



Regulador de recursos de SQL Server 2012

Artículo técnico de SQL Server

Autor: Guy Bowerman

Revisores técnicos: Jim van de Erve, Lindsey Allen (ZHU), Madhan Arumugam Ramakrishnan, Xin Jin

Fecha de publicación: julio de 2012

Se aplica a: SQL Server 2012

Resumen: SQL Server 2012 incluye mejoras en el Regulador de recursos que reflejan una necesidad creciente de que los servicios de bases de datos administrados de forma centralizada proporcionen una arquitectura multiempresa a los clientes que necesitan cargas de trabajo aisladas. El presente documento describe esas mejoras y por qué se han agregado. También incluye un tutorial independiente que puede realizar para familiarizarse con las nuevas características.

Copyright

Este documento se proporciona "tal cual". La información y los puntos de vista que se proporcionan en este documento, incluidas las direcciones URL y otras referencias a sitios web de Internet, pueden sufrir modificaciones sin previo aviso. Usted acepta el riesgo de utilizarlo.

En este documento no se proporciona ningún derecho legal sobre ninguna propiedad intelectual de ningún producto de Microsoft. Puede copiar y utilizar este documento para su propia referencia.

© 2012 Microsoft. Todos los derechos reservados.

Contenido

Introducción.....	4
Nuevas características del Regulador de recursos de SQL Server 2012	5
¿Por qué cambiar el Regulador de recursos?	5
Resumen de nuevas características	6
Caso de uso de aislamiento de cargas de trabajo de ejemplo.....	6
Parte 1: configuración inicial: crear usuarios y cargas de trabajo	7
Parte 2: configurar el Regulador de recursos	8
Parte 3: ejecutar la carga de trabajo.....	9
Parte 4: supervisión	9
Parte 5: cargas de trabajo que compiten entre sí.....	11
Prácticas recomendadas	18
Solución de problemas.....	19
Conclusión.....	20
Referencias.....	20

Introducción

El [Regulador de recursos de SQL Server](http://msdn.microsoft.com/library/bb933866.aspx) apareció por primera vez en Microsoft SQL Server 2008 Enterprise como una forma de proporcionar arquitectura multiempresa y aislamiento de recursos para instancias únicas de SQL Server que atendían varias cargas de trabajo cliente. La característica Regulador de recursos le permite establecer límites en cuanto a la cantidad de memoria y los recursos de CPU que las solicitudes entrantes pueden utilizar, y proporciona una manera de aislar y restringir las consultas descontroladas, agregando un seguimiento de recursos específico para las anulaciones y ofreciendo un rendimiento predecible. Para obtener más información acerca de esta característica, vea [Regulador de recursos](http://msdn.microsoft.com/library/bb933866.aspx) (<http://msdn.microsoft.com/library/bb933866.aspx>) en los Libros en pantalla de SQL Server.

Con SQL Server 2012 puede proporcionar un aislamiento más completo de los recursos de CPU para las cargas de trabajo e imponer límites en cuanto al uso de la CPU para lograr un mayor nivel de previsibilidad, así como controlar una mayor proporción de asignaciones de memoria de SQL Server.

Grupos de recursos de servidor, grupos de cargas de trabajo y clasificación

El Regulador de recursos de SQL Server introdujo el concepto de *grupos de recursos de servidor* como una implementación fundamental del aislamiento de recursos dentro de una instancia de SQL Server. Los grupos de recursos de servidor se controlan con Transact-SQL y otras interfaces de administración como [SMO](#); se les pueden asignar recursos mínimos y máximos de CPU y de memoria mediante las instrucciones CREATE RESOURCE POOL y ALTER RESOURCE POOL.

SQL Server 2012 admite hasta 62 grupos definidos por el usuario, con respecto a los 18 de SQL Server 2008. También hay dos grupos de recursos de servidor integrados: uno denominado **interno** que está reservado para las tareas del sistema y no se puede configurar, y un grupo configurable por el usuario denominado **predeterminado** en el que se ejecutarán las cargas de trabajo de forma predeterminada.

Se puede asociar cada grupo de recursos de servidor de usuario a uno o varios *grupos de cargas de trabajo*, que son entidades lógicas que representan una o varias cargas de trabajo cliente. Es posible asociar sesiones entrantes a estos grupos de cargas de trabajo mediante una función **clasificadora** definible por el usuario que se ejecuta después del inicio de sesión; las sesiones entrantes pueden llamar a funciones del sistema para evaluar diversas propiedades de inicio de sesión como el nombre de usuario, el nombre de la estación de trabajo y el nombre de la base de datos. En el diagrama siguiente se muestra el uso conjunto de estos componentes para clasificar las conexiones entrantes en grupos de recursos de servidor.

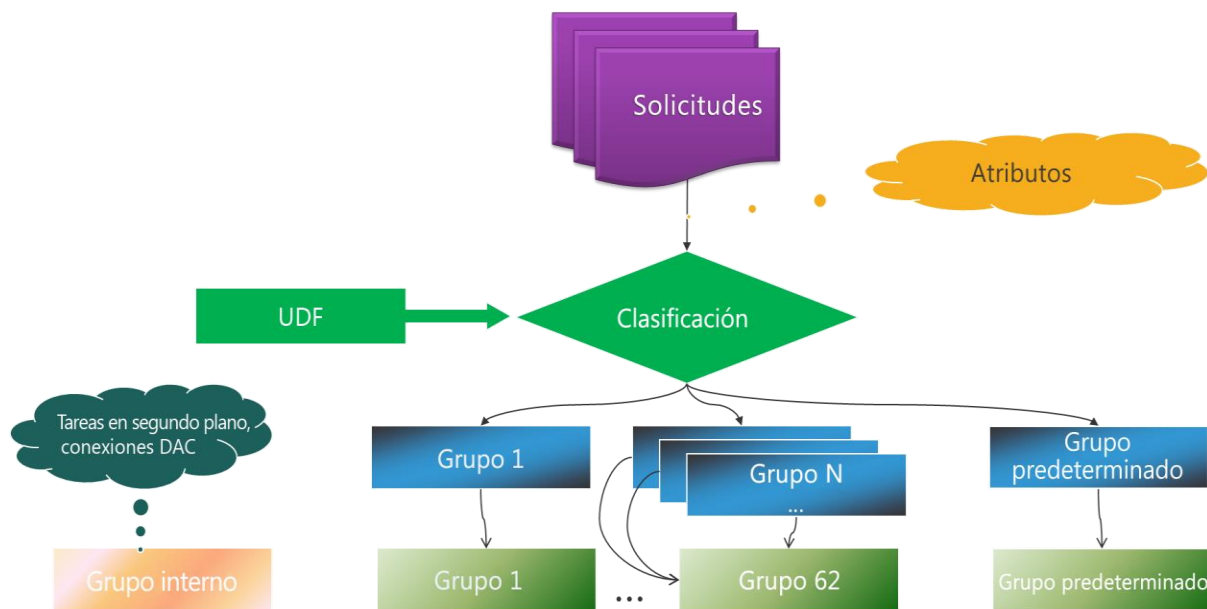


Ilustración 1: clasificación de sesiones del Regulador de recursos

Para obtener más información, incluida una introducción detallada de las características del Regulador de recursos en SQL Server 2008 y SQL Server 2008 R2, vea las notas del producto [Usar el Regulador de recursos](http://msdn.microsoft.com/library/ee151608.aspx) (http://msdn.microsoft.com/library/ee151608.aspx) de Aaron Bertrand y Boris Baryshnikov.

