

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

REPLIKACE DAT

Replikace

2

- ❑ Replikace je kopírování dat z databáze do více (geograficky vzdálených) míst za účelem podpory distribuovaných aplikací.
- ❑ Replikuje se
 - ❑ celá tabulka,
 - ❑ fragment tabulky, který vznikne jako výsledek dotazu.
- ❑ V 90. letech se objevily u velkých výrobců databázových systémů replikační servery, které se staly zatím nejefektivnějším přístupem k implementaci distribuovaných databází.

Vlastnosti replikace

3

- ❑ DDBS založený na replikaci musí:
 - ❑ dopravit replikovaná data (kdekoli a kdykoli),
 - ❑ automaticky synchronizovat kopie po chybě,
 - ❑ co nejrychleji propagovat změny,
 - ❑ chránit transakční a logickou integritu dat,
 - ❑ replikovat data z (do) heterogenních serverů,
 - ❑ podporovat návrh aplikace, který je v souladu s pravidly podniku.

Výhody replikace

4

- ❑ Replikace může zlepšit výkon a zvýšit dostupnost aplikací, protože lze použít alternativní kopii dat.
- ❑ Uživatelé mohou použít lokální databázi namísto vzdáleného serveru a minimalizovat tak síťový provoz.
- ❑ V případě úplného výpadku některých částí distribuované databáze mohou být kopie stále přístupné.
- ❑ Poskytují uživatelům lokální aktualizované kopie dat.

Problémy replikace

5

- ❑ Jak se rozhodneme kolik kopií udržovat?
- ❑ Čím více kopií vytvoříme, tím je náročnější úprava dat, ale tím efektivnější jsou dotazy.
- ❑ Jak udržíme kopie v identickém stavu?
- ❑ Co se stane, je-li porucha na síti a různé kopie v síti mají možnost samostatného vývoje? Jak data po odstranění poruchy sjednotíme?
- ❑ V důsledku distribuce dat se může stát, že transakce zahrnují více míst. Co se stane, když jedna komponenta transakce bude chtít transakci zrušit a jiná bude chtít transakci řádně ukončit?

Příklad

6

Předpokládejme, že máme tabulku R , ke které přistupujeme z n míst. N -té místo vyšle za sekundu q_i dotazů na R a u_i požadavků na změnu dat v tabulce R .

❑ Cena dotazu

= **d** , pokud je dotaz vyslán ze stejného místa v síti, ve kterém je umístěna tabulka R .

= **$10d$** , pokud je dotaz vyslán z jiného místa v síti než je umístěna tabulka R .

❑ Cena změny

= **z** , pokud je změna požadována ze stejného místa

= **$10z$** , pokud je změna požadována z jiného místa v síti než je umístěna tabulka R .

Příklad - výsledek

7

- ❑ Jak na základě daných parametrů rozhodnete, na která místa v síti se má tabulka R replikovat?

Optimální místa pro umístění replik jsou ta, pro které platí vztah:

$$9dq_i + 9zu_i \geq Z$$

Z je cena všech změn dat požadovaných z každého místa v síti na data umístěná na jiném místě: $Z = \sum 10zu_i$

Preferujeme místa, ve kterých probíhá hodně aktivit, ale čím větší je frekvence požadovaných změn, tím méně replik vytvoříme.

Příklad – ilustrace – pouze 2 místa

8

Místo 1

Dotazy lokální:

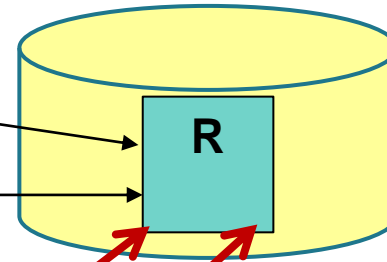
- Cena dotazu d
- Frekvence dotazů q_1

Změny lokální

- Cena změny z
- Frekvence změn u_1

$$\text{Celková cena} = d * q_1$$

$$\text{Celková cena} = z * u_1$$



Pokud vytvoříme repliku R, tak obě operace budou probíhat lokálně a úspora je $9dq_2 + 9zu_2$

Místo 2 neobsahuje R

Dotazy na jiné místo v síti:

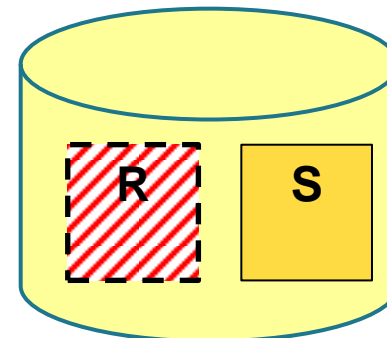
- Cena dotazu $10d$
- Frekvence dotazů q_2

