Log transakcyjny: rozmiar, shrink, zapchanie

Poziom: średni **Dotyczy: SQL Server 2016-2022** **Czas: ~25 min**

Diagnoza, dlaczego plik logu transakcyjnego rośnie mimo backupów, kiedy shrink jest bezpieczny, a kiedy jest ostatecznością, oraz jak wygląda odbudowa uszkodzonego pliku LDF.

Skróty użyte na tej stronie (6)

Skrót	Co znaczy
DBCC	Database Console Commands: rodzina poleceń MSSQL do sprawdzania i naprawy spójności bazy, na przykład DBCC CHECKDB
LDF	Log Data File: plik dziennika transakcji bazy MSSQL
RPO	Recovery Point Objective: ile danych wolno stracić przy awarii
SET	Switch Embedded Teaming: zespół kart sieciowych zbudowany w przełączniku wirtualnym Hyper-V; łączy dwie karty w jedno łącze bez ustawień po stronie przełącznika fizycznego, więc bez LACP
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
T-SQL	Transact-SQL: dialekt SQL używany w Microsoft SQL Server, wywodzący się z języka zapytań oryginalnego silnika Sybase

Fiszka

TYLKO ODCZYT

Sprawdzenie zajętości logu we wszystkich bazach: DBCC SQLPERF

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

Sprawdzamy rozmiar i procent zajętości pliku logu dla każdej bazy na instancji naraz — pierwsze zapytanie, kiedy dysk z logami zaczyna się zapełniać, a nie wiemy jeszcze, która baza jest winowajcą.

Przykład:

```
Database Name   Log Size (MB)   Log Space Used (%)   Status
--------------- --------------- -------------------- --------
master          19.99           38.12                0
CDN_Zielona          2048.00         94.87                0
CDN_Zielona_TEST          64.00           12.30                0

(3 rows affected)
```

DBCC = Database Console Commands: rodzina poleceń MSSQL do sprawdzania i naprawy spójności bazy, na przykład DBCC CHECKDB

TYLKO ODCZYT

Sprawdzenie przyczyny zapchania logu: log_reuse_wait_desc

```
SELECT name AS NazwaBazy, log_reuse_wait_desc AS PrzyczynaOczekiwania
FROM sys.databases;
```

Sprawdzamy, dlaczego dana baza nie zwalnia miejsca w logu mimo wykonywanych backupów — kolumna `log_reuse_wait_desc` widoku `sys.databases` mówi wprost, na co silnik czeka, zanim uzna wolne miejsce za odzyskane. To pierwsze zapytanie w każdej sytuacji „log rośnie, backupy logu idą, a miejsce się nie zwalnia".

Przykład:

```
NazwaBazy   PrzyczynaOczekiwania
----------- --------------------
master      NOTHING
CDN_Zielona      REPLICATION
CDN_Zielona_TEST      NOTHING

(3 rows affected)
```

ZMIENIA SYSTEM

Shrink pliku logu do zadanego rozmiaru: DBCC SHRINKFILE

```
USE NazwaBazyDanych;
GO
DBCC SHRINKFILE (NazwaBazyDanych_Log, 1);
GO
```

Zmniejszamy fizyczny rozmiar pliku logu do rozmiaru podanego w megabajtach — tutaj do 1 MB, czyli praktycznie minimum. Ta operacja działa wyłącznie na już nieużywanej, wolnej przestrzeni logu; nie da się zmniejszyć logu poniżej ilości miejsca faktycznie zajętego przez aktywne transakcje.

> (gdzie `NazwaBazyDanych_Log` to logiczna nazwa pliku logu, odczytana zapytaniem z sekcji Rozpoznanie instancji i baz — nie nazwa pliku fizycznego `.ldf`)

DBCC = Database Console Commands: rodzina poleceń MSSQL do sprawdzania i naprawy spójności bazy, na przykład DBCC CHECKDB

ZMIENIA SYSTEM

Wygenerowanie komend shrink dla wszystkich plików na instancji

```
SELECT
    'USE [' + d.name + N']' + CHAR(13) + CHAR(10)
    + 'DBCC SHRINKFILE (N''' + mf.name + N''', 0, TRUNCATEONLY)'
    + CHAR(13) + CHAR(10) + CHAR(13) + CHAR(10)
FROM sys.master_files mf
JOIN sys.databases d ON mf.database_id = d.database_id;
```

Generujemy gotowe komendy `DBCC SHRINKFILE` dla wszystkich plików na wszystkich bazach naraz, zamiast pisać je ręcznie bazę po bazie — wynik kopiujemy z okna rezultatów i wklejamy do nowego okna zapytania, gdzie uruchamiamy go jako zwykły skrypt T-SQL. Dopisanie warunku `WHERE d.database_id > 4` na końcu ogranicza wynik do baz użytkownika, pomijając cztery bazy systemowe.