En este artículo se examinan las nuevas características presentadas en SQL Server 2012, se explica por qué se han agregado y se describe cómo usarlas.

Nuevas características del Regulador de recursos de SQL Server 2012

¿Por qué cambiar el Regulador de recursos?

A medida que el número de núcleos de CPU por socket aumenta y el costo de los grandes equipos disminuye, hemos oído comentarios como el siguiente de asociados de hospedaje de SQL Server que usan el Regulador de recursos para aislar recursos de SQL para sus clientes: *"Con los nuevos servidores que tienen 10 y 12 núcleos por CPU, estamos desperdiciando recursos de CPU en los servidores porque estamos limitados por el número máximo de grupos"*. Los proveedores de servicios de hosting que desean proporcionar a sus clientes ofertas rentables de base de datos como servicio basadas en SQL Server necesitan tener más flexibilidad para repartir los recursos disponibles en equipos grandes entre las cargas de trabajo y minimizar los problemas que se producen cuando las cargas de trabajo compiten entre sí por recursos y una carga de trabajo tiene un impacto negativo sobre los recursos disponibles para otra carga de trabajo; es lo que se denomina problema del "vecino ruidoso".

Como los equipos y el software de virtualización son cada vez más eficientes, el número y el tamaño de las instancias de SQL Server que se virtualizan y administran en nubes privadas aumentan. Por tanto, cada vez es más importante proporcionar a los usuarios recursos compartidos con un aislamiento más eficiente. Las cargas de trabajo tienen que ejecutarse en recursos con particiones dentro de una instancia de SQL Server sin experimentar problemas de vecinos ruidosos.

Otra tendencia que hemos visto es un aumento del uso de anulaciones por uso de recursos. El seguimiento y el cobro por el uso de recursos internos dentro de las empresas y por parte de los proveedores de hospedaje públicos son cada vez más habituales, con la consecuente necesidad de proporcionar una facturación predecible a los consumidores de estos recursos.

Resumen de nuevas características

Se han agregado las características siguientes a SQL Server 2012 con el fin de abordar estas necesidades:

- Aumento del número máximo de grupos de recursos de servidor de 20 a 64, lo que permite repartir equipos mayores entre más cargas de trabajo.
- Adición de una opción CAP_CPU_PERCENT a los grupos de recursos de servidor para establecer un límite estricto sobre el uso de recursos de CPU, lo que proporciona mayor previsibilidad.
- Adición de una opción AFFINITY a los grupos de recursos de servidor que permite establecer afinidades entre grupos de recursos de servidor individuales y uno o varios programadores y nodos NUMA, lo que proporciona mayor aislamiento de los recursos de CPU.
- Control de varias asignaciones de página por parte del Administrador de memoria de SQL Server, además de asignaciones de página únicas. Esto se debe a un cambio de diseño importante realizado en SQL Server 2012, convirtiéndose el Administrador de memoria en la interfaz central del sistema operativo para la asignación de páginas. Para obtener más información sobre los cambios realizados en el Administrador de memoria, vea [Blog del equipo de SQLOS e infraestructura de nube](http://blogs.msdn.com/b/sqlosteam/) (<http://blogs.msdn.com/b/sqlosteam/>).

Para mostrar el funcionamiento de las nuevas características de control de recursos y cómo usarlas, he aquí una demostración sencilla que muestra en acción el aislamiento de cargas de trabajo.

Caso de uso de aislamiento de cargas de trabajo de ejemplo

En este escenario, usted hospeda una nube privada y mantiene un servicio de base de datos centralizado que atiende a dos departamentos: Sales (Ventas) y Marketing. Estos departamentos comparten la misma base de datos entre dos cargas de trabajo que usan mucha CPU. Para realizar un seguimiento de los costos, hace un seguimiento de los recursos que emplea cada departamento. Después cobra a los departamentos individualmente por los recursos de CPU que utilizan.

Ambos departamento necesitan un rendimiento predecible para sus consultas. El departamento de Ventas desea que sus consultas tengan siempre los recursos que necesitan e idealmente desea una experiencia aislada en la que tengan disponible el 100 por cien de un núcleo de CPU en todo momento. Al mismo tiempo, sus usuarios deben compartir la instancia de base de datos con otros departamentos. Mientras tanto, el departamento de Marketing tiene consultas de menor prioridad. También necesitan un rendimiento predecible, pero dentro de un límite de recursos permitidos.

Su trabajo consiste en proporcionar aislamiento de cargas de trabajo, rendimiento predecible y facturación predecible en este entorno.

Nota: para mostrar claramente las nuevas características, este tutorial se centra en los recursos de CPU. No incluye el control de la memoria, que también se puede realizar mediante el Regulador de recursos de SQL Server.

El siguiente tutorial se puede recorrer paso a paso. Para ello, empiece con un equipo o una máquina virtual que tenga instalado SQL Server 2012 Enterprise, Evaluation o Developer y esté configurados con dos núcleos de CPU.

Parte 1: configuración inicial: crear usuarios y cargas de trabajo

En primer lugar, cree los inicios de sesión de usuario para las cargas de trabajo de Ventas y Marketing. (Se da por supuesto que la autenticación de modo mixto de SQL Server está habilitada para esta instancia como una propiedad de instancia).

```
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
```

La primera parte de la demostración muestra cargas de trabajo que compiten entre sí por tiempo de CPU; por eso, antes de empezar a usar el Regulador de recursos, establezca una afinidad de SQL Server con una única CPU de modo que todas las cargas de trabajo deban compartir una única CPU.

```
sp_configure 'show advanced', 1
GO
RECONFIGURE
GO
sp_configure 'affinity mask', 1
GO
RECONFIGURE
GO
```

Parte 2: configurar el Regulador de recursos

Se usará el Regulador de recursos de SQL Server para asociar las cargas de trabajo de Ventas y Marketing a grupos de recursos de servidor diferentes. De esta forma puede hacer un seguimiento de su uso de los recursos por separado con Monitor de rendimiento. Sin embargo, no se realizará ninguna otra configuración del Regulador de recursos inicialmente; es decir, no se asignará ningún recurso de CPU a los grupos de recursos de servidor todavía, por lo que los usuarios no apreciarán ninguna diferencia entre usar el Regulador de recursos y no usarlo.

