteilen können, wäre dies ein Weg, den man auch gehen könnte: Zwei Server mit E5-2643-Prozessoren sind mit Sicherheit schneller als ein Server mit E5-2697-Prozessoren – obwohl der E5-2697-Prozessor-Server trotzdem etwas mehr Kerne hat. Das liegt zum einen an der Taktfrequenz, mit der jeder einzelne Kern betrieben wird, zum anderen aber auch – grob vereinfachend gesagt – daran, dass bei der Verteilung über zwei Server einfach mehr freie Ressourcen verfügbar sind, beispielsweise beim Zugriff auf den Hauptspeicher. Um es auf einen einfachen Satz zu bringen: Dieselbe Anzahl an Kernen bringt mehr Leistung, wenn sie auf zwei Server verteilt ist und mit höherer Taktfrequenz läuft.
- Zu den Kosten: Wenn Sie mit der Lizenzierung pro Kern arbeiten, spielt die Anzahl der Kerne schon eine wesentliche Rolle. Bei der Enterprise-Edition kostet jeder Kern ungefähr 7000 Euro (die Preise können je nach Ausprägung Ihres Lizenzvertrags deutlich abweichen, ich habe hier einen gegoogelten Straßenpreis angesetzt. Wenn ein Zwei-Sockel-System jeweils voll bestückt ist, müssen in einem Fall zwölf, in dem anderen 28 Kerne lizenziert werden. Mit 84 000 Euro bzw. 196 000 kommen da schon ganz spannende Zahlenwerte heraus. Das bedeutet, dass man sich schon sehr genau überlegen muss, wie viele Kerne man wirklich lizenzieren muss, denn das sind jetzt schon Zahlenwerte, die sich deutlich im Budget niederschlagen. Meine Aussage von weiter vorn, dass es überhaupt keinen Sinn macht, ein paar 100 Euro zu sparen und den Prozessor mit der niedrigeren Taktfrequenz zu nehmen, wird durch diese Zahlen noch verstärkt, denn der eigentliche Kostenfaktor liegt in den Lizenzen für SQL Server – zumindest wenn Sie die Enterprise-Edition benötigen. Bevor Sie jetzt mit blassem Gesichtsausdruck dieses Buch zuklappen, möchte ich Sie bitten, ein wenig weiter hinten die Ausführungen über die Lizenzierung nachzulesen, denn Sie müssen ja nicht notwendigerweise überall Enterprise-Editionen verwenden ...
- In dem vorvorherigen Punkt der Aufzählung hatte ich angeregt, anstelle von einem Server mit Prozessoren mit vielen Kernen falls möglich zwei Server mit Prozessoren mit weniger Kernen zu verwenden. Dieser Gedanke kann auch von der Kostenbetrachtung her attraktiv sein, denn schließlich kostet die Lizenzierung eines Kerns dasselbe Geld, egal ob die Taktfrequenz hoch oder niedrig ist. Mit den beiden genannten Prozessoren hätten Sie also entweder einen Server mit 28 Kernen ($2 \cdot 14$) oder zwei Server mit insgesamt 24 Kernen ($2 \cdot 2 \cdot 6$). Die Zwei-Server-Variante ist bezüglich der Lizenzen sogar 28 000 Euro günstiger, wobei das gesparte Geld natürlich in zusätzlicher Server-Hardware aufgeht, aber für diese Summe können Sie schon ganz ordentliche Hardware kaufen.

3.1.2.2.3 Prozessoren für Server mit vier und acht Sockeln

An der Spitze der Leistungspyramide stehen die Xeon-Prozessoren der E7-Familie. Bild 3.6 zeigt technische Daten von Prozessoren für Systeme mit vier bzw. acht Sockeln. Die Argumentationslinien bezüglich Anzahl der Kerne und Taktfrequenz sind hier dieselben wie im vorherigen Abschnitt bezüglich der E5-Familie. Mit dem E7-8893 lässt sich also ein relativ hoch getaktetes System mit 48 Kernen (acht Prozessoren mit je sechs Kernen) bauen, was bezüglich der Parallelität und der Verarbeitungsgeschwindigkeit pro Abfrage das obere Ende der Fahnenstange markiert. Zum Vergleich habe ich den Prozessor E7-4890 aufge-

führt, mit dem ein System mit 60 Kernen (vier Prozessoren mit je 15 Kernen) aufgebaut werden kann. Wir haben hier mehr Kerne, aber eine niedrigere Taktfrequenz. Die Argumentation ist dieselbe wie im vorherigen Abschnitt.

	Intel® Xeon® Processor E7-8893 v2 (37.5M Cache, 3.40 GHz)	Intel® Xeon® Processor E7-4890 v2 (37.5M Cache, 2.80 GHz)
▶ Product Name	Intel® Xeon® Processor E7-8893 v2 (37.5M Cache, 3.40 GHz)	Intel® Xeon® Processor E7-4890 v2 (37.5M Cache, 2.80 GHz)
▶ Code Name	Ivy Bridge	Ivy Bridge
Essentials		
Performance		
▶ # of Cores	6	15
▶ # of Threads	12	30
▶ Processor Base Frequency	3.4 GHz	2.8 GHz
▶ Max Turbo Frequency	3.7 GHz	3.4 GHz
▶ TDP	155 W	155 W
Memory Specifications		
▶ Max Memory Size (dependent on memory type)	1536 GB	1536 GB
▶ Memory Types	DDR3-1066/1333/1600	DDR3-1066/1333/1600
▶ # of Memory Channels	4	4
▶ Max Memory Bandwidth	85 GB/s	85 GB/s
▶ Physical Address Extensions	46-bit	46-bit
▶ ECC Memory Supported †	Yes	Yes
Expansion Options		
Package Specifications		
▶ Max CPU Configuration	8	4
▶ T _{CASE}	77°C	77°C

Bild 3.6 Prozessoren für Systeme mit vier und acht Sockeln

3.1.2.2.4 Unterschiede zwischen den Familien

Bei den vorherigen Ausführungen habe ich mich ziemlich auf Anzahl der Kerne und Taktfrequenzen gestützt. Es gibt allerdings zwischen den Prozessorfamilien noch weitere signifikante Unterschiede, die zum Teil in Bild 3.7 zu sehen sind. Dies sind beispielsweise Unterschiede in der Geschwindigkeit der Busse, der Geschwindigkeit des Speicherzugriffs, des maximal verwendbaren Speichers und etliches andere mehr, was eben nicht nur Auswirkungen auf den isoliert betrachteten Prozessor hat, sondern auch auf den gesamten damit aufgebauten Server.

Da die Kosten für Systeme mit vier und acht Sockeln deutlich nach oben gehen, ist zu diskutieren, ob es nicht wirtschaftlicher ist, mehrere Systeme aus der Zwei-Sockel-Familie zu nehmen und sozusagen einen Scale-out-Ansatz zu verfolgen. Das funktioniert natürlich nur, wenn sich Ihre Datenbankanwendungen tatsächlich auf verschiedene Systeme verteilen lassen.

Product Name	Intel® Xeon® Processor E3-1241 v3 (8M Cache, 3.50 GHz)	Intel® Xeon® Processor E5-2697 v3 (35M Cache, 2.60 GHz)	Intel® Xeon® Processor E7-8893 v2 (37.5M Cache, 3.40 GHz)	Intel® Xeon® Processor E7-4890 v2 (37.5M Cache, 2.80 GHz)
Code Name	Haswell	Haswell	Ivy Bridge	Ivy Bridge
Cache	8 MB Intel® Smart Cache	35 MB Intel® Smart Cache	37.5 MB	37.5 MB
Bus Type	DMI2	QPI	QPI	QPI
System Bus	5 GT/s	9.6 GT/s	8 GT/s	8 GT/s
# of QPI Links	0	2	3	3
Instruction Set	64-bit	64-bit	64-bit	64-bit
Instruction Set Extensions	SSE 4.1/4.2, AVX 2.0	AVX 2.0	Intel® Advanced Vector Extensions (Intel® AVX)	Intel® Advanced Vector Extensions (Intel® AVX)
Embedded Options Available	No	No	No	No
Lithography	22 nm	22 nm	22 nm	22 nm
Scalability	15 Only	25	S85	S45
Thermal Solution Specification	PCG 2013D			
Recommended Customer Price	BOX : \$273.00 TRAY : \$262.00	BOX : \$2706.00 TRAY : \$2702.00	TRAY : \$6841.00	TRAY : \$6619.00
Datasheet	Link	Link	Link	Link
VID Voltage Range		0.65V–1.30V		
Product Brief		Link	Link	Link
Additional Information		Link	Link	Link
Performance				
# of Cores	4	14	6	15
# of Threads	8	28	12	30
Processor Base Frequency	3.5 GHz	2.6 GHz	3.4 GHz	2.8 GHz
Max Turbo Frequency	3.9 GHz	3.6 GHz	3.7 GHz	3.4 GHz
TDP	80 W	145 W	155 W	155 W
Memory Specification				
Max Memory Size (dependent on memory type)	32 GB	768 GB	1536 GB	1536 GB
Memory Types	DDR3-1333/1600	DDR4-1600/1866/2133	DDR3-1066/1333/1600	DDR3-1066/1333/1600
# of Memory Channels	2	4	4	4
Max Memory Bandwidth	25.6 GB/s	68 GB/s	85 GB/s	85 GB/s

Bild 3.7 Zwischen den Prozessorfamilien gibt es diverse Unterschiede – auch jenseits von Kernen und Taktfrequenzen.

3.1.2.3 Hyperthreading – ja oder nein

Fast alle Server-Prozessoren unterstützen Hyperthreading. Dieses Thema ist unter SQL-Administratoren durchaus nicht unumstritten, folgende sehr vorsichtig formulierte Aussagen halten intensiveren Recherchen und insbesondere der Überprüfung im richtigen Leben stand:

- Hyperthreading kann im SQL Server-Umfeld verwendet werden ab der Prozessorgeneration Nehalem – diese ist erschienen im Jahr 2009. Mit anderen Worten war vor 2009 Hyperthreading mit SQL Server nicht empfehlenswert, mit aktuellen Prozessoren kann man durchaus damit liebäugeln.
- Hyperthreading limitiert die Leistung jeder einzelnen Abfrage auf der CPU zugunsten einer höheren Parallelität. Das ist auch einleuchtend, da durch Hyperthreading nicht etwa neue Kerne entstehen, sondern ein Kern „anders“ benutzt wird. Generell gilt, dass OLTP-Szenarien, die im Allgemeinen aus viel weniger komplexen Abfragen bestehen, von Hyperthreading durchaus deutlich profitieren, während es bei OLAP-Anwendungen (also Szenarien in Verbindung mit Analysis Services) nur geringe Vorteile gibt.
- Und jetzt das Wichtigste: Wenn Sie wirklich sicher sein möchten, ob Hyperthreading in konkret Ihrem Szenario zu Vorteilen führt, können Sie es im Grunde genommen nur tes-

ten. Sie müssen also einen aussagekräftigen Lasttest einmal mit und einmal ohne Hyperthreading und die Ergebnisse messen. Wenn Sie ein wenig mittels Google recherchieren, werden Sie sehen, dass sowohl Intel als auch diverse Serverhersteller genau diese Tests durchgeführt haben. Die Ergebnisse sprechen dafür, Hyperthreading einzuschalten, aber keine größeren Wunder zu erwarten. Ich möchte aber darauf hinweisen, dass es in Ihrem konkreten Fall auch anders sein könnte – sowohl in die eine als auch die andere Richtung. Mir ist klar, dass es organisatorisch nicht so einfach ist, einen Lasttest auszuführen. Wenn Sie allerdings in einer stark belasteten Umgebung wirklich Sicherheit brauchen, führt kein Weg daran vorbei.

3.1.2.4 MAXDOP – Maximum Degree of Parallelism

Im Rahmen der Betrachtung der Prozessoren muss unbedingt auf die Einstellung MAXDOP hingewiesen werden. Die Abkürzung steht für *Maximum Degree of Parallelism*, in der deutschen Lokalisierung des SQL Servers heißt dieser Konfigurationspunkt *Max. Grad an Parallelität* und findet sich in den erweiterten Eigenschaften jeder Instanz (Bild 3.8). Es geht bei dieser Einstellung darum, wie viele Prozessoren (oder das, was wie ein Prozessor aussieht) der SQL Server bei einer Abfrage verwendet. Die Standardeinstellung ist null, was bedeutet, dass der SQL Server für eine Abfrage so viele Prozessoren verwenden darf, wie er gern möchte. Diese Einstellung imitiert übrigens nicht die Anzahl der Prozessoren, die der Server insgesamt benutzt, sondern bezieht sich auf die einzelnen Prozessoren, die bei der Erstellung des Abfrageplans berücksichtigt werden.

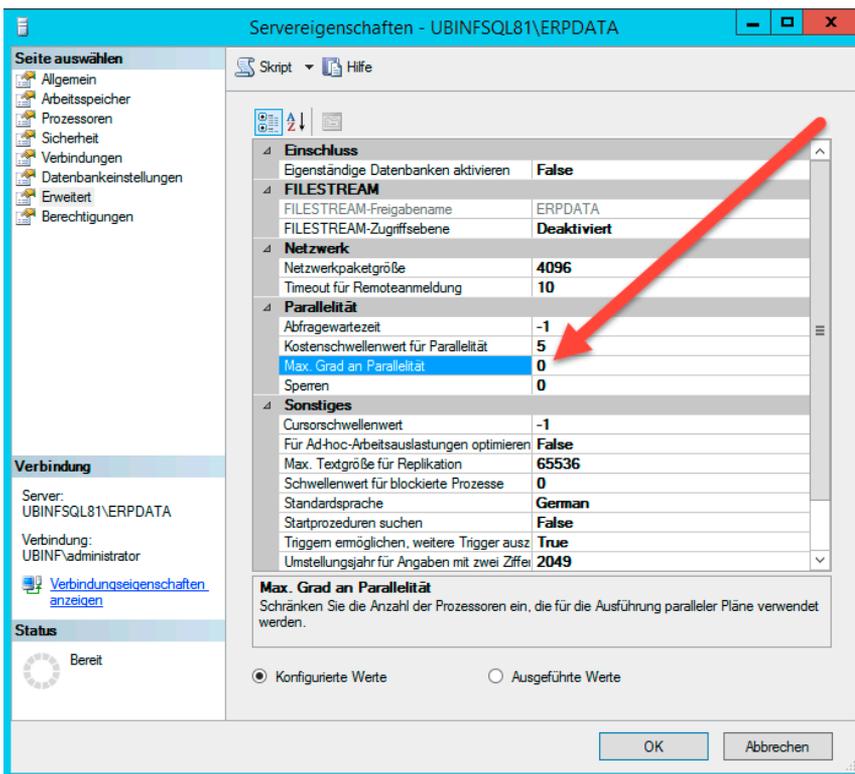


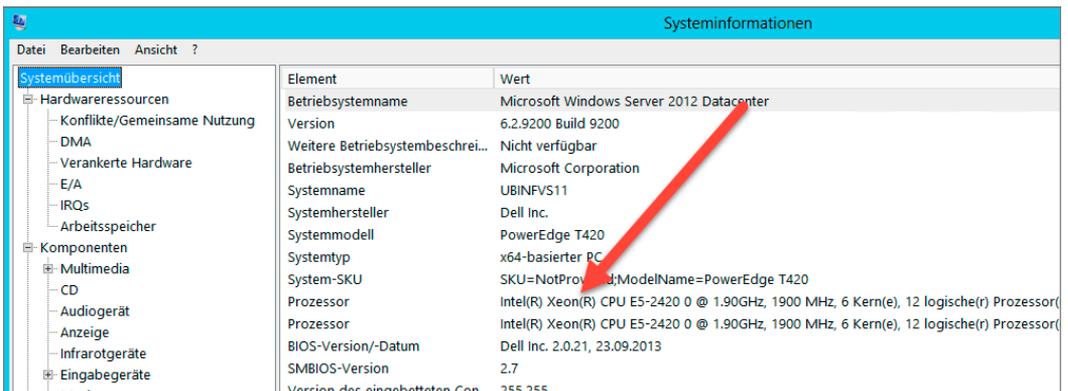
Bild 3.8 Die MAXDOP-Einstellung ist sehr wichtig und findet sich in den Eigenschaften der Instanz.

Wenn der Buchautor so ein Thema anfängt, kann man davon ausgehen, dass die Standard-einstellung von null nicht optimal ist. In der Tat gibt es einen Support-Artikel von Microsoft, der relativ eindeutige Vorgaben macht. Ich zitiere aus <http://support.microsoft.com/kb/2806535/>:

- For servers that use more than eight processors, use the following configuration:
MAXDOP=8
- For servers that use eight or fewer processors, use the following configuration:
MAXDOP=0 to N
Note In this configuration, N represents the number of processors.
- For servers that have NUMA configured, MAXDOP should not exceed the number of CPUs that are assigned to each NUMA node.
- For servers that have hyperthreading enabled, the MAXDOP value should not exceed the number of physical processors.
- For servers that have NUMA configured and hyperthreading enabled, the MAXDOP value should not exceed number of physical processors per NUMA node.

Wie ist dieser Support-Artikel also zu interpretieren?

Ein „Prozessor“ in diesem Zusammenhang ist ein Kern. Ich habe mit MSInfo32 die Konfiguration einer physikalischen Maschine ausgelesen, das Ergebnis sehen Sie in Bild 3.9: Zwei Prozessoren mit sechs Kernen ergeben zwölf Prozessoren, demzufolge müsste der MAXDOP-Wert auf acht gestellt werden.



Element	Wert
Betriebssystemname	Microsoft Windows Server 2012 Datacenter
Version	6.2.9200 Build 9200
Weitere Betriebssystembeschri...	Nicht verfügbar
Betriebssystemhersteller	Microsoft Corporation
Systemname	UBINFVS11
Systemhersteller	Dell Inc.
Systemmodell	PowerEdge T420
Systemtyp	x64-basierter PC
System-SKU	SKU=NotPro...;ModelName=PowerEdge T420
Prozessor	Intel(R) Xeon(R) CPU E5-2420 0 @ 1.90GHz, 1900 MHz, 6 Kern(e), 12 logische(r) Prozessor(e)
Prozessor	Intel(R) Xeon(R) CPU E5-2420 0 @ 1.90GHz, 1900 MHz, 6 Kern(e), 12 logische(r) Prozessor(e)
BIOS-Version/-Datum	Dell Inc. 2.0.21, 23.09.2013
SMBIOS-Version	2.7
Version des eingebetteten Con...	255.255

Bild 3.9 Die Ausgabe von MSInfo32 zeigt kein Hardware-Layout des Servers.

