

Sveučilište u Rijeci, Fakultet informatike i digitalnih tehnologija

Sveučilišni diplomski studij Informatike

Duje Vidas

In Memory Database vs Normal Database

Eksperimentalni rad

Nositelj: dr. sc. Rok Pilvater

Rijeka, 24.12.2023

Usporedna procjena PostgreSQL-a i Redisa: prednosti, ograničenja i slučajevi uporabe

PostgreSQL

Arhitektura i prednosti: PostgreSQL djeluje kao robustan sustav za upravljanje relacijskim bazama podataka. Njegova arhitektura omogućuje ACID (Atomicity, Consistency, Isolation, Durability) usklađenost, osiguravajući cjelovitost podataka čak i u složenim transakcijama. To ga čini prikladnim za aplikacije koje zahtijevaju strogu dosljednost podataka, kao što su financijski sustavi ili upravljanje sadržajem.

Prednosti:

- **Relacijski model:** PostgreSQL-ov relacijski model omogućuje složeno strukturiranje podataka i odnose, olakšavajući sofisticirane upite i spajanja.
- **Skalabilnost:** nudi horizontalnu skalabilnost kroz dijeljenje i replikaciju strujanja, što mu omogućuje rukovanje većim skupovima podataka i istodobnim vezama.
- **Integritet podataka:** usklađenost s ACID-om osigurava dosljednost podataka čak i tijekom istodobnih transakcija ili kvarova sustava.

Nedostaci:

- **Priroda koja se temelji na disku:** Mehanizam za pohranu na disku može rezultirati malo sporijim operacijama čitanja i pisanja u usporedbi s bazama podataka u memoriji.
- **Složena konfiguracija:** Upravljanje i konfiguriranje PostgreSQL-a za visoku dostupnost i izvedbu može zahtijevati stručnost i trud.

Idealni slučajevi upotrebe:

- Aplikacije koje zahtijevaju snažan integritet podataka, složene odnose podataka i usklađenost s ACID-om.
- Sustavi koji se bave velikim skupovima podataka i transakcijama koje zahtijevaju relacijske sposobnosti.

Redis

Arhitektura i prednosti: Redis radi kao pohrana strukture podataka u memoriji, pohranjujući podatke prvenstveno u RAM. Njegov dizajn pohrane ključ-vrijednosti omogućuje munjevito brz pristup podacima i manipulaciju. Redisova arhitektura podržava različite podatkovne strukture, uključujući nizove, hashove, popise, skupove i sortirane skupove.

Prednosti:

- **Pohrana u memoriji:** Velike brzine pristupa zahvaljujući podacima koji se nalaze u memoriji, što ga čini iznimno brzim za operacije koje zahtijevaju puno čitanja.
- **Predmemoriranje:** Idealno za predmemoriranje podataka kojima se često pristupa, smanjujući opterećenje primarnih baza podataka i poboljšavajući ukupne performanse sustava.
- **Strukture podataka:** Svestrane strukture podataka omogućuju složene operacije poput atomskih brojača, upravljanja pločom s najboljim rezultatima i analitike u stvarnom vremenu.

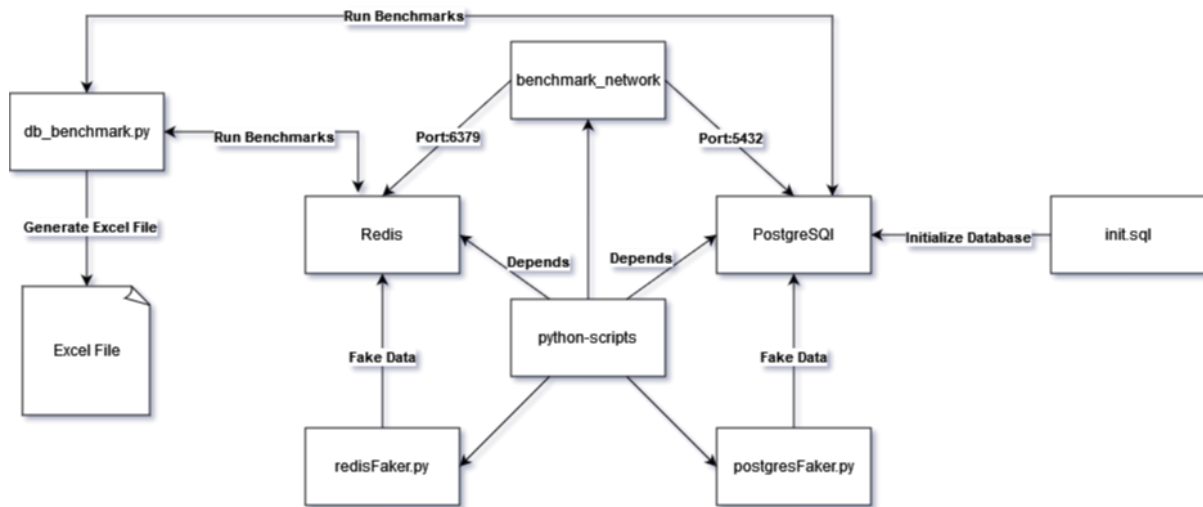
Nedostaci:

- **Ograničenje memorije:** Budući da je vezan za memoriju, Redis bi se mogao suočiti s ograničenjima u rukovanju iznimno velikim skupovima podataka u usporedbi sa sustavima koji se temelje na disku.
- **Složenost postojanosti:** Konfiguriranje opcija postojanosti za trajnost može uključivati složenosti.

Idealni slučajevi upotrebe:

- **Sloj predmemorije:** Idealan kao predmemorija za podatke kojima se često pristupa.
- **Analitika u stvarnom vremenu:** Pogodno za scenarije koji zahtijevaju brzi pristup podacima i manipulaciju, poput analitike u stvarnom vremenu i upravljanja sesijom.

Arhitektura sustava



Slika 1 Slika arhitecture sustava

Objašnjenje simulacije/lažnog opterećenja

1. Obrasci pristupa podacima: Varijacija omjera čitanja/pisanja

- **Uključene funkcije:** *simulate_read_write_ratio*, *redis_read_operation*, *redis_write_operation*, *postgres_read_operation*, *postgres_write_operation*
- **Obrazloženje:**
 - *simulate_read_write_ratio(ratio)*: Simulira različite omjere čitanja/pisanja, izračunavajući propusnost izvršavanjem operacija čitanja i pisanja.
 - *redis_read_operation()* i *redis_write_operation()*: Izvršava operacije čitanja i pisanja u Redisu, oponašajući obrasce pristupa podacima s nasumičnim ključevima i vrijednostima.
 - *postgres_read_operation()* i *postgres_write_operation()*: Slično Redisu, izvršavanje operacija čitanja i pisanja u PostgreSQL-u za simulaciju različitih obrazaca pristupa.

2. Emulacija neravnomjernog pristupa podacima: Distribucija ključa

- **Uključene funkcije:** *simulate_key_access_frequencies*
- **Obrazloženje:**
 - *simulate_key_access_frequencies()*: Demonstrira kako se različitim ključevima pristupa u Redisu i PostgreSQL-u koristeći unaprijed definirane scenarije. Procjenjuje razlike u vremenu odgovora na temelju specifičnih frekvencija pristupa ključu.

