Microsoft SQL Server Database Recovery Internals

RNDr. David Gešvindr, Ph.D.
MVP: Data Platform | MCSE: Data Platform | MCT
david@wug.cz
@gesvindr

Základní principy

- Zpracování transakcí na SQL Serveru je postaveno na přístupu **ARIES** (**Algorithms for Recovery and Isolation Exploiting Semantics**)
  - Autorem je Dr. C. Mohan (IBM)

- Jeho hlavní principy jsou:
  - **Write-ahead logging** – Jakákoliv změna je nejdříve zapsána do logu, před tím, než je zapsána na disk do datového souboru (garance atomičnosti a trvalosti změn)
  - **Repeating history during Redo** – Při restartu po havárii jsou identifikovány a zopakovány operace, jejichž změny se ztratily, aby byly zapsány potvrzené změny
  - **Logging changes during Undo** – Při rušení změn nepotvrzených transakcí tyto změny logujeme, aby při opakovaném restartu se akce neopakovaly

Zdroj: https://cs.stanford.edu/people/chrismre/cs345/rl/aries.pdf

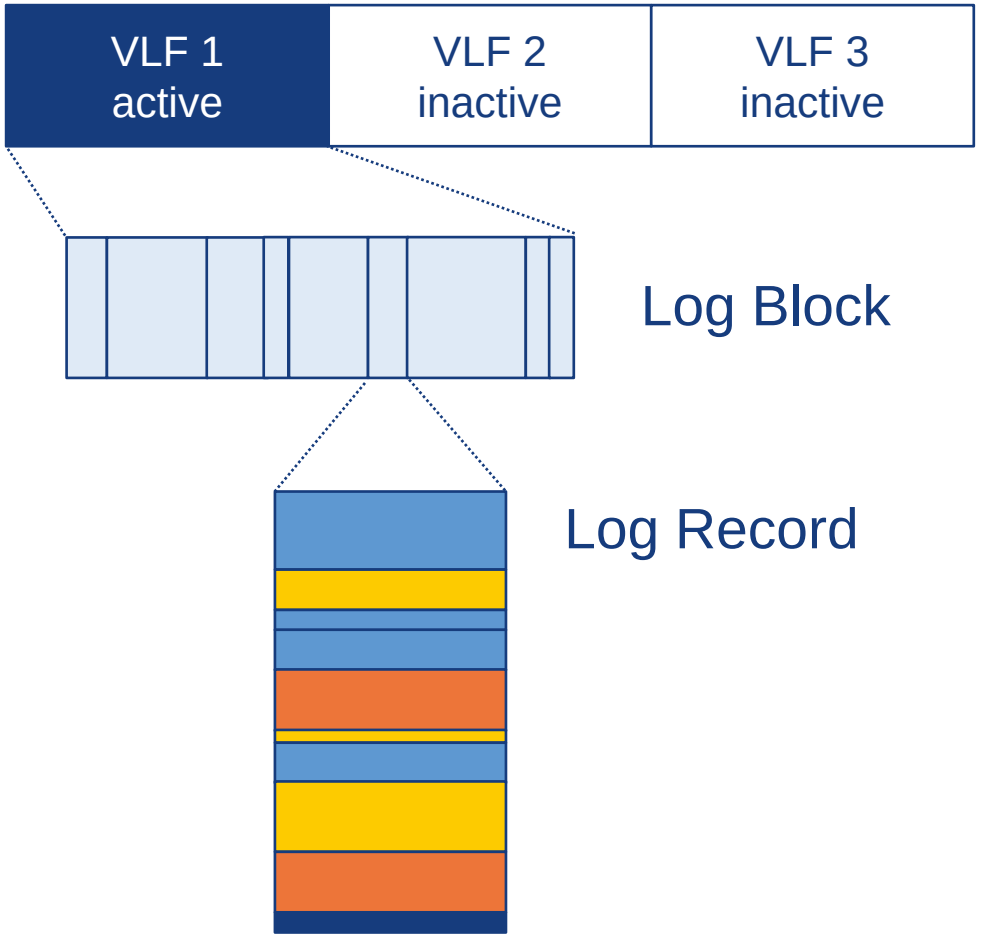
Transakční log

- Transakční log je pravděpodobně nejdůležitější součást SQL Serveru
- Bez transakčního logu bychom neměli v databázi:
  - Transakce – nebyl by mechanizmus, který by zajistil atomičnost a trvalost transakcí
  - Databáze by byla transakčně nekonzistentní, pravděpodobně poškozená při každé havárii
  - Nebylo by možné realizovat rollback transakce

