

Primerjava vzporednega programiranja v  
programskih jezikih C, Julia in Golang za reševanje  
NP-polnih problemov  
DISPOZICIJA DIPLOMSKE NALOGE

Gabrijel Okorn  
go7745@student.uni-lj.si

MENTOR: doc. dr. Boštjan Slivnik  
Fakulteta za računalništvo in informatiko  
Univerza v Ljubljani

25. julij 2024

### **Povzetek**

V diplomski nalogi bom raziskoval vzporedno programiranje, ki omogoča učinkovitejšo uporabo večprocesorskih sistemov, kar je ključno pri reševanju kompleksnih problemov.

Posvetil se bom trem programskim jezikom. To so C, Julia in Golang. Primerjal jih bom tako, da se bom poglobil v konstrukte in knjižnice, ki so temu namenjene. Med seboj jih bom primerjal tudi ob implementiranju treh NP-polnih problemov: Partitioning Problem z eno veliko množico, Partitioning Problem z več majhnimi množicami[1] in algoritmom Fruchterman Reingold[2].

Na podlagi analize rezultatov bom pripravil smernice in priporočila za programerje, ki se ukvarjajo z vzporednim programiranjem v teh jezikih, ter predloge za nadaljnje raziskave in optimizacije.

## **1 Motivacija za izbrano diplomsko temo**

Izhodišče za predlagano diplomsko temo je potreba po učinkovitem reševanju kompleksnih problemov z uporabo vzporednega programiranja. Omogoča nam uporabo več procesorskih jeder hkrati, kar lahko znatno pohitri reševanje problemov, ki so sicer časovno zelo zahtevni. Diplomaska naloga bo torej bralcu predstavila, kako se ti jeziki razlikujejo glede na učinkovitost, enostavnost uporabe ter porabo sistemskih virov.

V diplomski nalogi se bom sicer nanašal na lastne implementacije rešitev za NP-polne probleme. Vseeno pa to ne bo konkretni cilj diplomske naloge. To

bodo le primeri oziroma stranska nit s pomočjo katere si bom pomagal pisati razlago. Konkretni cilj, ki ga bo reševala moja diplomska naloga bo pomoč ob izbiri jezika in njegovih knjižnic ob reševanju pripadajočih problemov. Diplomska naloga bo demonstrirala uporabo različnih metod vzporednega programiranja, vključno z uporabo knjižnic in jezikovnih konstruktov. Med njimi so OpenMP za C, vgrajene funkcije za vzporedno programiranje v Juliji in Goroutines v Golangu. Naloga bo tudi analizirala, kako te metode vplivajo na učinkovitost reševanja problemov.

Pri reševanju problemov vzporedno programiranje zahteva boljše poznavanje delovanja programa v ozadju, kar delo dodatno oteži. Tako bo diplomska naloga programerjem s smernicami in razlago zagotavljala izboljšanje načina vzporednega programiranja. Prav tako bo pomagala akademikom, ki preučujejo učinkovitost različnih programskih jezikov in metod za vzporedno programiranje.

## **1.1 Pregled področja in sorodnih del**

Na področju vzporednega programiranja so bila izdana številna pomembna dela, ki obravnavajo optimizacijo algoritmov za večjedrne sisteme in uporabo specifičnih jezikovnih konstruktov za izboljšanje učinkovitosti.

Najbolj sorodna raziskava je gotovo diplomsko delo Janeza Pintarja z naslovom Programski jeziki za vzporedno programiranje[3]. V tem delu se je avtor posvetil primerjanju raziskovalnih programskih jezikov Chapel, Julia in X10. Jezike je med seboj primerjal iz dveh vidikov. Iz kvantitativnega vidika je meril čas izvajanja programov napisanih v teh jezikih. Naknadno je predstavil zahtevana predznanja za jezik in obrazložil v katerih primerih je uporaba le tega jezika primerna.

Moj mentor je soavtor knjige "Introduction to parallel computing: from algorithms to programming on state-of-the-art platforms"[4]. Skupaj z ostalimi avtorji so se osredotočili na metode OpenMP, MPI in OpenCL, ki omogočajo optimizacijo računalniške zmogljivosti na večjedrnih računalnikih, povezanih sistemih in GPU-jih. Knjiga poudarja pomembnost učinkovite zasnove, analize in programiranja vzporednih algoritmov za doseganje visoko zmogljivih računalniških operacij.

