

**Ressourcenkontrolle in SQL Server 2012**

Technischer Artikel zu SQL Server

**Autor:** Guy Bowerman

**Technische Bearbeiter:** Jim van de Erve, Lindsey Allen (ZHU), Madhan Arumugam Ramakrishnan, Xin Jin

**Veröffentlicht:** Juli 2012

**Betrifft:** SQL Server 2012

**Zusammenfassung:** SQL Server 2012 enthält Erweiterungen der Ressourcenkontrolle und kommt damit der wachsenden Nachfrage nach zentral verwalteten Datenbankendiensten nach, die Kunden die Bereitstellung eines mehrinstanzenfähigen Systems für isolierte Arbeitslasten ermöglichen. In diesem Dokument wird beschrieben, welches diese Erweiterungen sind und welche Motivation ihnen zugrunde lag. Hier können Sie eine Demonstration durchlaufen, um sich mit den neuen Funktionen vertraut zu machen.

Copyright

Die Informationen werden in ihrem derzeitigen Zustand zur Verfügung gestellt. Die in diesem Dokument enthaltenen Angaben und Ansichten, einschließlich URLs und anderer Verweise auf Internetwebsites, können ohne vorherige Ankündigung geändert werden. Ihnen obliegt das Risiko der Verwendung.

Mit diesem Dokument erwerben Sie keine Rechte an geistigem Eigentum in Microsoft-Produkten. Sie können dieses Dokument zu internen Zwecken und als Referenz kopieren und verwenden.

© 2012 Microsoft. Alle Rechte vorbehalten.

Inhalt

[Einführung 4](#_Toc354999505)

[Neue Funktionen der Ressourcenkontrolle in SQL Server 2012 5](#_Toc354999506)

[Warum wurde die Ressourcenkontrolle geändert? 5](#_Toc354999507)

[Neue Funktionen im Überblick 6](#_Toc354999508)

[Anwendungsfall für eine isolierte Arbeitslast 6](#_Toc354999509)

[Teil 1 – Ersteinrichtung – Benutzer und Arbeitsauslastungen erstellen 7](#_Toc354999510)

[Teil 2 – Ressourcenkontrolle einrichten 7](#_Toc354999511)

[Teil 3 – Arbeitsauslastung ausführen 8](#_Toc354999512)

[Teil 4 – Überwachung 9](#_Toc354999513)

[Teil 5 – Konkurrierende Arbeitsauslastungen 10](#_Toc354999514)

[Best Practices 16](#_Toc354999515)

[Problembehandlung 17](#_Toc354999516)

[Schlussfolgerung 18](#_Toc354999517)

[Referenzinformationen 19](#_Toc354999518)

# Einführung

Die [SQL Server-Ressourcenkontrolle](http://msdn.microsoft.com/de-de/library/bb933866.aspx) wurde mit Microsoft SQL Server 2008 Enterprise eingeführt, um Mehrinstanzenfähigkeit und Ressourcenisolation für einzelne SQL Server-Instanzen bereitzustellen, auf denen mehrere Clientarbeitslasten verarbeitet werden. Mit der Ressourcenkontrolle lassen sich Speichermenge und CPU-Ressourcen für eingehende Anforderungen einschränken. Darüber hinaus bietet sie die Möglichkeit, Endlosabfragen zu isolieren und einzugrenzen. Weitere Vorteile sind eine differenzierte Ressourcenüberwachung für die Rückbelastung der Kosten von IT-Services und die Bereitstellung voraussagbarer Leistung. Weitere Informationen zu diesem Leistungsmerkmal finden Sie unter [Ressourcenkontrolle](http://msdn.microsoft.com/library/bb933866.aspx) (http://msdn.microsoft.com/library/bb933866.aspx) in der SQL Server-Onlinedokumentation.

Mit SQL Server 2012 erzielen Sie eine umfassendere Isolierung von CPU-Ressourcen für Arbeitslasten und können die CPU-Nutzung deckeln, um Leistungsprognosen zu verbessern. Darüber hinaus können Sie die SQL Server-Speicherbelegung umfassender steuern.