- Fungování transakčního logu je současně jedna z nejméně známých částí SQL Serveru administrátorům a vývojářům
- „Transakční log není provozní log, nemažte ho vůbec nikdy“

Architektura transakčního logu

- Transakční log je interně rozdělen na **virtual log files (VLF)**
- Virtual log file je ve stavu **active** nebo **inactive/unused**
  - V databázi musí být vždy aspoň jeden aktivní VLF
- Virtual Log File je dále členěn na Log Block
  - Proměnná délka 512 B až 60 KB
  - Do Log Block jsou zapsány Log Records o různé délce pro různé transakce v pořadí podle toho, jak jsou generovány



Proces modifikace dat v databázi

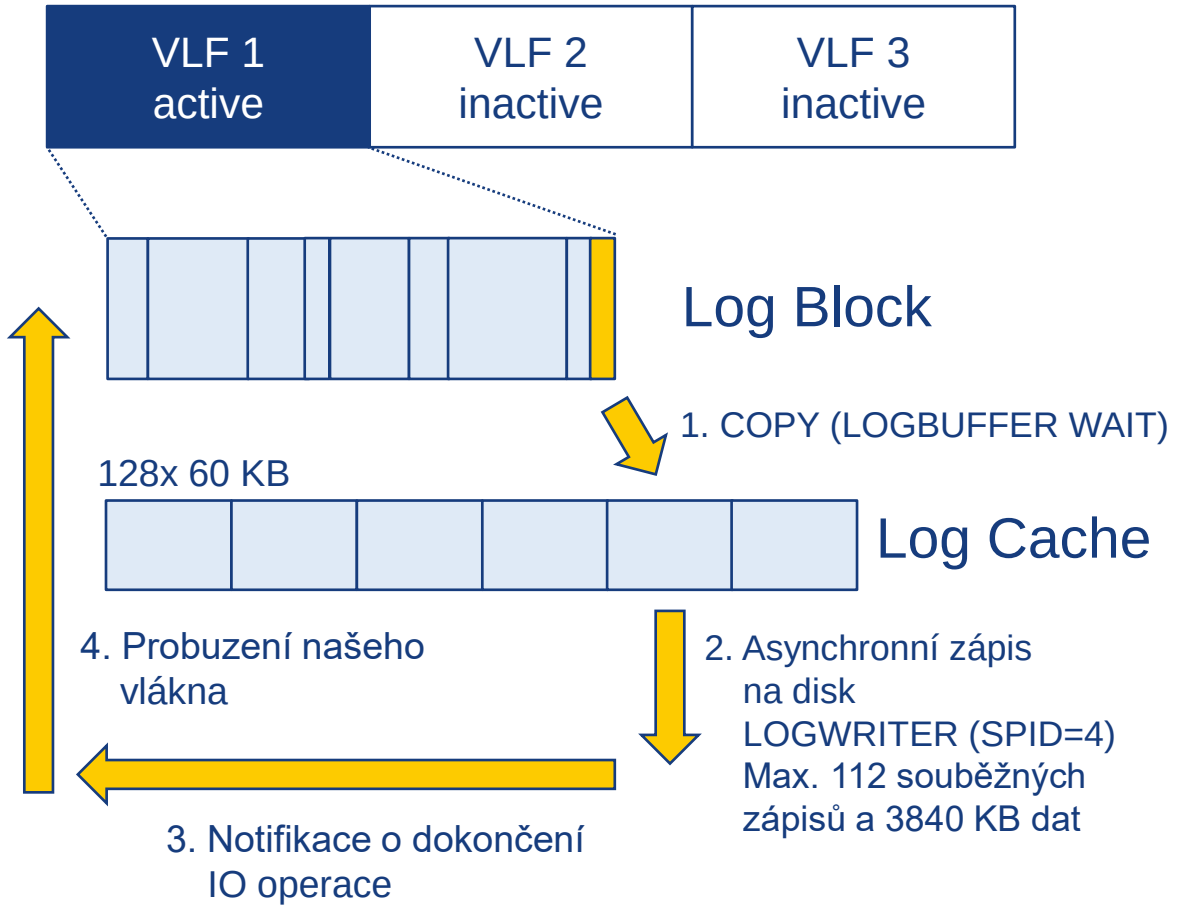
1. Uživatel odesílá UPDATE příkaz
2. Je otevřena transakce a tato informace je zapsána do transakčního logu
3. Jsou načteny ty datové stránky z disku do paměti, které budou modifikovány
4. Je provedena změna datové stránky v paměti a popis změny je zapsán do transakčního logu
5. Po dokončení všech změn v datových stránkách je do transakčního logu zapsán COMMIT transakce
6. Klientovi je odesláno potvrzení o dokončení transakce, což mu garantuje, že změny jsou trvale uloženy
7. Všechny změněné datové stránky jsou později propsány operací CHECKPOINT do datového souboru na disk (i nepotvrzené změny)

