

<http://www.sqlservercentral.com/articles/SQL+Server+Cluster+Management/103090/>

Printed 2013/11/04 10:59AM

Managing MaxMemory Values on an Active-Active Two Node Cluster

By [Michael Lacy](#), 2013/10/10

Okay, you have your brand new, two node, Windows Cluster built, and SQL Server installed and running on an active-active cluster. When you bought the hardware, you loaded-up on memory with each node having 256GB of the fastest memory money can buy. You've configured your two instances with a max memory value of 245GB and everything runs great - until one day one of the nodes has an error that places both instances onto one node. Now you have two instances, each with a max memory value of 245GB running on one piece of hardware with 256GB of memory. 245GB + 245GB is more than 256GB. Ouch! You could take the easy way out and make sure that the max memory on each instance adds-up to less than 256GB, but that means that when everything is working well, you're wasting about half of the available memory.

How do you maximize the use of your installed memory, and yet automatically reconfigure your max memory value settings when an unexpected fail-over occurs? You just need to link your two SQL Server instances, create some empty text files, create a few of SQL Server Agent jobs, and you're good-to-go.

My Windows Cluster has been correctly built and drive dependencies are correctly defined for my two SQL Server instances. Here's my server hardware setup:

- Node1=MyServer1: Should host my production instance which is named: ProdInst\ProdDBs and has disk E: (a SAN attached disk)
- Node2=MyServer2: Should host my reporting instance which is named: ReptInst\ReptDBs and has disk H: (a SAN attached disk)

(I have a C: drive and other SAN drives for my T-logs and TempDB, but this process does not reference those drives.)

First, link the two SQL Server instances. Edit and run this code on each SQL Server, linking both instances to one another. This was run on my production instance to link to my reporting instance:

```
USE [master]
GO
EXEC master.dbo.sp_addlinkedserver @server = N'ReptInst\ReptDBs', @srvproduct=N'SQL Server'
```

This was run on my reporting instance to link to my production instance:

```
USE [master]
GO
EXEC master.dbo.sp_addlinkedserver @server = N'ProdInst\ProdDBs', @srvproduct=N'SQL Server'
```

Note: RPC should be set to "True" on each linked server.

Next, create an empty text file on each node. Create these on your SAN disks that will fail-over per your Windows Cluster Manager configuration. I'll call these my "SAN Test Files".

- On SAN Disk – for ProdInst\ProdDBs: **E:\DBA\ProdInst.txt**
- On SAN Disk – for ReptInst\ReptDBs: **H:\DBA\ReptInst.txt**

Then, create two SQL Server Agent jobs on each SQL Server instance. One job to set the max memory value to what you want it to be when just one instance is on the node, and a second SQL Server Agent job to set the max memory value to what you'd like it to be if both SQL Server instances are running on the same node. If just one instance is on a node, I want the max memory value set to 245GB, and if both instances are running on the same node, I want the max memory value for my ProdInst\ProdDBs instance to be 173GB and my ReptInst\ReptDBs to be 72GB.

Memory is usually expressed in MBs, not GBs, so I use the following values:

- 245GB = 250880MB
- 173GB = 177152MB
- 72GB = 73728MB

The jobs to set the max memory values are named: 'Set MaxMemory Value on PRODDBS to 245GB' and 'Set MaxMemory Value on REPTDBS to 245GB'. Here is the code:

```
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
GO
EXEC sys.sp_configure N'max server memory (MB)', N'250880'
GO
RECONFIGURE WITH OVERRIDE
GO
EXEC sys.sp_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE
GO
```

Here is the second job, named: 'Set MaxMemory Value on PRODDBS to 173GB':

```
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
GO
EXEC sys.sp_configure N'max server memory (MB)', N'177152'
GO
RECONFIGURE WITH OVERRIDE
GO
EXEC sys.sp_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE
GO
```

I have a third job named: 'Set MaxMemory Value on REPTDBS to 72GB':

```
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
GO
EXEC sys.sp_configure N'max server memory (MB)', N'73728'
GO
RECONFIGURE WITH OVERRIDE
GO
```

```
EXEC sys.sp_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE
GO
```

I need one more SQL Server Agent job. I'll call this job: "Alert and Set MaxMemory on Restart". It exists on each SQL Server instance and is scheduled to "Start automatically whenever the SQL Server Agent starts". It will perform 2 job steps:

1. Send an email with the instance name, and the node that it is running on.
2. Set the max memory values appropriately if both instances are running on the same node, or independently with just one instance on each node.

"Alert and set MaxMemory on Restart" job steps:

Here are the details of this last job.

