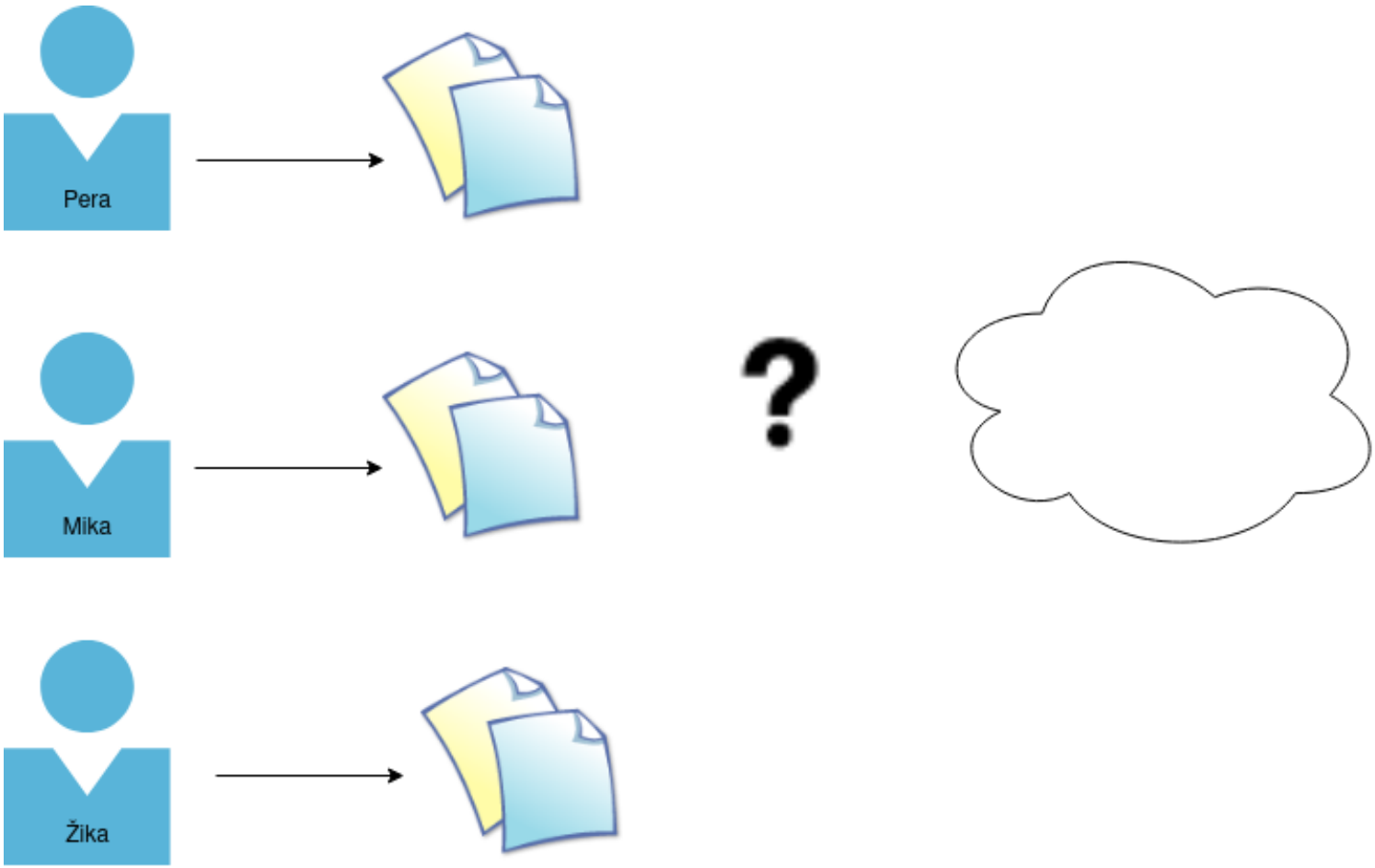
CRDT

SOA+NoSQL

Fakultet tehničkih nauka Novi Sad

26. decembar 2023.

Problem



Slika 1: Više korisnika edituje isti dokument

Problem

- Šta ako Pera i Mika kucaju istovremeno?
- Šta ako je Žiki loša konekcija? Da li se njegove promene mogu spojiti po povratku konekcije?
- Da li se svaka izmena mora sinhronizovati sa serverom?
- Da li mogu promene da se čuvaju do nekog momenta samo lokalno, pa da se onda spoje sa ostalim izmenama?

Rešenje

- CRDT

Conflict-free Replicated Data Type (CRDT)

- Struktura podataka koja pojednostavljuje distribuirane sisteme za skladištenje podataka i aplikacije za više korisnika.
- Bez obzira na modifikacije podataka na različitim replikama, CRDT omogućavaju da podaci uvek mogu biti spojeni u konzistentno stanje.

CRDT - osobine

- Aplikacija može da ažurira svaku repliku u bilo kom trenutku, konkurentno i bez sinhronizacije sa drugim replikama.
- Algoritam, koji se sam nalazi unutar strukture, razrešava konflikte koji se mogu pojaviti.
- Iako replike mogu imati različito trenutno stanje, garantuju da će eventualno iskonvergirati u konzistentno stanje.

Vrste

- **State-based**: replike propagiraju promene slanjem punog stanja objekata
- **Operation-based**: replika propagira promene tako što šalje operaciju svim replikama
- **Delta-based:** šalje se puno stanje u slušaju prve sinhronizacije, a nadalje se gleda vreme zadnje sinhronizacije i prosleđuju se promene stanja

State-based CRDT kao struktura

- Stanje S - moramo imati neku relaciju koja nam daje bar parcijalni poredak
- Funkcija Merge koja nam služi da spojimo stanje neke druge replike sa stanjem na trenutnoj replici
- Funkcija Update koja menja vrednost na replici

Kako merge funkcija rešava probleme sa početka?

- Da li nam je bitan redosled u kom se dešavaju izmene?

Kako merge funkcija rešava probleme sa početka?

- Da li nam je bitan redosled u kom se dešavaju izmene?
- Šta ako neka izmena kasni?

Kako merge funkcija rešava probleme sa početka?

- Da li nam je bitan redosled u kom se dešavaju izmene?
- Šta ako neka izmena kasni?
  - Komutativnost
  - Asocijativnost

Kako merge funkcija rešava probleme sa početka?

- Da li nam je bitan redosled u kom se dešavaju izmene?
- Šta ako neka izmena kasni?
  - Komutativnost
  - Asocijativnost
- Šta ako više puta stigne ista izmena?

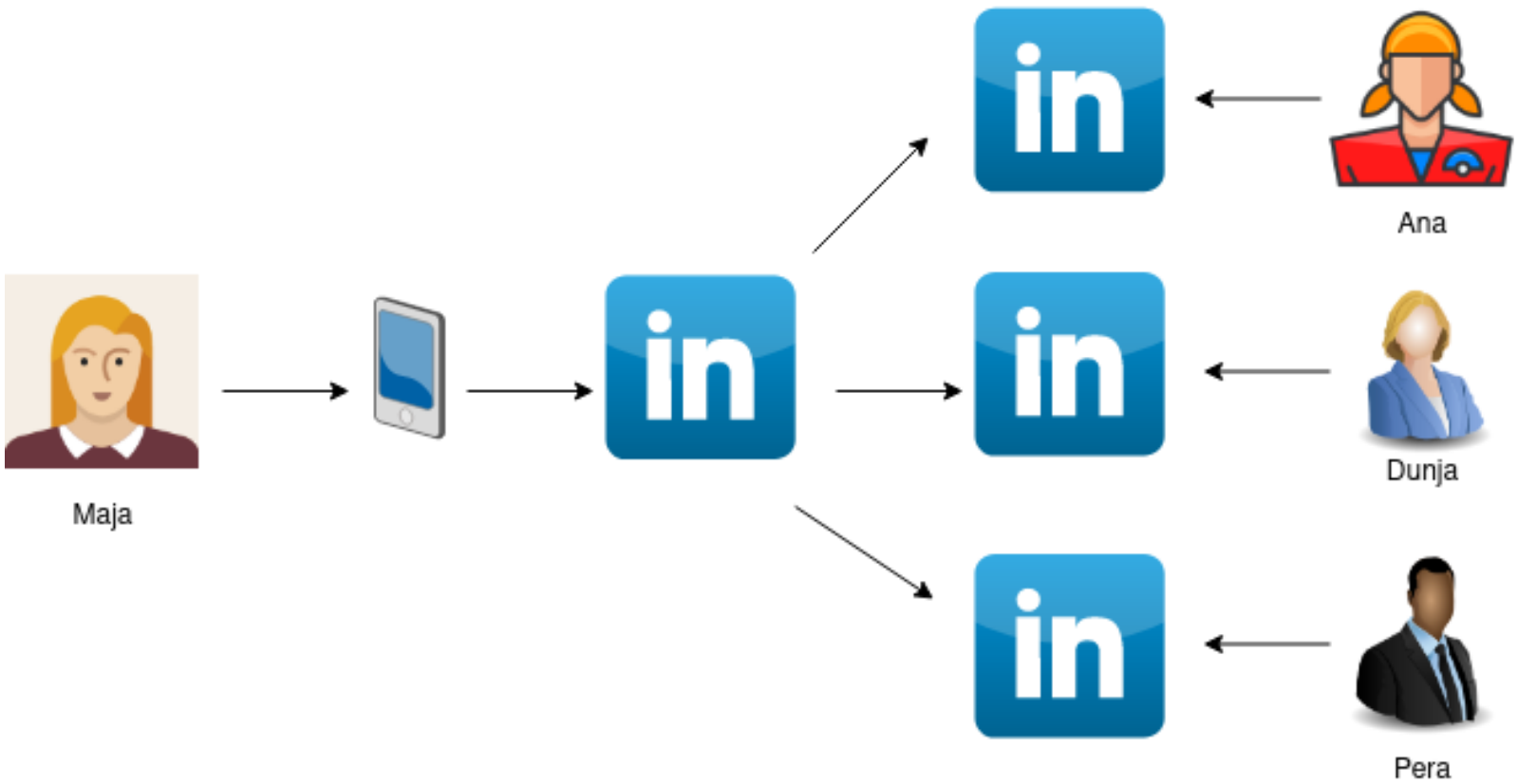
Kako merge funkcija rešava probleme sa početka?