Transaction Commit

- Po dokončení všech změn transakce a zavolání TRANSACTION COMMIT dojde k následujícímu:
  1. Vygenerování záznamu LOP_COMMIT_TRAN a ukončení Log Blocku
  2. Zápis všech záznamů před LOP_COMMIT_TRAN (včetně) až na disk do transakčního logu (WRITELOG WAIT)
  3. Pokud máte Availabiltiy Groups: Čeká se na zápis Log Blocku na synchronní replice na její disky
  4. Uvolnění všech zámků, které si transakce drží
  5. Potvrzení transakce klientovi

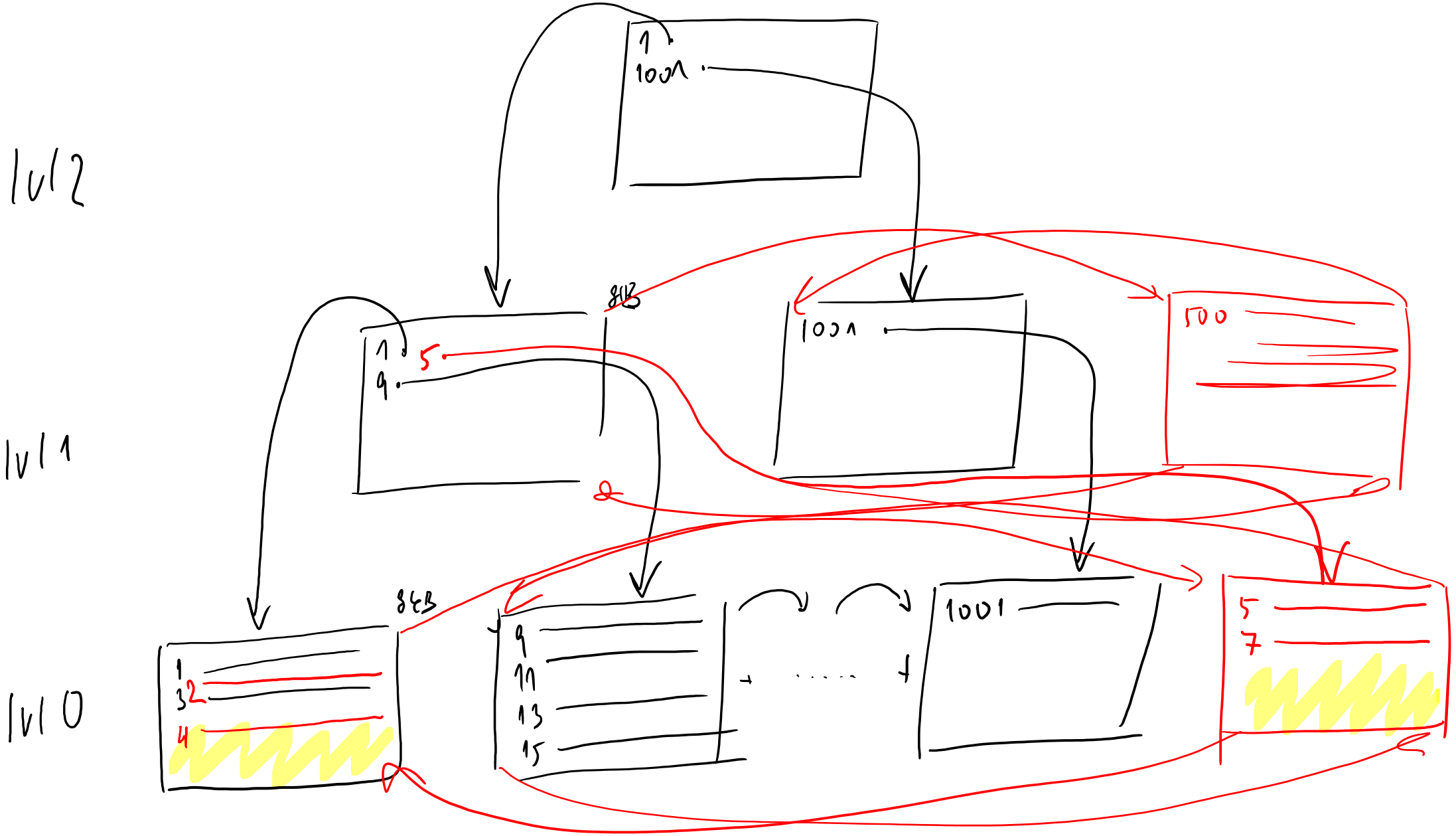
Zápis do transakčního logu

1. Log Block je okopírován do Log Cache
   - Může způsobit LOGBUFFER WAIT
   - Pevný počet 128x 60 KB log buffer
   - Naše vlákno přechází do stavu suspended na waitu WRITELOG
2. Proces LOGWRITER zahájí asynchronní zápis bloku do transakčního logu
   - SPID = 4 (SQL Server 2014+)
3. Další vlákno reaguje na dokončení IO operace a notifikuje naše vlákno
4. To se vrací do stavu RUNNING



WRITELOG Wait

- Čeká se na zápis Log Bufferu na disk
- Problém nemusí být automaticky na úrovni výkonu úložiště
- Nevytvářejte další soubory s transakčním logem
  - Zápis je sekvenční vždy do jednoho souboru
