|  |
| --- |
| FAKULTET TEHNIČKIH NAUKA UNIVERZITET U NOVOM SADU |
| Monte Carlo simulacija |
| Izveštaj za projektni zadatak |
|  |
| **Dušan Stević** |
| **9/5/2020** |



|  |
| --- |
|  |

Sadržaj

[Motivacija problema 3](#_Toc50225305)

[Projektni zadatak 3](#_Toc50225306)

[Simulacija izračunavanja broja π 4](#_Toc50225307)

[Osnovni pojmovi 4](#_Toc50225308)

[Input/output simulacije izračunavanja broja π 4](#_Toc50225309)

[Serijski pristup simulacije izračunavanja broja π 4](#_Toc50225310)

[Paralelni pristup simulacije izračunavanja broja π 5](#_Toc50225311)

[Vizuelizacija simulacije izračunavanja broja π 5](#_Toc50225312)

[Simulacija predikcije cene finansijske aktive 5](#_Toc50225313)

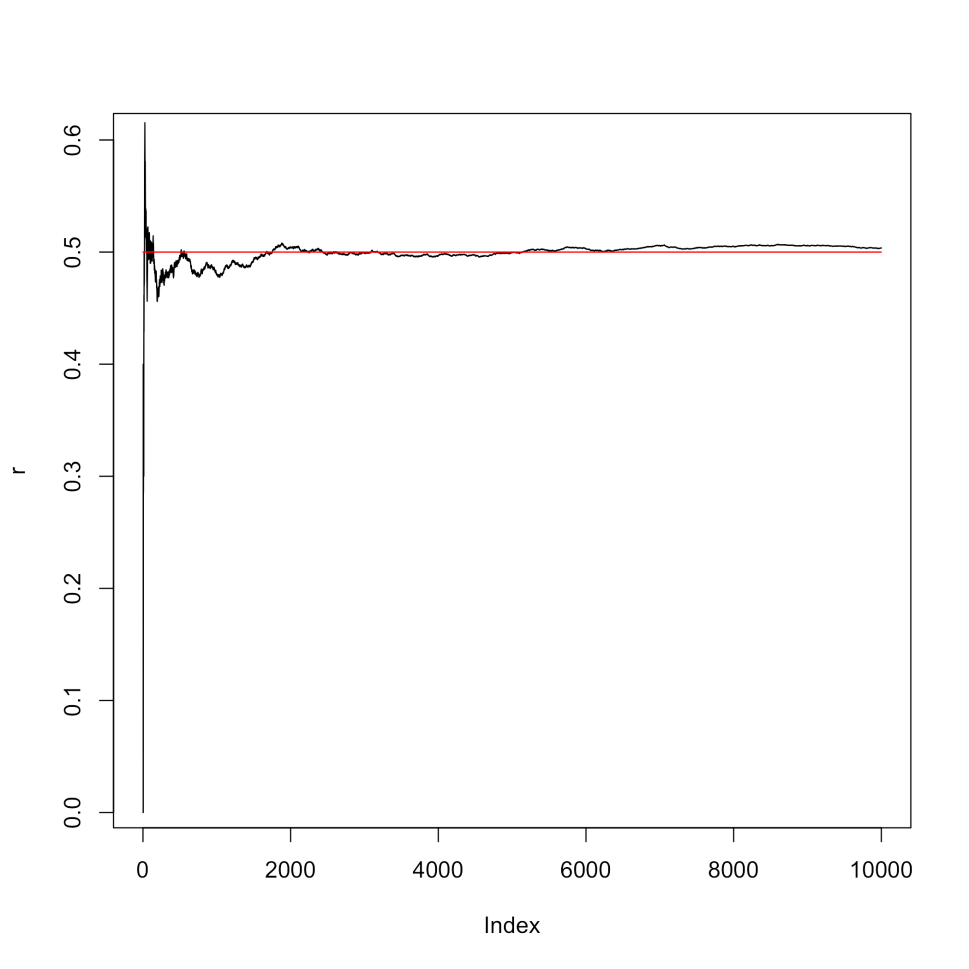
[Osnovni pojmovi 5](#_Toc50225314)

[Eksperimenti skaliranja 5](#_Toc50225315)

[Korisni linkovi i bibliografija 5](#_Toc50225316)