3. Konkurentne operacije: Demonstracija konkurentnosti

- **Uključene funkcije:** *execute_concurrent_operations_redis*, *execute_concurrent_operations_postgres*
- **Obrazloženje:**
 - *execute_concurrent_operations_redis()* i *execute_concurrent_operations_postgres()*: Simuliranje istodobne operacije čitanja i pisanja u Redisu i PostgreSQL-u, procjenjujući njihovo rukovanje simultanim operacijama.

4. Simulacije transakcija: Teške transakcije

- **Uključene funkcije:** *read_heavy_transaction_redis*, *read_heavy_transaction_postgres*, *write_heavy_transaction_redis*, *write_heavy_transaction_postgres*, *pg_perform_transactional_operations*, *redis_perform_transactional_operations*
- **Obrazloženje:**
 - *read_heavy_transaction_redis* i *read_heavy_transaction_postgres*: Simuliranje teške transakcije čitanja u Redisu i PostgreSQL-u izvršavanjem velikog broja operacija čitanja.
 - *write_heavy_transaction_redis* i *write_heavy_transaction_postgres*: Simuliranje teške transakcije pisanja u Redisu i PostgreSQL-u izvršavanjem velikog broja operacija pisanja.
 - *pg_perform_transactional_operations* i *redis_perform_transactional_operations*: Simuliranje transakcijske operacije u obje baze podataka, uključujući operacije čitanja, pisanja i ažuriranja.

5. Mjerila vremena odziva: Procjena učinka

- **Uključene funkcije:** Razne funkcije kroz skriptu
- **Obrazloženje:**
 - Funkcije za izračunavanje istodobnih vremena, vremena odgovora, vremena odziva učitavanja retka i metrike transakcijskih operacija pružaju uvid u performanse Redisa i PostgreSQL-a. Ove metrike, sažete u DataFrames, omogućuju usporedbu prosječne i srednje izvedbe.

Skripta koristi funkcije za simulaciju različitih scenarija, procjenjujući Redis i PostgreSQL u rukovanju različitim obrascima pristupa podacima, konkurentnosti, teškim transakcijama i metrikama performansi. Svaka funkcija emulira specifične aspekte ponašanja baze podataka, pomažući u razumijevanju i usporedbi njihove izvedbe pod različitim radnim opterećenjima.

Mjerenje vremena odziva i protoka

1. Mehanizam za bilježenje vremena:

- Skripta je koristila funkciju *time.time()* za hvatanje točnih vremenskih oznaka prije i poslije svake operacije, olakšavajući precizno mjerenje vremena početka i završetka.

2. Izvođenje operacije:

- Niz operacija čitanja i pisanja proveden je u obje baze podataka, koristeći namjenske funkcije za svaku vrstu operacije (*redis_read_operation()*, *redis_write_operation()*, *postgres_read_operation()*, *postgres_write_operation()*).

3. Izračun vremena odziva:

- Nakon svakog izvršenja operacije, skripta je izračunala potrošeno vrijeme (u sekundama) oduzimanjem vremena početka od vremena završetka: *end_time - start_time*.
- Ovaj proces bilježi individualna vremena odgovora za svaku operaciju čitanja i pisanja u PostgreSQL-u i Redisu.

4. Zbirni rezultati:

- Kako bi se utvrdila prosječna vremena odgovora, skripta je izračunala srednju vrijednost svih zabilježenih vremena odgovora za svaku vrstu operacije (čitanje/pisanje) u obje baze podataka.
- Formula za izračunavanje prosječnog vremena odziva: *Sum of Response Times / Number of Operations*.

5. Mjerenje propusnosti:

- Propusnost predstavlja brzinu operacija obrađenih unutar određenog vremenskog okvira, koja se obično mjeri u operacijama u sekundi (ops/sec).
- Propusnost je određena dijeljenjem ukupnog broja izvršenih operacija s ukupnim vremenom potrebnim za izvođenje tih operacija.
- Formula za izračun propusnosti: *Number of Operations / Total Time Taken*.

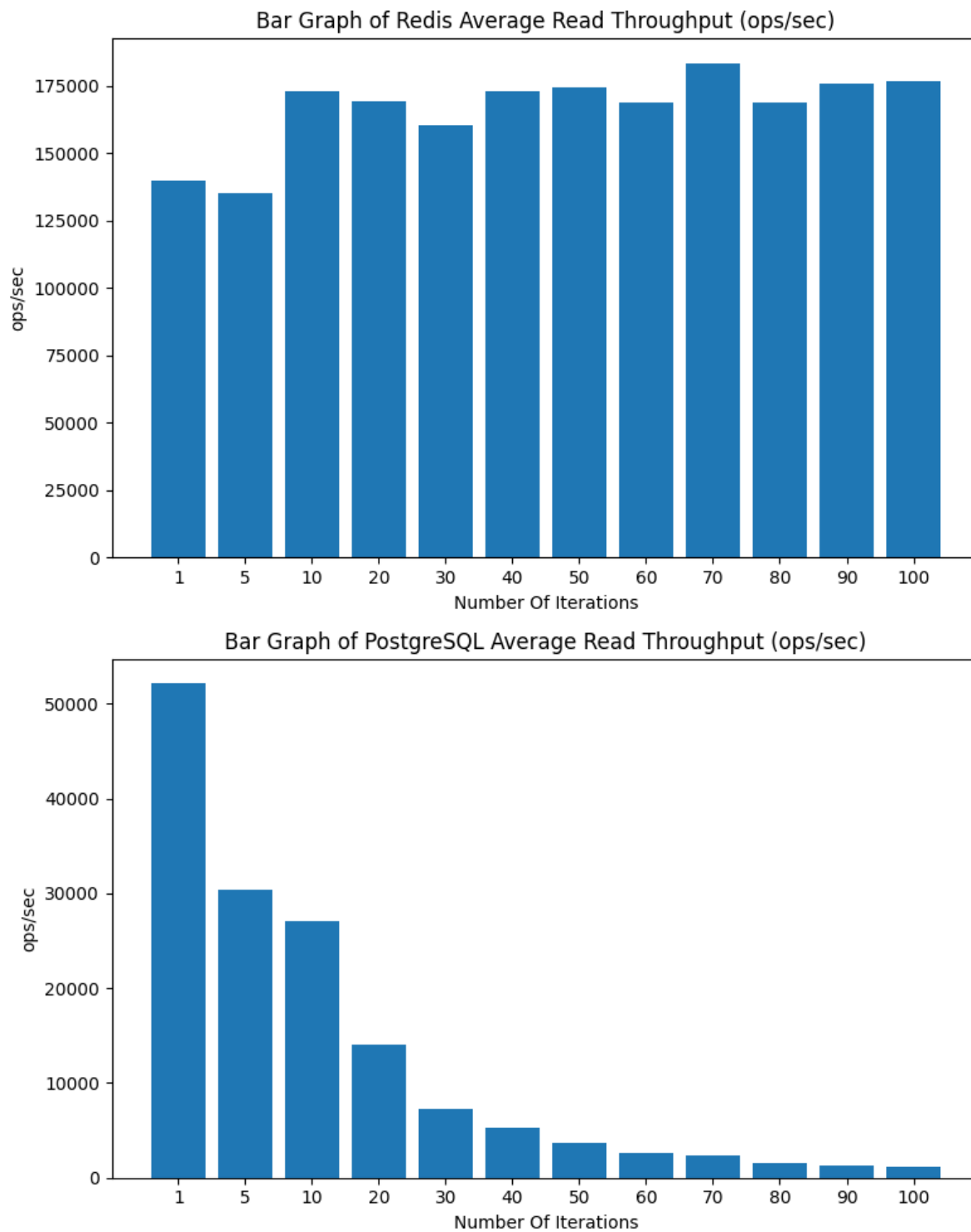
6. Percentilna analiza:

- Uz prosječna vremena odgovora, skripta je izvršila izračune za medijan vremena odgovora.

