HotStuff sa Adaptivnim *Pacemaker*-om

Boris Čuljak

Fakultet Tehničkih Nauka (FTN)
Primenjeni Algoritmi u Upravljačkim Sistemima

GitHub: `boriscu/Adaptive-Pacemaker-HotStuff`

25. decembar 2025.

Sadržaj

Zašto uopšte konsenzus?

Problem

U distribuiranom servisu replike mogu primati zahteve različitim redosledom zbog kašnjenja i nepredvidive isporuke poruka, pa stanje "propada".

Cilj

Obezbediti da sve ispravne replike izvršavaju **iste komande istim redosledom**.

Rešenje

Konsenzus protokol dogovara jedan globalni redosled (blokove/komande).

Doprinos (šta je urađeno)

- Objašnjen HotStuff u delimično sinhronom modelu (bezbednost + živost).
- Implementiran HotStuff u podesivom diskretno-događajnom simulatoru.
- Uveden adaptivni *pacemaker*: menja samo politiku *timeout*-a (ne menja pravila glasanja/zaključavanja).
- Evaluacija u više scenarija: skaliranje, greške, gubitak poruka, prekoračenje BFT praga.

Pojmovi koje koristimo (jednom i jasno)

SMR (*State Machine Replication* - replikacija mašine stanja)

Deterministički servis + isti redosled komandi $\Rightarrow$ isto stanje na svim replikama.

BFT (*Byzantine Fault Tolerant* - vizantijska tolerancija)

Neispravni čvor može lagati i ponašati se proizvoljno; cilj su **bezbednost** i **živost**.

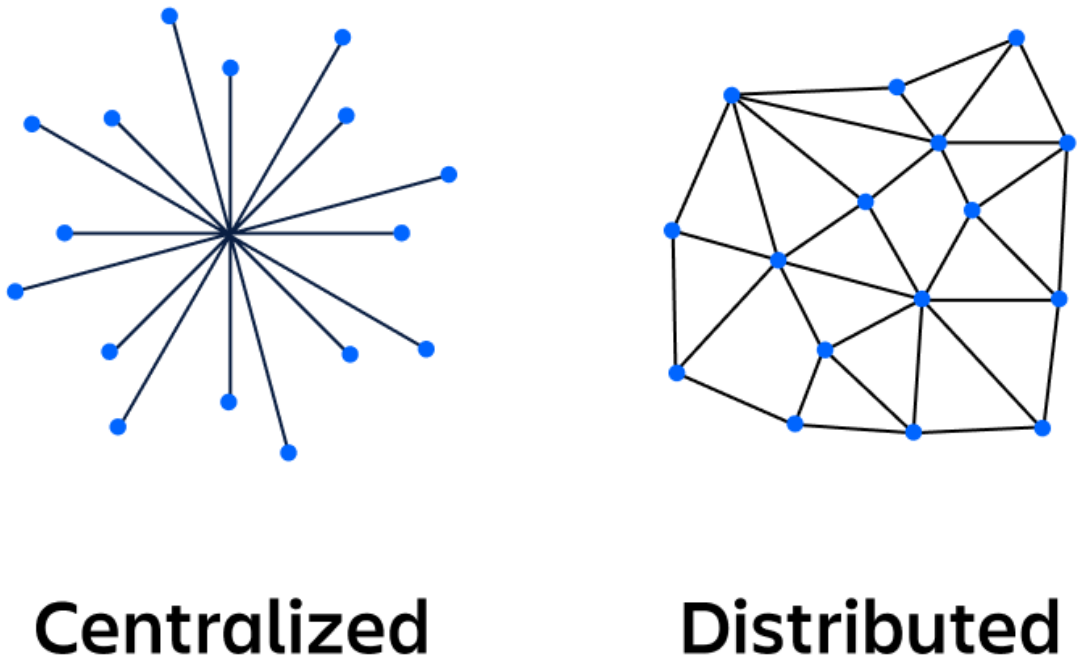
GST (*Global Stabilization Time* - trenutak globalne stabilizacije)

Posle GST poruke između ispravnih replika stižu u ograničenom vremenu Δ (delimična sinhronost).

Mandat i promena mandata

Mandat = period sa jednim vođom, **promena mandata** = prelazak na novog vođu zbog izostanka napretka.

Intuicija distribuiranog sistema



Slika 1: Centralizovano vs distribuirano: koordinacija postaje teža bez jedinstvene tačke kontrole.

Kvorumi i uslov $n \geq 3f + 1$

Standardni BFT uslov

Da bismo tolerisali do f vizantijskih replika, tipično je potrebno $n \geq 3f + 1$.