Aber Moment, da war doch noch was mit NUMA ... Blöderweise ist das Hardware-Layout der Maschine in der Ausgabe von MSInfo32 nicht zu erkennen. Wenn man die ganze Geschichte im Ressourcenmonitor anschaut, sieht man, dass sich die zwölf Kerne (alias physikalischen Prozessoren) auf zwei NUMA-Noten verteilen (Bild 3.10).

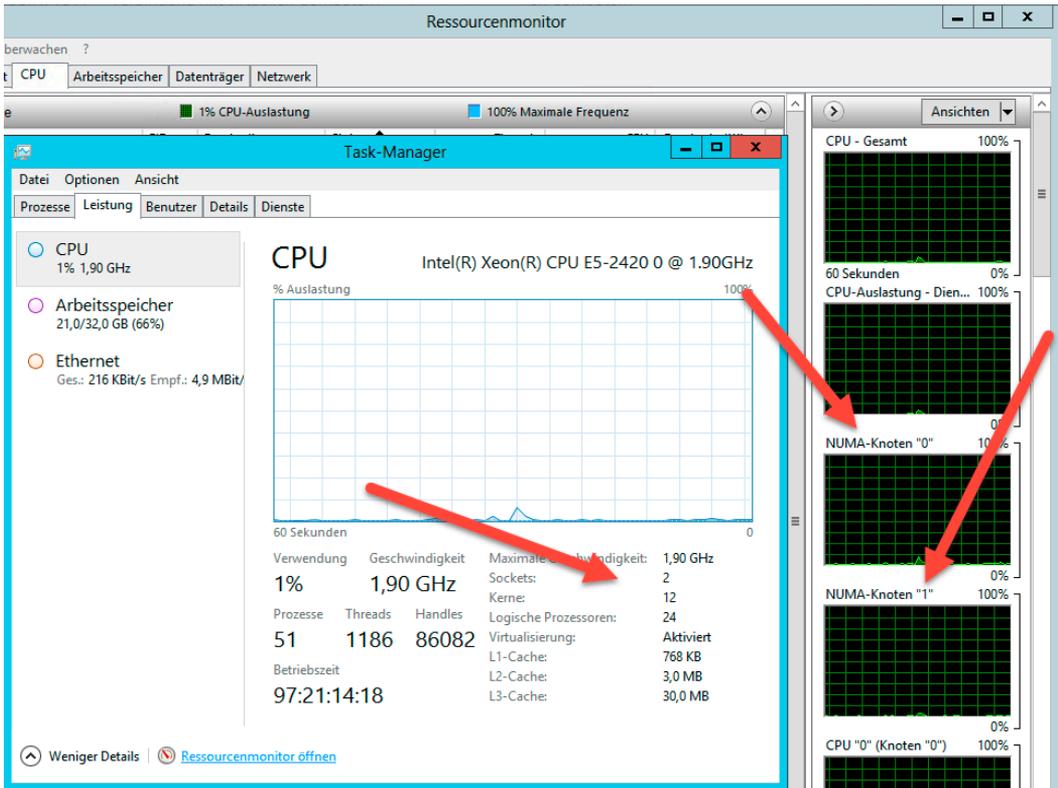


Bild 3.10 Im Ressourcenmonitor kann man erkennen, dass zwei NUMA-Knoten vorhanden sind.

Das Resultat dieser Erkenntnis ist, dass die Errechnung des optimalen Wertes für MAXDOP wie folgt aussieht:

- Zwei NUMA-Knoten zu je sechs physikalischen Prozessoren plus Hyperthreading
- Support-Artikel: „*For servers that have NUMA configured and hyperthreading enabled, the MAXDOP value should not exceed number of physical processors per NUMA node*“
- Fazit: **MAXDOP = 6**

Man kann im SQL Server Management Studio auch die Anzahl der NUMA-Knoten ermitteln. Die Konfigurationsseite *Prozessoren* der Eigenschaften der Instanz gibt in einer Baumansicht die NUMA-Knoten nebst vorhandener Prozessoren an. Dies ist in Bild 3.11 zu sehen, allerdings wurde dieser Screenshot auf einer virtuellen Maschine angefertigt und dort gibt es dieses Hardware-Layout nicht, demzufolge ist nur ein NUMA-Knoten vorhanden.

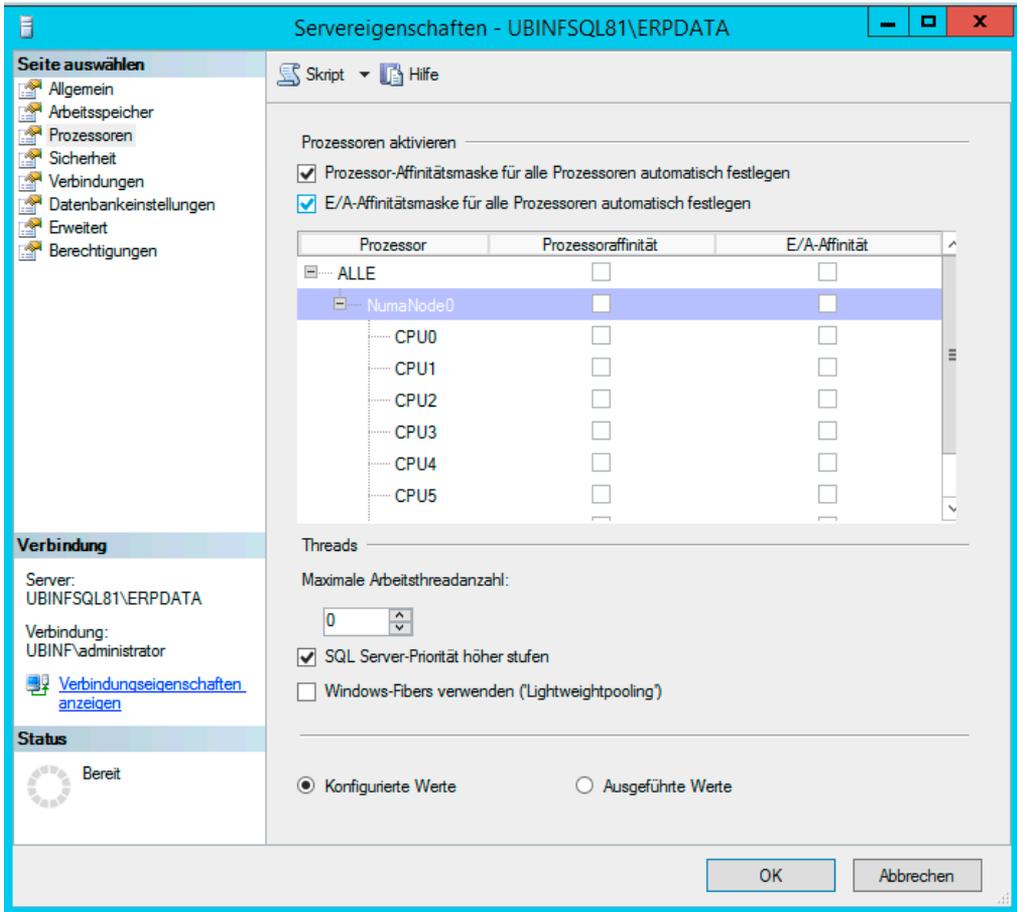


Bild 3.11 Im SQL Server Management Studio kann in den Eigenschaften des Prozessors auch eingesehen werden, wie viele NUMA-Knoten vorhanden sind.

3.1.2.5 Anzahl der Kerne pro Edition

Zu beachten ist, dass nicht jede Edition von SQL Server beliebig viele Prozessorkerne unterstützt. Die nachfolgend gezeigte Tabelle gibt die entsprechenden Werte an. Quelle dieser Aufstellung ist <http://msdn.microsoft.com/de-de/library/ms143760.aspx>.



HINWEIS: Es gibt einen Sonderfall: Für Lizenzen, die über Software Assurance zu SQL Server 2014 gekommen sind, könnte es Enterprise-Edition-Server geben, die nicht pro Prozessorkern lizenziert sind, sondern die alte Lizenzierung mit CALs abbilden. In einem solchen Fall unterstützt die Enterprise Edition lediglich 20 Kerne.

Für neue Lizenzen ist das übrigens nicht möglich, dort gibt es für die Enterprise Edition nur die Lizenzierung pro Prozessorkern.

SQL Server-Edition	Maximale von einer einzelnen Instanz verwendete Rechenkapazität (SQL Server Database Engine (Datenbankmodul))	Maximale von einer einzelnen Instanz verwendete Rechenkapazität (AS, RS)
Enterprise Edition: Kernbasierte Lizenzierung	Maximum des Betriebssystems	Maximum des Betriebssystems
Entwickler	Maximum des Betriebssystems	Maximum des Betriebssystems
Evaluation	Maximum des Betriebssystems	Maximum des Betriebssystems
Business Intelligence	Beschränkt auf weniger als vier Sockets oder 16 Kerne	Maximum des Betriebssystems
Standard	Beschränkt auf weniger als vier Sockets oder 16 Kerne	Beschränkt auf weniger als vier Sockets oder 16 Kerne
Web	Beschränkt auf weniger als vier Sockets oder 16 Kerne	Beschränkt auf weniger als vier Sockets oder 16 Kerne
Express	Beschränkt auf weniger als ein Socket oder vier Kerne	Beschränkt auf weniger als ein Socket oder vier Kerne
Express mit Tools	Beschränkt auf weniger als ein Socket oder vier Kerne	Beschränkt auf weniger als ein Socket oder vier Kerne
Express with Advanced Services	Beschränkt auf weniger als ein Socket oder vier Kerne	Beschränkt auf weniger als ein Socket oder vier Kerne

3.1.2.6 ... und in virtualisierten Umgebungen?

Die Überlegungen über die Anzahl der benötigten Prozessoren gelten analog auch, wenn Sie Ihren SQL Server in einer virtualisierten Umgebung betreiben. Der Unterschied ist dann natürlich, dass Sie die Mehrkern-Architekturen moderner Prozessoren durch die Virtualisierung „verdecken“, anders gesagt können Sie natürlich auswählen, wie viel Prozessoren in der virtuellen Maschine sichtbar sein sollen, die Differenzierung nach Sockeln, Kernen und gegebenenfalls Hyperthreading entfällt. Problematisch ist eventuell, dass Sie in der virtuellen Maschine nicht so viele Prozessoren verwenden können, beispielsweise Microsoft Hyper-V ermöglicht, bis zu acht Prozessoren zu verwenden. Das setzt der Skalierbarkeit natürlich deutliche Grenzen – die Frage ist, wie viele Prozessoren für die jeweiligen Anwendungsfälle des SQL Servers benötigt werden. Damit entscheidet sich, ob der SQL Server virtualisiert werden kann – so einfach ist das.

Hinzu kommt, dass die individuelle Leistung des einzelnen Prozessors durch die Virtualisierungsschicht notwendigerweise verringert ist.

Alles in allem muss das nicht dramatisch sein und ich möchte mich hier auch keinesfalls hinstellen und verkünden, dass Sie SQL Server nicht virtualisieren können. Jedoch muss deutlich angemerkt werden, dass die Skalierbarkeit deutlich schlechter ist, denn die beiden Kenngrößen für die Leistungsfähigkeit des SQL Servers bezüglich der Prozessoren sind nun eben:

- Parallelität der Verarbeitung: Je mehr Kerne/Prozessoren im SQL Server zur Verfügung stehen, desto mehr parallele Abfragen kann er performant verarbeiten.

- Leistung bei der Verarbeitung der einzelnen Abfrage: Je leistungsfähiger der einzelne Kern ist, desto schneller kann jede einzelne Abfrage verarbeitet werden. Beim SQL Server spielt sogar die Taktfrequenz der Prozessoren eine wesentliche Rolle (im Gegensatz zu vielen anderen Anwendungen), sodass die Minderleistung eines Prozessors in einer virtualisierten Umgebung durchaus eine Rolle spielen könnte.

Mir ist klar, dass die Virtualisierung signifikante Vorteile im Hinblick auf Administration und Betrieb bietet – ich bin ja auch davon überzeugt, dass Virtualisierung in vielen Fällen der beste Weg ist.

Es ist aber deutlich darauf hinzuweisen, dass die Performance-Einschränkungen, die durch die Virtualisierung entstehen, signifikanten Einfluss auf den SQL Server haben könnten. Insofern hilft es nur, eine Pilotumgebung zu bauen und dort Lasttests durchzuführen oder eben die Produktivumgebung sorgfältig zu beobachten und kontinuierlich kritische Performance-Kenndaten zu erfassen und auszuwerten.

3.1.2.7 Performance messen und überwachen

An dieser Stelle sei zu erwähnen, dass man die Prozessorleistung oder vorhandene Engpässe relativ gut mit dem Performance-Monitor messen kann. Überhaupt spielt die Überwachung der Performance im SQL Server-Umfeld eine ganz wesentliche Rolle. In diesem Buch ist diesem Themenkomplex ein eigenes Kapitel gewidmet („Monitoring und Überwachung“), auf das ich an dieser Stelle hinweisen möchte.

3.1.3 Hauptspeicher

Neben der Auslegung der Prozessoren bezüglich Taktfrequenz und Anzahl (nicht zu vergessen der Entscheidung für oder gegen Hyperthreading) ist die Menge des Speichers für die Performance des SQL Servers einer der entscheidenden Faktoren. Kurz gesagt kann man festhalten, dass der SQL Server alles das, was er nicht im Hauptspeicher halten kann, von der Festplatte nachladen muss. Bei einer Abfrage, die er hundertmal ausführt, ist es natürlich dramatisch viel schneller, wenn er die Daten einmal von der Platte liest und 99 mal aus dem Speicher bereitstellen kann, als hundertmal von der Platte zu lesen. Diese Aussage ist natürlich grob vereinfachend, so einfach sind die Zusammenhänge dann doch nicht, aber wenn Sie sich die Welt des SQL Servers ungefähr so vorstellen, haben Sie einen groben Anhaltspunkt, warum viel Speicher generell auch viel bringt.

Ich sehe hin und wieder SQL Server, die durchaus nicht wenig zu tun haben, die in einer virtuellen Maschine mit zwei Prozessoren und vier Gigabyte RAM (für alles!) laufen – das kann nicht schnell sein. Niemals!

Vermutlich haben Sie dieses Kapitel in der Hoffnung aufgeschlagen, dass ich Ihnen eine elegante Formel zur Berechnung des Hauptspeicherbedarfs für SQL Server anbiete. Tut mir leid, ich muss passen! Ich möchte Ihnen allerdings gerne folgende Gedanken auf den Weg geben:

- Wenn Sie die Microsoft-Dokumentation befragen, werden Sie die in Bild 3.12 gezeigte Seite finden, in der Microsoft Angaben zum Hauptspeicherbedarf macht. Die Aussage ist, dass es mindestens vier Gigabyte sein sollen und man es mit der Größe der Datenbank vergrößern soll, um die optimale Performance zu erreichen. Viel allgemeiner geht es

5

Backup und Restore

SQL Server ist nun eindeutig ein System, das Bewegungsdaten speichert. Und nicht nur das: Diese Daten sind stetiger und häufiger Änderungen unterworfen, demzufolge spielt die Datensicherung eine herausragende Rolle bei allen SQL Server-Szenarien.

Wie bereits zu Beginn des vorherigen Kapitels über Hochverfügbarkeit beschrieben, muss betrachtet werden, welche Datenverlustzeit akzeptabel ist und welche Wiederherstellungszeit erreicht werden muss. Wenn die Datenverlustzeit im Bereich von maximal einer Minute liegen darf, wird man mit Mitteln der Datensicherung das gesetzte Ziel nicht erreichen.

Der Gedanke hinter dem Anfertigen von Sicherungen ist, dass es eventuell notwendig werden könnte, eine Rücksicherung durchzuführen. Das ist so weit einleuchtend und wenig überraschend. In vielen mir bekannten Szenarien wird allerdings fleißig gesichert, ohne dass die Anforderungen genau hinterfragt worden wären. Das ist zwar besser als gar keine Sicherung, grundsätzlich sollten Sie sich vor dem Erstellen des technischen Datensicherungskonzepts im Klaren darüber sein, welche Anforderungen zu erfüllen sind. Teilweise gehen die Vorstellungen von IT-Verantwortlichen und den Fachabteilungen, deren Daten zu sichern sind, deutlich auseinander. In letzter Konsequenz ist es auch eine Frage des Geldes: Wenn die Geschäftsleitung der Meinung ist, dass der Datenbankserver nach spätestens einer halben Stunde wieder zur Verfügung stehen muss, sind für dieses Ziel auch entsprechende Mittel notwendig.

■ 5.1 Einige Gedanken und Fakten vorab

5.1.1 Servicelevel

Es kann keinen Abschnitt über Anforderungen an die IT-Umgebung geben, ohne dass das Thema Servicelevel angesprochen wird. Diese Vereinbarung regelt, in welcher Zeit und welchem Umfang eine Leistung erbracht wird, also beispielsweise:

- Wie schnell kann der Anwender eine Antwort auf seine Frage zu seinem Druckproblem aus Word erhalten?
- Wie schnell wird ein defekter PC ausgetauscht?

- Wie schnell wird die Funktionalität eines Dienstes (z. B. E-Mail-System) nach einem Ausfall wieder in vollem Umfang zur Verfügung stehen?

Bei einem Projektgespräch, beispielsweise zur Einführung eines SQL Server 2014-Systems als Ablösung für einen bestehenden SQL Server 2008, ist eine meiner ersten Fragen, welche Wiederherstellungszeiten erreicht werden müssen. Ganz klar: Das ist eine Frage, an der einerseits Geld, andererseits aber auch die Anwenderzufriedenheit, vielleicht sogar die Existenz des Unternehmens hängt. In vielen Fällen antworten mir die Verantwortlichen, dass mit der Geschäftsleitung oder der Fachabteilung nie über derlei Servicelevel geredet wurde oder aber diese zumindest nicht verbindlich vereinbart sind. Gut, für das, was nicht vereinbart wurde, kann auch niemand zur Rechenschaft gezogen werden. Diese Argumentation hinkt aber.

Geschäftsleitung und Anwender gehen zunächst davon aus, dass die Systeme immer funktionieren. Und wenn wider Erwarten doch eine Störung auftreten sollte, lässt sich diese ja wohl innerhalb einiger Minuten, höchstens aber einer halben Stunde beheben. Natürlich kann man ein System aufbauen, das diesen Anforderungen gerecht wird – das ist nur eine Frage der Bereitstellung einer hinreichend großen Menge Geldes. Man kann ja beliebig viele Beispiele aufführen:

- Wenn beispielsweise ein Datenbankserver für den Kontakt mit Kunden so wichtig ist, dass dieser höchstens für eine halbe Stunde ausfallen darf, kann ein solcher Servicelevel vereinbart werden, allerdings zieht das auch eine Investition von beispielsweise EUR 150 000 nach sich.
- Wenn die Benutzer erwarten, dass ein defekter PC innerhalb von einer Stunde ausgetauscht oder eine zusätzlich benötigte Software „sofort“ installiert wird, ist das ohne Weiteres möglich. Die Erfüllung eines so vereinbarten Servicelevels erfordert aber unter Umständen, dass eine weitere Person eingestellt oder aber ein externer Dienstleister mit einigen Aufgaben betraut wird.

