Accelerated Database Recovery

John Deardurff

Microsoft Customer Engineer (Global Technical Team)
Microsoft Certified Trainer (Regional Lead)
MVP: Data Platform (2016 – 2018)
Email: John.Deardurff@Microsoft.com
Twitter: @SQLMCT
Website: www.SQLMCT.com
GitHub: github.com\SQLMCT

Microsoft Most Valuable Professional
MVP
Reconnect
MICROSOFT CERTIFIED TRAINER
Microsoft
MCT
Trainer
2021 - 2022
MICROSOFT CERTIFIED PROFESSIONAL
Microsoft
MCSE
Data Management and Analytics
Earned 2016
Microsoft CERTIFIED
AZURE DATABASE ADMINISTRATOR
ASSOCIATE
Microsoft CERTIFIED
AZURE DATA ENGINEER
ASSOCIATE
Microsoft CERTIFIED
AZURE SOLUTIONS ARCHITECT
EXPERT

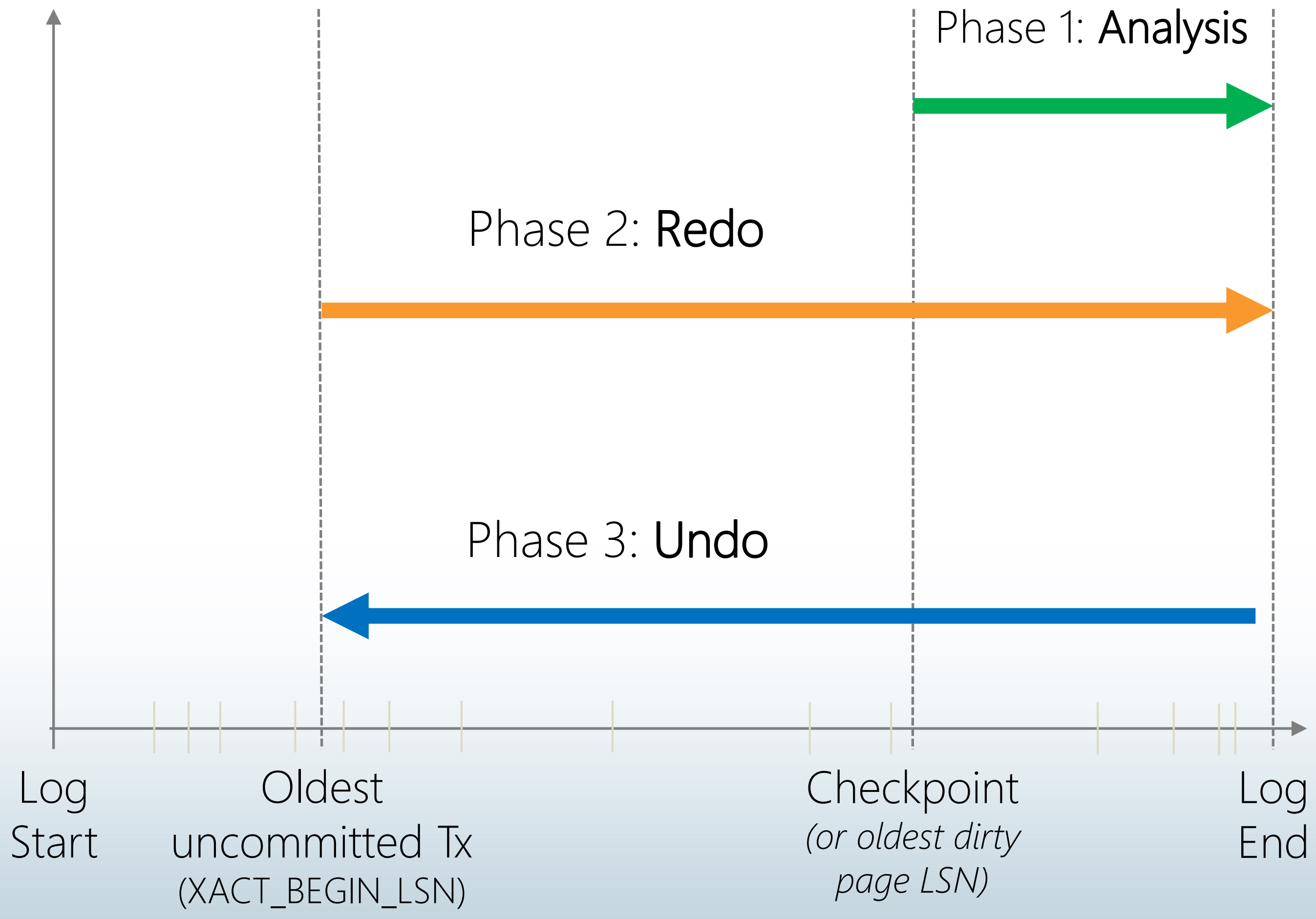
Accelerated Database Recovery

Accelerated Database Recovery is a new SQL Server Engine feature that greatly improves database availability by completely redesigning the current SQL Server recovery process.

Benefits of Accelerated Database Recovery

- Fast & Consistent Database Recovery
- Instantaneous Transaction Rollback
- Aggressive Log Truncation
- Available in Standard Edition

Current Database Recovery Process


Phase 1: **Analysis**
Phase 2: **Redo**
