

Ante Javor<sup>(1)</sup>Ivan Sekovanić<sup>(2)</sup>Tomislav Adamović<sup>(3)</sup>Viktor Matraković<sup>(4)</sup>

(1) Veleučilište u Bjelovaru,  
Trg Eugena Kvaternika 4,  
HR-43000 Bjelovar  
[ajavor@vub.hr](mailto:ajavor@vub.hr)

(2) Veleučilište u Bjelovaru,  
Trg Eugena Kvaternika 4,  
HR-43000 Bjelovar  
[isekovanic@vub.hr](mailto:isekovanic@vub.hr)

(3) Veleučilište u Bjelovaru,  
Trg Eugena Kvaternika 4,  
HR-43000 Bjelovar  
[tadamovic@vub.hr](mailto:tadamovic@vub.hr)

(4) Veleučilište u Bjelovaru,  
Trg Eugena Kvaternika 4,  
HR-43000 Bjelovar  
[vmatrakovic@vub.hr](mailto:vmatrakovic@vub.hr)

Zaprimljeno / Received

15. veljače 2026. / 15 February 2026

Prihvaćeno / Accepted

18. veljače 2026. / 18 February 2026

Autor za korespondenciju /

Corresponding author

Ante Javor

[ajavor@vub.hr](mailto:ajavor@vub.hr)

DOI: 10.70856/p.3.1.1



## Usporedba performansi operacija obilaska i agregacije svojstva u bazama podataka

**Sažetak:** U ovom radu provedena je empirijska usporedba performansi operacija obilaska grafa i agregacije svojstava čvorova na različitim sustavima baza podataka. Operacije obilaska predstavljaju temeljne operacije nad graf strukturama te izravno utječu na primjenjivost graf baza u analitičkim i interaktivnim sustavima. Evaluirani su obilasci prve, druge i treće razine nad sintetičkim grafom zvjezdane strukture koji sadrži 1,1 milijun čvorova i bridova. Analiza se temelji na mjerenju latencije izvršavanja upita, distribucije latencija i vršne potrošnje memorije. Rezultati pokazuju značajne razlike u performansama između analiziranih arhitektura, pri čemu nativne graf baze podataka ostvaruju niže i stabilnije latencije, dok Memgraph postiže najbolje rezultate pri dubljim razinama obilaska grafa.

**Ključne riječi:** graf baze podataka, operacije obilaska, operacije agregacije, Memgraph, baze podataka

## 1. Uvod

Graf baze podataka predstavljaju specijaliziranu klasu sustava koji su dizajnirani za učinkovitu pohranu, pretraživanje i analizu međusobno povezanih podataka modeliranih u obliku grafa. Za razliku od relacijskih baza podataka, koje analizu povezanih podataka provode putem operacija spajanja tablica i agregacije nad relacijama, graf baze koriste nativne mehanizme pohrane i obrade čvorova i bridova, optimizirane za izravni prolazak kroz odnose između entiteta [1].

Učinkovitost operacija obilaska grafa (engl. traversal) ima ključnu ulogu u primjenjivosti graf baza podataka u stvarnim sustavima. Neki od njih su analiza društvenih mreža, otkrivanje prijevera, sustavi za preporuke i grafove znanja. Navedeni produkcijski sustavi oslanjaju se na višestruke i često duboke obilaske grafa. Performanse ovih operacija izravno utječu na odzivnost, propusnost i skalabilnost sustava [2]. Uz samu operaciju obilaska, analitički graf upiti često uključuju i agregacijske operacije nad svojstvima čvorova i bridova, čime se dodatno povećava zahtjevnost izvršavanja upita.

Razvoj graf baza podataka doveo je do pojave različitih arhitekturnih pristupa, uključujući nativne graf baze, multimodalne sustave te relacijske baze koje graf operacije podržavaju kao dodatne značajke. Navedeni sustavi značajno se razlikuju u načinu pohrane podataka, izvršavanju upita i optimizacijama, što rezultira različitim performansnim karakteristikama kod specifičnih upita nad povezanim podacima.

Unatoč rastućem interesu za graf baze podataka, vidljivom kroz industrijske rang-liste poput DB-Engines Rankinga [3] i klasifikacije sustava u repozitorijima poput dbdb.io [4], neovisne i metodološki kontrolirane usporedbe performansi različitih arhitekturnih pristupa i dalje su relativno rijetke. Većina javno dostupnih usporedbi dolazi iz marketinških materijala proizvođača, dok empirijske studije temeljene na istim radnim opterećenjima i jedinstvenim metrikama predstavljaju značajan doprinos tehničkoj zajednici [5].

Cilj je ovog rada empirijski usporediti performanse operacija obilaska grafa i agregacije svojstava čvorova na različitim sustavima baza podataka. Poseban naglasak stavljen je na utjecaj dubine obilaska na latenciju izvršavanja upita kod graf baza, distribuciju latencija (engl. latency), te memorijski otisak sustava tijekom izvršavanja. Navedeni upiti predstavljaju temeljne blokove složenijih graf analiza ili analiza povezanih podataka.

Kako bi se osigurala neutralnosti evaluacije, većina analiziranih sustava anonimizirana je, uz opis njihovih ključnih arhitekturnih karakteristika [6]. Iznimku čini Memgraph, graf baza podataka temeljena primarno na memorijskoj pohrani, koja predstavlja središnji predmet analize.

## 2. Povezane studije

Istraživanje performansi baza podataka često se oslanja na standardizirane industrijske testove i sustavne analize arhitekturnih pristupa. Jedan od najznačajnijih napora u tom području predstavlja rad Linked Data Benchmark Councila (LDBC), koji je razvio industrijski prihvaćene testove za evaluaciju sustava nad povezanim podacima. Među najpoznatijim primjerima nalaze se Social Network Benchmark (SNB) i Graphalytics, koji definiraju kompleksna i realistična opterećenja za evaluaciju baza podataka [6].

