Zotavení z chyb

Databázové systémy

Zotavení z chyb v DBS

- Úloha:

  Po chybě obnovit poslední konzistentní stav databáze

- Třídy chyb:
  1. Lokální chyba v ještě nepotvrzené transakci
  2. Chyba se ztrátou hlavní paměti
  3. Chyba se ztrátou záložní paměti

Lokální chyba transakce

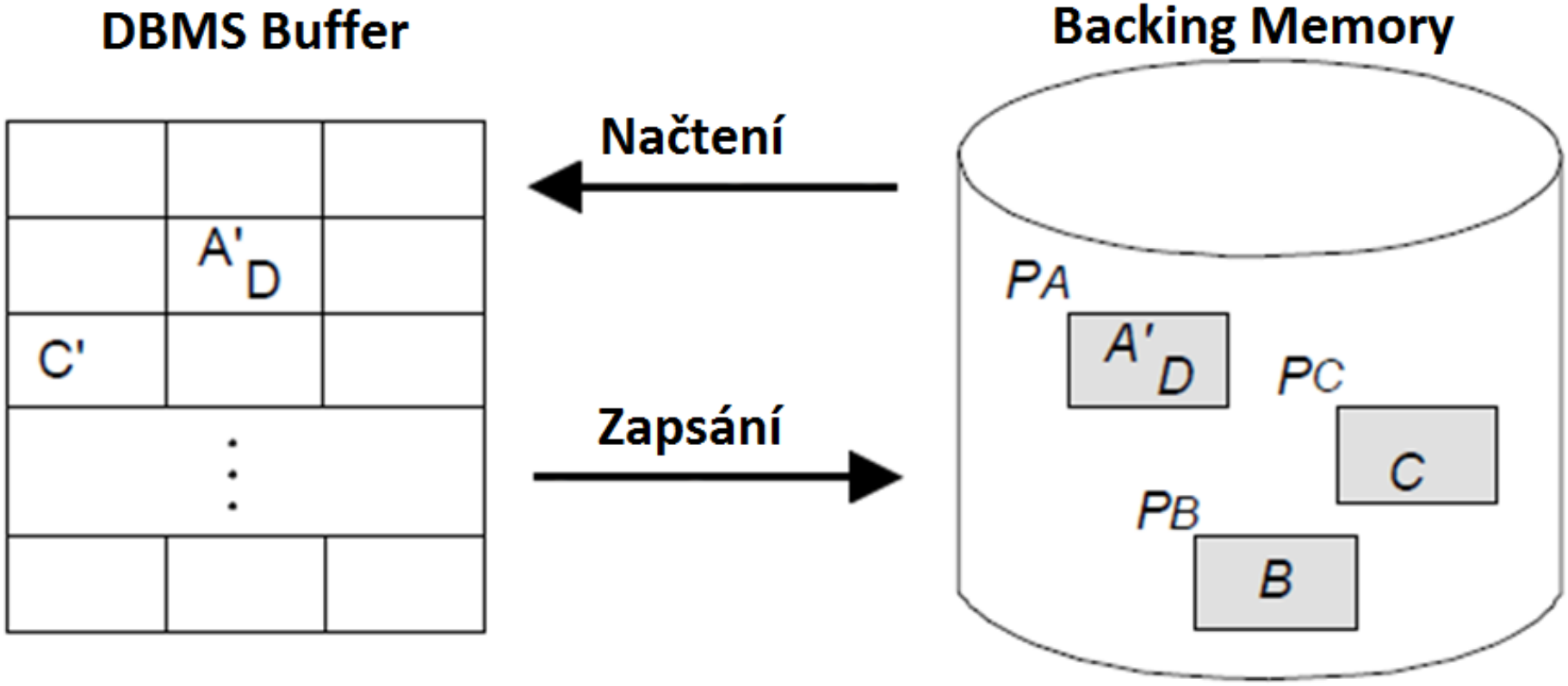
Typické chyby této třídy jsou:

- Chyba v aplikačním programu
- Explicitní přerušení (**abort**) uživatelem
- Systémové přerušení transakce
  - Např. aby se předešlo uvíznutí (deadlocku)