Mit der Vereinbarung eines Servicelevels setzt sich die IT zwar unter Druck, die zugesicherten Leistungen einzuhalten. Auf der anderen Seite schützt sie sich aber auch vor Anforderungen, die sie einfach nicht erfüllen kann. Natürlich geht jeder davon aus, dass Server nie ausfallen und wenn doch, innerhalb weniger Minuten wieder laufen. Natürlich erwarten die Benutzer, dass jede kleine Störung oder jede Anfrage mit höchster Priorität, also SOFORT bearbeitet wird. Die Vereinbarung von internen Serviceleveln trägt also dazu bei, dass klar ist, was die IT leisten muss, aber auch was von ihr erwartet werden kann. Im Übrigen kann nur mit Verweis auf einen geforderten Servicelevel eine Investition in bessere Verfügbarkeit oder in Management-Werkzeuge erfolgen.

Die Vereinbarung von Serviceleveln muss nun ja keinesfalls im Rahmen eines formellen internen Vertragsdokuments geschehen. Im mittelständischen Umfeld genügt es im Allgemeinen, wenn gemeinsam mit der Geschäftsleitung ein Protokoll angefertigt wird, in dem die wichtigsten Servicelevel definiert sind. Ob das dann Servicelevel heißt oder ein etwas weniger hochtrabender Begriff gewählt wird, sei jedem selbst überlassen. Meiner Erfahrung nach wird es aber ohne eine solche Vereinbarung wie folgt sein:

- Gefühlter Servicelevel aus Sicht der Geschäftsleitung für die Problembehebung bei SQL Server-Problemen: 30 Minuten
- ... und der IT-Verantwortliche setzt voraus, dass ein Ausfall von zwei Tagen gerade noch akzeptabel ist

Konsequenz: Im Fall des Falles wird es Ärger geben. Hätte man doch vorher darüber gesprochen ... Gut, vielleicht werden Hochverfügbarkeitsmaßnahmen für eine Viertelmillion Euro trotzdem nicht genehmigt, aber die Geschäftsleitung weiß, dass sie die 30-Minuten-Problembeseitigung nicht erwarten kann.

5.1.2 Wiederherstellungszeit

Zunächst möchte ich das Szenario der Wiederherstellung eines Servers, dessen lokale Plattensysteme so ausgefallen sind, dass ein Restore der Daten notwendig wird, betrachten. Dies könnte beispielsweise im Fall eines RAID-Controller-Defekts vorkommen. Ich gehe von einem Server mit einer Nutzkapazität von 300 GB aus. Auf dem Zeitstrahl ist der Vorgang dargestellt (Bild 5.1):

- Um 10:00 fällt das System aus.
- Kurz danach werden die ersten Störmeldungen eingehen. Bis die Ursache des Problems „Die Anwendung geht nicht mehr“ erkannt worden ist und die notwendigen Schritte eingeleitet worden sind, vergeht mit Sicherheit eine Stunde. Schließlich ist nicht ständig ein IT-Mitarbeiter in Wartestellung, wahrscheinlich wird zunächst eine Behebung des Fehlers versucht werden etc.

Ausfallzeit bis hierhin: 1 Stunde

- Sofern ein Service-Vertrag für die Instandsetzung der Hardware (!) vorliegt, wird diese nach sechs Stunden wieder funktionsbereit sein. Eine Wiederherstellungszeit von sechs Stunden ist der schnellste Standard-Servicelevel, der gemeinhin von Herstellern und Systemhäusern angeboten wird. (Ein Service-Vertrag, der eine Reaktionszeit von vier Stunden garantiert, ist niederwertiger als einer mit sechs Stunden Wiederherstellungszeit).

Ausfallzeit bis hierhin: sieben Stunden

- Ist die Hardware wieder funktionsbereit, wird ein gewisser Zeitraum, sagen wir eine Stunde, vergehen, bis tatsächlich mit der Rücksicherung begonnen werden kann. Schließlich muss die Backup-Software betriebsbereit gemacht, wahrscheinlich Bänder herausgesucht, kurzum einige Vorbereitungen müssen getroffen werden.

Ausfallzeit bis hierhin: acht Stunden

- Nun beginnt die eigentliche Rücksicherung. Eine Restore-Geschwindigkeit von 300 MB/Minute ist eine realistische Annahme (wenn Sie nicht gerade die komplette Backup-Hardware erneuert haben), woraus sich ergibt:

$$(300 \text{ GB} * 1024) / 300 \text{ MB} = 1024 \text{ Min.} = 17,07 \text{ Stunden}$$

Es muss also von einer Restore-Zeit von ungefähr 17 Stunden ausgegangen werden.

Ausfallzeit bis hierhin: 25 Stunden

- Nach Abschluss des Wiederherstellungsvorgangs müssen sicherlich noch einige Nacharbeiten vorgenommen werden. Ist alles gut geplant, dürfte das nicht allzu aufwendig sein, daher ist eine Stunde ein realistischer Schätzwert.

Ausfallzeit bis hierhin: 26 Stunden

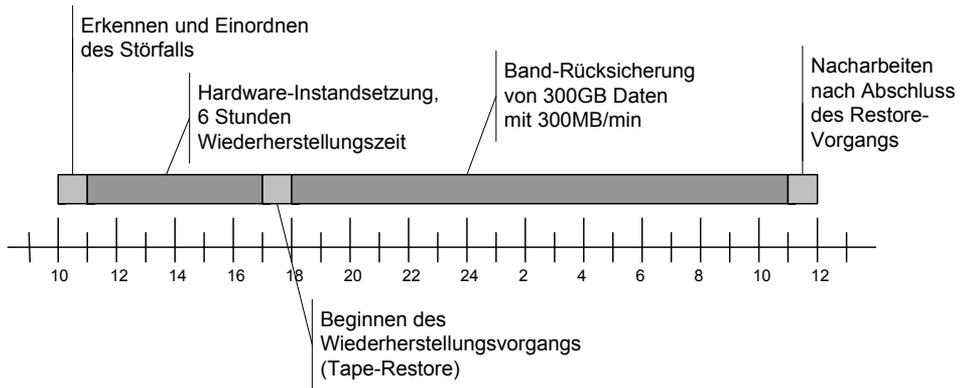


Bild 5.1 Wiederherstellung eines Systems

Dieses einfache Beispiel zeigt recht eindrucksvoll, welche enormen Risiken in den IT-Systemen stecken: Ein Ausfall eines kritischen Systems von mehr als 24 Stunden kann für viele Firmen akut existenzbedrohend sein, zumindest dürfte es als massive Störung gesehen werden. Letztendlich ist der zuvor geschilderte Ablauf noch recht optimistisch gewesen. Wenn während des Vorgangs, bei welchem Arbeitsschritt auch immer, Probleme auftreten, verlängert das die Wiederherstellungszeiten eventuell deutlich. Wenn Sie Optimierungspotenzial suchen, finden sich zwei Ansätze:

- Die Beschleunigung der Hardware-Wiederherstellung
- Die Beschleunigung der Rücksicherung

Ersteres lässt sich eventuell mit im Unternehmen gelagerter Ersatzhardware erreichen, es stellt sich hierbei allerdings die Frage, ob jederzeit ein Mitarbeiter, der Hardwareprobleme eines Servers erkennen und beheben kann, zur Verfügung steht. Die Beschleunigung der Rücksicherung ist natürlich ebenfalls möglich. Schnellere Backup-Hardware und sehr performante Serversysteme ermöglichen zwar höhere Wiederherstellungsgeschwindigkeiten, dennoch bleibt eine Rücksicherung größerer Datenmengen eine zeitaufwendige Angelegenheit.

Folgende Schlussfolgerung ergibt sich aus dieser Betrachtung:

- Sofern ein Server bzw. dessen Applikationen nicht länger als beispielsweise vier oder sechs Stunden ausfallen dürfen, ist dies mit einem normalen Sicherheits-/Wiederherstellungsszenario nicht zu schaffen.
- Vielleicht wird, entweder aus finanziellen Gründen oder weil die Verfügbarkeit für bestimmte Systeme lediglich eine untergeordnete Rolle spielt, entschieden, keine erweiterten Maßnahmen zu ergreifen. In diesem Fall sollte unbedingt schriftlich festgestellt und kommuniziert werden, dass es zu längeren Ausfällen kommen kann.

5.1.3 Datenverlustzeit

In vielen mittelständischen Unternehmen wird die Wiederherstellung der Systeme nicht mit so hoher Wichtigkeit belegt. Viel entscheidender ist es häufig sicherzustellen, dass keine Daten verloren werden. Ich möchte dazu ein Szenario auf dem Zeitstrahl betrachten (Bild 5.2):

- Die Datensicherung ist um 06:00 Uhr abgeschlossen.
- Um 8 Uhr nehmen die Benutzer die Arbeit auf und verändern die Daten.
- Am Nachmittag um 16 Uhr tritt ein Störfall auf. Dies ist ein schwerer Störfall, es gehen also beispielsweise die Festplattensysteme verloren (= Daten sind zumindest nicht mehr zu lesen).
- Wenn keine zusätzlichen Sicherungsmaßnahmen getroffen werden, bedeutet dies, dass die in diesen acht Stunden produzierten Daten verloren werden (von acht bis 16 Uhr).

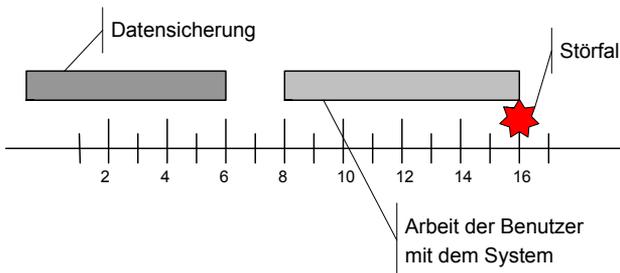


Bild 5.2
Die Datenverlustzeit

Bei der Betrachtung des Datenverlusts sind zwei Fälle zu beachten:

- Reproduzierbare Daten
- Nicht reproduzierbare Daten

Ein Beispiel für reproduzierbare Daten wären Buchungen von Eingangsrechnungen (die Papierrechnungen liegen ja noch vor und werden nochmals gebucht) oder eine CAD-Zeichnung, die natürlich auch ein zweites Mal angefertigt werden kann. Nicht reproduzierbar sind beispielsweise empfangene E-Mails (hat man nicht zufällig kurz vor dem Ausfall des Systems seinen Posteingang eingesehen, weiß man ja nicht, wer geschrieben hat, und kann nachfragen) oder die Auftragseingangsdaten eines Webshops.

Wenn die Anforderung an die IT-Abteilung herangetragen wird, dass ein Verlust von Daten auf einigen oder sogar allen Systemen nicht tragbar ist (Datenverlustzeit kleiner als eine Minute), müssen weitergehende Maßnahmen ergriffen werden; ein normales Sicherungs-/Wiederherstellungskonzept ist eindeutig nicht ausreichend.

Man sollte sich nicht über die scheinbare Sicherheit, die redundant ausgelegte Server oder mit RAID-Leveln konfigurierte Plattensysteme vorspiegeln, täuschen: Wir sprechen bei den Überlegungen zur Verfügbarkeit grundsätzlich vom schlimmstmöglichen Fall und dieser könnte so aussehen, dass das gesamte Festplattensystem irreparabel beschädigt wird.

5.1.4 Logische Fehler

In den zuvor beschriebenen Situationen habe ich stets Hardwareprobleme als Grund für den Wiederherstellungsfall angegeben. Völlig klar ist, dass auch logische Fehler zur Notwendigkeit, eine Sicherung einzuspielen, führen können. Diese logischen Fehler könnten beispielsweise durch Programmierfehler in Anwendungsprogrammen oder gespeicherten Prozeduren verursacht werden, aber auch massive Fehlbedienung durch Anwender ist als Ursache denkbar.

Langer Rede kurzer Sinn: Eine hochverfügbare Architektur mit gespiegelten Speichersystemen befreit Sie nicht von der Notwendigkeit, ein Datensicherungskonzept zu erarbeiten und dieses auch umzusetzen.

5.1.5 Katastrophenvorsorge

Wenn Sie Servicelevel für den Wiederanlauf eines ausgefallenen Servers und die Datenverlustzeit verabschieden, werden Sie zumeist von Störfällen ausgegangen sein, die lediglich einen einzelnen Server betreffen. Der Fall, dass Sie das ganze Rechenzentrum/Serverraum verlieren, ggf. auch noch das komplette Gebäude darum herum, ist zwar nicht sehr wahrscheinlich, er muss aber ebenfalls betrachtet werden:

Gilt die definierte Datenverlustzeit auch im Katastrophenfall? Das bedeutet, dass die Sicherung oder eine Kopie der Daten außerhalb des Gebäudes vorhanden sein muss. Bei einer Datenverlustzeit von einem Tag genügt es, aktuelle Bänder an einen auswärtigen Lagerungsort zu bringen. Soll auch im K-Fall eine Datenverlustzeit von maximal einer Stunde realisiert werden, müssen die Daten über Weitverkehrsverbindungen gesichert oder gespiegelt werden.

Weiterhin muss die zu erreichende Wiederherstellungszeit im Katastrophenfall betrachtet werden. Ist das Gebäude, in dem die Server normalerweise untergebracht sind, nicht mehr zugänglich (z. B. weil es überschwemmt oder abgebrannt ist), müssen notwendiger Weise die Server übergangsweise an einem anderen Standort aufgebaut werden. Auch wenn Sie Zugriff auf aktuelle Datensicherungen haben, benötigen Sie Hardware, auf der Sie SQL Server und andere Produkte installieren können – ganz zu schweigen von Clientsystemen, der Anbindung an das Internet und vielem anderen mehr. Eine kurzfristige Wiederherstellung der Funktionalität im Katastrophenfall ist im Normalfall nur dann möglich, wenn ein Ausweichrechenzentrum vorhanden und dementsprechend vorbereitet ist.

Viele Unternehmen legen übrigens fest, dass für einen begrenzten Störfall andere Servicelevel als beim Eintritt des Katastrophenfalls gelten. Das hängt natürlich sehr von den Anforderungen des Business-Modells des Unternehmens ab: Wer Dienstleistungen im Internet bereitstellt, kann sich sicherlich keinen tagelangen Ausfall leisten, egal welche Katastrophe passiert ist. Wer hingegen sein Geld mit der Fertigung von Bauteilen verdient, ist vermutlich nicht darauf angewiesen, dass das System zur Produktionssteuerung einsatzbereit ist – zumindest dann, wenn mit den IT-Systemen auch die Fertigungsanlagen vernichtet worden sind. In einem solchen Fall sind übrigens aber trotzdem die Kommunikationssysteme (z. B. E-Mail), die Kundendatenbank und dergleichen mehr von Interesse.

5.1.6 Genügt die Datensicherung (oder: Business Continuity vs. Disaster Recovery)?

Bei der Definition der Servicelevel werden Sie eventuell feststellen, dass Sie auf gewisse Grenzen stoßen: Eine Datenverlustzeit von maximal fünf Minuten ist mit der Datensicherung nicht zu realisieren, ebenso wenig werden Sie realistisch eine Wiederherstellungszeit von 30 Minuten erreichen können. Wenn die Anforderungen hinreichend hoch geschraubt werden, erreichen Sie also früher oder später den Punkt, an dem die Methoden der Datensicherung einfach nicht mehr ausreichen, vielmehr müssen Sie in diesem Fall auf Lösungen aus dem Themenbereich der Hochverfügbarkeit zurückgreifen. Hier stehen Ihnen dann technische Möglichkeiten zur Verfügung, um Datenverlust- und Wiederherstellungszeiten im Minutenbereich zu realisieren. Auf Neudeutsch lautet die Fragestellung, ob Sie eine Disaster-Recovery- oder eine Business-Continuity-Lösung realisieren möchten bzw. müssen. Datensicherung ist grundsätzlich im Bereich des Disaster Recoverys angesiedelt.

An dieser Stelle möchte ich nochmals betonen, dass auch die aufwendigste Business-Continuity-Lösung nicht die Datensicherung obsolet macht. Auch wenn Sie das volle Programm mit Datenbankspiegelung zwischen mehreren Standorten und dergleichen implementiert haben, könnte es doch Szenarien geben, in denen Sie die gute alte Datensicherung benötigen, beispielsweise bei logischen Fehlern.

■ 5.2 Datensicherung mit Bordmitteln

Der SQL Server stellt ausgereifte Möglichkeiten zur Verfügung, um zuverlässige Datensicherungen durchzuführen – auch ohne dass Sie teure Software hinzukaufen müssen. Ich möchte natürlich nicht verschweigen, dass professionelle Backup-Software in vielen Fällen absolut Sinn macht und Sie in größeren Umgebungen vermutlich nicht mit den Bordmitteln auskommen. Das liegt aber nicht daran, dass die vom SQL Server mitgebrachten Sicherungsmöglichkeiten in irgendeiner Form unzuverlässig wären, sondern dass die Verwaltung sehr aufwendig ist, wenn Sie viele Server betreuen und demzufolge sichern müssen.

Es ist in jedem Fall sinnvoll zu wissen, wie man mit den standardmäßig enthaltenen Mitteln Datensicherung durchführt, denn erstens kann es im Administratorleben immer wieder passieren, dass Datenbanken gesichert werden müssen, ohne dass eine Enterprise-Software zur Verfügung steht. Zweitens lernt man quasi nebenbei auch relativ viel über die Konzepte von SQL Server bezüglich der Sicherung – auch die teuerste Backup-Software greift in letzter Konsequenz auf diese Konzepte zurück.

5.2.1 Sicherungsstrategie

Das Sichern einer Datenbank im SQL Server ist simpel, allerdings müssen Sie sich im Vorfeld überlegen, welche Ziele Sie erreichen wollen (Datenverlustzeit, wie zuvor beschrieben), zudem sind einige Nebenaspekte zu betrachten, die im Grunde genommen gar nicht so



HINWEIS: Die Cloud-Integration von SQL Server 2014 dreht sich in erster Linie um Microsofts eigenes Cloud-System namens Azure. Mir ist vollkommen klar, dass die Cloud-Welt nicht nur aus Azure besteht, dieses Kapitel wird sich aber primär um SQL Server 2014 in der Azure-Welt drehen. Ähnliche Szenarien könnten mit anderen Cloud-Anbietern möglich sein.