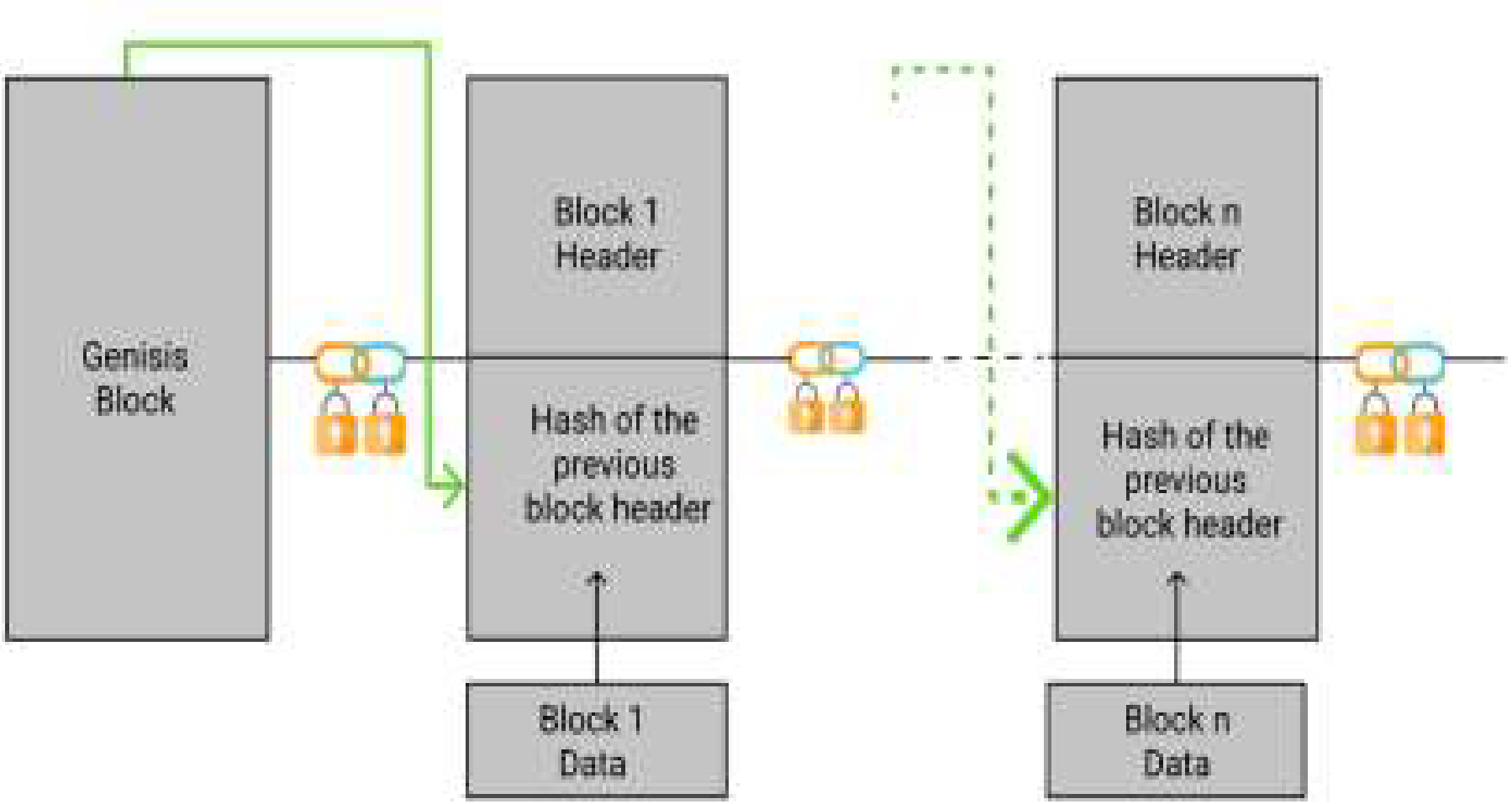
Kvorum

Protokol koristi kvorum od $2f + 1$ glasova. Bilo koja dva takva kvoruma seku se u najmanje $f + 1$ replika $\Rightarrow$ bar jedna ispravna replika je u preseku.

Šta to daje?

Sprečava da dve konfliktne istorije obe dobiju „dovoljno“ potvrde od ispravnih replika.

Blokčejn kao replikovani log (intuicija SMR-a)



Slika 2: Lanac blokova kao log: dogovor o istom lancu = dogovor o istom redosledu.

HotStuff u jednoj rečenici

HotStuff

Vođa predlaže blok, replike glasaju, a dokaz kvoruma se sumarizuje u **QC** (sertifikat kvoruma) koji nosi „najbolji“ napredak i kroz normalan rad i kroz promenu mandata.

Zašto je zanimljiv?

Dizajn teži **linearnom komunikacionom trošku** po mandatu i brzoj stabilizaciji nakon GST.

Osnovni objekti: Blok + QC

Blok

Sadrži komandu(e), pokazivač na roditelja i **opravdanje** (QC).

QC (*Quorum Certificate* - sertifikat kvoruma)

Kompaktan dokaz da je $2f + 1$ replika glasalo za isti blok u istoj fazi i mandatu.

Ključna ideja

QC je „prenosiva istina“: novi lider u sledećem mandatu ga koristi da bezbedno nastavi granu.

Bezbednost: zaključavanje + bezbedno glasanje

Lokalno stanje replike (intuicija)

- **highQC**: najveći QC koji replika zna (najbolji dokaz napretka)
- **lockedQC**: tačka zaključavanja (replika ne želi da pređe na konfliktnu granu)

Bezbedno glasanje (poenta)

Replika glasa ako predlog:

- nastavlja zaključanu granu, **ili**
- dolazi sa jačim opravdanjem (QC iz višeg mandata) koje „otključava“ prelazak.

Promena mandata (kako se oporavljamo od lošeg vođe)

Okidač

Ako nema napretka do isteka *timeout*-a, replika prelazi u novi mandat.

Šta se šalje novom lideru?

`NEW-VIEW` poruka koja sadrži **highQC** replike.