Oprava vrácením změn databáze způsobených touto dosud aktivní transakcí (**lokální undo**)

Chyba se ztrátou hlavní paměti

- Dvouúrovňová hierarchie úložišť - stránkování



Chyba se ztrátou hlavní paměti

- Je požadováno:
  - vrátit změny všech neukončených transakcí zapsané do „materializované“ databáze (**globální undo**)
  - všechny změny ukončených transakcí nezapsané do materializované databáze provést znovu (**globální redo**)

- Vyskytují se v intervalu řádu dní a lze je opravit v řádu minut za pomoci **protokolu** (logu)

Chyba se ztrátou záložní paměti

Nastává např. v těchto situacích:

- *head crash* – zničení disku s materializovanou DB
- Požár/zemětřesení – zničení disku
- Chyba systémového programu (např. ovladače disku)

Stávají se zřídka (měsíce, roky)

Obnova ze záložní kopie materializované DB + ze zálohy protokolu se změnami, ke kterým od té doby došlo

Hierarchie paměti: Výměna stránek bufferu

- Data používaná transakcí – na více stránkách
- Stránka v bufferu po dobu trvání přístupu, brání výměně -> označení „FIX“
- Změna dat -> označení „dirty“
- Ukončení operace -> odstranění „FIX“
- Strategie
  - ¬*steal:* nevymění stránky změněné dosud aktivními transakcemi
  - *steal:* stránku bez FIX lze nahradit
- ¬*steal:* změny nepotvrzených transakcí se nedostanou do materializované DB, při **Rollback** není třeba se starat o její stav
- *steal:* po **Rollback** je nutno již provedené změny v materializované DB vrátit zpět pomocí **Undo**

Hierarchie paměti: Zpětný zápis stránek z bufferu

- Strategie *force:* při **commit** transakce se zpět zapíší změněné stránky
- Strategie *¬force:*
  - zápis není nutný bezprostředně po potvrzení transakce, může proběhnout později
  - nutno zaznamenat více záznamů v protokolu

	force	***¬force***
¬steal	bez Redo, bez Undo	Redo, bez Undo
steal	bez Redo, Undo	Redo, Undo

  - *force + ¬steal:* lákavé, ale neekonomické při intenzívních změnách stránek také od ostatních aktivních transakcí

Hierarchie paměti: Strategie zpětného zápisu stránek

- *update-in-place*: stránka v bufferu odpovídá stránce na disku, do které se zapíše při změně
- *Twin-Block-Procedure:* stránka P v bufferu má v úložišti přiřazeny dvě stránky P^0 a P^1 - minulý a předminulý stav této stránky
  - Při zápisu se přepisuje předposlední stav
  - I při chybě během zpětného zápisu je stále k dispozici poslední stav

Protokolování změnových operací

Uvažujme nadále tuto konfiguraci systému:

- *steal*
- *¬force*
- *update-in-place*
- malá granularita zamykání: stránka v bufferu může zároveň obsahovat změny potvrzených transakcí i změny neukončených transakcí

„Resetovatelné“ historie

- *Scheduler* dohlíží na serializovatelnost transakcí
- Resetovatelné historie – založeny na závislostech operací čtení a zápisu
- V historii H transakce T_i čte od transakce T_j, pokud:
  - T_j zapíše data A, která následně přečte T_i
  - T_j není vrácena zpět před přístupem ke čtení od T_i
  - všechny ostatní přístupy k zápisu do A ostatními transakcemi jsou vráceny zpět před čtením od T_i
- Historie je ***resetovatelná***, pokud vždy zapisující T_j před čtením v T_i provede svůj **commit**
- Neboli, transakce smí provést svůj **commit** až po ukončení všech transakcí, od nichž sama četla
- Pokud by to neplatilo, nebylo by možné vrátit zpět zapisující transakci, protože transakce, která čte, by pak byla potvrzena s oficiálně neexistující hodnotou A.