- Da li nam je bitan redosled u kom se dešavaju izmene?
- Šta ako neka izmena kasni?
  - Komutativnost
  - Asocijativnost
- Šta ako više puta stigne ista izmena?
  - Idempotentnost

Kako merge funkcija rešava probleme sa početka?

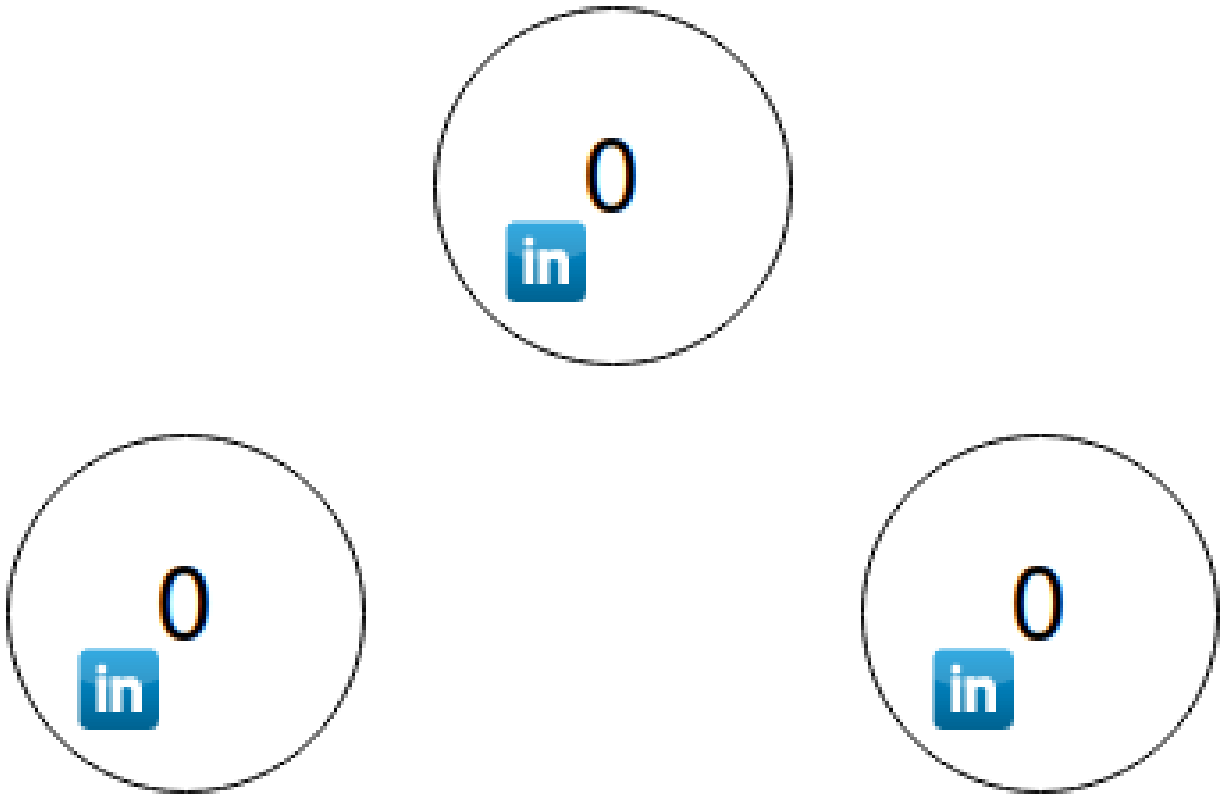
- Da li nam je bitan redosled u kom se dešavaju izmene?
- Šta ako neka izmena kasni?
  - Komutativnost
  - Asocijativnost
- Šta ako više puta stigne ista izmena?
  - Idempotentnost
- Ove tri osobine nam garantuju da ćemo uvek moći spojiti stanja na replikama u konzistentno stanje, pa ne moramo da brinemo o protokolu za prenos.
- Drugim rečima, omogućava nam da asinhrono delimo lokalna stanja replika.