**Ressourcenpools, Arbeitsauslastungsgruppen und Klassifizierung**Mit der SQL Server-Ressourcenkontrolle wurde das Konzept der *Ressourcenpools* als grundlegende Implementierung der Ressourcenisolation innerhalb einer SQL Server-Instanz eingeführt. Ressourcenpools werden über Transact-SQL und andere Verwaltungsschnittstellen wie [SMO](http://msdn.microsoft.com/de-de/library/ms162169.aspx) gesteuert. Die Anweisungen CREATE RESOURCE POOL und ALTER RESOURCE POOL werden verwendet, um minimale und maximale CPU- und Arbeitsspeicherressourcen zuzuweisen.

SQL Server 2012 unterstützt bis zu 62 benutzerdefinierbare Pools – ein deutlicher Anstieg gegenüber den 18 unterstützten Pools in SQL Server 2008. Zwei integrierte Ressourcenpools stehen zur Verfügung: zum einen ein interner Pool (**internal**), der für Systemtasks reserviert und nicht konfigurierbar ist, und zum anderen ein benutzerkonfigurierbarer Pool (**default**), in demstandardmäßig Arbeitslasten ausgeführt werden.

Jeder Benutzerressourcenpool kann einer oder mehreren *Arbeitsauslastungsgruppen* zugeordnet werden. Das sind logische Entitäten, die eine oder mehrere Clientarbeitslasten darstellen.Angemeldete Sitzungen können diesen Arbeitsauslastungsgruppen über eine vom Benutzer definierbare **Klassifizierungsfunktion** zugeordnet werden, die nach der Anmeldung ausgeführt wird und Systemfunktionen zur Überprüfung verschiedener Anmeldeinformationen wie Benutzername, Arbeitsstationsname und Datenbankname aufrufen kann. Im folgenden Diagramm wird veranschaulicht, wie diese Komponenten zusammenwirken, um eingehende Verbindungen in Ressourcenpools zu klassifizieren.



**Abbildung 1**: Klassifizierung der Sitzung durch die Ressourcenkontrolle

Weitere Informationen, einschließlich einer fundierten Einführung in die Ressourcenkontrolle unter SQL Server 2008 und SQL Server 2008 R2, finden Sie im Whitepaper [Verwenden der Ressourcenkontrolle](http://msdn.microsoft.com/library/ee151608.aspx) (http://msdn.microsoft.com/library/ee151608.aspx) von Aaron Bertrand und Boris Baryshnikov.

In diesem Artikel wird beschrieben, welche neuen Funktionen mit SQL Server 2012 eingeführt wurden, warum sie hinzugefügt wurden und wie sie in der Praxis eingesetzt werden.

# Neue Funktionen der Ressourcenkontrolle in SQL Server 2012

## Warum wurde die Ressourcenkontrolle geändert?

Vor dem Hintergrund, dass die Anzahl der CPU-Kerne pro Sockel steigt und die Kosten für leistungsfähige Computer sinken, haben wir von unseren SQL Server-Hostingpartnern, die mithilfe der Ressourcenkontrolle SQL-Ressourcen für ihre Kunden isolieren, immer wieder Kommentare wie diese gehört: „*Obwohl neuere Standardserver über 10 und 12 Kerne pro CPU verfügen, lassen wir CPU-Ressourcen auf unseren Servern ungenutzt, da uns durch die maximale Anzahl der Pools Grenzen gesetzt sind.*“
Hostinganbieter, die ihren Kunden eine kosteneffiziente SQL Server-Datenbank als Dienst anbieten möchten, benötigen mehr Flexibilität, um die Ressourcen leistungsstarker Computer zwischen Arbeitslasten aufzuteilen, Ressourcenkonflikte zwischen Arbeitsauslastungen zu beseitigen und den so genannten „Noisy Neighbor“-Effekt zu vermeiden (bei dem eine Arbeitslast die Ressourcenverfügbarkeit benachbarter Arbeitslasten beeinträchtigt).

Mit zunehmender Leistung der Computer und Virtualisierungssoftware steigen auch Anzahl und Größe der SQL Server-Instanzen, die virtualisiert und in privaten Clouds gehostet werden. Es wird immer wichtiger, Nutzern gemeinsamer Ressourcen effizientere Möglichkeiten der Isolierung anzubieten. Arbeitsauslastungen müssen auf partitionierten Ressourcen innerhalb einer SQL Server-Instanz lauffähig sein, ohne dass der „Noisy Neighbor“-Effekt eintritt.