Struktura záznamů protokolu

Záznam změnové operace:

- *Redo* informace
- *Undo* informace
- *LSN (Log Sequence Number) – jedinečné ID*
- *Transaction ID* volající transakce
- *PageID* – kde byla změna
- *PrevLSN* – odkaz na předchozí záznam odpovídající transakce (jen kvůli efektivitě)

Příklad: Dvě transakce a protokol

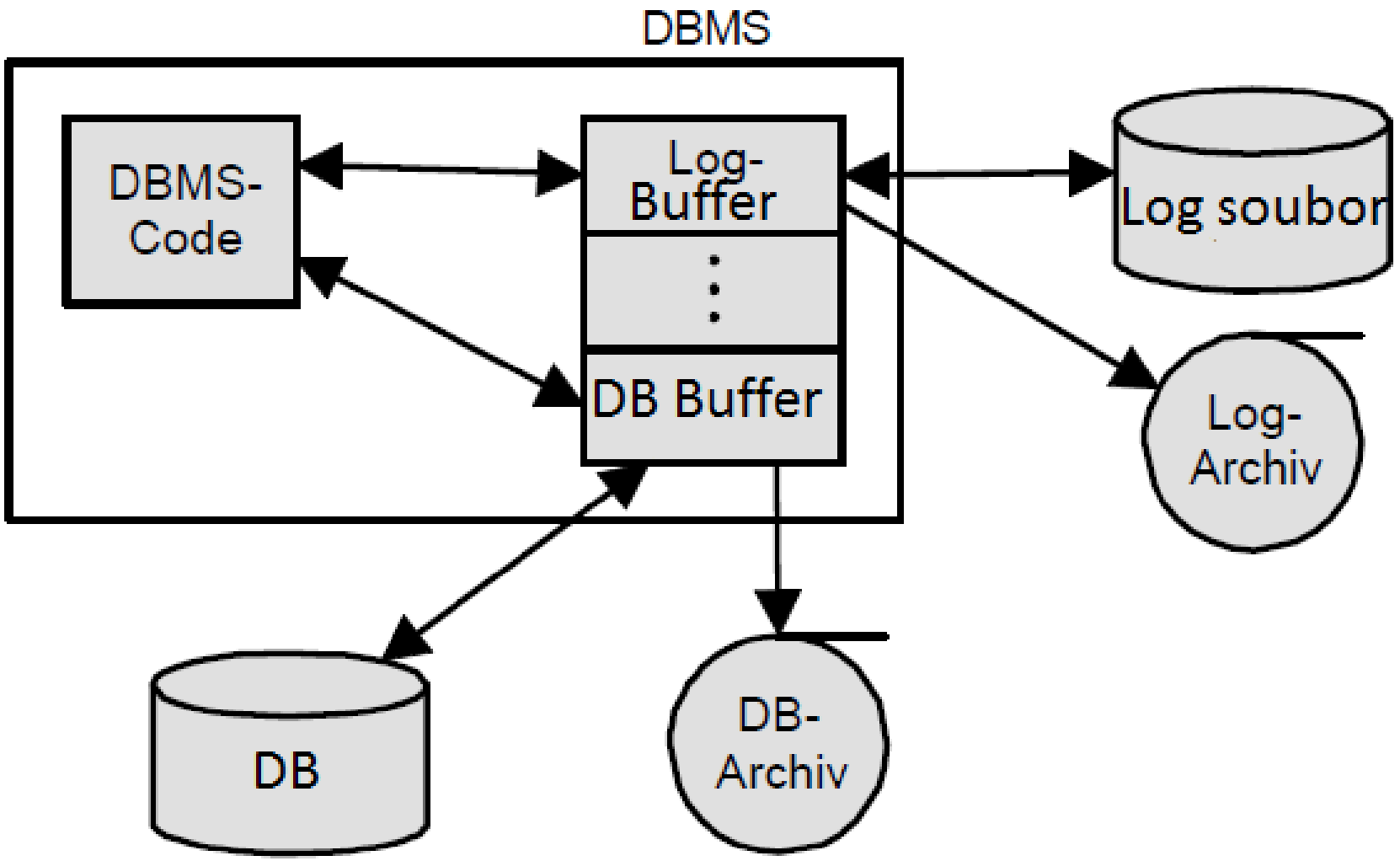
	T_1	T_2	Log
			[LSN, TA, PageID, Redo, Undo, PrevLSN]
1.	**BOT**		[#1, T_1, **BOT**, 0]
2.	$r(A, a_1)$		
3.		**BOT**	[#2, T_2, **BOT**, 0]
4.		$r(C, c_2)$	
5.	$a_1 := a_1 - 50$		
6.	$w(A, a_1)$		[#3, T_1, P_A, A-=50, A+=50, #1]
7.		$c_2 := c_2 + 100$	
8.		$w(C, c_2)$	[#4, T_2, P_C, C+=100, C-=100, #2]
9.	$r(B, b_1)$		
10.	$b_1 := b_1 + 50$		
11.	$w(B, b_1)$		[#5, T_1, P_B, B+=50, B-=50, #3]
12.	**commit**		[#6, T_1, **commit**, #5]
13.		$r(A, a_2)$	
14.		$a_2 := a_2 - 100$	
15.		$w(A, a_2)$	[#7, T_2, P_A, A-=100, A+=100, #4]
16.		**commit**	[#8, T_2,**commit**, #7]