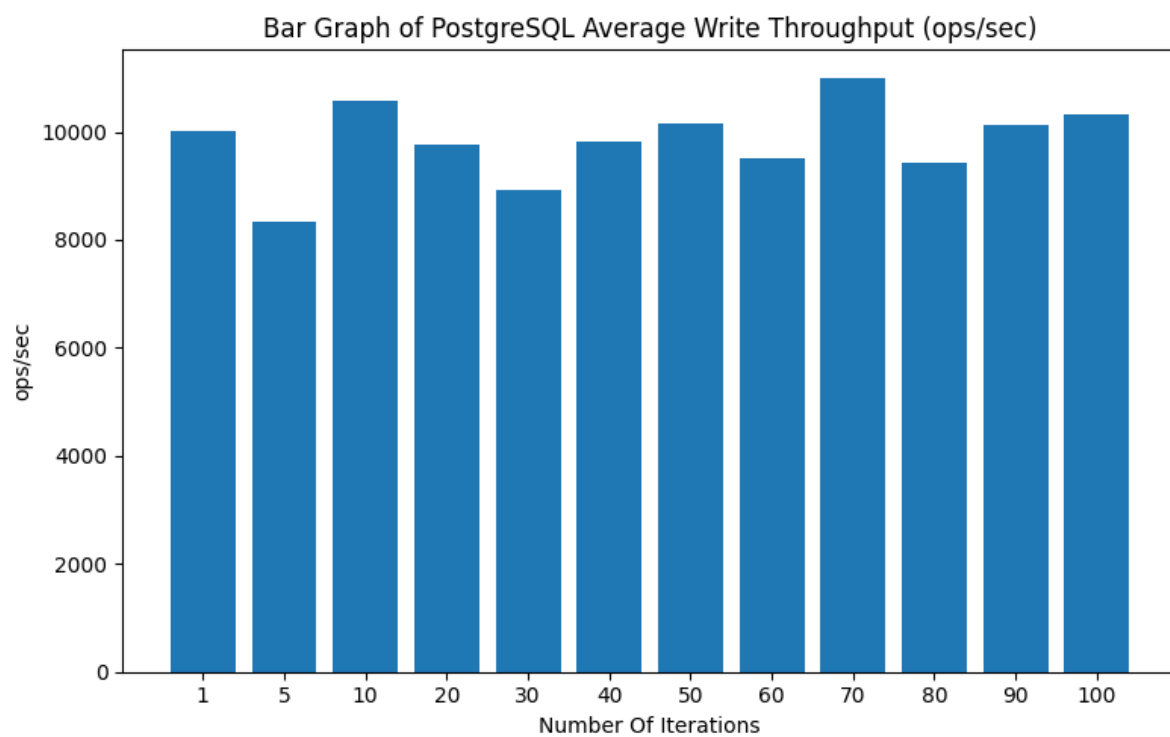
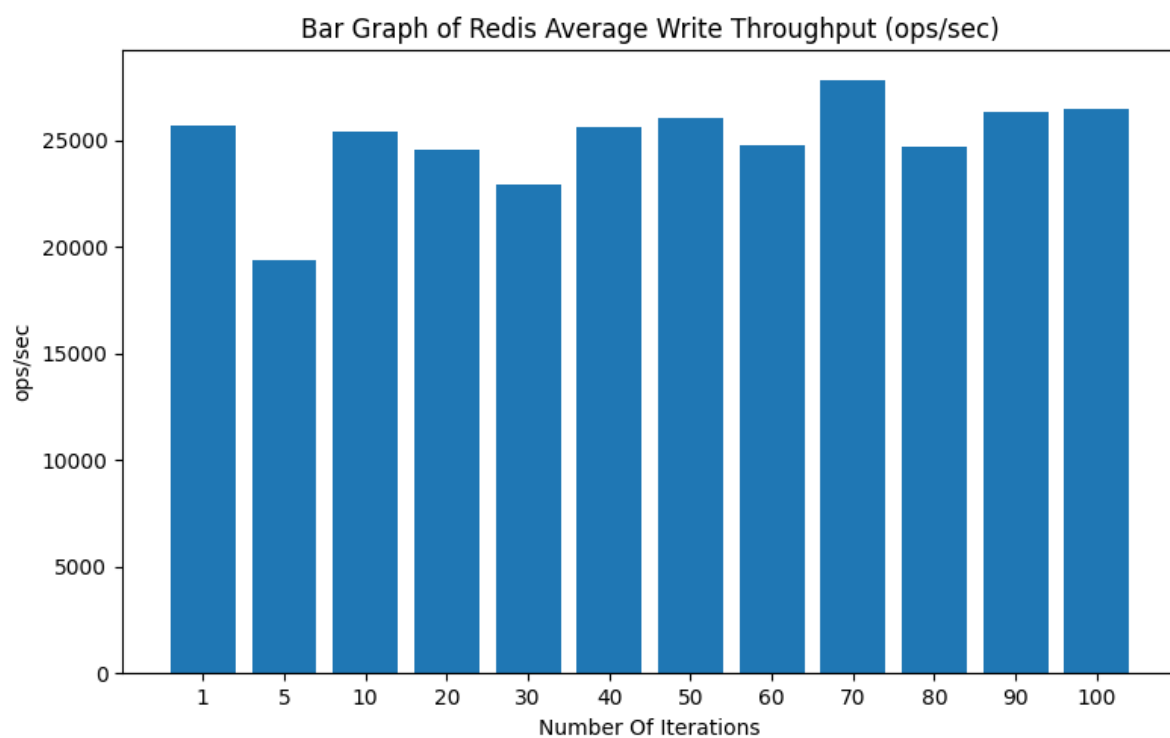
7. Usporedba i vizualizacija:

- Naposljetku, skripta je agregirala, usporedila i predstavila vremena odgovora i metriku protoka između Redis-a i PostgreSQL-a.
- Ove metrike organizirane su u obliku tablice kako bi se olakšalo jasnije tumačenje i usporedba.

