



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ




Лука Петковић

**Симулација двоструког клатна и
анализа перформанси паралелних
имплементација**

ДИПЛОМСКИ РАД

Основне академске студије

Нови Сад, 2025

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Број:
	ЗАДАТАК ЗА ЗАВРШНИ РАД	Датум:

(Податке уноси предметни наставник - ментор)

Студијски програм:	Софтверско инжењерство и информационе технологије		
Студент:	Лука Петковић	Број индекса:	SV 16/2021
Степен и врста студија:	Основне академске студије		
Област:	Електротехничко и рачунарско инжењерство		
Ментор:	Игор Дејановић		
НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ЗАВРШНИ РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА: <ul style="list-style-type: none"> - проблем – тема рада; - начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна; 			

НАСЛОВ ЗАВРШНОГ РАДА:

Симулација двоструког клатна и анализа перформанси паралелних имплементација
--

ТЕКСТ ЗАДАТКА:

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
--

Руководилац студијског програма:	Ментор рада:

Примерак за: <input type="checkbox"/> - Студента; <input type="checkbox"/> - Ментора
--



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:

Идентификациони број, **ИБР**:

Тип документације, **ТД**:

Монографска документација

Тип записа, **ТЗ**:

Текстуални штампани материјал

Врста рада, **ВР**:

Дипломски - бечелор рад

Аутор, **АУ**:

Лука Петковић

Ментор, **МН**:

Др Игор Дејановић, редовни професор

Наслов рада, **НР**:

Симулација двоструког клатна и анализа перформанси паралелних имплементација

Језик публикације, **ЈП**:

српски/ћирилица

Језик извода, **ЈИ**:

српски/енглески

Земља публикавања, **ЗП**:

Република Србија

Уже географско подручје, **УГП**:

Војводина

Година, **ГО**:

2025

Издавач, **ИЗ**:

Ауторски репринт

Место и адреса, **МА**:

Нови сад, трг Доситеја Обрадовића 6

Физички опис рада, **ФО**:

(поглавља/страна/ цитата/табела/слика/графика/прилога)

9/21/4/0/6/0/2

Научна област, **НО**:

Електротехничко и рачунарско инжењерство

Научна дисциплина, **НД**:

Примењене рачунарске науке и информатика

Предметна одредница/Кључне речи, **ПО**:

Двоструко клатно, паралелна обрада, Python, Rust, Go

УДК

Чува се, **ЧУ**:

У библиотеци Факултета техничких наука, Нови Сад

Важна напомена, **ВН**:

Извод, **ИЗ**:

Рад представља симулацију двоструког клатна и анализу перформанси паралелних имплементација. Коришћена је метода Рунге–Кута четвртог реда за нумеричко решавање једначина. Имплементације у Python, Rust и Go језицима упоређене су по времену извршавања, убрзању и скалабилности, уз анализу предности сваког приступа.

Датум прихватања теме, **ДП**:

Датум одбране, **ДО**:

01.01.2025

Чланови комисије, **КО**:

Председник: Др Петар Петровић, ванредни професор

Члан: Др Марко Марковић, доцент

Члан, ментор: Др Игор Дејановић, редовни професор

Потпис ментора



UNIVERSITY OF NOVI SAD • FACULTY OF TECHNICAL SCIENCES
21000 NOVI SAD, Trg Dositeja Obradovića 6

KEY WORDS DOCUMENTATION

Accession number, **ANO**:
Identification number, **INO**:
Document type, **DT**: Monographic publication
Type of record, **TR**: Textual printed material
Contents code, **CC**:
Author, **AU**: Luka Petković
Mentor, **MN**: Igor Dejanović, Phd., full professor
Title, **TI**: Double pendulum simulation with performance comparison of parallel implementations