Iako navedeni testovi omogućuju sveobuhvatnu analizu performansi, njihova primjena zahtijeva značajnu infrastrukturnu pripremu, složene procedure generiranja podataka te velike hardverske resurse. Zbog toga su ovakva testiranja često ograničena na industrijske ili akademske projekte s dostupnim resursima, a rezultati nisu uvijek jednostavno reproducibilni u manjim eksperimentalnim okruženjima.

Suprotno tome, ovaj rad primjenjuje lakši i kontroliraniji eksperimentalni pristup, usmjeren na temeljne operacije obilaska grafa različitih dubina te agregaciju svojstava čvorova. Time se omogućuje izolirana analiza osnovnih performansnih karakteristika sustava uz manji volumen podataka i jednostavniju reprodukciju eksperimenata.

Angles i Gutierrez [7] proveli su sveobuhvatan pregled modela graf baza podataka, identificirajući ključne razlike u podatkovnim modelima i arhitekturama koje izravno utječu na performanse graf upita. Njihov rad pruža teorijsku osnovu za razumijevanje kompromisa između različitih pristupa pohrani i izvršavanju graf operacija.

Besta et al. [5][8] dodatno su proširili ovu perspektivu analizom sustava za pohranu i obradu grafova, klasificirajući ih prema arhitekturi, strategijama izvršavanja upita i optimizacijama za različite vrste graf analiza, te naglašavaju važnost arhitekturnih odluka u kontekstu skalabilnosti i performansi.

U odnosu na navedene studije, ovaj rad nadopunjuje postojeću literaturu empirijskom evaluacijom aktualnih verzija sustava baza podataka, s fokusom na osnovne operatore obilaska i agregacije. Iako je evaluacija ograničena na manji skup upita, analizirani obrasci predstavljaju

temeljne blokove složenijih graf analiza, čime su dobiveni rezultati relevantni za širi raspon graf primjena.

### 3. Metodologija

U cilju osiguravanja objektivne, ponovljive i metodološki utemeljene usporedbe performansi graf baza podataka, eksperimentalno okruženje, način izvođenja testova i korištene metrike precizno su definirani. U nastavku su opisani svi relevantni aspekti metodologije. Izvorni kod i rezultati koji obuhvaćaju cijelu metodologiju dostupni su kod autora na zahtjev.

#### 3.1. Testno okruženje

Svi testovi provedeni su na računalnom sustavu s operacijskim sustavom macOS na Apple Silicon arhitekturi M2 Pro [10].

Testirani sustavi pokretani su unutar Docker [11] kontejnerskog okruženja. Korištenje kontejnera osigurava izolaciju procesa, konzistentnost runtime okruženja te jednake uvjete izvođenja za sve baze podataka uključene u test.

#### 3.2. Sustavi na testu

Evaluirano je pet različitih sustava baza podataka koje predstavljaju različite karakteristike u rješavanju problema povezanih podataka te su vidljivi u tablici 3.1.

Tablica 3.1: Pregled karakteristika baza podataka u testu [4], [5], [7].

Naziv sustava	Podatkovni model	Arhitektura pohrane	Kratki opis arhitekture
Memgraph	Label property graph	Memorijska (in-memory) uz perzistenciju	Nativna graf baza podataka s primarnim izvršavanjem u memoriji i opcionalnom perzistencijom podataka
Graf baza (hibridna)	Label property graph	Hibridna (memorija + disk)	Nativna graf baza podataka koja kombinira memorijsku i diskovnu pohranu radi uravnoteženja performansi i kapaciteta
Graf baza (u memoriji)	Label property graph	Memorijska (in-memory)	Nativna graf baza podataka s potpunim zadržavanjem grafa u memoriji i opcionalnom perzistencijom podataka

<i>Multimodalna baza podataka</i>	<i>Multimodalni</i>	<i>Diskovna</i>	<i>Multimodalna baza podataka koja podržava dokumente, objekte i graf model podataka</i>
<i>Relacijska baza podataka</i>	<i>Relacijski / objektno-relacijski</i>	<i>Diskovna</i>	<i>Objektno-relacijska baza podataka koja graf operacije ostvaruje pomoću rekurzivnih CTE upita</i>

### 3.3. Podaci u testu

Za potrebe evaluacije generiran je sintetički graf zvjezdane strukture. Skup podataka sadrži jedan centralni čvor (pivot) s identifikatorom *o*, koji predstavlja polazišnu točku za sve operacije obilaska grafa.

Graf je organiziran u tri razine dubine. Prva razina (R1) sastoji se od 10 000 čvorova izravno povezanih s centralnim čvorom. Svaki čvor u razini R1 posjeduje jedinstveni identifikator *id* te svojstva *first\_name*, *last\_name* i *account\_balance*. Nadalje, svaki čvor u razini R1 povezan je s 10 potomaka, čime se formira druga razina (R2) koja ukupno sadrži 100 000 čvorova. Analogno tome, svaki čvor u razini R2 povezan je s dodatnih 10 potomaka, čime se formira treća razina (R3), koja predstavlja najširu razinu grafa i ukupno sadrži 1 000 000 čvorova. Izuzev centralnog čvora, razine su podijeljene u eksponente broja 10, pa je rast eksponenta linearan, dok je rast broja čvorova eksponencijalan. Navedeni rast između razina predstavlja predvidive eksponencijalne skokove u količini podataka za obradu koji će se odraziti i na samim latencijama.