Das Spannende im Themenbereich SQL Server 2014 und Cloud ist die relativ große Vielfalt von möglichen Szenarien. Es geht längst nicht nur darum, dass man eine SQL-Datenbank in der Cloud erzeugen und vom Webserver darauf zugreifen kann. In der Tat gibt es, aus einem funktionalen Blickwinkel betrachtet, drei Szenarien:

- Eine Datenbank wird in der Cloud betrieben und für eine eigenständige Lösung verwendet, die mit dem Rest der Unternehmens-IT nichts oder nur wenig zu tun hat.
- Die Datenbanken in der Cloud sind im weitesten Sinne eine Erweiterung der Unternehmens-IT, beispielsweise dadurch, dass bestimmte Datenbanken auf einem System in der Cloud bereitgestellt werden, aber gleichzeitig voll in den Unternehmenskontext integriert sind. Ein anderes Beispiel ist, dass Spiegel einer Datenbank in der Cloud angesiedelt sind.
- Das zweite Szenario umfasst Cloud-Dienste, die beispielsweise zur Speicherung von Sicherungen außerhalb des Unternehmens dienen.

Wir sehen im Markt derzeit rasantes Wachstum der über die Cloud bereitgestellten Möglichkeiten, momentan lassen sich die Lösungen in eines dieser drei Szenarien einordnen, was aber nicht bedeutet, dass es nicht mittelfristig auch noch ganz andere Ideen geben könnte.

■ 6.1 Azure SQL-Datenbank

Stellt man sich die Frage, warum man einen Dienst in der Cloud, in diesem Zusammenhang natürlich speziell SQL Server in der Cloud, nutzen möchte, kommt man zu einigen Argumenten:

- Niemand muss sich um die Bereitstellung des Dienstes kümmern, also keine Server-Hardware beschaffen, kein Betriebssystem installieren, keine SQL Server installieren.
- Keiner muss sich um den Betrieb des Dienstes kümmern.
- Der Betrieb der Infrastruktur muss nicht mehr betreut werden.
- Die Cloud ist extrem skalierbar.
- Die Dienste in der Cloud sind in sich redundant, d.h., die Mechanismen in der Cloud sorgen dafür, dass Datenverlustzeit und Wiederherstellungszeit annähernd bei null sind,
- ... und das Ganze ist auch noch kostengünstig.

Ob die Cloud jetzt sämtliche Wünsche, Erwartungen und Hoffnungen gleichzeitig erfüllen kann, muss zumindest diskutiert werden, aber man kommt schon relativ nah dran.

Bild 6.1 zeigt im Azure-Managementportal die Rubrik für Azure SQL-Datenbanken, vormals SQL Azure genannt. Es gibt zwei Möglichkeiten, SQL Server-Dienste in Azure bereitzustellen, die Azure SQL-Datenbanken sind die Technologie, die den zuvor formulierten Erwartungen an die Cloud am nächsten kommt.

Sie werden im Verlauf dieses Kapitels sehen, dass das Bereitstellen einer SQL Server-Datenbank noch nie so einfach und schnell vonstatten ging wie mit Azure SQL-Datenbanken – allerdings passt dieser Cloud-Dienst nicht zu allen Anforderungen.

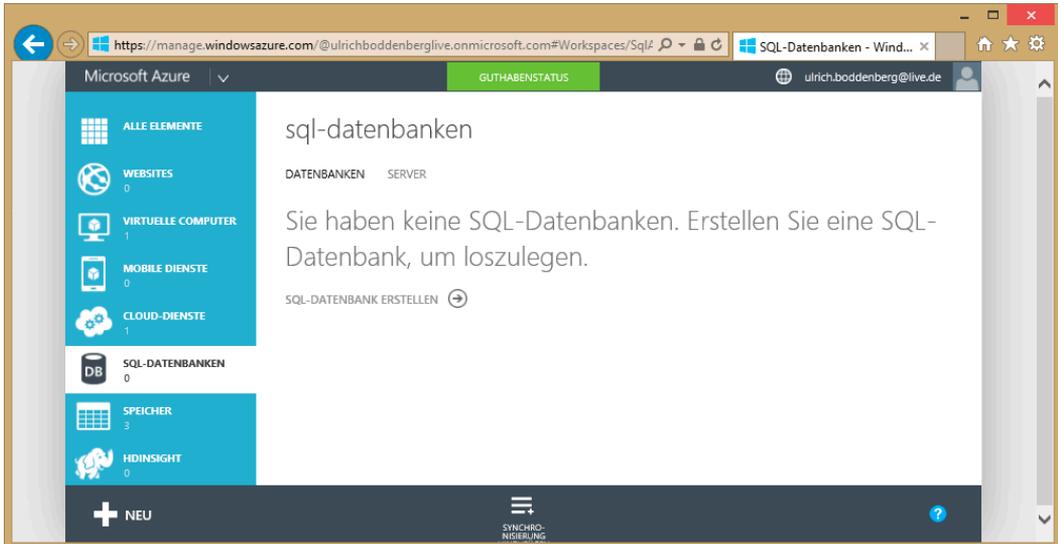


Bild 6.1 Azure SQL-Datenbanken stellen einen der vielen Dienste in der Azure-Cloud dar.

6.1.1 Abgrenzung zu SQL Server auf Azure-VM

Möchten Sie SQL Server-Datenbanken bereitstellen, gibt es prinzipiell drei Wege:

- Sie stellen die Datenbank auf einem Server bereit, der bei Ihnen im Rechenzentrum betrieben wird. Aber es ist zunächst unerheblich, ob der SQL Server von der physikalischen Maschine oder in einer virtuellen Maschine läuft. Natürlich ist das taktisch nicht unerheblich, für die hier vorgenommene Betrachtung spielt es aber keine Rolle.
- Die zweite Möglichkeit ist der Betrieb des SQL Servers auf einer virtuellen Maschine, die in der Cloud gehostet wird. Microsoft bietet zu diesem Zweck den Dienst Azure für virtuelle Computer, andere Cloud-Anbieter haben ähnliche Angebote, beispielsweise Amazon EC2 oder die Compute Engine der Google Cloud-Plattform. Im Grunde genommen ist es kein signifikanter Unterschied, ob die virtuelle Maschine in Ihrem Rechenzentrum oder in der Cloud läuft. Ein paar interessante und nennenswerte Aspekte gibt es zwar zu besprechen, aber im Grunde genommen bewegt sich alles, zumindest bezüglich des SQL Servers, in bekannten Bahnen.
- Die dritte Möglichkeit – und diese kommt dann dem eigentlichen Cloud-Gedanken am nächsten – sehen Sie in Form der Azure SQL-Datenbanken. Bei dem vorgenannten Szenario

rio arbeiten Sie nach wie vor in erster Linie mit einem Server (der jetzt zwar ein Teil der in der Cloud gehosteten virtuellen Maschine ist), der ein gewisses Maß an Administration braucht. Bei Azure SQL-Datenbanken haben Sie mit der darunter liegenden Server-Infrastruktur nichts mehr zu tun. Die Cloud stellt Ihnen eine SQL Server-Datenbank zur Verfügung – und Sie müssen sich weder um die Administration des darunter liegenden Betriebssystems kümmern noch um die des SQL Servers. Die Cloud sorgt für Ausfallsicherheit und Skalierung.

Wenn man diese Auflistung liest und es generell infrage kommt, Datenbankdienste in der Cloud bereitzustellen, kommt man natürlich schnell zu dem Ergebnis, dass Variante drei am attraktivsten ist – schließlich haben Sie im Grunde genommen nichts anderes zu tun, als den Dienst zu nutzen.

Allerdings gibt es ein paar Aspekte, die auch für die Variante zwei sprechen, wenn Sie die Cloud nutzen möchten. Insbesondere ist dies darin begründet, dass Azure SQL-Datenbanken nicht hundertprozentig kompatibel zu einem „herkömmlichen“ SQL Server sind. Der nächste Abschnitt wird sich insbesondere mit den Einschränkungen von Azure SQL-Datenbanken beschäftigen; in diesem Abschnitt möchte ich vor allem herausarbeiten, wann Sie Azure SQL-Datenbanken nutzen und wann Sie einen SQL Server verwenden, der auf einem in Azure virtuellen Computer bereitgestellt ist.

Tabelle 6.1 Gegenüberstellung eines SQL Servers auf einer virtuellen Maschine in Azure und einer Azure SQL-Datenbank

	SQL Server auf Azure-VM	Azure-Datenbank
Aufwand bei der Bereitstellung der Lösung		
Migration bestehender Anwendungen	Einfach, denn die Datenbank kann mehr oder weniger auf den auf einer virtuellen Maschine in Azure laufenden SQL Server verschoben werden	Eventuell aufwendiger. Azure basiert natürlich auf dem SQL Server, hat aber einerseits ein paar Einschränkungen, andere Aspekte sind „einfach anders“ umzusetzen.
Entwicklung neuer Anwendungen	Entspricht in etwa dem Aufwand, den Sie auch bei einem in Ihrer eigenen Umgebung betriebenen SQL Server hätten	Einfacher, sowohl bei der Bereitstellung als auch im Betrieb, weil viele Aspekte durch die Cloud übernommen werden
Kosten		
Administration der Hardware	Entfällt	Entfällt
Administration von Betriebssystem und Datenbank	Muss durch einen Administrator Ihres Unternehmens mit den entsprechenden Fachkenntnissen erledigt werden. Die Bereitstellung in der Cloud vereinfacht bzw. verbessert hier also nichts.	Entfällt
Hochverfügbarkeit der virtuellen Maschine	Automatisiert durch Azure (99,9% Uptime SLA at commercial release)	Entfällt, weil bei Azure SQL-Datenbanken keine virtuelle Maschine dediziert zugewiesen ist

	SQL Server auf Azure-VM	Azure-Datenbank
Hochverfügbarkeit der Datenbank	Kann man durch zusätzliche virtuelle Maschinen erreichen, die dann Technologien wie AlwaysOn-Verfügbarkeitsgruppen, Datenbankspiegelung oder den Transaktionsprotokollversand nutzen	Standard-Feature (99,9% DB uptime SLA)
Direkte Kosten (Geld für die Nutzung von Azure)	Die Bereitstellung von virtuellen Maschinen in Azure ist nicht ganz billig, denn Azure ist ein „Enterprise Grade“-Dienst, zudem sind mit den Preisen der virtuellen Maschinen auch die darin enthaltenen Lizenzen abgedeckt. Es darf bezweifelt werden, dass man in seinem lokalen Rechenzentrum die Preise von Azure bei realistischer Kalkulation unterbieten kann, trotzdem bleibt hier festzuhalten, dass die aufgerufenen Preise nicht unbedingt gering sind.	Eine Datenbank, die über Azure SQL-Datenbanken bereitgestellt wird, ist billiger, als wenn für diese eine Datenbank eine virtuelle Maschine in Azure bereitgestellt wird. Der Kostenpunkt ist generell günstiger, muss aber im Einzelfall geprüft werden.
Skalierung		
Scale-Up	Entsprechend der größten in Azure verfügbaren virtuellen Maschine (16 Kerne, 112 GB RAM, bis zu 16 TB Festplattenplatz – diese Werte erhöhen sich kontinuierlich)	Wird nicht angeboten
Scale-Out	Manuell beispielsweise über lesbare Replikate bei AlwaysOn-Verfügbarkeitsgruppen, Scalable Shared Databases, Distributed Partitioned Views und Data-Dependent Routing (muss alles manuell durch einen Administrator eingerichtet werden, Applikationen müssen diese Features unterstützen)	SQL Database Federation (automatisiert in der Datenschicht, Applikationen müssen Federation unterstützen)
Kontrolle, Modifikation und Anpassung		
Betriebssystem und virtuelle Maschine	Volle Kontrolle	Keine Kontrolle, keine Einflussmöglichkeiten
SQL Server-Datenbank-Kompatibilität, Customization	Alle Features von SQL Server 2014 sind verfügbar, außerdem Datenbankmodule SSIS, SSAS, SSRS	Ein großer Teil der Möglichkeiten des SQL Server-Datenbankmoduls sind verfügbar, aber nicht alle! SSIS, SSAS, SSRS stehen nicht zur Verfügung, eventuell zukünftig als separate Azure-Dienste

(Fortsetzung nächste Seite)

	SQL Server auf Azure-VM	Azure-Datenbank
Hybrider Betrieb		
Mitgliedschaft in der Domäne und Nutzung von Windows-Authentifizierung	Ja	Nicht möglich
Data Synchronization via Azure Data Sync	Möglich	Möglich
Verwaltbarkeit		
Resource Governance & Security Level	SQL-Instanz, virtuelle Maschine	Logische Datenbank-Server
Verwendbare Werkzeuge	Die bekannten SQL Server-Tools sind ohne Einschränkung verwendbar.	Die meisten SQL Server-Tools sind verwendbar, teilweise mit Einschränkungen.

Ich habe eine Weile überlegt, ob ich am Ende dieses Abschnitts versuche, die doch recht lang geratene Tabelle in einem kurzen Fazit zusammenzufassen. Das ist nicht einfach, weil die Welt erstens bunt ist und zweitens in einem tatsächlichen Projekt viele spezielle Aspekte zu berücksichtigen sind. Trotzdem möchte ich folgende Punkte nennen:

- Wenn Sie eine Anwendung entwickeln, deren Datenbank nicht in den technischen Unternehmenskontext (beispielsweise Windows-Authentifizierung) integriert werden muss, ist es eine ziemlich gute Idee, sehr genau zu schauen, ob Sie mit Azure SQL-Datenbanken arbeiten können. Die auf der ganzen Breite günstigere Kostensituation, verbunden mit den Skalierungs- und Verfügbarkeitsvorteilen eines reinrassigen Cloud-Dienstes, machen das sehr attraktiv.
- Wenn Sie auf maximale Kompatibilität zu einem „klassischen“ SQL Server angewiesen sind oder aber Features wie beispielsweise SSAS nutzen möchten, werden Sie am SQL Server auf einer virtuellen Maschine nicht herumkommen – dieser ist schlicht und ergreifend ein „klassischer“ SQL Server und verhält sich dementsprechend auch so.

6.1.2 Einschränkungen der Azure SQL-Datenbanken

In diesem Abschnitt möchte ich auf die Einschränkungen eingehen, die Azure SQL-Datenbanken gegenüber einem „normalen“ SQL Server aufweisen.



HINWEIS: Die nachfolgenden Ausführungen haben den Stand August 2014. Da die Entwicklung in der Cloud recht dynamisch voranschreitet, ist es denkbar, dass die eine oder andere Einschränkung zum Lesezeitpunkt nicht mehr existiert. Konzeptbedingt werden die Azure SQL-Datenbanken niemals komplett dem „normalen“ SQL Server entsprechen, denn das würde bedeuten, dass man etliche Vorteile, die der Dienst als „echte Cloud-Lösung“ bietet, ausfüllen bzw. abschaffen würde. Generell würde ich empfehlen, dass Sie die aktuelle Microsoft-Dokumentation konsultieren, falls sich in Ihren Überlegungen eine der nachfolgend genannten

Einschränkungen als Show-Stopper herausstellt – vielleicht wird es ja zum Leszeitpunkt in Azure SQL-Datenbanken unterstützt.

In Tabelle 6.2 sind die Funktionen aufgeführt, die für Azure SQL-Datenbanken nicht zur Verfügung stehen. Auf den ersten Blick ist die Tabelle erschreckend lang, auf den zweiten Blick werden Sie feststellen, dass es sich vor allem um diverse Zusatzfunktionen handelt, die jenseits der Basisdatenbank-Funktionalität liegen. Auch etliche Features wie beispielsweise die Datenbankspiegelung stehen nicht zur Verfügung – diese macht für Azure SQL-Datenbanken allerdings auch überhaupt keinen Sinn, weil die Datenbanken in Azure ohnehin von der Cloud redundant sind.

Tabelle 6.2 Nicht unterstützte Funktionen,

Quelle: <http://msdn.microsoft.com/de-de/library/azure/ff394115.aspx>

SQL Server-Hilfsprogramm
SQL Server PowerShell Provider. PowerShell-Skripte können auf einem lokalen Computer ausgeführt werden und eine Verbindung mit Microsoft Azure SQL-Datenbank herstellen.
Master Data Services
Change Data Capture
Datenüberwachung
Datenkomprimierung
Erweiterte Ereignisse
Erweiterung von räumlichen Typen und Methoden mittels Common Language Runtime (CLR)
Verwaltung externer Schlüssel/erweiterbare Schlüsselverwaltung
FILESTREAM-Daten
Integrierte Volltextsuche
Große benutzerdefinierte Aggregate (User-Defined Aggregates, UDAs)
Große benutzerdefinierte Typen (User-Defined Types, UDTs)
Sammlung von Leistungsdaten (Datensammler)
Richtlinienbasierte Verwaltung
Ressourcenkontrolle
SQL Server-Replikation
Transparente Datenverschlüsselung
Common Language Runtime (CLR) und benutzerdefinierte CLR-Typen
Datenbankspiegelung
Service Broker
Tabellenpartitionierung
Typisiertes XML und XML-Indizierung werden nicht unterstützt. Der XML-Datentyp wird von Microsoft Azure SQL-Datenbank unterstützt.
Sicherung und Wiederherstellung
Replikation
Erweiterte gespeicherte Prozeduren
SQL Server-Agent/-Aufträge

Tabelle 6.3 zeigt eine Auflistung der unterstützten T-SQL-Funktion, wobei Sie hier im Wesentlichen alle Features, die man für den Zugriff auf Datenbanken nun mal braucht, finden. Wenn Sie also eine Datenbank vom SQL Server auf Azure SQL-Datenbanken importieren, sollte dies ohne größere Änderungen in der Applikation zu machen sein – zumindest wenn Sie Standard-T-SQL nutzen.

An dieser Stelle ein kleiner Hinweis für Entwickler: ORM-Tools bieten im Allgemeinen spezielle Modi für Azure SQL-Datenbanken an – zumindest sollten sie das mittlerweile ...

Tabelle 6.3 Unterstützte und teilweise unterstützte T-SQL-Funktionen,
Quelle: <http://msdn.microsoft.com/de-de/library/azure/ee336250.aspx>

Konstanten
Einschränkungen
Cursor
Erweiterung von räumlichen Datentypen und Methoden per CLR
Indexverwaltung und Neuerstellung von Indizes
Lokale temporäre Tabellen
Reservierte Schlüsselwörter
Räumliche Daten und Indizes
Gespeicherte Prozeduren
Statistikverwaltung
Transaktionen
Trigger
Tabellen, Joins und Tabellenvariablen
Transact-SQL-Sprachelemente; Beispiele:
▪ Erstellen/Löschen von Datenbanken
▪ Erstellen/Ändern/Löschen von Tabellen
▪ Erstellen/Ändern/Löschen von Benutzern und Anmeldungen
Benutzerdefinierte Funktionen
Sichten

Tabelle 6.4 listet die nicht unterstützten T-SQL-Funktionen auf. Zumeist sind dies administrative Funktionen, was so weit einleuchtend ist, da sie so „tief“ mit Azure SQL gar nicht zur Datenbank bzw. zum Datenbankserver kommen. Ansonsten werden verteilte Abfragen und verteilte Transaktionen nicht unterstützt, was auch einleuchtend ist, weil mit Azure SQL-Datenbanken die einzelne Datenbank stets für sich alleine steht. Verbindungen jedweder Art zu anderen Datenbanken in Azure SQL sind nicht möglich.