Przykład:

```
(No column name)
------------------------------------------------------
```

```
USE [CDN_Zielona]
DBCC SHRINKFILE (N'CDN_Zielona_log', 0, TRUNCATEONLY)

USE [CDN_Zielona_TEST]
DBCC SHRINKFILE (N'CDN_Zielona_TEST_log', 0, TRUNCATEONLY)

(2 rows affected)
```

Uważaj: shrink z opcją `TRUNCATEONLY` zwalnia miejsce niezajęte na końcu pliku bez przenoszenia danych — bezpieczniejszy i szybszy niż pełny shrink, ale nie zmniejsza pliku, jeśli wolna przestrzeń nie leży fizycznie na jego końcu.

DBCC = Database Console Commands: rodzina poleceń MSSQL do sprawdzania i naprawy spójności bazy, na przykład DBCC CHECKDB · **T-SQL** = Transact-SQL: dialekt SQL używany w Microsoft SQL Server, wywodzący się z języka zapytań oryginalnego silnika Sybase ·**join** = połącz dwa pliki tekstowe po wspólnej kolumnie, jak złączenie tabel w bazie danych

Z życia wzięte

Z ŻYCIA WZIĘTE

W przedostatni dzień miesiąca dysk z logami na sql01 nagle robi się ciasny, mimo że backup logu CDN_Zielona leci co godzinę jak zawsze. Sprawdzamy log_reuse_wait_desc i widzimy wartość REPLICATION, chociaż nikt z zespołu nie pamięta, żeby ktokolwiek konfigurował replikację na tej bazie. Okazuje się, że rok wcześniej ktoś testował replikację transakcyjną do raportowania i odpiął ją niepełnie, zostawiając widmowego subskrybenta w logu. Po wykonaniu sp_removedbreplication log natychmiast zaczyna się zwalniać, tuż przed najgorętszymi dniami zamknięcia miesiąca.

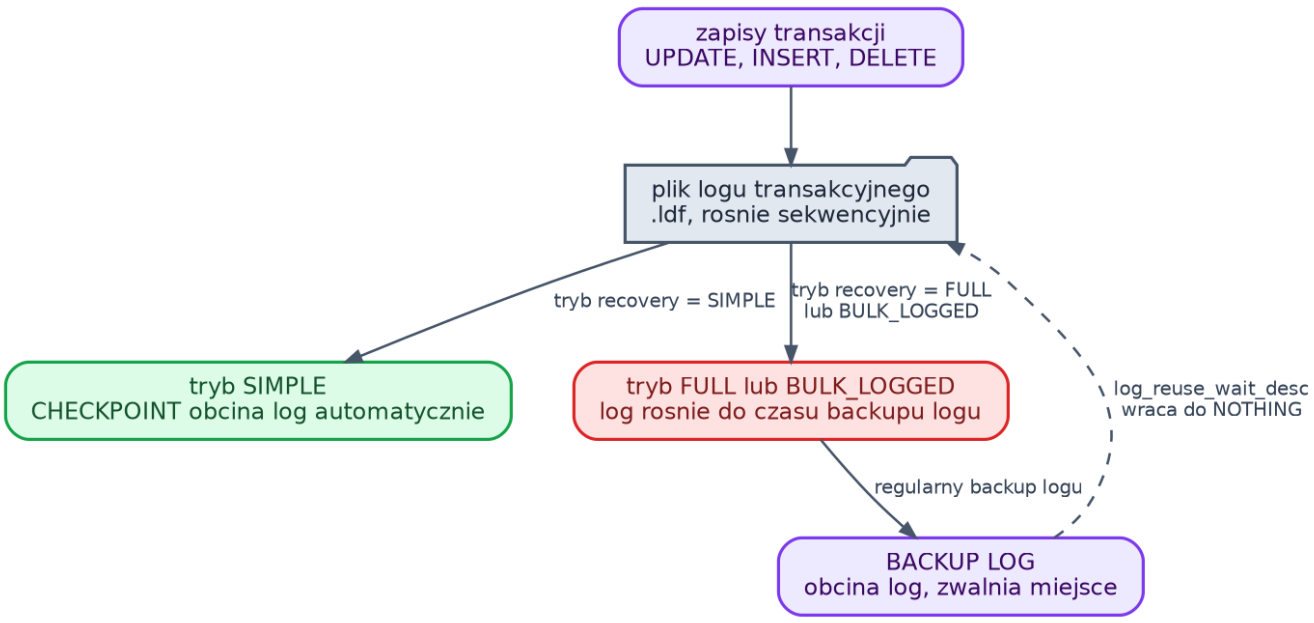
Po co to istnieje

Pojęcie: ponowne użycie loga (log reuse). Log transakcyjny działa jak pierścień — silnik zapisuje do niego sekwencyjnie, a miejsce zajęte przez najstarsze rekordy może zostać ponownie wykorzystane dopiero wtedy, gdy nie są już potrzebne do niczego: ani do odzyskiwania po awarii, ani do backupu logu, ani do żadnego mechanizmu, który je jeszcze czyta. Kolumna `log_reuse_wait_desc` opisuje właśnie to — na co silnik czeka, zanim uzna dany fragment logu za bezpieczny do nadpisania.