#### Statistika podataka u testu:

- **Ukupan broj čvorova:** 1 110 001  
(1 pivot + 10 000 R1 + 100 000 R2 + 1 000 000 R3)
- **Ukupan broj bridova:** 1 110 000  
(svaki čvor, osim pivot čvora, povezan je jednim usmjerenim bridom bez svojstava)
- **Svojstva čvorova:** *id, first\_name, last\_name, account\_balance, node\_type*

Odabrani zvjezdasti uzorak grafa omogućuje determinističko ponašanje upita te eliminira strukturnu varijabilnost kao faktor utjecaja na performanse. Time se omogućuje precizno mjerenje utjecaja dubine obilaska grafa na latenciju izvršavanja i potrošnju memorije, uz jasnu i ponovljivu interpretaciju rezultata.

### 3.4. Upiti

Svi upiti računaju sumu varijable `account_balance` preko dostupnih čvorova od pivot čvora. Upiti ovakvog tipa predstavljaju karakteristične operacije izračuna agregatnog utjecaja čvorova na određenoj razini.

U programskom kodu 3.1. prikazani su upiti u Memgraph bazi podataka:

```
#### Suma razine 1
```

```
MATCH (start:Person {id: 0})-[:CONNECTED_TO]->(neighbor:Person)
```

```
RETURN SUM(neighbor.account_balance) AS total_account_balance
```

```
#### Suma razine 2
```

```
MATCH (start:Person {id: 0})-[:CONNECTED_TO*1..2]->(neighbor:Person) RETURN
```

```
SUM(neighbor.account_balance) AS total_account_balance
```

```
#### Suma razine 3
```

```
MATCH (start:Person {id: 0})-[:CONNECTED_TO*1..3]->(neighbor:Person)
```

```
RETURN SUM(neighbor.account_balance) AS total_account_balance
```

*Programski kod 3.1: Upiti u Memgraph bazi podataka*

Za sve testirane sustave napravljeni su indeksi na svojstvima čvora `id`. Time se osigurava da se mjerenja temelje na realističnim uvjetima uporabe te da rezultati ne odražavaju razlike u performansama uzrokovane neindeksiranim pristupom osnovnim identifikatorima čvorova.

### 3.5. Prikupljanje metrike

Svaki upit izvršen je kroz 100 iteracija na bazi podataka. Višestruko ponavljanje upita smanjuje utjecaj nedeterminističkih faktora, uključujući optimizacije tijekom izvođenja, zagrijavanje baze i pozadinske sistemske aktivnosti. Rezultati testa temelje se na agregiranim statističkim vrijednostima dobivenim iz svih iteracija.

Primarna je metrika performansi latencija, definirana kao ukupno vrijeme izvršavanja upita izraženo u milisekundama. Latencija predstavlja ključni pokazatelj odzivnosti sustava, osobito u kontekstu interaktivnih i real-time aplikacija temeljenih na graf bazama podataka.

Osim prosječne latencije, analizirana je i distribucija vremena izvršavanja, uključujući minimalnu vrijednost, medijan, p50, p90, p95, p99 percentil, maksimalnu vrijednost, aritmetičku sredinu i standardnu devijaciju. Analiza distribucije omogućuje uvid u varijabilnost performansi te identifikaciju rijetkih, ali potencijalno problematičnih, sporih izvršavanja koja prosječne vrijednosti često ne odražavaju i skrivaju.

Potrošnja memorije praćena je korištenjem vršne potrošnje memorije procesa koja pokreće bazu i predstavlja ukupni trošak izvršavanja operacije, dok je memorija diska zanemarena.

Vršna memorija definirana je? kao maksimalna vrijednost rezidentnog skupa procesa (RSS) tijekom izvršavanja upita, izražena u megabajtima. Ova metrika ključna je za procjenu skalabilnosti i učinkovitosti upravljanja memorijom, budući da visoka vršna potrošnja može ograničiti paralelizam i povećati infrastrukturne troškove.

Kako bi se osigurala ekvivalentnost izvršavanja upita, provedena je validacija rezultata upita, pri čemu je verificirana ispravnost ukupnog account balance iznosa. Ova provjera osigurava da izmjerene razlike u performansama nisu posljedica nekorektnog ili nepotpunog izvođenja upita, već stvarnih razlika u implementaciji i optimizaciji sustava. Time se osiguralo da je svaki sustav za pohranu pročitao i agregirao vrijednost s čvora.

#### 4. Rezultati mjerenja

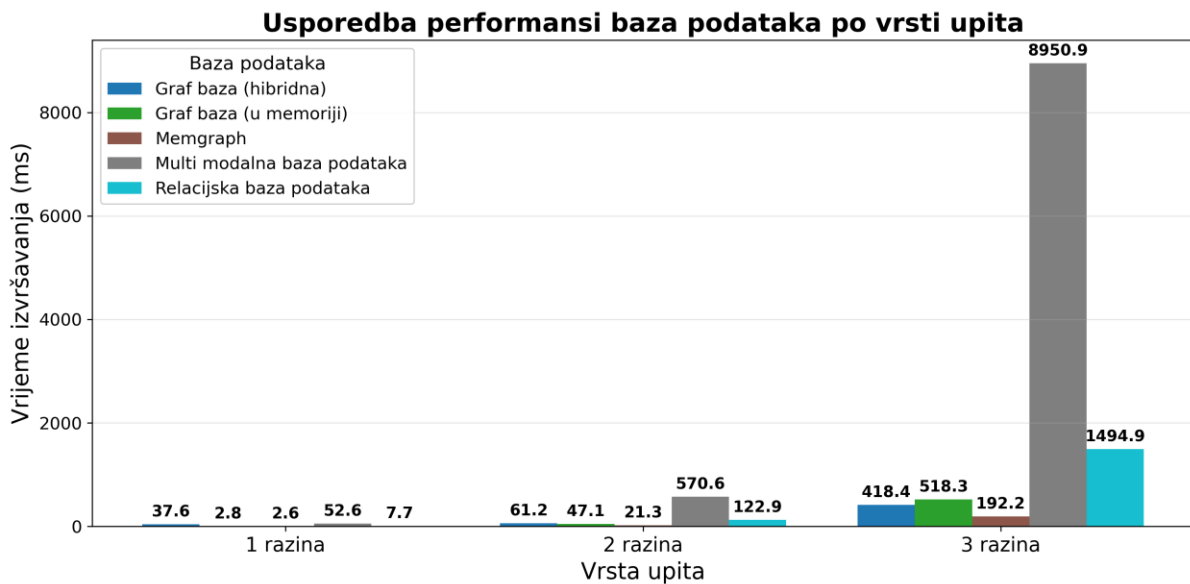
Tablica 4.1. sadrži sve rezultate mjerenja na jednom mjestu izraženim u milisekundama. Iz tablice je vidljivo kako određeni sustavi skaliraju operacije obilaska i agregacije po različitim razinama.