Tabelle 6.4 Nicht unterstützte T-SQL-Funktionen,

Quelle: <http://msdn.microsoft.com/de-de/library/azure/ee336250.aspx>

Common Language Runtime (CLR)
Platzierung von Datenbankdateien
Datenbankspiegelung
Verteilte Abfragen
Verteilte Transaktionen
Dateigruppenverwaltung
Globale temporäre Tabellen
SQL Server-Konfigurationsoptionen
SQL Server Service Broker
Systemtabellen
Ablaufverfolgungsflags

6.1.3 Azure SQL-Datenbank anlegen

Um eine Azure SQL-Datenbank anzulegen, klicken Sie im Azure-Managementportal auf das Pluszeichen links unten (siehe Bild 6.2), wodurch Sie zu dem in Bild 6.2 dargestellten Menü gelangen. Dort wählen Sie das Anlegen einer neuen SQL-Datenbank und entscheiden sich für das benutzerdefinierte Erstellen.

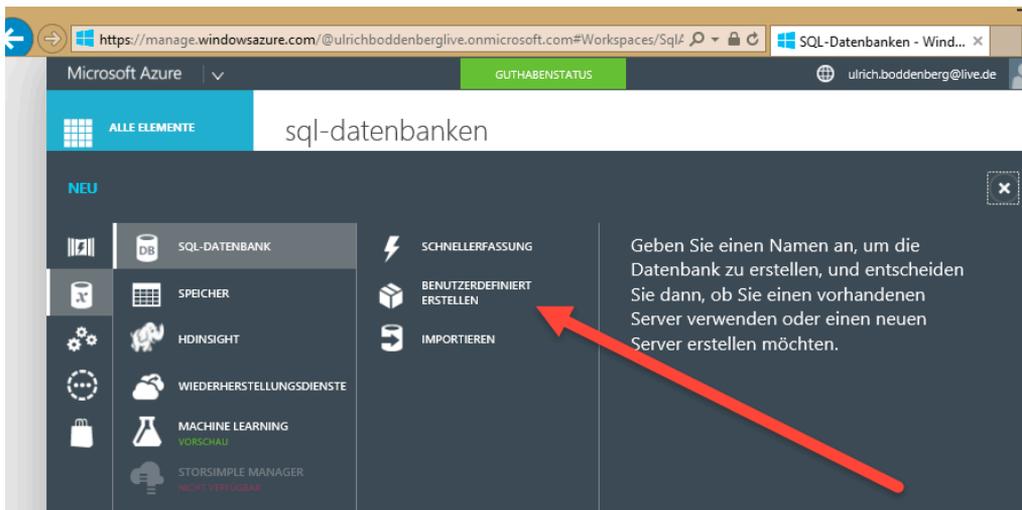


Bild 6.2 Das Anlegen einer neuen Azure SQL-Datenbank startet im Azure-Managementportal.

Das Erstellen der neuen Azure SQL-Datenbank findet in einem einfachen Dialog statt, der in Bild 6.3 zu sehen ist:

- Notwendigerweise erhält die Datenbank einen Namen, über den sie später auch angesprochen wird.

- Sie wählen ein Abonnement, über das die Abrechnung läuft. Es ist möglich, dass Ihrem Konto mehrere Abonnements zugeordnet sind, daher macht diese Auswahlmöglichkeit Sinn.
- Der nächste Punkt ist die Auswahl der Edition. Zum Zeitpunkt, in dem ich diesen Absatz schreibe (August 2014), gibt es die Editionen Web und Business. Microsoft hat bereits angekündigt, dass es im April 2015 drei neue Editionen geben wird, die etwas anders konfiguriert werden können, dort gibt es zum Beispiel Leistungsstufen. Die neuen Editionen kann man sich bereits heute als Vorschau-Feature ansehen, was ich ein wenig später beschreibe. Im Moment unterscheiden sich die Editionen in der maximalen Datenbankgröße.
- Wenn Sie die Edition ausgewählt haben, können Sie die maximale Größe der Datenbank festlegen. In der Business-Edition ist der Maximalwert im Moment 150 Gigabyte. In der zukünftigen Premium-Edition wird eine Azure SQL-Datenbank bis zu 500 Gigabyte groß sein können. Zu beachten ist, dass Microsoft nur die tatsächliche Größe der Datenbanken in Rechnung stellt und nicht die eingestellte Maximalgröße. Ich möchte hier aber deutlich sagen, dass die Art und Weise der Abrechnung sich auch ändern kann – Sie sollten also die Preisliste konsultieren, bevor Sie Ihr gesamtes weiteres berufliches Schicksal an diese Zeile hängen.
- Die nächste Einstellung, nämlich die Sortierung, ist aus dem normalen SQL Server hinlänglich bekannt.
- Interessant ist die letzte Auswahlmöglichkeit des Dialogs, hier geht es um den zu verwendenden Server. Das wird auf den ersten Blick etwas merkwürdig anmuten, denn ich habe ja zuvor lang und breit geschildert, dass Sie mit der Administration eines Servers bei der Nutzung der Azure SQL-Datenbanken nichts zu tun haben. Wie Sie gleich sehen werden, brauchen Sie bei dem „Server“ auch nicht mehr einzustellen als Anmeldeinformationen und den gewünschten Rechenzentrums-Standort. Sehen Sie den hier einzutragenden Server also eher als virtuellen Container denn als etwas, an dem Sie Administrationsarbeiten leisten müssen.

NEUE SQL-DATENBANK – BENUTZERDEFINIERT ERSTELLEN

Datenbankeinstellungen angeben

NAME
sqlbook1

ABONNEMENT
Windows Azure MSDN ? Visual Studio Premiurr

EDITION
WEB BUSINESS

MAXIMALE GRÖÖE
150 GB

SORTIERUNG
SQL_Latin1_General_CP1_CI_AS

SERVER
Neuer SQL-Datenbankserver

Bild 6.3

Anlegen einer neuen Datenbank. Wenn Sie die erste Datenbank anlegen, müssen Sie einen neuen SQL-Datenbankserver erstellen.

Bild 6.4 zeigt den Dialog zum Erstellen eines SQL-Datenbanksservers für Azure SQL-Datenbanken. Sie sehen, dass ich Ihnen nicht zu viel bzw. zu wenig versprochen habe, viel ist hier wirklich nicht zu konfigurieren:

- Anmeldenname und Anmeldekennwort ist, denke ich, selbsterklärend. Sie benötigen diese Informationen, wenn Sie auf diesem Server beispielsweise mit SQL Server Management Studio zugreifen möchten. Weitere Benutzerkonten können dann später angelegt werden.
- Obligatorisch ist die Auswahl einer Region, also des Azure-Rechenzentrums, in dem dieser Server nebst seiner Datenbanken betrieben werden soll. Es kommen immer mal wieder Rechenzentren hinzu, derzeit kommen für europäische Benutzer vor allem die Rechenzentren Nordeuropa (Dublin) und Westeuropa (Amsterdam) infrage. Es hindert Sie aber auch niemand daran, beispielsweise Brasilien auszuwählen.
- In vielen Fällen kreieren Sie die Datenbank auch deshalb, weil Sie mit anderen Azure-Diensten arbeiten möchten und diese eine Datenbank benötigen. Setzen Sie dann bitte die entsprechende Checkbox.

SERVER ERSTELLEN

SQL-Datenbankservereinstellungen

ANMELDENNAME
uboddenberg

ANMELDEKENNWORT
●●●●●●

KENNWORT BESTÄTIGEN
●●●●●●

REGION
Nordeuropa

ERMÖGLICHEN SIE WINDOWS AZURE-DIENSTEN DEN ZUGRIFF AUF DEN SERVER.

Bild 6.4 Erstellen eines SQL-Datenbanksservers für Azure SQL

Sie können übrigens für jede Datenbank einen separaten Datenbankserver anlegen, das macht kostentechnisch keinen Unterschied, da die Datenbanken Kosten verursachen und nicht diese „Pseudo-Server“. Sie werden im weiteren Verlauf dieses Kapitels sehen, dass es schon ein wenig mehr Konfigurationsoptionen für diesen Datenbankserver gibt, als im Dialog zur Einrichtung (Bild 6.4) zu sehen ist, beispielsweise die für die Administration zulässigen IP-Adressen. Auch dies könnte ein Kriterium sein, um zusätzliche Datenbankserver anzulegen. Andererseits gilt wie immer (und eben auch hier): Viele Server verursachen viel administrativen Aufwand.



HINWEIS: Das Innovationstempo in der Cloud ist recht hoch, das Microsoft-Azure-Team baut kontinuierlich neue Dienste in das Angebot ein. Teilweise ändert sich auch etwas an den Diensten, so ist beispielsweise angekündigt, dass für Azure SQL-Datenbanken die Editionen Web und Business entfallen und stattdessen die Editionen Basic, Standard und Premium erhältlich sind. Zur Zeit der Abfassung dieses Kapitels haben diese Editionen den Status einer Preview. Wenn Sie sie anschauen und damit testen möchten, müssen Sie das entsprechende Vorschau-Feature aktivieren. Ich zeige Ihnen, wie das gemacht wird, denn das ist sicher auch für andere Verwendungszwecke interessant – für Datenbankanwendungen gibt es zurzeit (August 2014) noch ein weiteres interessantes Feature, nämlich das Auditing, das auch über die Vorschau aktiviert werden kann.

Navigieren Sie zu <http://azure.microsoft.com/en-us/services/preview/> und suchen das zu aktivierende Vorschau-Feature heraus – das sieht dann so aus wie in Bild 6.5 gezeigt.

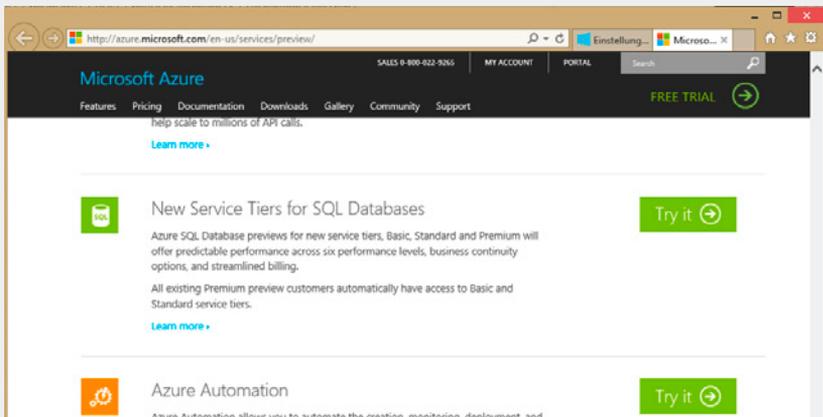


Bild 6.5 Die Liste der Vorschau-Features, die Sie Ihrem Azure-Abonnement hinzufügen können

Nach einer kurzen Weile werden Sie aufgefordert, sich anzumelden, und dann erhalten Sie den in Bild 6.6 gezeigten Dialog zur Auswahl des Azure-Abonnements, dem das Vorschau-Feature zugeordnet werden soll. Hintergrund: Dem Konto könnten durchaus mehrere Abonnements zugeordnet sein.

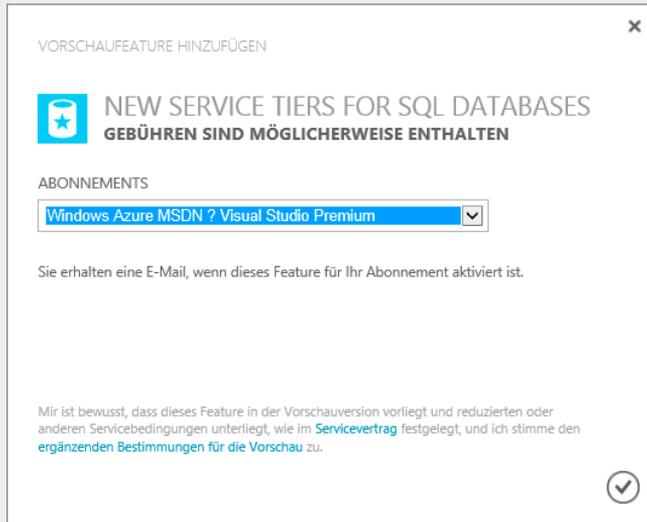


Bild 6.6 Wählen Sie das Abonnement aus, dem Sie das Feature hinzufügen möchten.

Wenn Sie nochmals einen Blick in die Liste der Vorschau-Features werfen, werden Sie sehen, dass das Feature als aktiv angezeigt wird (Bild 6.7).

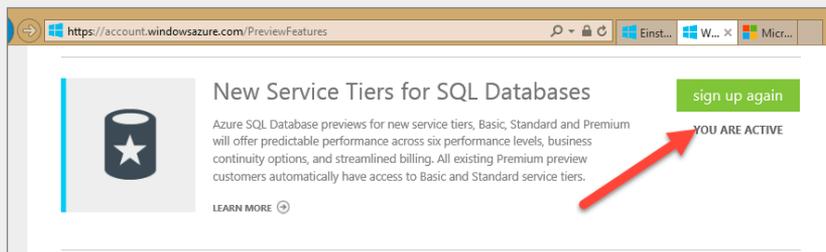


Bild 6.7 Nun wird in der Liste angezeigt, dass das Vorschau-Feature aktiv ist.

Rufen Sie nun das Erstellen einer Azure SQL-Datenbank auf, werden Sie sehen, dass Vorschau-Editionen angezeigt werden. In Bild 6.8 habe ich eine der Vorschau-Editionen ausgewählt, wie Sie sehen, gibt es nun eine zusätzliche Einstellmöglichkeit (Leistungsebene) und außerdem kann die maximale Größe auf 500 Gigabyte gestellt werden.

Bild 6.8 In den Datenbankeinstellungen können Sie jetzt auch eine Datenbank der Vorschau-Edition anlegen – da gibt es übrigens auch andere Konfigurationsmöglichkeiten.

Wenn Sie diese Zeilen nach April 2015 lesen, handelt es sich bei den hier als Vorschau-Editionen sichtbaren SQL-Datenbank-Kategorien vermutlich um die Standardeinstellungen. Vielleicht gibt es dann aber auch schon etwas Neues, wie gesagt, die Innovationszyklen bei Cloud-Diensten sind sehr kurz.

6.1.4 Administration des Datenbankservers

Vielleicht erscheint Ihnen die Überschrift dieses Kapitels etwas merkwürdig, denn ich habe ja zuvor lang und breit ausgeführt, dass es bei den Azure SQL-Datenbanken eben keinen Server gibt, der administriert werden muss. Im Grunde genommen ist dem auch so, es gibt aber einige Einstellungen für Azure-Datenbanken, die unterhalb des Konstrukts „Server“ zusammengefasst sind.

Wie in Bild 6.9 zu sehen, gibt es eine Registerkarte **SERVER**, die zu einer Auflistung der im Rahmen der Datenbankanlage erstellten Server führt. Diese Server haben automatisch generierte Namen, sind daher nicht allzu intuitiv, was aber auch kein Problem darstellt – Sie werden diese Servernamen nicht allzu häufig aufrufen müssen ...

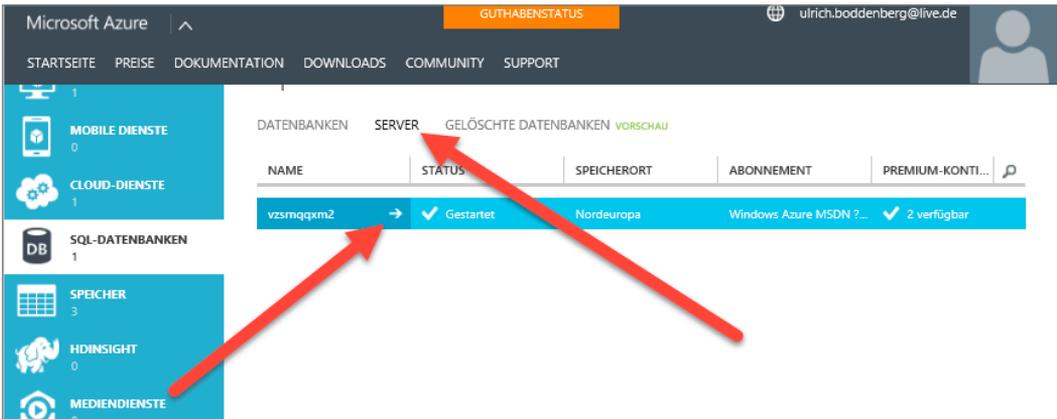


Bild 6.9 In der Rubrik SQL-Datenbanken gibt es auch die Registerkarte Server.

Bild 6.10 zeigt die Konfigurations-Webseite des Servers. Es gibt hier einige Registerkarten, die ich nun nicht im Detail besprechen möchte – ich würde Ihnen aber empfehlen, einfach mal darüber zu schauen. Interessant ist allerdings der Inhalt der auf dem Bild gezeigten Registerkarte **KONFIGURIEREN**, denn diese Einstellung ist einigermaßen entscheidend für den weiteren Erfolg Ihrer Arbeit mit den Azure SQL-Datenbanken. Sie tragen hier die IP-Adressen ein, die Zugriff auf diesen Datenbankserver, nebst der darauf laufenden Datenbanken, haben sollen. Wie Sie sehen, ist nicht nur die Eingabe einzelner IP-Adressen, sondern die Festlegung von Bereichen möglich.

