

Replikace

Přednášky z distribuovaných systémů

Pro a proti replikaci

1. Zvýšení spolehlivosti.
2. Zvýšení výkonnosti.
3. Nutnost zachování škálovatelnosti systému co do počtu komponent i geografické rozlehlosti.
4. Výkonnost se snižuje, protože s opravou jedné kopie souvisí i oprava ostatních kopií pro zachování konzistentnosti.

Vztah ke škálovatelnosti (1)

- Pro zlepšení škálovatelnosti jsou používány replikace a cache.
- Škálovatelnost je obecně prezentována jako problém průchodnosti.
- Replikace dat v blízkosti potenciálních uživatelů může zlepšit výkonnost a zlepšit škálovatelnost.

Potenciální problémy:

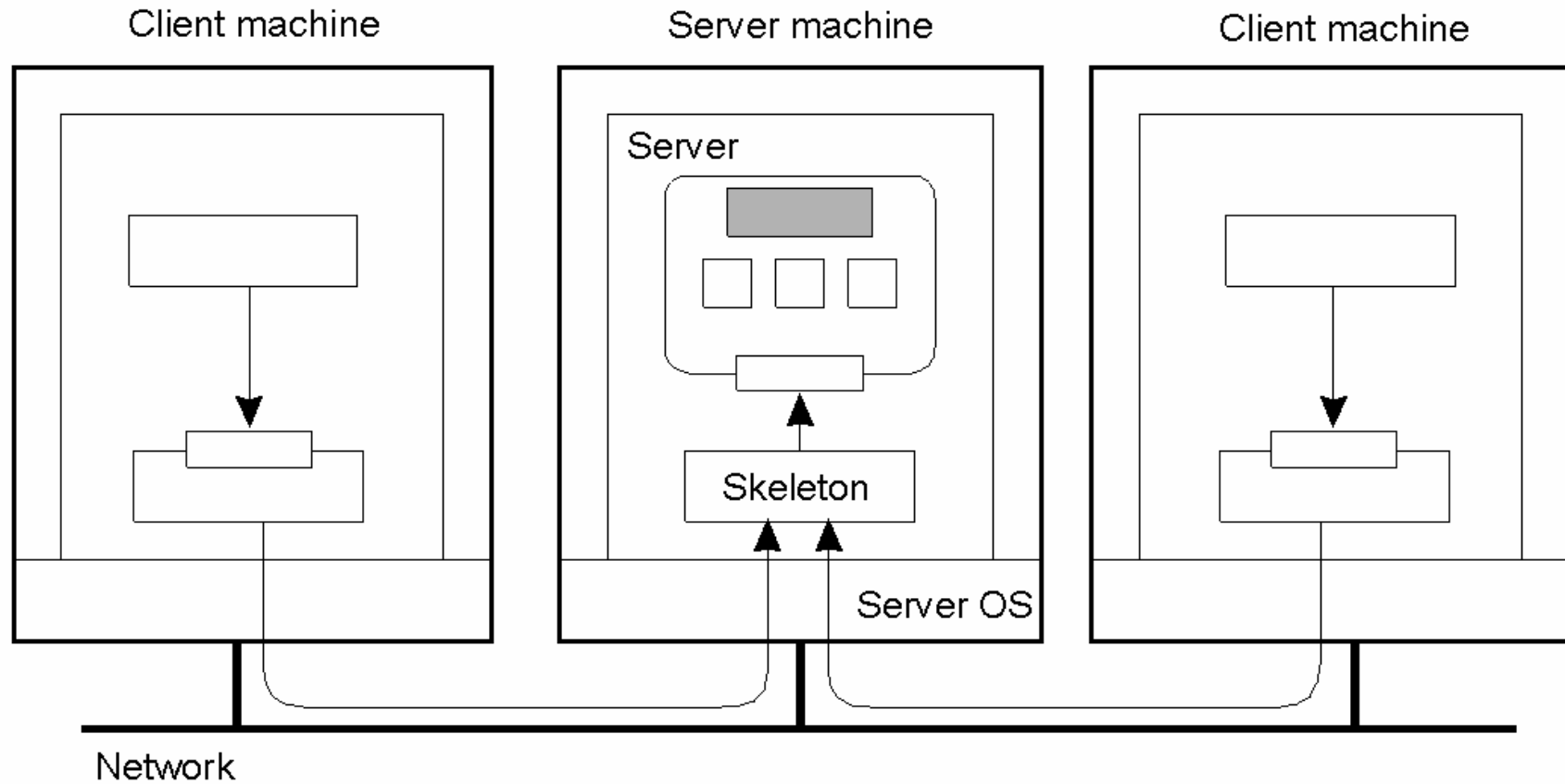
- Šířka pásma požadovaná pro aktualizaci kopií může být velká.
- Udržování konzistentních replikovaných kopií může být problémové z pohledu škálovatelnosti.

Proto je třeba **dodržovat určité dohody při replikaci dat.**

Vztah ke škálovatelnosti (2)

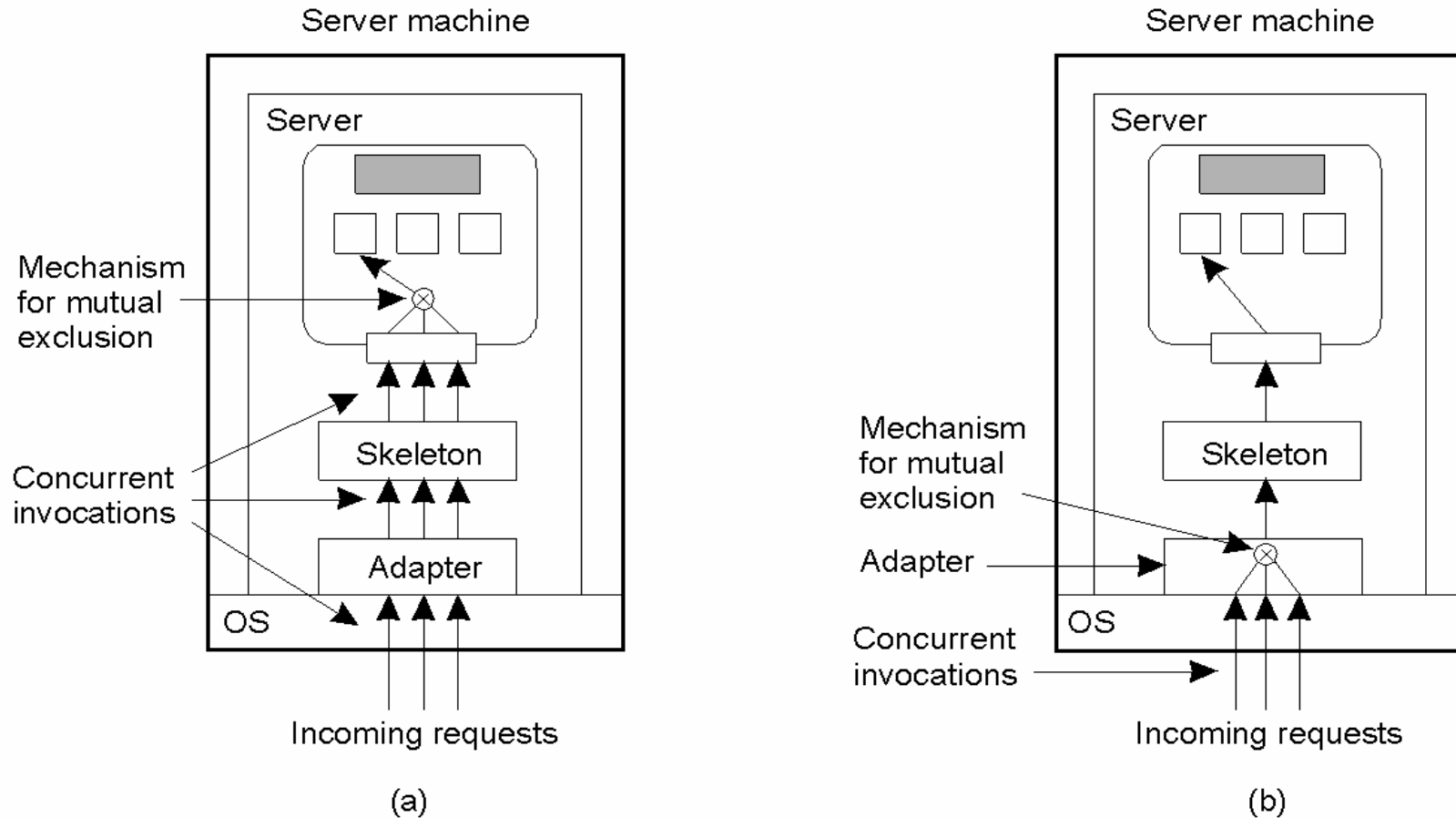
- Vícenásobné kopie.
 - zvyšují výkonnost a redukují dobu přístupu
 - zvyšují také režii pro udržení konzistentnosti
 - příklad: N - krát replikované objekty
 - frekvence čtení R , frekvence zápisu W
 - je-li $R \ll W$ vysoká konzistentnost zvyšuje režii, zprávy jsou zbytečné
- Zdrojem řešení je udržování konzistentnosti
 - volba vhodné sémantiky
 - těsná konzistentnost vyžaduje globálně synchronizované hodiny
- Řešení: snížíme požadavky na konzistentnost
 - jsou k dispozici různé stupně konzistentnosti