Šta radi novi lider?

Bira najveći pristigli QC i predlaže blok koji ga nastavlja (da bi ispravne replike mogle da glasaju).

Basic HotStuff: faze (šta se dešava u jednom mandatu)

Jedan blok prolazi kroz faze

$$\texttt{PREPARE} \rightarrow \texttt{PRE-COMMIT} \rightarrow \texttt{COMMIT} \rightarrow \texttt{DECIDE}$$

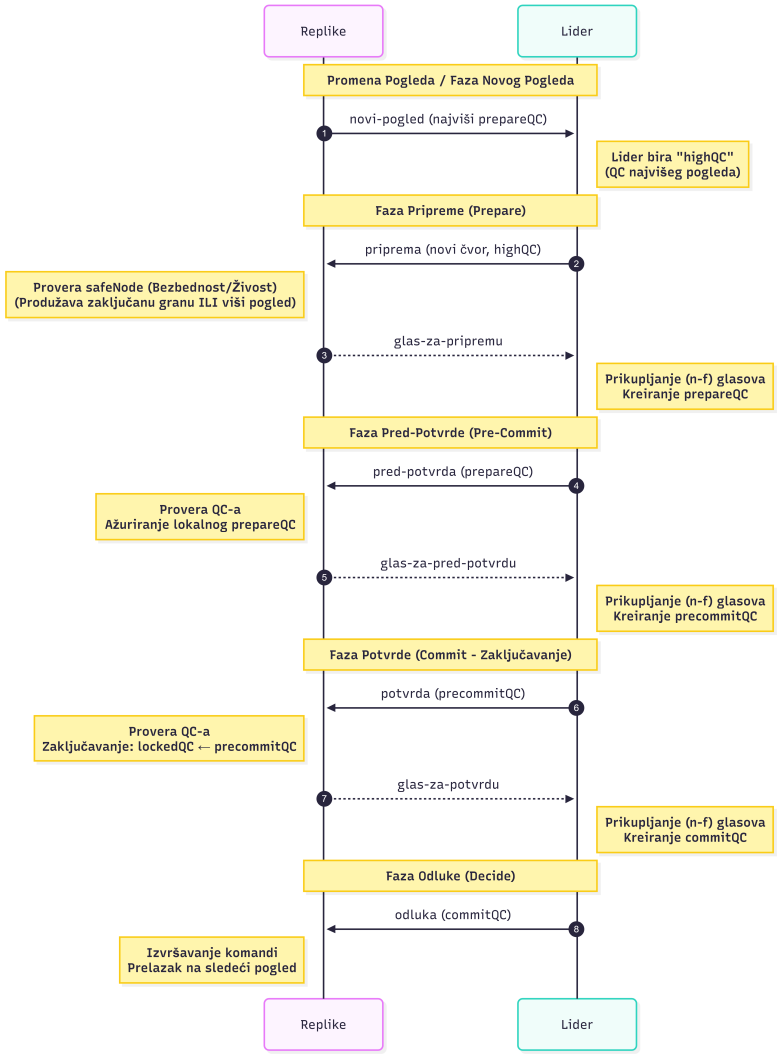
Intuicija

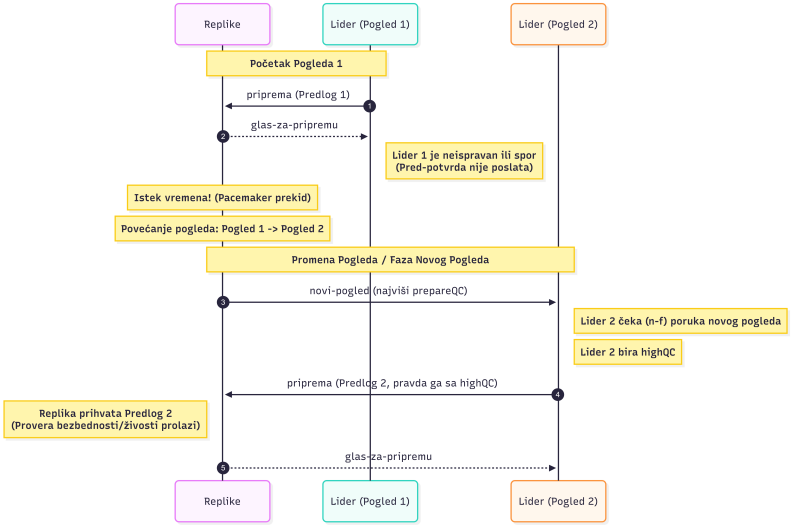
U svakoj fazi vođa prikuplja kvorum glasova i formira QC. Zaključavanje tipično nastaje kada replika vidi `precommitQC`.

Šta dobijamo?

Determinističku finalnost: kada dođe do komitovanja, odluka je nepovratna za ispravne replike.