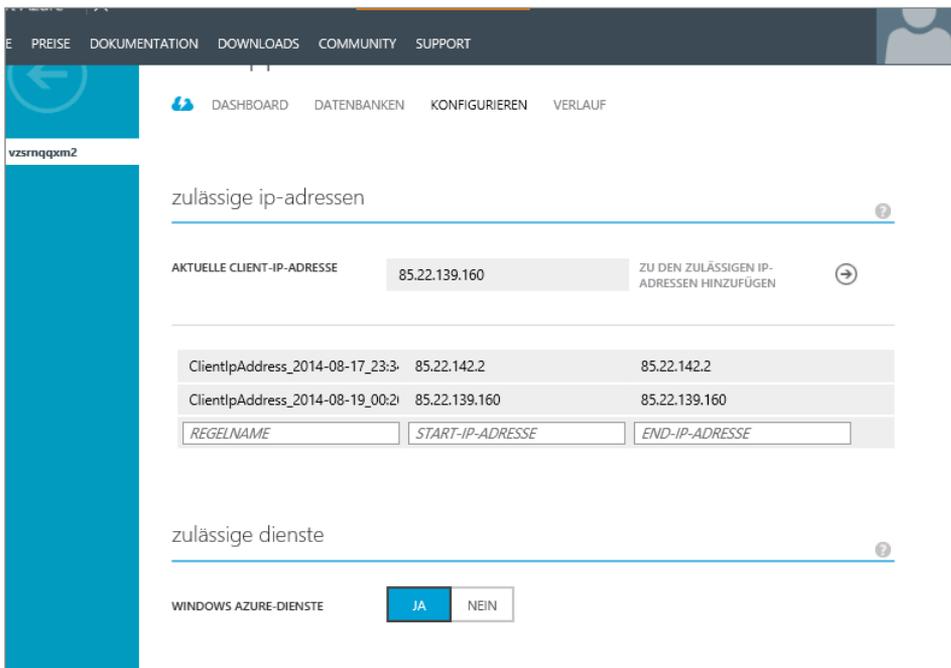


Bild 6.10 Hier werden die IP-Adressen bzw. Adressbereiche, die zum Zugriff berechtigt sind, eingetragen.

Weiterhin sehen Sie die Einstellmöglichkeit, ob Azure-Dienste Zugriff erhalten sollen. Das wurde bereits vom Assistenten zum Anlegen des Datenbankservers abgefragt, die damals getroffene Entscheidung kann also hier bei Bedarf revidiert werden.



HINWEIS: Normalerweise sollte es im professionell genutzten Umfeld keinen Fall geben, in dem von Datenbank-Clients mit dynamischen IP-Adressen auf den Azure SQL-Datenbankserver zugegriffen werden muss. Ich gehe davon aus, dass moderne Anwendungen mindestens dreischichtig sind, also die Clients zunächst auf eine Datenzugriffsschicht zugreifen und diese dann auf den Datenbankserver zugreift. Auch wenn die Clients mit dynamischen IP-Adressen ausgestattet sind, sollte die Datenzugriffsschicht (beispielsweise ein Webservice) eine statische IP-Adresse haben, die dann im Azure SQL-Datenbankserver wie in Bild 6.10 gezeigt eingetragen werden kann.

Falls Sie – beispielsweise in einem Testszenario – mit dynamischen IP-Adressen arbeiten müssen, wäre es eine Möglichkeit, den kompletten infrage kommenden IP-Bereich, den der Provider dynamisch vergibt, einzutragen. Das ist zwar nicht perfekt, eine andere Wahl wird Ihnen allerdings nicht bleiben.

Wenn Sie auf die Startseite des Datenbankservers (dahin gelangen Sie beim Aufruf des Datenbankservers) schauen, werden Sie einen Link namens *Server verwalten* finden (Bild 6.11). Dieser Link führt zu einer Silverlight-basierten Applikation, die eine detaillierte Verwaltung des SQL Servers und der Datenbanken ermöglicht. Diese Applikation erfordert, wie in Bild 6.12 gezeigt, eine Anmeldung mit den Credentials, die Sie beim Anlegen des Datenbankservers angegeben haben, siehe Bild 6.4.

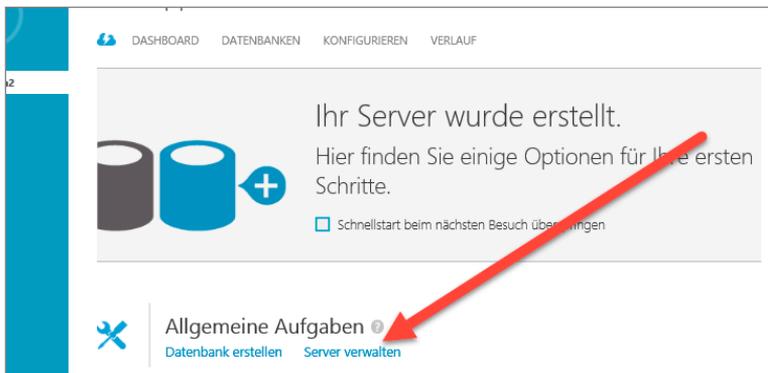


Bild 6.11 Auf der Startseite findet sich ein Link zum weitergehenden Verwalten des Servers.

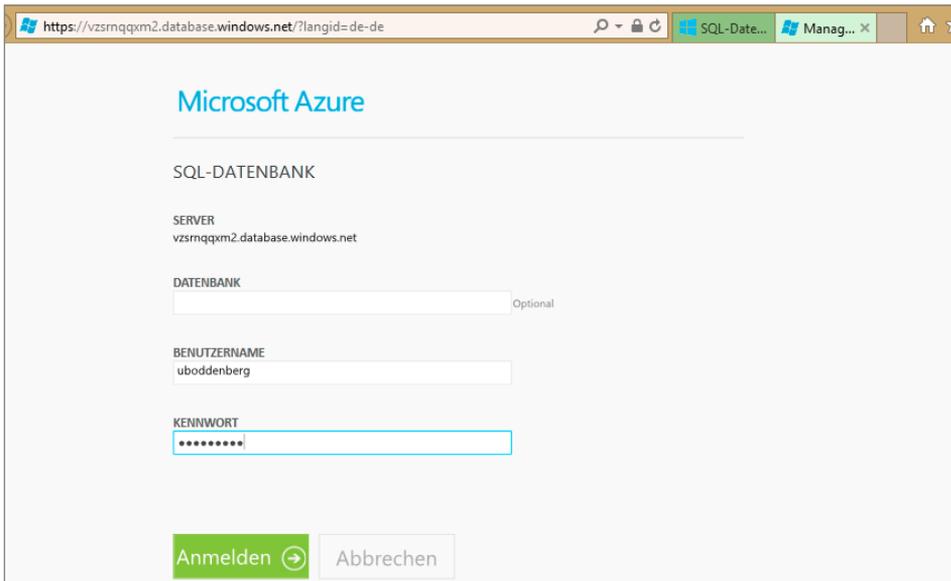


Bild 6.12 Die Anmeldeseite der Serververwaltung

Bild 6.13 zeigt einen Blick auf die Verwaltungsapplikation, in der Sie beispielsweise Datenbanken anlegen und diese sehen sowie diverse andere datenbankbezogene Aufgaben ausführen können. Diese kleine Anwendung ist nun zugegebenermaßen nicht die Krönung einer Ergonomie und Sie haben auch nicht die Möglichkeiten eines SQL Management Studios mit einem lokalen SQL Server – dennoch ist sie zur Durchführung der üblichen Standardaufgaben ganz hilfreich.

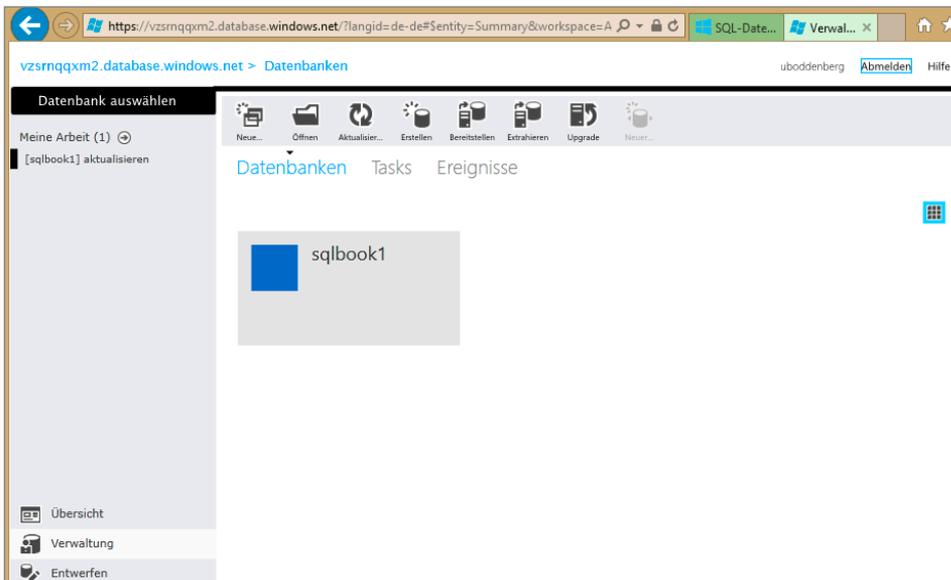


Bild 6.13 In der Verwaltung können beispielsweise Datenbanken angelegt werden.



HINWEIS: Eine Silverlight-Applikation läuft in Ihrem Browser, also auf Ihrem Client. Demzufolge muss die IP-Adresse des Clients in der Liste der zulässigen Adressen eingetragen sein, siehe Bild 6.10.

Als SQL-Administratoren sind wir seit Generationen des Produktes gewöhnt, mit dem SQL Server Management Studio zu arbeiten, weshalb natürlich die Frage nahelegt, ob diese Möglichkeit mit den Azure SQL-Datenbanken besteht. Die Antwort ist, wie so häufig, das klassische „Ja, aber“: Sie können mit dem SQL Management Studio eine Verbindung zu einem Azure SQL-Datenbankserver, nebst der darauf vorhandenen Datenbanken, aufbauen. Sie werden aber feststellen, dass Sie dramatisch weniger Möglichkeiten zur Administration haben werden. Zum einen entfallen sämtliche Konfigurationsmöglichkeiten, die auf die eigentliche Instanz bezogen sind, weil Sie eben bei Azure SQL-Datenbanken auf selbige keinen Zugriff haben – jedenfalls nicht administrativ. Zum anderen werden Sie feststellen, dass Sie auch bei der Administration der Datenbank beispielsweise die grafischen Design-Werkzeuge nicht zur Verfügung haben und stattdessen mit SQL-Befehlen arbeiten müssen. Das ist jetzt keine Katastrophe – sollte es jedenfalls für einen SQL-Administrator nicht sein, aber schon ein Verlust an Komfort.

Wenn Sie also mit dem SQL Server Management Studio eine Verbindung zu einem Azure SQL-Datenbankserver aufbauen möchten, tragen Sie dessen Namen in den Verbindungsaufbau-Dialog ein und geben dort weiterhin die Anmeldeinformationen, die Sie beim Anlegen des Datenbankservers vorgegeben haben (siehe Bild 6.4), an. Wenn die IP-Adresse des Computers, mit dem Sie zugreifen, in der Liste der zulässigen Adressen eingetragen ist, wird einige Augenblicke später der Server im Management Studio geöffnet sein (Bild 6.14).

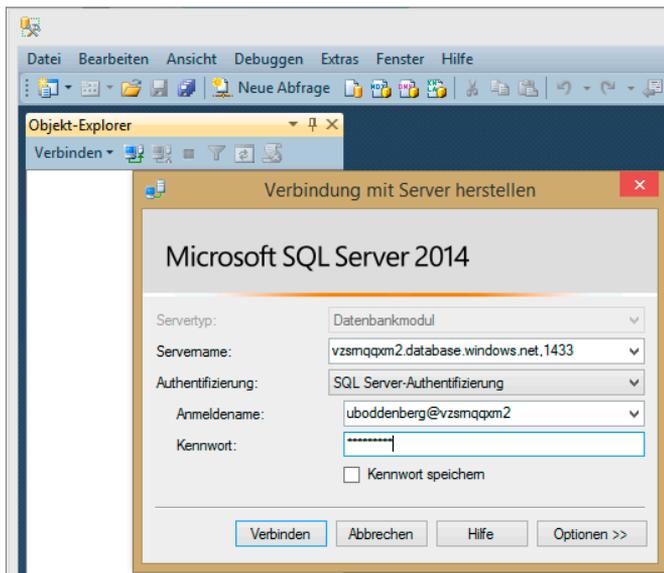


Bild 6.14 Verbindung herstellen mit dem SQL Server Management Studio



HINWEIS: Es wäre nun natürlich blöd, wenn man, um Azure SQL-Datenbanken zu administrieren, auf einem SQL Server angemeldet sein müsste. Daher gibt es das SQL Server Management Studio auch „einzeln“, also ohne SQL Server. Wenn Sie von einem vollständigen SQL Server-Installationsmedium versuchen, nur das Management Studio zu installieren, wird das nicht gehen. Es mag sein, dass es dafür den einen oder anderen „Hack“ gibt. Einfacher ist aber, wenn Sie bei Microsoft ein Installationspaket herunterladen, das nur das Management Studio enthält. Sie finden das in Ihrem Download-Portal (VLSC oder MSDN).

Bild 6.15 zeigt Ihnen den Azure SQL-Datenbankserver im SQL Server Management Studio. Sie werden sehen, dass es viele Aspekte, die sonst angezeigt werden, nicht gibt, um nur einige zu nennen: Es gibt im Bereich Systemdatenbanken nur die Master-Datenbank, es gibt keinen SQL Server Agent und unterhalb des Knotens Verwaltung findet sich kein einziger Eintrag. Das ist alles so korrekt, denn unter Azure SQL werden die fehlenden Komponenten bzw. Administrationsmöglichkeiten auch nicht gebraucht. Das ist ja gerade der Vorteil der cloud-basierten Lösung, dass der Cloud-Anbieter sich für Sie über diese Dinge seinen Kopf zerbricht.

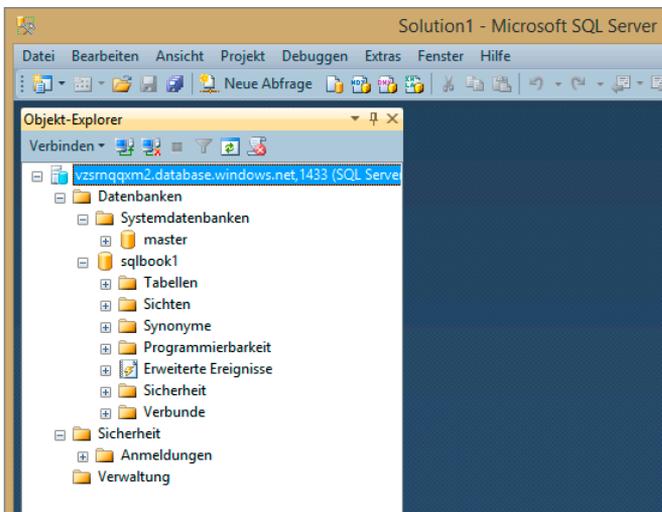


Bild 6.15

Zugriffe auf den Azure SQL-Datenbankserver mit dem Management Studio

Wollen Sie im SQL Server Management Studio in die Administration und Modifikation einer Datenbank einsteigen, werden Sie sehen, dass hier nicht die üblichen grafischen Werkzeuge angeboten werden, sondern ein exemplarischer SQL-Befehl, den Sie mit Ihren Inhalten füllen können bzw. müssen. Das ist für jemanden, der mit SQL ein wenig Erfahrung hat, nicht wirklich ein Problem. Sie sollten sich jedoch darüber schon im Klaren sein, dass man mit Azure SQL-Datenbanken und dem Management Studio nicht den gewohnten Komfort zur Verfügung hat (Bild 6.16).

Wenn man das im Gesamtzusammenhang bewerten muss, halte ich das aber nicht für problematisch, denn die Zeit, die man beim Betrieb einer SQL-Datenbank auf Azure mit Administrationsaufgaben verbringt, ist dramatisch geringer als beim konventionell bereitgestellten SQL Server – das ist auch genau der Sinn einer Cloud-Lösung.

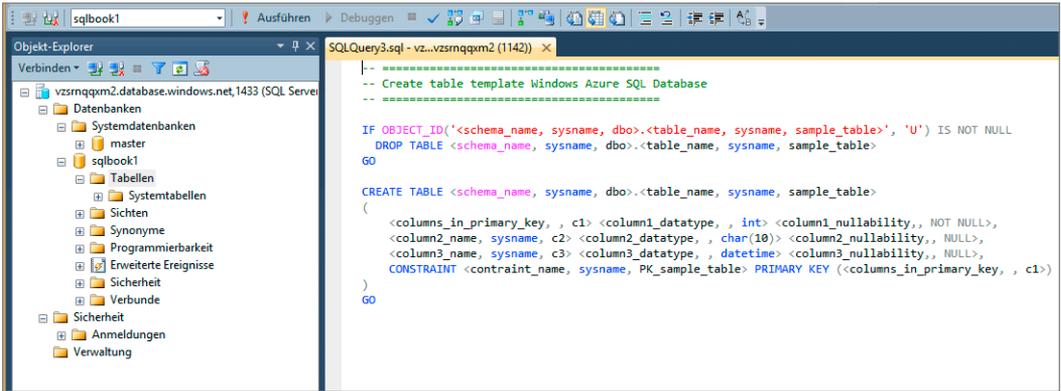


Bild 6.16 Zum Anlegen einer neuen Tabelle müssen Sie das vordefinierte SQL-Statement anpassen.

6.1.5 Datenbank in Azure SQL bereitstellen

In den meisten Fällen werden Datenbanken zunächst in der lokalen Umgebung erstellt und dann als Azure SQL-Datenbank bereitgestellt. Das SQL Management Studio bietet hierfür eine angenehme Funktionalität, nämlich den Befehl **DATENBANK AUF WINDOWS AZURE SQL-DATENBANK BEREITSTELLEN**, den Sie im Kontextmenü einer Datenbank finden (Bild 6.17). Sie können also beispielsweise als Entwickler mit dem lokalen Server in der Entwicklungsumgebung hier Datenbankdesign durchführen und transportieren dann bequem mit wenigen Mausklicks die Datenbank in die Cloud.

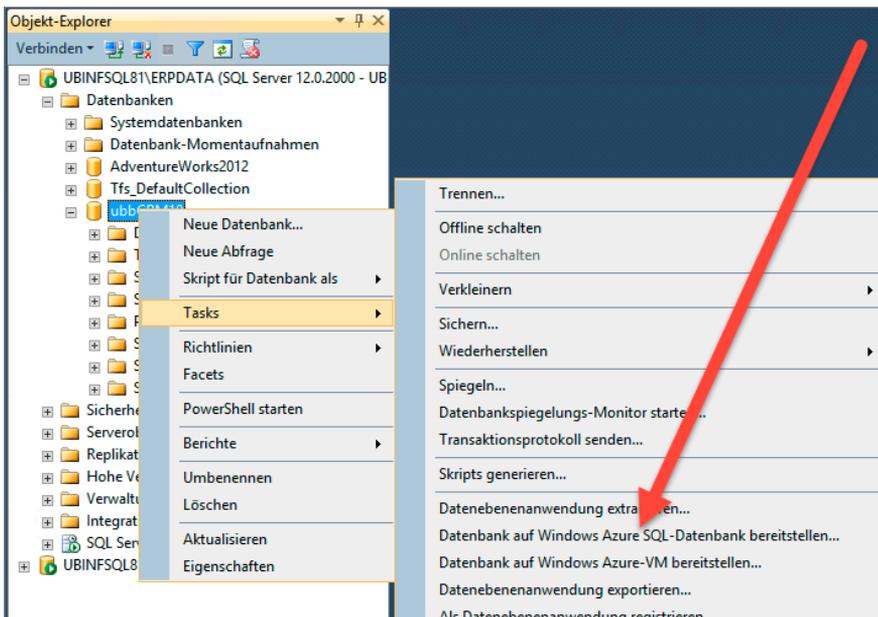


Bild 6.17 Eine vorhandene Datenbank kann mittels dieses Befehls auf einer SQL-Datenbank bereitgestellt werden.

