Database Backup Compression using SQL Server 2008 R2

Author : contactng

Database backup compression is a feature provided by Microsoft which reduces the native backup size by up to 66% depending upon the type of data stored in a database and similar other factors. By default backup compression is disabled as it has some limitations which needs to be looked as per environment setup and if suites one’s need then it can be enabled as default or a parameter can be added to backup script to take compressed backups on the go.

Backup Compression was first introduced in SQL Server 2008 (only for Enterprise Edition), now this feature is extended to SQL Server 2008 R2 for both Enterprise as well as Standard edition hence it is good to understand the basics of compression and how to make a full use of it,

Below is Microsoft link which explain the basics of it and where to start,

Backup Compression (SQL Server)

Following restrictions are applicable; hence one should ensure that it is not the case with their environment,

•Compressed and uncompressed backups cannot co-exist in a media set.

•Previous versions of SQL Server cannot read compressed backups.

•NTbackups cannot share a tape with compressed SQL Server backups.

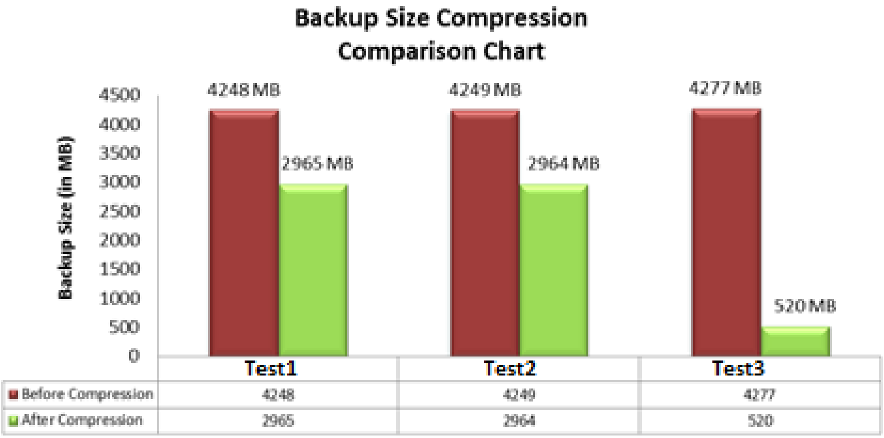
Now when a technology has pros, then there are cons as well, hence same applicable with Backup Compression, as it will improve IO, backup time, space requirement, etc. but it all comes at a price of performance impact, so if one’s database size is comparatively less like within 100GB or so and backups are happening in maintenance window, then all fine, else one may have to use Resource Governor to restrict the CPU usage, a nice article explaining how to configure it is provided by Microsoft at following link,

How to: Use Resource Governor to Limit CPU Usage by Backup Compression (Transact-SQL)

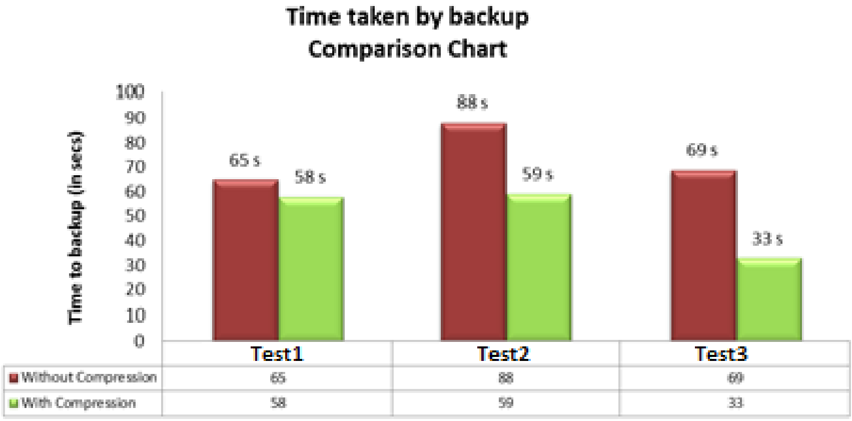
Now, question arises that how much compression is possible, answer is it depends on data type in database, I did test run on my servers, for some database compression was up to 90% but for some it is only 30%, below are details,

Backup Compression test analysis

- Backup size on disk is reduced by up to 90% for Test1 database and by 30% for Test 2 and 3 database, overall by combining savings of all databases we are able to achieve 50% space savings on disk.



- Time taken by backup after compression is considerably less which helps in completing backup activities early, currently due to limited database size I observed improvement in seconds but when database will grow better improvement can be observed,



Similarly, when trying to restore database, I observed improvement in total restore time, below is analysis of same,

Restore validation test analysis

- Database restore with compression is successfully tested by copying backup files to other server and restoring over there. 57% improvement is observed while restoring compressed database backup file over uncompressed backup file, it is good sign if large database needs a restore, also better restore time will improve overall Minimum Time To Recovery (MTTR) during disaster scenario

Restore Date	Restore Type	Restore DB Name	Time Taken by Restore (in Sec)	% Improvement in restore time
12/20/2013 8:39 AM	Full-Uncompressed	TestDB	75.615	57.80
12/20/2013 8:51 AM	Full-Compressed	TestDB	31.911	

Now, to check on performance impact, did ran the task manager performance and saw 10-20% higher CPU when using compression, it is normal as far as backups are taken in maintenance window else need to use resource manager.

Final thing is how to check the compression ratio, for same, found a useful blog which provided query to generate report and identify the compression ration,

[Trending Database Growth from Backups]()

Script is copied for reference as below,

```
SELECT

[database_name] AS “Database”,

DATEPART(month,[backup_start_date])
AS “Month”,

AVG([backup_size]/1024/1024)
AS “Backup Size MB”,

AVG([compressed_backup_size]/1024/1024)
AS “Compressed Backup Size MB”,

AVG([backup_size]/[compressed_backup_size])
AS “Compression Ratio”

FROM msdb.dbo.backupset

WHERE [database_name] =
N’master’
```

```
AND [type] =
‘D’

GROUP
BY [database_name],DATEPART(mm,[backup_start_date]);
```

Overall benefits of compressions are quite good which also improves overall MTTR (Minimum time to Recovery) thus recommended solution for environments with sufficient maintenance window or with use of resource governor.

Hope it helps.

New Year Wishes to all reading this first post mine in Year 2014!