Replikace objektů (1)



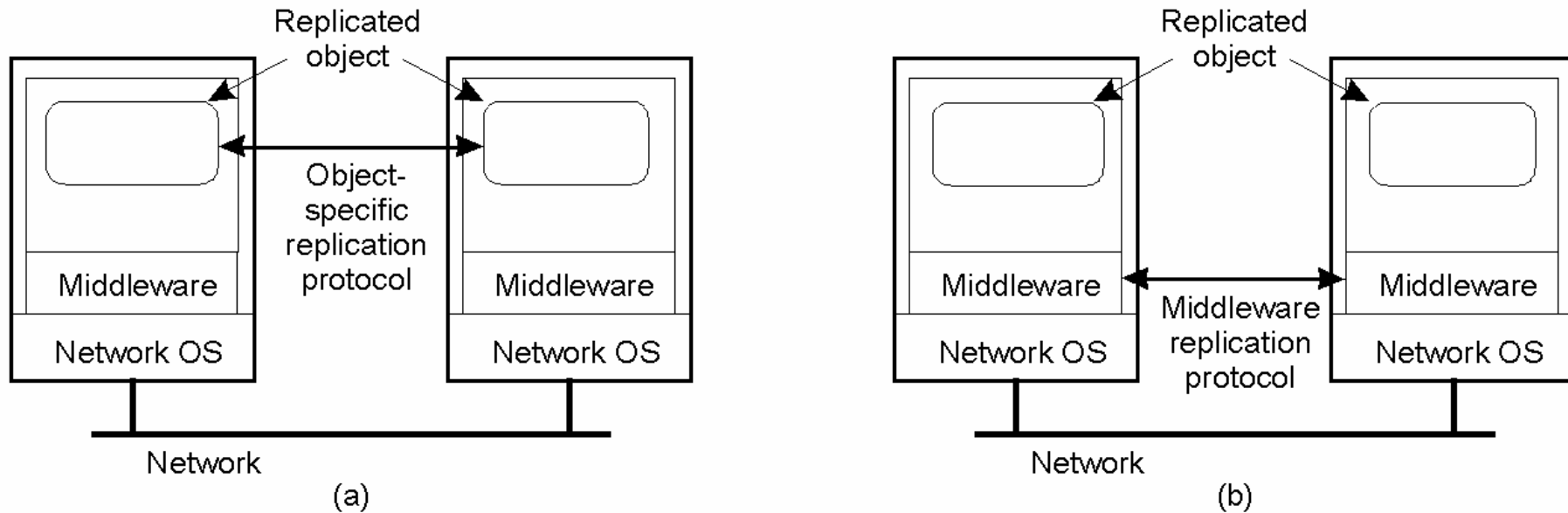
Organizace distribuovaných vzdálených objektů sdílených dvěma různými klienty.

Replikace objektů (2)



- a) Vzdálené objekty schopné zpracovávat souběžné vyvolání sebe sama.
- b) Vzdálený objekt pro který je vyžadován adaptér objektu ke zpracování souběžných volání.

Replikace objektů (3)

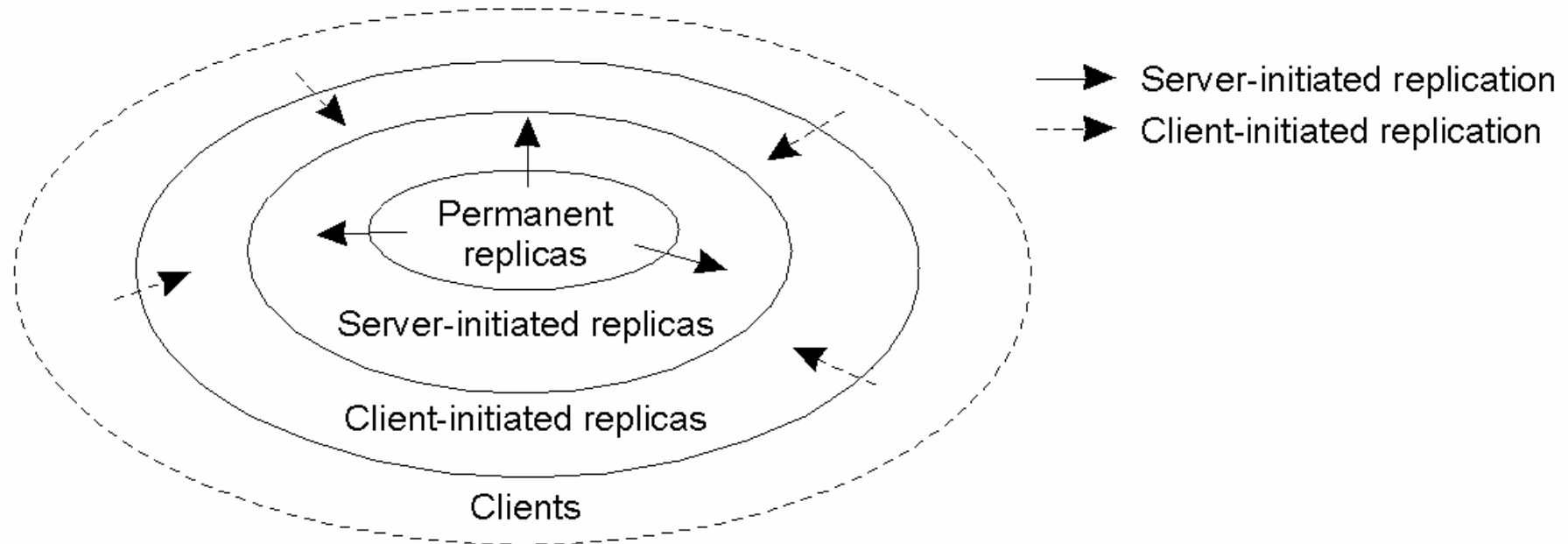


- a) Distribuovaný systém pro replikovatelné distribuované objekty – aplikace je schopná replikace.
- b) Distribuovaný systém schopný managementu replik – replikace zpracovává middleware.