Změny na jiném místě v síti

- Cena změny $10z$
- Frekvence změn u_2

$$\text{Celková cena} = 10 * d * q_2$$

$$\text{Celková cena} = 10 * z * u_2$$



Příklad – řešení (pokračování)

9

- ❑ Tím, že jsme vytvořili repliku tabulky R v místě 2, tak zlevnily dotazy i aktualizace dat v tabulce R; musíme ale udržovat repliku synchronizovanou.
- ❑ Synchronizace vyžaduje propagovat všechny změny v tabulce R realizované v místě 1 na místo 2 a obráceně.
- ❑ Náklady Z na propagaci změn jsou dané vzorcem:
$$Z = 10zu_1 + 10zu_2$$
- ❑ Úspora ze zavedení replik by měla převýšit náklady, potřebné na synchronizaci replik, proto repliku tabulky R by bylo vhodné vytvořit v případě, že platí
$$9dq_2 + 9zu_2 \geq Z$$

Formy replikace

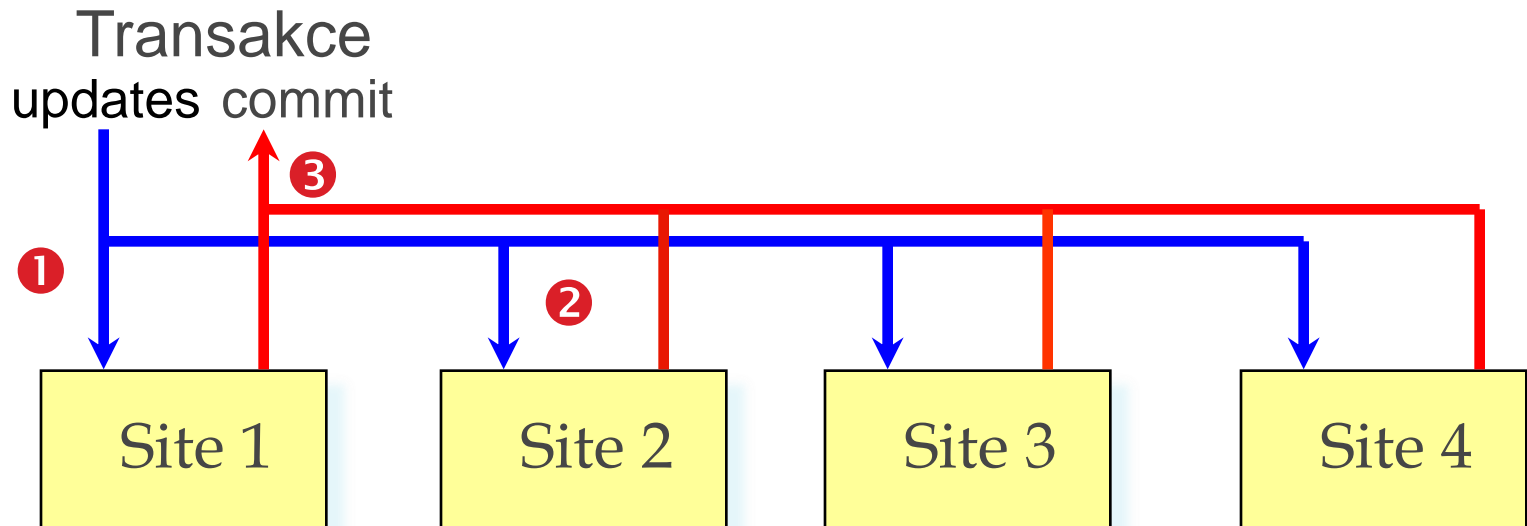
10

- ❑ Podle místa kde začíná aktualizace
 - ❑ centralizované – aktualizace může začínat pouze v jednom místě (master-slave),
 - ❑ distribuované (aktualizace může začít kdekoliv).
- ❑ Podle šíření aktualizace (propagace změn)
 - ❑ synchronní probíhá současně s vkládáním dat – replikace je vlastně součástí transakce
 - ❑ asynchronní probíhá pokud nastane nějaká událost (metoda pull nebo push).

