

11. előadás

Konzisztencia és többszörözés

3. rész

Hibatűrés

1. rész

Konzisztencia és többszörözés

3. rész

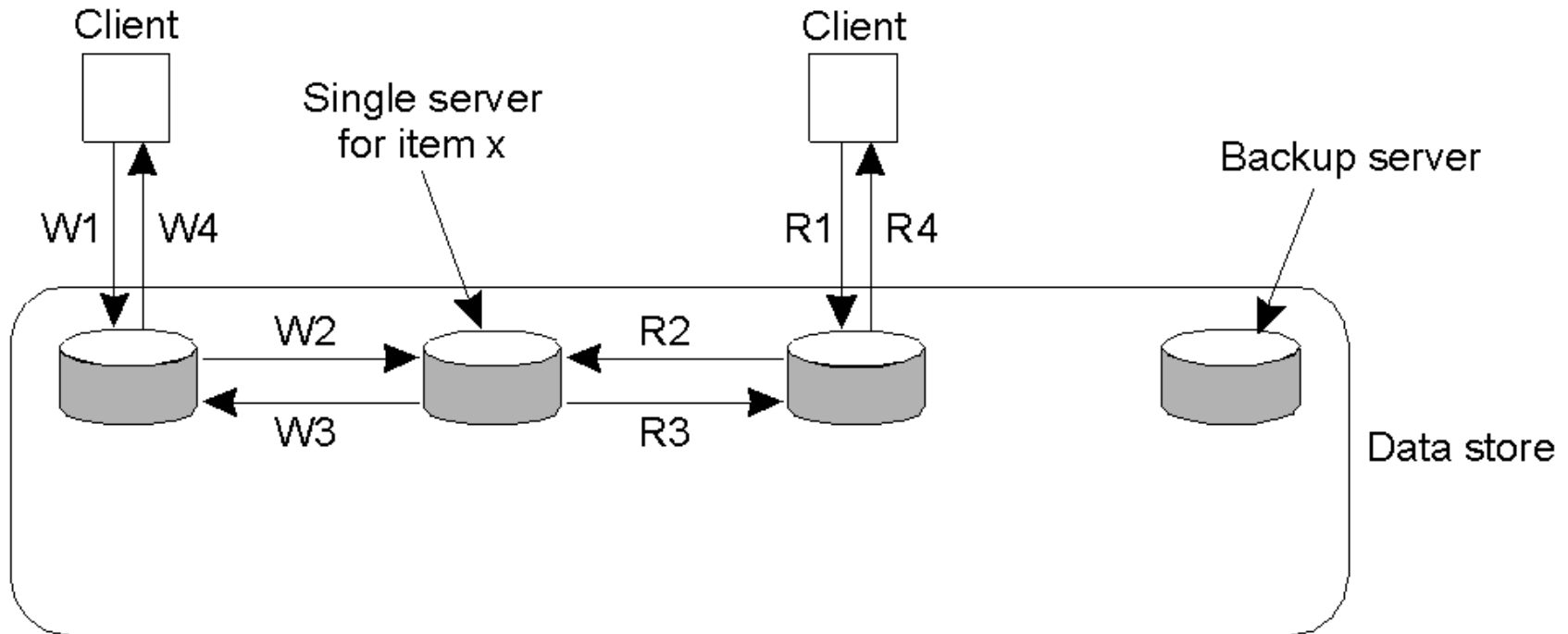
Konzisztenciaprotokollok

(soros)

Elsődleges másolaton alapuló protokollok

Az adattár minden elemének van elsődleges másolata, aki felel az írási műveletek koordinálásáért.

Távoli írás protokollok (1)