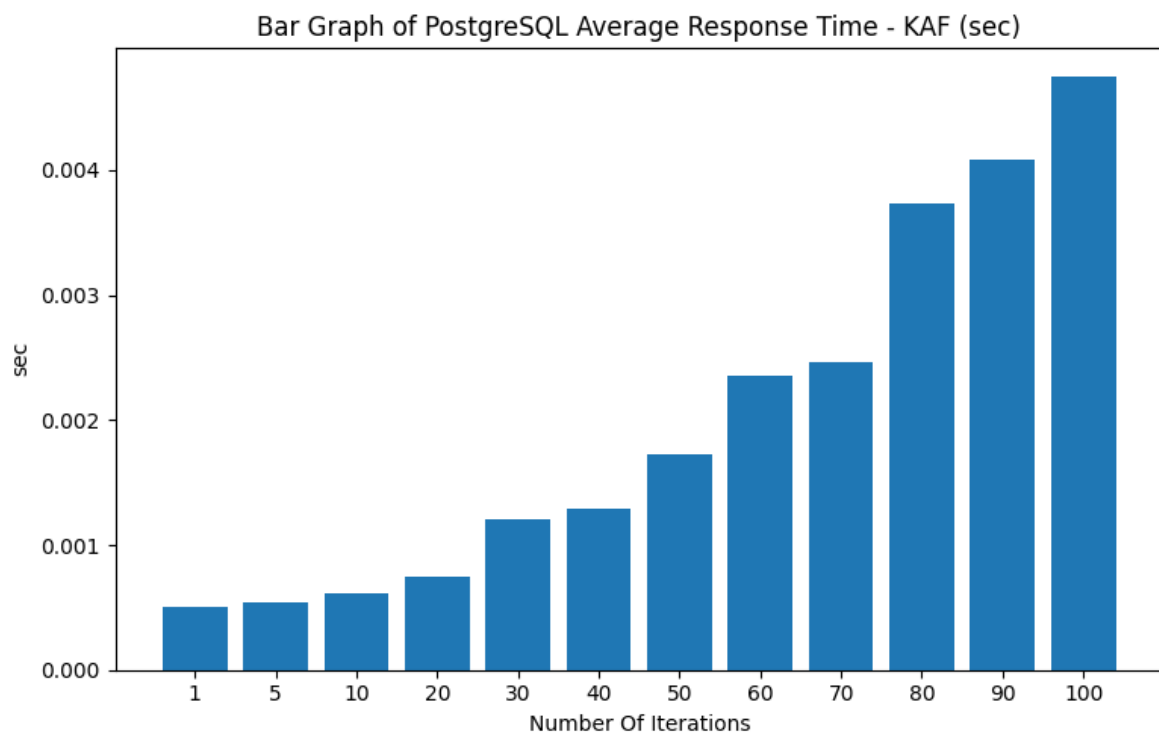
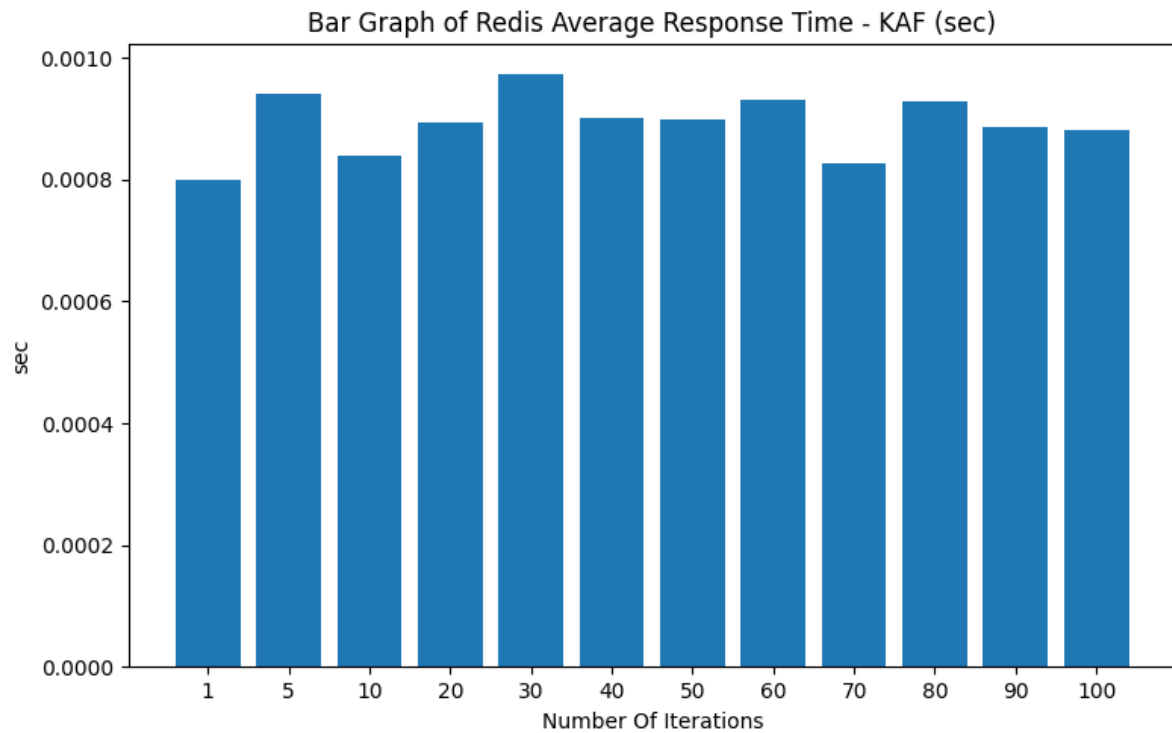
Mjerenja