Tablica 4.1: Latencije i statistike vremena izvršavanja upita za sve testirane sustave baza podataka i tipove upita u milisekundama (ms).

Baza podataka	Tip upita	Min (ms)	p50 (ms)	p90 (ms)	p95 (ms)	p99 (ms)	Max (ms)	Pros. (ms)
Memgraph	R1	1,33	1,62	3,41	8,94	16,63	17,62	2,61
Memgraph	R2	15,44	16,62	25,68	30,83	116,67	135,33	21,33
Memgraph	R3	154,82	166,43	206,26	262,98	754,08	765,06	192,23
Graf baza (u memoriji)	R1	1,65	1,89	2,22	2,59	3,87	83,75	2,77

Graf baza (u memoriji)	R <sub>2</sub>	43,32	46,37	50,61	52,74	58,89	75,90	47,15
Graf baza (u memoriji)	R <sub>3</sub>	460,50	506,61	560,01	584,63	684,53	852,11	518,30
Graf baza (hibridna)	R <sub>1</sub>	5,02	16,01	58,13	162,76	462,39	546,89	37,60
Graf baza (hibridna)	R <sub>2</sub>	34,45	42,88	95,94	110,87	214,36	630,86	61,21
Graf baza (hibridna)	R <sub>3</sub>	320,17	355,95	523,78	684,54	1493,05	1514,55	418,41
Multimodalna baza podataka	R <sub>1</sub>	43,03	47,29	58,20	74,21	108,52	267,61	52,64
Multimodalna baza podataka	R <sub>2</sub>	462,97	542,05	671,14	731,77	998,79	1053,72	570,58
Multimodalna baza podataka	R <sub>3</sub>	6845,21	8517,10	10532,04	11883,05	14950,96	16787,39	8950,90
Relacijska baza podataka	R <sub>1</sub>	5,74	7,18	8,91	9,48	18,59	22,82	7,70
Relacijska baza podataka	R <sub>2</sub>	109,04	118,31	127,68	136,05	200,71	393,26	122,89
Relacijska baza podataka	R <sub>3</sub>	1201,01	1417,52	1757,62	1824,66	2856,38	4194,41	1494,90

Postoji vidljiv skok u latencijama na upitu R<sub>3</sub>. Slika 4.1. stavlja u fokus samo stupac prosječnih vrijednosti latencija.