Ein weiterer Trend sind die zunehmenden Ausgleichsbuchungen der Kosten von IT-Services. Die Nutzung interner Ressourcen zu überwachen und abzurechnen, steht in Unternehmen und bei öffentlichen Hostinganbietern heute auf der Tagesordnung. Das stärkt die Nachfrage nach einem zuverlässigen Abrechnungssystem, das die Nutzung der Ressourcen genau nachhält.

## Neue Funktionen im Überblick

Um diesem Bedarf gerecht zu werden, wurden SQL Server 2012 die folgenden Funktionen hinzugefügt:

* Maximale Anzahl von 64 Ressourcenpools (statt 20) und Aufteilung der Ressourcen leistungsfähiger Computer auf mehrere Arbeitslasten
* CAP\_CPU\_PERCENT: Festlegen einer festen Obergrenze für die Nutzung von CPU-Ressourcen und bessere Voraussagbarkeit
* AFFINITY: Zuordnen einzelner Ressourcenpools zu einem oder mehreren Schedulern und NUMA-Knoten, um die Isolierung von CPU-Ressourcen zu verbessern
* Steuerung der Zuordnung mehrerer Seiten durch den SQL Server-Speicher-Manager zusätzlich zur Zuordnung einzelner Seiten. Dies ist das Ergebnis einer wichtigen Umstrukturierung in SQL Server 2012, in deren Zuge der Speicher-Manager zur zentralen Betriebssystem-Schnittstelle für die Seitenzuordnung wird. Weitere Informationen zu den Änderungen im Hinblick au den Speicher-Manager finden Sie im [SQLOS- und Cloudinfrastruktur-Teamblog](http://blogs.msdn.com/b/sqlosteam/) (http://blogs.msdn.com/b/sqlosteam/).

Um die Wirkungsweise der neuen Funktionen in der Ressourcenkontrolle zu veranschaulichen, zeigen wir die Isolierung von Arbeitslasten anhand einer einfachen Demonstration.

# Anwendungsfall für eine isolierte Arbeitslast

In diesem Szenario hosten Sie eine private Cloud und verwalten einen zentralen Datenbankdienst, der die beiden Abteilungen Vertrieb und Marketing unterstützt. Diese Abteilungen greifen für ihre CPU-intensive Arbeitslast auf dieselbe Datenbank zu. Zur Kostenkontrolle wird die Ressourcennutzung der beiden Abteilungen überwacht. Anschließend berechnen Sie die CPU-Ressourcennutzung für jede der Abteilungen.

Beide Abteilungen sind auf voraussagbare Leistung für ihre Abfragen angewiesen. Dem Vertrieb sollten die benötigten Ressourcen uneingeschränkt zur Verfügung stehen. Im Idealfall sollten jederzeit 100 % eines CPU-Kerns als isolierte Kapazität verfügbar sein. Gleichzeitig muss der Vertrieb die Datenbankinstanz jedoch mit anderen Abteilungen teilen. Die Marketingabteilung führt Abfragen grundsätzlich mit niedrigerer Priorität aus. Obwohl auch sie auf vorhersagbare Leistung angewiesen ist, bewegt sich der Ressourcenbedarf in einem festen Rahmen.

Die Aufgabe besteht darin, die Isolation der Arbeitsauslastung sowie voraussagbare Leistung und Abrechnung in dieser Umgebung bereitzustellen.

**Hinweis**: Zur Verdeutlichung der neuen Funktionen liegt der Schwerpunkt dieser Demonstration auf den CPU-Ressourcen. Auf die Speicherverwaltung, die ebenfalls über die SQL Server-Ressourcenkontrolle gesteuert werden kann, wird hier nicht eingegangen.

Die folgende Demonstration können Sie wie ein Lernprogramm schrittweise durchlaufen. Beginnen Sie mit einem Computer oder einer virtuellen Maschine, der bzw. die mit SQL Server 2012 Enterprise, Evaluation oder Developer sowie zwei CPU-Kernen konfiguriert ist.

## Teil 1 – Ersteinrichtung – Benutzer und Arbeitsauslastungen erstellen

Zunächst erstellen Sie die Benutzeranmeldungen für die Arbeitsauslastungen „Sales“ und „Marketing“. (Dies setzt voraus, dass für die Instanz die SQL Server-Authentifizierung im gemischten Modus als Instanzeigenschaft aktiviert ist.)