Language of text, **LT**: Serbian
Language of abstract, **LA**: Serbian
Country of publication, **CP**: Republic of Serbia
Locality of publication, **LP**: Vojvodina
Publication year, **PY**: 2025
Publisher, **PB**: Author's reprint
Publication place, **PP**: Novi Sad, Dositeja Obradovica sq. 6
Physical description, **PD**: 9/21/4/0/6/0/2
(chapters/pages/ref./tables/pictures/graphs/appendixes)
Scientific field, **SF**: Electrical and Computer Engineering
Scientific discipline, **SD**: Applied computer science and informatics
Subject/Key words, **S/KW**: double pendulum, parallel processing, Python, Rust, Go

UC

Holding data, **HD**: The Library of Faculty of Technical Sciences, Novi Sad, Serbia

Note, **N**:

Abstract, **AB**: This paper presents a simulation of the double pendulum and a performance comparison of parallel implementations. The fourth-order Runge–Kutta method was used for numerical integration. Implementations in Python, Rust, and Go were evaluated by execution time, speedup, and scalability, highlighting each language's advantages.

Accepted by the Scientific Board on, **ASB**:

Defended on, **DE**: 01.01.2025

Defended Board, **DB**: President: Petar Petrović, Phd., assoc. professor
Member: Marko Marković, Phd., asist. professor
Member, Mentor: Igor Dejanović, Phd., full professor

Menthor's sign



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

ИЗЈАВА О НЕПОСТОЈАЊУ СУКОБА ИНТЕРЕСА

Изјављујем да нисам у сукобу интереса у односу ментор – кандидат и да нисам члан породице (супружник или ванбрачни партнер, родитељ или усвојитељ, дете или усвојеник), повезано лице (крвни сродник ментора/кандидата у правој линији, односно у побочној линији закључно са другим степеном сродства, као ни физичко лице које се према другим основама и околностима може оправдано сматрати интересно повезаним са ментором или кандидатом), односно да нисам зависан/на од ментора/кандидата, да не постоје околности које би могле да утичу на моју непристрасност, нити да стичем било какве користи или погодности за себе или друго лице било позитивним или негативним исходом, као и да немам приватни интерес који утиче, може да утиче или изгледа као да утиче на однос ментор-кандидат.

У Новом Саду, дана _____

Ментор

Кандидат

Садржај

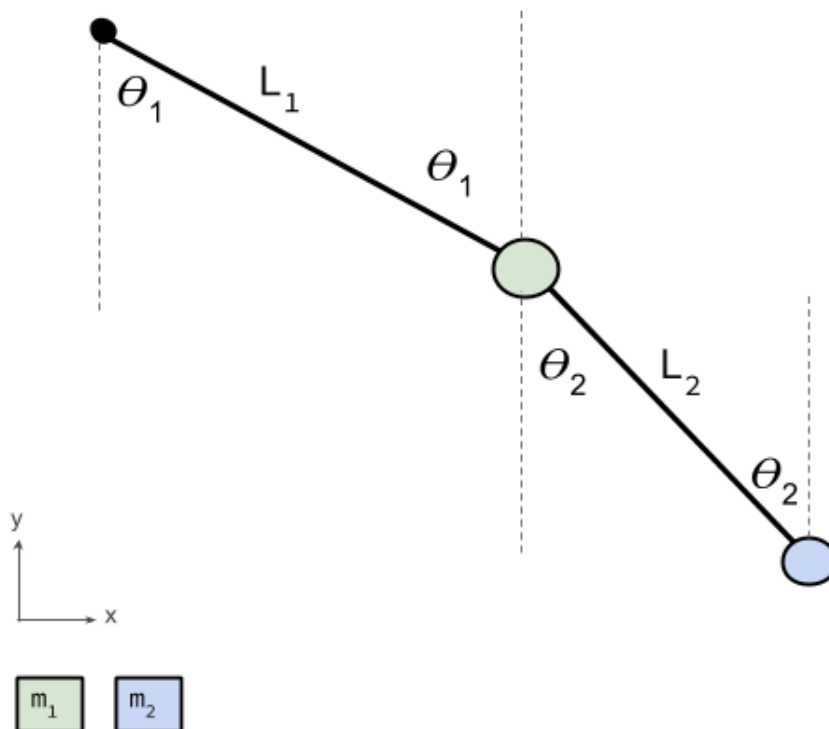
1	Увод	1
1.1	Шта је двоструко клатно?	1
1.2	Циљ и структура рада	1
2	Теоријске основе	3
2.1	Хаотични системи	3
2.2	Физички модел двоструког клатна	3
2.3	Нумеричке методе интеграције	3
2.4	Паралелизација и модели скалирања	5
3	Закључак	9
	Списак слика	11
	Списак листинга	13
	Списак коришћених скраћеница	15
	Списак коришћених појмова	17
	Биографија	19
	Литература	21

