

WSUS - Windows Server Update Services

Свободная замена и мобильный вариант для засылки в разные отдаленные регионы где нет интернета: <http://www.wsusoffline.net/>

Ускоряем работу WSUS

Взято тут: <https://habrahabr.ru/post/329440/>

Настройка Internet Information Server

Даем больше памяти приложению.

Открываем «Диспетчер служб IIS», переходим в «Пулы приложений», выбираем WsusPool и открываем «Дополнительные параметры». Устанавливаем «Лимит выделенной памяти» в 0 и перезапускаем IIS.

Настройка TempDB в MSSQL

Как рекомендовано тут:

<https://support.microsoft.com/ru-ru/help/2154845/recommendations-to-reduce-allocation-contention-in-sql-server-tempdb-database>

«As a general rule, if the number of logical processors is less than or equal to 8, use the same number of data files as logical processors. If the number of logical processors is greater than 8, use 8 data files and then if contention continues, increase the number of data files by multiples of 4 (up to the number of logical processors) until the contention is reduced to acceptable levels or make changes to the workload/code.»

Т.е. если 8 или меньше логических процессоров - делаем по 1 файлу на каждый процессор. Если больше - умножаем на 4.

Открываем SQL Server Management Studio, далее «Базы данных» → «Системные базы данных» → tempdb, открываем свойства и выбираем «Файлы». Добавляем требуемое количество файлов, желательно разнести их по разным дискам. Начальный размер делаем у всех файлов одинаковым, автоувеличение делаем «С шагом по 10Мб, без ограничений».

Очистка базы WSUS

Очистка производится через «Параметры» → «Мастер очистки сервера» и желательно это делать регулярно, не реже раза в месяц. Если в процессе работы падает с ошибкой - увеличиваем кол-во памяти для IIS и в мастере очистки запускаем по одному пункту.

Если это не помогает, надо выполнить очистку напрямую в базе. Для этого необходимо

подключиться к экземпляру SQL Server инструментом Management Studio. Management Studio стала отдельным продуктом. Его можно скачать по этой ссылке. Если вы используете Windows Internal Database, необходимо поставить Management Studio на сервер с WSUS. Для подключения к экземпляру используется строка:

```
\\.\pipe\MICROSOFT##WID\tsql\query
```

Для очистки базы выполните 4 волшебные команды:

```
EXEC SUSDB.dbo.spDeclineExpiredUpdates;1
```

```
EXEC SUSDB.dbo.spDeclineSupersededUpdates;1
```

Для команды spCompressUpdate используется «обёртка»:

```
USE SUSDB
DECLARE @var1 INT, @curitem INT, @totaltodelete INT
DECLARE @msg nvarchar(200)
CREATE TABLE #results (Col1 INT) INSERT INTO #results(Col1)
EXEC spGetUpdatesToCompress
SET @totaltodelete = (SELECT COUNT(*) FROM #results)
SELECT @curitem=1
DECLARE WC Cursor FOR SELECT Col1 FROM #results
OPEN WC
FETCH NEXT FROM WC INTO @var1 WHILE (@@FETCH_STATUS > -1)
BEGIN SET @msg = cast(@curitem as varchar(5)) + '/' + cast(@totaltodelete as
varchar(5)) + ': Compressing ' + CONVERT(varchar(10), @var1) + ' ' +
cast(getdate() as varchar(30))
RAISERROR(@msg,0,1) WITH NOWAIT
EXEC spCompressUpdate @localUpdateID=@var1
SET @curitem = @curitem +1
FETCH NEXT FROM WC INTO @var1
END
CLOSE WC
DEALLOCATE WC
DROP TABLE #results
```

Такая же обертка для spDeleteUpdate:

```
USE SUSDB
DECLARE @var1 INT, @curitem INT, @totaltodelete INT
DECLARE @msg nvarchar(200)
CREATE TABLE #results (Col1 INT) INSERT INTO #results(Col1)
EXEC spGetObsoleteUpdatesToCleanup
SET @totaltodelete = (SELECT COUNT(*) FROM #results)
SELECT @curitem=1
DECLARE WC Cursor FOR SELECT Col1 FROM #results
OPEN WC
FETCH NEXT FROM WC INTO @var1 WHILE (@@FETCH_STATUS > -1)
BEGIN SET @msg = cast(@curitem as varchar(5)) + '/' + cast(@totaltodelete as
```

```
varchar(5)) + ': Deleting ' + CONVERT(varchar(10), @var1) + ' ' +  
cast(getdate() as varchar(30))  
RAISERROR(@msg,0,1) WITH NOWAIT  
EXEC spDeleteUpdate @localUpdateID=@var1  
SET @curitem = @curitem +1  
FETCH NEXT FROM WC INTO @var1  
END  
CLOSE WC  
DEALLOCATE WC  
DROP TABLE #results
```

Во время работы «обёрток» клиенты прекращают получать обновления. Вы можете в любой момент прервать выполнение скрипта без потери прогресса. Для того, чтобы продолжить процесс, не забудьте удалить временную таблицу:

```
DROP TABLE #results
```