1. Cree los grupos de recursos de servidor.

```
CREATE RESOURCE POOL SalesPool
CREATE RESOURCE POOL MarketingPool
```

2. Cree los grupos de cargas de trabajo.

```
CREATE WORKLOAD GROUP SalesGroup
USING SalesPool

CREATE WORKLOAD GROUP MarketingGroup
USING MarketingPool
GO
```

3. Cree una función clasificadora que se desencadene cuando un usuario se conecte. La función asocia la conexión a un grupo de cargas de trabajo específico según el inicio de sesión del usuario entrante (es decir, si el inicio de sesión procede de Ventas o de 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
```

4. Active ahora el Regulador de recursos.

```
ALTER RESOURCE GOVERNOR RECONFIGURE
GO
```


Parte 3: ejecutar la carga de trabajo

El bucle siguiente muestra una carga de trabajo que hace un uso intensivo de la CPU para ilustrar claramente los efectos de usar el Regulador de recursos. Es decir, solo hace que la CPU funcione.

```
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
```

Si guarda este bucle como un archivo con el nombre workload.sql, puede iniciar la carga de trabajo de Ventas en un símbolo del sistema de la manera siguiente.

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

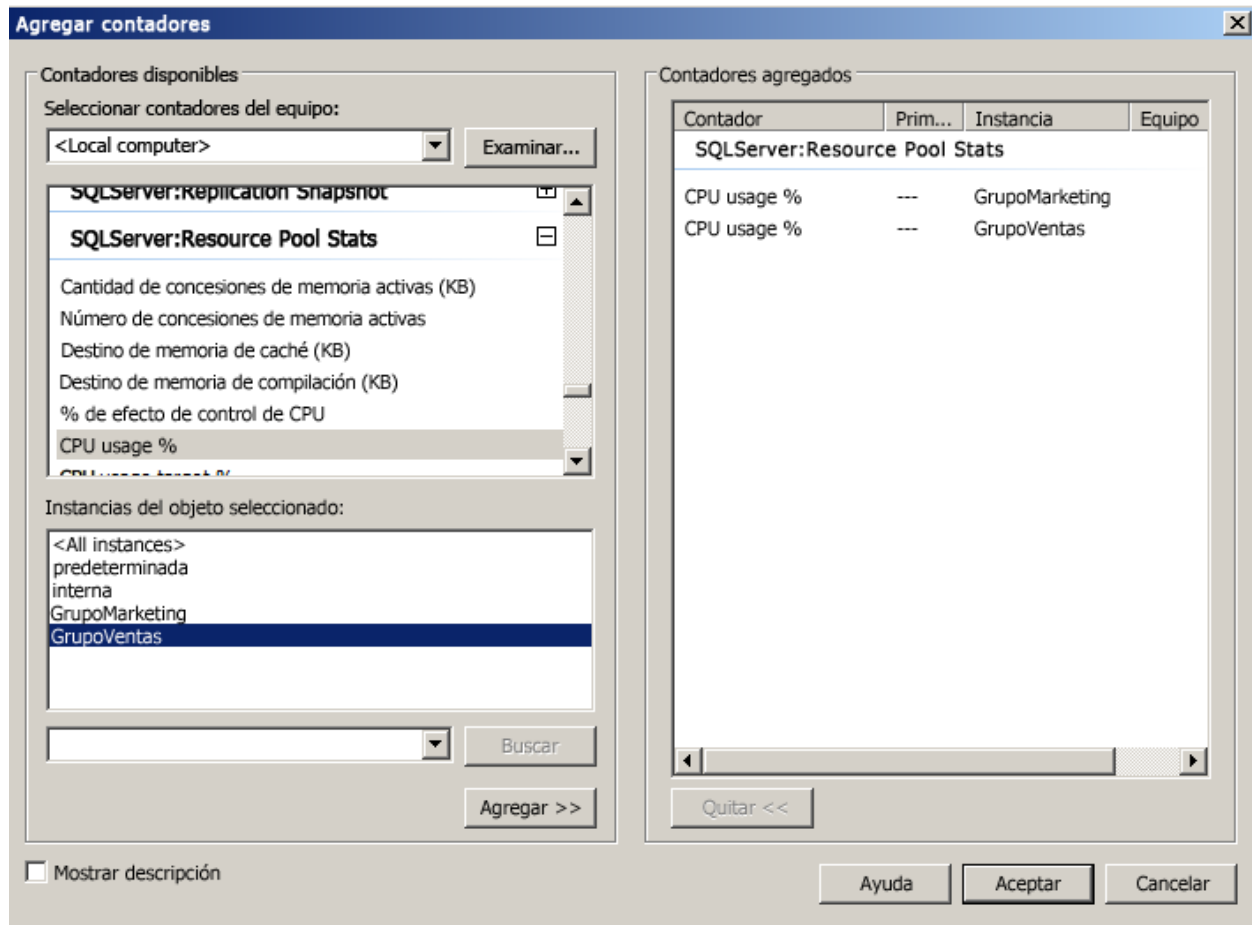
Con esta carga de trabajo en ejecución, el uso de la CPU para SQL Server se dispara directamente hasta el 100 por cien del único núcleo con el que SQL Server tiene afinidad.

Parte 4: supervisión

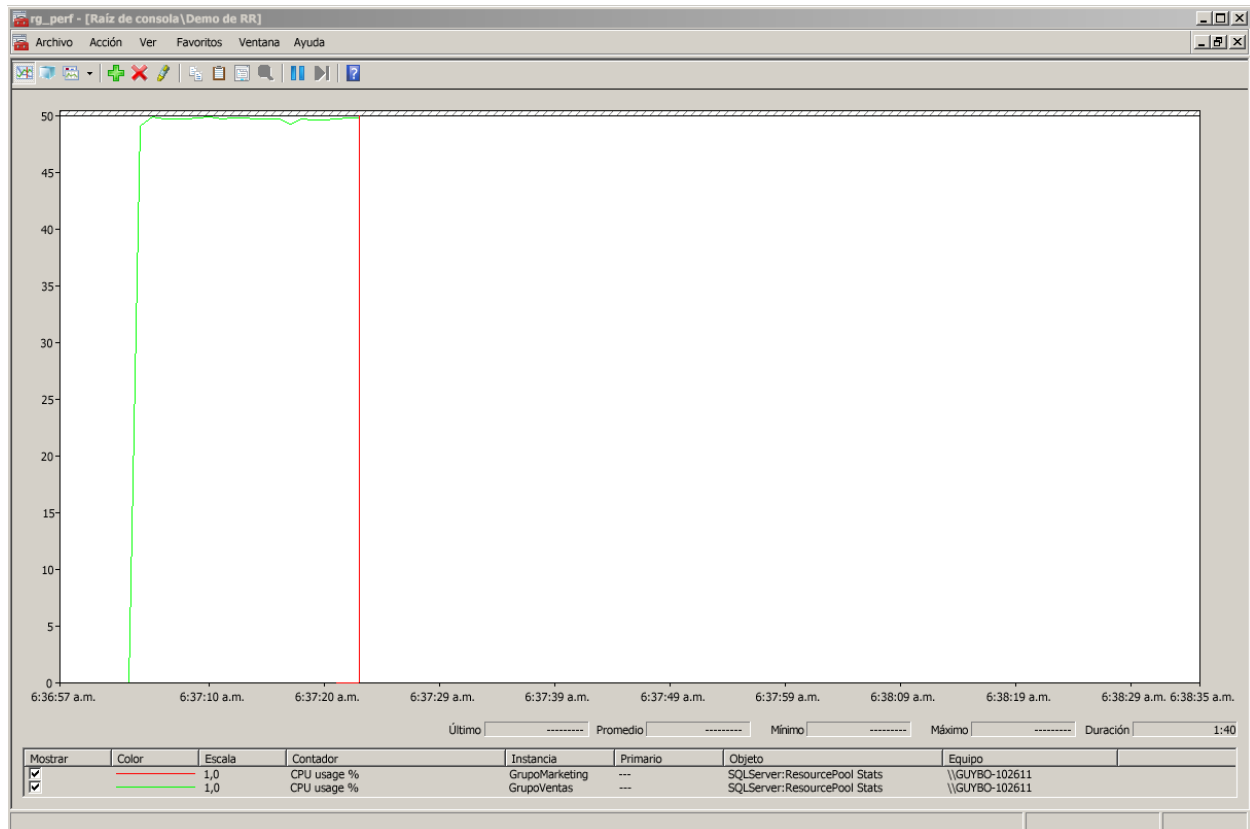
Puede hacer un seguimiento del uso de los recursos por grupo mediante Monitor de rendimiento. El contador de rendimiento correspondiente es:

```
SQLServer:Resource Pool Stats->CPU usage
```

Puede agregar un contador para hacer un seguimiento de cada grupo mediante el cuadro de diálogo **Agregar contadores**.



Observe que en un equipo con dos CPU, si SQL Server tiene afinidad con una única CPU, el uso de CPU del grupo de Ventas llegará a un máximo del 50 por ciento en el gráfico (es decir, el 100 por cien de una CPU).



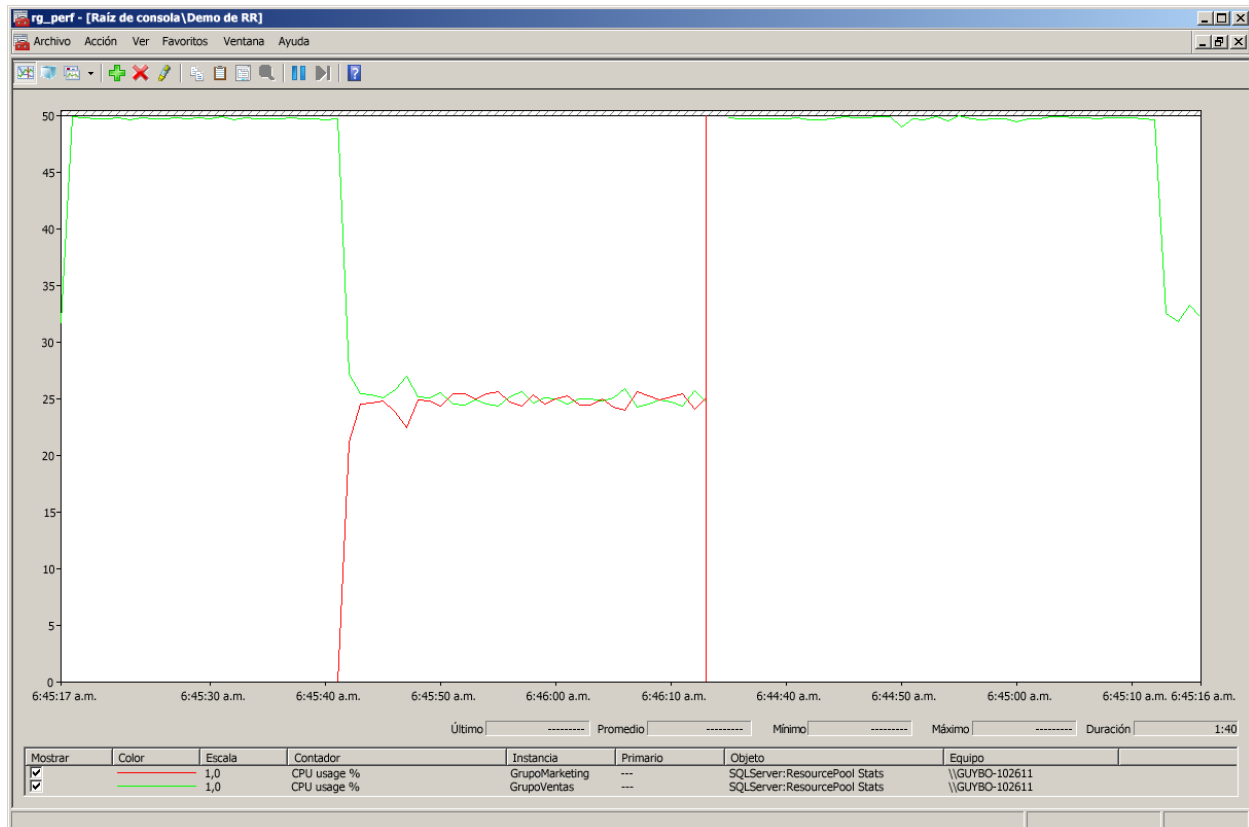
Siempre que vea los contadores gráficos en este tutorial, duplíquelos para obtener el porcentaje de uso de CPU de un núcleo.

Parte 5: cargas de trabajo que compiten entre sí

Agregue otra carga de trabajo para representar Marketing.

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

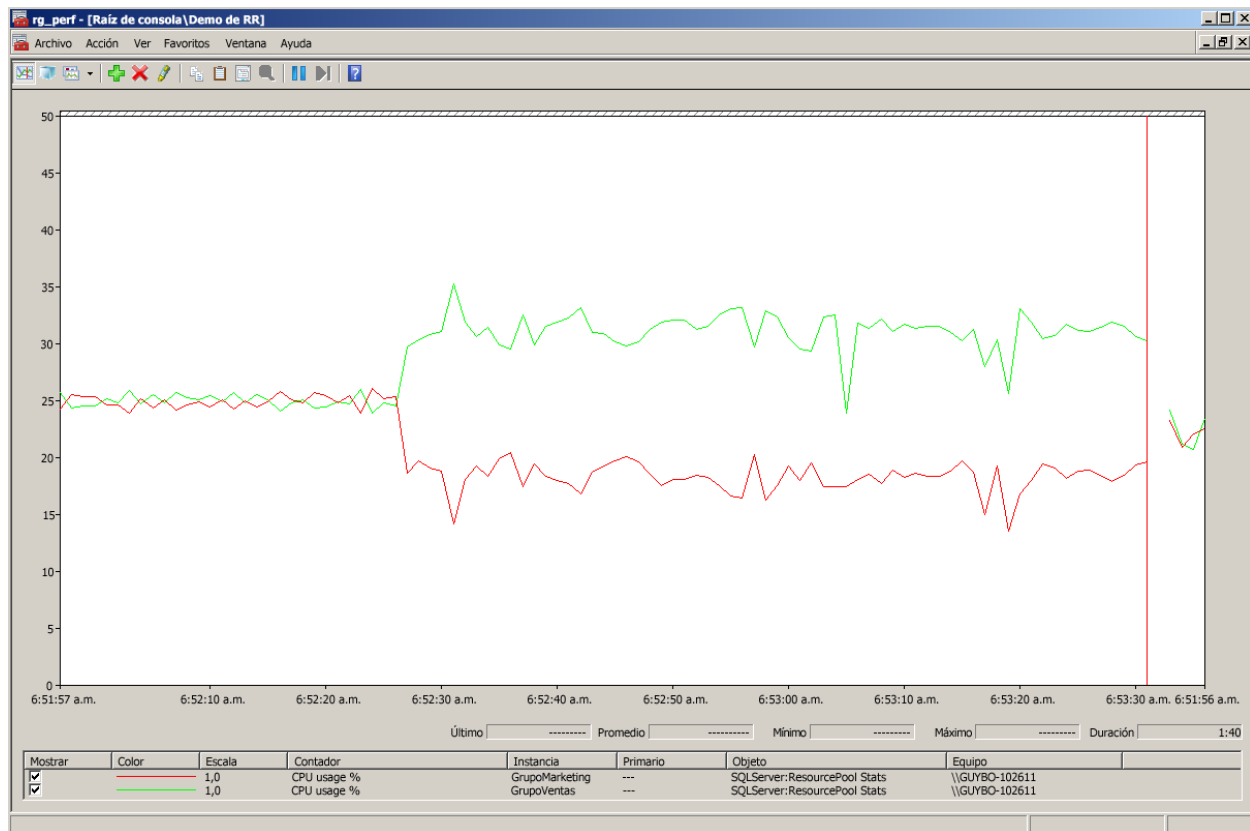
Ahora las dos cargas de trabajo compiten entre sí y los recursos de CPU se reparten equitativamente entre los grupos de recursos de servidor de Ventas (verde) y de Marketing (rojo).



Puesto que las consultas de Ventas tienen prioridad, puede asegurarse de que tienen asignada una proporción mayor de los recursos de CPU si ajusta el grupo de recursos de servidor de Ventas y establece un valor mínimo de CPU. Establezca este valor en el 70 por ciento para el grupo de Ventas.