- Ověřte odezvu zápisu na disk (sys.dm_io_virtual_file_stats) a délku fronty požadavků pro zápis na disk
- Ověřte velikost transakcí (pozor na veliké množství velmi malých transakcí)
- Ověřte množství page splitů probíhajících na serveru
- Zkontrolujte existenci zbytečných indexů
  - Loguje se zápis do každého indexu


lvl 2
lvl 1
lvl 0
1
1001
8kB
1
9
5
1001
500
2
8kB
1
32
4
9
11
13
15
1001
5
7

CHECKPOINT

- CHECKPOINT je operace, která propíše všechny změněné datové stránky databáze z operační paměti zpět na disk
  - Zapisuje díky tomu jak potvrzené, tak nepotvrzené změny v datech na disk
- Do transakčního logu se loguje začátek i konec CHECKPOINTu
  - Loguje se: LSN začátku CHECKPOINTu, LSN nejstarší aktivní transakce (+replikace), seznam aktivních transakcí, LSN konce CHECKPOINTu
- Spouštění CHECKPOINTu probíhá automaticky podle množství aktivity na SQL Serveru

Čtení z transakčního logu

- Sekvenční čtení
  - Záloha transakčního logu
  - Ostatní zálohy
  - Transakční replikace, change data capture
  - CHECKPOINT v simple recovery modelu

- Náhodné čtení
  - Rollback transakce
  - Database Recovery
  - Vytváření databázového snapshotu, DBCC CHECKDB