# Motivacija problema

Monte Carlo simulacija spada u kategoriju probabilističkih aproksimativnih tehnika za simulaciju i predikciju. Zasnovana je na [teoriji velikih brojeva](https://en.wikipedia.org/wiki/Law_of_large_numbers#:~:text=In%20probability%20theory%2C%20the%20law,a%20large%20number%20of%20times.&text=For%20example%2C%20while%20a%20casino,a%20large%20number%20of%20spins.) i [teoriji slučajnog uzrokovanja](https://en.wikipedia.org/wiki/Simple_random_sample). Osnovna ideja koja se krije iza Monte Carlo simulacije je da sa porastom broja simulacija eksperimentalna verovatnoća se približava (konvergira) teorijskoj verovatnoći (npr.kada bacamo novčić ako imamo jako puno pokušaja bacanja onda će odnos pismo glava biti pibližno 50%-50%).



Slika Teorija velikih brojeva na primeru bacanja novčića

# Projektni zadatak

[Monte Carlo simulacija](https://en.wikipedia.org/wiki/Monte_Carlo_method) ima širok dijapazon primene. U ovo radu Monte Carlo simulacija je primenjena u dve obalsti:

1. Simulacija izračunavanja broja π
2. Simulacija predikcije cene finansijske aktive (akcije, obveznice, derivati, kripto valute, itd.)

Osnovni zadatak u ovom projektu bio je razvoj serijske i paralelne verzije softverskog rešenja koje će omogućiti matematičku simulaciju izračunavanja broja π kao i simulaciju izračunavanja buduće vrednosti finansijske aktive. Serijske i paralelne verzije programa ostvarene su u [Python](https://www.python.org) i [Golang](https://golang.org) programskim jezicima. Pored razvoja serijske i paralelne verzije programa posebna pažnja je posvećena i vizuelizaciji dobijenih rešenja. Vizuelizacija rešenja zasnovana je na programskom jeziku [Pharo](https://pharo.org) uz korišćenje graphic engine-a [Roassal](http://agilevisualization.com). Takođe veoma važnu stavku u projektu čine i [eksperimenti skaliranja](https://www.kth.se/blogs/pdc/2018/11/scalability-strong-and-weak-scaling/). Kako je reč o projektu koji obuhvata paralelno programiranje i [računarstvo visokih performansi](https://en.wikipedia.org/wiki/High-performance_technical_computing) bilo je potrebno sprovesti detaljne eksperimente kako bi se utvrdio uticaj broja procesnih jedinica i veličine posla na ubrzanje paralelnog programa u odnosu na serijski program. Eksperimenti skaliranja obuhvataju eksperimente jakog i slabog skaliranja. Cilj eksperimenata jakog i slabog skaliranja je da pokažemo kako se ovi algoritmi ponašanja na stvanom hardveru.

# Simulacija izračunavanja broja π

## Osnovni pojmovi

Monte Carlo simulacija može da posluži u izračunavanju broja π. Ako imamo kvadrat i u kvadratu imamo krug odnos površina kruga i kvadrata je π/4. Ovaj odnos približno možemo da dobijemo ako površinu kvadrata "izbombardujemo" tačkama i onda napravimo odnos tačke koje su upale u krug/ukupan broj tačaka. Množenjem ovog odnosa sa 4 dobija se približna vrednost broja π koja je sve tačnija i tačnija kako povećavamo broj simulacija.

## Input/output simulacije izračunavanja broja π

1. Input: Pseudo-slučajno generisane koordinate tačaka (x i y) kojima se “bombarduje” površina kvadrata
2. Output: Aproksimirana vrednost broja π

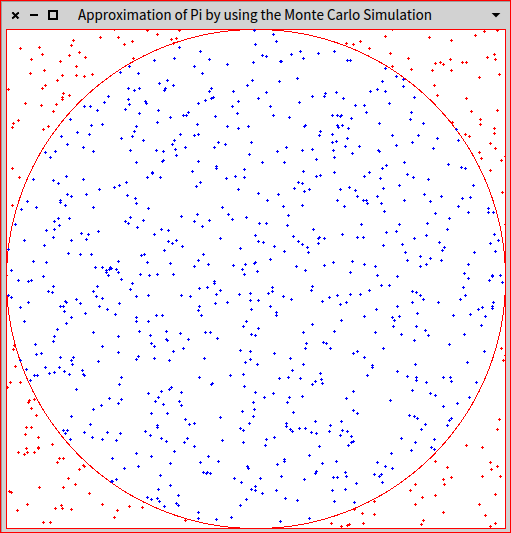
## Serijski pristup simulacije izračunavanja broja π

