

Performance Story about Traceflag T1118 and the cluster registry checkpoint service



Ewan Fairweather (Azure CAT) 13 Dec 2008 1:26 AM

1

I've just got back from a week working with a customer. The customer was a large retailer using BizTalk to perform processing over AS2. They had just installed a new QA environment and wanted the operational environment to be optimized for their application. I had a gotcha with SQL Clustering which I wasn't aware of and I wanted to share.

As with any BizTalk installation, I started from the ground up and switched their SQL Storage from Raid 5 (which is very bad for write performance - which BizTalk does a lot of) to RAID 10. Over time in the lab I've also found that changing the NTFS cluster size can gain performance improvements. Typically we use 64KB for data LUNS (which host the BizTalk Database MDF files) and 4KB NTFS cluster size for log file LUNS (LDF's).

Optimizing TEMPDB is another of the optimizations that we tend to do. Specifically splitting temp db into multiple equal size files, 1 per Logical Proc available on the server. To reduce contention further on SQL Server and prevent almost all single page allocations Trace Flag T1118 is then applied.

I used SQL 2005 configuration manager to make the change by opening the tool, clicking SQL Server 2005 Services and then double clicking SQL Server (instance name). In the SQL Server (Instance name) properties dialog box on the advanced tab I added the following text at the end of the existing string: -T1118. Crucially what I'd forgotten to include was the semi colon required. So what I'd actually done was change the following

```
...-lc:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\mastlog.ldf
```

to

```
...-lc:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\mastlog.ldf-T1118
```

In other words the SQL Server would not attempt to load a log file for the Master database with name mastlog.ldf-T1118. Still unbeknown that I had made this error we rebooted the cluster so an OS hotfix could take affect.

When the server rebooted SQL Server would not start and the following error was in the SQL Error log

```
Can't open file "c:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\mastlog.ldf -T1118"
```

We removed the T1118 from the Configuration Manager properties used previously and attempted to bring the SQL Service back online. It failed again with the same error in the error log. We examined the SQL configuration manager properties and the -T1118 is still appended to the Master log file name. We try this several times even going into the registry directly (HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\Parameters) to try and change it. But no matter what we do it stayed there.

Those of you who are cluster gurus reading this are probably laughing because you know what has happened. If SQL is clustered it is the Windows Clustering Service which controls its execution and manages based on the cluster config which node it will start on. In order that SQL Server functions correctly when failed over between servers, the cluster service stores a checkpoint registry on the cluster quorum disk.

In this case when the change was made and the cluster was rebooted, this incorrect configuration was stored on the quorum disk. Because changes to the checkpoint can only be made when the service (in this case SQL) is online the change I was making to remove T1118 was never getting saved and the incorrect config was consistently erasing our right config. Talk about frustrating.

Fortunately there is a work around on this, which one of my colleagues in the Clustering Support Team was able to help me out with.

1. Disable the cluster checkpoint for the specific registry key:

```
c:\> cluster res "SQL Server" /removecheck: "Software\Microsoft\Microsoft SQL Server\MSSQL\MSSQLSERVER"
```

2. Correct the trace flag using SQL configuration manager or the registry

3. Re-enable Cluster checkpoint for the specific registry key:

```
c:\> cluster res "SQL Server" /addcheck: "Software\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLSERVER"
```

Comments



Nick Heppleston 15 Dec 2008 9:15 AM #

Hi Ewan,

Many thanks for the excellent content. The whole NTFS cluster size is new to me - can you point me to any documents that discuss this topic in more detail so we can optimise out SQL Server data and log

partitions?

Cheers, Nick.