W1. Write request

W2. Forward request to server for x

W3. Acknowledge write completed

W4. Acknowledge write completed

R1. Read request

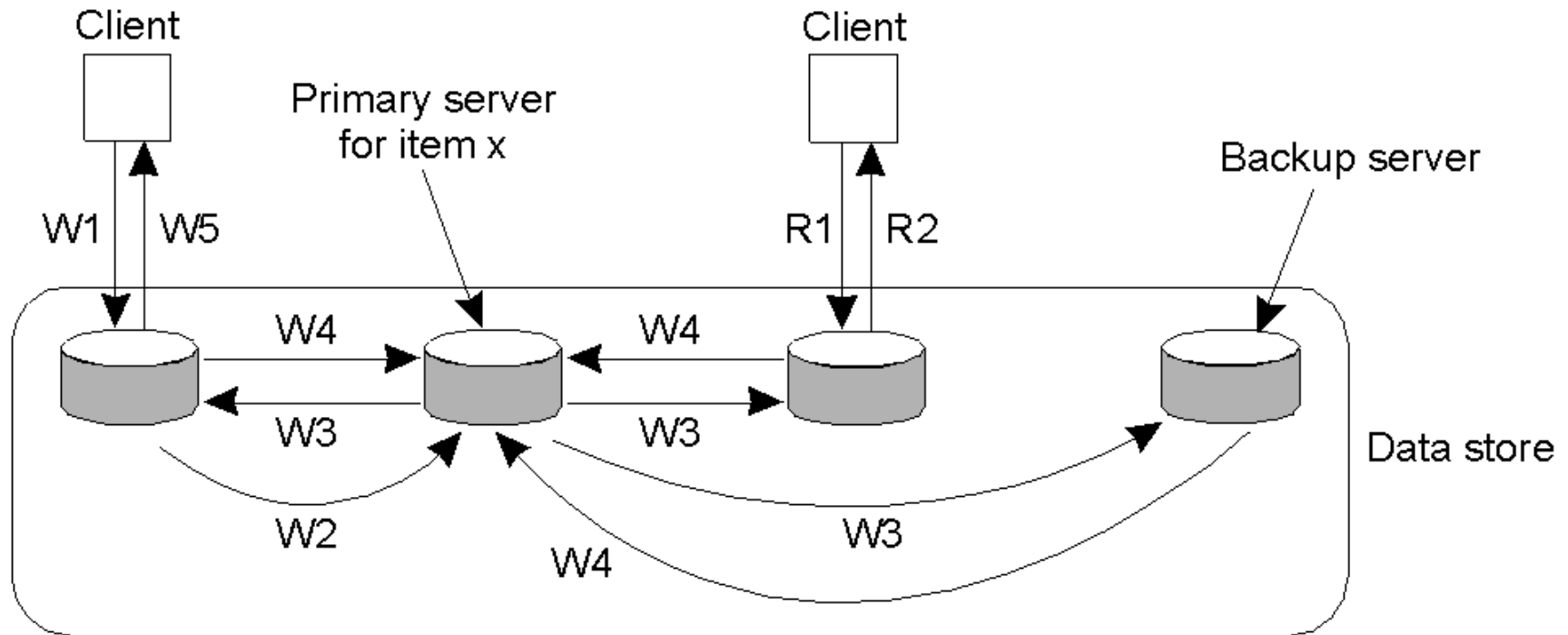
R2. Forward request to server for x

R3. Return response

R4. Return response

Elsődleges távoli írás-protokoll, az olvasási és írási műveletek elvégzésére kizárólagosan hivatott rögzített szerverrel.

Távoli írás protokollok (2)

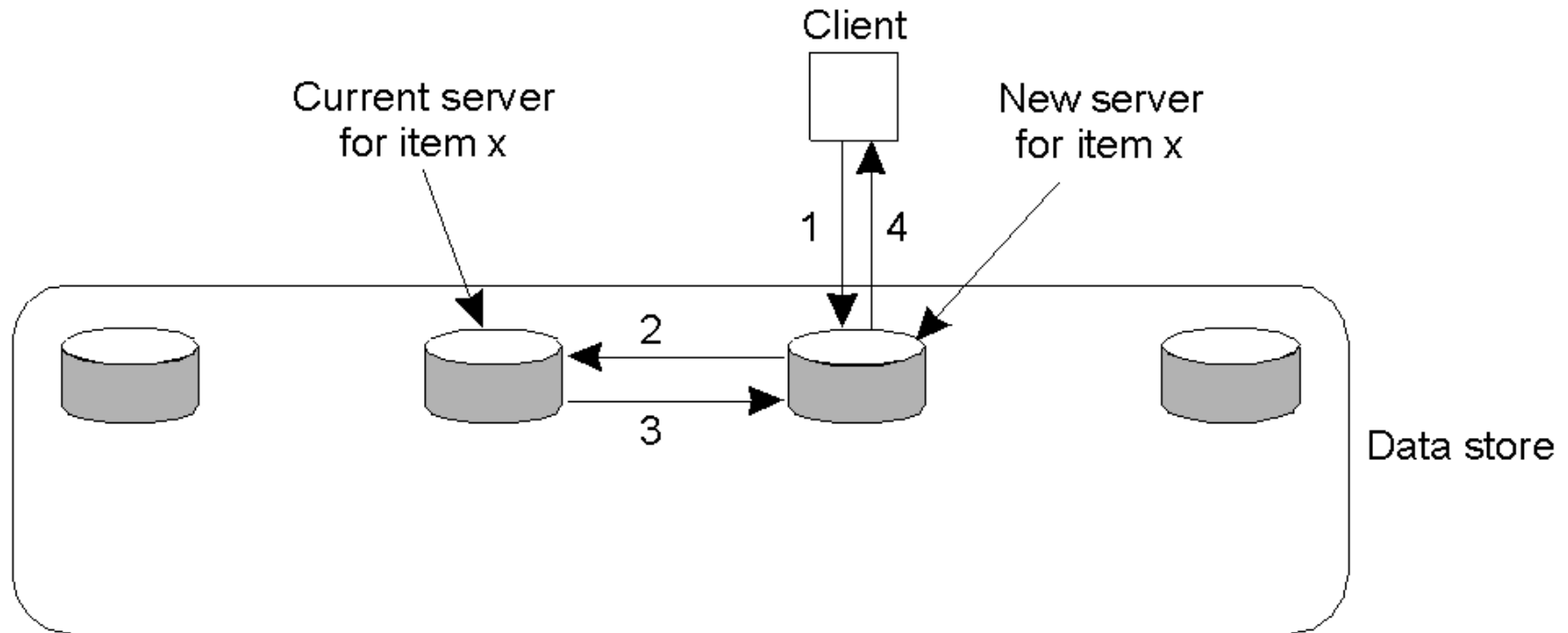


W1. Write request
W2. Forward request to primary
W3. Tell backups to update
W4. Acknowledge update
W5. Acknowledge write completed

R1. Read request
R2. Response to read

Az elsődleges másolatprotokoll alapelve.