Velikost transakčního logu

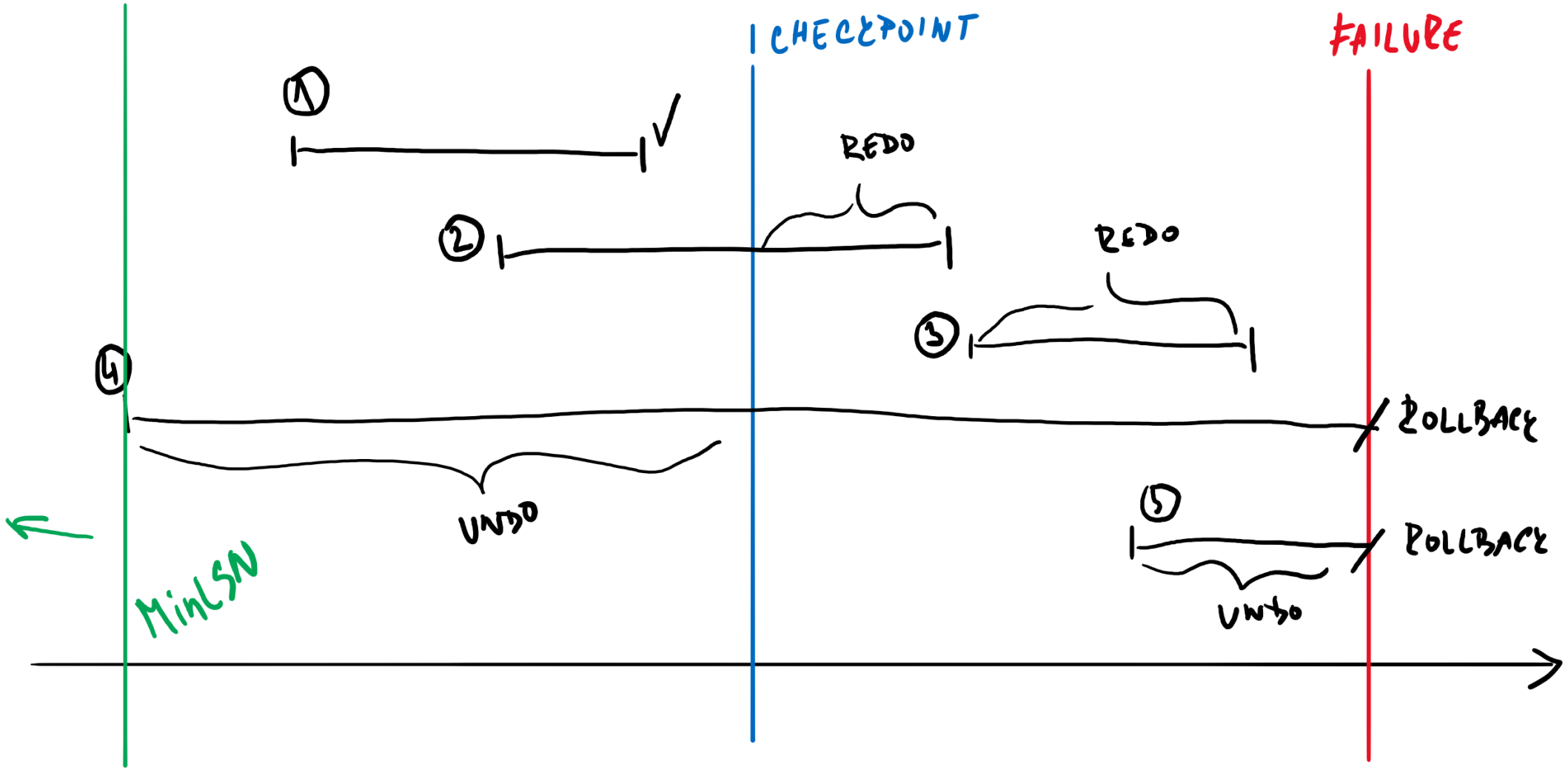
- Velikost transakčního logu je ovlivněna:
  - Recovery modelem
    - Ve Full Recovery Modelu závisí na frekvenci záloh transakčního logu
  - Velikostí transakcí
    - Dlouho trvající transakce nám zablokuje uvolňování transakčního logu do okamžiku jejího dokončení
    - Záleží jaké další transakce probíhají v jejím průběhu