Distribuční protokoly

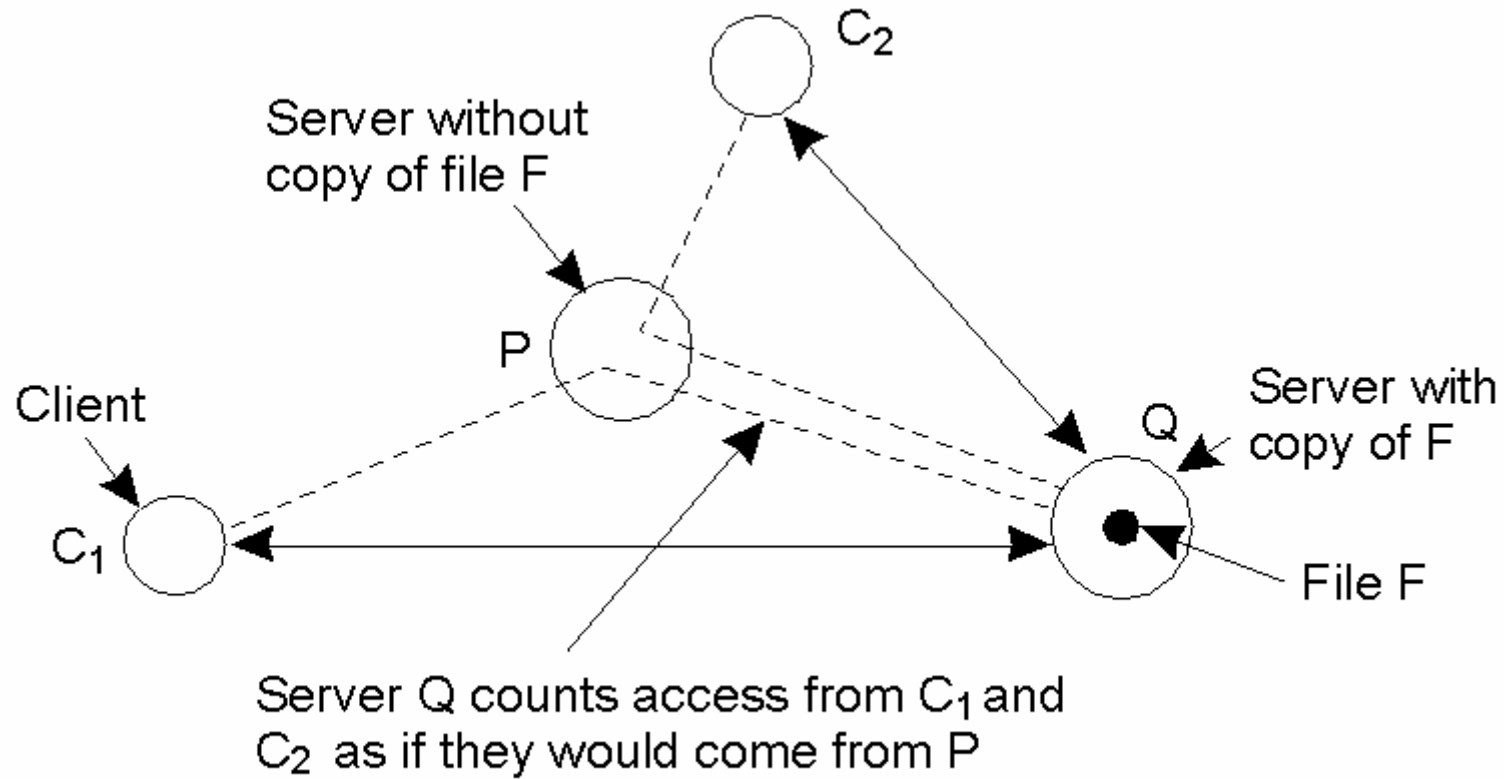
- Rozumná otázka na této úrovni zní: pokud je zvolen model konzistentnosti, jak jsou aktuální opravy šířeny v DS?
- K odpovědi na tuto otázku se musíme podívat na několik distribučních protokolů.
- Avšak předtím ještě musíme udělat rozhodnutí týkající se vhodného umístění replik.

Uspořádání replikací dat



Logická organizace různých typů replikací datové paměti znázorněná jako tři soustředné kruhy.

Repliky iniciované serverem



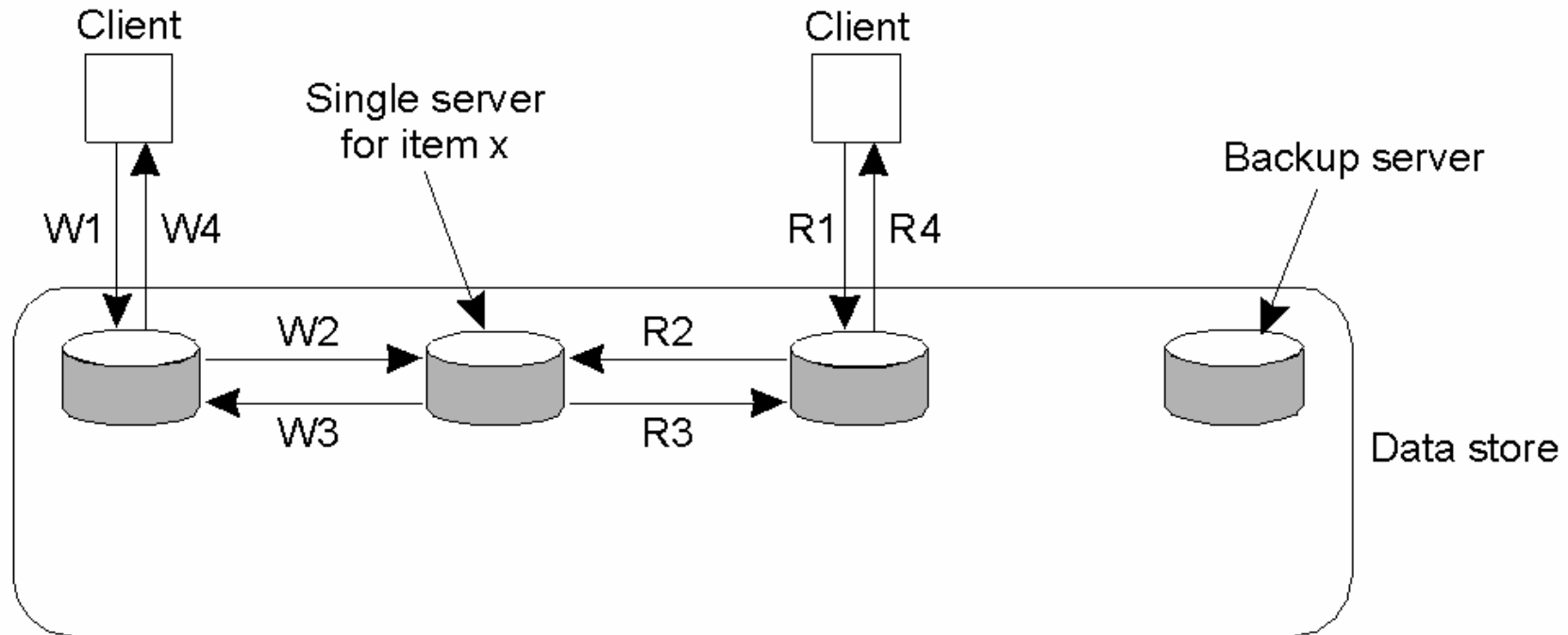
Čítání požadavků přístupu od různých klientů.