Job step 1: (This code is basically the same on either instance, but must be edited for you database mail values) (*see edit comments*). This step sends an email with the instance name, and the node that it is running on.

```
-- Send an email with the instance name, and the node that it is running on.
USE master
DECLARE @importance AS VARCHAR(6)
DECLARE @body AS NVARCHAR(1000)
DECLARE @subject AS NVARCHAR(1000)
DECLARE @instanceName AS VARCHAR(100) DECLARE @nodeName AS NVARCHAR(100)
DECLARE @recipientsList VARCHAR(100) SELECT @recipientsList = 'myemail@myworkplace.com' -- Edit
SELECT @instanceName = @@SERVERNAME
SELECT @nodeName = CAST(SERVERPROPERTY('ComputerNamePhysicalNetBIOS') AS NVARCHAR(200))
SELECT @subject = '!!! ALERT !!! - Fail over happened for instance: '+@instanceName
SELECT @body = 'Fail over happened for instance: '+@instanceName + '. This instance is currently running on node: '+@nodeName
SELECT @importance = 'High'
EXEC msdb.dbo.sp_send_dbmail
    @profile_name = 'ProdDBs Profile', -- Edit for your database mail profile
    @recipients = 'myemail@myworkplace.com', -- Edit for your email operator profile
    @subject = @subject,
    @body = @body,
    @body_format = 'HTML',
    @importance=@importance
```

Job step 2 for my ProdInst\ProdDBs instance: This sets the appropriate MaxMemory values. If my ProdInst\ProdDBs and my ReptInst\ReptDBs are both on the same server then set ProdInst\ProdDBs to use 173GB of memory and set ReptInst\ReptDBs to use 72GB of memory.

```
-- Set appropriate MaxMemory values. If my ProdInst\ProdDBs and my ReptInst\ReptDBs are both on the
-- same server then set ProdInst\ProdDBs to use 173GB of memory and set ReptInst\ReptDBs to use
-- 72GB of memory.
SET NOCOUNT ON
DECLARE @FileName varchar(255)
DECLARE @File_Exists int
/* Edit below as necessary. You are testing to see if two instances are running on one node. So for my ProdInst\ProdDBs instance I war
SELECT @FileName='H:\DBA\ReptInst.txt' EXEC Master.dbo.xp_fileexist @FileName, @File_Exists OUT
IF @File_Exists = 1
BEGIN
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'max server memory (MB)', N'177152'
RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE
EXEC [REPTINST\REPTDBS].msdb.dbo.sp_start_job @job_name = 'Set MaxMemory Value on REPTDBS to 72GB'
END
ELSE
BEGIN
PRINT 'PRODINST\PRODDBS and REPTINST\REPTDBS are NOT running on the same node'
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'max server memory (MB)', N'250880'
RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE
EXEC [REPTINST\REPTDBS].msdb.dbo.sp_start_job @job_name = 'Set MaxMemory Value on REPTDBS to 245GB'
END
```

Job step 2 for my ReptInst\ReptDBs instance: This job sets memory values.

```
-- Set appropriate MaxMemory values. If my ProdInst\ProdDBs and my ReptInst\ReptDBs are both on the
-- same node then set ProdInst\ProdDBs to 173GB of memory and set ReptInst\ReptDBs to use 72GB
-- of memory.
SET NOCOUNT ON
DECLARE @FileName varchar(255)
DECLARE @File_Exists int
/* Edit below as necessary. You are testing to see if two instances are running on one node. So for my ReptInst\ReptDBs instance I wou
SELECT @FileName='E:\DBA\ProdInst.txt' EXEC Master.dbo.xp_fileexist @FileName, @File_Exists OUT
IF @File_Exists = 1
BEGIN
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'max server memory (MB)', N'73728'
RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE
EXEC [PRODINST\ProdDBs].msdb.dbo.sp_start_job @job_name = 'Set MaxMemory Value on PRODDBS to 173GB'
END
ELSE
BEGIN
PRINT 'REPTINST\REPTDBS and PRODINST\PRODSDBS are NOT running on the same node'
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'max server memory (MB)', N'250880'
RECONFIGURE WITH OVERRIDE
EXEC sys.sp_configure N'show advanced options', N'0' RECONFIGURE WITH OVERRIDE
EXEC [PRODINST\PRODSDBS].msdb.dbo.sp_start_job @job_name = 'Set MaxMemory Value on PRODDBS to 245GB'
END
```

That's it. Now your max memory values should always be correctly set, to allow you to use the maximum amount of memory, whether they're running alone on a node, or if

both instances are running on the same node.

Copyright © 2002-2013 Simple Talk Publishing. All Rights Reserved. [Privacy Policy](#), [Terms of Use](#), [Report Abuse](#).