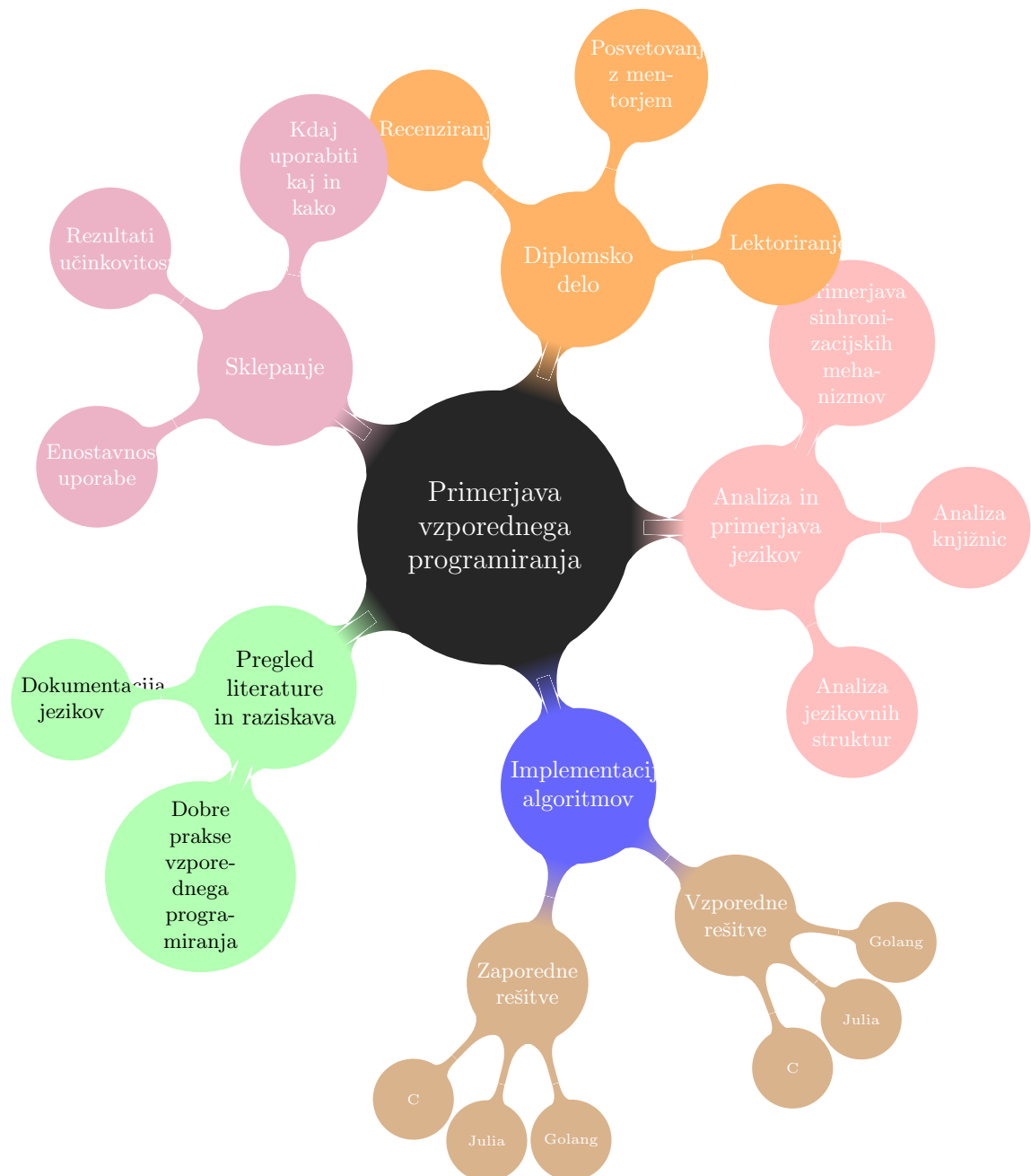
## **1.2 Zakaj je predlagani mentor primeren**

Doc. dr. Boštjan Slivnik je priznan strokovnjak na področju programskih jezikov. Prav tako pa je tudi nosilec predmetov Programiranje 2 in Prevaljniki. Tudi sam sem bil udeleženec pri teh predmetih. Profesor nam je poglobil znanje o ozadju izvajanja programov v različnih programskih jezikih in njihovi implementaciji.

V preteklosti je bil mentor študentom, ki so se v diplomskem delu posvetili

vzporednem programiranju.

## 2 Miselni diagram za izbrano temo diplomske naloge



### 3 Predvideni prispevki diplomske naloge

Rezultat diplomske naloge bo celovita analiza vzporednega programiranja v programskih jezikih C, Julia in Golang. Na tem področju bo moja diplomska naloga v pomoč bralcu z naslednjimi prispevki:

- smernice za programerje - katere vgrajene funkcije in katere konstrukte jezika uporabiti v katere namene
- delovanje programskega jezika - vpogled v to, kako se napisana koda izvede in kako se različni mehanizmi vzporednih programov med seboj razlikujejo v ozadju izvajanja
- analiza učinkovitosti (kvantitativna merila) - primerjava časovne učinkovitosti in porabe sistemskih virov ob uporabi raznolikih pristopov

Kvantitativne rešitve, ki jih bom pridobil v namen diplomske naloge bom preveril z merjenjem časovne učinkovitosti, tako da bom merjenja izvedel večkrat in na istem računalniku in bom tako zagotovil koherentne in zanesljive rezultate.

Nekaj več dela pa bom imel pri analitični primerjavi jezikovnih konstruktov in vgrajenih funkcij. Tu namreč rezultati ne bodo kvantitativni in zanje ne bo kriterija na katerega bi se lahko zanesel, da bi ocenil pristnost mojih rešitev. V ta namen se bom potrudil in kar se da zanesljivo raziskal dokumentacijo jezikov in drugih znanstvenih prispevkov ter v skladu z njimi poskusil bralcu predati kar se da korektno znanje.