Porovnání protokolů push a pull

	Operace push	Operace pull
Stav serveru	Seznam klientů replik a cache paměť	žádný
Posílané zprávy	Oprava nebo zpožděné vykonání opravy	Hlasování a oprava
Doba reakce na klientu	Bezprostřední nebo čas vykonání opravy	Čas vykonání opravy

Porovnání mezi protokoly založenými na modelu push a pull operacích v případě systému s více klienty a jedním serverem.

Protokoly se vzdáleným zápisem (1)



W1. Write request

W2. Forward request to server for x

W3. Acknowledge write completed

W4. Acknowledge write completed

R1. Read request

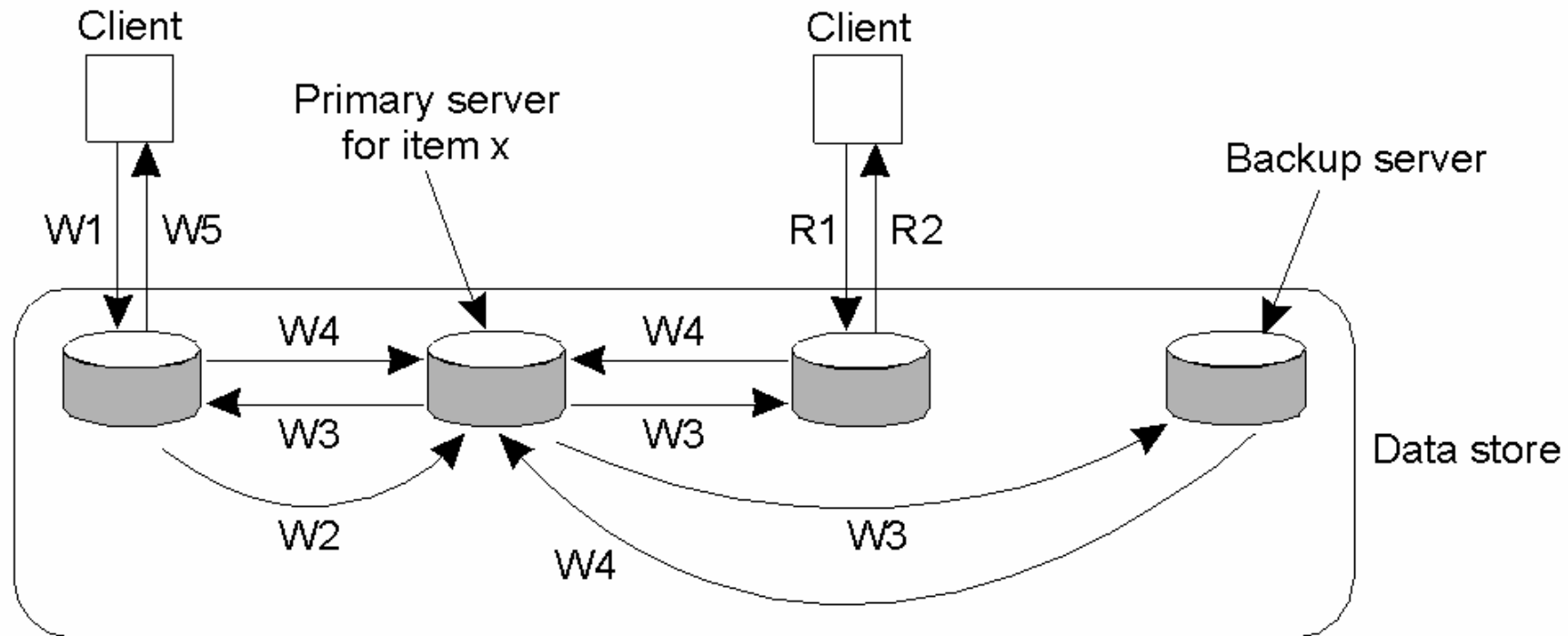
R2. Forward request to server for x

R3. Return response

R4. Return response

Protokol vzdáleného zápisu do primární kopie s fixním serverem do kterého jsou posílány všechny operace čtení i zápisu.

Protokoly se vzdáleným zápisem (2)

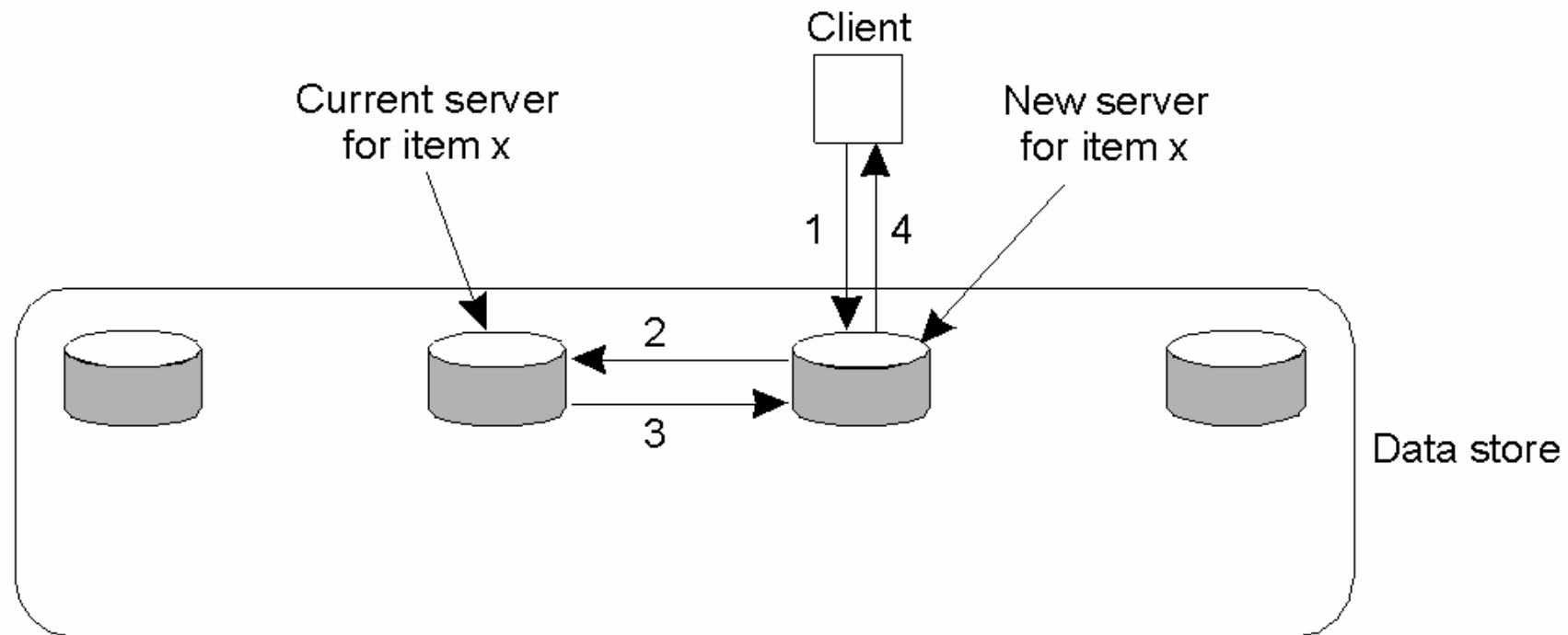


W1. Write request
W2. Forward request to primary
W3. Tell backups to update
W4. Acknowledge update
W5. Acknowledge write completed

R1. Read request
R2. Response to read

Princip protokolu se zálohou primární kopie.

