









DBA

SQL Server Administration

Part4



Mohammad Reza Gerami
mrgerami@aut.ac.ir
gerami@virasec.ir



Auto Growth and Sizing of Transactions log

Auto Growth



SQL Server expands the size of a database (data and log file) based on the auto growth setting of an individual file to avoid space issues in the existing transactions. Once an auto-growth event occurs, SQL Server holds up transaction processing for the time being. We should monitor and manage the auto growth setting for a database. If the auto-growth event takes a little longer time, it might influence database performance and availability.

زمانیکه که SQL Server برای انجام یک عملیات (مانند Insert کردن رکورد در دیتابیس) فضای کافی ندارد و نیاز به فضای جدیدی در دیسک دارد این رویداد اتفاق می افتد و فضای جدیدی در دیسک برای فایل دیتابیس رزرو می شود.

Auto Growth



In case of any auto-growth event, SQL Server looks for additional space in the disk where the data or log file exists. If we have a very frequent auto-growth event, it causes physical fragmentation as well. The physical fragmented database takes a long time to complete the transactions. We should take action to minimize the auto-growth events for highly transactional databases.

By default, Auto Growth is set to grow 10% percent of existing file size in the SQL Server database. It might be suitable for small databases, however, if you look at the perspective of a large database file, it is not the right configuration.

Auto Growth



Database Properties - model

Select a page

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties

Script ? Help

Database name: model

Owner: sa

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Size (MB)	Autogrowth / Maxsize	Page
modeldev	ROWS...	PRIMARY	8	By 10 percent, Unlimited	C...
modellog	LOG	Not Applicable	8	By 10 percent, Unlimited	C...

Auto Growth

Let's look at the following examples. In example 1, SQL Server needs to expand the data file by 100 GB while in example 2, SQL Server expands data file by 10 GB. It will take a long time to expand the data file by 100 GB, and it might put additional load on the database as well.



Example 1	Example 2
Database File Size – 1 TB	Database File Size – 100 GB
Auto Growth – By 10%	Auto Growth – By 10%
Auto Growth File size increment: 100 GB	Auto Growth File size increment: 10 GB

Auto Growth



We should take the following approach to avoid auto-growth events.

- ❖ Proactively grow data and log file size for the SQL Server database. If database files have sufficient free space, it will not cause any auto growth
- ❖ Monitor auto growth events and set appropriate size for the auto growth in fixed MB instead of percentage (%) growth

Auto Growth



Suppose we set the data file auto growth to 512 MB instead of a 10% setting. In the following table, we can see that auto growth does not require any dependency on the data file size if we use fixed MB growth.

Example 1	Example 2
Database File Size – 1 TB	Database File Size – 100 GB
Auto Growth – By 512 MB	Auto Growth – By 512 MB
Auto Growth File size increment: 512 MB	Auto Growth File size increment: 512 MB



- ❖ در پایگاه داده و سیستم هایی که Performance بسیار مهم است تنظیم صحیح Auto Growth بسیار مهم است.
- ❖ مانیتورینگ وضعیت Auto Growth دیتابیس در بازه های روزانه و ماهانه برای رسیدن به مقدار درست پارامتر Auto Growth بسیار مهم است.
- ❖ گاهی ممکن است در طی تغییرات سیستم (افزودن امکانات به سیستم نرم افزاری) نمودار رشد حجم دیتابیس نیز تغییر کند که باید پارامتر Auto Growth نیز با این تغییرات تطابق داشته باشد و تغییر داده شود.
- ❖ رخداد Auto Growth یک رخداد پرهزینه است که تاثیر مستقیمی روی کارایی سیستم خواهد داشت، بنابراین درست تنظیم نشدن آن ممکن است آثار سوء زیادی روی کارایی سیستم داشته باشد.