### 4 Uporabljena metodologija

Ob izdelavi diplomske naloge bom za zgled implementiral nekaj algoritmov, pri čemer bom uporabil:

- OpenMP - knjižnica za vzporedno programiranje v jeziku C
- Distributed.jl - knjižnica za vzporedno programiranje v Julia jeziku, ki omogoča enostavno distribucijo nalog med več procesorskih jeder
- Goroutines - orodje za vzporedno programiranje v Golangu

Testne podatke bom pridobil na internetu ali iz strokovnih del. Moji NP-polni problemi so relativno znani, zato bom lahko našel pristne testne primere in rešitve.

### 5 Razdelitev potrebnega dela na aktivnosti

#### 1. Raziskava in načrtovanje

- (a) Pregled literature in sorodnih del (3 dni)
  - (b) Študija jezikovnih konstruktov in knjižnic (5 dni)
  - (c) Raziskava na področju sinhronizacijskih mehanizmov (3 dni)
2. **Implementacija zaporednih rešitev** (1 teden)  
*Pred to aktivnostjo je potrebno dokončati raziskavo in načrtovanje.*
3. **Implementacija vzporednih rešitev** (1 teden)  
*Pred to aktivnostjo je potrebno dokončati implementacijo zaporednih rešitev.*
4. **Testiranje in zbiranje podatkov** (1 dan)
- (a) Priprava testnih podatkov
  - (b) Večkratno izvajanje testov in zbiranje podatkov
  - (c) Merjenje časovne učinkovitosti
- Pred to aktivnostjo je potrebno dokončati implementacijo vzporednih rešitev.*
5. **Analiza rezultatov**
- (a) Analiza porabe sistemskih virov (1 dan)
  - (b) Primerjava rezultatov med jeziki (1 teden)
- Pred to aktivnostjo je potrebno dokončati testiranje in zbiranje podatkov.*
6. **Dokumentacija in pisanje diplomske naloge** (2 tedna) - *pisanje uvoda, pregleda literature, primerjave, rezultatov in zaključka*  
*Pred to aktivnostjo je potrebno dokončati vse ostale aktivnosti.*

## 6 Preliminarno kazalo

1. **Uvod** - *pomen vzporednega programiranja in opis ciljev diplomske naloge in hipoteze, ki jih bo raziskava preverila.*
2. **Programski jeziki** - *opis vsakega od jezikov in analiza jezikovnih orodij in konstruktov ter njihovih knjižnic*
  - (a) **C, Julia, Golang**
3. **Primerjava** - *primerjava osnovnih jezikovnih struktur in knjižnic: primerjanje in ananliziranje vsakega jezika z vsakim.*
  - (a) **Osnovne jezikovne strukture**

- (a) **Knjižnice za vzporedno programiranje**
  - i. *C: Uporaba OpenMP za vzporedno programiranje.*
  - ii. *Julia: Distributed.jl in drugi orodja za vzporedno izvajanje.*
  - iii. *Golang: Standardne knjižnice za goroutine in sinhronizacijo.*
- (b) **Sinhronizacijski mehanizmi**
  - i. *C: Uporaba semaforjev, mutexov in drugih mehanizmov.*
  - ii. *Julia: Pregled sinhronizacijskih orodij, kot so kanali in dogodki.*
  - iii. *Golang: Kanali in druge vgrajene možnosti za sinhronizacijo.*
- (c) **Enostavnost uporabe**
  - i. *Primerjava kompleksnosti programiranja v vsakem jeziku.*
  - ii. *Krivulja učenja za programerje pri prehodu med jeziki.*
- (d) **Učinkovitost izvajanja**
  - i. *Primerjava časa prevajanja in izvajanja programov.*
  - ii. *Analiza performančnih razlik med jeziki pri različnih tipih nalog.*

#### 4. Rezultati

- (a) **Primernost sinhronizacijskih mehanizmov** - opredelitev mehanizmov, ki so primerni za reševanje določenih problemov.

#### 5. Sklepne ugotovitve - povzetek ključnih ugotovitev in priporočil za uporabnike obravnavanih jezikov in orodij

#### 6. Literatura - seznam uporabljenih virov, pripravljen z BibLaTEX.

## Literatura

- [1] *Partitioning Problem*. 2024. URL: [https://en.wikipedia.org/wiki/Partition\\_problem](https://en.wikipedia.org/wiki/Partition_problem) (pridobljeno 2. 10. 2024).
- [2] Tomihisa Kamada, Satoru Kawai in sod. »An algorithm for drawing general undirected graphs«. V: *Information processing letters* 31.1 (1989), str. 7–15.
- [3] Janez Pintar. »Programski jeziki za vzporedno programiranje«. Diplomsko naloga. Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2019.
- [4] Roman Trobec in sod. *Introduction to parallel computing: from algorithms to programming on state-of-the-art platforms*. Springer, 2018.