Problem 2



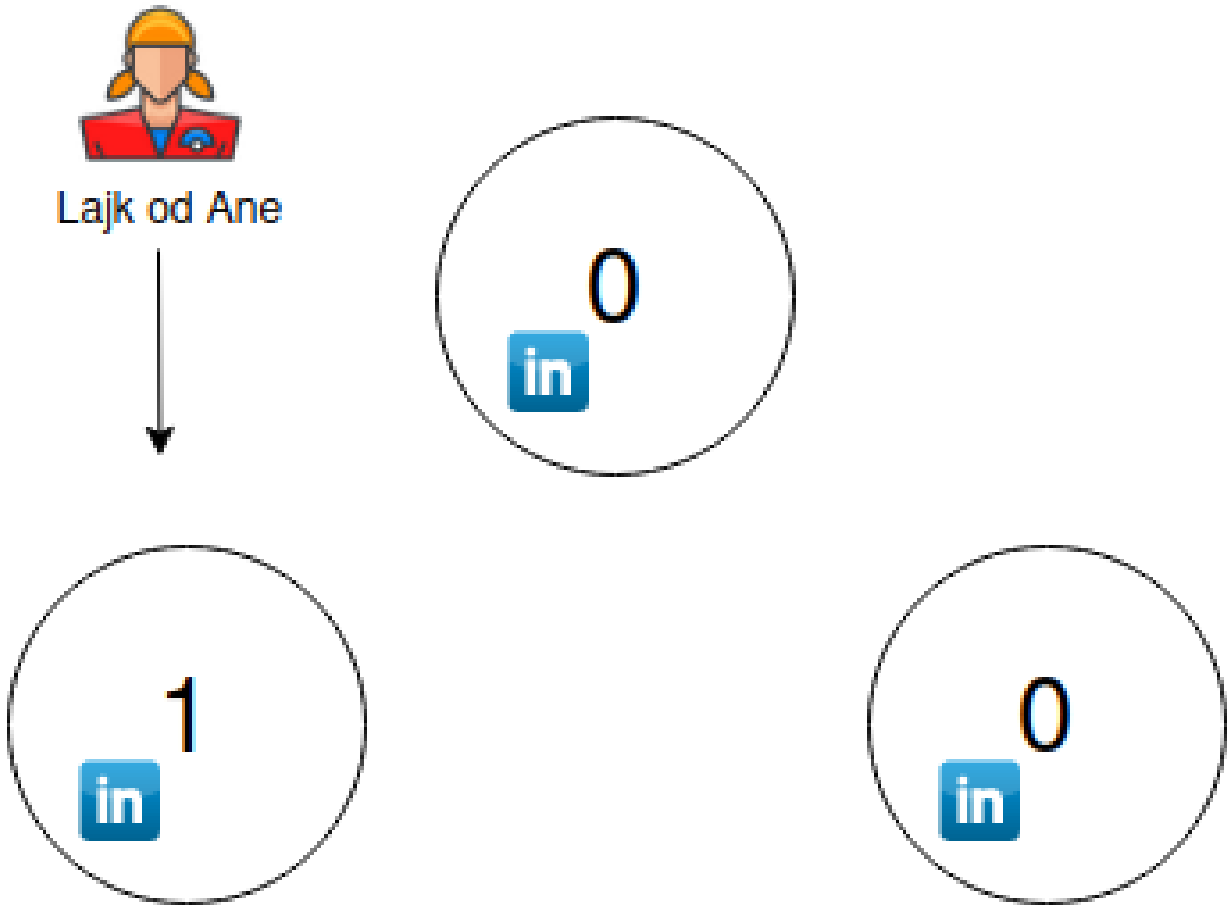
Slika 2: Lajkovi na objavi

Replike - distribuirani brojač



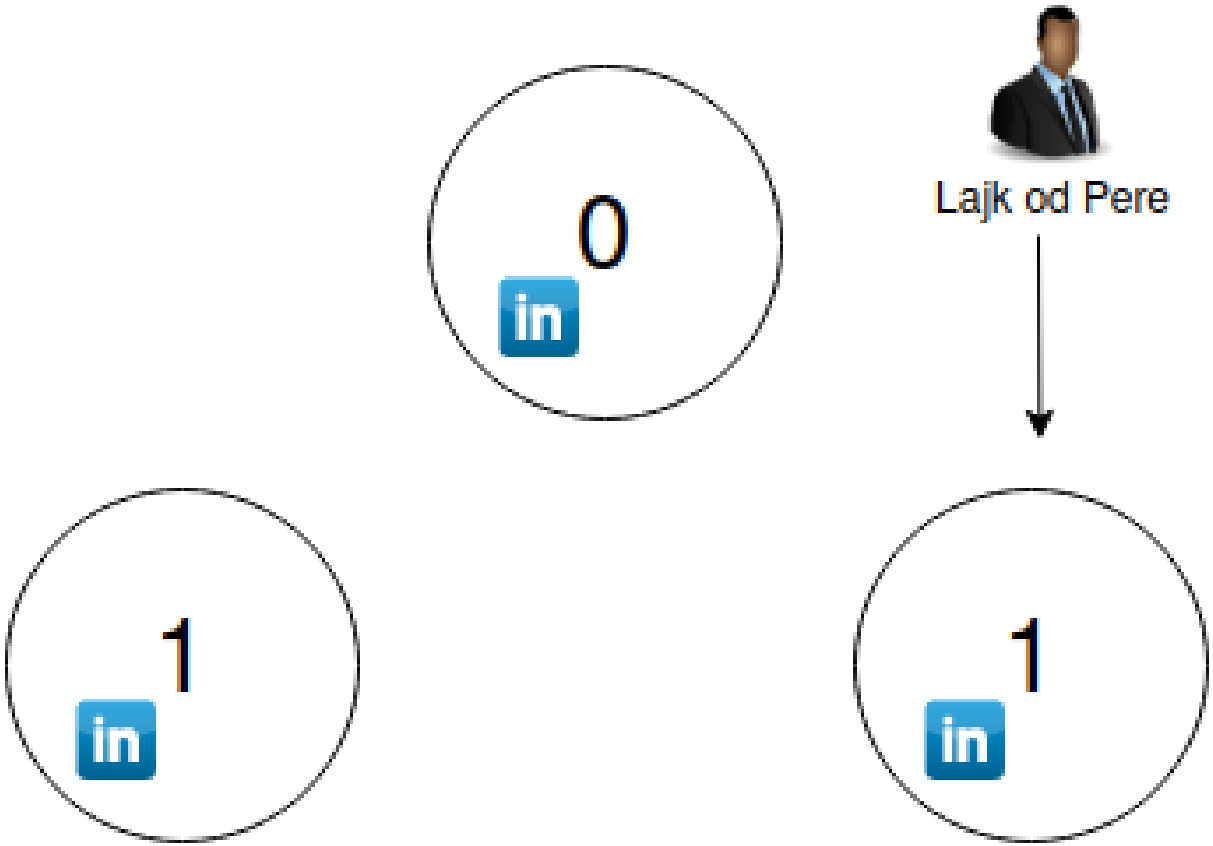
Slika 3: Početno stanje na replikama

Replike - distribuirani brojač



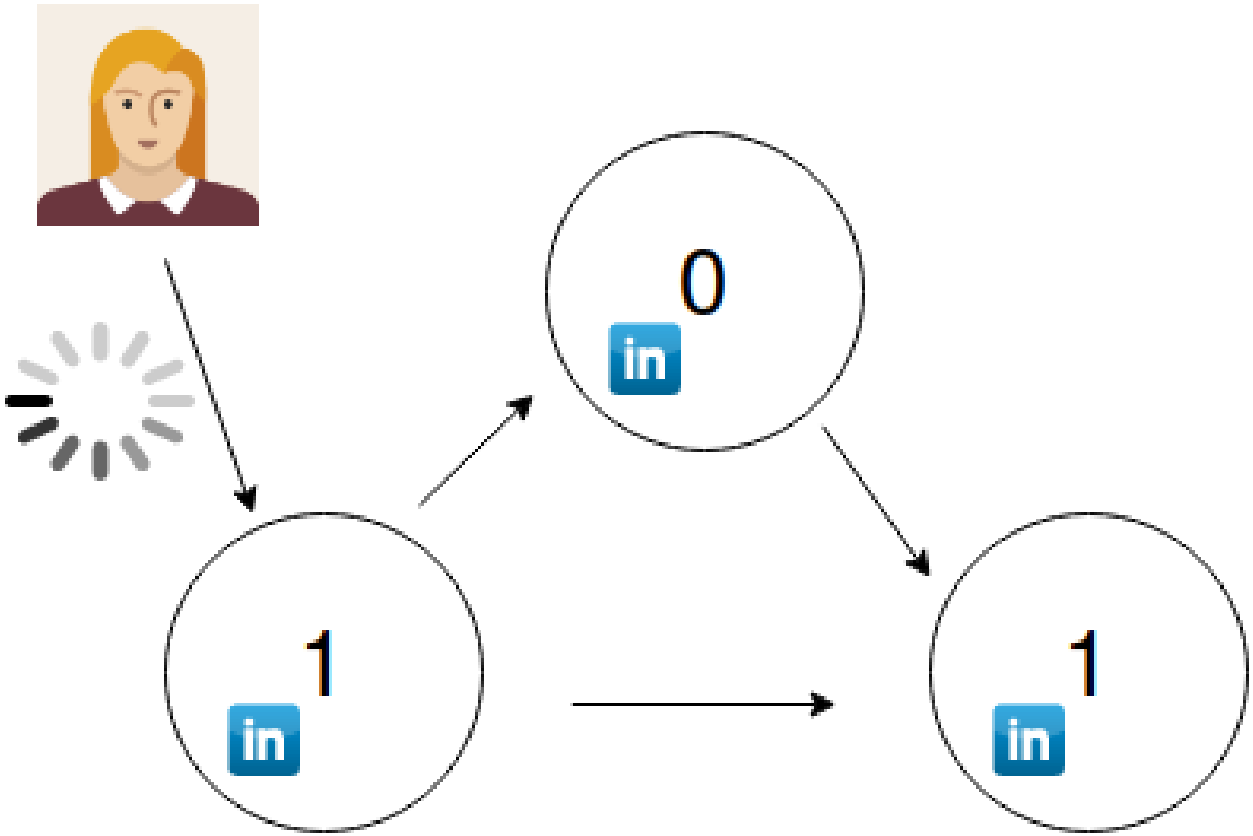
Slika 4: Lajk od Ane

Replike - distribuirani brojač



Slika 5: Lajk od Pere

Problem 2 - Maja hoće da zna broj lajkova



Slika 6: Zahtev za broj lajkova

Problem 2

- Ako prilikom svakog zahteva moramo da imamo prvo sinhronizaciju između replika, korisnik je dugo ćekati na odgovor.

Problem 2

- Ako prilikom svakog zahteva moramo da imamo prvo sinhronizaciju između replika, korisnik je dugo ćekati na odgovor.
- Bolje da vratimo korisniku brz odgovor, bez obzira što možda nije u potpunosti tačan.