CREATE LOGIN Sales WITH PASSWORD = 'UserPwd', CHECK\_POLICY = OFF

CREATE LOGIN Marketing WITH PASSWORD = 'UserPwd', CHECK\_POLICY = OFF

CREATE USER Sales FOR LOGIN Sales;

CREATE USER Marketing FOR LOGIN Marketing;

GO

Der erste Teil der Demonstration zeigt, wie die Arbeitsauslastungen um CPU-Zeit konkurrieren. Bevor Sie die Ressourcenkontrolle verwenden, sollten Sie SQL Server einer einzelnen CPU zuordnen, damit alle Arbeitsauslastungen diese CPU gemeinsam verwenden.

sp\_configure 'show advanced', 1

GO

RECONFIGURE

GO

sp\_configure 'affinity mask', 1

GO

RECONFIGURE

GO

## Teil 2 – Ressourcenkontrolle einrichten

Die SQL Server-Ressourcenkontrolle wird verwendet, um den Arbeitsauslastungen „Sales“ und „Marketing“ separate Ressourcenpools zuzuordnen. Auf diese Weise können Sie die Ressourcennutzung mithilfe der Leistungsüberwachung separat verfolgen. Zunächst werden jedoch keine weiteren Einstellungen für die Ressourcenkontrolle vorgenommen, d. h., den Ressourcenpools werden noch keine CPU-Ressourcen zugeordnet, sodass die Benutzer nicht sehen, ob die Ressourcenkontrolle verwendet wird oder nicht.

1. Erstellen Sie die Ressourcenpools.

CREATE RESOURCE POOL SalesPool

CREATE RESOURCE POOL MarketingPool

1. Erstellen Sie die Arbeitsauslastungsgruppen.

CREATE WORKLOAD GROUP SalesGroup

USING SalesPool

CREATE WORKLOAD GROUP MarketingGroup

USING MarketingPool

GO

1. Erstellen Sie eine Klassifizierungsfunktion, die ausgelöst wird, wenn ein Benutzer eine Verbindung herstellt. Die Funktion ordnet die Verbindung einer bestimmten Arbeitsauslastungsgruppe zu. Das erfolgt auf Grundlage des Anmeldenamens des Benutzers (entweder „Sales“ oder „Marketing“).

CREATE FUNCTION CLASSIFIER\_V1()

RETURNS SYSNAME WITH SCHEMABINDING

BEGIN

 DECLARE @val varchar(32)

 SET @val = 'default';

 IF 'Sales' = SUSER\_SNAME()

 SET @val = 'SalesGroup';

 ELSE IF 'Marketing' = SUSER\_SNAME()

 SET @val = 'MarketingGroup';

 RETURN @val;

END

GO

ALTER RESOURCE GOVERNOR

WITH (CLASSIFIER\_FUNCTION = dbo.CLASSIFIER\_V1)

GO

1. Aktivieren Sie jetzt die Ressourcenkontrolle.

ALTER RESOURCE GOVERNOR RECONFIGURE

GO

## Teil 3 – Arbeitsauslastung ausführen

Anhand der folgenden einfachen Schleife wird eine CPU-intensive Arbeitsauslastung veranschaulicht. Es ist eindeutig erkennbar, wie sich die Verwendung der Ressourcenkontrolle auswirkt: außer dass die CPU arbeitet, passiert nichts.

SET NOCOUNT ON

DECLARE @i INT

DECLARE @s VARCHAR(100)

SET @i = 100000000

WHILE @i > 0

BEGIN

 SELECT @s = @@version;

 SET @i = @i - 1;

END

Wenn Sie diese Schleife als Datei unter dem Namen „workload.sql“ speichern, können Sie die Arbeitsauslastung für den Vertrieb („Sales“) an einer Eingabeaufforderung wie folgt starten.

start sqlcmd -S localhost -U Sales -P UserPwd -i "workload.sql"

Sobald die Arbeitsauslastung läuft, steigt die CPU-Auslastung für den Prozessorkern, dem SQL Server zugeordnet ist, sofort auf 100 % an.

## Teil 4 – Überwachung

Sie können die Ressourcennutzung poolweise mithilfe der Leistungsüberwachung verfolgen. Folgender Leistungsindikator wird verwendet:

**SQLServer:Statistiken für Ressourcenpools->CPU-Verwendung**

Sie können einen Leistungsindikator zur Überwachung jedes einzelnen Pools hinzufügen. Verwenden Sie dazu das Dialogfeld **Leistungsindikatoren hinzufügen**.