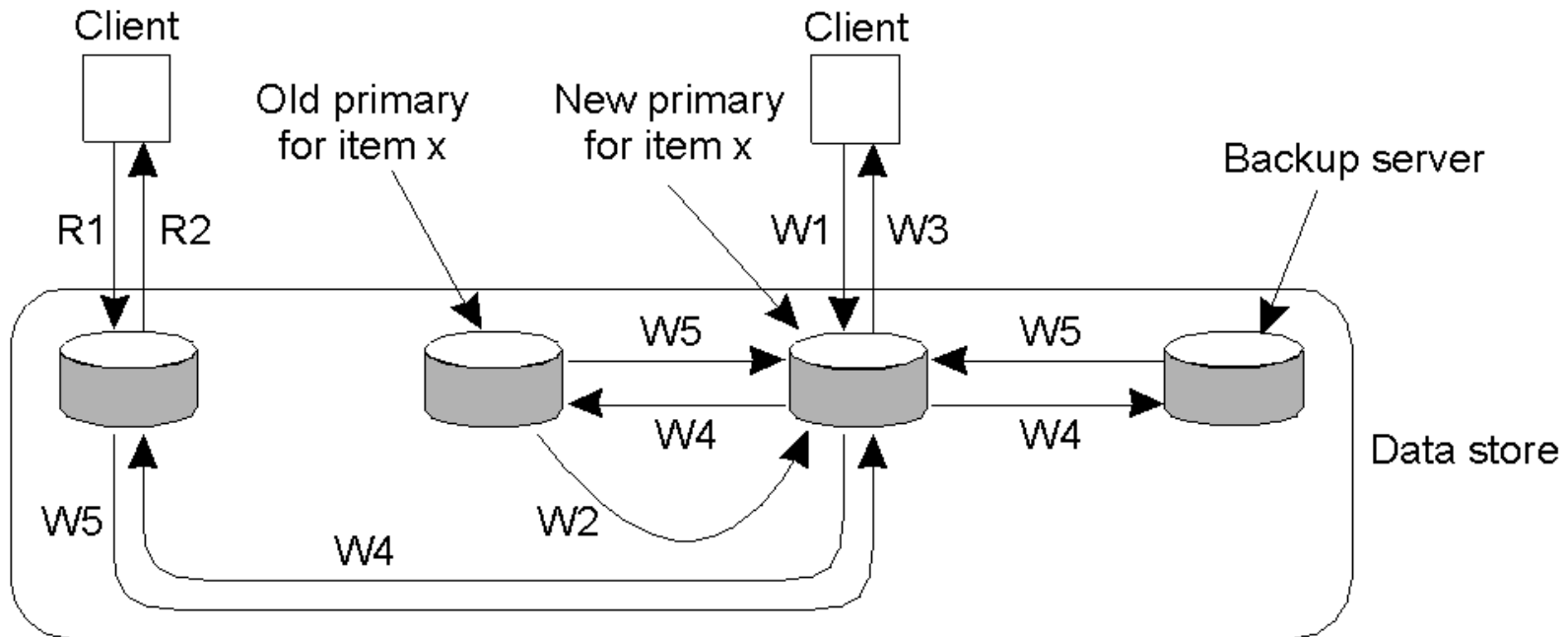
Helyileg író protokollok (1)



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

Elsődleges másolat alapú helyileg író protokoll, amelyben az adatelem egyetlen példánya vándorol.

Helyileg író protokollok (2)



W1. Write request
W2. Move item x to new primary
W3. Acknowledge write completed
W4. Tell backups to update
W5. Acknowledge update