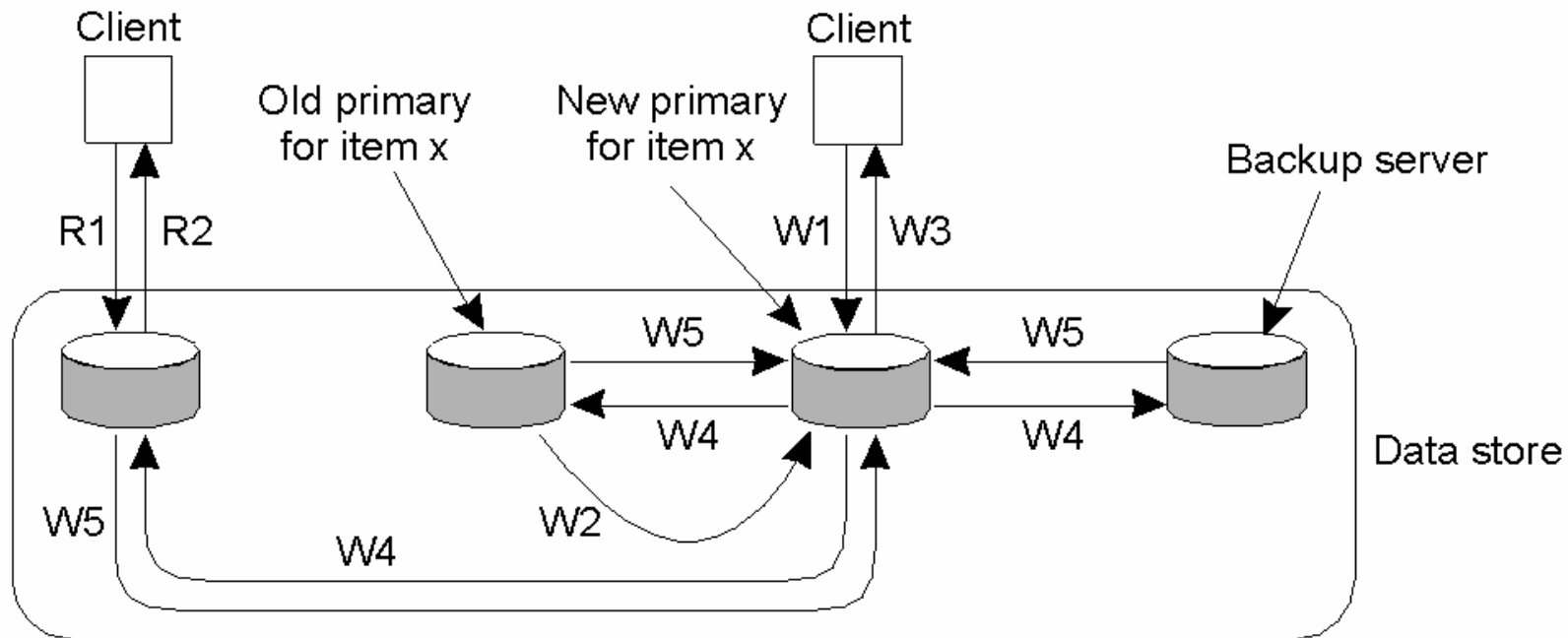
Protokoly s lokálním zápisem (1)



1. Read or write request
2. Forward request to current server for x
3. Move item x to client's server
4. Return result of operation on client's server

Protokol lokálního zápisu do primární kopie, kdy jedna kopie migruje mezi procesy.

Protokoly s lokálním zápisem (2)