1.1 Шта је двоструко клатно?

Двоструко клатно је физички систем који се састоји од два међусобно повезана клатна. Први сегмент је причвршћен за фиксну тачку, док је други окачен на крај првог сегмента. Кретање система описује комбинацију кретања оба клатна, при чему се сваки сегмент помера под утицајем гравитације и сила које проистичу из њихове међусобне везе.

Овакав систем има четири степена слободе – два угла и две угаоне брзине – што доводи до сложене динамике. Када се енергија система повећа, двоструко клатно почиње да показује хаотично понашање, односно изузетну осетљивост на почетне услове. Иако се ради о класичном механичком систему, његово понашање није предвидиво на дуге стазе, па се често користи као илустрација **хаотичних система** у физици и нумеричкој анализи.

Стање система у сваком тренутку може се описати четворком вредности: $(\theta_1, \omega_1, \theta_2, \omega_2)$ где су θ_1 (тета 1) и θ_2 (тета 2) углови сегмената у односу на вертикалу, а ω_1 (омега 1) и ω_2 (омега 2) њихове угаоне брзине. Ове променљиве су међусобно повезане системом нелинеарних диференцијалних једначина које немају аналитичко решење, па се решавање врши **нумеричким методама**.



Слика 1: Основни дијаграм двоструког клатна

1.2 Циљ и структура рада

Основни циљ овог рада јесте развој и анализа система за симулацију двоструког клатна у више програмских језика, са посебним освртом на поређење перформанси паралелних имплементација. Истражују се

разлике у брзини извршавања, ефикасности и скалабилности између имплементација у језицима *Python*, *Rust* и *Go (Golang)*, као и утицај различитих модела паралелизма на резултате симулације.

Рад има за циљ да:

- демонстрира примену нумеричких метода на хаотичном систему,
- анализира ефекте паралелне обраде код различитих програмских језика,
- упореди брзину и ефикасност имплементација,
- истакне практичне предности сваког језика у области научних симулација.

Структура рада организована је на следећи начин. У другом поглављу описане су теоријске основе система двоструког клатна, укључујући физички модел, једначине кретања и примењену нумеричку методу. Треће поглавље приказује дизајн решења и концепт ансамбл-паралелизације. Четврто поглавље садржи опис имплементација у *Python*, *Rust* и *Go* језицима, са детаљним анализама јаких и слабих скалинга. У петом поглављу дато је упоредно поређење резултата и дискусија. На крају, у шестом поглављу налази се закључак са прегледом резултата и предлогом за даљи рад.

2.1 Хаотични системи

Хаотични системи представљају класу динамичких система чије се понашање карактерише изузетном осетљивошћу на почетне услове. Иако се њихова еволуција управља детерминистичким једначинама, и најмања промена у почетним параметрима доводи до драстично различитих резултата. Овај феномен је познат као **ефекат лептира** (*butterfly effect*) и чини да такви системи делују непредвидиво на дуже стазе.

Двоструко клатно је један од најпознатијих примера хаотичног система у класичној механици. Упркос томе што се заснива на једноставним законима физике, његово кретање може бити веома сложено, са путањама које се ретко понављају. Овакав систем је од посебног значаја у нумеричкој анализи, јер омогућава тестирање стабилности, тачности и перформанси различитих метода интеграције.