1. Na pseudo-slučajan način odredimo koordinatu x na interval [0,1]
2. Na pseudo-slučajan način odredimo koordinatu y na interval [0,1]
3. Dobili smo tačku sa koordinatama (x,y)
4. Radimo sa jediničnom kružnicom čiji je centar u koordinatnom početku (0,0), a radijus r = 1
5. Proveravamo da li je udaljenost novo formirane tačke manja od 1
   1. x2+y2<r2 odnosno u slučaju jedinične kružnice x2+y2<1 (tačka se nalazi unutar kružnice)
   2. x2+y2=r2 odnosno u slučaju jedinične kružnice x2+y2=1 (tačka se nalazi na kružnici)
   3. x2+y2>r2 odnosno u slučaju jedinične kružnice x2+y2>1 (tačka se nalazi van kružnice)
6. Ukoliko se tačka nalazi unutar kružnice uvećavamo brojač tačaka unutar kružnice
7. Ukupan broj simulacija predstavlja ukupan broj tačaka kojima se “bombarduje” površina kvadrata opisanog oko jedinične kružnice
8. Zbog odnosa površine kruga i površine kvadrata koja iznosi π/4 odnos broja tačaka unutar kružnice i ukupnog broja tačaka množimo sa 4 kako bismo dobili aproksimiranu vrednost broja π
9. Izračunali smo približno vrednost broja π koja je sve tačnija i tačnija kako povećavamo broj simulacija odnosno broj tačaka sa kojim “bombardujemo” površinu kvadrata

## Paralelni pristup simulacije izračunavanja broja π

1. Podelimo ukupan broj simulacija tj. ukupan broj tačaka sa kojim “bombardujemo” površinu kvadrata na npr. četvrtine
2. Svaki task obrađuje jednu četvrtinu tačaka
3. Svi taskovi se obavljaju u paraleli
4. Zbog odnosa površine kruga i površine kvadrata koja iznosi π/4 odnos broja tačaka unutar kružnice i ukupnog broja tačaka množimo sa 4 kako bismo dobili aproksimiranu vrednost broja π
5. Izračunali smo približno vrednost broja π koja je sve tačnija i tačnija kako povećavamo broj simulacija odnosno broj tačaka sa kojim “bombardujemo” površinu kvadrata

## Vizuelizacija simulacije izračunavanja broja π



Slika Vizuelizacija simulacije izračunavanja broja π u programskom jeziku Pharo n = 1000

# Simulacija predikcije cene finansijske aktive

## Osnovni pojmovi

Asdfasdfa

# Eksperimenti skaliranja

Eksperimenti jakog skaliranja

safdafsdf

Eksperimenti slabog skaliranja

Asdfasdfas

# Korisni linkovi i bibliografija

|  |  |
| --- | --- |
| **Oblast** | **Link** |
| **Python** | <http://www.igordejanovic.net/courses/tech/Python/> |
| <https://github.com/vladaindjic/ntp-2020/blob/master/napredni-python/code/konkurentno_programiranje/pregled.md> |
| <https://www.python.org> |
| <http://www.igordejanovic.net/courses/ntp/napredni-python/> |
| <https://www.youtube.com/watch?v=IEEhzQoKtQU> |
| <https://www.youtube.com/watch?v=fKl2JW_qrso&t=622s> |
| **Go** | <http://www.igordejanovic.net/courses/tech/GoLang/index.html> |
| <https://golang.org> |
| <https://www.youtube.com/watch?v=C8LgvuEBraI> |
| <https://www.youtube.com/watch?v=LvgVSSpwND8> |
| <https://www.youtube.com/watch?v=kjr3mOPv8Sk&t=6s> |
| **Pharo** | <http://www.igordejanovic.net/courses/tech/Pharo/index.html> |
| <https://pharo.org> |
| <http://agilevisualization.com> |
| <https://www.youtube.com/watch?v=-Pk4q5oMdLo> |
| <https://www.youtube.com/watch?v=iXUZiFtnxK8&t=47s> |
| **Monte Carlo** | <https://www.investopedia.com/terms/m/montecarlosimulation.asp> |
| <https://en.wikipedia.org/wiki/Monte_Carlo_method> |
| **Skaliranje** | <https://www.kth.se/blogs/pdc/2018/11/scalability-strong-and-weak-scaling/> |
| **Finansijski podaci** | <https://finance.yahoo.com> |