```
ALTER RESOURCE POOL SalesPool
WITH (MIN_CPU_PERCENT = 70)
GO
ALTER RESOURCE GOVERNOR RECONFIGURE
GO
```

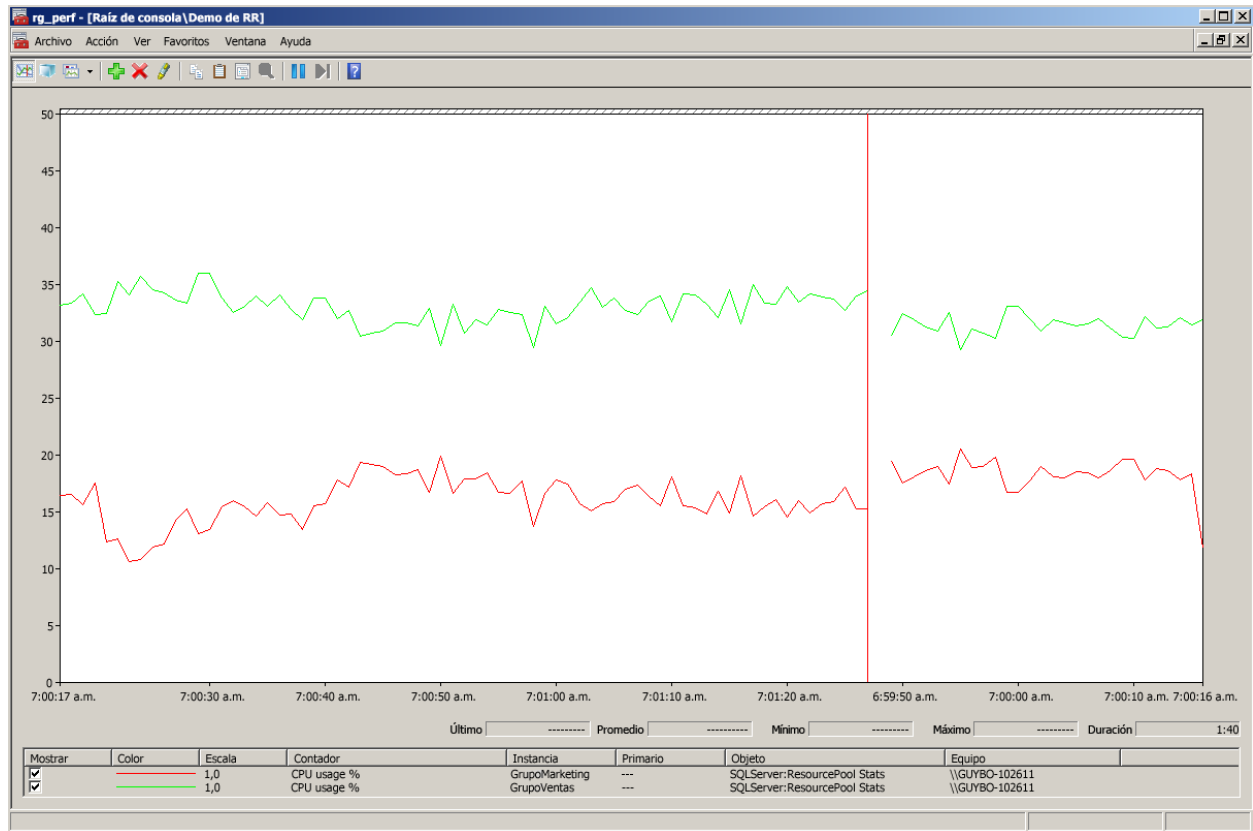
Ahora que la carga de trabajo de Ventas está obteniendo claramente más recursos de CPU, y casi el 70 por ciento de una única CPU, la carga de trabajo de Marketing se está acercando al 30 por ciento.



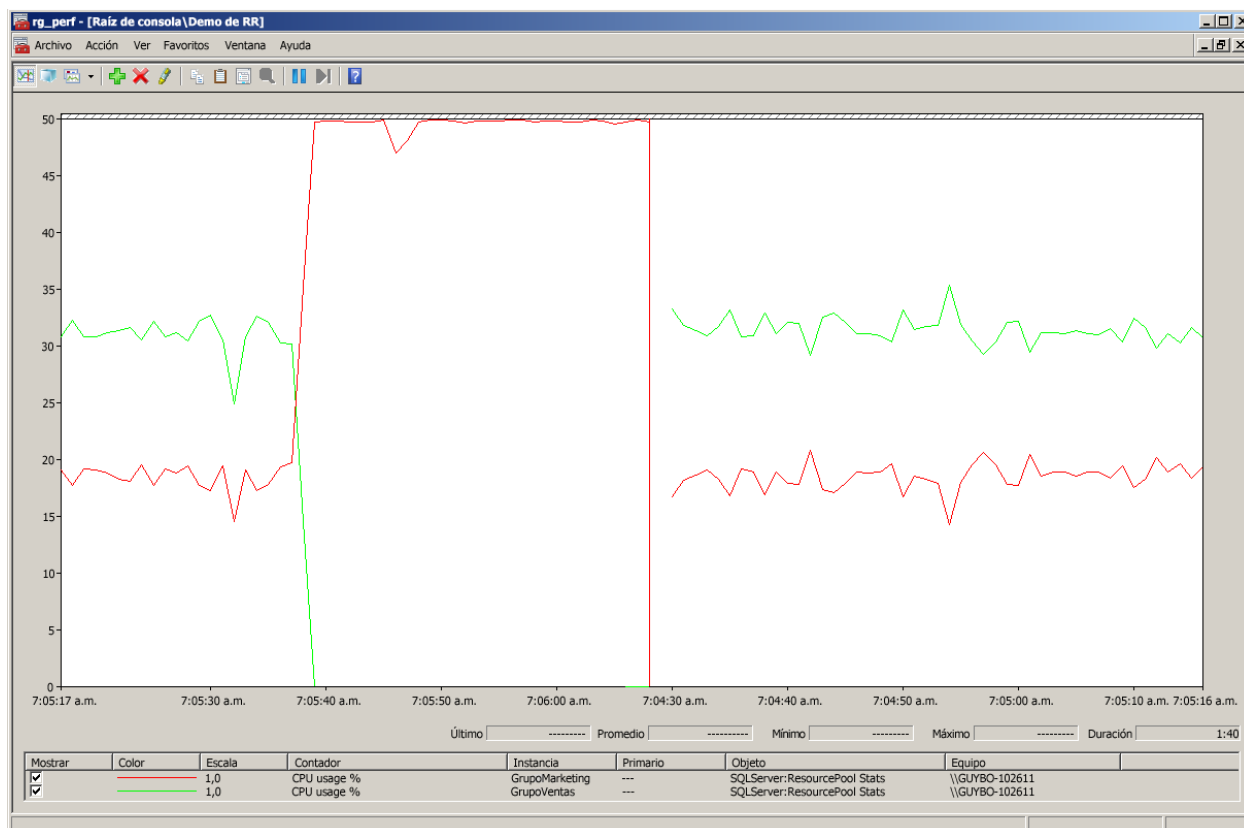
Cuando ambas cargas de trabajo aprovechan al máximo una única CPU como esta, sigue siendo difícil que el programador de SQL Server reparta el trabajo en partes proporcionales y asigne a la carga de trabajo de Ventas exactamente el 70 por ciento o más de la CPU. Más adelante en este documento se describe cómo usar una nueva característica, la afinidad del programador, para aislar cargas de trabajo en programadores individuales o en grupos de programadores. Sin embargo, para poder mostrar cómo puede usar el Regulador de recursos con el fin de repartir cargas de trabajo que se ejecutan en CPU únicas o compartidas, suponga que el número de CPU es limitado y que estas cargas de trabajo tienen que compartir un único programador que se ejecuta en una sola CPU. Puede hacer que la carga de trabajo de Ventas esté más cerca del 70 por ciento si establece un valor máximo del 30 por ciento para el grupo de Marketing. El valor de MAX_CPU_PERCENT indica al programador cómo asignar los recursos cuando hay contención entre las cargas de trabajo.