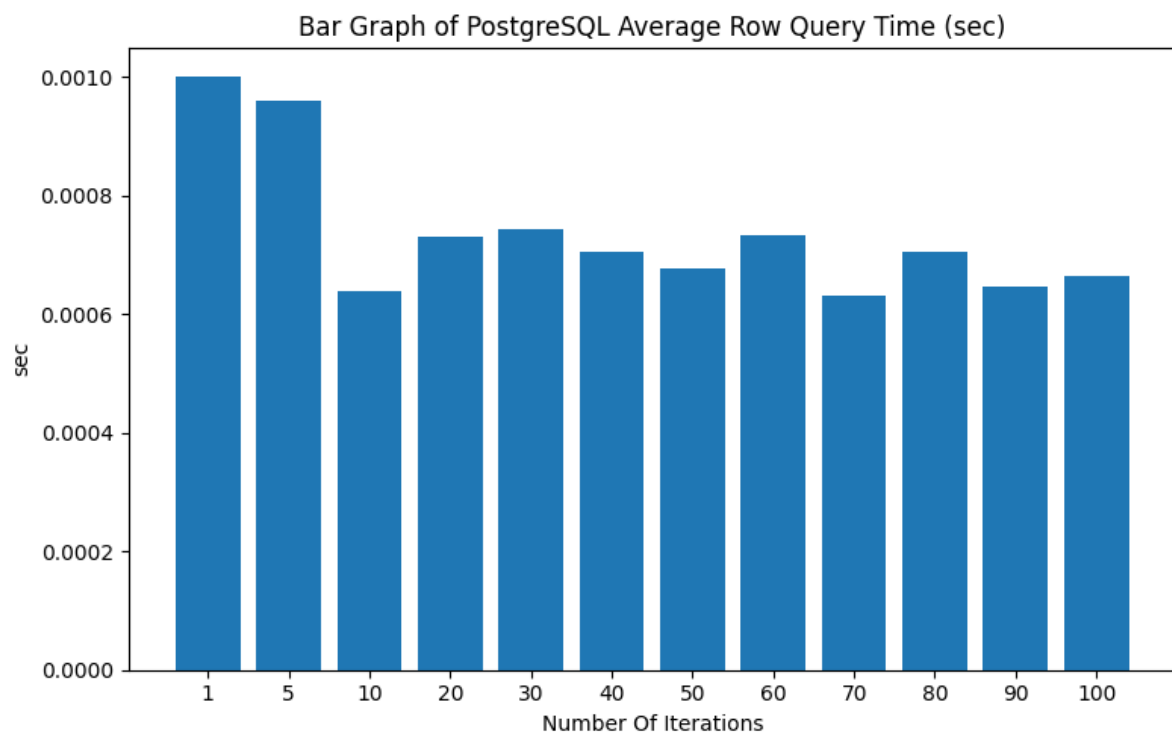
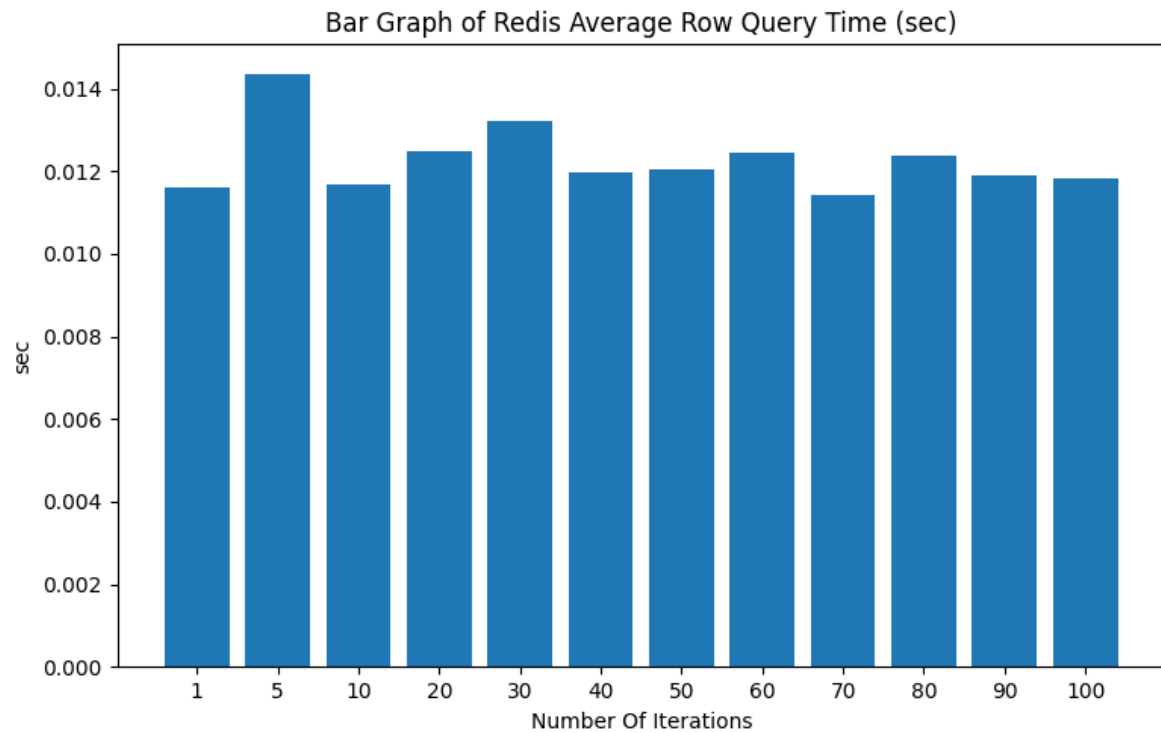
Slika 2 Redis u odnosu na PostgreSQL: usporedba prosječnog protoka čitanja (ops/sec).



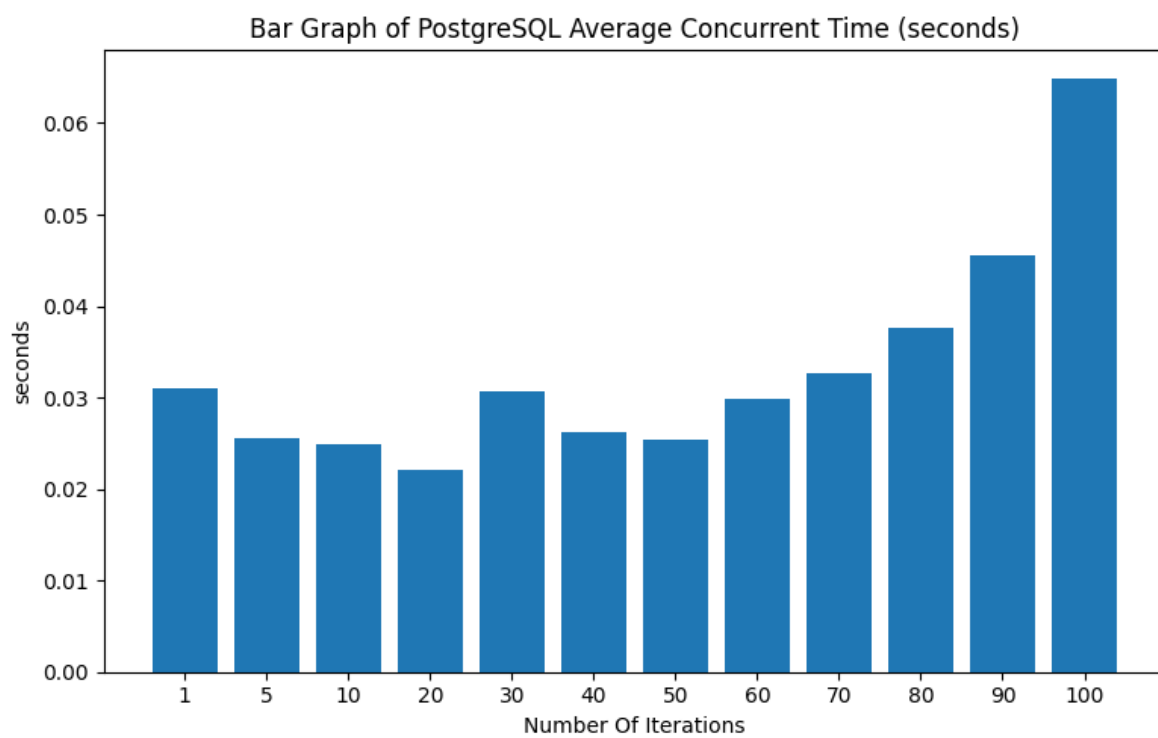
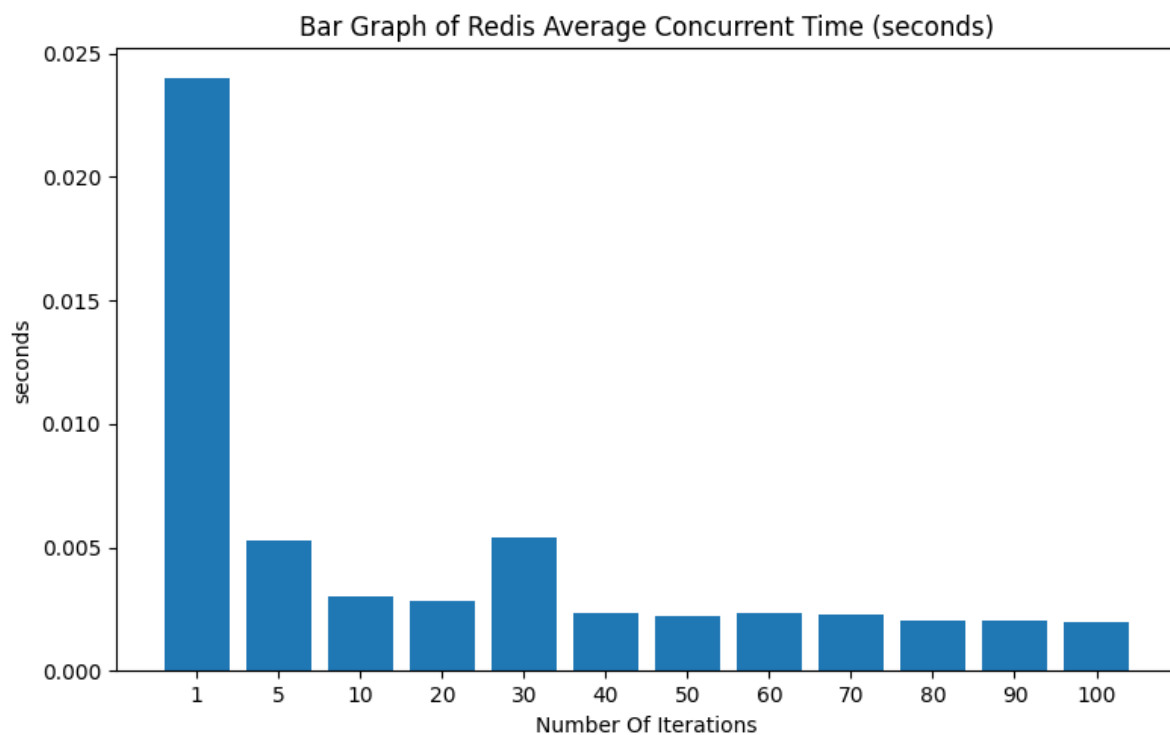
Slika 3 Redis u odnosu na PostgreSQL: usporedba prosječnog protoka pisanja (ops/sec).