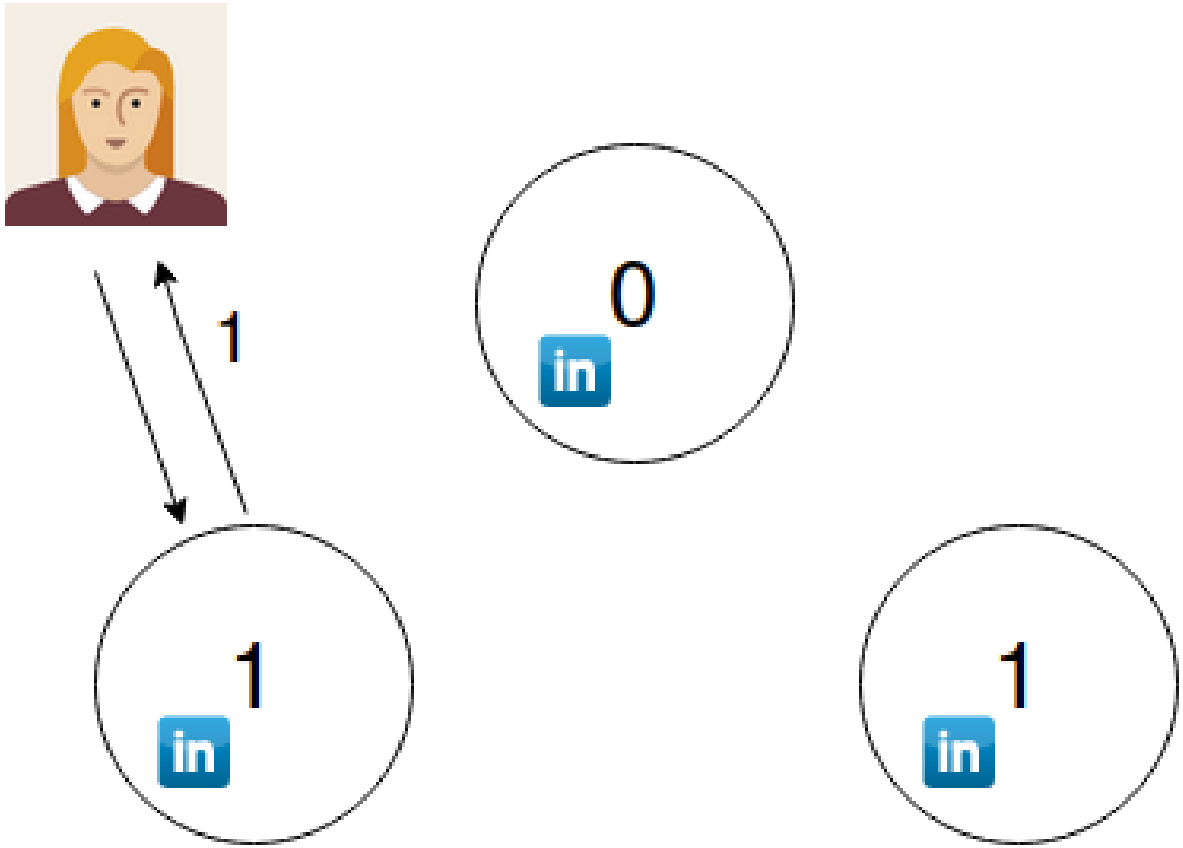
Problem 2

- Ako prilikom svakog zahteva moramo da imamo prvo sinhronizaciju između replika, korisnik je dugo ćekati na odgovor.
- Bolje da vratimo korisniku brz odgovor, bez obzira što možda nije u potpunosti tačan.
- Sinhronizacija replika se radi dok čvorovi nisu opterećeni (npr. gossip)
- S vremena na vreme će se replike sinhronizovati, a kad stigne zahtev, replika vraća vrednost koja se kod nje nalazi - eventualno konzistentno.

Rešenje problema2

- Postavimo CRDT na svaku repliku.
- On predstavlja šta replika misli da je stvaran ukupan broj lajkova.
- Stvarna ukupna vrednost ne mora da postoji ni na jednoj replici u sistemu.
- To je ok, dok god promene koje korisnik vidi imaju smisla.