Wenn ein Computer mit zwei CPUs verwendet wird, auf dem SQL Server einer einzelnen CPU zugeordnet ist, steigt die CPU-Auslastung des Pools „Sales“ auf maximal 50 % (das entspricht 100 % einer CPU).



Sobald Sie also Leistungsindikatoren in dieser Demonstration begegnen, verdoppeln Sie das Ergebnis, um die prozentuale CPU-Nutzung eines Kerns zu erhalten.

## Teil 5 – Konkurrierende Arbeitsauslastungen

Fügen Sie eine weitere Arbeitsauslastung für „Marketing“ hinzu.

start sqlcmd -S localhost -U Marketing -P UserPwd -i "workload.sql"

Die beiden Arbeitsauslastungen stehen jetzt in Konkurrenz zueinander, und die CPU-Ressourcen werden gleichmäßig auf die Ressourcenpools „Sales“ (grün) und „Marketing“ (rot) aufgeteilt.



Da die Vertriebsabfragen Priorität haben, können Sie sicherstellen, dass diesen ein größerer Anteil der CPU-Ressourcen zugewiesen wird. Legen Sie dazu für den Ressourcenpool „Sales“ einen CPU-Mindestwert fest. Legen Sie diesen Wert für den Pool „Sales“ auf 70 % fest.

ALTER RESOURCE POOL SalesPool

WITH (MIN\_CPU\_PERCENT = 70)

GO

ALTER RESOURCE GOVERNOR RECONFIGURE

GO

Die Arbeitsauslastung „Sales“ nutzt jetzt eindeutig einen größeren Teil der CPU-Ressourcen und nahezu 70 % einer einzelnen CPU, während sich die Arbeitsauslastung „Marketing“ um 30 % bewegt.



Wenn beide Arbeitsauslastungen eine CPU auf diese Weise maximal beanspruchen, ist es für den SQL Server-Scheduler schwierig, die Arbeit proportional zu verteilen und der Arbeitsauslastung „Sales“
genau 70 % oder mehr der CPU-Leistung zuzuteilen. An späterer Stelle wird die Verwendung einer neuen Funktion mit Namen „Scheduler Affinity“ erörtert. Diese Funktion wird verwendet, um Arbeitsauslastungen für einzelne Scheduler oder Gruppen von Schedulern zu isolieren. Um zu verdeutlichen, wie Sie mit der Ressourcenkontrolle Arbeitsauslastungen aufteilen, die unter einer einzelnen oder gemeinsam genutzten CPUs laufen, stellen Sie sich vor, dass die Anzahl der CPUs begrenzt ist und dass diese Arbeitsauslastungen gemeinsam einen einzelnen Scheduler verwenden, der für eine einzelne CPU ausgeführt wird. Sie können die Arbeitsauslastung „Sales“ dem Wert 70 Prozent annähern, indem Sie für den Pool „Marketing“ einen Höchstwert von 30 % festlegen. Der MAX\_CPU\_PERCENT-Wert informiert den Scheduler darüber, wie Ressourcen zuzuweisen sind, wenn ein Konflikt zwischen Arbeitsauslastungen besteht.

ALTER RESOURCE POOL MarketingPool

WITH (MAX\_CPU\_PERCENT = 30)

GO

ALTER RESOURCE GOVERNOR RECONFIGURE

GO

Die Aufteilung der CPU-Ressourcen liegt jetzt viel näher an dem gewünschten Verhältnis von 70/30.



Was würde aber mit der Arbeitsauslastung „Marketing“ passieren, wenn die Arbeitsauslastung „Sales“
auf einen niedrigeren Wert sinkt oder ganz heruntergefahren wird? Bleibt es in diesem Fall beim Maximalwert von 30 %? Die folgende Abbildung zeigt, wie sich eine solche Änderung auswirkt.



Nein. Die MAX\_CPU\_PERCENT-Einstellung ist ein *opportunistischer* Maximalwert. Falls CPU-Kapazität frei wird, wird sie von der Arbeitslast zu 100 % genutzt. Der „Maximalwert“ greift nur bei einem CPU-Ressourcenkonflikt. Wenn die Arbeitsauslastung „Sales“ heruntergefahren wird, nutzt die Arbeitsauslastung „Marketing“ ab diesem Zeitpunkt 100 % der CPU.