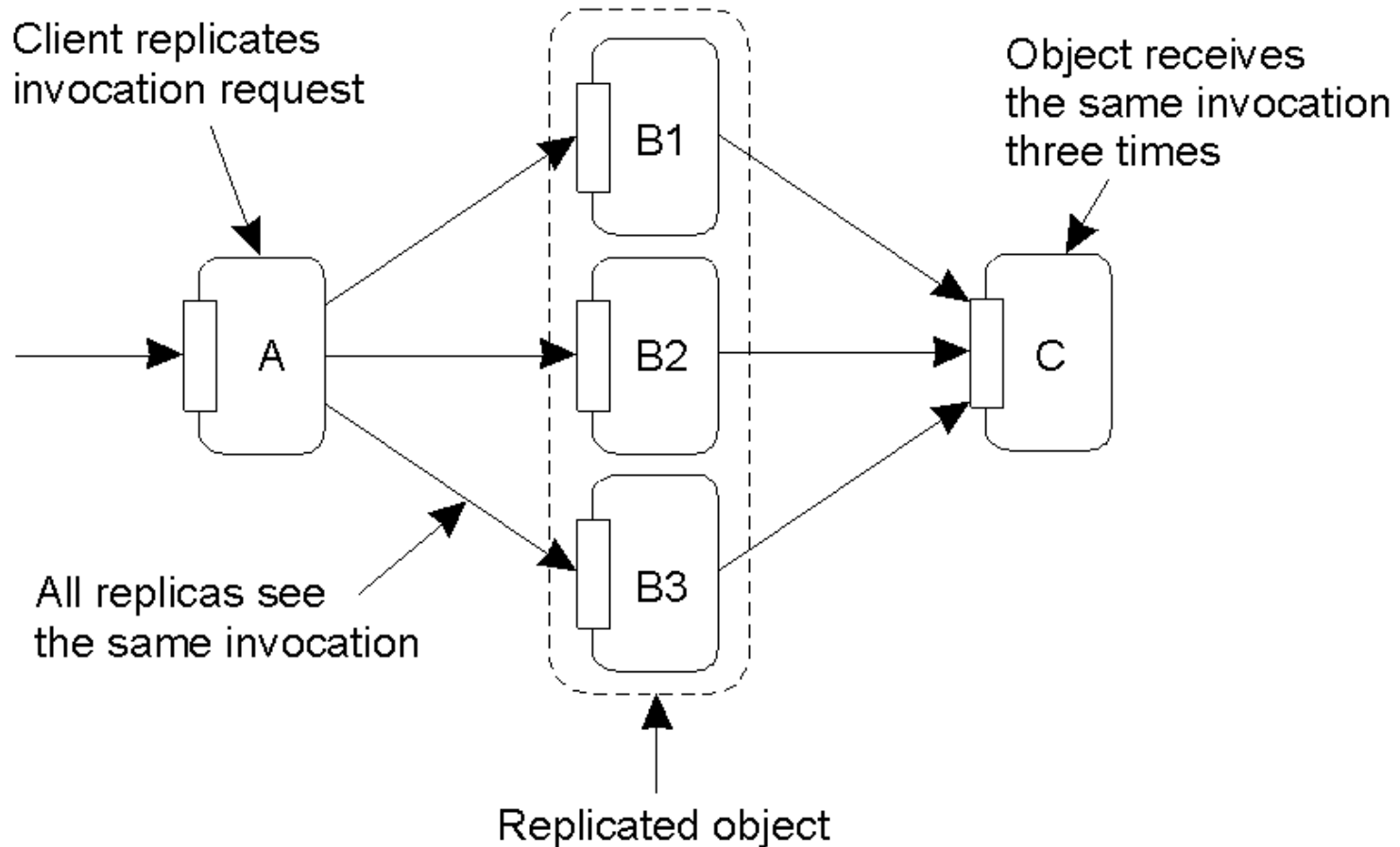
R1. Read request
R2. Response to read

A frissítést végrehajtani kívánó folyamatok között vándorló elsődleges másolat protokollja.

Többszálú írás-protokollok

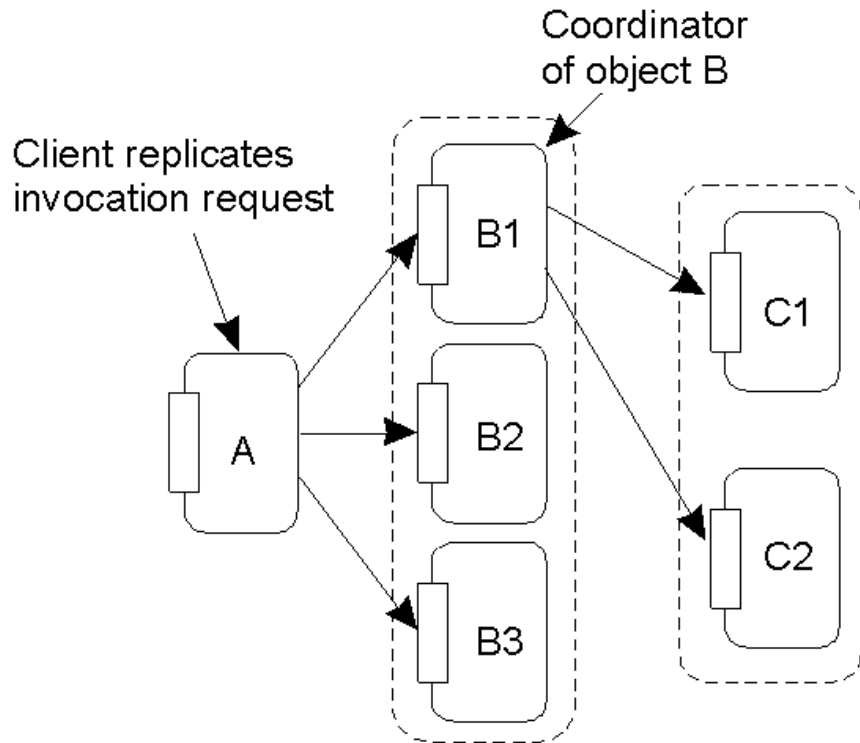
Az írási művelet több másolaton is
végrehajtható

Aktív többszörözés (1)