W1. Write request

W2. Move item x to new primary

W3. Acknowledge write completed

W4. Tell backups to update

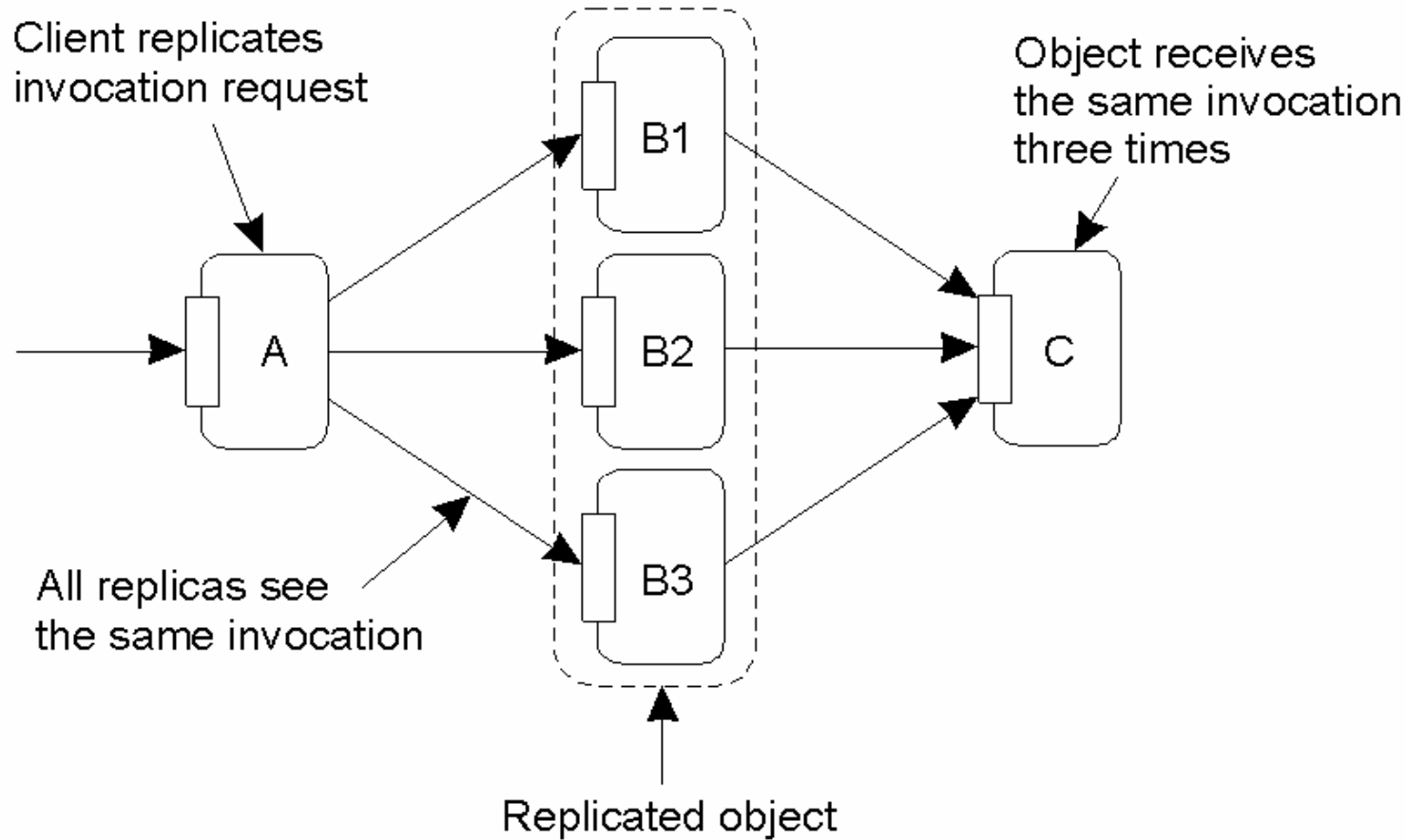
W5. Acknowledge update

R1. Read request

R2. Response to read

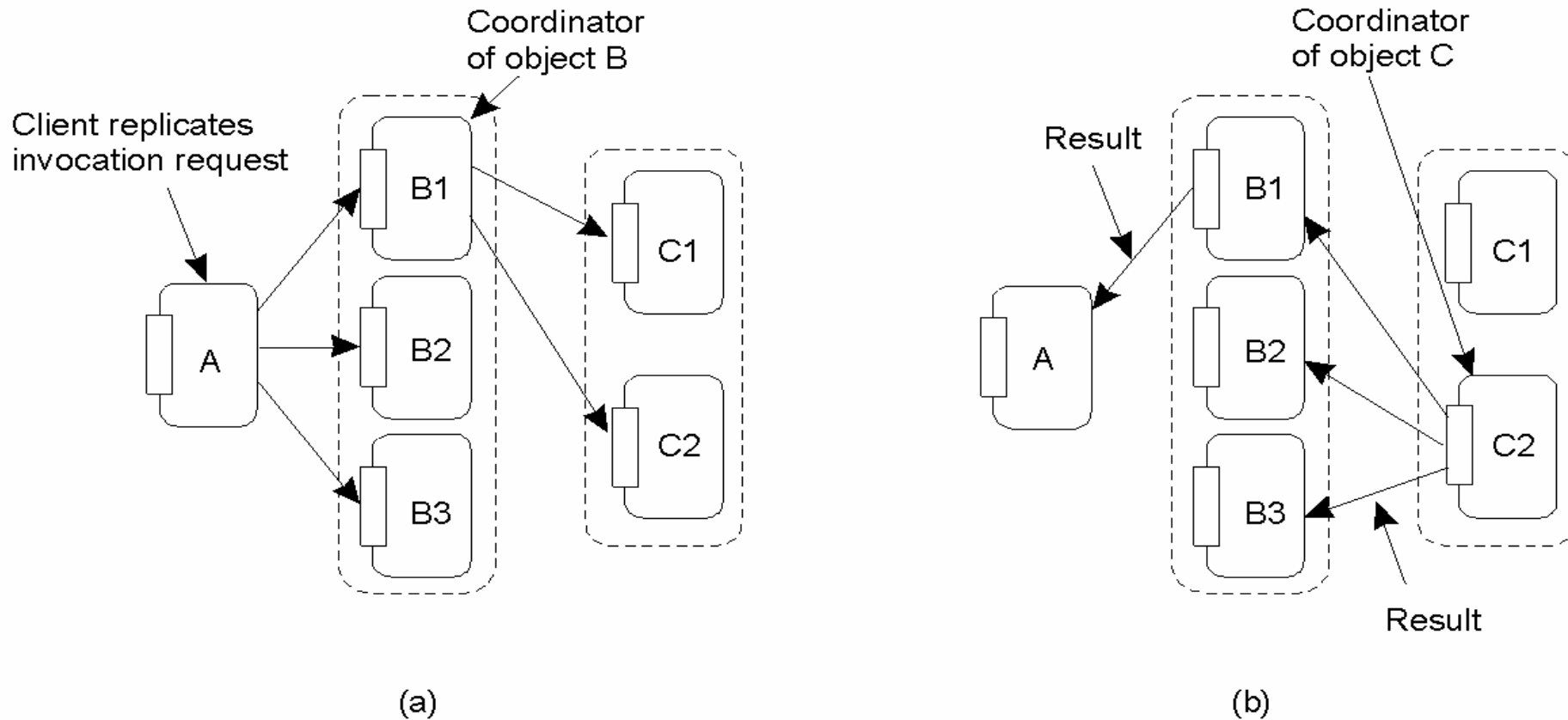
Protokol primární kopie, kdy primární kopie migruje k procesům, čekajícím na provedení opravy.

Aktivní replikace (1)



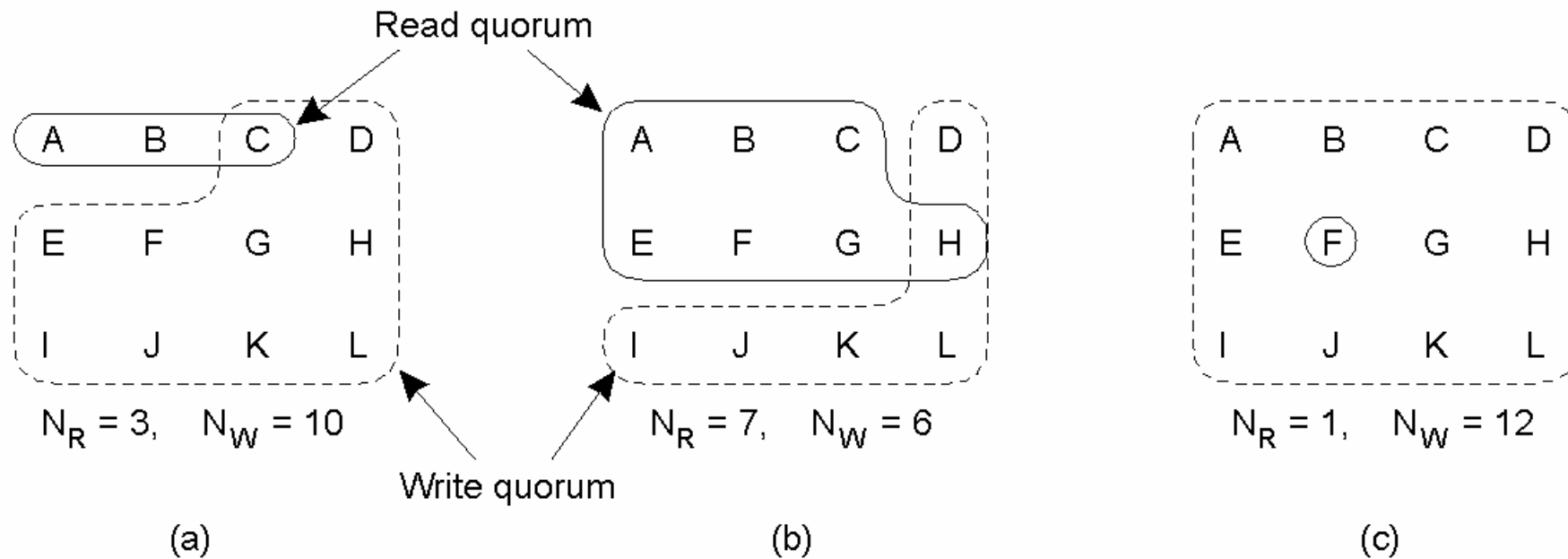
Problém replikovaných vyvolání.

Aktivní replikace (2)



- a) Posílání požadavku vyvolání z replikovaného objektu.
- b) Vrácení odpovědi replikovanému objektu.