Basic HotStuff: uspeh vs otkaz lidera (slika)

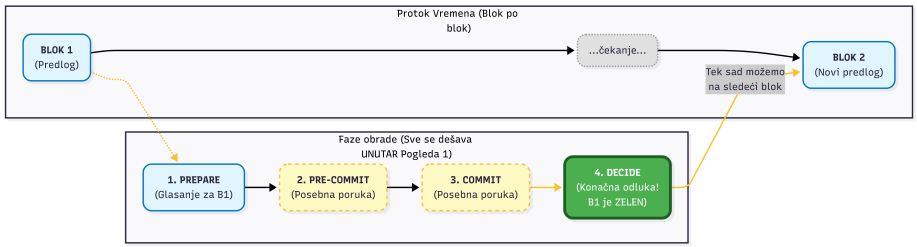




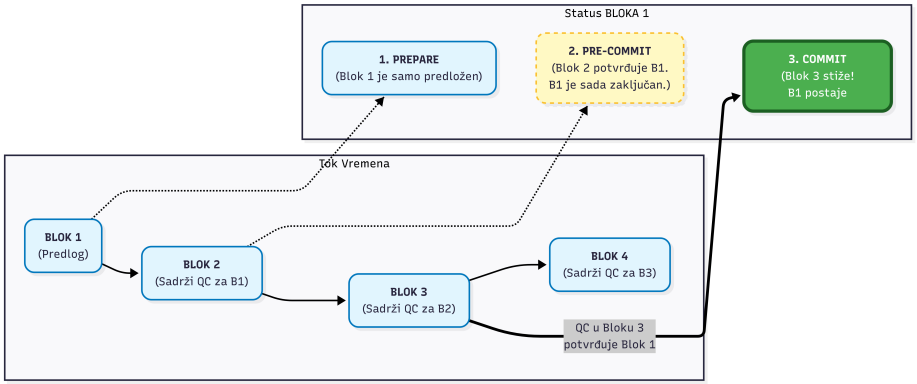
Otkaz/spor lider: isteci timeout, promena mandata, nastavak preko najvećeg QC-a.

Normalan tok: QC-ovi se formiraju i dolazi do finalizacije.

Chained HotStuff: protočnost (pipeline) umesto „čekanja“



Basic: jedan blok ide kroz sve faze.



Chained: QC-ovi na novim blokovima „guraju“ potvrđivanje starijih.

Poenta

Nakon popunjavanja *pipeline*-a, protokol može potvrđivati približno jedan blok po mandatu u stabilnom režimu.

Pacemaker i „optimistična odzivnost“

Pacemaker (modul živosti)

Upravlja *timeout*-ima, promenom mandata i izborom vođe.

Optimistička odzivnost

Nakon GST, brzina napredovanja prati realna mrežna kašnjenja (ne mora da čeka konzervativno velike fiksne *timeout*-e).

Diskretna simulacija

Mrežni model (konceptualno)

- kašnjenja poruka (stabilno/nestabilno)
- mogućnost blokiranja isporuke (partition)
- raspoređivanje događaja preko prioritetnog reda

Adaptivni pacemaker

Ideja

Timeout treba da se prilagođava lokalno posmatranom trajanju mandata (npr. vreme do QC-a).

EMA (*Estimated Mean Average*) + margina

$$T_{\text{next}} = \beta \cdot (\alpha \cdot T_{\text{obs}} + (1 - \alpha) \cdot T_{\text{prev}})$$

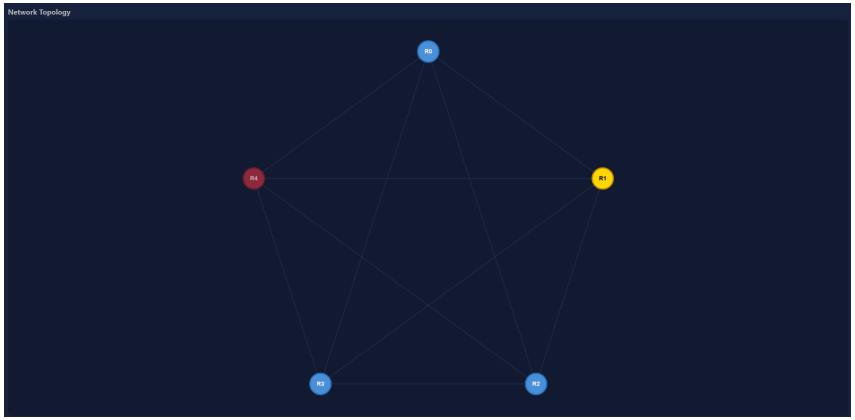
Backoff pri neuspehu

Kod uzastopnih isteka timeout-a, timeout se eksponencijalno uvećava do gornje granice.