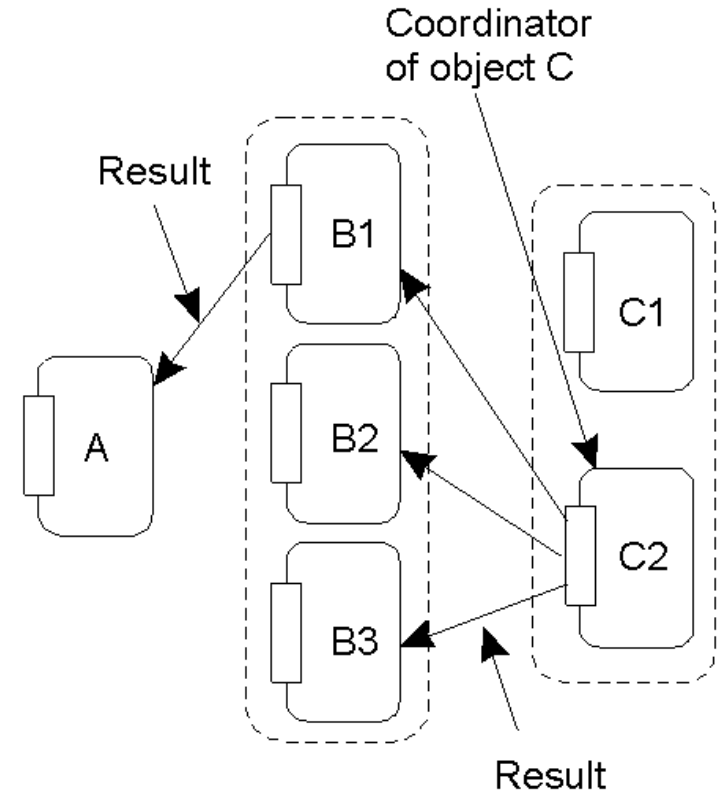


A többszörözött hívás problémája.

Aktív többszörözés (2)



(a)



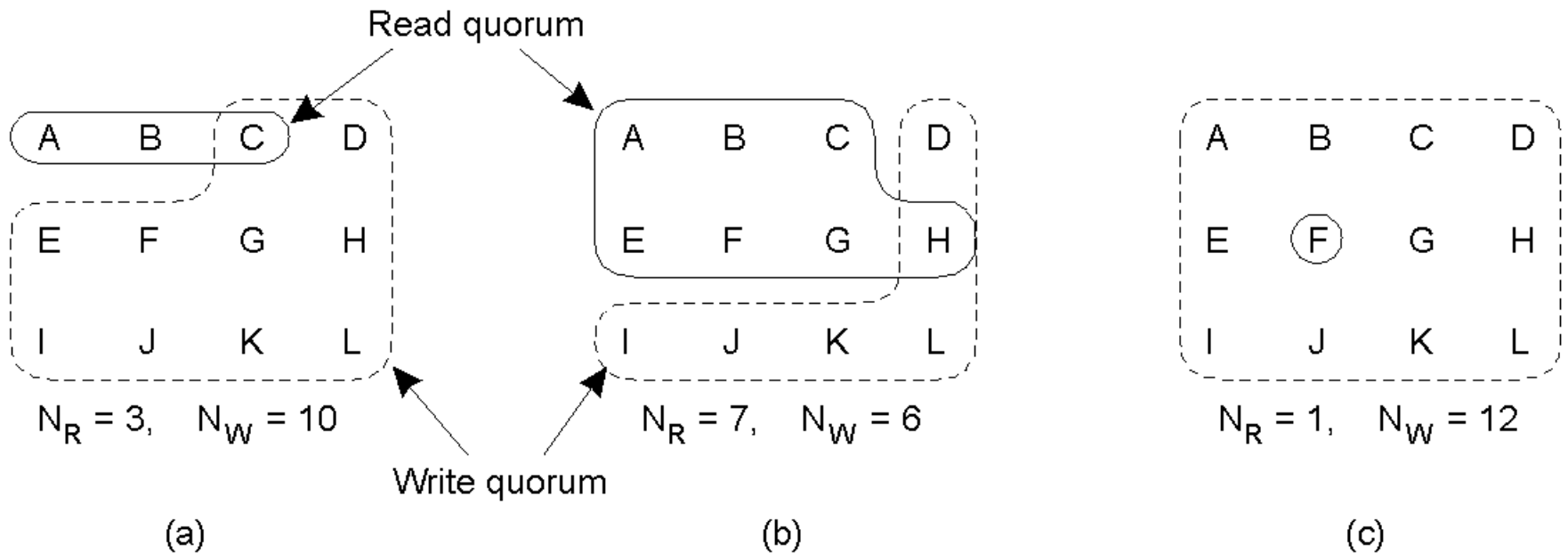
(b)

- a) Továbbítja a többszörözött objektumtól érkező hívási kérést egy másik többszörözött objektumnak.
- b) A válasz visszaküldése az egyik többszörözött objektumtól a másikkak.