Wartość `log_reuse_wait_desc`	Co oznacza
`NOTHING`	log może być swobodnie zawracany, nic go nie wstrzymuje
`CHECKPOINT`	oczekiwanie na najbliższy checkpoint — zwykle chwilowe
`LOG_BACKUP`	baza w trybie `FULL` lub `BULK_LOGGED` czeka na backup logu, który go obetnie
`ACTIVE_TRANSACTION`	trwa długa, niezatwierdzona transakcja trzymająca stare rekordy logu
`REPLICATION`	agent dystrybucji albo log reader replikacji nie odczytał jeszcze wszystkich rekordów
`DATABASE_MIRRORING` / `AVAILABILITY_REPLICA`	replika mirroringu albo Always On nie potwierdziła jeszcze odebrania rekordów
`BACKUP`	trwa aktywny backup pełny lub różnicowy bazy

Jak to działa

Baza pracująca w trybie `FULL` albo `BULK_LOGGED`, opisanym na stronie Architektura MSSQL: instancja, baza, pliki MDF/NDF/LDF, nie zwalnia miejsca w logu samym `CHECKPOINT` — jedynym sposobem na obcięcie logu jest regularny backup logu, opisany na stronie Backup baz danych. Log, który rośnie mimo wykonywanych backupów, prawie zawsze oznacza jedną z pozostałych przyczyn z tabeli wyżej — najczęściej porzuconą, nieistniejącą już konfigurację replikacji, która wciąż jest zarejestrowana jako subskrybent logu.



Tryb recovery decyduje, czy log obcina CHECKPOINT, czy tylko BACKUP LOG (kliknięcie otwiera pełny rozmiar)

Naprawa zapchanego logu przy replikacji

Kiedy `log_reuse_wait_desc` pokazuje `REPLICATION`, mimo że sama replikacja została już odpięta po stronie aplikacyjnej, silnik wciąż czeka na potwierdzenie odczytu przez log reader. Resetujemy tę informację:

```
EXEC sp_repldone @xactid = NULL, @xact_segno = NULL, @numtrans = 0, @time = 0, @reset = 1;
```

Jeżeli to nie pomaga, bo replikacja została skonfigurowana i porzucona bez czystego rozpięcia, usuwamy jej ślad z bazy w całości:

```
USE NazwaBazy;
GO
EXEC sp_removedbreplication N'NazwaBazy';
GO
CHECKPOINT;
GO
```

Dodatkowo wyłączamy flagę publikacji na poziomie bazy, jeśli nadal jest ustawiona:

```
USE master;
GO
EXEC sp_replicationdboption @dbname = N'NazwaBazy', @optname = N'publish', @value = N'false';
GO
```

Po wykonaniu tych trzech kroków ponownie sprawdzamy `log_reuse_wait_desc` — wartość powinna wrócić do `NOTHING`.

Krok po kroku: bezpieczny shrink logu po jednorazowym skoku rozmiaru

Shrink logu jest operacją ostateczną, nie rutynową — plik logu, który raz urósł do dużego rozmiaru pod wpływem jednorazowej operacji masowej, i tak urośnie do podobnego rozmiaru ponownie przy

następnej takiej operacji, więc regularne zmniejszanie go tylko powoduje ciągłe, kosztowne powiększanie od nowa (autogrowth). Sięgamy po shrink wyłącznie po jednorazowym, nietypowym skoku rozmiaru, którego nie spodziewamy się powtórzyć.

1. Wykonujemy pełny backup bazy, zanim cokolwiek zmienimy — opisane na stronie Backup baz danych.
2. Sprawdzamy bieżący tryb recovery, żeby po operacji przywrócić dokładnie ten sam.
3. Tymczasowo przełączamy bazę na `SIMPLE`, żeby móc obciąć nadmiarową część logu, która w trybie `FULL` czekałaby na backup logu:

```
USE NazwaBazyDanych;
GO
ALTER DATABASE NazwaBazyDanych SET RECOVERY SIMPLE;
GO
```

1. Wykonujemy shrink do rozsądnego rozmiaru:

```
DBCC SHRINKFILE (NazwaBazyDanych_Log, 1);
GO
```

1. Przywracamy oryginalny tryb recovery:

```
ALTER DATABASE NazwaBazyDanych SET RECOVERY FULL;
GO
```

Uważaj: krok 3 przerywa ciągłość łańcucha backupów logu — od momentu przełączenia na `SIMPLE` do momentu wykonania kolejnego backupu pełnego lub różnicowego po powrocie do `FULL`, odtwarzanie do punktu w czasie nie jest możliwe. Ta procedura jest do wykonania w oknie serwisowym, nigdy w środku dnia roboczego na produkcji o wysokim RPO.