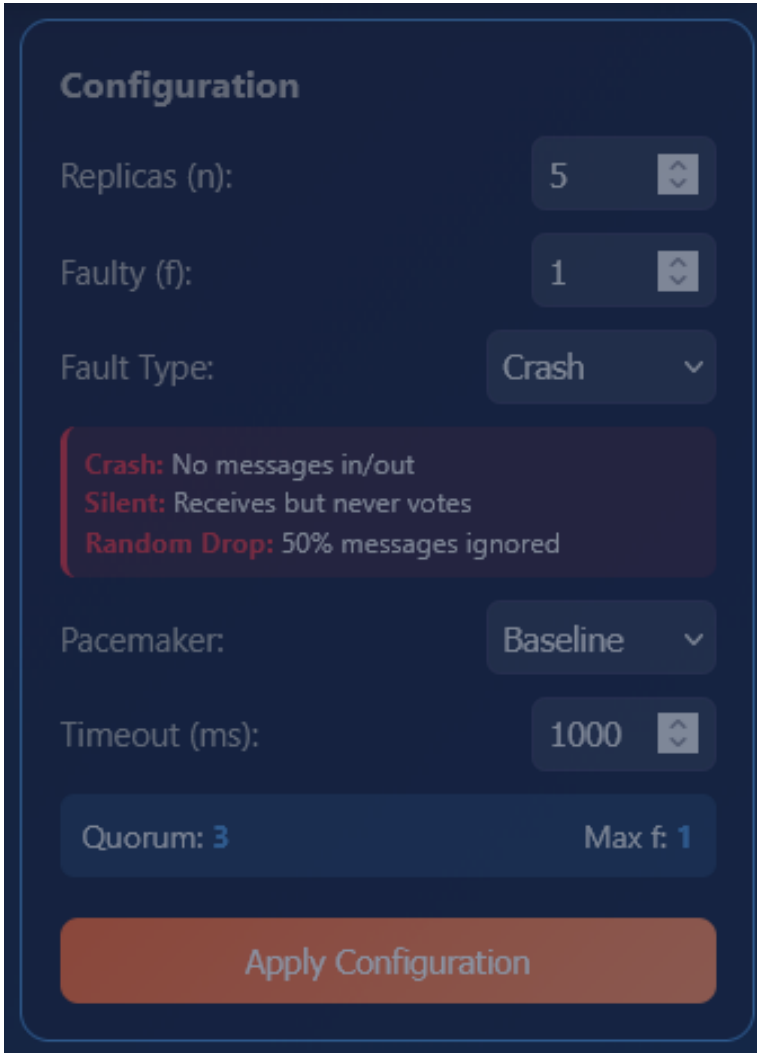
Dashboard

- Podešavanje n, f, modela greške i strategije pacemaker-a
- Praćenje lidera, poruka i lokalnog stanja replika
- Brza validacija ponašanja pre pokretanja batch eksperimenata

Dashboard - Primer

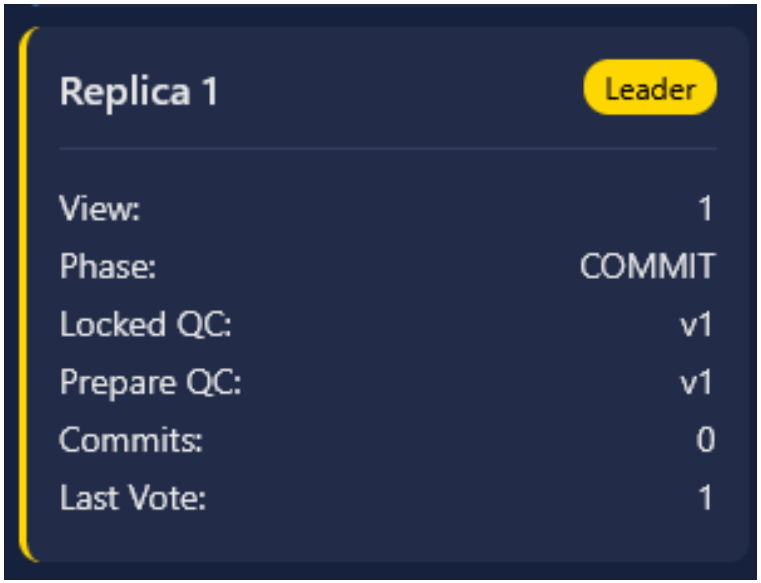


Topologija (lider / neispravne replike).

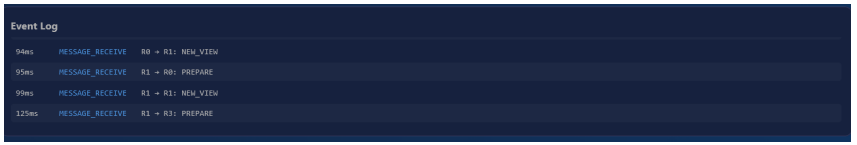


Panel za konfiguraciju.