```
ALTER RESOURCE POOL MarketingPool
WITH (MAX_CPU_PERCENT = 30)
GO
ALTER RESOURCE GOVERNOR RECONFIGURE
GO
```

Ahora, la división de recursos de CPU es mucho más cercana a la relación 70/30 deseada.



Pero ¿qué ocurriría con la carga de trabajo de Marketing si la carga de trabajo de Ventas disminuyera o se detuviera? ¿Permanecería en el valor máximo del 30 por ciento? En la ilustración siguiente se muestran los resultados de este tipo de cambio.

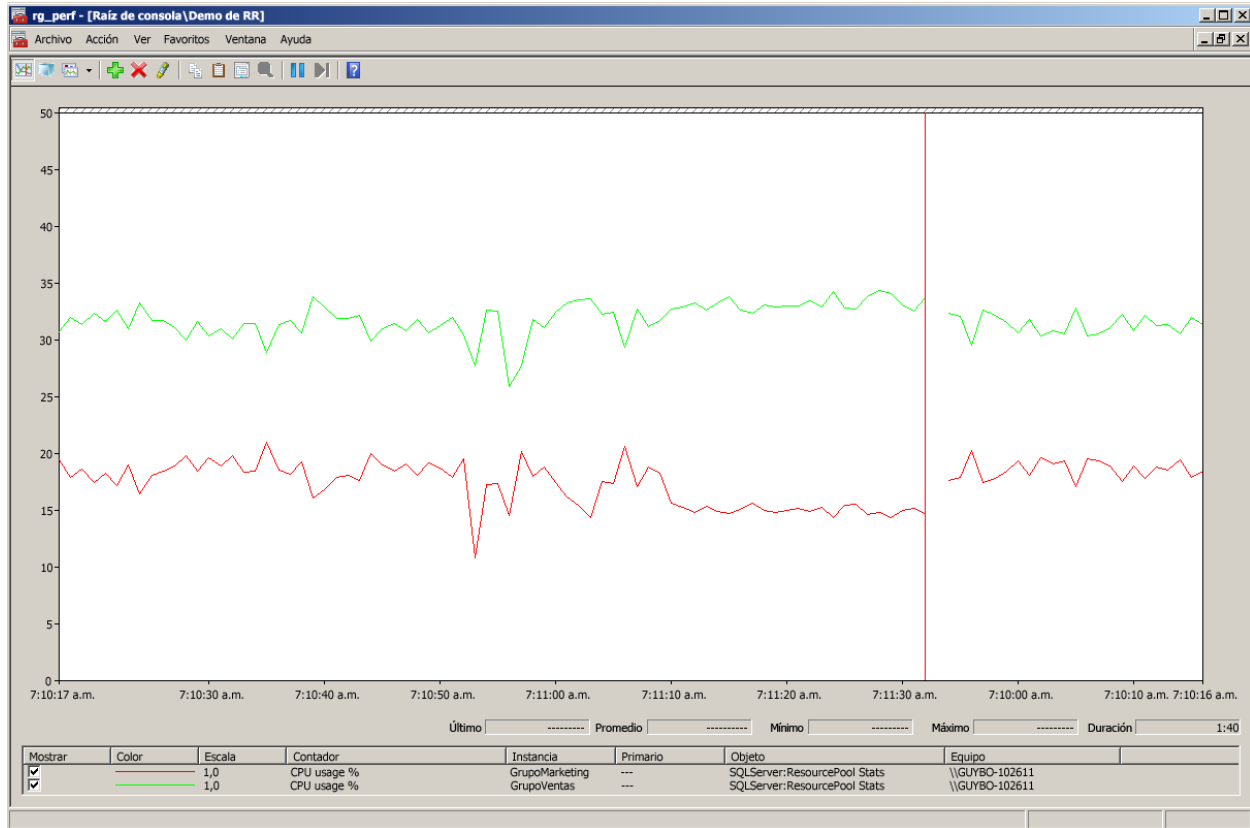


No. El valor de MAX_CPU_PERCENT es un máximo *oportunista*. Si hay capacidad disponible de CPU, la carga de trabajo la utiliza hasta el 100 por cien. El valor "máximo" solo se aplica cuando hay contención de CPU, de modo que si la carga de trabajo de Ventas se desactiva, la carga de trabajo de Marketing empieza a usar el 100 por cien de la CPU.

En muchas circunstancias esto es exactamente lo que se necesita, porque no suele ser aconsejable que la CPU permanezca inactiva. Pero como en este caso se está cobrando a Marketing por su uso de recursos y desean tener una facturación predecible, no desean disponer de más del 30 por ciento de un núcleo. Imagine que desea reservar recursos para otras cargas de trabajo y mantener siempre la carga de trabajo de Marketing en el 30 por ciento o menos. Puede hacerlo mediante una nueva característica del Regulador de recursos de SQL Server 2012 denominada la configuración **CAP_CPU_PERCENT** del grupo de recursos de servidor, que establece un límite máximo de uso de la CPU. Establezca un límite máximo del 30 por ciento para el grupo de Marketing.