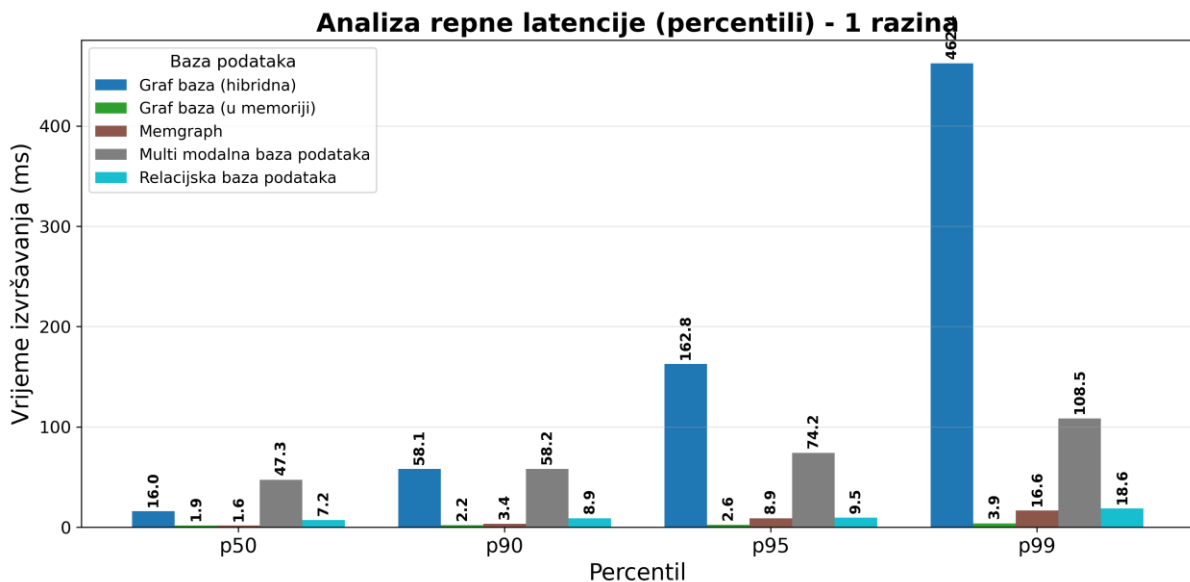


*Slika 4.1: Prosječna latencija po sustavu i upitu*

#### 4.1. Utjecaj dubine obilaska na repnu latenciju

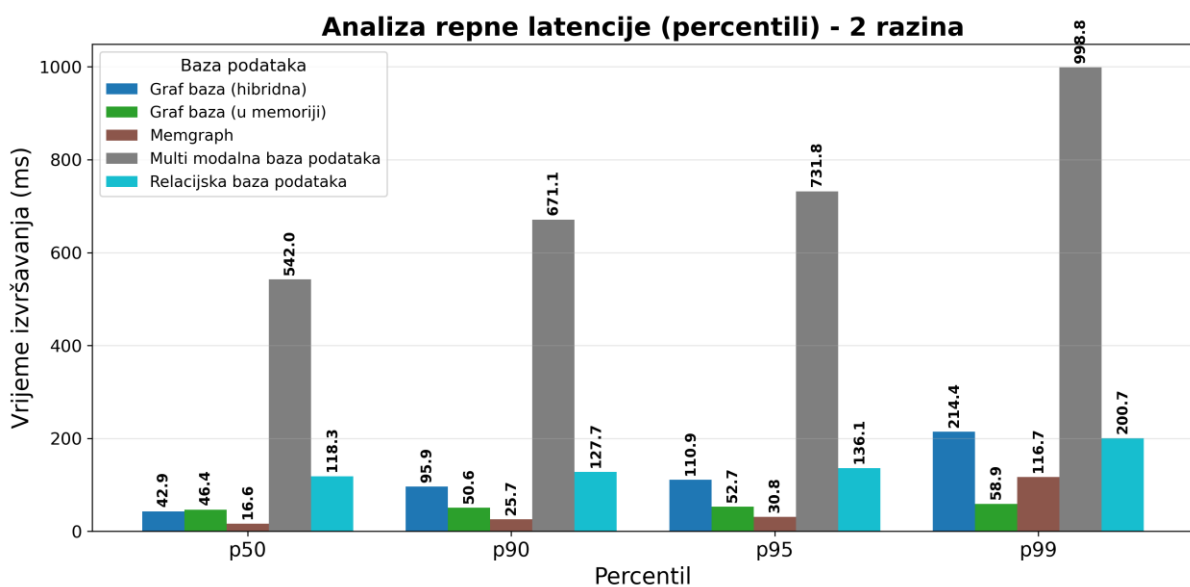
Repna latencija (engl. tail latency) otkriti će sve spore upite koji se ne vide na prosječnim vrijednostima, te su često skriveni, a mogu bitno utjecati na responzivnost sustava. Slika 4.2. prikazuje repnu latenciju za upite obilaska R<sub>1</sub> za sve sustave. Rezultati pokazuju da native graf baze podataka temeljene na memorijskoj pohrani postižu najniže latencije i najmanju varijabilnost vremena izvršavanja. Ovakvo ponašanje očekivano je s obzirom na korištenje izravnih pokazivača između čvorova i izbjegavanje skupih operacija spajanja.

Kod drugih baza uočava se veća varijabilnost latencija, osobito u višim percentilima (p<sub>95</sub> i p<sub>99</sub>). Multimodalna i relacijska baza pokazuju znatno više latencije već pri upitima R<sub>1</sub>, što ukazuje na dodatnu kompleksnost i manje optimizirane mehanizme za obilazak grafa. Hibridna graf baza pokazuje visoke latencije na plitkom obilasku R<sub>1</sub>.



*Slika 4.2: Repna latencija na upitu R1*

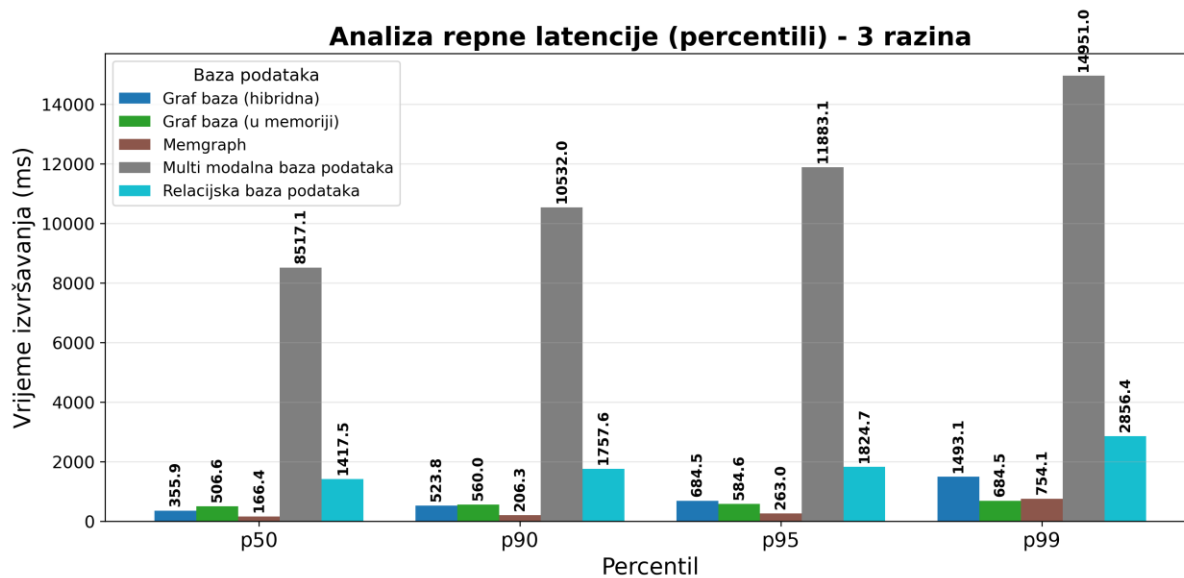
Povećanjem dubine obilaska upitom R2, na slici 4.3. razlike između sustava postaju izraženije. Dok native graf baze zadržavaju relativno linearan rast latencije, multimodalna i relacijska baza bilježe znatno strmiji porast vremena izvršavanja. Ovakav trend ukazuje na rastući trošak izvršavanja rekurzivnih ili iterativnih operacija nad graf strukturom.