Synchronní replikace

11

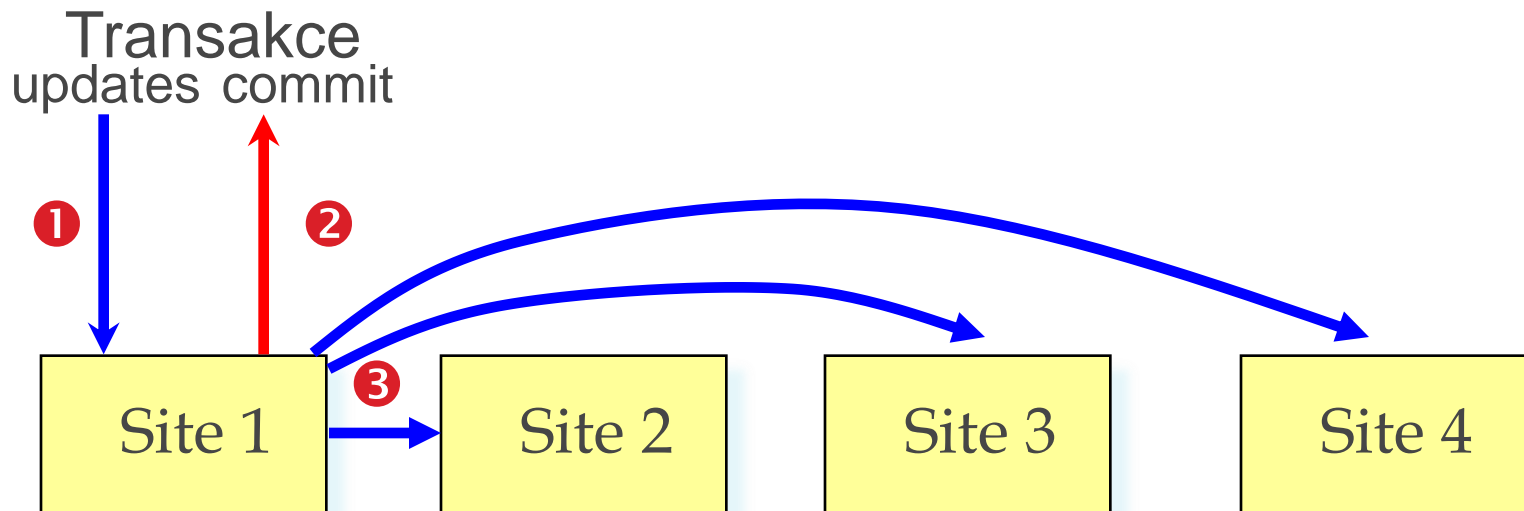
- ❑ Změny se propagují v rámci transakce, realizující změny. Požadované vlastnosti transakcí (ACID) platí pro úpravy na všech místech. Po řádném ukončení aktualizace obsahují všechny repliky stejné hodnoty.
- ❑ ROWA protokol: Read-one/Write-all
- ❑ Nevýhody – transakce může být řádně ukončena až po provedení všech změn.