Protokoly založené na hlasování



Algoritmu hlasování:

- a) Korektní volba množiny pro čtení a zápis
- b) Volba, která může vézt ke konfliktu typu zápis – zápis
- c) Správná volba, známá jako ROWA (read one, write all)

Orca

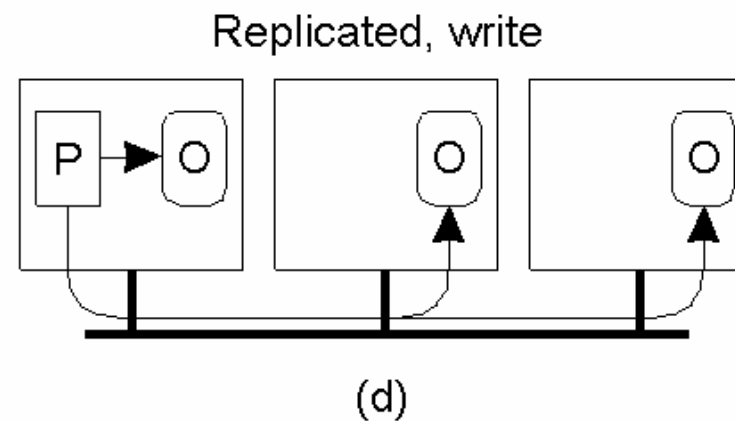
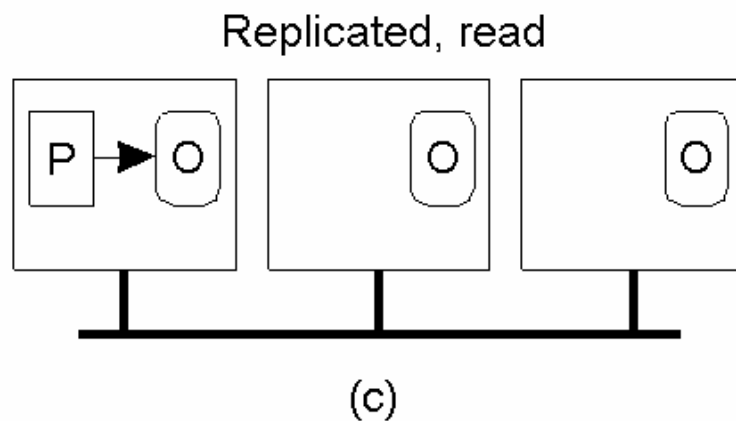
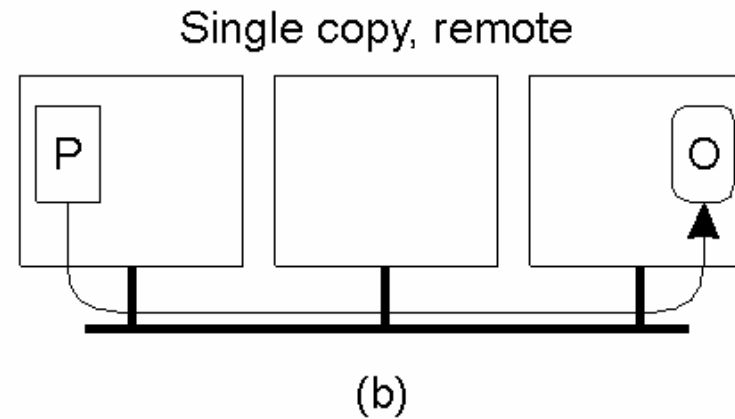
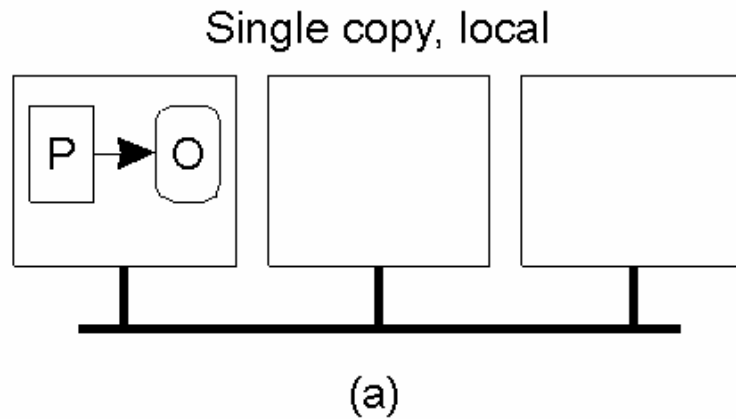
```
OBJECT IMPLEMENTATION stack;
  top: integer;                                # variable indicating the top
  stack: ARRAY[integer 0..N-1] OF integer      # storage for the stack

  OPERATION push (item: integer)              # function returning nothing
  BEGIN
    GUARD top < N DO
      stack [top] := item;                    # push item onto the stack
      top := top + 1;                         # increment the stack pointer
    OD;
  END;

  OPERATION pop():integer;                    # function returning an integer
  BEGIN
    GUARD top > 0 DO
      top := top - 1;                         # suspend if the stack is empty
      RETURN stack [top];                    # decrement the stack pointer
    OD;
  END;
  BEGIN
    top := 0;                                # return the top item
  END;                                         # initialization
```

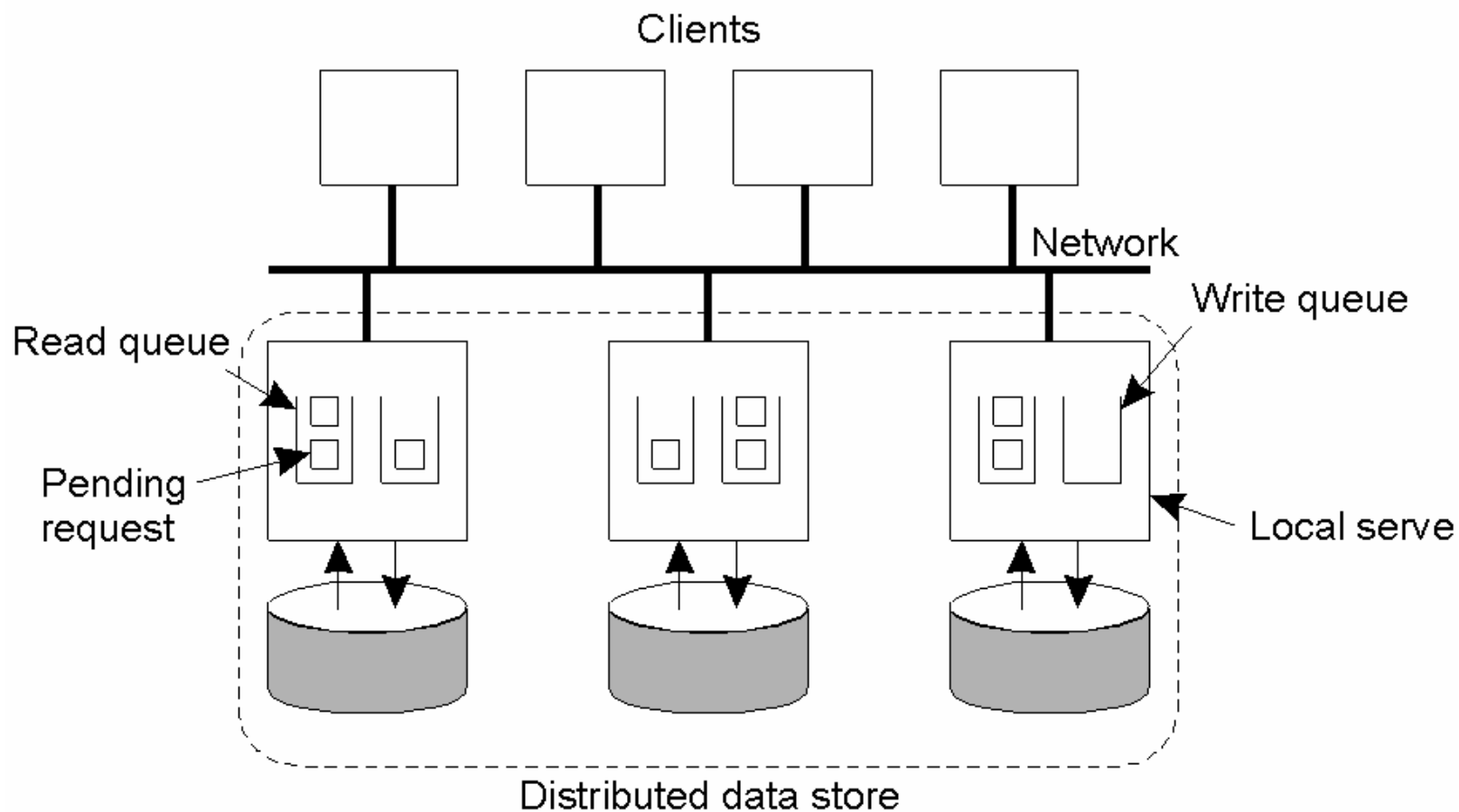
Zjednodušený objekt zásobníku v Orca s interními daty a dvěma operacemi.