- ❖ در پایگاه داده و سیستم هایی که Performance بسیار مهم است تنظیم صحیح Auto Growth بسیار مهم است.
- ❖ مانیتورینگ وضعیت Auto Growth دیتابیس در بازه های روزانه و ماهانه برای رسیدن به مقدار درست پارامتر Auto Growth بسیار مهم است.
- ❖ گاهی ممکن است در طی تغییرات سیستم (افزودن امکانات به سیستم نرم افزاری) نمودار رشد حجم دیتابیس نیز تغییر کند که باید پارامتر Auto Growth نیز با این تغییرات تطابق داشته باشد و تغییر داده شود.
- ❖ رخداد Auto Growth یک رخداد پرهزینه است که تاثیر مستقیمی روی کارایی سیستم خواهد داشت، بنابراین درست تنظیم نشدن آن ممکن است آثار سوء زیادی روی کارایی سیستم داشته باشد.

Auto Growth



❖ عدد ثابتی برای تنظیم Auto Growth وجود ندارد و مقدار آن بسته به بار کاری سیستم متغیر است، و باید طوری انتخاب شود که Auto Growth دائماً اتفاق نیفتد و از طرفی زمان اجرایش آنقدر طولانی نباشد که باعث بلاک سایر تراکنش ها بشود، بهتر است Auto Growth به شکل درصدی انتخاب نشود چون سر بار محاسباتی دارد و عدد ثابت آنقدر کوچک نباشد که با مشکل Log fragmentation مواجه شوید.

<https://www.comparitech.com/net-admin/sql-server-monitoring-tools/>

Auto Growth



Lab Demo

- ❖ Create a new Database with normal size
 - ❖ Import a file lower than DB size
 - ❖ Check the Log and Analysis the Growing
 - ❖ Report
 - ❖ sp_helpdb [yourDBname]
 - ❖ dbcc sqlperf (logspace)
 - ❖ Import another file bigger than your free space
 - ❖ Check the Log and Analysis the Growing
- ❖ Fix this problem with good decision about DB size



Transactions log

Transactions log



Every SQL Server database has a transaction log that records all transactions and the database modifications that are made by each transaction. The transaction log is a **critical component** of the database and, if there is a system failure, the transaction log might be required to bring your database back to a consistent state.

Warning

Never delete or move this log unless you fully understand the ramifications of doing so.

Transactions log



Transaction Log Logical Architecture

The SQL Server transaction log operates logically as if the transaction log is a string of log records.

Each log record is identified by a **log sequence number (LSN)**.

Each new log record is written to the logical end of the log with an LSN that is higher than the LSN of the record before it.

Log records are stored in a serial sequence as they are created such that if LSN2 is greater than LSN1, the change described by the log record referred to by LSN2 occurred after the change described by the log record LSN1.

Transactions log

Transaction Log Logical Architecture



پسوند LogFile در پایگاه داده ldf است.

LogFile ها در پایگاه داده حاوی Log Record های ایجاد شده به ازای انجام هر عملیات که منجر به تغییر Data شود می باشد.

مکانیسم Transaction و انجام دسته ای دستورات TSQL توسط این نوع از فایلها در یک پایگاه داده انجام می شود. نوع دسترسی در LogFile بصورت ترتیبی می باشد و بنابراین با وجود امکان تعریف چند LogFile برای پایگاه داده، افزایش تعداد این نوع فایل در یک پایگاه داده هیچ کمکی به افزایش کارایی نخواهد کرد. اما بهتر است به منظور بالا بردن امنیت داده و کارایی، این فایلها بر روی دیسک جداگانه نسبت به DataFile ها ذخیره و نگهداری شوند. در ساختار فیزیکی LogFile از page استفاده نمی شود و شامل رکوردهایی است که به ازاء هر تغییر در پایگاه داده درج می شود.

Transactions log



بررسی معماری منطقی LogFile

ذخیره داده ها به صورت sequential
ثبت یک یا چند رکورد به ازای هر عملیات تغییر
ثبت LSN - Log Sequence Number به ازای هر رکورد که شناسه آن محسوب می شود.
وجود Transaction ID یکسان به ازای گروههای یکسان
ثبت Before Image : و After Image به ازای هر تغییر مقدار قدیمی و جدید را درج می کند.