Das ist in den meisten Fällen auch beabsichtigt, da es in der Regel unerwünscht ist, wenn die CPU nicht ausgelastet ist. Da die Ressourcen dem Pool „Marketing“ jedoch als Kosten angerechnet werden, die Kosten jedoch kontrollierbar bleiben sollen, wird die CPU-Nutzung bei maximal 30 Prozent gedeckelt. Angenommen, Sie möchten Ressourcen für andere Arbeitsauslastungen reservieren und die Arbeitsauslastung „Marketing“ deshalb immer bei 30 % oder darunter halten. In diesem Fall können Sie eine neue Funktion der Ressourcenkontrolle in SQL Server 2012 verwenden: die Ressourcenpool-Einstellung **CAP\_CPU\_PERCENT**, durch die eine feste Obergrenze für die CPU-Nutzung festgelegt wird. Legen Sie für den Pool „Marketing“ eine feste Obergrenze von 30 % fest.

ALTER RESOURCE POOL MarketingPool

WITH (CAP\_CPU\_PERCENT=30)

GO

ALTER RESOURCE GOVERNOR RECONFIGURE

GO

Die Linie im CPU-Diagramm flacht daraufhin für den Pool „Marketing“ ab und steigt nicht mehr über 30 %. Das garantiert der Marketingabteilung vorhersagbare Abrechnungszahlen.



Nehmen wir nun an, dass die Vertriebsabteilung isoliertere Kapazität benötigt und einen CPU-Kern durchgehend zu 100 % nutzen will. In diesem Fall kommt eine weitere neue Funktion der Ressourcenkontrolle in SQL Server 2012 ins Spiel: die Scheduler Affinity. Entfernen Sie zu diesem Zweck die Affinitätsmaske, die Sie auf SQL Server angewendet haben, und überlassen Sie dem SQL Server-Modul beide Computer-CPUs zur Nutzung.

sp\_configure 'affinity mask', 0

GO

RECONFIGURE

GO

Jetzt können Sie jedem Pool einen eigenen Scheduler zuordnen, indem Sie die neue AFFINITY-Option verwenden. In diesem Fall wird jeder Arbeitsauslastung ein CPU-Kern zur Verfügung gestellt.

ALTER RESOURCE POOL SalesPool

WITH (AFFINITY SCHEDULER = (0))

GO

ALTER RESOURCE POOL MarketingPool

WITH (AFFINITY SCHEDULER = (1))

GO

ALTER RESOURCE GOVERNOR RECONFIGURE

GO

Nachdem Sie diesen Code ausgeführt haben, laufen die Arbeitsauslastungen auf unterschiedlichen CPUs. Solange die feste Obergrenze besteht, verwendet die Arbeitsauslastung „Marketing“ weiterhin 30 % eines Kerns, während die Arbeitsauslastung „Sales“ 100 % des anderen Kerns nutzt. Mit „Scheduler Affinity“ verlaufen die Linien für die CPU-Nutzung jedes Ressourcenpools jetzt ebenmäßig, was auf eine vollständige und konsistente Isolierung von CPU-Ressourcen zwischen den Arbeitsauslastungen hinweist.



Die Arbeitsauslastungen „Sales“ und „Marketing“ werden jetzt sozusagen auf zwei isolierten Maschinen ausgeführt und greifen in voraussagbarer Weise auf die nötigen Ressourcen zu. Diese Demonstration verdeutlicht anhand eines einfachen Beispiels, wie einige der neuen Funktionen der Ressourcenkontrolle in SQL Server 2012 dazu beitragen, Ressourcen effizienter zu isolieren.

# Best Practices

**Betrachten wir nun den gesamten Ressourcenverbrauch der Anwendung.**
Die Demonstration stellt eine idealisierte Arbeitslast dar, die keine Abhängigkeit zu weiteren Ressourcen wie Arbeitsspeicher und E/A aufweist. Sie können die Ressourcenkontrolle verwenden, um feste Obergrenzen für Arbeitsspeicher festzulegen und um die Computerressourcen aufzuteilen. Die MAX\_MEMORY\_PERCENT-Einstellung ist im Prinzip eine feste Obergrenze für die Arbeitsspeichernutzung und keine opportunistische Einstellung. Eine Ausnahme bei der Arbeitsspeicherkontrolle bildet der Pufferpool, der gemeinsam verwendet wird, um die Gesamtabfrageleistung zu optimieren. Die Ressourcenkontrolle verwaltet derzeit keine E/A, sodass andere Methoden zur Minimierung des „Noisy Neighbor“-Effekts für die E/A verwendet werden müssen, z. B. Speichermaximierung, Striping, E/A-Isolation und Optimierung des zufälligen Zugriffsverfahrens (Random I/O) durch effiziente SANs.