Phase 3: **Undo**
Log Start
Oldest uncommitted Tx (XACT_BEGIN_LSN)
Checkpoint *(or oldest dirty page LSN)*
Log End

Determines the state of each transaction in the system at the time SQL Server stopped.

Returns the database to the state it was in at the time the SQL Server stopped.

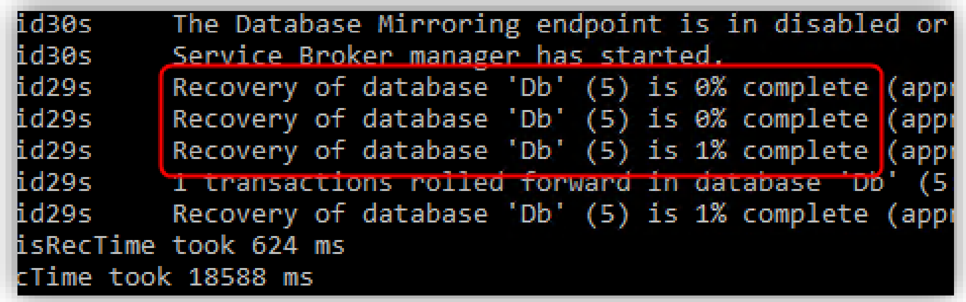
[DB is PARTIALLY available after Redo]

Rolls back any uncommitted transaction(s) individually.

[DB is FULLY available after Undo]

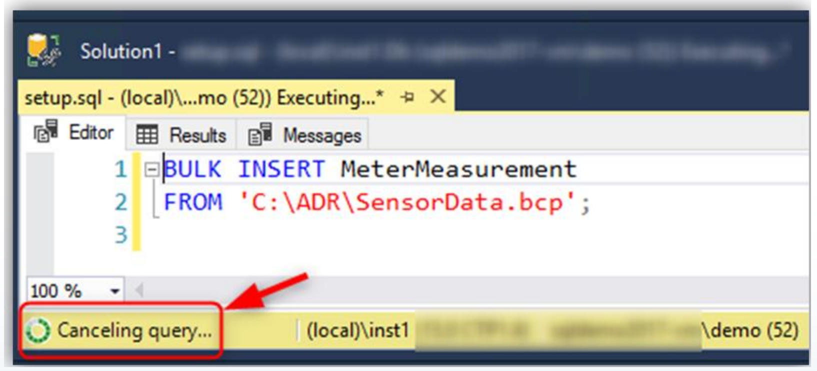
Most common implications

- Recovery time is roughly proportional to the longest running transaction.



```
id30s      The Database Mirroring endpoint is in disabled or
id30s      Service Broker manager has started.
id29s      Recovery of database 'Db' (5) is 0% complete (appr
id29s      Recovery of database 'Db' (5) is 0% complete (appr
id29s      Recovery of database 'Db' (5) is 1% complete (appr
id29s      1 transactions rolled forward in database 'Db' (5
id29s      Recovery of database 'Db' (5) is 1% complete (appr
isRecTime took 624 ms
cTime took 18588 ms
```

- Rolling back large batch operations (such as bulk insert) takes a long time.