2.2 Физички модел двоструког клатна

Двоструко клатно се састоји од два сегмента дужина l_1 и l_2 , са масама m_1 и m_2 . Први сегмент је окачен о фиксну тачку, док је други сегмент окачен на крај првог. Положај система описује се угловима θ_1 и θ_2 у односу на вертикалу. Систем ћемо посматрати у две димензије због једноставности, а сем тога, занема-рићемо све стране силе попут силе отпора ваздуха или силе трења.

Енергија система се састоји од кинетичке (T) и потенцијалне (V) енергије:

$$E = T + V$$

Кинетичка енергија:

$$T = \frac{1}{2}m_1(l_1\omega_1)^2 + \frac{1}{2}m_2[(l_1\omega_1)^2 + (l_2\omega_2)^2 + 2l_1l_2\omega_1\omega_2\cos(\theta_1 - \theta_2)]$$

Први члан се односи на кинетичку енергију горњег клатна. Други члан се односи на кинетичку енергију доњег клатна које укључује:

- сопствено кретање
- кретање због ротације горњег сегмента
- међусобни унакрсни члан који настаје јер се вектори брзина не поклапају у правцу

Потенцијална енергија:

$$V = -m_1gl_1\cos(\theta_1) - m_2g[l_1\cos(\theta_1) + l_2\cos(\theta_2)]$$

Потенцијална енергија се рачуна у односу на најнижи положај. Први члан се односи на висину масе m_1 , а други на висину масе m_2 која зависи од оба угла. Знак минус стоји испред јер се енергија система смањује кад тела иду надоле.

Ове једначине немају аналитичко решење, па се у пракси примењују нумеричке методе.

2.3 Нумеричке методе интеграције

Када кажемо да систем „нема аналитичко решење“, то значи да не постоји затворени математички израз који би омогућио да се вредности променљивих добију директним рачунањем. Другим речима, функције које описују кретање тела не могу се изразити у облику једноставних формула (попут синуса, косинуса или експоненцијала), већ се решење мора приближно израчунати.

Због тога се у пракси користе **нумеричке методе интеграције**, које омогућавају приближно решавање система диференцијалних једначина у малим временским корацима. Оне омогућавају да се за познато

почетно стање система — почетне углове (θ_1, θ_2) и угаоне брзине (ω_1, ω_2) — израчунају нове вредности у наредним корацима времена ($t + \Delta t$).

Једноставне методе, попут Euler-ове, често производе велике грешке јер користе само једну процену нагиба функције у датом кораку. То доводи до акумулације грешке и нестабилности током дужих симулација, посебно код хаотичних система попут двоструког клатна.

Да би се добила већа тачност без већег повећања сложености, у овом раду примењује се **метода Рунге–Кута четвртог реда (RK4)**. Ова метода представља компромис између једноставности имплементације и нумеричке стабилности. Основна идеја је да се вредност функције у следећем кораку одређује као пондерисана средина више „процена“ нагиба унутар истог временског интервала.

Ако је општа форма једначине:

$$\frac{dy}{dt} = f(t, y)$$

онда се алгоритам RK4 дефинише на следећи начин:

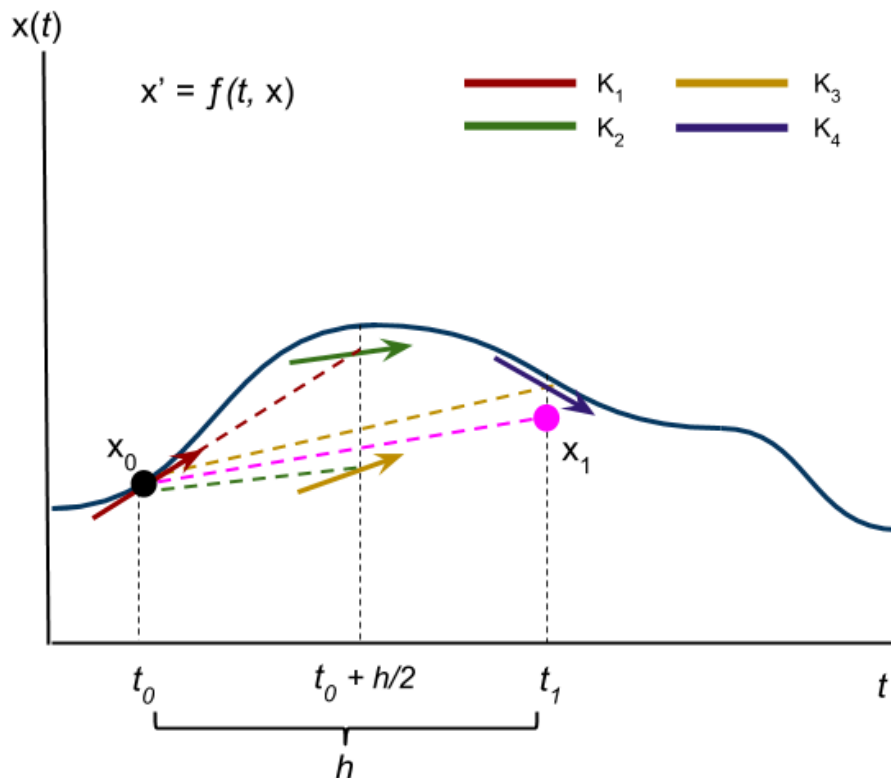
$$k_1 = f(t, y)$$

$$k_2 = f\left(t + \frac{1}{2}h, y + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(t + \frac{1}{2}h, y + \frac{1}{2}hk_2\right)$$

$$k_4 = f(t + h, y + hk_3)$$

$$y_{\{n+1\}} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4)$$



Слика 2: Приказ Runge–Kutta (RK4) метода

На слици је приказано како се у сваком кораку интеграције израчунавају четири различите процене нагиба (k_1, k_2, k_3, k_4) које заједно дају бољу апроксимацију правог решења функције $x(t)$. Тачке $t_0, t_0 + \frac{h}{2}$ и t_1 представљају положаје у времену у којима се израчунавају међурезултати. Коначна вредност x_1 добија се као пондерисана средина ових нагиба.

Метода Рунге–Кута се показала као стабилна и ефикасна чак и код система са израженом нелинеарношћу и хаотичним понашањем, попут двоструког клатна, што је чини изузетно погодном за ову симулацију.

2.4 Паралелизација и модели скалирања

Иако је *Runge–Kutta* метода изузетно погодна за тачно нумеричко решавање система диференцијалних једначина, она по својој природи није лако паралелизујућа. Разлог је што сваки временски корак зависи од резултата претходног — да би се израчунало стање у тренутку t_{n+1} , потребно је да буде познато стање у тренутку t_n . Ово уводи секвенцијалну зависност у израчунавање, која ограничава могућност директног распоређивања задатка на више процесора.

Да би се ипак искористиле предности савремених вишејезгарних процесора, примењује се тзв. **ансамбл паралелизација**. Уместо да се једна симулација раздваја на мање делове, истовремено се покреће више независних симулација са различитим почетним условима (различите комбинације углова и угаоних брзина). На овај начин сваки процес или нит извршава потпуно независан ток израчунавања, а резултати се на крају обједињују ради анализе понашања система.

Оваква стратегија има два кључна циља:

- боље искоришћење рачунарских ресурса (сви процесорски језгри су активни);
- омогућавање статистичке анализе резултата на већем броју различитих почетних услова.

Да би се ефикасност паралелизације објективно проценила, користе се два класична модела скалирања: **јако скалирање** и **слабо скалирање**.