Logické vs. fyzické protokolování

- Fyzické protokolování – místo Undo a Redo se ukládají *Before-Image* a *After-Image*
- Logické protokolování – zadání *Undo* a *Redo* operací (jako předchozí příklad)
  - *Before-Image* se generuje provedením *Undo* operace z *After-Image*
  - *After-Image* se generuje provedením *Redo* operace z *Before-Image*
- Z *LSN* lze poznat, zda je *Before-Image* nebo *After-Image* obsažen v materializované DB
  - Nová *LSN* se ukládají v rezervované části stránky, později jsou zapsána do DB spolu se stránkou

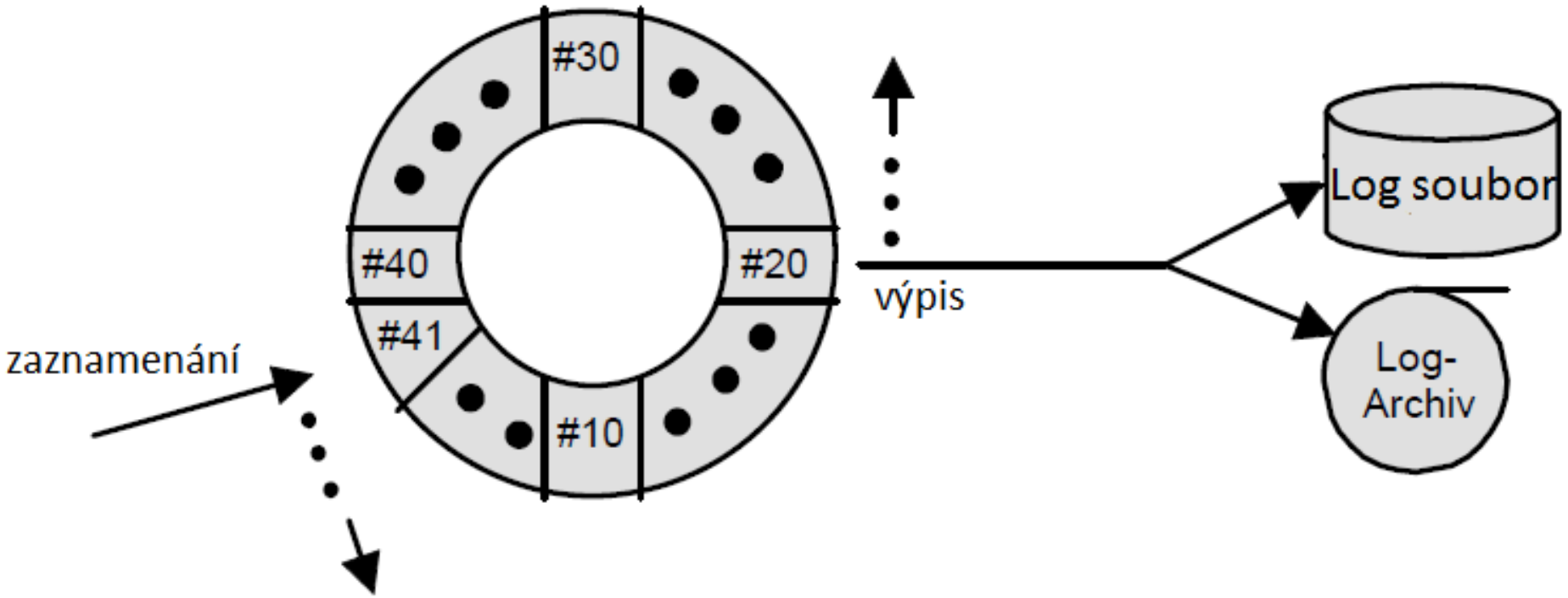
  Pro určitý záznam protokolu:
  - Je-li *LSN* stránky menší než v protokolu, jde o *Before-Image*
  - Je-li *LSN* stránky větší nebo rovno *LSN* v protokolu, *After-Image* již byl materializován