Asynchronní replikace

12

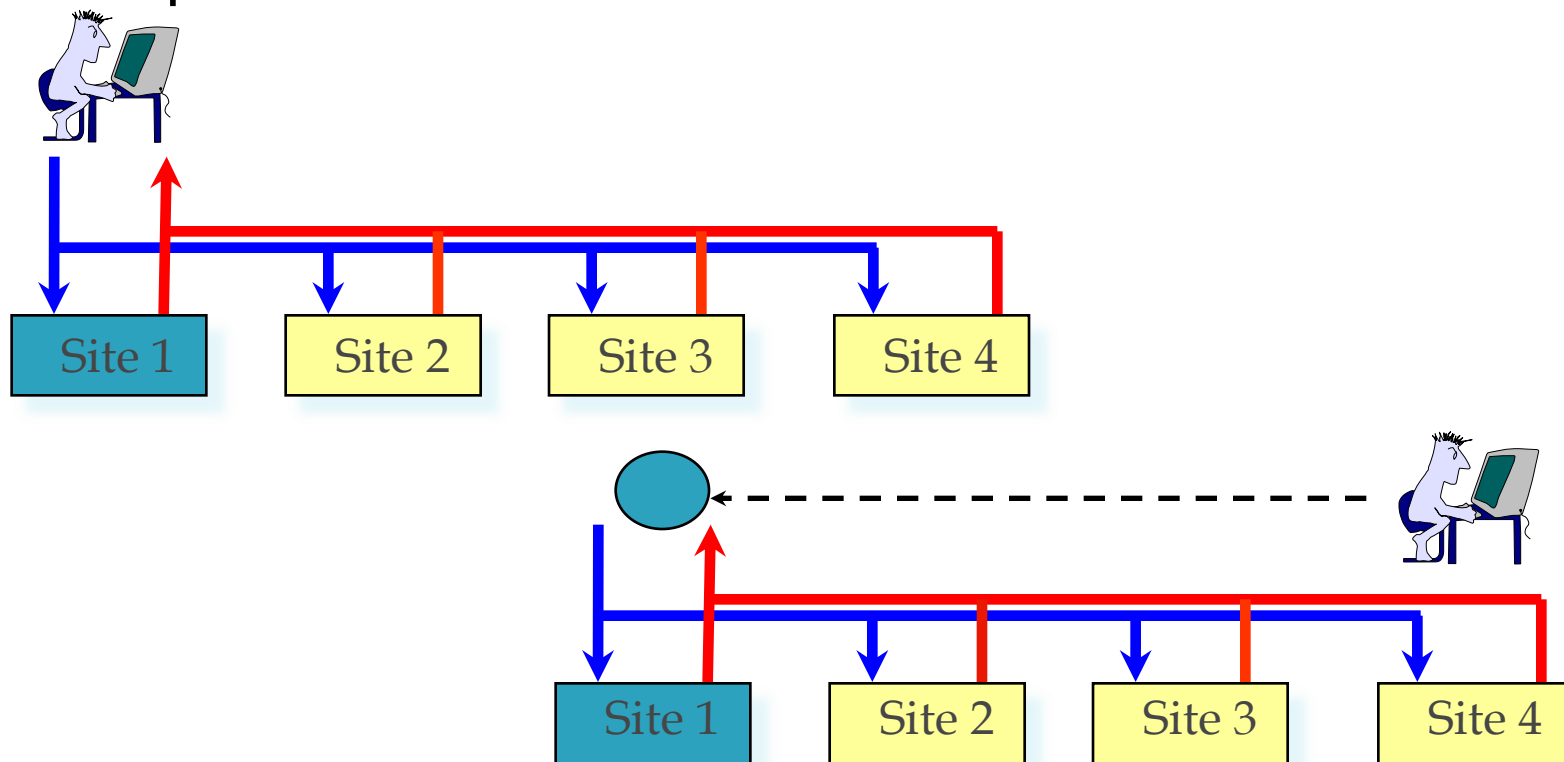
- ❑ Asynchronní replikace nejdříve vykoná aktualizaci na jednom místě (master, primary copy). Poté, co transakce skončí jsou změny propagovány na ostatní kopie. Tj. transakce nečeká na ukončení propagace změn, ale je ukončena ihned po realizaci změn na primární kopii.
- ❑ Repliky mohou být po určitý čas v nekonzistentním stavu.
- ❑ Úprava dat na primární kopii je rychlejší.



Centralizované repliky

13

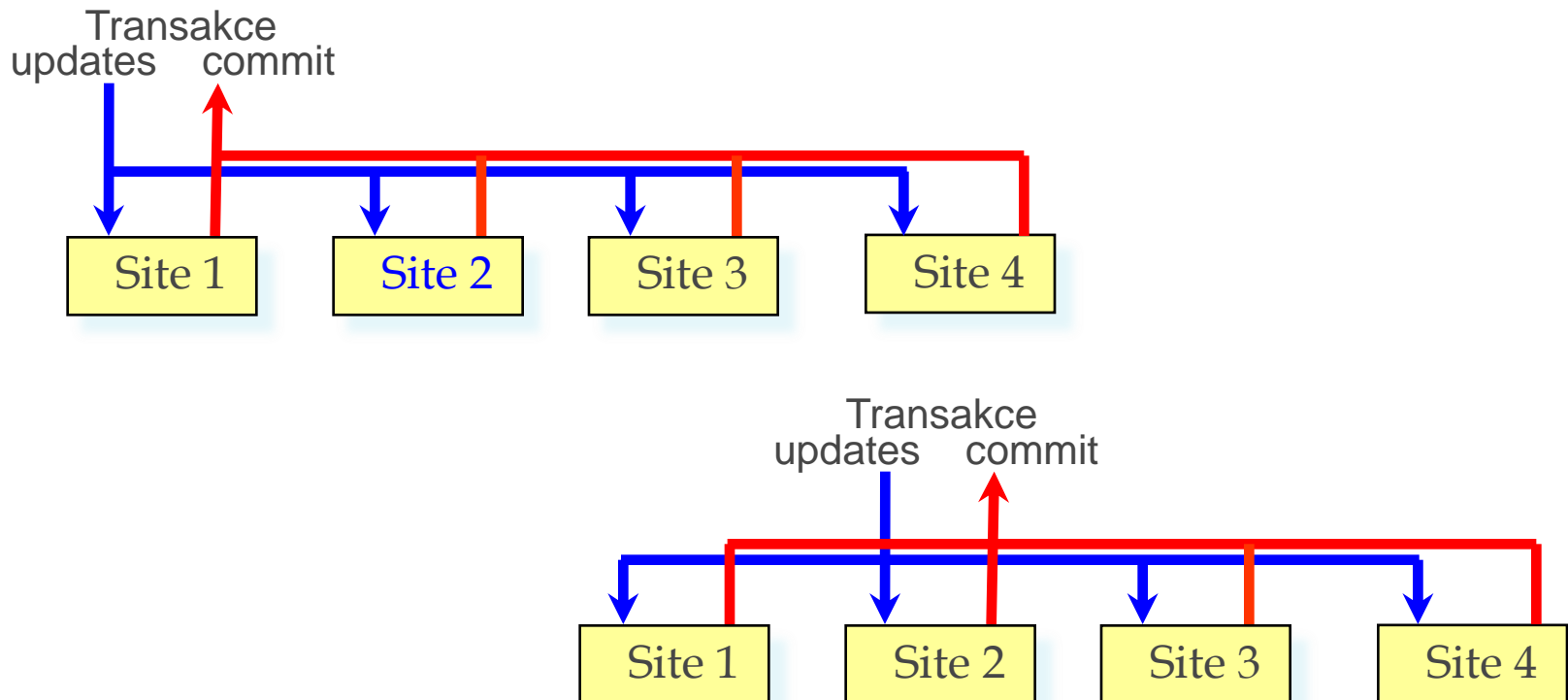
Pouze jedna kopie může být upravována (primary, master), všechny ostatní (slaves) jsou upravovány na základě dat v master kopii.