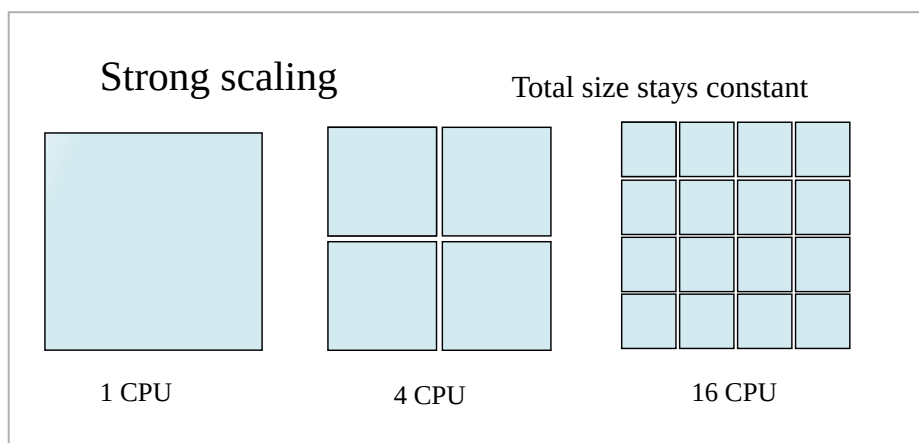
2.4.1 Јако скалирање

Јако скалирање подразумева извршавање истог задатка (фиксирани величине проблема) на различитом броју процесора. Циљ је да се измери убрзање које се постиже повећањем броја процесора:

$$S_p = \frac{T_1}{T_p}$$

где је T_1 време извршавања на једном процесору, а T_p време извршавања на p процесора.

У идеалном случају, убрзање би требало да буде линеарно ($S_p = p$), али у пракси долази до одступања услед додатних трошкова синхронизације, преноса података и управљања нитима.



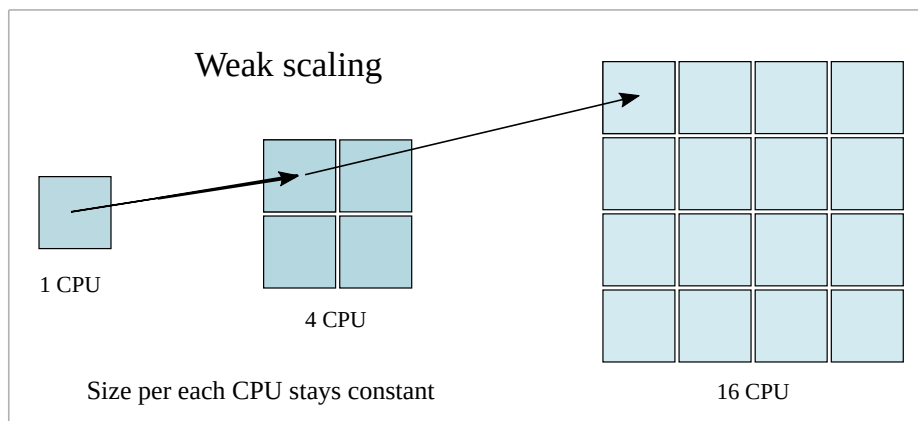
Слика 3: Јако скалирање

2.4.2 Слабо скалирање

Слабо скалирање подразумева пропорционално повећање величине проблема са бројем процесора. У овом моделу, сваки процесор има приближно исти количински део посла као у случају једног процесора. Циљ је да се време извршавања задржи приближно константно:

$$E_p = \frac{T_1}{T_p}$$

где E_p представља ефикасност система за p процесора. Добра имплементација треба да показује малу промену у времену извршавања када се величина проблема повећава у складу са бројем процесора.



Слика 4: Слабо скалирање

Ови модели биће примењени у експериментима који следе, како би се проценила скалабилност имплементација у језицима *Python*, *Rust* и *Go*.