Odbudowa uszkodzonego pliku LDF

Uważaj: poniższa procedura jest krokiem ratunkowym stosowanym wyłącznie wtedy, gdy plik logu jest fizycznie uszkodzony i baza nie chce się otworzyć, a nie ma aktualnego backupu do odtworzenia. Odbudowa loga od zera zeruje informację o niezakończonych transakcjach — baza po takiej operacji bywa niespójna, dane niezatwierdzonych transakcji mogą zostać utracone bezpowrotnie.

1. Przełączamy bazę w tryb offline. Jeśli operacja trwa dłużej niż kilkanaście sekund, sprawdzamy zapytaniem z Sesje, zapytania i blokady, czy jakaś sesja blokuje przełączenie, i w razie potrzeby kończymy ją poleceniem `KILL`.

```
ALTER DATABASE NazwaBazy SET OFFLINE WITH ROLLBACK IMMEDIATE;
```

1. Odbudowujemy plik logu pod nową nazwą — dobrą praktyką jest zapisanie najpierw do nowego pliku, a nie nadpisywanie oryginału na miejscu:

```
ALTER DATABASE NazwaBazy
REBUILD LOG ON (NAME = NazwaLogicznaPlikuLog, FILENAME = 'SciezkaDoNowegoPliku.ldf');
```

1. Przełączamy bazę z powrotem online:

```
ALTER DATABASE NazwaBazy SET ONLINE;
```

1. Jeśli baza pokazuje się jako `(Restricted User)`, przełączamy ją na `MULTI_USER`, zgodnie z opisem na stronie Tryby pracy bazy: read-only, single user, offline.

1. Sprawdzamy tryb recovery — po odbudowie loga baza wraca zwykle w trybie `SIMPLE`, niezależnie od tego, w jakim trybie pracowała wcześniej:

```
ALTER DATABASE NazwaBazy SET RECOVERY FULL;
```

> W przeciwnym razie backup logów nie będzie się wykonywać, mimo że harmonogram zadania SQL Agent pozostaje aktywny.

1. Weryfikujemy stan loga:

```
SELECT name, log_reuse_wait_desc FROM sys.databases;
```

Jeśli po pierwszej odbudowie plik nadal sprawia problemy, powtarzamy krok 2 z nazwą oryginalną, dopiero po zmianie nazwy nowego pliku na docelową (dopisując do starego pliku sufiks `_OLD`) i po ponownym przejściu przez cykl online/offline.

Częste błędy

Regularny shrink logu jako element rutynowej konserwacji. Log skurczony do minimum, a potem od razu ponownie potrzebny do dużej operacji, rośnie z powrotem przez wielokrotne

autogrowth — każde powiększenie to zapis zer na nowej części pliku i pauza w zapisie, więc rutynowy shrink w dłuższej perspektywie szkodzi wydajności bardziej, niż gdyby log po prostu zostawić dużym.

Przełączenie na `SIMPLE` na stałe zamiast tymczasowo, żeby „raz na zawsze" pozbyć się problemu z logiem. Trwała zmiana trybu recovery z `FULL` na `SIMPLE` wyłącza możliwość odtworzenia do punktu w czasie i backupu logów — decyzja o trybie recovery należy do polityki RPO danej bazy, nie do doraźnego rozwiązania problemu z miejscem na dysku.

Odbudowa loga jako pierwsza reakcja na zapchany log. Rebuild loga jest ostatecznością przy uszkodzeniu pliku, nie sposobem na zwolnienie miejsca — do tego celu służy backup logu albo, w uzasadnionych przypadkach, shrink opisany wyżej.

Powiązane: Tryby pracy bazy: read-only, single user, offline · Backup baz danych · Log transakcyjny: WAL, redo, transaction log · Sesje, zapytania i blokady

Źródła: dokumentacja Microsoft Learn, sekcja „DBCC SHRINKFILE", „sys.databases (Transact-SQL)" i „ALTER DATABASE (Transact-SQL) SET Options"

Revision #44
Created 2025-03-08 10:56:52 UTC by Tomasz Konieczny
Updated 2026-10-26 14:07:46 UTC by Tomasz Konieczny