Dashboard - Primer

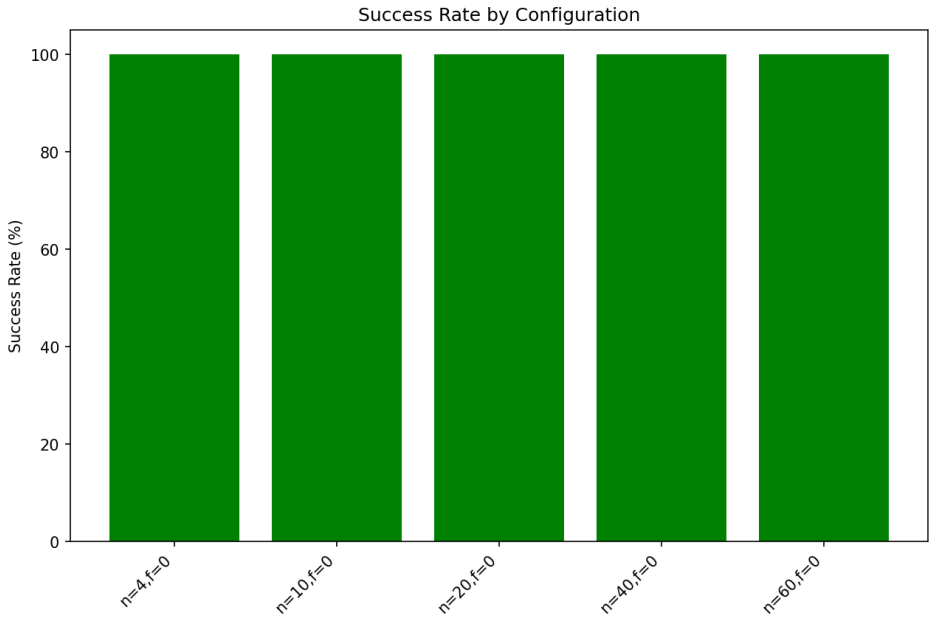


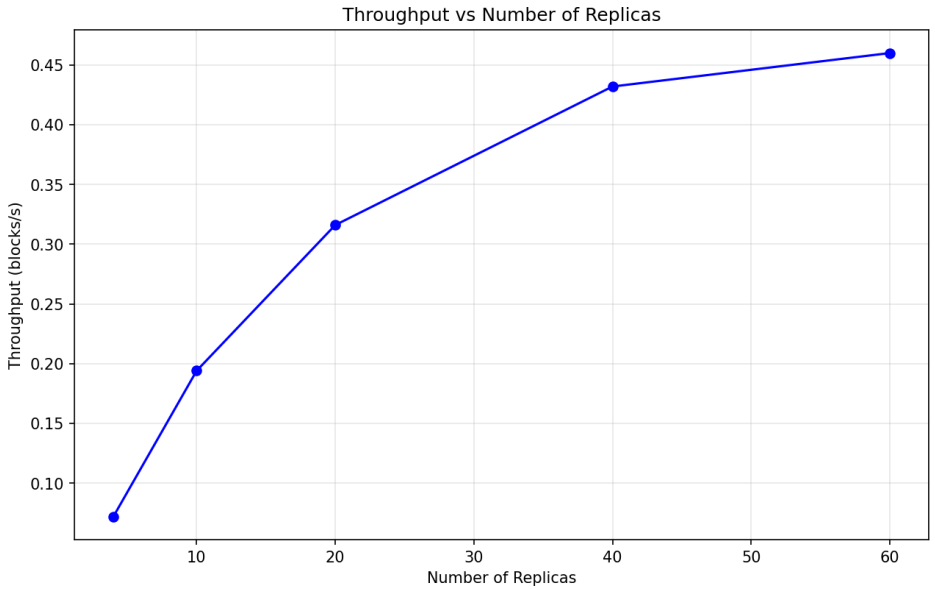
Inspektor replike (mandat, faza, QC-ovi).



Log događaja/poruka.