Database Recovery

- Než je databáze zpřístupněna klientům (stav ONLINE), je nutné synchronizovat datový soubor a transakční log

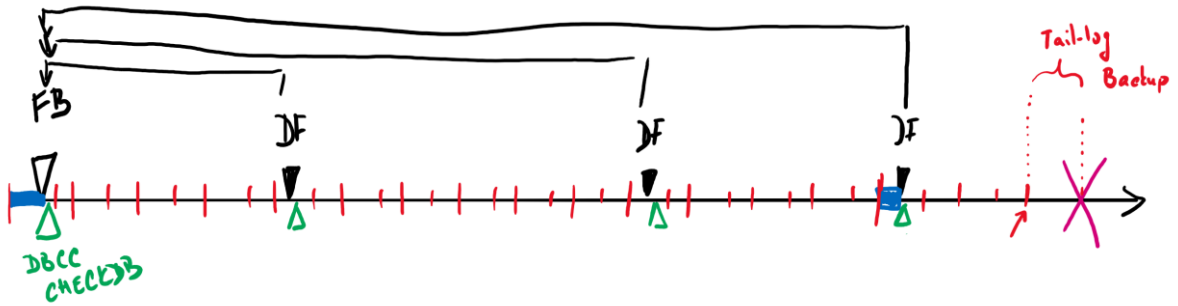
- V datovém souboru mohou chybět změny již potvrzených transakcí
  - Porušení vlastnosti trvalost databázové transakce
- V datovém souboru mohou být zapsány změny nepotvrzených transakcí
  - Operace CHECKPOINT propisuje všechny změněné datové stránky z paměti (pokud jejich poslední změna je již zapsána v logu na disku)
  - Narušení atomičnosti transakce a také integrity celé databáze

Database Recovery



Recovery Model

- Full Recovery Model
  - Logují se všechny prováděné operace
  - Záznamy mohou být z transakčního logu uvolněny po provedení zálohy logu
- Bulk-logged Recovery Model
  - Podpora pro minimally logged operace, některé dávkové operace mají zjednodušené logování
  - Záznamy mohou být z transakčního logu uvolněny po provedení zálohy logu
- Simple Recovery Model
  - Podpora pro minimally logged operace
  - Nepotřebné záznamy se uvolňují samy