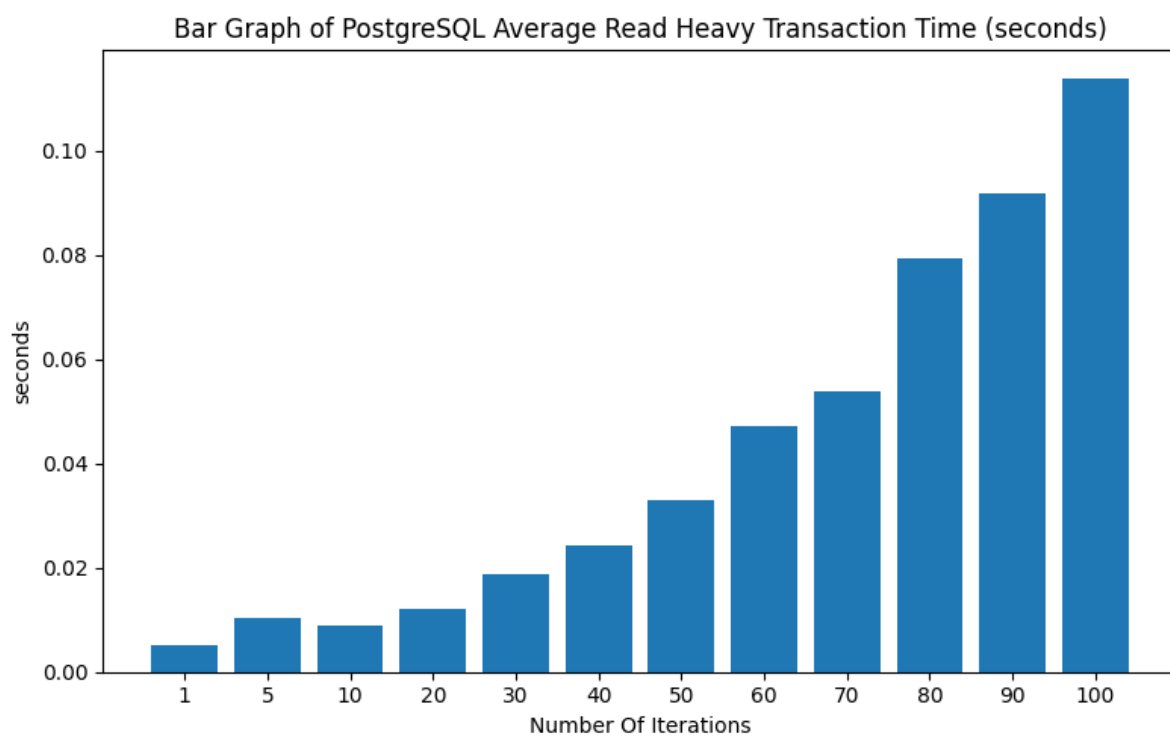
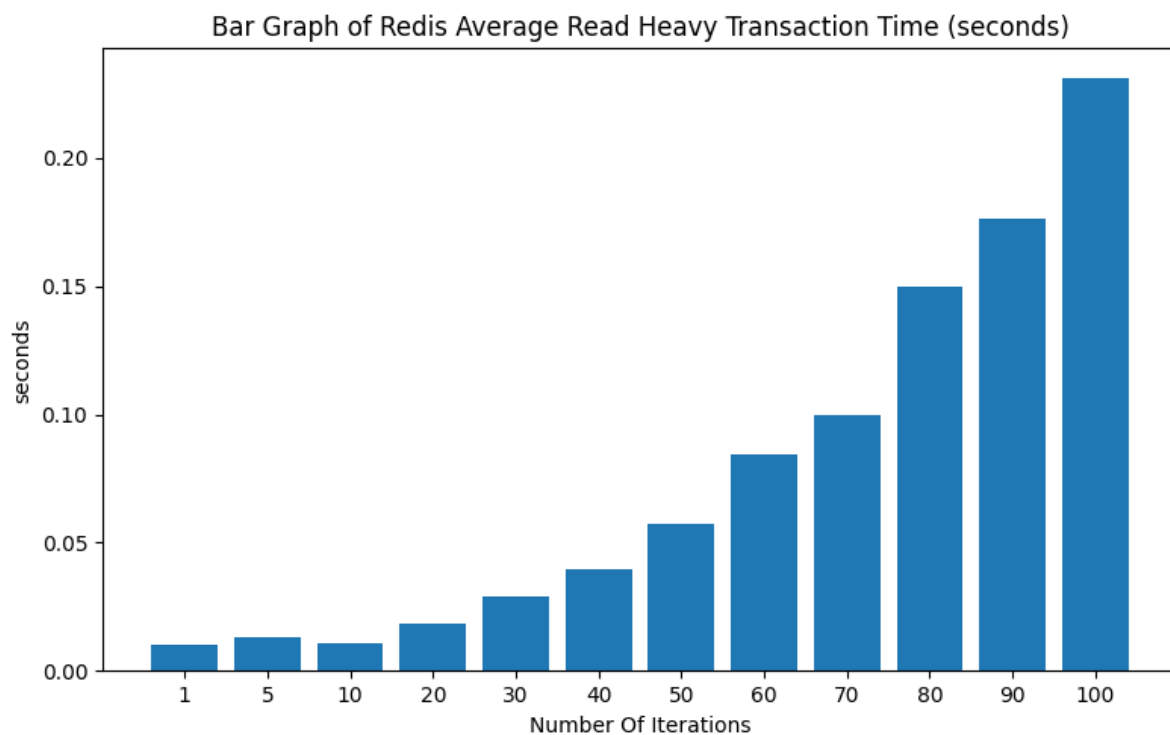
Slika 4 Redis u odnosu na PostgreSQL: usporedba prosječnog vremena odgovora - KAF (sek).