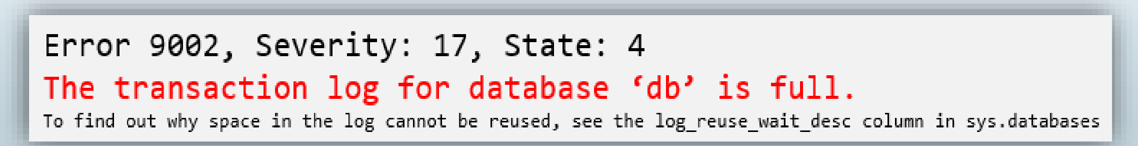
Solution1 - [redacted]

setup.sql - (local)\...mo (52)) Executing...*

```
BULK INSERT MeterMeasurement
FROM 'C:\ADR\SensorData.bcp';
```

Canceling query... (local)\inst1 [redacted] \demo (52)

- Transaction log may run out of space during long-running transactions.



```
Error 9002, Severity: 17, State: 4
The transaction log for database 'db' is full.
To find out why space in the log cannot be reused, see the log_reuse_wait_desc column in sys.databases
```

How to enable ADR?

Azure SQL Database

It's ON by default.

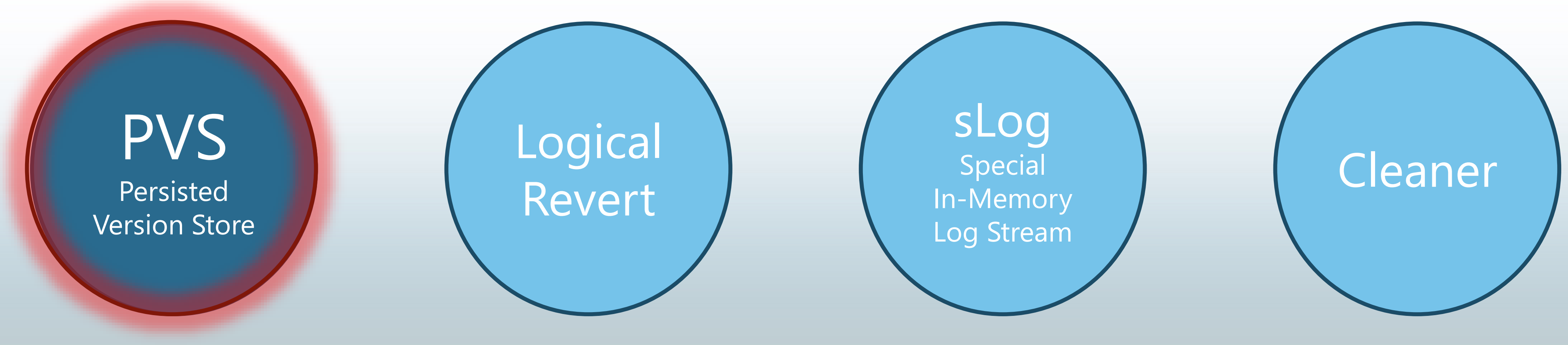
SQL Server 2019

```
ALTER DATABASE <db_name> SET
ACCELERATED_DATABASE_RECOVERY = ON
(PERSISTENT_VERSION_STORE_FILEGROUP = [VersionStoreFG])
```

