Shrink - Incremental Shrink

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:27 AM PST

Contents

Issue

This TSG helps with shrinking data files in Azure SQL Database. Depending on the size of the database and data distribution in the data pages, a shrink operation might overwhelm the configured I/O capacity and exceed the transaction log size. The approach that is shown here allows you to shrink the data file in smaller batches inside a loop, and the loop will run until the target size has been reached.

Investigation / Analysis

The code below is copy from Yochanan's stored procedure that has been publicly shared at Incremental Shrink.txt. Please check there as well for any updated, newer version.

Mitigation

```
CREATE OR ALTER PROCEDURE usp_IncrementalShrink
@DesiredFileSize int=0,
@ShrinkChunkSize int=5,
@dbFileID int =0
as
begin
/*********************************************
Incremental Shrink for data file - SQL Server, Azure SQL, Azure Managed Instance
*********************************************/
/*-----------------------------------------------
Change Log:
   2022-07-12 - Change it from script to stored procedure
              - Add functionality to go through all data files
   2022-07-06 - more accurate current size validation.

*/-----------------------------------------------
set nocount on
declare @AllocatedSpaceMB int
declare @UsedSpaceMB int
declare @UnusedSpaceMB int
declare @ErrorIndication int=0
declare @dbFileType sysname
declare @lastSize int
declare @SqlCMD nvarchar(max)
declare @MSG nvarchar(100)
declare @iFileList table(i int)
declare @iTMP table(i int)
declare @iFileID int
declare @iCurrentSizeTarget int

set @MSG = convert(nvarchar,getdate())+' - Starting incremental shrink procedure'; raiserror(@msg,0,0) with no

/* @dbFileID=0 -> All Files, or actual data file ID */
insert into @iFileList select file_id from sys.database_files where type=0/*Rows*/ and (@dbFileID=0 or file_id

-- check if there is paused resumable index operation on this DB
-- existance of these types of operations block the shrink operation from reducing the file size
if (SELECT count(*) FROM sys.index_resumable_operations)>0 set @ErrorIndication=3

if @ErrorIndication=3  raiserror('[Error] Paused resumable index rebuild was detected, please abort or complet

/*Go throgh all files pending to be shrinked*/
WHILE (select count(*) from @iFileList)>0
Begin
        set @MSG = REPLICATE('-',50); raiserror(@msg,0,0) with nowait

        /*Iterate on specific file*/
        delete top (1) from @iFileList output  deleted.i into @iTMP
        select top 1 @iFileID=i from @iTMP

        set @MSG = 'Running shrink file on file ID = ' + CONVERT(varchar,@iFileID) +char(13) ; raiserror(@msg,

        SELECT
                @AllocatedSpaceMB = SIZE/128.0
                , @UsedSpaceMB = cast(fileproperty(name, 'SpaceUsed') AS int)/128.0
                , @UnusedSpaceMB = (SIZE/128.0) - cast(fileproperty(name, 'SpaceUsed') AS int)/128.0
        FROM sys.database_files
        WHERE file_id = @iFileID

        set @MSG = char(9)+'Information about file ID = ' + CONVERT(varchar,@iFileID) ; raiserror(@msg,0,0) wi
        set @MSG = char(9)+char(9)+'Allocated Space MB = ' + CONVERT(varchar,@AllocatedSpaceMB) ; raiserror(@m
        set @MSG = char(9)+char(9)+'Used Space MB = ' + CONVERT(varchar,@UsedSpaceMB) ; raiserror(@msg,0,0) wi
        set @MSG = char(9)+char(9)+'Unused Space MB = ' + CONVERT(varchar,@UnusedSpaceMB) ; raiserror(@msg,0,0

        set @lastSize = @AllocatedSpaceMB+1
        while @AllocatedSpaceMB > @DesiredFileSize /*check if we got the desired size*/ and @lastSize>@Allocat
        begin
                set @MSG = char(9)+char(9)+char(9)+convert(nvarchar,getdate()) + ' - Calling ShrinkFile' ; rai
```

```
                select @lastSize = size/128.0
                from sys.database_files
                where file_id=@iFileID

                /*Calculate next target size and make sure we do not go below 0*/
                set @iCurrentSizeTarget = @AllocatedSpaceMB-@ShrinkChunkSize
                set @iCurrentSizeTarget = iif(@iCurrentSizeTarget>0, @iCurrentSizeTarget,0)

                set @sqlCMD = N'dbcc shrinkfile('+cast(@iFileID as varchar(7))+','+ convert(nvarchar,@iCurrent
                --print @sqlCMD
                exec(@sqlCMD)

                select @AllocatedSpaceMB = size/128.0
                from sys.database_files
                where file_id=@iFileID

                set @MSG = char(9)+char(9)+char(9)+convert(nvarchar,getdate()) + ' - ShrinkFile completed. cur
        end

        delete from @iTMP
End

set @MSG = convert(nvarchar,getdate())+' - Finished incremental shrink procedure'; raiserror(@msg,0,0) with no
END
```

More Information

This is an alternative approach (older version of the Incremental Shrink.txt from above):

```
set nocount on
declare @CurrentFileSize int
declare @DesiredFileSize int
declare @ShrinkChunkSize int
declare @ActualSizeMB int
declare @ErrorIndication int
declare @dbFileID int = 1
declare @lastSize int
declare @SqlCMD nvarchar(max)
declare @MSG nvarchar(100)

/*set this values for the current operation, size is in MB*/
set @DesiredFileSize = 8
set @ShrinkChunkSize = 8
select @CurrentFileSize = size/128 from sysfiles where fileid=@dbFileID
select @ActualSizeMB = (sum(total_pages) / 128)  from sys.allocation_units

set @msg = 'Current File Size: ' + cast(@CurrentFileSize as varchar(10)) + 'MB'
raiserror(@msg,0,0) with nowait

set @msg = 'Actual used Size: ' + cast(@ActualSizeMB as varchar(10)) + 'MB'
raiserror(@msg,0,0) with nowait

set @msg = 'Desired File Size: ' + cast(@DesiredFileSize as varchar(10)) + 'MB'
raiserror(@msg,0,0) with nowait

set @msg = 'Interation shrink size: ' + cast(@ShrinkChunkSize as varchar(10)) + 'MB'
raiserror(@msg,0,0) with nowait

set @ErrorIndication =
case
   when @DesiredFileSize > @CurrentFileSize then 1
   when @ActualSizeMB > @DesiredFileSize then 2
   else 0 end

if @ErrorIndication=1  raiserror('[Error] Desired size bigger than current size',0,0) with nowait
if @ErrorIndication=2  raiserror('[Error] Actual size is bigger than desired size',0,0) with nowait
if @ErrorIndication=0  raiserror('Desired Size check - OK',0,0) with nowait

set @lastSize = @CurrentFileSize+1
while @CurrentFileSize > @DesiredFileSize /*check if we got the desired size*/ and @lastSize>@CurrentFileSize
begin
   set @msg = cast(getdate() as varchar(100)) + ' - Iteration starting'
   raiserror(@msg,0,0) with nowait

   select @lastSize = size/128 from sysfiles where fileid=@dbFileID
   set @sqlCMD = 'dbcc shrinkfile('+cast(@dbFileID as varchar(7))+','+ cast(@CurrentFileSize-@ShrinkChunkSize
   exec(@sqlCMD)

   select @CurrentFileSize = size/128 from sysfiles where fileid=@dbFileID
   set @msg = cast(getdate() as varchar(100)) + ' - Iteration completed. current size is: ' + cast(@CurrentFil
   raiserror(@msg,0,0) with nowait
end
print 'Done'
```

How good have you found this content?