

Beschleunigte Datenbank Wiederherstellung (ADR) mit SQL Server 2019

Kategorie
sql server

Mit dem SQL Server 2019 wurde von Microsoft ein neues Feature mit dem Namen **ADR** (Accelerated Database Recovery) vorgestellt. Eigentlich handelt es sich dabei nicht um ein neues Feature, sondern viel mehr um die Überarbeitung der Wiederherstellungs-Mechanismen, wie man sie aus den Vorgängerversionen kennt.

Eine neue Komponente mit dem Namen **Persistenter Versionsspeicher (PVS)** wird benutzt, um Zeilen nach Änderungen über einen längeren Zeitraum noch verfügbar zu machen. In Verbindung mit sog. **logischer Wiederherstellung** kann die Transaktionsverarbeitung erheblich beschleunigt werden. Zusätzlich sorgt ein neuer **sLog** (eng. secondary **log** stream, also ein sekundärer Protokolldatenstrom) dafür, dass Wartezeiten im Prozess der Transaktionsverarbeitung verkürzt werden.

Aaron
Priesterrath

Mit Hilfe dieses neuen Aufbaus gelingt es Microsoft die Wiederherstellung erheblich zu verbessern. Sowohl für einzelne Datenbanken, als auch Pooldatenbanken – egal ob on-premise oder in der Cloud.

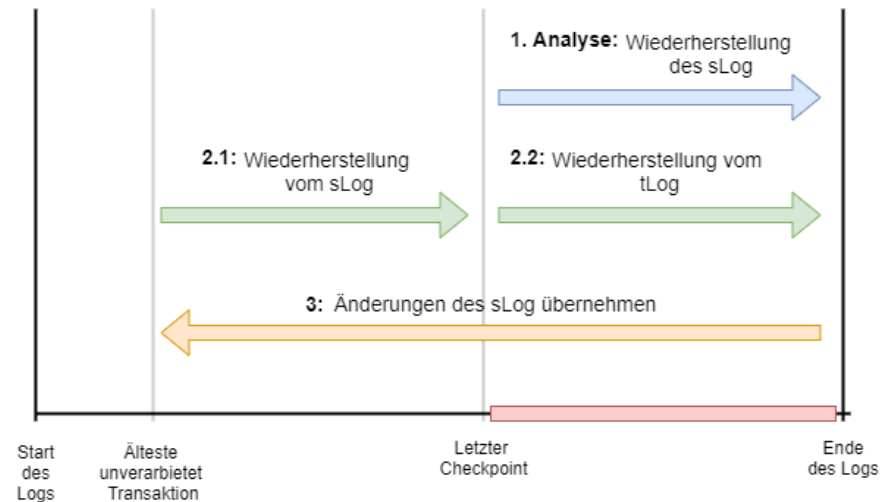
Nach eigener Aussage ist diese Überarbeitung vor allem sehr interessant für Kunden

- × die über eine Arbeitsbelastung mit zeitintensiven Transaktionen verfügen,
- × bei denen aktive Transaktionen in manchen Fällen dazu führen, dass die Transaktionsprotokolle sehr groß werden, oder
- × bei denen es vorkommt, dass die Datenbank über einen längeren Zeitraum nicht verfügbar war, da die Wiederherstellung zu lange gedauert hat (wie beispielsweise bei einem unerwarteten Neustart des Servers oder einem manuellen Transaktionsrollback).

Der Beitrag von Microsoft bezüglich ADR ist [hier](#) zu finden.

Wiederherstellungsprozess

Im folgenden möchten wir den Ablauf des neuen Mechanismus weiter durchleuchten:



Der Ablauf beginnt mit der **Analyse**. Diese Phase ist der ersten Phase im **alten** Mechanismus sehr ähnlich, mit dem kleinen Unterschied, dass zusätzlich der sLog wiederhergestellt wird.

In der zweiten Phase kann nun der zuvor wiederhergestellte sLog dafür genutzt werden, den Log-Bereich von der **ältesten unverarbeiteten Transaktion** bis zum **letzten Checkpoint** zu verarbeiten. Der tLog (Transaktions-Protokollstrom) wird nur für den Bereich vom letzten Checkpoint bis zum **Ende des Logs** benutzt.

In der letzten Phase kann nun mit Hilfe des **sLog** rückwirkend alle Änderungen verarbeitet werden. Dabei beschränkt sich der Bereich in dem der tatsächliche Transaktions-Log verarbeitet werden muss auf ein Minimum (im Diagramm der rot hinterlegte Balken).

ADR im Test

Die tatsächliche Auswirkung des ADR wird bereits in einem kleinen, trivialen Test deutlich, wie beispielsweise in [diesem Artikel](#) von [mssqltips.com](#) beschrieben.

In dem Artikel werden zum Test zwei Datenbanken erzeugt:

```
CREATE DATABASE AlteWiederherstellung;
GO
CREATE DATABASE NeueWiederherstellung;
ALTER DATABASE NeueWiederherstellung SET ACCELERATED_DATABASE_RECOVERY = ON;
GO
```