DBCC LOGINFO;

Transactions log



DBCC LOGINFO;

با اجرای دستور بالا لیستی از LogFile ها با ID یکسان نمایش داده می شود. علت وجود چندین LogFile با ID یکسان این است که SQL SERVER با افزایش اندازه LogFile، یک LogFile جدید ایجاد می کند که اینکار بصورت مجازی انجام می شود. این مورد در شکل زیر به خوبی قابل مشاهده است

	FileId	FileSize	StartOffset	FSeqNo	Status	Parity	CreateLSN
1	2	253952	8192	29	0	64	0
2	2	253952	262144	30	2	64	0
3	2	253952	516096	0	0	0	0
4	2	278528	770048	0	0	0	0

Transactions log



SP_HELPFILE

در مورد بالا برای رسیدن به این مورد که پایگاه داده دارای تنها **LogFile** می باشد کفایت با اجرای دستور SP_HELPFILE لیست فایل های پایگاه داده را بررسی نماییم.

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	vsfs_academy	1	C:\Program Files (x86)\Microsoft SQL Server\MSSQ...	PRIMARY	1280 KB	Unlimited	1024 KB	data only
2	vsfs_academy_log	2	C:\Program Files (x86)\Microsoft SQL Server\MSSQ...	NULL	1024 KB	2147483648 KB	10%	log only

Transactions log



Virtual log file (VLF) creation follows this method:

- If the next growth is less than 1/8 of current log physical size, then create 1 VLF that covers the growth size (Starting with SQL Server 2014 (12.x))
- If the next growth is more than 1/8 of the current log size, then use the pre-2014 method:
 - ❖ If growth is less than 64MB, create 4 VLFs that cover the growth size (e.g. for 1 MB growth, create four 256KB VLFs)
 - ❖ If growth is from 64MB up to 1GB, create 8 VLFs that cover the growth size (e.g. for 512MB growth, create eight 64MB VLFs)
 - ❖ If growth is larger than 1GB, create 16 VLFs that cover the growth size (e.g. for 8 GB growth, create sixteen 512MB VLFs)

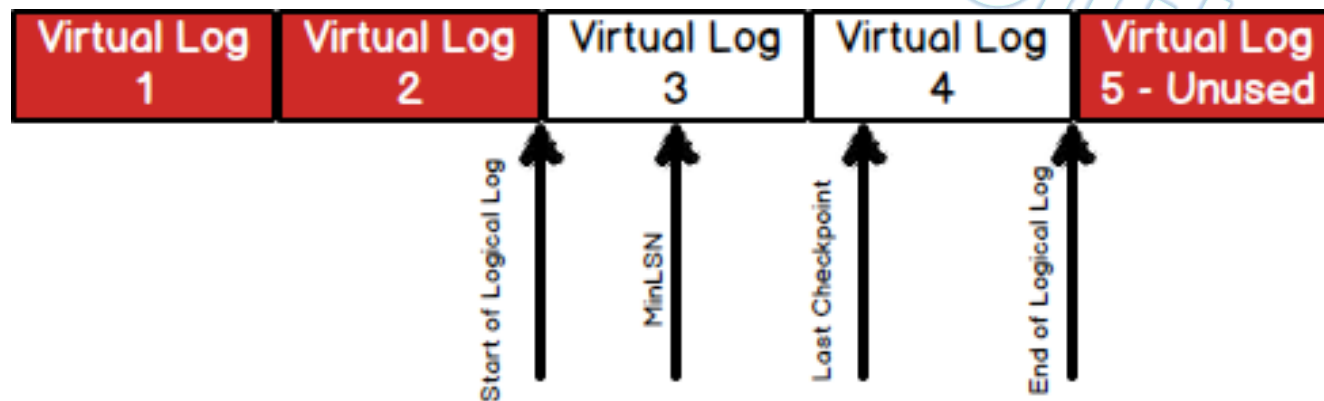
Transactions log



A log record is no longer needed in the transaction log if all of the following are true:

- ❖ The transaction of which it is part has committed
- ❖ The database pages it changed have all been written to disk by a checkpoint
- ❖ The log record is not needed for a backup (full, differential, or log)
- ❖ The log record is not needed for any feature that reads the log (such as database mirroring or replication) [1]