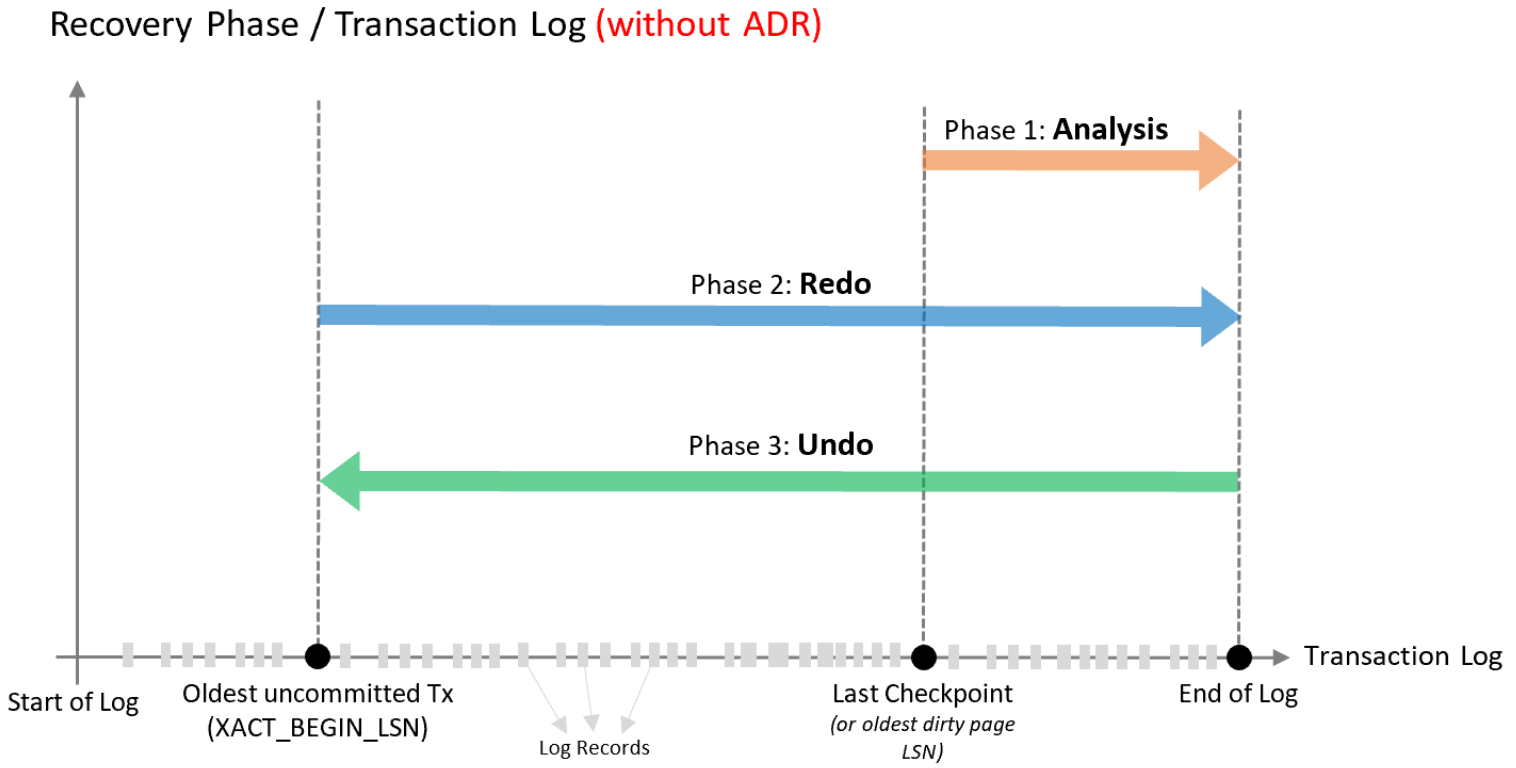


Výhody Full Recovery Modelu

- Full Recovery model a zálohy transakčního logu činí databázi odolnější proti ztrátě dat v následujících scénářích:
- **Poškození datového souboru**
  - Pokud je ztracen/poškozen datový soubor, ale nikoliv transakční log, je možné zazálohovat zatím nezazálohovanou část logu (Tail-log Backup) a obnovit tak databázi bez ztráty jediné potvrzené transakce
  - Je možné provést i Page Restore, kdy jsou obnoveny pouze poškozené stránky, které jsou zálohami logu uvedeny do konzistentního stavu se zbytkem databáze
- **Obnovení databáze do specifického časového okamžiku**
  - Záloha transakčního logu umožňuje provést obnovení stavu databáze k vybranému časovému okamžiku
  - Nelze obnovit jen část databáze (tabulku), vždy obnovujeme celou databázi

Accelerated Database Recovery (ADR)

- Recovery proces v SQL Serveru sestává ze 3 fází:
  1. Analytická fáze
  2. Redo fáze
  3. Undo fáze
- Délka recovery procesu je přímo úměrná velikosti nejstarší nepotvrzené transakce
  - Může trvat i hodiny!
  - Veliký soubor s logem



Zdroj: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-accelerated-database-recovery