Rešenje problema2

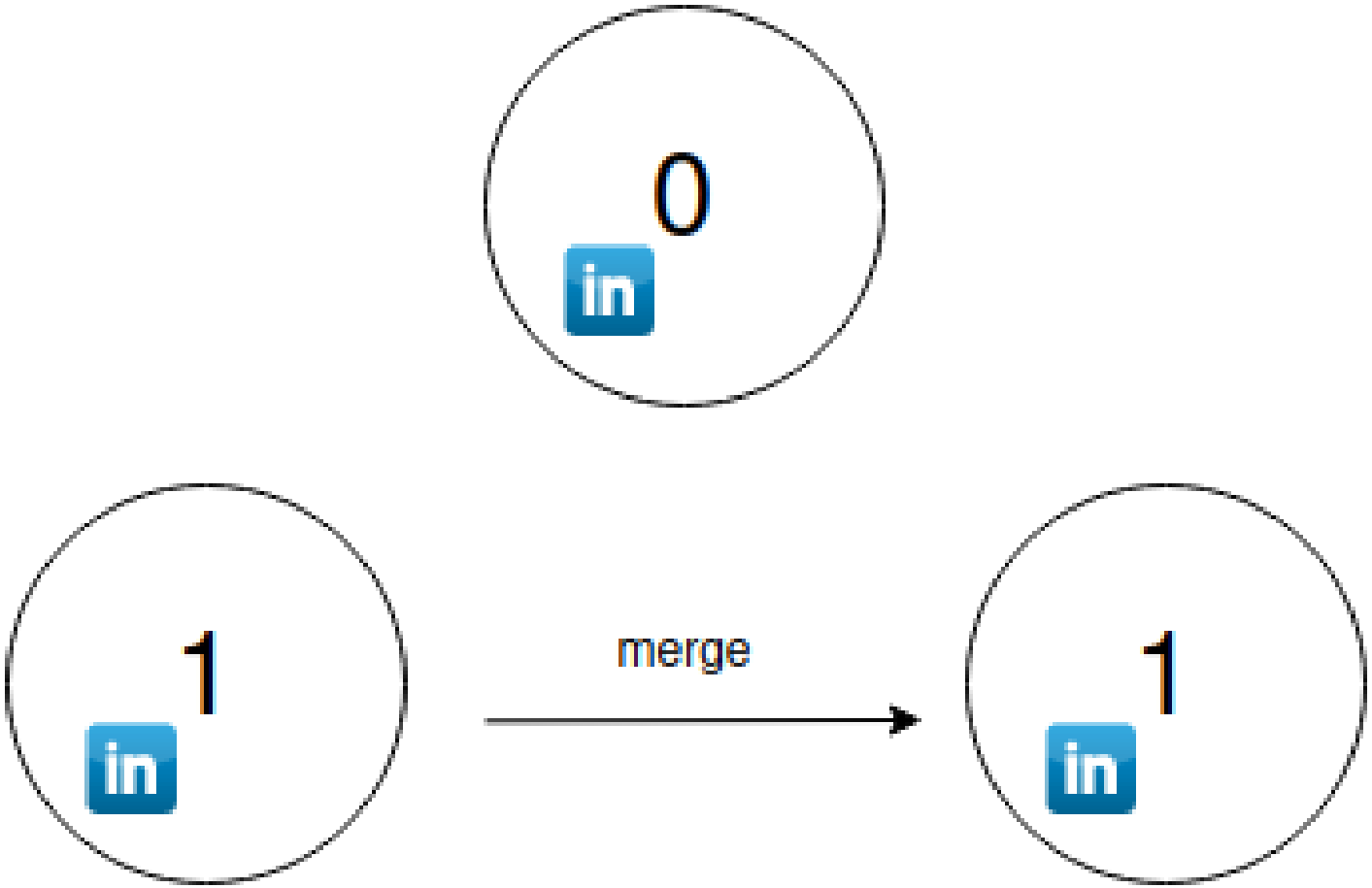


Slika 7: Maja vidi odgovor 1, stvaran broj lajkova 2

Sinhronizacija replika - G Counter

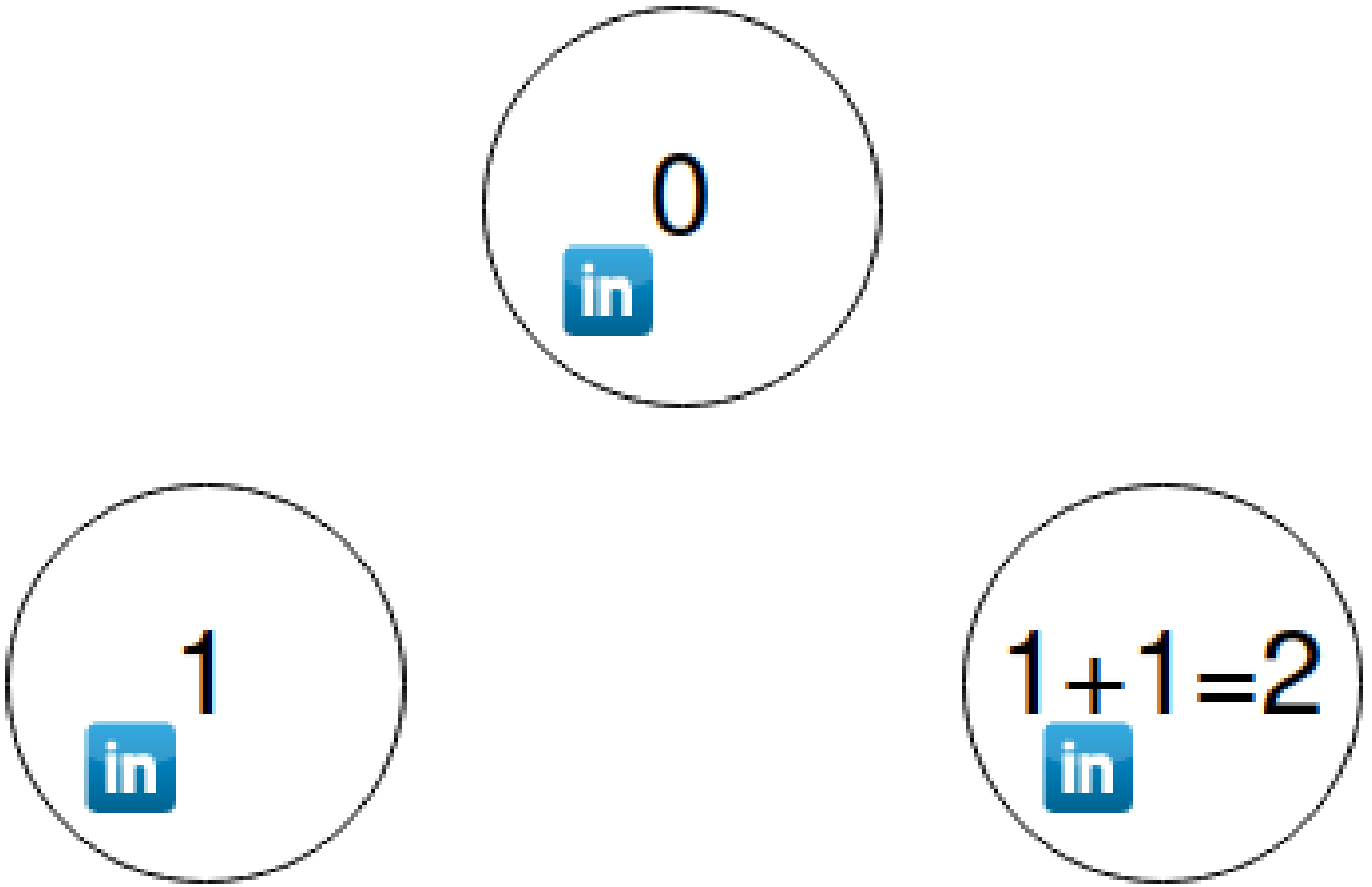
- Grow - only counter
- Broj lajkova se može samo povećati (pretpostavka da korisnik ne može da poništi lajk)
- Kako spojiti vrednosti na replikama?

G Counter - naivan pristup



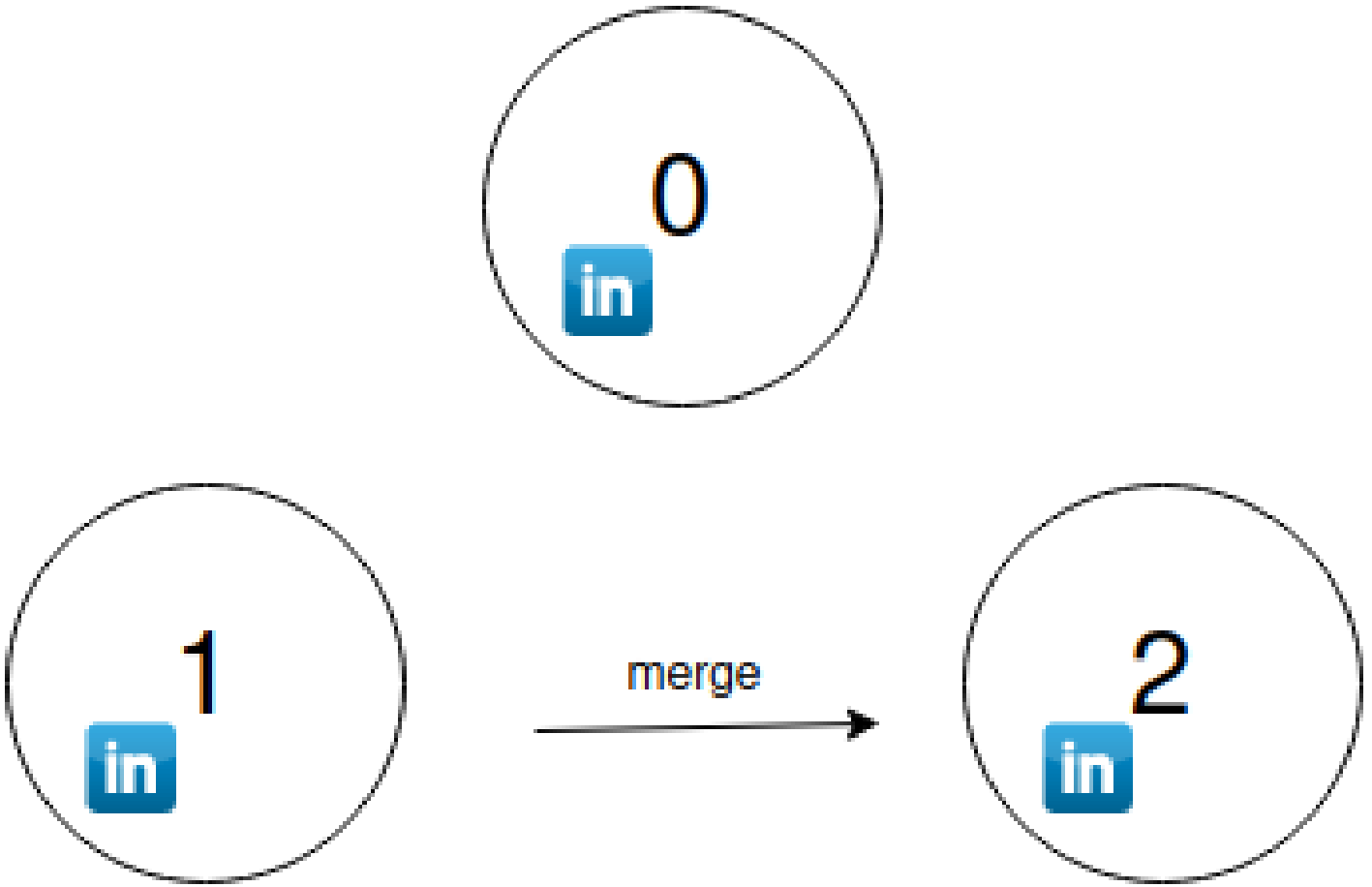
Slika 8: Replika 1 šalje svoje stanje replici 3