Logical log is an active part of the transaction log. A Log Sequence Number (LSN) identifies every transaction in the transaction log. The MinLSN is the starting point of the oldest active transaction in the online transaction log



Transactions log



Recommendations

Following are some general recommendations when you are working with transaction log files:

- The automatic growth (autogrow) increment of the transaction log, as set by the `FILEGROWTH` option, must be large enough to stay ahead of the needs of the workload transactions. The file growth increment on a log file should be sufficiently large to avoid frequent expansion. A good pointer to properly size a transaction log is monitoring the amount of log occupied during:
 - The time required to execute a full backup, because log backups cannot occur until it finishes.
 - The time required for the largest index maintenance operations.
 - The time required to execute the largest batch in a database.

Transactions log



When setting **autogrow** for data and log files using the `FILEGROWTH` option, it might be preferred to set it in **size** instead of **percentage**, to allow better control on the growth ratio, as percentage is an ever-growing amount.

- Keep in mind that transaction logs cannot leverage Instant File Initialization, so extended log growth times are especially critical.
 - As a best practice, do not set the `FILEGROWTH` option value above 1,024 MB for transaction logs.
- The default values for `FILEGROWTH` option are:

Version

Starting with SQL Server 2016 (13.x)

Starting with SQL Server 2005 (9.x)

Prior to SQL Server 2005 (9.x)

Default values

Data 64 MB. Log files 64 MB.

Data 1 MB. Log files 10%.

Data 10%. Log files 10%.

Transactions log



maintain a transaction log in SQL Server

A transaction log maintenance is important task in SQL Server administration. Monitoring is recommended on daily basis or even more often is a SQL Server database has high amount of traffic. The transaction log space can be monitored by using the DBCC SQLPREF command:

```
DBCC SQLPERF(LOGSPACE);  
GO
```

	Database Name	Log Size (MB)	Log Space Used (%)	Status
1	master	0.7421875	53.68421	0
2	tempdb	0.4921875	50.39682	0
3	model	1.117188	41.95804	0
4	msdb	2.492188	31.97492	0
5	IODB	0.9921875	43.06102	0
6	vsfs_ts	0.9921875	55.61024	0
7	vsfs_academy	0.9921875	41.19094	0
8	malaria	99.99219	0.6221189	0
9	IODB1	1.992188	24.87745	0

Transactions log

```
DBCC SQLPERF(LOGSPACE);  
GO
```



	Database Name	Log Size (MB)	Log Space Used (%)	Status
1	master	0.7421875	53.68421	0
2	tempdb	0.4921875	50.39682	0
3	model	1.117188	41.95804	0
4	msdb	2.492188	31.97492	0
5	IODB	0.9921875	43.06102	0
6	vsfs_ts	0.9921875	55.61024	0
7	vsfs_academy	0.9921875	41.19094	0
8	malaria	99.99219	0.6221189	0
9	IODB1	1.992188	24.87745	0

- *Database Name* – Name of the database for the log statistics displayed
- *Log Size (MB)* – Current size allocated to the log. This value is always smaller than the amount originally allocated for log space because the Database Engine reserves a small amount of disk space for internal header information
- *Log Space Used (%)* – Percentage of the log file currently occupied with transaction log information
- *Status* – Status of the log file. Always 0 [3]