Testület alapú protokollok (1)

- Többszörözött írás támogatása szavazás alapján
- Olvasás és írás előtt a többiektől engedélyt kell kérni
- Olvasási testület: N_R
- Írási testület: N_W
- $N_R + N_W > N$
- $N_W > N/2$

Testület alapú protokollok (2)



A szavazó algoritmus három példája:

- Az írási és olvasási testület taglétszámának helyes megválasztása
- Lehetséges írás-írás ütközéshez vezető taglétszám
- A ROWA (read one, write all) néven ismert helyes taglétszámválasztás

Gyorsítótár-egyezőségi protokollok (1)

- Egyöntetűség-észlelő stratégia – mikor veszik észre a tár eltérő voltát
 - statikus
 - dinamikus
- Mikor történik az észlelés?
 - tranzakció előtt
 - tranzakció közben
 - tranzakció lezárásakor

Gyorsítótár-egyezőségi protokollok (2)

- Egyöntetűség-biztosítási stratégia – hogyan maradnak konzisztensek az adatok
 - megosztott adat csak a szerveren
 - változáskor érvényesítő üzenet
 - változáskor frissítés terjesztése
 - keresztülíró gyorsítótár
 - visszaíró gyorsítótár

Hibatűrés

1. rész

Alapelvek (1)

Üzembiztos rendszer

- Elérhetőség (availability)
- Megbízhatóság (reliability)
- Biztonságosság (safety)
- Kezelhetőség (maintainability)

Alapelvek (2)

- Meghibásodott rendszer
- Hiba
- Hibaok
- Hibatűrő képesség

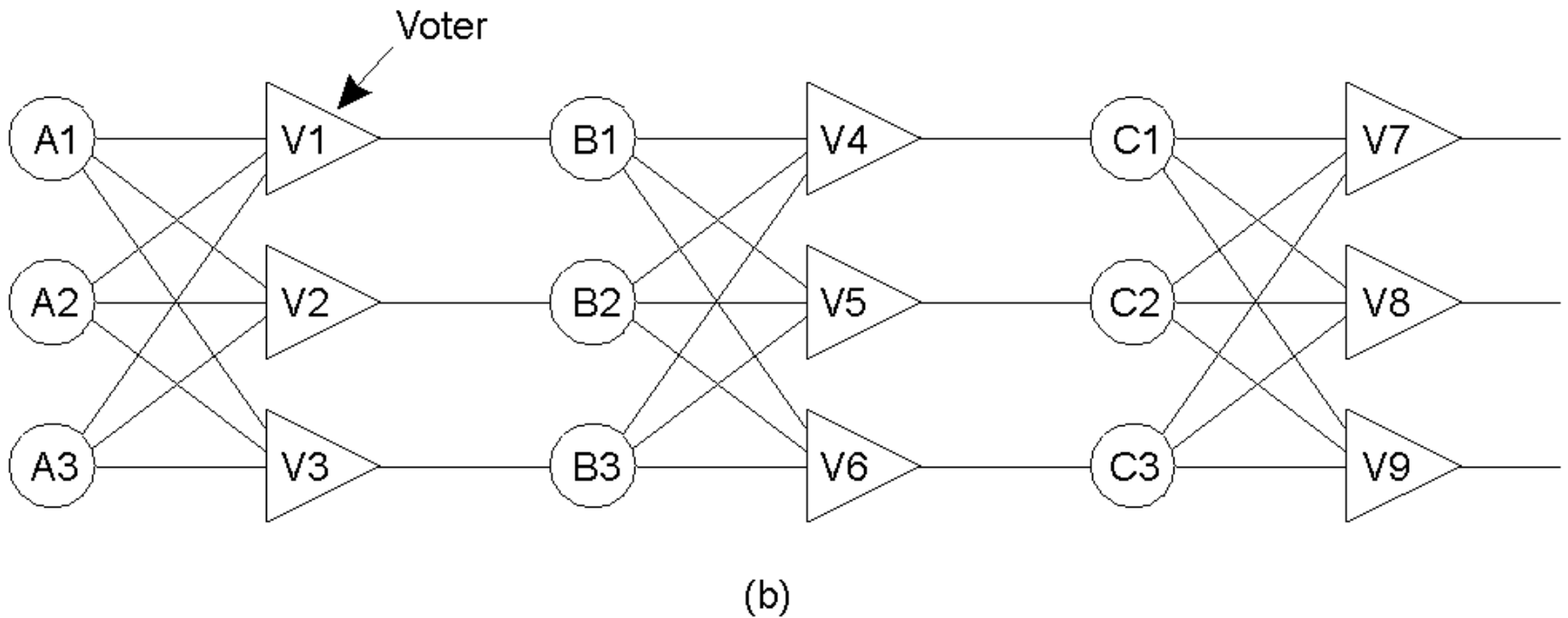
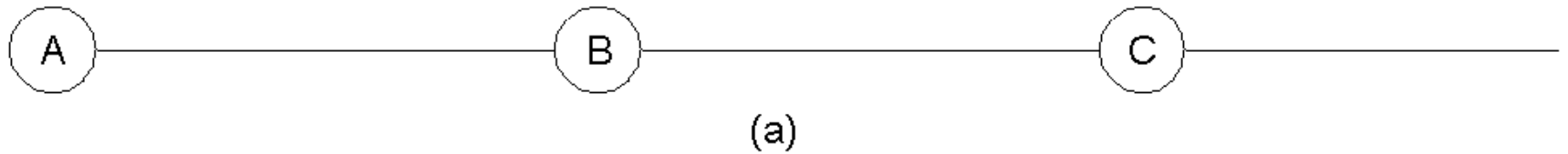
- Hibák csoportosítása
 - alkalmi hiba
 - ismétlődő hiba
 - állandó hiba