Scenario 1: skaliranje bez grešaka

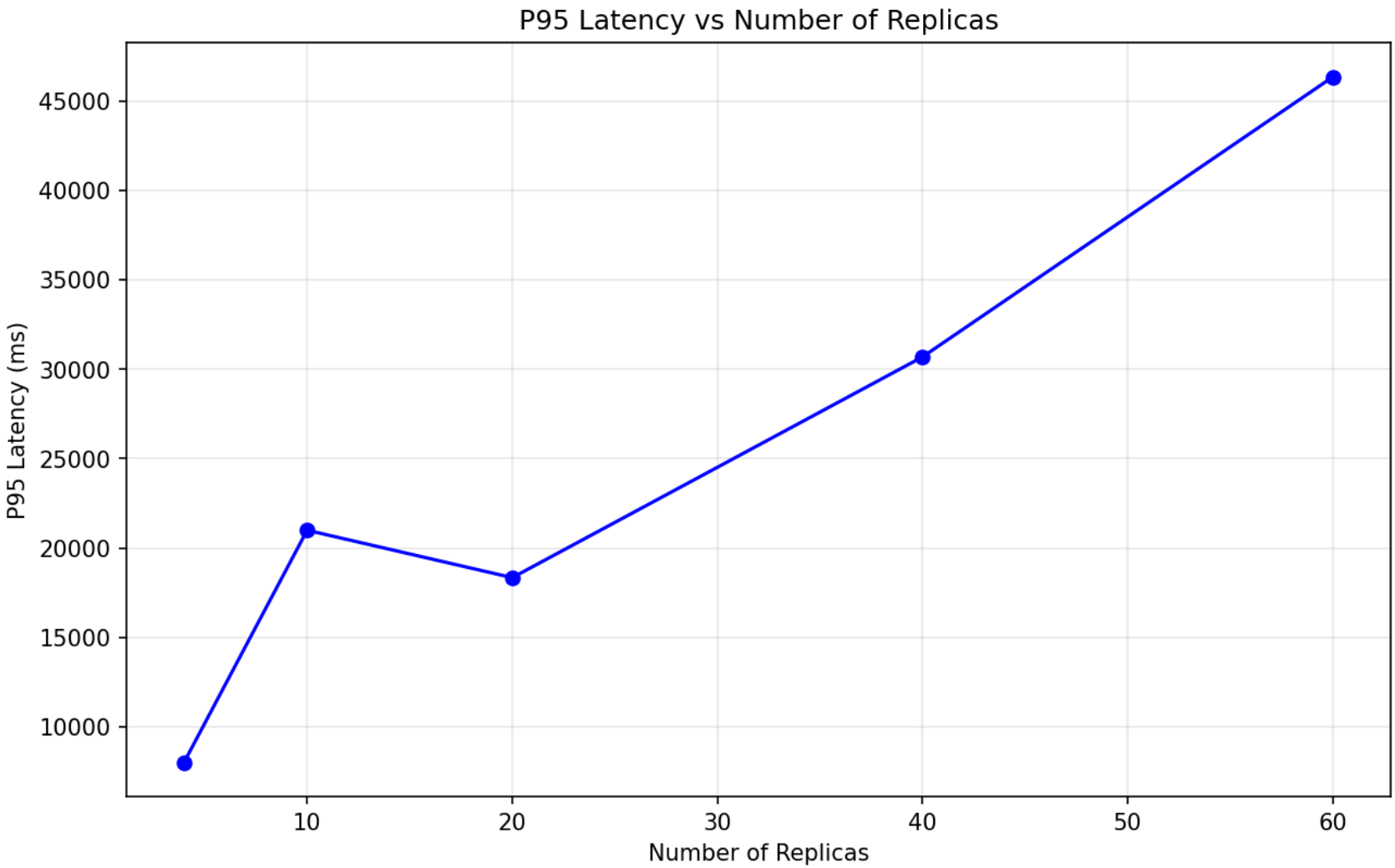




Nalaz

Sva pokretanja uspešna; propusnost raste sa n uz zasićenje (koordinacioni trošak lidera).

Scenario 1: mrežno kašnjenje



Slika 3: Repna latencija (p95) raste sa veličinom komiteta.

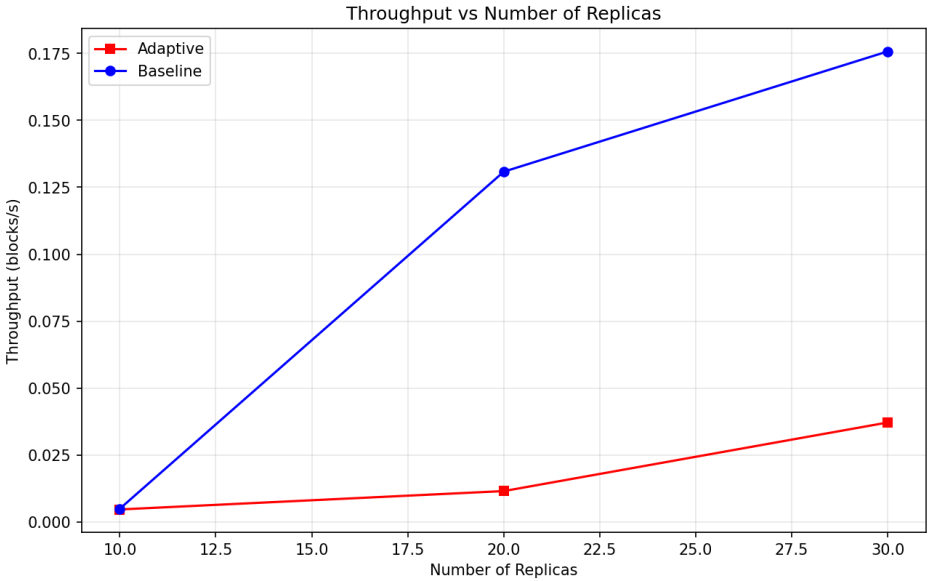
Scenario 2: greške ($n = 10, f = 3$)

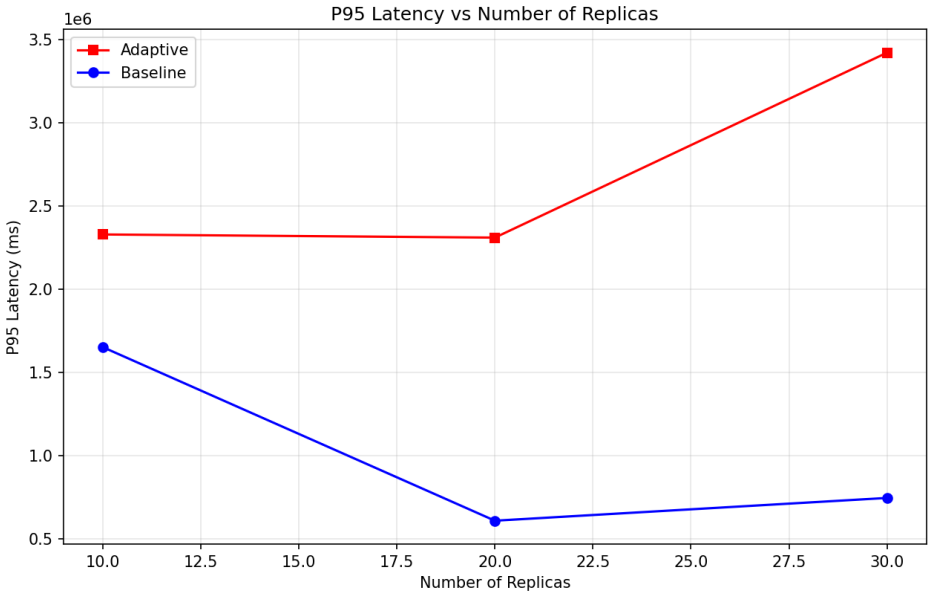
Greška	Uspeh	Blokovi	Timeout-i	Propusnost
Nema (ref)	100%	∼20	∼975	0%
Crash	100%	3.4	699	-83%
Silent	100%	4.8	995	-76%
RandomDrop	100%	15.0	985	-25%

Intuicija

Crash/Silent često onemogućavaju kvorum; RandomDrop ponekad „pusti“ dovoljno poruka da se kvorum formira.