Accelerated Database Recovery Components

Persisted Version Store

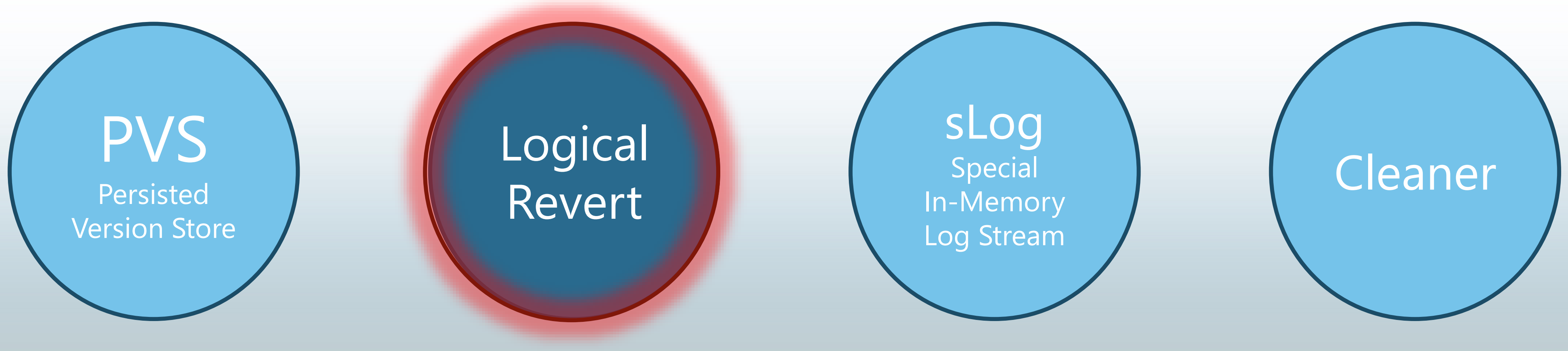
- Persists row versions in the database itself rather than TempDB.
- The version can be stored in-row or off-row within the database, it will vary according to the row size;
- Versions have the previous state of the data and the Transact-ID of the version;
- Fast UNDO, instead of rolling back the active transactions (traditional recovery process) the row version is marked as Terminated.



Accelerated Database Recovery Components

Logical Revert

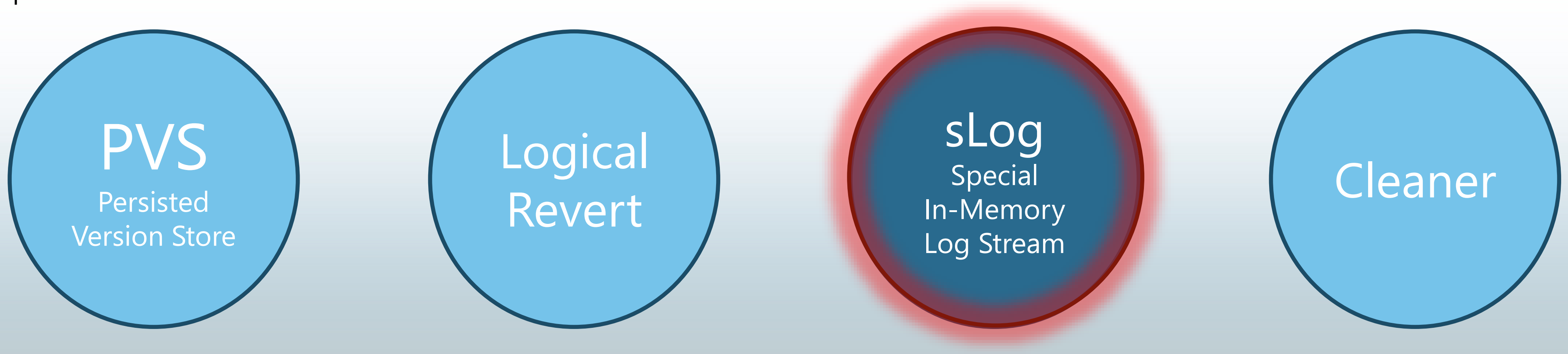
- Asynchronous process that performs row-level version-based Undo;
- Keeps track of all terminated transactions;
- Performs rollback using recent committed transactions from PVS;
- Release all locks immediately after transaction termination.