```
ALTER RESOURCE POOL MarketingPool
WITH (CAP_CPU_PERCENT=30)
GO
ALTER RESOURCE GOVERNOR RECONFIGURE
GO
```

Ahora, el gráfico de la CPU para el grupo de Marketing se aplanará y ya no supera el 30 por ciento. El departamento de Marketing obtendrá su facturación predecible.



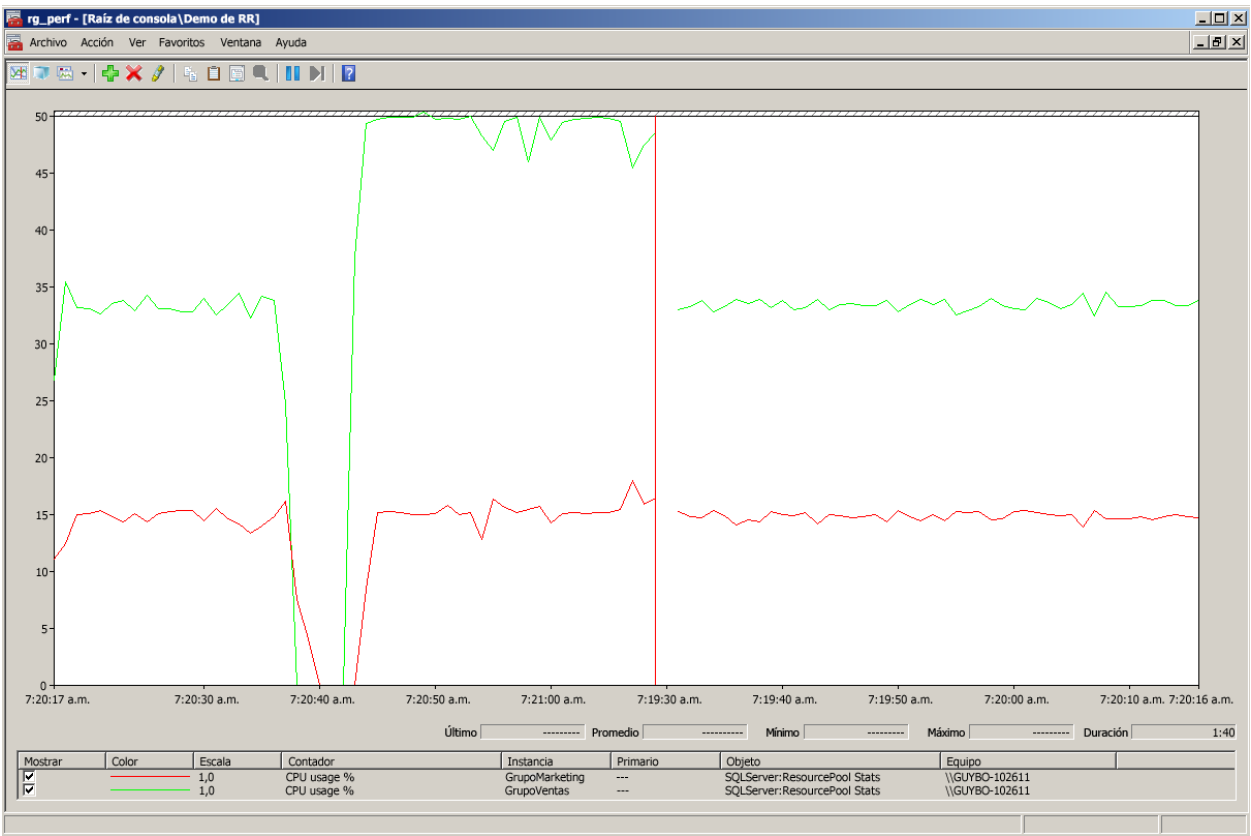
Suponga que el departamento de Ventas necesita una experiencia más aislada y desea tener disponible el 100 por cien de un núcleo de CPU en todo momento. Pueden usar otra característica nueva del Regulador de recursos de SQL Server 2012: la afinidad del programador. Para verlo, quite la máscara de afinidad que aplicó a SQL Server y permita que el motor de SQL Server use ambas CPU del equipo.

```
sp_configure 'affinity mask', 0
GO
RECONFIGURE
GO
```

Ahora puede establecer una afinidad de cada grupo con su propio programador mediante la nueva opción AFFINITY del grupo de recursos de servidor. En este caso, la opción pone un solo núcleo de CPU a disposición de cada carga de trabajo.