Distribované repliky

14

Změny mohou být iniciovány na libovolné replice. Každé místo, ve kterém je umístěná replika, může upravovat data.



Formy replikace

15

Synchronní

- + Všechny kopie jsou konzistentní (identické)
- + Lokální kopie poskytují aktuální data
- + Změny jsou atomické
- Transakce musí upravit všechny repliky
 - Delší čas potřebný pro realizaci změn
 - Nízká dostupnost

Asynchronní

- + Transakce je vždy lokální (rychlá odezva)
- + Data jsou po určitý čas nekonzistentní
- Lokální kopie nemusí mít vždy aktuální data
- Změny na všech kopiích nejsou vždy garantovány
- Replikace není transparentní

Centralizovaná

- + Není nutná synchronizace míst (probíhá na master kopii)
- + Vždy existuje jedno místo, kde jsou zaznamenány všechny změny
- Zatížení master kopie může být vysoké
- Lokální kopie nemusí mít vždy aktuální data

Distribuovaná

- + V každém místě může probíhat transakce
- + Zátěž je rozložena rovnoměrně
- Kopie je nutné synchronizovat

Replikační protokoly

16

Předchozí přístupy lze skombinovat do 4 různých replikačních protokolů:

Synchronní	Synchronní centralizované	Synchronní distribuované
Asynchronní	Asynchronní centralizované	Asynchronní distribuované
	Centralizované	Distribuované

Replikační strategie

17

Synchronní	<ul style="list-style-type: none">+ Změny nemusí být koordinované+ Nenastávají inkonzistence- Delší čas odezvy- Užitečné pokud potřebujeme málo úprav- Lokální kopie jsou pouze ke čtení	<ul style="list-style-type: none">+ Nenastávají inkonzistence+ Elegantní (symetrické řešení)- Dlouhý čas odezvy- Úpravy musí být koordinované
	<ul style="list-style-type: none">+ Není potřebná koordinace+ Krátký čas odezvy- Lokální kopie nejsou aktuální- Inkonzistence	<ul style="list-style-type: none">+ Není centrální koordinace+ Krátký čas odezvy- Inkonzistence- Úpravy mohou být ztraceny
Centralizované		Distribuované

Využití replikace

18

- ❑ distribuce dat na síťové servery, včetně mobilních a připojených jen příležitostně
- ❑ sjednocení dat s centrálním serverem
- ❑ rozdělení procesu na více než jeden server
- ❑ informační tok z jednoho serveru na další
- ❑ sdílení dat mezi více místy

V praxi je také nutno rozhodnout, zda všechny pobočky potřebují aktuální informace, nebo zda některým stačí třeba informace ze včerejšího dne. Musíme posoudit, kdo a kdy potřebuje které informace, kdo a kdy je aktualizuje a kdo a kdy je vytváří.