G Counter - naivan pristup



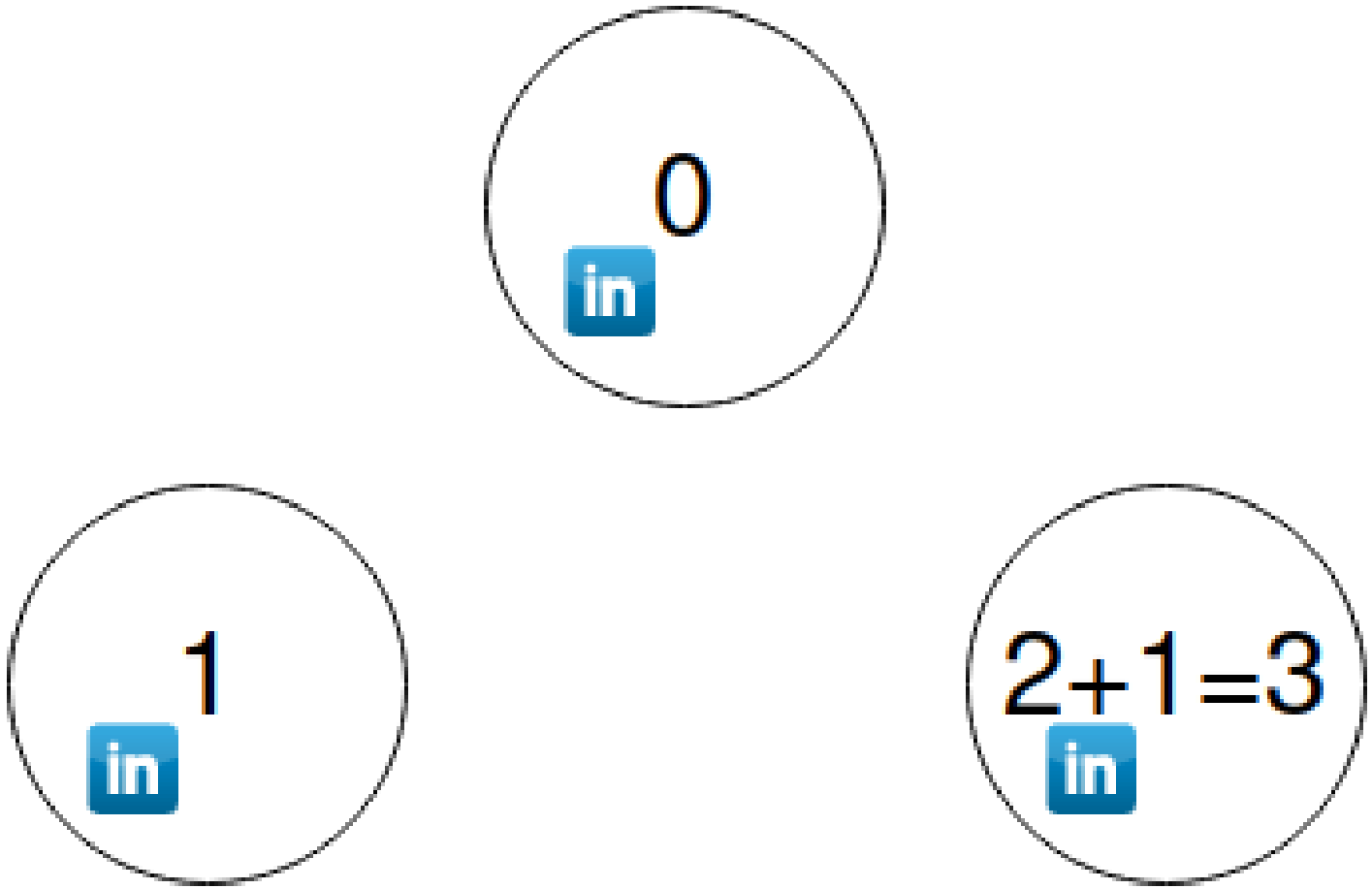
Slika 9: Replika 3 ažurira svoje stanje

G Counter - naivan pristup



Slika 10: Replika 1 ponovo šalje svoje stanje

G Counter - naivan pristup



Slika 11: Pogrešan rezultat

Šta je problem?

- Sabiranje nije idempotentno.

Šta je problem?

- Sabiranje nije idempotentno.
- Da li je max idempotentno?

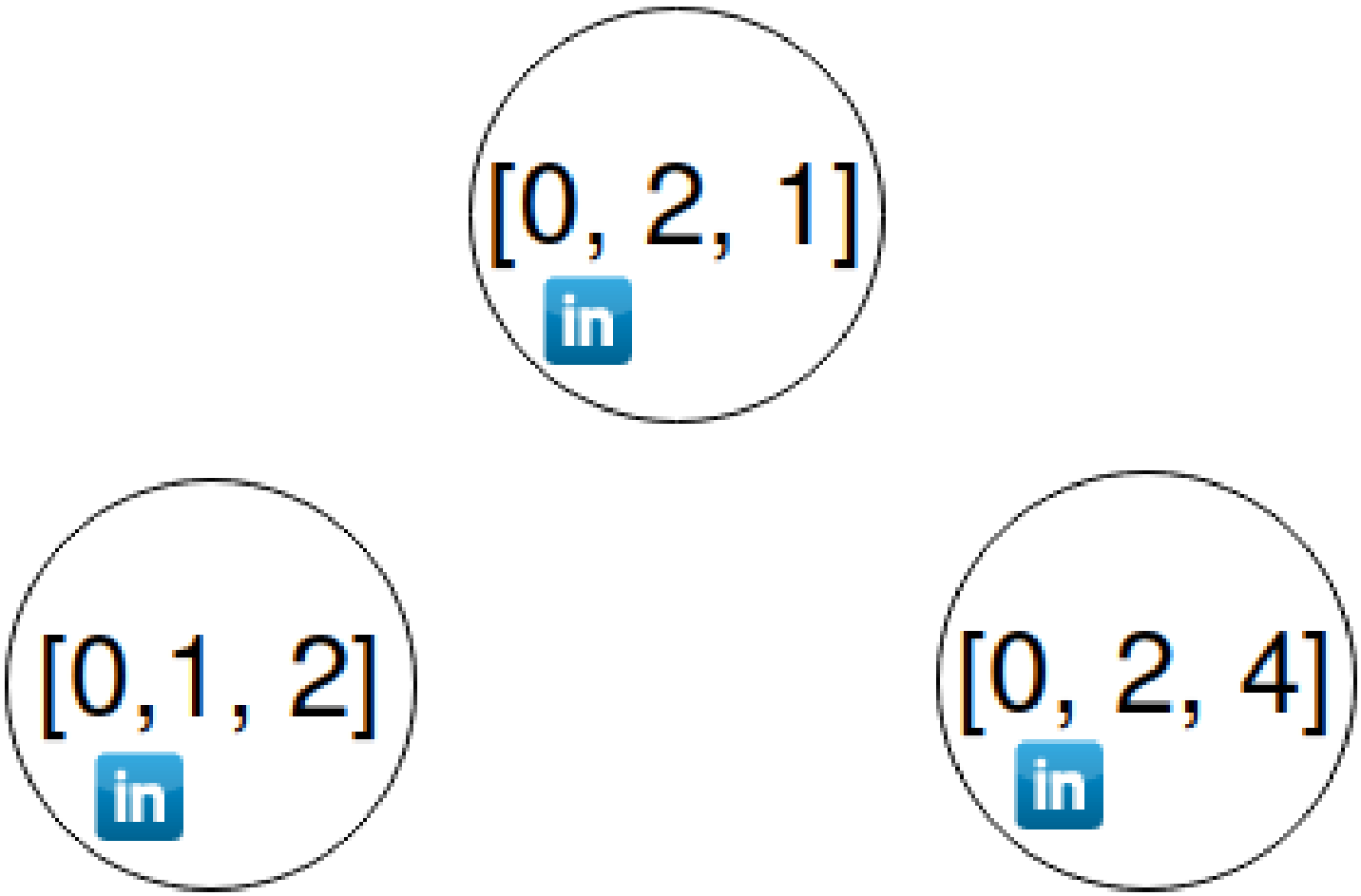
Šta je problem?

- Sabiranje nije idempotentno.
- Da li je max idempotentno?
- Da li možemo da iskoristimo max?

G Counter - dobro rešenje

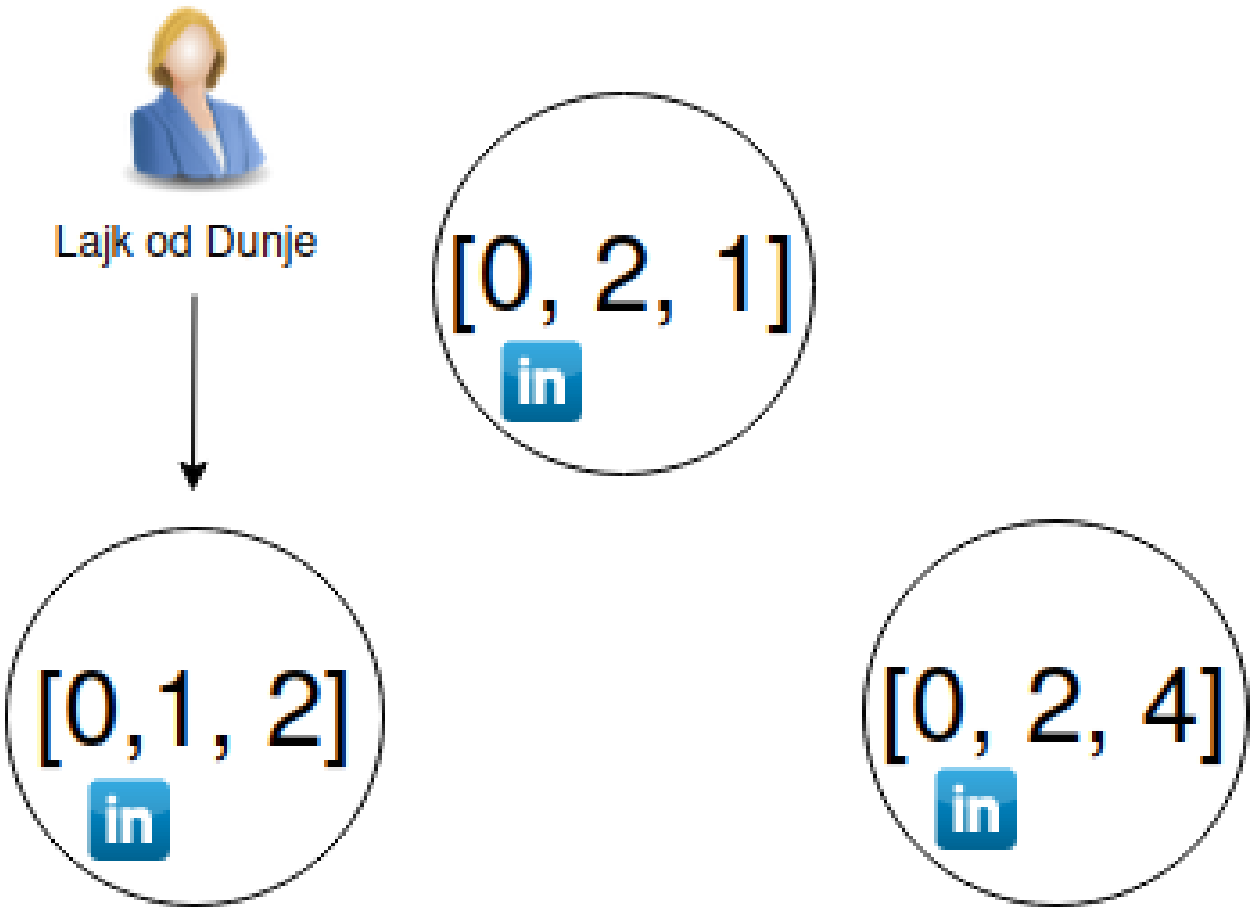
- Svaka replika će čuvati kao stanje informaciju o brojaču na svim replikama.
- Predstavljamo ga vektorom - npr. [1, 0, 3]
- Kad na repliku stigne zahtev za lajk, replika uvećava samo element u vektoru koji predstavlja njen brojač.
- Merge radi tako što traži max elemenata na istim indeksima.