Accelerated Database Recovery Components

sLog

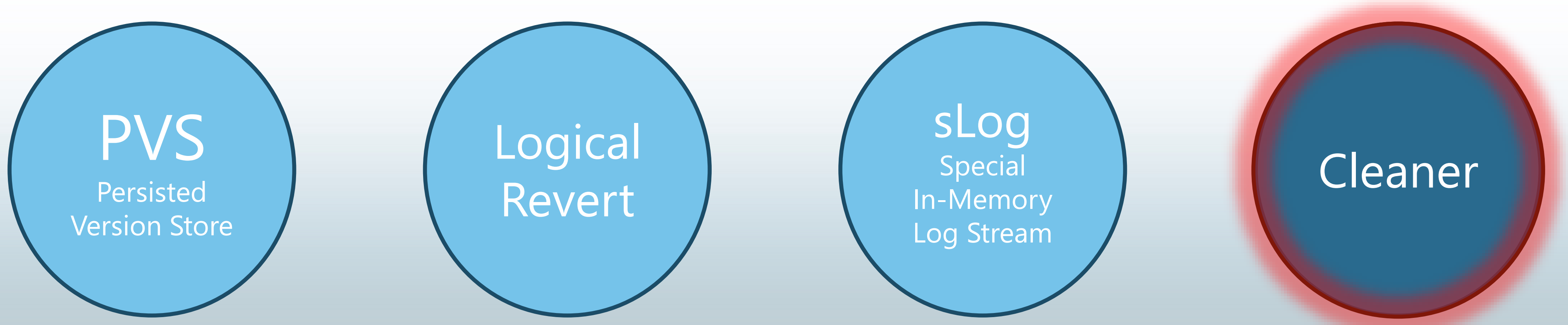
- Secondary in-memory log stream that stores log records for non-versioned operations (e.g.: metadata cache invalidation, lock acquisitions);
- Persisted on disk by been serialized during SQL checkpoint;
- Is periodically truncated as transactions commits;
- It accelerates the redo and undo by processing only the non-versioned operations;



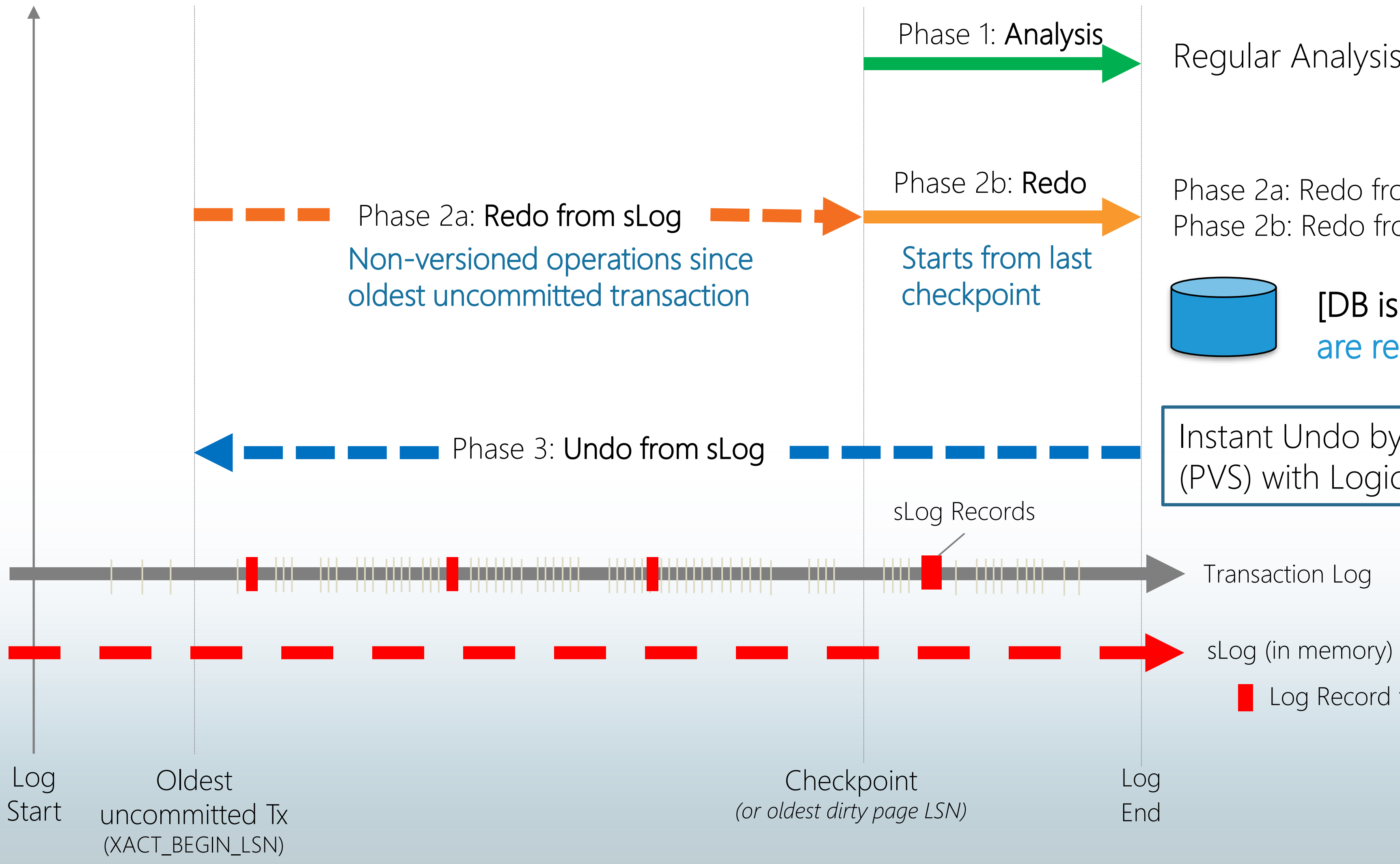
Accelerated Database Recovery Components