*Slika 4.3: Repna latencija na upitu R2*

Kod upita R3 na slici 4.4. razlike u arhitekturi sustava dolaze do punog izražaja. Native graf baze, iako bilježe značajan porast latencije zbog eksponencijalnog rasta broja obilazaka, i dalje su

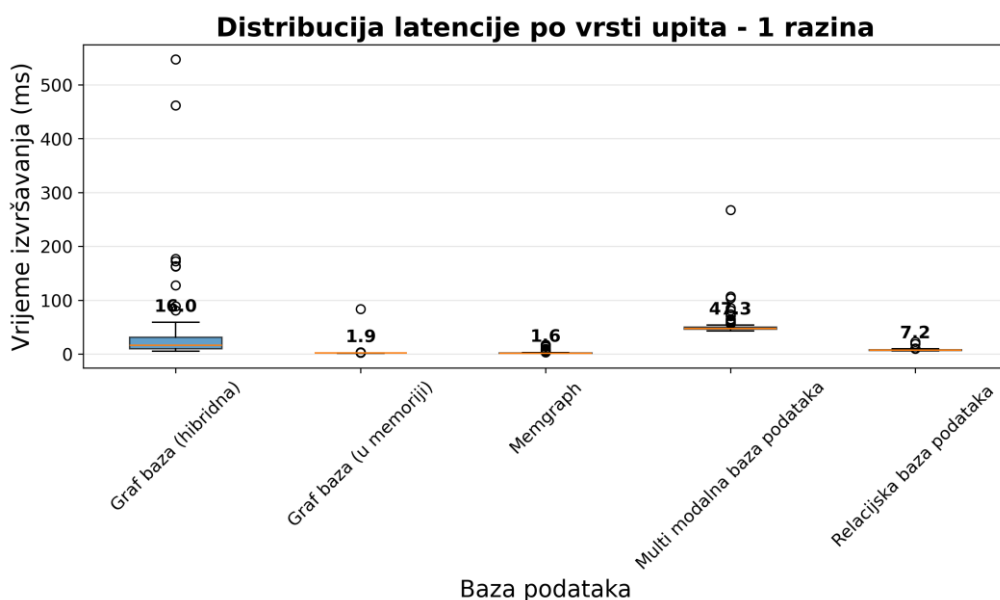
značajno brže u usporedbi s multimodalnim i relacijskim sustavima. Multimodalna baza pokazuje najvišu latenciju i najveću varijabilnost, a prati ju relacijska baza.



Slika 4.4: Repna latencija na upitu R3

#### 4.2. Distribucija latencija i stabilnost sustava

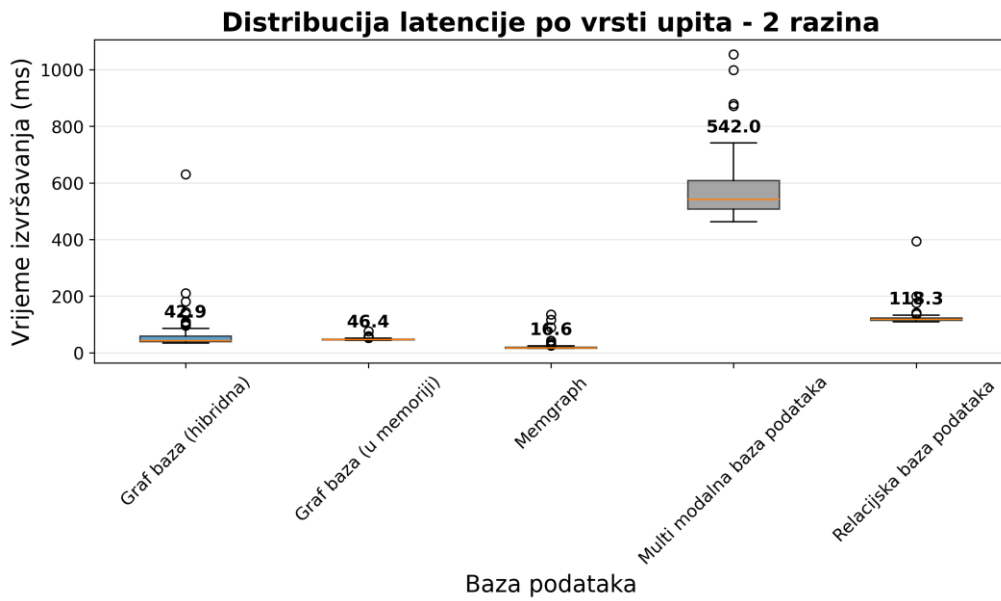
Analiza distribucije latencija otkriva važne razlike u stabilnosti izvršavanja upita. Nativne graf baze pokazuju usku distribuciju latencija s relativno malim razlikama između medijana i viših percentila. Takvo ponašanje ukazuje na predvidljivo izvršavanje i manju osjetljivost na varijacije u produkcijskom okruženju. Distribucija latencija za upite R1 vidljiva je na slici 4.5.