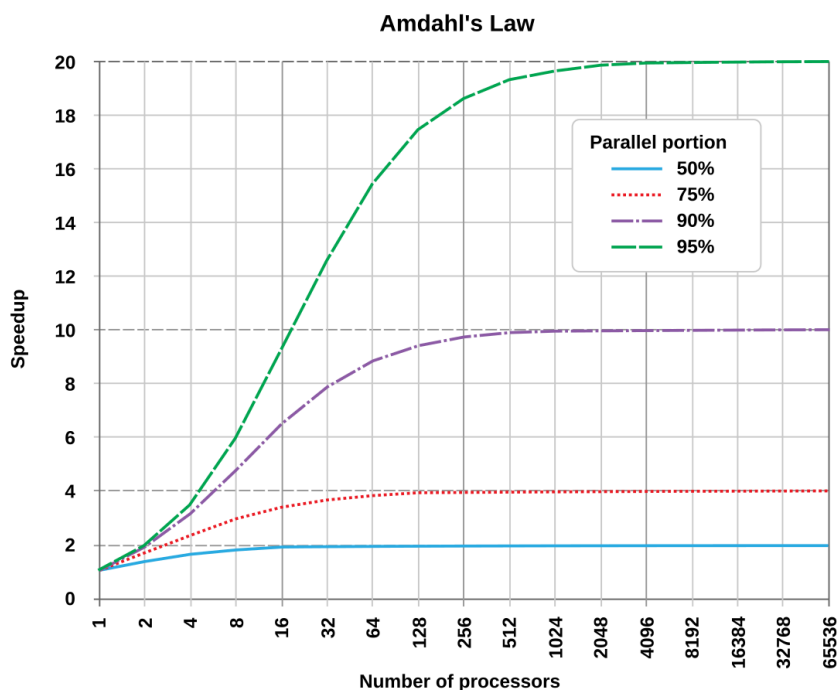
2.4.3 Амдалов и Густафсонов закон

Ефикасност паралелних програма се не може повећавати неограничено. Чак и када се број процесора повећава, део програма који се не може паралелизовати поставља природно ограничење убрзања. Ово ограничење описује **Амдалов закон**:

$$S_p = \frac{1}{(1-f) + \frac{f}{p}}$$

где је:

- S_p – убрзање које се постиже коришћењем p процесора,
- f – део програма који се може паралелизовати,
- $(1 - f)$ – серијски део програма који мора бити извршен секвенцијално.



Слика 5: Амдалов закон

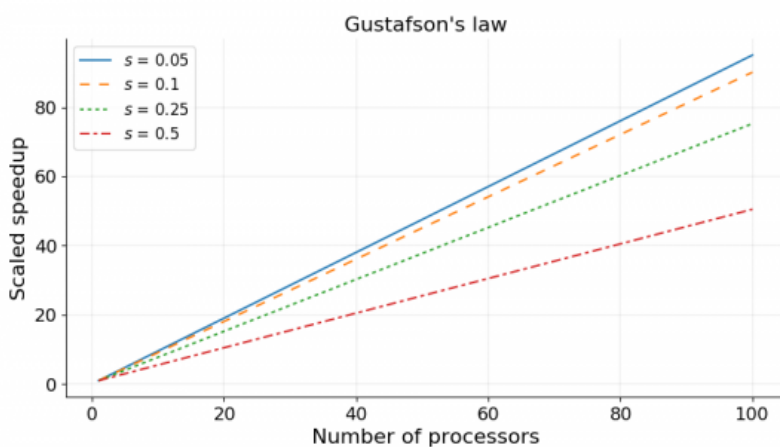
Из Амдаловог закона произилази да чак и ако је 95% програма паралелно, теоријски максимум убрзања није већи од 20×, без обзира на број процесора. Зато је у пракси важно минимизовати серијске делове програма и комуникационе трошкове између процеса.