Hierarchie úložišť pro zabezpečení dat



Zápis informace do protokolu

- Před provedením změny musí být vytvořen záznam protokolu
- Záznamy protokolu se ukládají v log-bufferu v paměti
- Moderní DBS: log-buffer implementován jako kruhový buffer
- Záznamy protokolu se zapisují zároveň do dočasného souboru na disku a do archivu protokolu např. na magnetickou pásku


#30
#40
#20
#41
#10
zaznamenání
výpis
Log soubor
Log-
Archiv

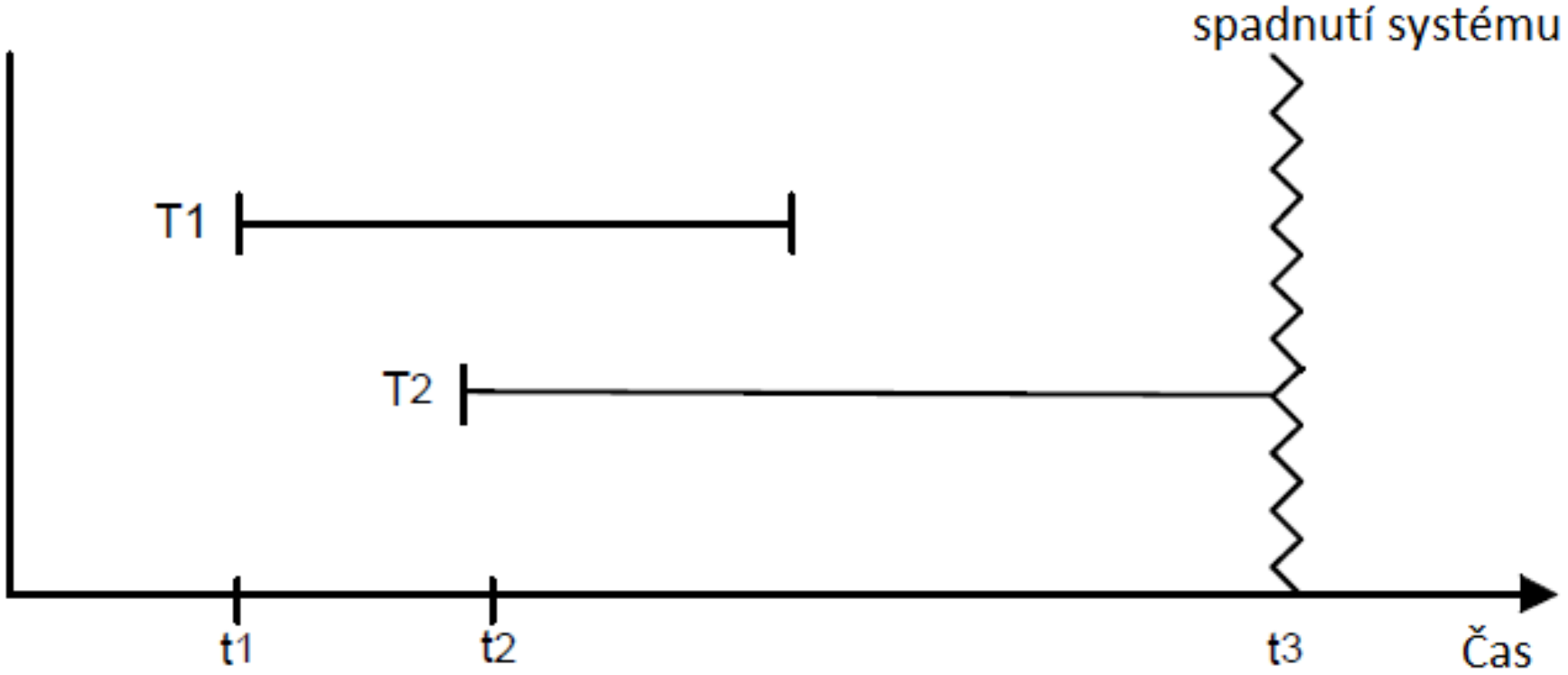
Princip WAL

Při zápisu do protokolu platí princip WAL (Write Ahead Log):

- Předtím než je transakce pevně zapsána (commit), musí být zapsány všechny příslušné záznamy protokolu. To je vyžadováno, aby bylo možné provést *Redo* po chybě.
- Předtím než je modifikovaná stránka v bufferu nahrazena, musí být všechny k ní příslušející záznamy protokolu zapsány do log-souboru.