G Counter - dobro rešenje



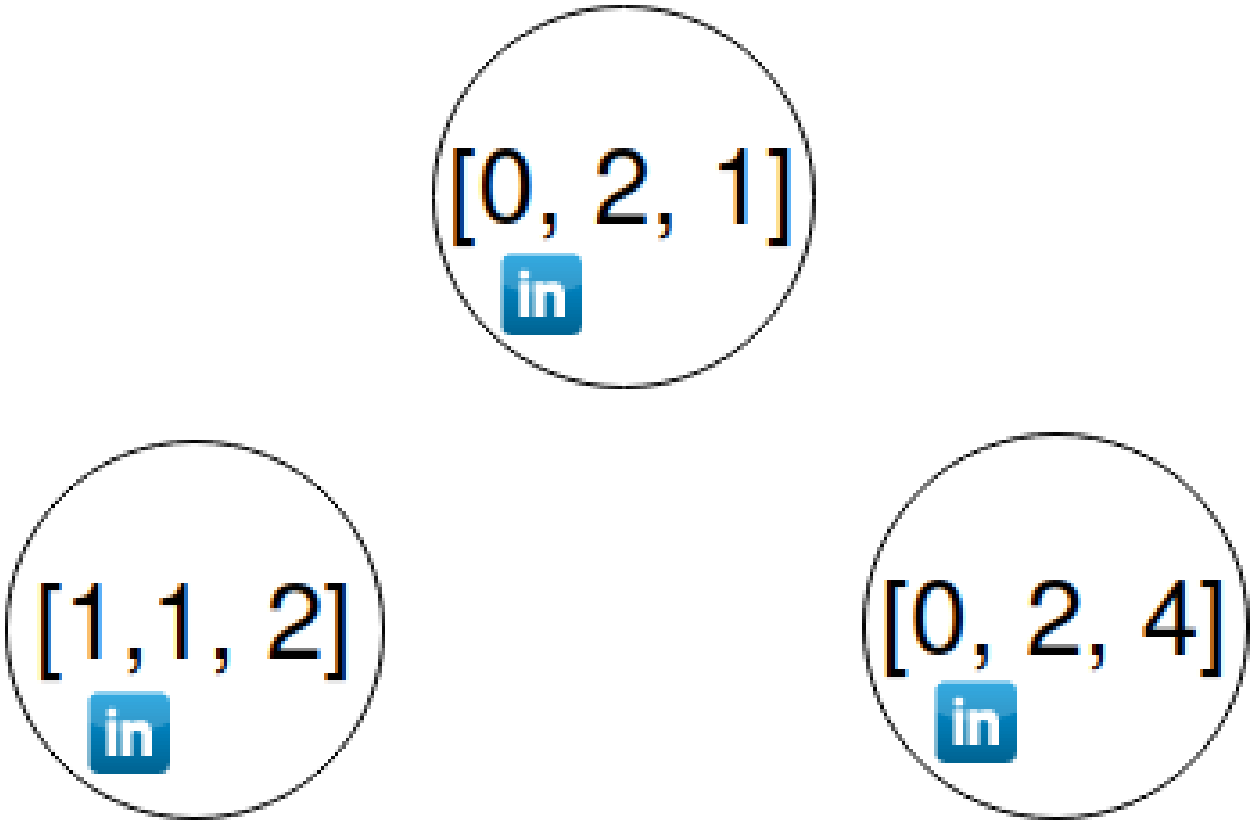
Slika 12: Trenutno stanje

G Counter - dobro rešenje



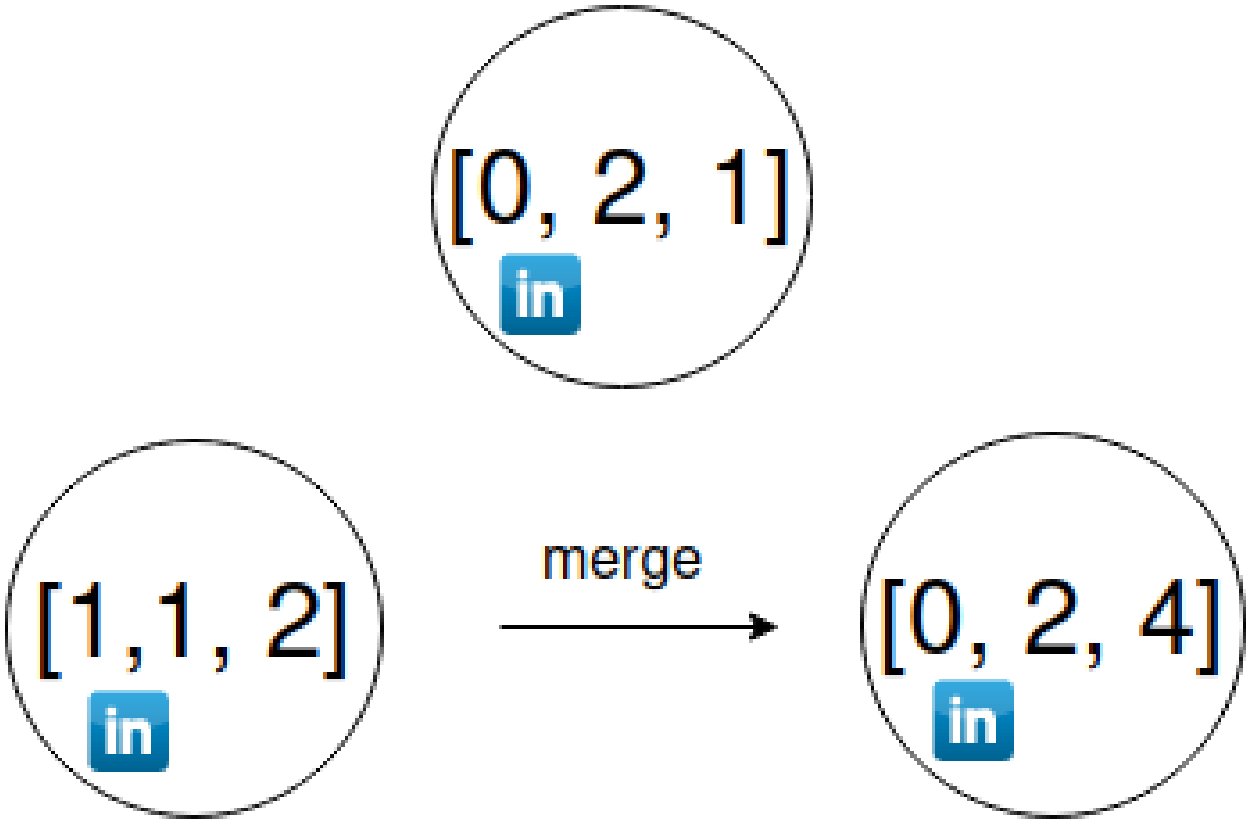
Slika 13: Novi lajk na replici 1

G Counter - dobro rešenje



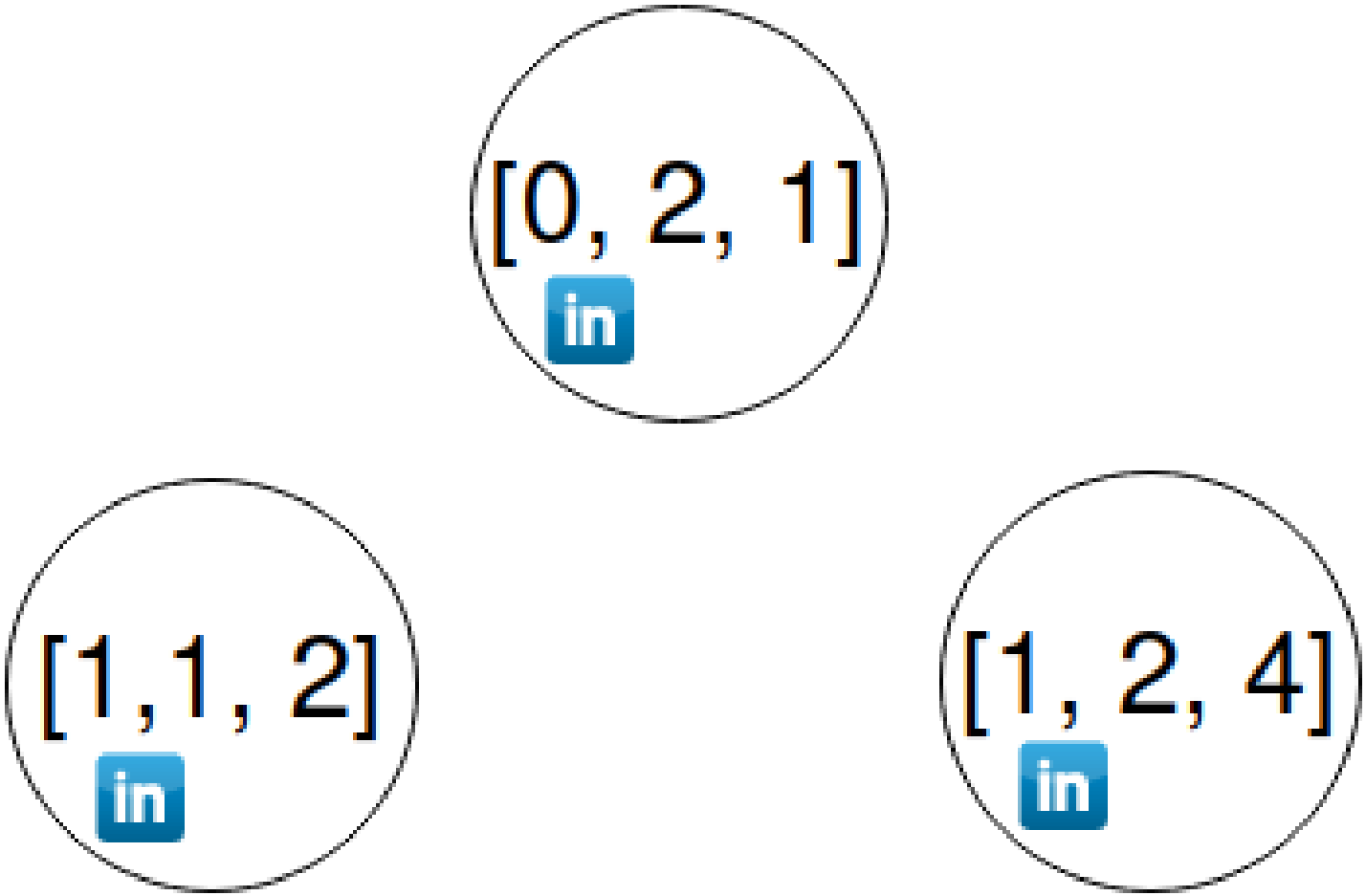
Slika 14: Novo trenutno stanje

G Counter - dobro rešenje



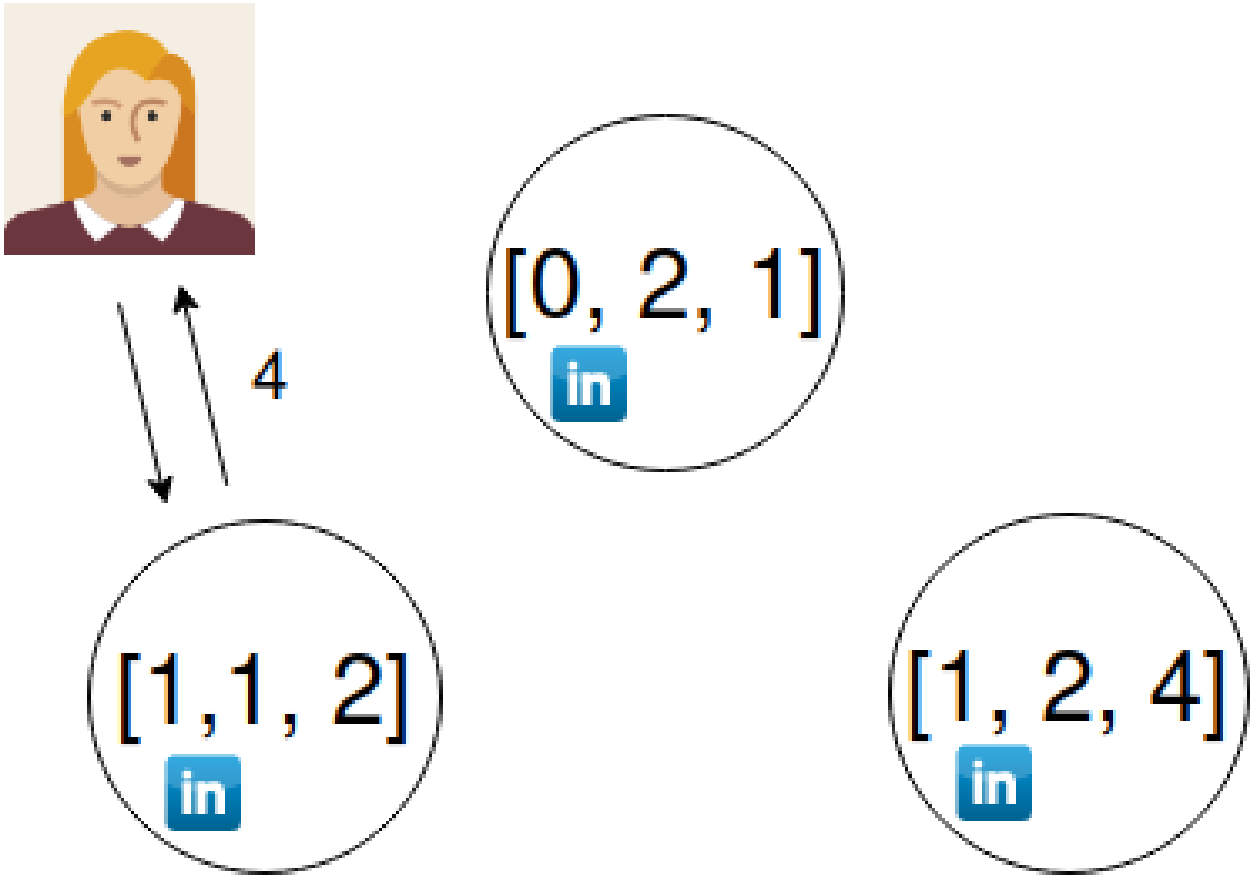
Slika 15: Replika 1 šalje svoje stanje replici 3

G Counter - dobro rešenje



Slika 16: Novo trenutno stanje

G Counter - dobro rešenje



Slika 17: Maja dobija kao odgovor sumu brojeva u vektoru

Problem 3

- Šta ako neko može da ostavi "negativan" lajk?

Problem 3

- Šta ako neko može da ostavi "negativan" lajk?
- PN Counter

Problem 3

- Šta ako neko može da ostavi "negativan" lajk?
- PN Counter
- Koristi dva G Countera, jedan kao brojač inkrementa, a drugi dekrementa.

Primena CRDT

- Koriste se za deljene dokumente, onjaln četove..
- Koriste ga najčešće distribuirane NoSQL baze:
  - Redis
  - Riak
  - Cosmos
- Cassandra koritsti LWW (Last Write Wins) Element Set CRDT

Korisni linkovi

- https://github.com/neurodrone/crdt
- https://www.youtube.com/watch?v=OOlnp2bZVRs
- https://crdt.tech/resources
- https://github.com/alangibson/awesome-crdt
- https://mattweidner.com/2023/09/26/crdt-survey-2.html
- https://redis.com/blog/diving-into-crdts/