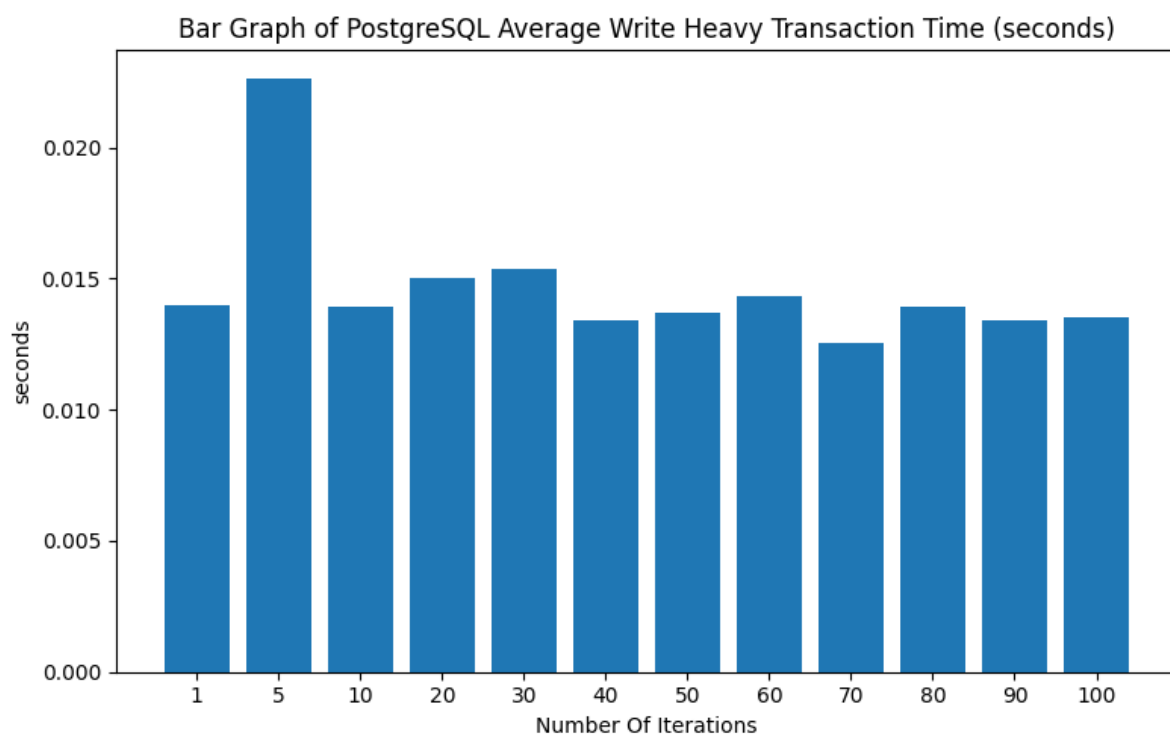
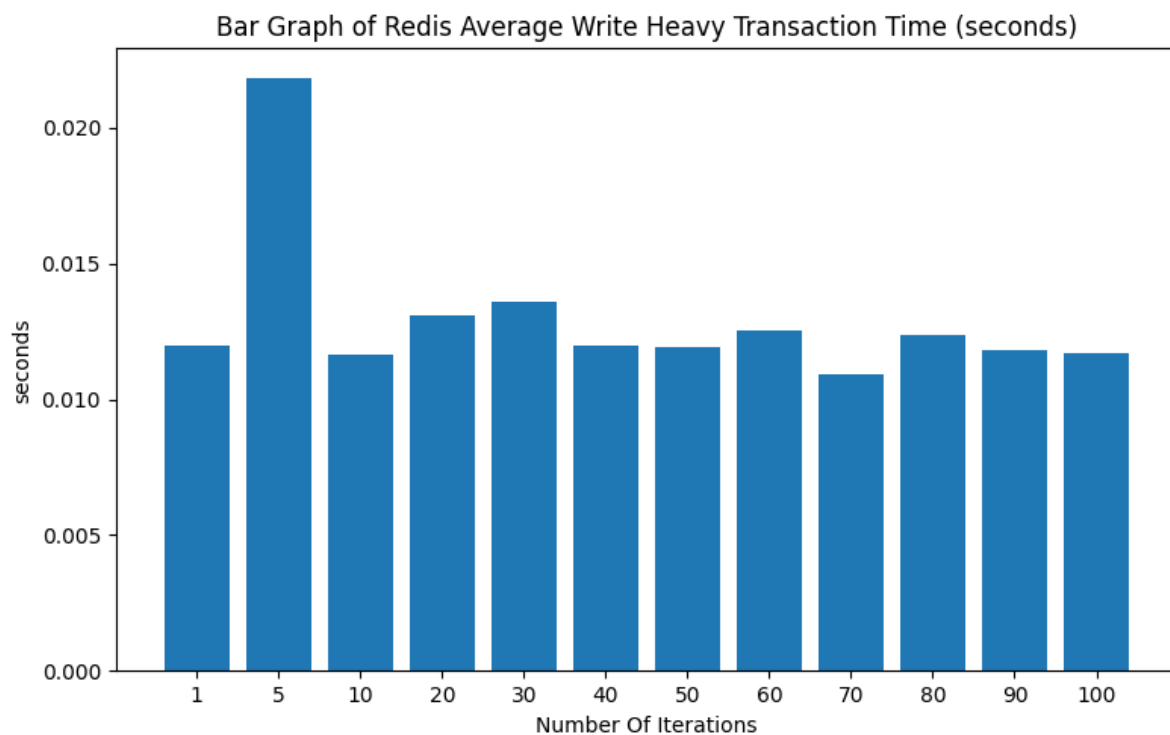
Slika 5 Redis u odnosu na PostgreSQL: Usporedba prosječnog vremena upita retka (sec)



Slia 6 Redis u odnosu na PostgreSQL: Usporedba prosječnog vremena istovremenog rada (sekunde).



Slika 7 Redis u odnosu na PostgreSQL: usporedba prosječnog vremena teške transakcije čitanja (sekunde)



Slika 8 Redis u odnosu na PostgreSQL: usporedba prosječnog vremena teške transakcije pisanja (sekunde)

Ovi skupovi podataka prikazuju rezultate benchmarkinga za Redis i PostgreSQL kroz različite iteracije.