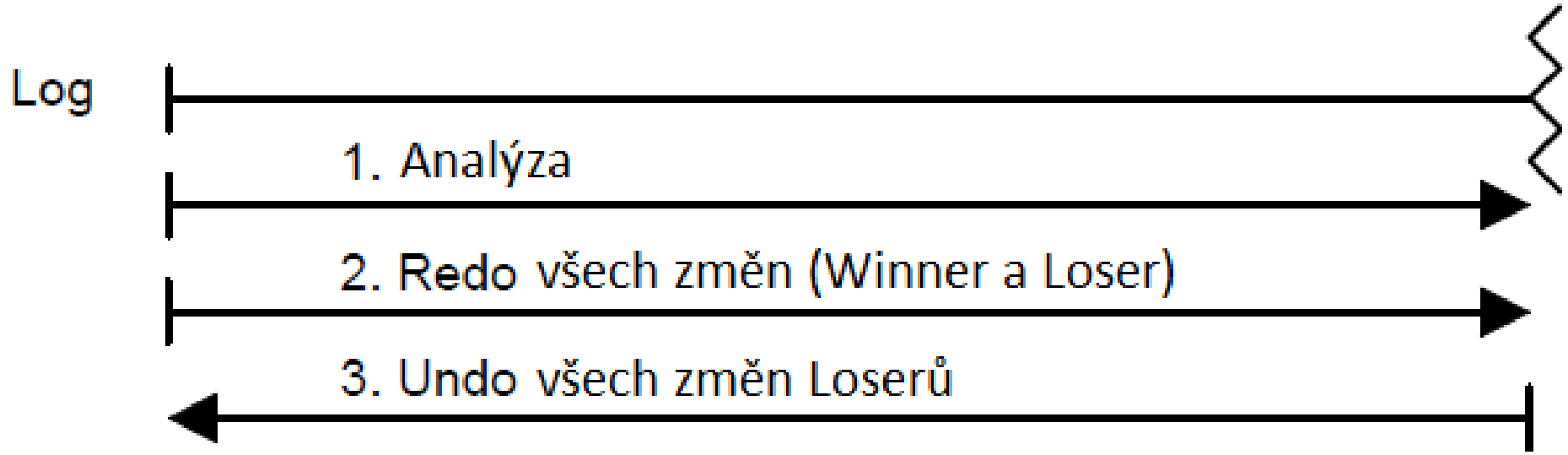
Zotavení po chybě

- Dva typy transakcí při „spadnutí“ systému



- T1 … Winner -> *Redo* T2 … *Loser* -> *Undo*

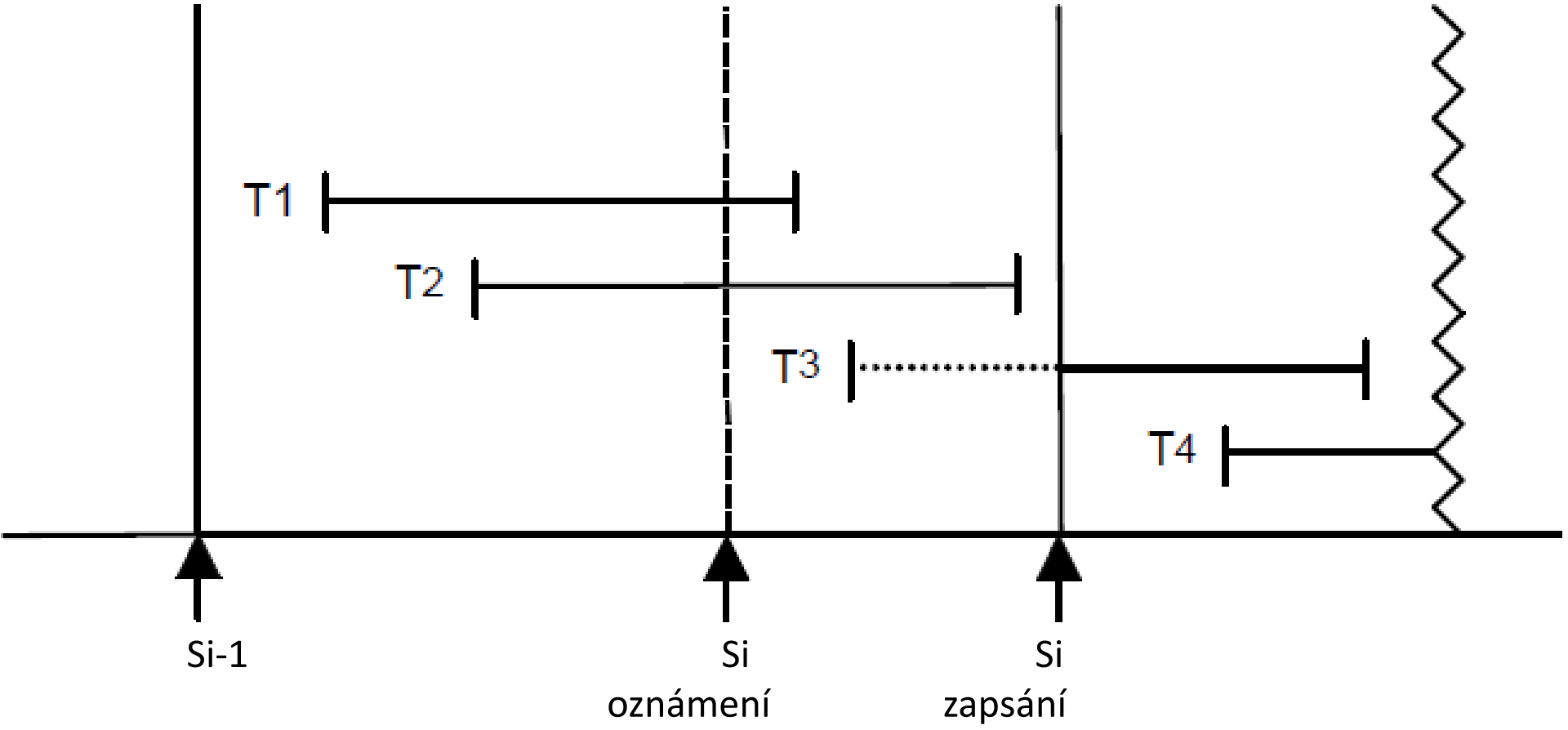
Zotavení – 3 fáze



- 1: *Winner* prokáže v logu úspěšný commit, u *Loserů* není

Body obnovy

- transakčně konzistentní stavy zpoždění T3



Ztráta materializované databáze

- Dva druhy zotavení po chybě
  - Materializovaná DB + dočasný log soubor
  - DB Archív + log archív

    => Konzistentní DB

Zabezpečení dat u SQL Serveru 2000

```
EXEC sp_addumpdevice
     'disk',
     'unidump',
     'c:\dump\unidump.dat'

backup database uni to unidump
```

```
restore database uni
from unidump
```

```
EXEC sp_addumpdevice
     'disk',
     'unilog',
     'c:\dump\unilog.dat'

backup log uni to unilog
```

```
restore log uni
from unilog
```