Међутим, Амдалов модел се користи у случају фиксне величине проблема. У стварности, повећање броја процесора често омогућава решавање већих проблема у истом временском оквиру. Ову идеју описује **Густафсонов закон**:

$$S_p = p - \alpha(p - 1)$$

где је:

- S_p – постигнуто убрзање,
- p – број процесора,
- α – серијски удео извршавања (однос времена серијског дела и укупног времена)



Слика 6: Густафсонов закон

Густафсонов закон показује да се у пракси убрзање може приближити линеарном, ако се величина проблема повећава заједно са бројем процесора, што је карактеристично за **слабо скалирање**. Ово објашњење боље одражава реалне услове рада савремених система, када се повећање броја процесора користи за обраду већих количина података, а не само за брже извршавање истог посла.

Комбинацијом Амдаловог и Густафсоновог модела могуће је добити реалнију процену ефикасности паралелних система и боље разумети понашање симулација при повећању броја процесора.

Глава 3

Закључак

У закључку дајте кратак преглед онога шта урађено, са освртом на проблеме који су решени, предности и мане решења и правце даљег развоја.

Списак слика

Слика 1	Основни дијаграм двоструког клатна	1
Слика 2	Приказ Runge–Kutta (RK4) метода	4
Слика 3	Јако скалирање	5
Слика 4	Слабо скалирање	6
Слика 5	Амдалов закон	7
Слика 6	Густафсонов закон	7

Списак листинга

Списак коришћених скраћеница

Скраћеница	Опис
API	Application Programming Interface (апликациони програмски интерфејс)
AWS	Amazon Web Services (Амазон веб сервиси)
CI/CD	Continuous Integration / Continuous Delivery (континуирана интеграција / континуирана испорука)
CORS	Cross-Origin Resource Sharing (размена ресурса између извора и дестинације различитог порекла)
CSS	Cascading Style Sheets (језик за описивање стилова)
DOM	Document Object Model (објектни модел документа)
DTO	Data Transfer Object (објекат за пренос података)
HTTP	HyperText Transfer Protocol (протокол за пренос хипертекста)
JSON	JavaScript Object Notation (формат за размену података)
JWT	JSON Web Token (сигурносни токен заснован на JSON формату)
RLS	Row-Level Security (сигурност на нивоу реда)
REST	Representational State Transfer (скуп правила за комуникацију између клијента и сервера)
RPC	Remote Procedure Call (позив удаљене процедуре)
SQL	Structured Query Language (структурирани упитни језик)
TLS	Transport Layer Security (безбедност транспортног слоја)
UML	Unified Modeling Language (језик за моделовање дијаграма)
URL	Uniform Resource Locator (јединствени идентификатор и локатор ресурса)
UI	User Interface (кориснички интерфејс)
UUID	Universally Unique Identifier (универзално јединствени идентификатор)
WAL	Write-Ahead Logging (записивање операција унапред)

Списак коришћених појмова

Појам	Објашњење
θ_1	Угао првог сегмента двоструког клатна у односу на вертикалу
θ_2	Угао другог сегмента двоструког клатна у односу на вертикалу
ω_1	Угаона брзина првог сегмента клатна
ω_2	Угаона брзина другог сегмента клатна
Хаотични систем	Систем који показује осетљивост на почетне услове, што доводи до непредвидивог понашања током времена
Рунге–Кута метода	Нумеричка метода четвртог реда за решавање система диференцијалних једначина
Скалабилност	Мера способности алгоритма да ефикасно користи повећање рачунарских ресурса

Биографија

Зовем се Лука Петковић и рођен сам 14. јула 2002. годину у Новом Саду. Завршио сам Гимназију “Јован Јовановић Змај” 2021. године, смер обдарени ученици у математичкој гимназији, и добитник сам Вукове дипломе. Након тога, уписао сам Факултет Техничких наука, смер Софтверско инжењерство и информационе технологије, преко буџета. У току студирања, завршио сам тронедељну праксу у “Schneider Electric”-у, као и двомесечну праксу у компанији “Syneschron”. Положио сам све предмете основних академских студија са просеком оцена 9.36. Тренирам стони тенис.

Литература