**Vermeiden Sie gemischte Arbeitsauslastungen auf denselben CPUs.**
Wenn Arbeitsauslastungen miteinander konkurrieren, ist es für den Scheduler einfacher, eine gerechte Aufteilung vorzunehmen und eine Obergrenze für ähnliche Arbeitsauslastungen festzusetzen. Wenn Sie Ressourcen für stark veränderliche Arbeitsauslastungen kontrollieren, z. B. bei einer Mischung aus hohem CPU-Verbrauch und hoher E/A, sollten Sie die Partitionierung dieser Arbeitsauslastungen mithilfe der Scheduler Affinity-Option in Betracht ziehen, anstatt Maximal- und Deckelungswerte für die CPU festzulegen.

**Konfigurieren Sie die Poolaffinität nach jeder Änderung der CPU-Konfiguration neu.**Ordnen Sie einen Ressourcenpool einer Gruppe von SQL Server-Schedulern zu, um sie auf spezifischen CPUs zu isolieren. Was geschieht aber, wenn der Computer heruntergefahren wird und CPUs wegfallen? Stellen Sie sich vor, Ihre SQL Server-Instanz läuft auf einer virtuellen Maschine und die CPU-Auslastung ist niedrig. Sie konfigurieren die virtuelle Maschine mit einer geringeren Anzahl CPUs neu.

Im besten Fall sind die Scheduler Affinity-Einstellungen nicht mehr gültig, weil die Scheduler, denen ein Ressourcenpool zugewiesen ist, jetzt möglicherweise auf andere CPUs verweisen. Im schlimmsten Fall verweisen sie auf nicht mehr vorhandene CPUs, was zu Fehlern in der Ressourcenkontrolle führen oder die Affinitätsstandardeinstellung „Auto“ zur Folge haben kann.

Konfigurieren Sie die Scheduler Affinity-Einstellungen deshalb immer neu, und starten Sie die Ressourcenkontrolle erneut, nachdem Sie die CPU-Konfiguration geändert haben.

**Befolgen Sie zusätzliche Best Practices-Leitlinien.**
Weitere Informationen zu den Funktionen der Ressourcenkontrolle in SQL Server 2008 finden
Sie im Abschnitt „Best Practices“ des Whitepapers [Verwenden der Ressourcenkontrolle](http://msdn.microsoft.com/library/ee151608.aspx) (http://msdn.microsoft.com/library/ee151608.aspx).

# Problembehandlung

**Sind neue Tools zur Problembehandlung erhältlich?**SQL Server 2012 verfügt über eine neue dynamische Verwaltungssicht, die Sie bei Problemen mit der Ressourcenpoolaffinität und bei der Überwachung der Affinitätseinstellungen unterstützt:

sys.dm\_resource\_governor\_resource\_pool\_affinity

|  |  |  |
| --- | --- | --- |
| Spaltenname | Datentyp | Beschreibung |
| Pool\_id | **int** | Die ID des Ressourcenpools.  |
| Processor\_group | **smallint** | Die ID der logischen Windows-Prozessorgruppe.  |
| Scheduler\_mask | **bigint** | Eine binäre Maske, die die dem Pool zugeordneten Scheduler darstellt.  |

**Ich erhalte Scheduler Affinity-Fehler.**Wenn Sie für einen Pool in den Scheduler Affinity-Einstellungen festlegen, dass die Scheduler 12 bis 14 verwendet werden sollen, wird folgender Fehler angezeigt:

Der SCHEDULER-Bereich, der SCHEDULER 12 bis 14 angibt, umfasst mindestens einen SCHEDULER, der für die aktuelle Instanz nicht verfügbar ist. Der maximal für diese Instanz verfügbare SCHEDULER hat die Nummer 12.