Scenario 3: fiksni vs adaptivni pacemaker (RandomDrop 50%)

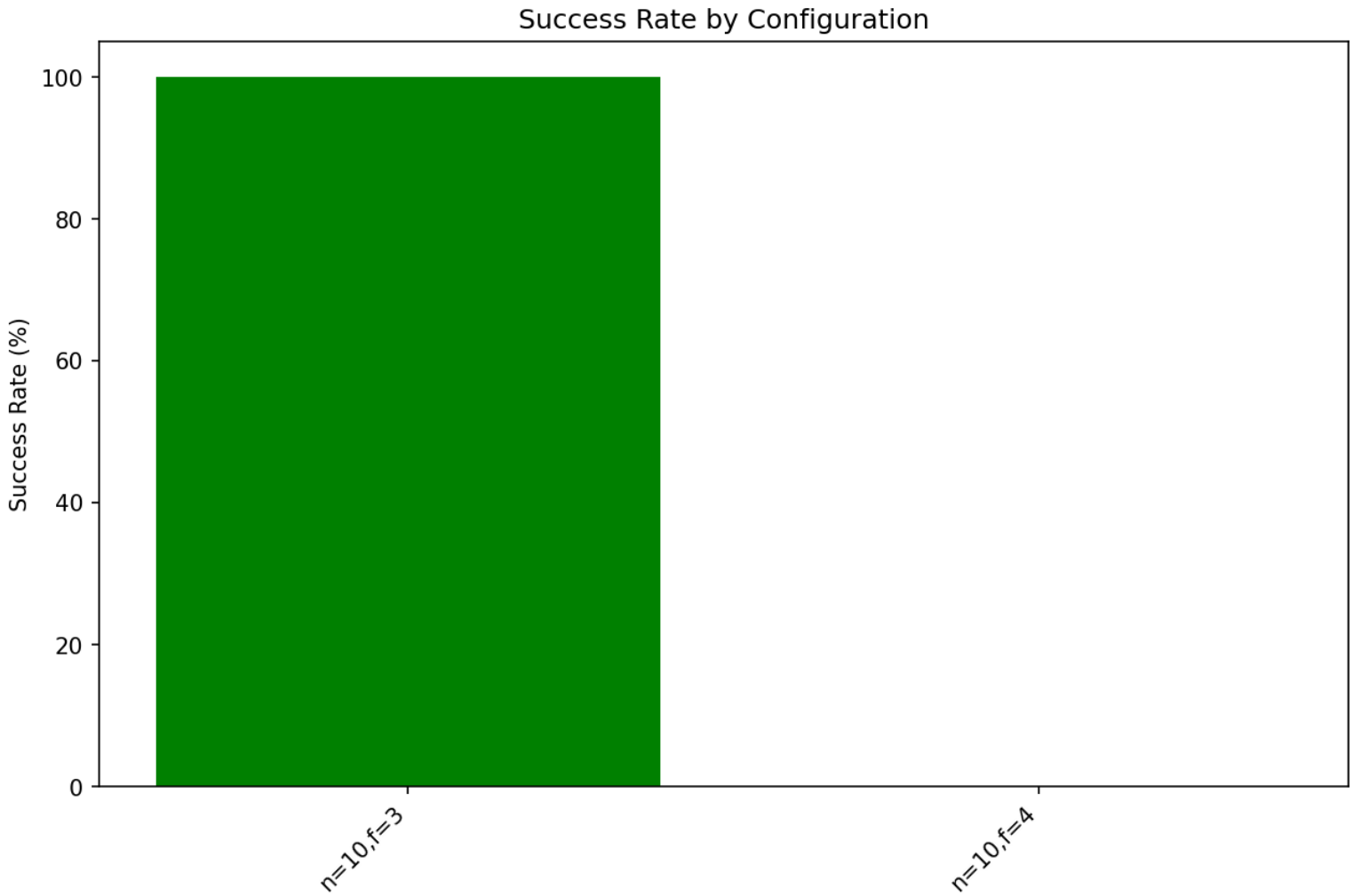




Nalaz

U režimu gubitka poruka, fiksni timeout daje veću propusnost i manju repnu latenciju.

Scenario 4: prekoračenje BFT praga ($n = 10, f = 4$)



Slika 4: Nakon prekoračenja praga, nema komitovanja (nema dovoljno glasova za kvorum).

Zaključci

- HotStuff kombinuje QC + zaključavanje + bezbedno glasanje da obezbedi bezbednost.
- Promena mandata prenosi napredak preko najvećeg QC-a i pomaže oporavak od lošeg lidera.
- Diskretna simulacija omogućava kontrolisane scenarije i jasne metrike (propusnost, latencija, timeout dinamika).
- Adaptivni timeout je koristan u promenljivim kašnjenjima, ali pri stohastičkom gubitku poruka „duže čekanje“ ne mora pomoći.

Literatura


M. Yin et al., *HotStuff: BFT Consensus with Linearity and Responsiveness*, PODC 2019.


C. Dwork, N. Lynch, L. Stockmeyer, *Consensus in the Presence of Partial Synchrony*, JACM 1988.


M. Castro, B. Liskov, *Practical Byzantine Fault Tolerance*, OSDI 1999.

Hvala!

Pitanja?