Cleaner

- Asynchronous process that periodically cleans row versions that are not needed.



Accelerated Database Recovery process


Phase 1: **Analysis**
Phase 2a: **Redo from sLog**
Non-versioned operations since oldest uncommitted transaction
Phase 2b: **Redo**
Starts from last checkpoint
Phase 3: **Undo from sLog**
sLog Records
Transaction Log
sLog (in memory)
Log Start
Oldest uncommitted Tx (XACT_BEGIN_LSN)
Checkpoint *(or oldest dirty page LSN)*
Log End

Regular Analysis + Reconstructs sLog

Phase 2a: Redo from **sLog**
Phase 2b: Redo from **Transaction Log**

[DB is FULLY available and all locks are released after Redo]

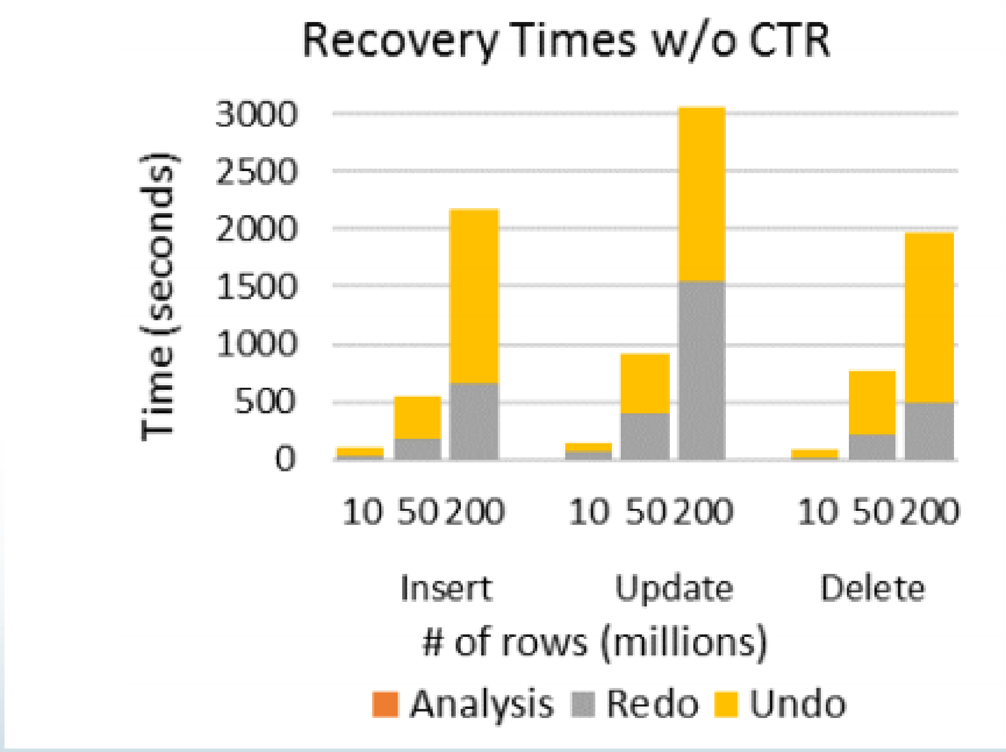
Instant Undo by using sLog and Persisted Version Store (PVS) with Logical Revert.