Das bedeutet, dass Sie versuchen, den Ressourcenpool nicht vorhandenen Schedulern zuzuordnen (beispielsweise nachdem eine Konfiguration auf einen Computer mit weniger CPUs verschoben wurde). Es empfiehlt sich, die genauen Zuweisungen zwischen Scheduler und CPU aus der DMV **sys.dm\_os\_schedulers** zu übernehmen.

**Ich kann die im SMO-Objektmodell definierten, neuen Funktionen der Ressourcenkontrolle nicht sehen.**Wenn Sie Windows PowerShell oder C# verwenden, um die Ressourcenkontrolle mit SQL Server Management Objects (SMO) zu konfigurieren, können Sie die neuen Funktionen in SQL Server 2012 nicht nutzen. Die neuen SMO-Klassen werden voraussichtlich in SQL Server 2012 Service Pack 1 (SP1) verfügbar sein.

**Wenn ich eine Ressourcenpooldefinition in SQL Server Management Studio ausgebe, werden die neuen Funktionen nicht angezeigt.**
Die erweiterte Skripterstellung in SQL Server Management Studio, durch die neue Funktionen der Ressourcenkontrolle wie CAP\_CPU\_PERCENT und Scheduler Affinity angezeigt werden, wurden in SQL Server 2012 SP1 hinzugefügt.

# Schlussfolgerung

Die neuen Funktionen der Ressourcenkontrolle in SQL Server 2012 machen es einfacher, CPU und Arbeitsspeicherressourcen eines Computers aufzuteilen, um Arbeitsauslastungen zu isolieren und die Performance einzelner Arbeitslasten in einer mehrinstanzenfähigen Umgebung besser zu prognostizieren. Moderne Computer sind mit immer mehr CPU-Kernen und Arbeitsspeicher ausgestattet und haben das Potenzial, einer größeren Anzahl von Nutzern isolierte Ressourcen zur Verfügung zu stellen. Aus diesem Grund wurde die maximale Anzahl der Ressourcenpools in SQL Server 2012 von 20 auf 64 erhöht. Gemeinsam sorgen diese Funktionen für bessere Mehrinstanzenfähigkeit, indem sie die Aufteilung der Rechenressourcen eines Computers auf eine größere Anzahl isolierterer Pools unterstützen.

Weitere Informationen zur Ressourcenkontrolle in SQL Server 2012 finden Sie unter [Ressourcenkontrolle](http://msdn.microsoft.com/library/bb933866.aspx)
[(](file:///C%3A%5CUsers%5Cguybo%5CAppData%5CLocal%5CMicrosoft%5CWindows%5CTemporary%20Internet%20Files%5CContent.Outlook%5CN4K2BAZ1%5C%28)http://msdn.microsoft.com/library/bb933866.aspx) in der SQL Server-Onlinedokumentation.

## Referenzinformationen

[Übersicht über die Ressourcenkontrolle](http://msdn.microsoft.com/de-de/library/bb933866.aspx)

[Verwenden der Ressourcenkontrolle](http://msdn.microsoft.com/de-de/library/ee151608.aspx) von Aaron Bertrand und Boris Baryshnikov

[SQL Server Management Objects (SMO)-Programmierungshandbuch](http://msdn.microsoft.com/de-de/library/ms162169.aspx)

[SQL Server-Cloudinfrastruktur (Teamblog)](http://blogs.msdn.com/b/sqlosteam/)

**Weitere Informationen:**

<http://www.microsoft.com/sqlserver/>: SQL Server-Website

<http://technet.microsoft.com/de-de/sqlserver/>: SQL Server TechCenter

<http://msdn.microsoft.com/de-de/sqlserver/>: SQL Server DevCenter

War dieses Dokument hilfreich? Geben Sie uns Ihr Feedback. Bewerten Sie dieses Dokument auf einer Skala von 1 (schlecht) bis 5 (ausgezeichnet), und begründen Sie Ihre Bewertung. Beispiel:

* Bewerten Sie es hoch aufgrund guter Beispiele, ausgezeichneter Screenshots, klarer Formulierungen oder aus einem anderen Grund?
* Bewerten Sie es niedrig aufgrund schlechter Beispiele, ungenauer Screenshots oder unklarer Formulierungen?

Dieses Feedback hilft uns, die Qualität von veröffentlichten Whitepapers zu verbessern.

Feedback senden