Meghibásodásmodellek

Meghibásodás típusa	Leírás
Összeomlás (crash)	A szerver megáll, de a leállítás pillanatáig helyesen működött
Kiesés (omission) <i>Fogadási (receive) kiesés</i> <i>Küldési (send) kiesés</i>	A szerver nem válaszol a beérkező üzenetekre A szerver nem kapja meg a beérkező üzeneteket A szerver nem tud üzenetet küldeni
Időzítési hiba (timing)	A szerver válaszküldésének időpontja kívül esik a meghatározott időintervallumon
Válaszhiba (response) <i>Értékhiba</i> <i>Állapotátmeneti hiba</i>	A szerver hibás választ küld A válaszüzenet értéke rossz A szerver eltér a helyes vezérlési sorrendtől
Váratlan hiba (arbitrary)	A szerver tetszőleges időpontban tetszőleges válaszokat tud előállítani

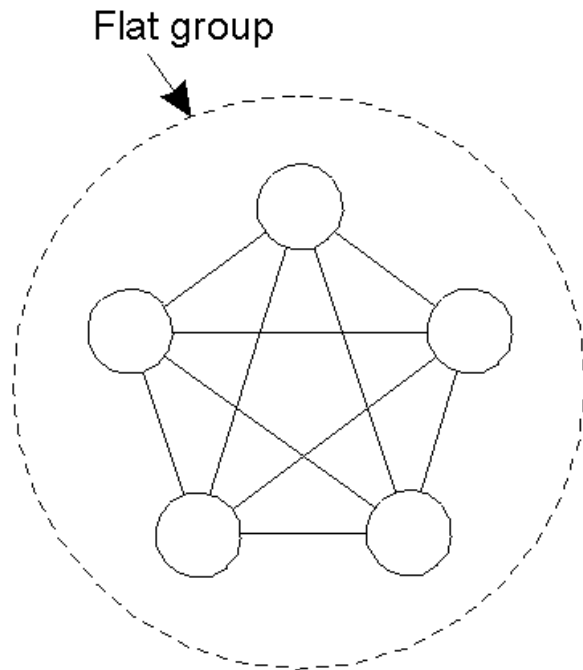
A meghibásodások különböző típusai.

Meghibásodás elfedése redundanciával

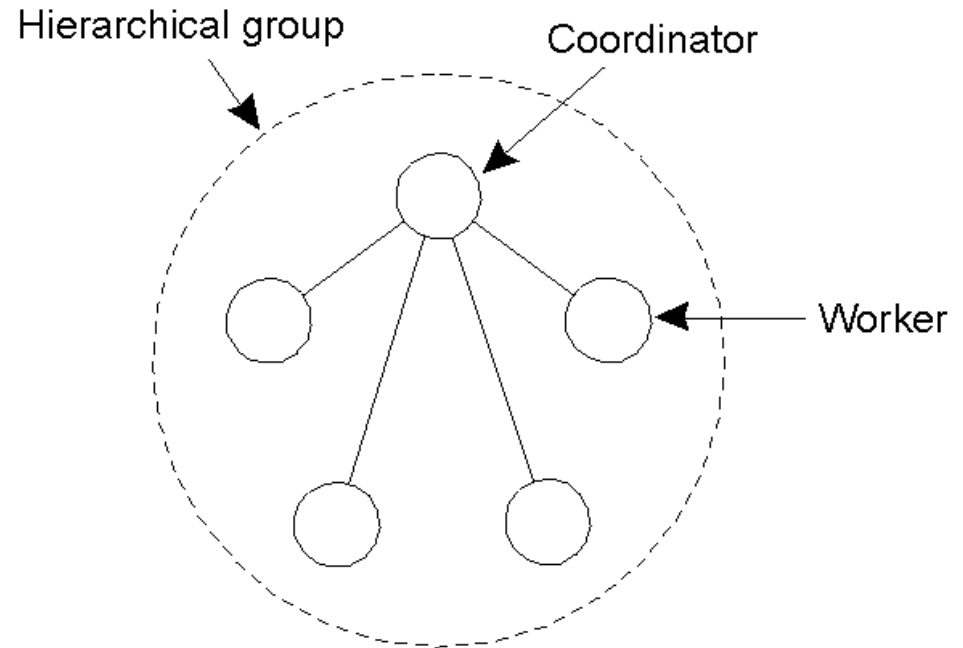


Hármas moduláris redundancia.

Egyenlő és hierarchikus csoportok



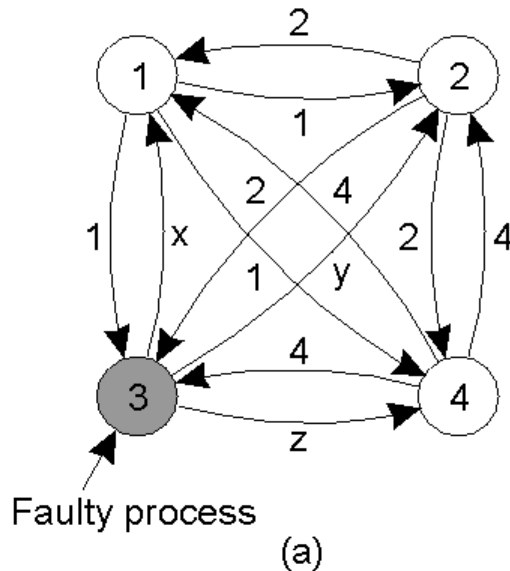
(a)