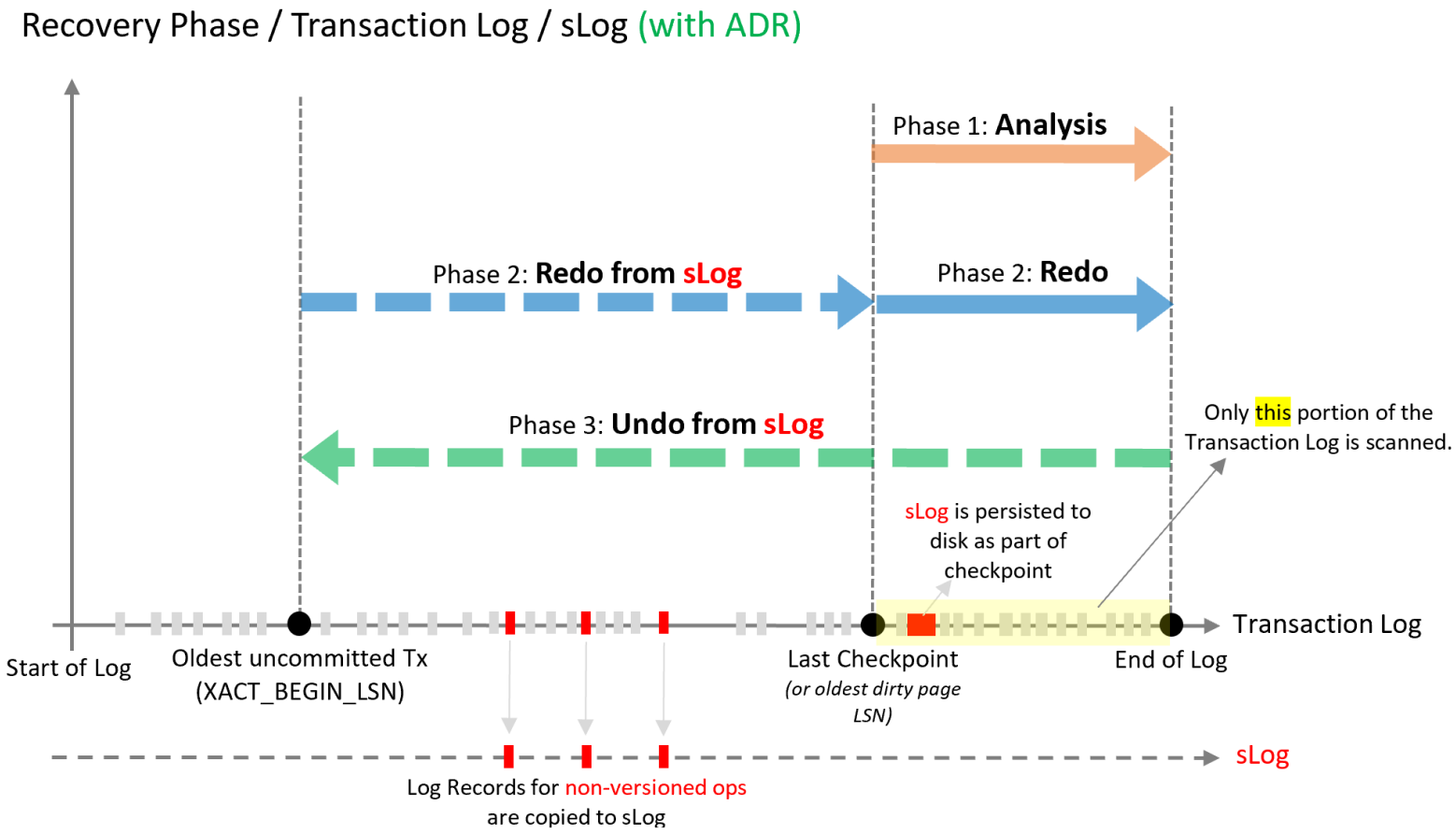
Accelerated Database Recovery (ADR)

- **Persistent Version Store (PVS)**
  - Nový mechanizmus, který ukládá verze měněných řádků jako součást databáze bez využití tempdb
- **Logical Revert**
  - Asynchronní proces, který provede velmi rychlé **undo** s pomocí návratu k předchozím verzím řádků
- **sLog**
  - Druhotný in-memory stream záznamů v transakčním logu, který loguje pouze neverzovatelné operace
  - Je velmi malý, je v paměti a při checkpointu se ukládá na disk
- **Cleaner**
  - Asynchronní proces, který odmazává nepotřebné verze datových stránek

Zdroj: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-accelerated-database-recovery

Accelerated Database Recovery (ADR)

- Výhody ADR:
  - Rychlá a konzistentní doba recovery
  - Instantní rollback transakcí
  - Agresivní uvolňování transakčního logu



Zdroj: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-accelerated-database-recovery

Dotazy

RNDr. David Gešvindr, Ph.D.
MVP: Data Platform | MCSE: Data Platform | MCT
david@wug.cz
@gesvindr