Bild 6.18 zeigt den ersten Dialog des Assistenten zum Veröffentlichen der Datenbank in der Cloud:

- Zunächst geben Sie die Parameter der Serververbindung an, dazu erscheint nach Klick auf den Schalter **VERBINDEN** der übliche Dialog zum Eingeben der Credentials zum Verbindungsaufbau mit einem SQL Server, gemeint ist hier der Azure SQL-Datenbankserver (siehe auch Bild 6.14). Falls Sie bisher noch keinen Datenbankserver auf Azure angelegt haben, müssen Sie das zunächst erledigen. Das geschieht in der Weboberfläche von Azure wie zuvor gezeigt.
- Dann können Sie einen beliebigen Namen für die Datenbank in Azure vorgeben.
- Da eine neue Datenbank in Azure angelegt wird, müssen Sie die Edition um die Datenbankgröße wählen. Bei den neuen Editionen (Basic, Standard, Premium) gibt es noch andere Parameter als nur die maximale Datenbankgröße.
- Da die Datenbank während des Anmeldevorgangs zwischengespeichert wird, müssen Sie einen temporären Speicherort mit genügend Platz angeben.

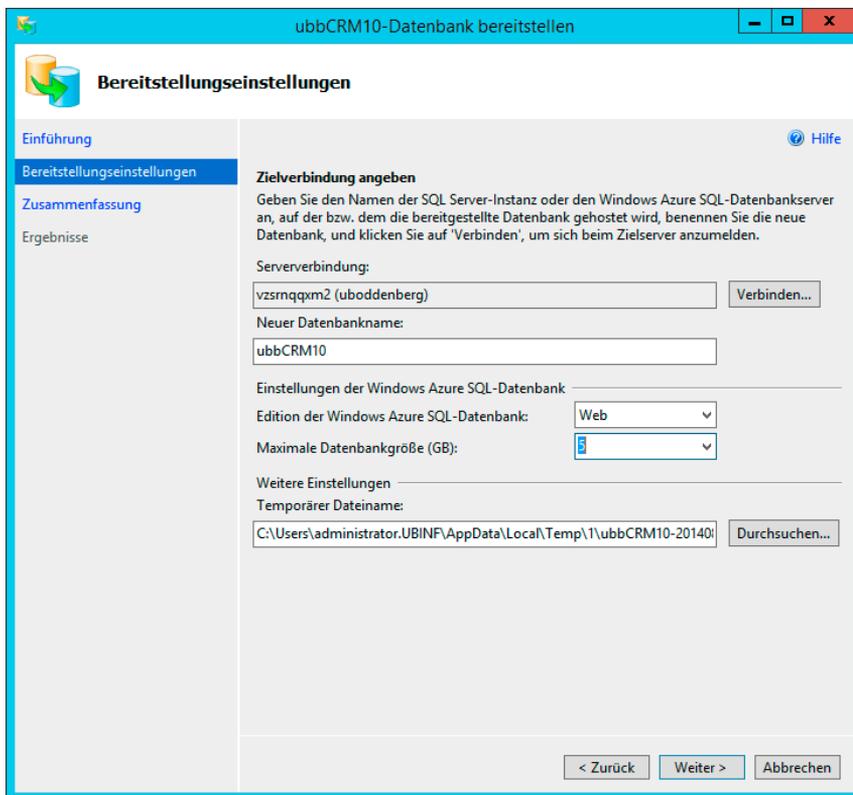


Bild 6.18 Im Assistenten wird unter anderem die Edition der Datenbank auf Azure ausgewählt.

Sobald Sie alle Eingaben gemacht haben, beginnt der Assistent zu arbeiten. Wie Sie in Bild 6.19 sehen können, erledigt er eine Menge Aufgaben. Die Ausführungsdauer wird aber naturgemäß hauptsächlich von der Größe der Datenbank und der Upload-Zeit zu Azure bestimmt.

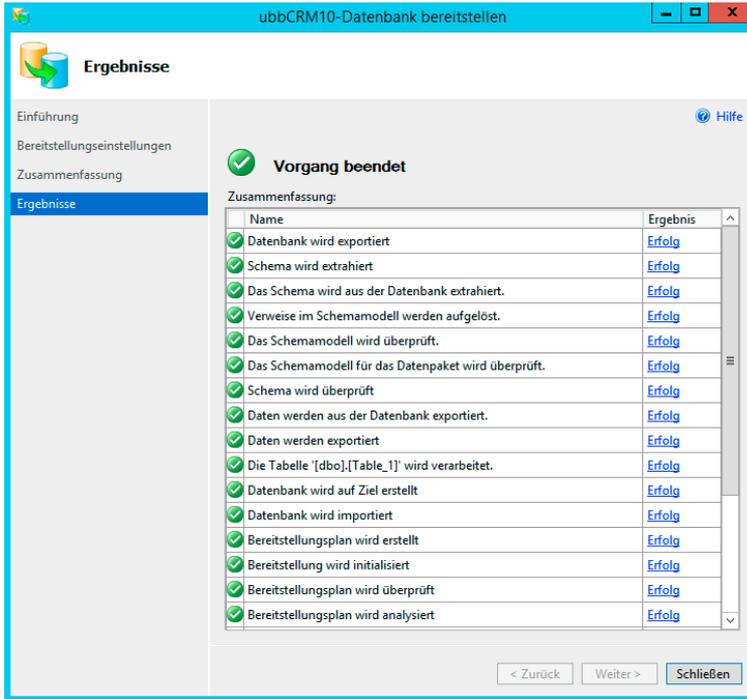


Bild 6.19 Diverse Schritte werden vom Assistenten ausgeführt.

Bild 6.20 zeigt das Ergebnis der Bemühungen in der Azure Verwaltungsoberfläche: Die Datenbank ist in der Liste vorhanden und ist einem Server und einer Edition zugeordnet. Der Speicherort (auf der Abbildung Nordeuropa) ist durch den Datenbankserver, dem die Datenbank zugeordnet ist, vorgegeben.

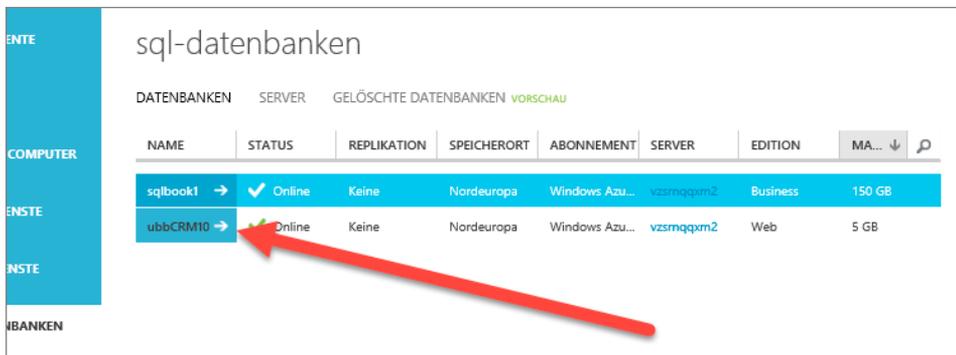


Bild 6.20 Die Datenbank ist nun in Azure vorhanden und kann verwendet werden.



HINWEIS: Ich habe Ihnen zuvor das Beispiel einer Bereitstellung einer vorhandenen Datenbank zu Azure SQL-Datenbanken in einer funktionierenden Form gezeigt. Weiter vorn in diesem Kapitel habe ich einige Einschränkungen aufgelistet, die Azure SQL-Datenbanken gegenüber einem normalen SQL Server haben. Wenn Sie eine Datenbank versuchen bereitzustellen, die eine oder mehrere dieser nicht unterstützten Features verwendet, wird der Assistent mit einem oder mehreren Fehlern abbrechen. Das sieht dann etwa so aus, wie in Bild 6.21 gezeigt.

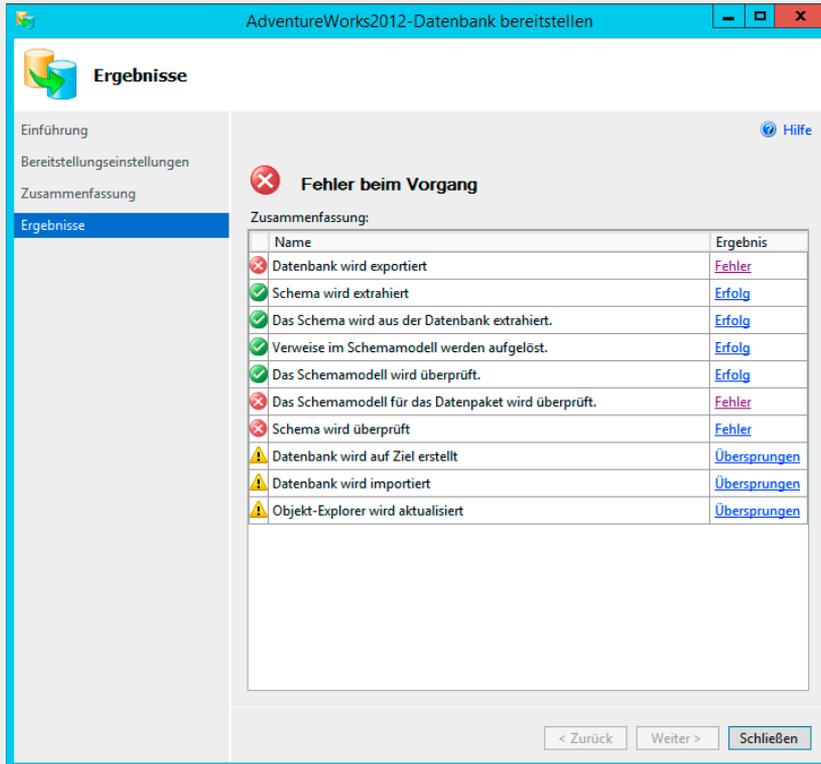


Bild 6.21 So sieht es aus, wenn die Übertragung der Datenbank nach Azure nicht funktioniert.

Glücklicherweise können die aufgetretenen Fehler einfach durch Mausklick aufgerufen werden. Meiner Erfahrung nach sind die Meldungen zumeist sogar einigermaßen aussagekräftig. Leider ist es so, dass die Kenntnis, warum ein Fehler aufgetreten ist, nicht unbedingt dabei hilft, die Ursache einfach und schnell zu beheben. Insofern bleibt die Erkenntnis, dass nicht jede Datenbank problemlos nach Azure SQL-Datenbanken zu transportieren ist. In der Praxis ist das aber nicht wirklich so ein riesiges Problem, dafür gibt es zwei Gründe:

- Möchten Sie eine bestehende Datenbank in der Cloud bereitstellen, diese aber aufgrund der Einschränkungen von Azure SQL-Datenbanken dort nicht bereitgestellt werden kann, bleibt Ihnen auf jeden Fall der Weg über eine virtuelle Maschine in Azure mit SQL Server. Da das ein „normaler“ SQL Server ist, gibt es dabei keine Einschränkungen. Verfahren und Vorgehensweisen stelle ich Ihnen ein wenig später in diesem Buch vor.
- Und vor allem: Eine cloud-basierte Lösung besteht im Normalfall immer aus mehreren Komponenten: der Datenbank, der Datenzugriffsschicht, einem oder mehreren Applikationsservern, einem oder mehreren Webservern. Vermutlich wird eine seit 20 Jahren bestehende Lösung, die für die Bereitstellung auf lokalen Servern im Unternehmen gedacht war, ohnehin nicht als echte Cloud-Lösung verwendet werden können – zumindest nicht ohne größere „Umbauarbeiten“. Insofern stellt sich die Frage des Übernehmens einer bestehenden Datenbank, die nichtunterstützte Features verwendet, vermutlich ohnehin nicht allzu oft. Die zuvor gezeigte Management Studio-Funktionalität ist aber trotzdem spannend, eben um speziell für die Cloud-Verwendung entwickelte Datenbanken bereitzustellen, also vom Entwicklungs-SQL-Server in die Cloud. Und wenn eine Datenbank genau für die Verwendung in der Cloud entwickelt wird, wird sie natürlich auch nicht nichtunterstützte Features aufweisen.

6.1.6 Nutzung der Azure SQL-Datenbanken durch Azure-Dienste

Viele Nutzungsszenarien der Azure SQL-Datenbanken werden darauf beruhen, dass andere Azure-Dienste eine solche Datenbank zum Betrieb benötigen. Da es bei allen Diensten immer dieselbe Vorgehensweise ist und der Schwerpunkt dieses Buches auch nicht auf Azure an sich, sondern den SQL-Datenbanken liegt, genügt es, ein kleines Beispiel vorzuführen.

Wenn Sie eine neue Azure-Website erstellen, haben Sie die Möglichkeit, eine neue SQL-Datenbank zu erstellen oder die Website mit einer bestehenden zu verknüpfen. Den entsprechenden Dialog sehen Sie in Bild 6.22. Viel Aufregendes ist auch nicht zu berichten, bis auf dass eben eine Combobox vorhanden ist, in der Sie Ihre Wünsche bezüglich der Datenbank kundtun können.

Bild 6.23 zeigt den Dialog, der erscheint, wenn Sie sich für das Erstellen einer neuen Datenbank entschieden haben:

- Ein sehr intuitiver Name, der im Dialog vorgegeben ist, kann natürlich verändert werden ...
- Ist bereits ein Datenbankserver für die Azure SQL-Datenbank vorhanden, wie auf der Abbildung, können Sie diesen auswählen. Alternativ können Sie auch das Erzeugen eines neuen Servers initiieren. Auf der Abbildung sehen Sie, dass es eine Warnmeldung gibt, wenn die Website in Westeuropa liegt, der Datenbankserver aber in Nordeuropa betrieben wird. Wohlgemerkt: Es handelt sich um eine Warnmeldung, man kann das so konfigurieren, macht aber im Normalfall keinen Sinn, sollte also vermieden werden.
- Dann müssen Sie noch vorgeben, mit welchen Credentials sich die Website am Datenbankserver anmelden soll. Es ist durchaus nicht so, dass die Azure-Komponenten automatisch gegenseitigen Vollzugriff hätten.

NEUE WEBSITE - BENUTZERDEFINIERT ERSTELLEN

Website erstellen

URL
UBB-Website11 ✓ .azurewebsites.net

WEBHOSTINGPLAN
Neuen Webhostingplan erstellen

REGION
Westeuropa

DATENBANK
Neue SQL-Datenbank erstellen

NAME DER DATENBANKVERBINDUNGSZEICHENFOLGE ⓘ
DefaultConnection ✓

Aus der Quellcodeverwaltung veröffentlichen ⓘ

2

Bild 6.22 Beim Anlegen einer neuen Website in Azure können Sie eine zugehörige Datenbank neu erstellen oder auch eine vorhandene verknüpfen.

⚠ Datenbank und Website befinden sich nicht in derselben Region.

NEUE WEBSITE - BENUTZERDEFINIERT ERSTELLEN

Datenbankeinstellungen angeben

NAME
UBBWebsA5rg123UX

SERVER
vzsmqxm2 (Speicherort = North Europe)

SERVERANMELDENAME
uboddenberg

KENNWORT FÜR DIE SERVERANMELDUNG

ERWEITERTE DATENBANKEINSTELLUNGEN KONFIGURIEREN

1

Bild 6.23 Wenn Sie einen bestehenden Server auswählen und der liegt in einer anderen Region, erscheint diese Warnmeldung.



HINWEIS: Denken Sie bitte daran, dass in den Einstellungen des Azure SQL-Datenbanksservers der Zugriff von Azure-Diensten gestattet sein muss, siehe Bild 6.10.