Slika 4.5: Distribucija latencije R1

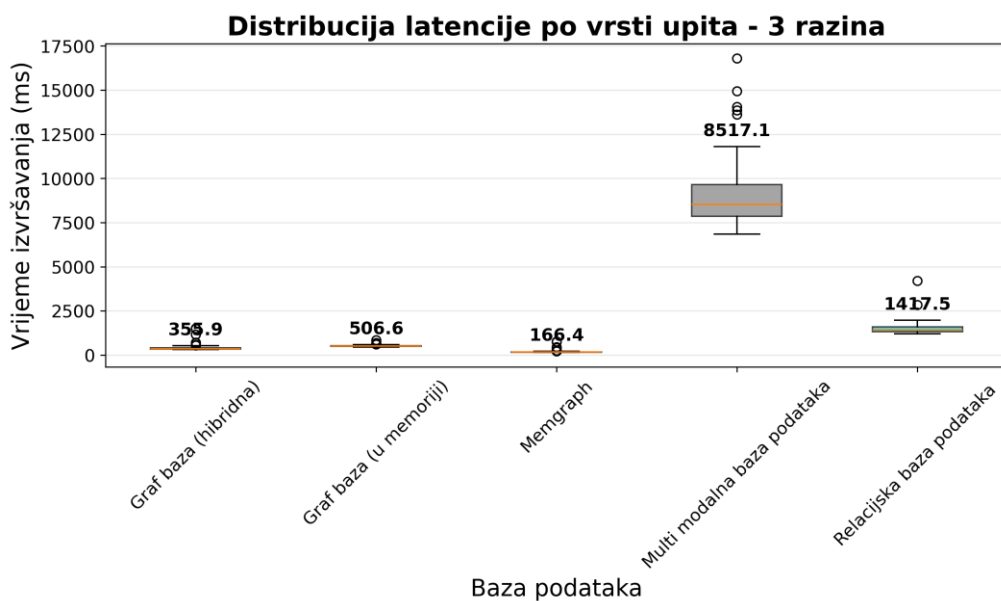
Iz slike je vidljivo kako na upitu R1 najveću varijabilnost distribucije ima hibridna graf baza, međutim taj upit nije toliko zahtjevan nad samom bazom.

Slika 4.6. pokazuje varijabilnost distribucije za upit R2. Na navedenoj slici vidljivo je kako se varijabilnost distribucije latencija kod graf baza značajno smanjila.



Slika 4.6: Distribucija latencije R2

Kod upita R3 na slici 4.7. moguće je primijetiti kako varijabilnost latencija multimodalne i relacijske baze podataka raste.



Slika 4.7: Distribucija latencije R3

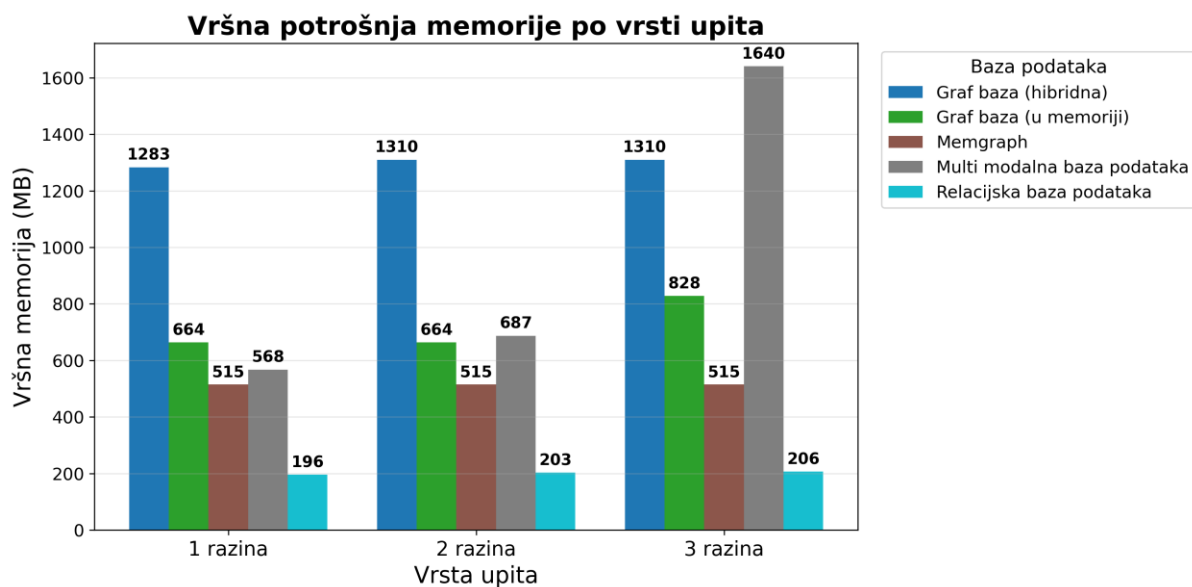
Multimodalna i hibridna graf baza pokazuju znatno šire distribucije, osobito kod R2 i R3 upita. Visoke vrijednosti p<sub>99</sub> i maksimalnih latencija sugeriraju povremena spora izvršavanja, koja u produkcijskim sustavima mogu negativno utjecati na korisničko iskustvo i ukupnu propusnost sustava.

#### 4.3. Memorijski otisak sustava

Slika 4.8. prikazuje vršnu potrošnju memorije tijekom izvršavanja upita različitih dubina. Nativne graf baze koje se temeljene na memorijskoj pohrani očekivano koriste veći inicijalni memorijski otisak, ali pokazuju stabilno ponašanje tijekom izvršavanja upita, bez naglih memorijskih skokova.

Hibridne i multimodalna baza pokazuju veće fluktuacije u potrošnji memorije, što se može pripisati internim strukturama za međuspremanje (engl. buffering), i privremene podatke tijekom izvršavanja obilaska. Multimodalna baza pokazuje značajan rast memorijske potrošnje s povećanjem dubine obilaska, dok relacijska baza bilježi relativno nižu potrošnju memorije, ali uz znatno veće latencije izvršavanja.