Řízení sdílených objektů v Orca



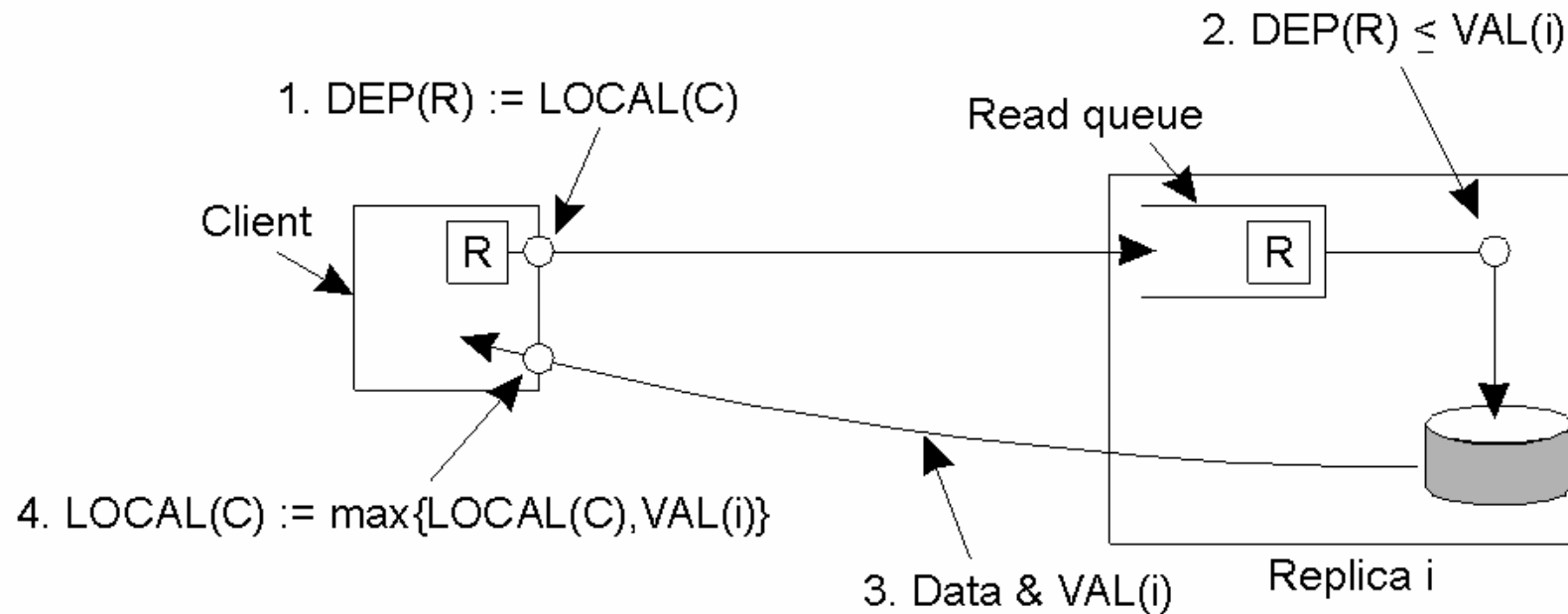
Čtyři případy procesu P provádějícího operace nad objektem O v Orca.

Příčinně konzistentní líná replikace



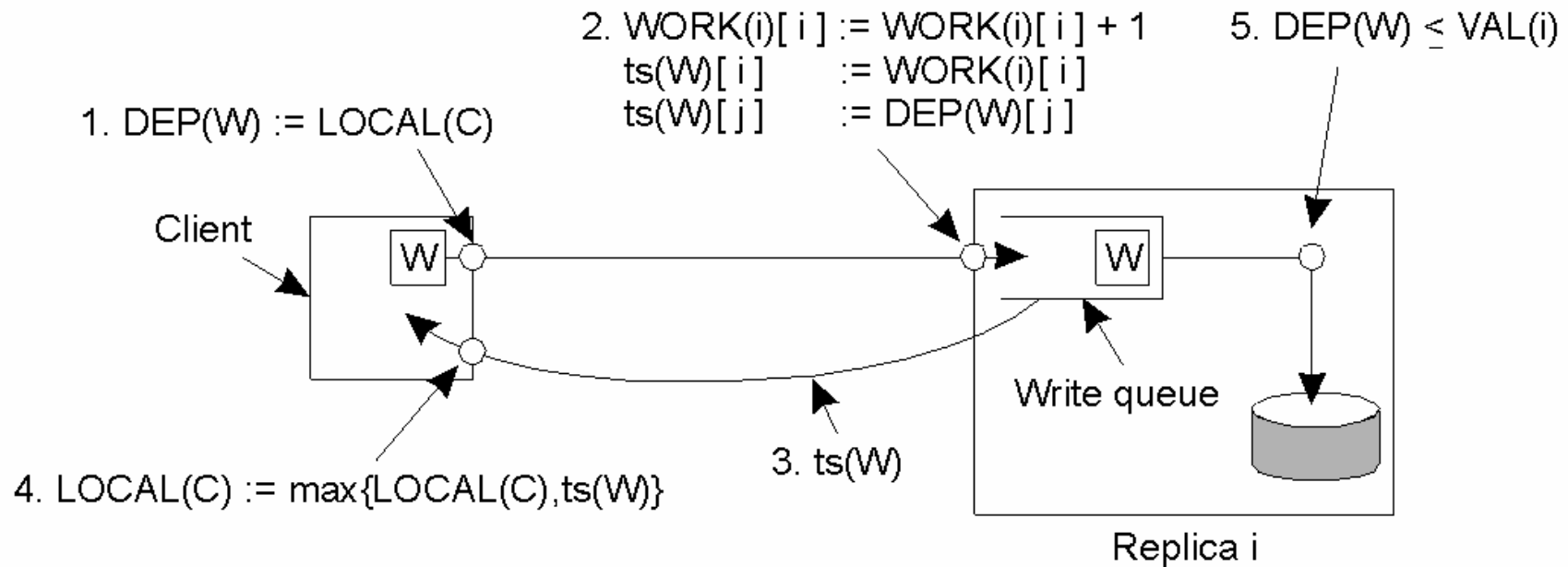
Obecná organizace distribuované paměti. Klienti na sebe také přebírají zpracování konzistentní komunikace.

Zpracování operací čtení



Provádění operace čtení nad lokální kopií.

Zpracování operací zápisu



Provádění operace zápisu nad lokální kopií.