Index

Symbole

<\$nopage>Logs siehe Transaktionsprotokolle 84
<\$nopage>Protokolle siehe Transaktionsprotokolle 84
-m 284
%Privileged Time 537
%Processor Time 537
%Usage 535

A

Abfragen analysieren 539
Abfragestatistik 449
Abonnement 558
- anlegen 574, 592
- einrichten 606
- hinzufügen 572
Abonnent 557
Active Directory 16, 28
Active Directory-Benutzer und -Computer 38
Administration 354
Administrativen Zugriff ermöglichen 387
Administratoren der SQL Server-Instanz 17
ADSIEdit 38
AdventureWorks 32
Advisor 46
Agenten 306
AlwaysOn-Failovercluster Instanz 120f.
AlwaysOn konfigurieren 154
AlwaysOn-Status-Ereignisse 174
AlwaysOn-Verfügbarkeitsgruppen 120f., 134, 167, 171, 417
AlwaysOn vorbereiten 152

Amazon EC2 343
Analysis Services 5
Anforderungen 118
- berücksichtigen 2
Anlegen des neuen Speicherkontos 265
Anlegen des Speichers in Azure 404
Anmeldung 23, 57
Ansätze Verfügbarkeit 120
Anzahl der Kerne 66
- pro Edition 74
Arbeitsauslastungsgruppe 476
Arbeitsspeicher 79
Architektur
- Transaktionsprotokolle 87
Article 557
Artikel 557
Audit 469
Ausführungsplan 548
Authentifizierung 22
Authentifizierungsmodus 15
Automatische Registrierung des SPN 38
Automatische Vergrößerung 82
Autorisierung 22
Available Mbytes 535
Average Latch Wait Time (ms) 538
Average Seek Time 93
Avg. Disk Read Queue Length 536
Avg. Disk Sec/Read 536
Avg. Disk Sec/Write 536
Avg. Disk Write Queue Length 536
Azure 265, 309, 342, 366, 373, 381, 404
Azure-Dienste 364
Azure-Netz 377
Azure-Rechenzentrum 265, 351

Azure-Region 368
 Azure-Replikat hinzufügen 160
 Azure-Speicher 265
 Azure SQL bereitstellen 360
 Azure SQL-Datenbanken 342f., 354, 359, 364
 – anlegen 349
 – Einschränkungen 346
 Azure SQL-VM 394
 Azure-Storage 271, 402
 Azure Storage Explorer 409
 Azure-VM 343
 – als Notfallrechenzentrum 416

B

Backup 233
 – auf Azure 309
 – in die Cloud 415
 Backup-Strategie 242
 Bandgerät 288
 Bandroboter 302
 Bandwechsel 338
 Baseline 535
 Basisinstallation 152
 Batch Processings 114
 Batch Requests/Sec 538
 Bedarf an Prozessoren 61
 Benachrichtigungen 433, 440
 benannte Instanzen 521
 Benutzerzuordnung 26
 Berechtigungen 7, 33
 Betrieb 226
 Betriebszustand 423
 Bibliothek 302
 Blöcke 81
 Blockgröße 105, 107
 Browser-Dienst 525
 Buffer Cache hit ratio 535f.
 Business 1
 Business Continuity 239
 Business Intelligence 5
 Business Intelligence Edition 110, 114

C

CALs 61, 109
 Checkpoint Pages/Sec 535
 Cisco 373

Cloud 264
 Cloud-Adapter 390, 400
 Cloud-Technologie 5
 Cluster einrichten 137
 Cluster erstellen 196
 Cluster prüfen 195
 Clusterdatenträger 210
 Clustered Index Scan 550
 Clustererstellungs-Assistent 143
 Clusterquorum 179
 Clusterquorum Einstellungen konfigurieren 147
 Clusterressourcengruppe 205
 Clusterrolle 210
 Code First 541
 Compute Engine 343
 Concatenation 551
 Context Switches/sec 537

D

Dashboard 173
 Data Protection Manager 2012 R2 286
 DATA-Verzeichnis 20
 Database Engine Services 11
 Database First 541
 Dateien 81f.
 Dateigruppen 82
 Datenauswertung 464
 Datenbank in Azure SQL bereitstellen 360
 Datenbanken mit Azure-Storage 402
 Datenbank verkleinern 252
 Datenbank zwischen SQL Servern übertragen 501
 Datenbankdateien 17, 81
 – verschieben 494
 Datenbank-E-Mail 426, 433
 – verwenden 431
 Datenbankintegrität überprüfen 252
 Datenbankspiegelung 121f., 230, 421
 Datenbankzugriff mit ORM-Werkzeugen 540
 Datenquellen bei der Business Intelligence Edition 114
 Datensammlersätze 529
 Datensammlung 440, 446
 – einrichten 441
 Datensammlungssätze 443
 Datensicherung mit Bordmitteln 239
 Datenträgerverwendung 446

Datenverlustzeit 3, 117
 Datenverzeichnisse 17
 Datenzugriffskonto 16
 db_datareader 26
 db_datawriter 26
 db_owner 26
 dbo 35
 Delegation 37
 Disaster Recovery 239
 Dienstkonten 8, 13
 Differenziell 242
 Dimensionierung
 – IOPS 97, 99
 – RAID-Level 100
 – RAID-Set 99
 – Warteschlange 97
 Disk 536
 Disk-Layout 95
 Distributed Transaction Coordinator 197
 distribution 564
 Distributor 557
 DMVs 451, 453, 483
 – für ... 453
 Domänenkonten als Dienstkonten 13
 DPM 286, 294
 DTC 197
 durchschnittliche Warteschlangenlänge 99
 Dynamic Management Views 451
 dynamische TCP-Ports 518

E

E3-Familie 63
 E5-Familie 66
 E5-Prozessoren 65
 E7-Familie 68
 ECC-Speicher 65
 Editionen 78, 110
 Editionsvergleich 74, 562
 Einzelbenutzermodus 282f., 285
 Einzelne Datenbanken migrieren 50
 E-Mail 426
 E-Mail-Adressen außerhalb Ihrer Organisation 429
 Entwurfsansicht eines Wartungsplans 261
 erster Clusterknoten 201
 Erweiterte Ereignisse 454
 Erweiterte Grundlagen 7

Exchange 429
 Extended Events 454

F

Failover 176, 229
 – nach Absturz 177
 Failovercluster 122, 182
 Failovercluster-Feature installieren 136
 Failovercluster-Manager 138, 166, 195, 209
 Failoverclustering 181, 194
 Failoverclusterinstallation 202
 Festplatten 59
 – Blockgröße 105
 – Disk-Layout 95
 – IOPS 97
 – Leistungsindikatoren 98
 – RAID-Level 100
 – Systempartition 95
 – Warteschlange 97
 Festplattenlayout 85
 Festplattenperformance 86
 – Write Penalty 86
 Festplatten-System 81
 Feuerwehrrübung 118f., 270
 Fibre-Channel 123
 Fragmentierung des Index 252
 Free pages 535f.
 fsutil 108
 Full Scans/sec 537

G

Gateway-Subnetz 370
 geclusterte SQL Server-Instanzen 216
 gemischter Modus 16
 geografisch redundanter Speicher 265, 406
 Geplantes Failover 176
 Google Compute Engine 343
 Größe der Zuordnungseinheit 107
 Grundlagen 7

H

Hardware 59
 – RAID-Controller 92
 Hardware-Wiederherstellung 118
 Hauptspeicher 60, 76, 78

High Density Virtualization 114
 HTTP/servername 41
 Hyperthreading 64, 70, 76
 Hyper V-Replikation 124 f., 127

I

Identitäten 22
 Impersonation 37
 Index neu organisieren 252
 In die Cloud sichern 264
 Inplace-Upgrade 43
 – des kompletten Servers 46
 Installation 152
 – von Patches 490
 Installieren 8
 Instanzen 7, 19
 Instanzfunktionen 11
 Instanznamen 12
 Instanzstammverzeichnis 20, 208
 Integration-Services 5, 58
 Integritäts-Explorer 487
 IOPS 93, 97, 99, 105
 I/Os 91
 IPsec 374
 iSCSI 123
 iSCSI-Datenträger 193
 iSCSI einrichten 183
 iSCSI-Initiator 183 f., 189, 191
 iSCSI-Target 184, 188, 191
 – einrichten 185

J

Juniper 373

K

Katastrophenvorsorge 238
 Kerberos 36
 Kerberos-Delegierung 37
 Kerne pro Edition 74
 Klassifizierungsfunktion 476
 klist 41
 Knoten einem SQL Server-Failovercluster
 hinzufügen 211
 Kompatibilitätsgrad 600
 Konfigurations-Manager 516

Konfigurationsüberprüfungs-Assistent 138
 Konflikte 608
 – anzeigen 610
 Konfliktbehandlung 610
 Konsolidieren der SQL Server-Landschaft 4
 Kopieren von Datenbanken 505
 Kopiersicherung 242
 Kosten 1 f.

L

Lastsimulator 61
 Lazy Writes/Sec 535
 Leistungsindikatoren 527
 – Festplatten 98
 Leistungsüberwachung 527
 Live-Ansicht 464
 Lizenzen 74
 Lizenzierung 109
 – DPM 286
 – virtueller Maschinen 113
 Lizenzmodelle 110
 Lock Requests/sec 538
 Logdateien 17
 Logfiles 90, 491, 499
 – werden unendlich groß 491
 Logical Sequence Numbers 271
 Logins/sec 538
 Logische Fehler 238
 Logouts/sec 538
 Lokale Netzwerke 367
 lokal redundant 265
 LSNs 271, 276
 Lücke 278

M

Majority Node Set 135
 Massenprotokolliert 491
 Master-Datenbank 283
 MAXDOP 71 f.
 Maximum Degree of Parallelism 71
 Medienoptionen 244
 Memory 535
 Merge-Replikation 558, 561, 598
 Mergeveröffentlichung 599
 messen 109
 Microsoft Azure 342

- Microsoft Data Protection Manager 2012 R2 286
 - Microsoft Exchange 429
 - Migration 44
 - Mini LSN 87
 - minimaler ServerArbeitsspeicher 79
 - Minimum Recovery Log Sequence Number 87
 - MNS-Cluster 135
 - Model First 541
 - Momentaufnahmeveröffentlichung 558 f.
 - Monitoring 414, 423
 - MSDTC installieren 197
 - MSOLAPSvc.3/servername 41
 - MSSQLSERVER 12
 - MSSQLSvc/servername 41
- N**
- NAT 374
 - Nehalem 70
 - Netzwerk 60
 - erstellen 367
 - Netzwerkprotokolle prüfen 517
 - Neue Anmeldung 23
 - Notfallrechenzentrum 416
 - NTFS-Blöcke 82
 - NTLM 36
 - NUMA 72
 - Number of Deadlocks/sec 538
 - Nutzung der Azure SQL-Datenbanken 364
- O**
- Obergrenze für die Speicherverwendung 79
 - objectSid 26
 - Offlinefähigkeit 554
 - Operatoren anlegen 431
 - ORM-Werkzeug 540
 - Orphaned Logins 515
- P**
- Page Life Expectancy 535
 - Page reads/sec 535 f.
 - Pages Input/Sec 535
 - Page Splits/sec 537
 - Pages/Sec 535
 - Page writes/sec 535 f.
 - Paging File 535
 - Parallelität der Verarbeitung 63
 - PasswordHash 514
 - Patches 490
 - Peer-zu-Peer-Veröffentlichung 558, 598
 - Performance 1, 94, 414
 - Blockgröße (Festplatten) 105
 - Festplatten 86
 - messen und bewerten 526
 - messen und überwachen 76
 - RAID-Level 86, 100
 - Write Penalty 86
 - Performance-Monitor 482, 527
 - PhysicalDisk 536
 - Planung der Backup-Strategie 242
 - Platten 91
 - Plattenspeicher 301
 - Plattentechnologie 92
 - Preiskalkulator 265
 - Preisliste für Azure 382
 - Problem mit den SQL-Anmeldungen 511
 - Processor Queue Length 537
 - Produktkatalog 553
 - Pro-Kern-Lizenzform 110
 - Protokolle 81, 424
 - anzeigen 473
 - Protokolldatei-Viewer 425
 - Protokollfragment 279
 - Protokollfragmentsicherung 280, 282
 - Protokollversand 120, 121
 - Prozessoren 59 ff., 73, 537
 - für kleine Server 63
 - für mittlere Systeme 65
 - für Server mit vier und acht Sockeln 68
 - Prozessorfamilien, Vergleich 69
 - Prozessorkunde 62
 - Prozessorlizenzen 61
 - Prozessorlizenzierung 110
 - public 25
 - Publication 557
 - Publisher 557
 - Pull-Abonnement 575, 592
 - Pull-Abonnenten hinzufügen 581
 - Push-Abonnements 575, 592
 - Push-Abonnenten 573

Q

Quorumdatenträger 206
 Quorumvoten 150
 Quorumzeugen 148

R

RAID 91
 RAID-Controller 82
 RAID-Level 86, 102
 – Einfluss auf die Performance 100
 – Write Penalty 86
 RAID-Sets 17
 Rechnen macht erfolgreich 99
 Recovery 274
 Registrierung des SPN 38
 Replikat hinzufügen 160
 Replikation 553
 – initiieren 609
 – Technische Vorbereitung 555
 – über das Web 612
 Replikationskomponenten 555
 Replikationskonflikte 608
 Replikationskonfliktbehandlung 610
 Replikationskonflikt-Viewer 611
 Replikationstypen 558, 561
 Reporting Services 5
 Resource Governor 475
 Ressourcenkontrolle 475
 Ressourcenpool 476
 Restore 233
 RESTORE DATABASE [databasename] WITH
 RECOVERY 274
 RESTORE WITH NORECOVERY 273, 275
 RESTORE WITH RECOVERY 273
 Richtpreise 110
 Rotational Latency 93
 rowguid 609
 RRAS 373
 Rücksicherung der Vollsicherung 270
 Rücksicherung mit inkrementellen und
 Transaktionsprotokollsicherungen
 275
 Rücksicherungsvorgang 270

S

SA-Benutzer 17
 Scale-out 69
 Schema 31
 Schutzgruppen 327
 SCOM 485
 Securables 34
 Segenswünsche 182
 Seiten 81
 – Aufbau 82
 Seitenheader 82
 Server, Azure SQL-Datenbanken 354
 Serveraktivität 447
 Serverarbeitsspeicher 79
 Server-Hardware
 – RAID-Controller 92
 Serverrollen 25
 Serversysteme
 – IOPS 93
 – RAID 91
 – SAS 94
 – SATA 94
 – SCSI 94
 Serverüberwachungsspezifikation 470
 Service-Klassen 3
 Servicelevel 233f.
 Service Principal Name 36, 41
 servicePrincipalName lesen 40
 servicePrincipalName schreiben 40
 Service-Provider Lizenzen 109
 Service-Ticket 36
 setspn 38
 Shared Storage 181, 191
 Sicherheit | Anmeldung 23
 Sicherheit | Schemas 33
 Sichern/Wiederherstellen 51
 Sicherung durchführen 244
 Sicherung mit Wartungsplan 248
 Sicherungsfähige Elemente 34
 Sicherungsstrategie 239
 Sicherungstyp Differenziell 241
 Sicherungstyp Transaktionsprotokoll 241
 Sicherungstyp Vollständig 241
 Sicherungszeitachse 281
 sid 513
 SIDs 26, 30, 57
 Single-User-Modus 282

Site-to-Site-VPN 367
 Snapshot 559, 604
 Snapshot-Replikation 558 f., 563 f.
 Sortierung 14
 Speicherausbau 77
 Speicherkonto 265, 404
 Speicheroptimierte Tabellen 42
 SPLA 109
 SPN 36, 38, 41
 – automatisch registrieren 38
 – manuell registrieren 38
 SQL-Anmeldungen 511, 274
 SQL-Benutzer 22, 28
 SQL-Cluster installieren 200
 SQL Compilations/sec 538
 SQL-Konten 17
 SQL_Latin1_General_CP1_CI_AS 14
 SQL Profiler 539, 546
 SQL Re-Compilations/sec 538
 SQL Server 366
 SQL Server\
 – Access Methods 537
 – Buffer Manager 535 f.
 – General Statistics 538
 – Latches 538
 – Locks 538
 – SQL Statistics 538
 SQL Server-Agent 18, 432
 SQL Server-Agent-Dienst 250
 SQL Server-Browser-Dienst 525
 SQL Server Configuration Manager 8
 SQL Server-Failoverclusterinstallation 202
 SQL Server-Identitäten 16
 SQL Server installieren 152
 SQL Server-Konfigurations-Manager 516
 SQL Server Management Studio 358
 SQL Server-Netzwerks 204
 SQL Server-Protokolle 246, 424
 SQL Server umbenennen 516
 SQL Server- und Windows-Authentifizierungsmodus 22
 SQL-Sortierung 14
 sqlcmd 283
 SQLCMD 284
 sqlservr.exe 20
 SSIS 58
 Stabilität 1
 Standard Edition 110

Standardinstanz 12
 Startparameter 284
 Statistiken aktualisieren 252
 Stolen pages 536
 Storage
 – Write Penalty 100
 Storage-Systeme im SAN 536
 Subscriber 557
 Subscription 558
 SUSER-NAME() 601
 Synchronisation über das Web 612
 Synchronisierungsstatus anzeigen 609
 sysadmin 25
 System Center Operations Manager 485
 Systemdatenbanken verschieben 501
 Systempartition 95
 Systemsichten 451

T

Tabellenzeilen für die Veröffentlichung filtern 601
 Taktfrequenz 66
 Target Server Memory(KB) 535
 TCP/IP-Protokolls 518
 TCP-Ports 516
 Telnet-Client 517
 tempdb 89
 TempDB 18
 Transaktionsprotokolle 84, 218, 242, 276
 – Architektur 87
 – Festplattenlayout, Auswirkungen auf das 85
 – Sinn und Zweck 84
 – Virtuelle Protokolldatei 87
 Transaktionsprotokollsicherung 247, 278, 491
 Transaktionsprotokollversand 217, 220, 420
 Transaktionsreplikation 558, 560, 584
 Transaktionsveröffentlichung mit aktualisierbaren Abonnements 558
 Trennen/Anfügen 509
 Trennen/Offline und anfügen 55
 Troubleshooting 489

U

Übertragungen/s 99
 Überwachen 109

- Überwachung 226, 336, 481
 - einrichten 469
- umbenennen 516
- UNION vs. UNION ALL 550
- Uniqueidentifier-Spalten 601, 605
- Upgrade 43
- Upgrade Advisor 46
- Upgrade-Pfade 44
- User Connections 538

V

- Verbindung zum SQL Server aufbauen 516
- Verbindungszeichenfolge 58
- Verfügbarkeit 1, 117, 120
- Verfügbarkeitsgruppenlistener 162
- Vergrößerungseinstellungen 83, 90
- Verkleinern von Dateien 492
- Verlaufscleanup 252, 256
- Verleger 557
- Verleger- und Verteilereigenschaften 570
- Veröffentlichung 557
 - einrichten 565, 587
 - vorbereiten 599
- Veröffentlichungstyp 588
- Veröffentlichungszugriffsliste 582
- Verteiler 557
- Verwaltungs-Data Warehouse 441
- VHDX-Datei 188
- virtualisierte Umgebungen 75, 113
- Virtualisierung 75, 120, 122
- virtuelle Maschine 343, 377
 - in Azure 366
- Virtuelle Netzwerke 367
- Virtuellen SQL Server erstellen 380
- Virtuelle Protokolldatei 87
- Vollständig 241, 491
- Vorhandene Datenbank überschreiben 273
- VPN 367
- VPN-Geräteskript herunterladen 373

W

- Warnungen aktivieren 433
- Warnungen konfigurieren 436
- Warteschlange 97f.
- Wartungscleanup 252, 258
- Wartungspläne 18, 58, 248, 252
- Wartungsplan erfolgreich angelegt 260
- Weitere Instanzen installieren 216
- Weitere Knoten 211
- Wiederherstellen der Master-Datenbank 283
- Wiederherstellung eines Systems 118
- Wiederherstellungsmodell 167, 219, 240
- Wiederherstellungsmodus 275
- Wiederherstellungsstatus 54
- Wiederherstellungszeit 3, 117
- Windows-Authentifizierung 16
- Windows-Authentifizierungsmodus 16, 22
- Windows-Benutzer 22f.
- Windows-Firewall 19, 390
 - für benannte Instanzen 521
 - für SQL Server-Browser-Dienst anpassen 525
 - konfigurieren 519
- Windows-Sortierung 14
- WITH NORECOVERY 277
- WITH RECOVERY 277
- WITH REPLACE 285
- Worktables Created/Sec 537
- Write Penalty 86, 100

X

- Xeon E3-Familie 63
- Xeon E5-Prozessoren 65

Z

- Zeilenoffset 82
- Zeitachse 272
- Zeugenserver konfigurieren 146
- Zugriff auf AlwaysOn-Verfügbarkeitsgruppe 171
- Zugriff auf die geclusterte SQL Server-Instanz 216
- Zuordnungseinheit 107