Переиндексация базы

Для переиндексации базы используйте следующий скрипт:

```
/*  
***  
This sample T-SQL script performs basic maintenance tasks on SUSDB  
1. Identifies indexes that are fragmented and defragments them. For certain  
   tables, a fill-factor is set in order to improve insert performance.  
   Based on MSDN sample at  
http://msdn2.microsoft.com/en-us/library/ms188917.aspx  
   and tailored for SUSDB requirements  
2. Updates potentially out-of-date table statistics.  
*****  
**/  
  
USE SUSDB;  
GO  
SET NOCOUNT ON;  
  
-- Rebuild or reorganize indexes based on their fragmentation levels  
DECLARE @work_to_do TABLE (  
    objectid int  
    , indexid int  
    , pagedensity float  
    , fragmentation float  
    , numrows int  
)  
  
DECLARE @objectid int;  
DECLARE @indexid int;  
DECLARE @schemaname nvarchar(130);
```

```
DECLARE @objectname nvarchar(130);
DECLARE @indexname nvarchar(130);
DECLARE @numrows int
DECLARE @density float;
DECLARE @fragmentation float;
DECLARE @command nvarchar(4000);
DECLARE @fillfactorset bit
DECLARE @numpages int

-- Select indexes that need to be defragmented based on the following
-- * Page density is low
-- * External fragmentation is high in relation to index size
PRINT 'Estimating fragmentation: Begin. ' + convert(nvarchar, getdate(),
121)
INSERT @work_to_do
SELECT
    f.object_id
    , index_id
    , avg_page_space_used_in_percent
    , avg_fragmentation_in_percent
    , record_count
FROM
    sys.dm_db_index_physical_stats (DB_ID(), NULL, NULL , NULL, 'SAMPLED')
AS f
WHERE
    (f.avg_page_space_used_in_percent < 85.0 and
f.avg_page_space_used_in_percent/100.0 * page_count < page_count - 1)
    or (f.page_count > 50 and f.avg_fragmentation_in_percent > 15.0)
    or (f.page_count > 10 and f.avg_fragmentation_in_percent > 80.0)

PRINT 'Number of indexes to rebuild: ' + cast(@@ROWCOUNT as nvarchar(20))

PRINT 'Estimating fragmentation: End. ' + convert(nvarchar, getdate(), 121)

SELECT @numpages = sum(ps.used_page_count)
FROM
    @work_to_do AS fi
    INNER JOIN sys.indexes AS i ON fi.objectid = i.object_id and fi.indexid
= i.index_id
    INNER JOIN sys.dm_db_partition_stats AS ps on i.object_id = ps.object_id
and i.index_id = ps.index_id

-- Declare the cursor for the list of indexes to be processed.
DECLARE curIndexes CURSOR FOR SELECT * FROM @work_to_do

-- Open the cursor.
OPEN curIndexes

-- Loop through the indexes
WHILE (1=1)
BEGIN
```

```
FETCH NEXT FROM curIndexes
INTO @objectid, @indexid, @density, @fragmentation, @numrows;
IF @@FETCH_STATUS < 0 BREAK;

SELECT
    @objectname = QUOTENAME(o.name)
    , @schemaname = QUOTENAME(s.name)
FROM
    sys.objects AS o
    INNER JOIN sys.schemas as s ON s.schema_id = o.schema_id
WHERE
    o.object_id = @objectid;

SELECT
    @indexname = QUOTENAME(name)
    , @fillfactorset = CASE fill_factor WHEN 0 THEN 0 ELSE 1 END
FROM
    sys.indexes
WHERE
    object_id = @objectid AND index_id = @indexid;

IF ((@density BETWEEN 75.0 AND 85.0) AND @fillfactorset = 1) OR
(@fragmentation < 30.0)
    SET @command = N'ALTER INDEX ' + @indexname + N' ON ' + @schemaname
+ N'.' + @objectname + N' REORGANIZE';
ELSE IF @numrows >= 5000 AND @fillfactorset = 0
    SET @command = N'ALTER INDEX ' + @indexname + N' ON ' + @schemaname
+ N'.' + @objectname + N' REBUILD WITH (FILLFACTOR = 90)';
ELSE
    SET @command = N'ALTER INDEX ' + @indexname + N' ON ' + @schemaname
+ N'.' + @objectname + N' REBUILD';
PRINT convert(nvarchar, getdate(), 121) + N' Executing: ' + @command;
EXEC (@command);
PRINT convert(nvarchar, getdate(), 121) + N' Done.';
END

-- Close and deallocate the cursor.
CLOSE curIndexes;
DEALLOCATE curIndexes;

IF EXISTS (SELECT * FROM @work_to_do)
BEGIN
    PRINT 'Estimated number of pages in fragmented indexes: ' +
cast(@numpages as nvarchar(20))
    SELECT @numpages = @numpages - sum(ps.used_page_count)
    FROM
        @work_to_do AS fi
        INNER JOIN sys.indexes AS i ON fi.objectid = i.object_id and
fi.indexid = i.index_id
        INNER JOIN sys.dm_db_partition_stats AS ps on i.object_id =
```

```
ps.object_id and i.index_id = ps.index_id

    PRINT 'Estimated number of pages freed: ' + cast(@numpages as
nvarchar(20))
END
GO
--Update all statistics
PRINT 'Updating all statistics.' + convert(nvarchar, getdate(), 121)
EXEC sp_updatestats
PRINT 'Done updating statistics.' + convert(nvarchar, getdate(), 121)
GO
```

[windows](#), [server](#), [wsus](#), [windows server update services](#), [обновления](#)

From:

<https://wiki.rtzra.ru/> - RTzRa's hive

Permanent link:

<https://wiki.rtzra.ru/software/windows/wsus>

Last update: **2017/05/26 14:20**