Als nächstes wird in zwei unterschiedlichen Abfragefenstern eine zeitaufwendige Abfrage ausgeführt. Diese Abfrage wartet auf das Erreichen einer bestimmten Uhrzeit, bevor sie ausgeführt wird. Diese muss angepasst werden.

```
WAITFOR TIME '13:45:00';
USE AlteWiederherstellung;
GO
DROP TABLE IF EXISTS dbo.fl1, dbo.fl2, dbo.fl3;
SELECT s2.* INTO dbo.fl1 FROM sys.all_columns AS s1 CROSS JOIN sys.all_objects AS s2;
SELECT s2.* INTO dbo.fl2 FROM sys.all_columns AS s1 CROSS JOIN sys.all_objects AS s2;
SELECT s2.* INTO dbo.fl3 FROM sys.all_columns AS s1 CROSS JOIN sys.all_objects AS s2;

WAITFOR TIME '13:45:00';
USE NeueWiederherstellung;
GO
DROP TABLE IF EXISTS dbo.fl1, dbo.fl2, dbo.fl3;
SELECT s2.* INTO dbo.fl1 FROM sys.all_columns AS s1 CROSS JOIN sys.all_objects AS s2;
SELECT s2.* INTO dbo.fl2 FROM sys.all_columns AS s1 CROSS JOIN sys.all_objects AS s2;
SELECT s2.* INTO dbo.fl3 FROM sys.all_columns AS s1 CROSS JOIN sys.all_objects AS s2;
```

In einem dritten Abfragefenster muss ein **SHUTDOWN** provoziert werden, um eine Unterbrechung der beiden zuvor ausgeführten Abfragen zu erzwingen und damit eine Wiederherstellung hervorzurufen.

```
WAITFOR DELAY '00:02:30';
SHUTDOWN WITH NOWAIT;
```

Nachdem alle Abfragen ausgeführt wurden, kann mit Hilfe des **Configuration Managers** der SQL Server erneut gestartet werden. Im **error-log** sind die Unterschiede bereits deutlich zu erkennen:





103	2020-02-07 13:43:50.270	spid5s	1 transactions rolled back in database 'NeueWiederherstellung' (8/0). This is an informational message only. No user action is required.
104	2020-02-07 13:43:50.280	spid5s	Recovery is writing a checkpoint in database 'NeueWiederherstellung' (8). This is an informational message only. No user action is required.
105	2020-02-07 13:43:51.100	spid5s	Recovery completed for database 'NeueWiederherstellung' (database ID 8) in 5 seconds (analysis 2537 ms, redo: 1146 ms, undo: 87 ms, system undo: 0 ms, regular undo: 0 ms). This is an informational message only. ...
106	2020-02-07 13:43:51.110	spid5s	[Dbid 8] ADR enabled for the database.
107	2020-02-07 13:43:51.140	spid5s	Parallel redo is shutdown for database 'NeueWiederherstellung' with worker pool size [2].
108	2020-02-07 13:43:55.050	spid29s	Recovery of database 'AlteWiederherstellung' (8) is 0% complete (approximately 2547 seconds remain). Phase 2 of 3. This is an informational message only. No user action is required.
109	2020-02-07 13:44:04.680	spid29s	Recovery of database 'AlteWiederherstellung' (8) is 11% complete (approximately 137 seconds remain). Phase 2 of 3. This is an informational message only. No user action is required.
110	2020-02-07 13:44:04.700	spid29s	4 transactions rolled forward in database 'AlteWiederherstellung' (8/0). This is an informational message only. No user action is required.
111	2020-02-07 13:44:04.710	spid29s	Recovery of database 'AlteWiederherstellung' (8) is 11% complete (approximately 137 seconds remain). Phase 3 of 3. This is an informational message only. No user action is required.
112	2020-02-07 13:44:08.480	spid5s	1 transactions rolled back in database 'AlteWiederherstellung' (8/0). This is an informational message only. No user action is required.
113	2020-02-07 13:44:08.490	spid5s	Recovery is writing a checkpoint in database 'AlteWiederherstellung' (8). This is an informational message only. No user action is required.
114	2020-02-07 13:44:08.840	spid5s	Recovery completed for database 'AlteWiederherstellung' (database ID 8) in 23 seconds (analysis 8507 ms, redo: 9623 ms, undo: 3776 ms, system undo: 0 ms, regular undo: 3672 ms). This is an informational message ...

Die Wiederherstellung der Datenbank **AlteWiederherstellung** hat insgesamt **23 Sekunden** gedauert. Davon wurden 8507 ms in der Analyse verbracht, 9623 ms im **redo** und abschließend 3776 ms im **undo**.

Im Vergleich dazu dauerte die Wiederherstellung der Datenbank **NeueWiederherstellung** gerade einmal 5 Sekunden, wovon 2537 ms in der Analyse, 1146 ms im **redo** und lediglich 87 ms im **undo** verbracht wurden.

Und auch in der Dateigröße lässt sich ein deutlicher Unterschied zwischen den beiden Datenbanken feststellen: Die Log-Datei der Datenbank **NeueWiederherstellung** benötigt nur ein Viertel des Speichers der Log-Datei der Datenbank **AlteWiederherstellung**. Dafür ist die tatsächliche Datenbank-Datei (.mdf) rund ein Viertel Größer.

Hier können Sie die Neuerungen deutlich erkennen:

 NeueWiederherstellung	07.02.2020 13:59	SQL Server Databa...	3.022.848 KB
 NeueWiederherstellung_log	07.02.2020 14:00	SQL Server Databa...	663.552 KB
 AlteWiederherstellung	07.02.2020 13:44	SQL Server Databa...	2.433.024 KB
 AlteWiederherstellung_log	07.02.2020 14:00	SQL Server Databa...	2.498.560 KB

Abschluss

Wie bereits oben beschrieben handelt es sich bei dem zuvor durchgeführten Testen um keine besonders aussagekräftige Umgebung um die Auswirkung von ADR in einem "realistischen" Szenario beurteilen zu können. Doch bereits jetzt wird deutlich, dass Microsoft eine enorme Verbesserung im Vergleich zu den alten Mechanismen gelungen ist, was in Zukunft das Arbeiten mit dem SQL Server konsistenter und angenehmer gestalten wird.