Za svaku iteraciju zabilježeno je nekoliko mjernih podataka:

1. Prosječna propusnost čitanja (ops/sek):

- Za Redis postoji relativno stabilna izvedba kroz iteracije, s uočenim manjim fluktuacijama. U početku postoji blagi silazni trend, nakon čega slijedi lagani oporavak.
- PostgreSQL, s druge strane, prikazuje dosljedno smanjenje propusnosti čitanja sa svakom iteracijom, što ukazuje na potencijalnu degradaciju performansi.

2. Prosječna propusnost pisanja (ops/sek):

- Redis pokazuje fluktuirajući obrazac u protoku pisanja, pokazujući različite razine kroz iteracije, ali bez jasnog uzlaznog ili silaznog trenda.
- Suprotno tome, PostgreSQL stalno doživljava pad u propusnosti pisanja, što ukazuje na potencijalno usko grlo ili ograničenje performansi u scenarijima intenzivnog pisanja.

3. Prosječno vrijeme odziva - KAF (sek):

- I Redis i PostgreSQL pokazuju sve veći trend vremena odgovora tijekom iteracija. Međutim, vrijeme odgovora PostgreSQL-a značajno se povećava u usporedbi s Redisom. To sugerira da bi se PostgreSQL s povećanjem radnog opterećenja mogao više boriti s održavanjem nižih vremena odgovora nego Redis.

4. Prosječno vrijeme upita retka (sek):

- Obje baze podataka prikazuju dosljednu uzlaznu putanju u vremenu upita. Međutim, PostgreSQL doživljava značajno veće povećanje vremena upita od Redisa. To bi moglo značiti da bi se, kako se radno opterećenje povećava, PostgreSQL mogao više boriti s izvedbom upita nego Redis.

5. Prosječno vrijeme istovremenog rada (sekunde):

- Redis općenito pokazuje trend pada u istovremenom vremenu kroz iteracije. To bi moglo značiti poboljšanu učinkovitost ili optimizaciju u rukovanju istodobnim operacijama.
- Istodobno vrijeme PostgreSQL-a pokazuje fluktuacije, ali naginje prema rastućem trendu, što ukazuje na potencijalne izazove u učinkovitom rukovanju istodobnim operacijama s povećanjem radnog opterećenja ili ponavljanja.

6. Prosječno vrijeme teške transakcije čitanja (sekunde):

- Obje baze podataka pokazuju postupno povećanje vremena transakcije za čitanje, što ukazuje na potencijalno povećanje vremena potrebnog za transakcije usmjerene prema čitanjima.
- Čini se da PostgreSQL ima dosljedniji i malo strmiji porast u usporedbi s Redisom.

7. Prosječno vrijeme teške transakcije pisanja (sekunde):

- Slično vremenu teške transakcije čitanja, obje baze podataka prikazuju uzlazni trend vremena teške transakcije pisanja.
- PostgreSQL pokazuje dosljedniji porast tijekom iteracija, što sugerira izraženiji izazov u rukovanju transakcijama s intenzivnim pisanjem kako se radno opterećenje povećava.

Ova zapažanja pružaju uvid u karakteristike izvedbe Redisa i PostgreSQL-a u različitim intenzitetima radnog opterećenja. Očito je da dok Redis općenito održava stabilnije metrike performansi kroz iteracije, PostgreSQL se suočava s izazovima kao što je pad propusnosti i sve dulje vrijeme odgovora i upita kako se radno opterećenje povećava

Sveukupno, čini se da Redis održava bolje performanse u smislu propusnosti, vremena odgovora i istodobnih operacija u ovim iteracijama u usporedbi s PostgreSQL-om. Međutim, obje baze podataka doživljavaju povećanje vremena odgovora, vremena upita i vremena transakcije kako iteracije napreduju.

Određivanje je li bolje koristiti PostgreSQL ili Redis ovisi o konkretnom slučaju upotrebe i prioritetima za aplikaciju ili sustav.

1. **Za potrebe visoke propusnosti i malog vremena odziva:**
 - Ako se aplikacija uvelike oslanja na visoku propusnost za operacije čitanja i pisanja, dok zahtijeva nisko vrijeme odziva, Redis bi mogao biti bolji izbor. Dosljedno pokazuje veću propusnost i relativno niža vremena odgovora kroz iteracije u usporedbi s PostgreSQL-om.
2. **Transakcijsko intenzivne aplikacije:**
 - Ako aplikacija uključuje brojne transakcije s velikim brojem čitanja ili pisanja, obje baze podataka pokazuju povećanje vremena transakcije tijekom iteracija. Međutim, Redis održava nešto bolje performanse u ovim scenarijima.
3. **Izvedba složenog upita:**
 - Ako aplikacija uključuje složene upite ili podatke koji zahtijevaju opsežno postavljanje upita, PostgreSQL bi mogao imati izazove jer pokazuje strmiji porast vremena upita u usporedbi s Redisom.
4. **Rukovanje istodobnim operacijama:**
 - Redis pokazuje trend smanjenja istovremenog vremena, što potencijalno ukazuje na učinkovitije rukovanje istodobnim operacijama. PostgreSQL, međutim, pokazuje fluktuacije koje naginju prema povećanju istodobnog vremena.
5. **Ukupna stabilnost i dosljednost:**
 - Redis pokazuje stabilnije metrike performansi kroz iteracije u usporedbi s PostgreSQL-om. Nastoji zadržati ili neznatno poboljšati svoju metriku, dok PostgreSQL doživljava stalan pad propusnosti i povećanje vremena odgovora.

Razmatranja:

- **Model podataka i struktura:** PostgreSQL nudi bogate SQL mogućnosti i relacijska je baza podataka, dok je Redis pohrana ključ-vrijednost. Bolje je razmotriti koji model podataka bolje odgovara zahtjevima aplikacije.
- **Složenost operacija:** Ako aplikacija uključuje složene manipulacije podacima, relacijske upite i transakcije, PostgreSQL bi mogao biti prikladniji unatoč trendovima izvedbe.
- **Skalabilnost:** Redis je često hvaljen zbog svoje lakoće skalabilnosti i performansi u distribuiranim sustavima. Ako je skalabilnost značajna briga, Redis bi mogao biti od prednosti.

- **Upornost:** Redis podržava postojanost, ali prvenstveno naglašava izvedbu u memoriji. Ako je postojanost podataka uz dobru izvedbu ključna, PostgreSQL-ova pohrana i trajnost na disku možda bi bolje odgovarali.
- **Predmemoriranje u odnosu na integritet podataka:** Redis se obično koristi za predmemoriju zbog svoje brzine, dok PostgreSQL osigurava integritet podataka i dosljednost svojom usklađenošću s ACID-om.

U konačnici, izbor između Redisa i PostgreSQL-a ovisi o specifičnim zahtjevima u pogledu performansi, modela podataka, skalabilnosti, postojanosti i prirodi operacija koje će aplikacija izvoditi. Obje baze podataka imaju svoje jake i slabe strane, stoga je bitno uskladiti ih s potrebama projekta kako biste donijeli informiranu odluku.