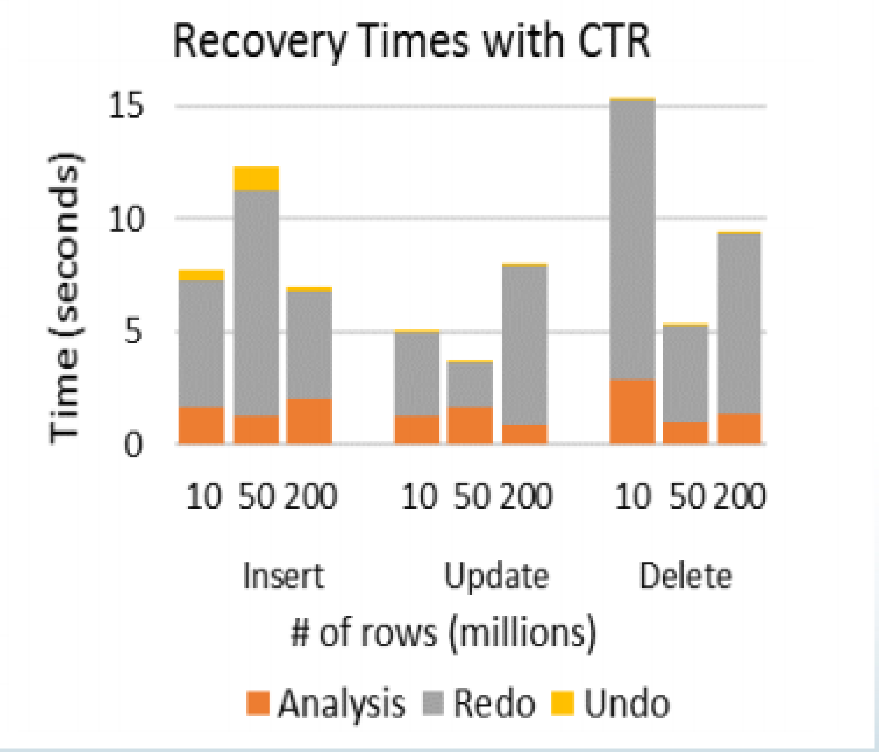
🟥 Log Record for non-versioned operation

Demo Time

Recovery Time Comparison

[Constant Time Recovery in SQL Server]





Accelerated Dabase Recovery FAQ

Will my database be larger?

- Yes. Monitor to determine difference.
- According to the CTR whitepaper, 50 million modifications add 1GB to database.

Will it affect performance?

- It depends. Write-heavy (OLTP) workloads are most susceptible.
- According to the CTR whitepaper, 13.8% utilization for Update heavy workloads, 2.4% for normal workloads.

How is PVS different than the version store in TempDB?

- PVS stores versions in the user database rather than TempDB
- If ADR is enabled, PVS is used to support SNAPSHOT and READ_COMMITTED_SNAPSHOT_ISOLATION transactions

How does this affect Availability Groups?

- PVS and log records replicate to secondaries, secondary communicates oldest versions needed to primary
- ADR can speed up failover because Undo becomes fast
- If the secondary must be restarted without ADR, TempDB is lost so versions are lost and queries must wait for data to commit on primary, with ADR, versions are persisted, so no delay before queries can be served

Questions?