```
ALTER RESOURCE POOL SalesPool
WITH (AFFINITY SCHEDULER = (0))
GO
ALTER RESOURCE POOL MarketingPool
WITH (AFFINITY SCHEDULER = (1))
GO
ALTER RESOURCE GOVERNOR RECONFIGURE
GO
```


Después de ejecutar este código, las cargas de trabajo se programan en varias CPU. Ahora, con el límite todavía vigente, la carga de trabajo de Marketing sigue usando el 30 por ciento de un núcleo, mientras que la carga de trabajo de Ventas usa el 100 por cien del otro núcleo. Con la afinidad del programador, las líneas de uso de CPU para cada grupo de recursos de servidor son ahora lisas, mostrando un aislamiento completo y coherente de los recursos de CPU entre estas cargas de trabajo.



Ahora, en lo que respecta a las cargas de trabajo de Ventas y Marketing, se ejecutan en dos equipos aislados, obteniendo los recursos que necesitan de manera predecible. Este tutorial ha mostrado un ejemplo simple de cómo se pueden usar algunas de las características nuevas del Regulador de recursos de SQL Server 2012 para proporcionar un aislamiento de recursos más eficiente.

Prácticas recomendadas

Considere el consumo total de recursos de la aplicación.

La demostración del presente documento representa una carga de trabajo idealizada, sin dependencias de otros recursos como la memoria y E/S. En el caso de la memoria, el Regulador de recursos puede proporcionar límites máximos y repartir los recursos del equipo. La configuración MAX_MEMORY_PERCENT del grupo de recursos de servidor es realmente un límite máximo del uso de memoria en lugar de un valor oportunista. Una excepción al control de memoria es el grupo de búferes, que se comparte para optimizar el rendimiento global de la consulta. El Regulador de recursos no administra actualmente la E/S, por lo que es necesario emplear otros métodos para minimizar los problemas de vecinos ruidosos con la E/S, como maximizar la memoria, creación de bandas, aislamiento de E/S y optimización para E/S aleatoria con SAN eficientes.

Evite compartir cargas de trabajo mixtas en las mismas CPU.

Cuando las cargas de trabajo compiten entre sí, es más fácil que el programador reparta equitativamente y limite las cargas de trabajo similares. Si está controlando recursos para cargas de trabajo muy variables, por ejemplo una combinación de uso elevado de CPU con alta E/S, considere la posibilidad de repartir esas cargas de trabajo mediante la afinidad del programador en lugar de establecer valores máximos y límites de CPU.

Reconfigure siempre la afinidad del grupo después de cambiar la configuración de CPU.

Se crea la afinidad de un grupo de recursos de servidor con un conjunto de programadores de SQL Server con la expectativa de aislarlos de determinadas CPU. ¿Qué ocurre si se apaga el equipo y se quitan las CPU? Por ejemplo, la instancia de SQL Server se está ejecutando en una máquina virtual y el uso de CPU es bajo, por lo que reconfigura la máquina virtual con menos CPU.

En el mejor de los casos, la configuración de afinidad del programador ya no es exacta porque los programadores con los que tiene afinidad un grupo de recursos de servidor quizás no apunten ahora a CPU diferentes. En el peor de los casos pueden apuntar a CPU inexistentes, lo que produciría errores del Regulador de recursos o la afinidad se establecería de forma predeterminada en automática.

Por tanto, reconfigure siempre la afinidad del programador y reinicie el Regulador de recursos después de realizar cambios en la configuración de CPU.

Siga las prácticas recomendadas adicionales.

Para obtener más información sobre las instrucciones que abarcan características del Regulador de recursos de SQL Server 2008, vea la sección sobre las prácticas recomendadas de las notas del producto [Usar el Regulador de recursos](http://msdn.microsoft.com/library/ee151608.aspx) (<http://msdn.microsoft.com/library/ee151608.aspx>).

Solución de problemas

¿Hay nuevas herramientas que ayuden con la solución de problemas?

Para solucionar problemas relacionados con la afinidad de grupos de recursos de servidor y supervisar la configuración de afinidad, SQL Server 2012 incluye una nueva DMV:

`sys.dm_resource_governor_resource_pool_affinity`

Nombre de columna	Tipo de datos	Descripción
Pool_id	int	Identificador del grupo de recursos de servidor.
Processor_group	smallint	Identificador del grupo de procesadores lógicos de Windows.
Scheduler_mask	bigint	Máscara binaria que representa los programadores asociados al grupo.

Aparecen errores de afinidad del programador.

Suponga que cuando configura la afinidad del programador para que un grupo emplee los programadores 12 a 14 aparecen errores como este:

El intervalo de SCHEDULER que especifica SCHEDULER 12 a 14 incluye al menos un SCHEDULER que no está disponible en la instancia actual. El número máximo de SCHEDULER disponibles para esta instancia es 12.

Esto implica que está intentando establecer afinidad entre el grupo de recursos de servidor y programadores que no existen (por ejemplo, movió la configuración a un equipo que tiene menos CPU). Una buena manera de resolver este problema consiste en consultar la DMV **sys.dm_os_schedulers** para ver las asignaciones exactas de programador/CPU.

No veo las nuevas características del Regulador de recursos definidas en el modelo de objetos SMO.

Si emplea Windows PowerShell o C# para configurar el Regulador de recursos mediante Objetos de administración de SQL Server (SMO), no podrá hacer uso de las nuevas características de SQL Server 2012. Se espera que las nuevas clases de SMO estén disponibles en SQL Server 2012 Service Pack 1 (SP1).

Cuando creo un script de una definición de grupo de recursos de servidor en SQL Server Management Studio, no muestra las nuevas características.

Las mejoras de scripting de SQL Server Management Studio para mostrar las nuevas características del Regulador de recursos como CAP_CPU_PERCENT y la afinidad del programador se han agregado a SQL Server 2012 SP1.

Conclusión

Las nuevas características del Regulador de recursos de SQL Server 2012 facilitan el reparto de los recursos de CPU y de memoria de un equipo para proporcionar aislamiento de cargas de trabajo y un rendimiento predecible a las cargas de trabajo individuales en un entorno multiempresa. Como las especificaciones de los equipos individuales están aumentando en cuanto a núcleos de CPU y memoria, y por tanto tienen la posibilidad de ofrecer recursos aislados a un mayor número de clientes, el número máximo de grupos de recursos de servidor también ha aumentado en SQL Server 2012, pasando de 20 a 64. Conjuntamente, estas características proporcionan un mejor entorno multiempresa al admitir el reparto de los recursos de proceso de un equipo entre un número mayor de grupos más aislados.

Para obtener más información acerca del Regulador de recursos de SQL Server 2012, vea [Regulador de recursos](http://msdn.microsoft.com/library/bb933866.aspx) (<http://msdn.microsoft.com/library/bb933866.aspx>) en los Libros en pantalla de SQL Server.

Referencias

[Regulador de recursos](#)

[Usar el Regulador de recursos](#) de Aaron Bertrand y Boris Baryshnikov

[Guía de programación para objetos de administración de SQL Server \(SMO\)](#)

[Blog del equipo de infraestructura de nube de SQL Server](#)

Para obtener más información:

<http://www.microsoft.com/sqlserver/>: Sitio web de SQL Server

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

<http://msdn.microsoft.com/es-es/sqlserver/>: Centro para desarrolladores de SQL Server

¿Le sirvió de ayuda este documento? Proporcionémos su opinión. Díganos, en una escala de 1 (poco útil) a 5 (excelente), cómo calificaría este documento y por qué lo valora con esta puntuación. Por ejemplo:

- ¿Lo valora alto debido a que tiene buenos ejemplos, capturas de pantalla excelentes, una redacción comprensible u otra razón?
- ¿Lo valora bajo debido a que sus ejemplos son escasos, las capturas de pantalla son borrosas o su redacción es poco clara?

Esta información nos ayudará a mejorar la calidad de las notas del producto que publicamos.

[Enviar comentarios.](#)