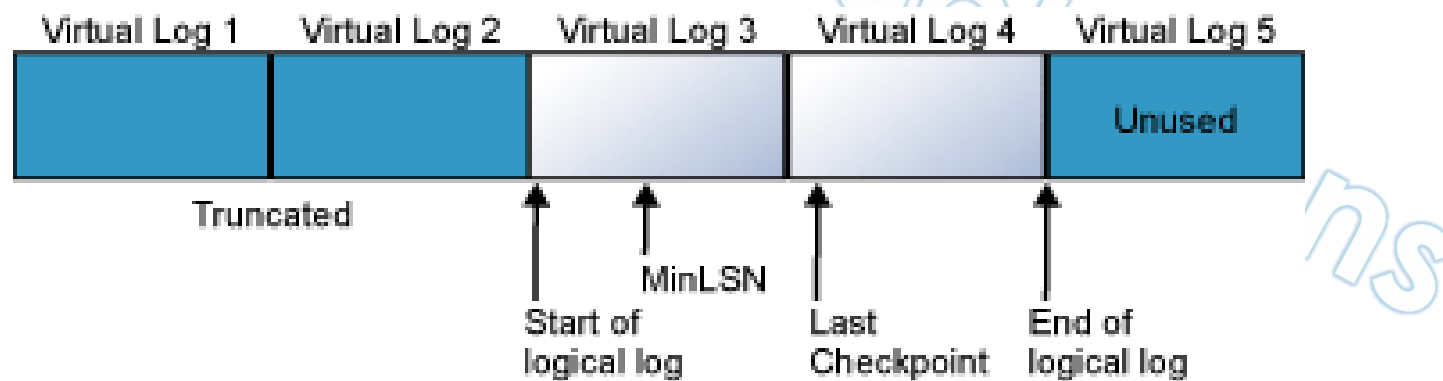
Transactions log



Transaction Log Logical Architecture

معماری فیزیکی Log File : درون Log File از تعدادی Virtual Log File یا VLF تشکیل شده است که داخل آن اطلاعات مربوط به Log Record ها ایجاد می شود. ایجاد VLF بستگی به Initial Size و همچنین نحوه رشد Log File دارد که در طی این جلسه درباره این باره بحث شد.

معماری منطقی Log File : کلیه تغییراتی که در SQL Server رخ می دهد چنانچه بخواهد به صورت Transactional انجام شود باید Log Record های مربوط به آن در Log File ثبت گردد



Transactions log



Transaction Log Logical Architecture

بررسی Buffer Pool: یکی از بزرگترین بخش‌های حافظه RAM متعلق به SQL Server است که محل قرار گیری Data Page، Index Page ها و ... می‌باشد.
برای مشاهده قسمت‌های مختلف حافظه متعلق به SQL Server می‌توانید از دستور زیر استفاده کنید

```
DBCC MEMORYSTATUS  
GO
```

زمانیکه شما یک کوئری جهت اجرا به DB Engine ارسال می‌کنید چنانچه Page های مورد انتظار این کوئری در Buffer Pool موجود باشد طی عملیات Logical Read توسط DB Engine مورد استفاده قرار می‌گیرد و اگر این Page ها در Buffer Pool موجود نباشد SQL Server طی عملیات Physical IO آنها را از Disk خوانده و به Buffer Pool منتقل می‌کند.

Transactions log



The transaction log should be backed up on the regular basis to avoid the auto growth operation and filling up a transaction log file. Space in a transaction log can be truncated (cleared) through SQL Server Management Studio by selecting Transaction Log as backup type or through CLI by executing the following command:

```
BACKUP LOG vsfs_Academy  
TO DISK = 'Q:\vsfs_academy.TRN'  
GO
```

Notes:

- 1.your database must be in full recovery mode
- 2.First you must take a backup and then backup log file.

Transactions log

LAB Demo



Vira Security Solutions

Transactions log



LAB Demo

Create a Test Database to test fn_dblog

```
--Create DB.  
USE [master];  
GO
```

```
CREATE DATABASE ReadingDBLog;  
GO
```

```
-- Create tables.  
USE ReadingDBLog;  
GO
```

```
CREATE TABLE [Location] (  
    [Sr.No] INT IDENTITY,  
    [Date] DATETIME DEFAULT GETDATE (),  
    [City] CHAR (25) DEFAULT 'Tehran');
```

```
USE ReadingDBLog;
```

```
GO SELECT COUNT(*) FROM fn_dblog(null,null)
```

Transactions log



```
USE ReadingDBLog;  
GO  
SELECT [Current LSN],  
       [Operation],  
       [Transaction Name],  
       [Transaction ID],  
       [Transaction SID],  
       [SPID],  
       [Begin Time]  
FROM fn_dblog(null,null
```