Ovi rezultati potvrđuju postojanje jasnog kompromisa između latencije i memorijskog otiska, pri čemu sustavi optimizirani za graf operacije koriste više memorije kako bi postigli niže vrijeme izvršavanja.



Slika 4.8: Vršna potrošnja memorije po bazi podataka i tipu upita

## 5. Zaključak

U ovom radu provedena je empirijska usporedba performansi operacija obilaska grafa i agregacije svojstava čvorova na sustavima baza podataka koji predstavljaju različite arhitekturne pristupe radu s graf strukturama. Evaluacija je provedena u kontroliranom i ponovljivom okruženju, koristeći sintetički graf zvjezdane strukture koji omogućuje izoliranu analizu utjecaja dubine obilaska na latenciju izvršavanja i memorijski otisak sustava.

Rezultati mjerenja pokazuju da arhitektura sustava ima ključan utjecaj na performanse graf upita. Nativne graf baze podataka, osobito one temeljene na memorijskoj pohrani, dosljedno postižu najniže latencije i najmanju varijabilnost vremena izvršavanja kroz sve razine obilaska. Povećanjem dubine obilaska razlike između sustava postaju sve izraženije, pri čemu multimodalni i relacijski sustavi bilježe znatno strmiji porast latencije u odnosu na native graf baze.

Analiza distribucije latencija pokazuje da native graf baze pružaju stabilnije i predvidljivije ponašanje, dok hibridni i multimodalni sustavi pokazuju veću varijabilnost i povremena spora izvršavanja, osobito kod dubljih obilazaka. Rezultati memorijske analize potvrđuju postojanje kompromisa između potrošnje memorije i latencije, gdje sustavi optimizirani za graf operacije koriste veći memorijski otisak kako bi postigli niže vrijeme izvršavanja.

Doprinos ovog rada ogleda se u metodološki utemeljenoj i reproduktivnoj empirijskoj evaluaciji temeljnih graf operacija na aktualnim sustavima baza podataka. Iako je evaluacija provedena na sintetičkom grafu pravilne strukture, analizirani obrasci obilaska i agregacije predstavljaju osnovne blokove složenijih graf upita. Složeniji graf upiti često će imati kombinacije više koraka obilaska i agregacija, što će dovesti do sličnih rezultata u produkcijskim sustavima.

Zaključno, rezultati potvrđuju da su native graf baze podataka najprikladniji izbor za sustave koji zahtijevaju česte i duboke operacije obilaska grafa uz agregaciju podataka, dok alternativni arhitekturni pristupi predstavljaju kompromis između fleksibilnosti, memorijskog otiska i performansi [1], [5], [7].

## 6. Popis literature

1. Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., & Vrgoč, D. (2017). Foundations of modern query languages for graph databases. *ACM Computing Surveys*, 50(5), 1–40. <https://doi.org/10.1145/3104031>
2. Sahu, S., Mhedhbi, A., Salihoglu, S., Lin, J., & Özsu, M. T. (2017). The ubiquity of large graphs and surprising challenges of graph processing. *Proceedings of the VLDB Endowment*, 11(4), 420–431. <https://doi.org/10.14778/3184470.3184475>
3. DB-Engines. (2024). DB-Engines ranking. <https://db-engines.com/en/ranking>
4. Pramod, M., Pavlo, A., et al. (2024). dbdb.io: A catalog of databases and their architectural characteristics. <https://dbdb.io>
5. Besta, M., et al. (2023). Demystifying graph databases: Analysis and taxonomy of data organization, system designs, and graph queries. *ACM Computing Surveys*, 56(2), 1–40. <https://doi.org/10.1145/3561039>
6. Erling, O., et al. (2015). The LDBC social network benchmark: Interactive workload. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 619–630). <https://doi.org/10.1145/2723372.2742786>
7. Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, 40(1), 1–39. <https://doi.org/10.1145/1322432.1322433>
8. Besta, M., et al. (2023). Practice of streaming processing of dynamic graphs: Concepts, models, and systems. *IEEE Transactions on Parallel and Distributed Systems*, 34(6), 1860–1876. <https://doi.org/10.1109/TPDS.2023.3258491>
9. Memgraph. (2024). Memgraph documentation. <https://memgraph.com/docs>
10. Microsoft. (2024). Recursive common table expressions (CTE) in SQL. <https://learn.microsoft.com/en-us/sql/t-sql/queries/with-common-table-expression-transact-sql>
11. Apple Inc. (2024). Apple silicon overview. Apple Developer Documentation. <https://developer.apple.com/documentation/apple-silicon>
12. Docker Inc. (2024). Docker engine overview. Docker documentation. <https://docs.docker.com/engine/>

## PERFORMANCE COMPARISON OF TRAVERSAL AND PROPERTY AGGREGATION OPERATIONS IN DATABASES

**Abstract:** *This paper presents a performance comparison of graph traversal and property aggregation operations across different database systems. Graph traversal operations represent fundamental building blocks of graph processing and have a direct impact on the applicability of graph databases in analytical and interactive workloads. Traversals of one-hop, two-hop, and three-hop depth were evaluated using a synthetic star-shaped graph consisting of 1.1 million nodes and edges. The evaluation focuses on query execution latency, latency distribution, and peak memory consumption. The results reveal significant performance differences among the evaluated architectures, showing that native graph databases achieve lower latency and more stable execution, with Memgraph demonstrating the best performance for deeper graph traversals.*

**Keywords:** *graph databases, traversal operations, aggregation operations, Memgraph, databases*