(b)

- a) Kommunikáció az egyenlő csoportban
- b) Kommunikáció az egyszerű hierarchikus csoportban

Közös megegyezés a hibás rendszerekben (1)



1 Got(1, 2, x, 4)
 2 Got(1, 2, y, 4)
 3 Got(1, 2, 3, 4)
 4 Got(1, 2, z, 4)

(b)

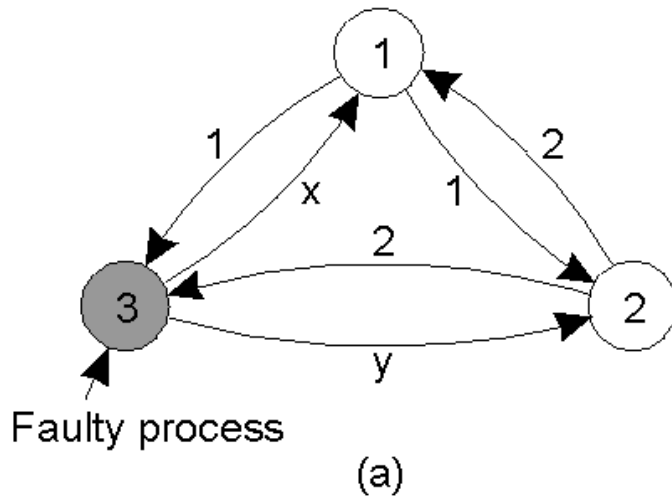
<u>1 Got</u>	<u>2 Got</u>	<u>4 Got</u>
(1, 2, y, 4)	(1, 2, x, 4)	(1, 2, x, 4)
(a, b, c, d)	(e, f, g, h)	(1, 2, y, 4)
(1, 2, z, 4)	(1, 2, z, 4)	(i, j, k, l)

(c)

A bizánci tábornok problémája három lojális és egy áruló tábornok esetén.

- a) A tábornokok jelentik a saját haderejük nagyságát
- b) A négy tábornok által összegyűjtött információ-vektorok
- c) A harmadik lépésben megkapott vektorok

Közös megegyezés a hibás rendszerekben (2)



1 Got(1, 2, x)
 2 Got(1, 2, y)
 3 Got(1, 2, 3)

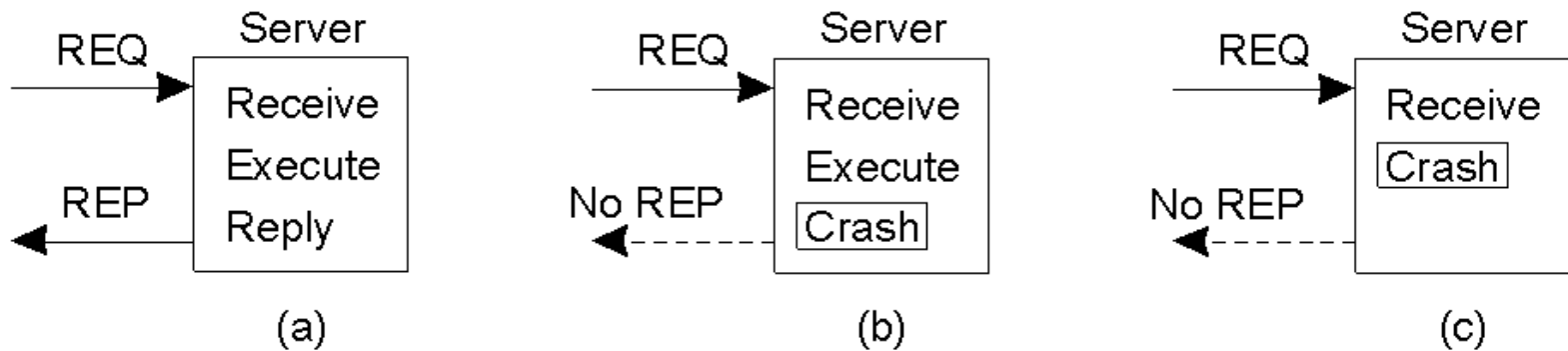
(b)

1 Got	2 Got
(1, 2, y)	(1, 2, x)
(a, b, c)	(d, e, f)

(c)

Ugyanaz, mint az előző ábra, de ezúttal két lojális és egy áruló tábornok esetére.

A szerver összeomlása (1)



A szerver működése a kliens-szerver kommunikációban

- a) Alapeset
- b) Összeomlás a végrehajtás után
- c) Összeomlás a végrehajtás előtt

A szerver összeomlása (2)

Kliens	Szerver					
	M -> P stratégia		P -> M stratégia			
Újraküldési stratégia	MPC	MC(P)	C(MP)	PMC	PC(M)	C(PM)
Mindig	DUP	OK	OK	DUP	DUP	OK
Soha	OK	ZERO	ZERO	OK	OK	ZERO
Csak ha jött ACK	DUP	OK	ZERO	DUP	OK	ZERO
Csak ha nem jött ACK	OK	ZERO	OK	OK	DUP	OK

A szerver és a kliens stratégiák különböző kombinációi
szerverhiba esetén