

---

# MONTE CARLO SIMULATION: GO VS PYTHON BENCHMARK

---

Assignment 4: MSDS 458 Fall 2024

Hannah Graham

Northwestern University

GitHub Repo: [https://github.com/hannah-r-graham/GoVsPython\\_MonteCarloStudy\\_Assignnment4\\_MSDS460\\_business\\_analytics.git](https://github.com/hannah-r-graham/GoVsPython_MonteCarloStudy_Assignnment4_MSDS460_business_analytics.git)

Email: Hannahrg24@gmail.com

## Abstract

In recent years, Go Lang created by Google Inc, has created immense stir and popularity amongst developers in contrast with Python language that has been widely used for decades and remains the top language in Repos on GitHub. How do these two languages compare with computational ability from simple to complex mathematical operations and string computations. This comparison found that out of 8 computational tests that include math operations and string operations, Go outperformed Python in speed in 75% of tests when comparing average times of calculations over 100,000 simulations. This is similar to other benchmark experiments that have already been completed. Further analysis on how performance changes based on complexity of tasks and Go's integrability in current software systems is needed.

# Table of Contents

Abstract ..... i

Introduction ..... 1

Literature Review ..... 1

Methods ..... 1

Results ..... 3

Conclusions ..... 6

References ..... 7

Appendix A ..... 8

# Introduction

With so many programming languages available for use today which does a developer choose? The answer lies in finding the right language for the job. How do you know what is the right language for which job? In this benchmark study, we seek to compare the computational speeds of Python and Go on mathematical functions and string functions, two very common use cases with programming languages.

Go, formerly known as Golang, was developed at Google in 2009 as a high-level general-purpose language with a strong affinity for readability, multiprocessing, and run-time efficiency similar to the C language (Wikipedia 2024). In comparison, Python was first released in 1991 and has such widespread use that it recently took over JavaScript as the top language used on GitHub in 2024 (GitHub Blog 2024). In other words, Go is the new kid on the block but is rapidly gaining popularity. While there are many considerations and restrictions when assessing languages, a primary concern is the time it takes to perform various tasks. How do these two models compare in computation time on common mathematical and string computations?

## Literature Review

Comparing programming languages and their performance is very common. Exaloop completed a comparison of speed and execution similar to this study, among other parameters. Using a simple test case of counting one to a million, Go outperformed Python in speed and in concurrency tasks (Exaloop 2024). Ruby Roid Labs recently did an in-depth comparison on Go and Python and found similar results to Exaloop – due to Go’s functionality of translating code into machine readable code before execution, Go is much quicker than Python. Add in Go’s ability and emphasis to parallel process, and Go is faster than Python (Bibikov 2024).

## Methods

Using Monte Carlo simulation in conjunction with random numerical variables and strings, Python and Go will be compared on eight different functions where each function will be timed with the standard time packages for Go and Python languages. Those times will then be summed, then divided by the sample size to get the average time for each test.

There are five mathematical tests:

- $a + b$
- $a * b$
- $a ** b$
- $b * \text{sqrt}(a)$
- Matrix Multiplication (A dot product B)

There are three string function tests:

- Concatenation
- Substring Search
- Regex Matching

A Monte Carlo simulation will be performed on each test as follows:

- Number of samples = 100,000
- Matrix size = 10
- String size = 1,000

**Packages uses:**

Python:

- Pandas
- Numpy
- Random
- Time
- Matplotlib.pyplot
- String
- Re

Go:

- "encoding/csv"
- "fmt"
- "log"
- "math"
- "math/rand"
- "os"
- "regexp"

- "strings"
- "time"
- "gonum.org/v1/plot"
- "gonum.org/v1/plot/plotter"
- "gonum.org/v1/plot/vg"

## Results

CSV of results can be found here: [https://github.com/hannah-r-graham/GoVsPython\\_MonteCarloStudy\\_Assignment4\\_MSDS460\\_business\\_analytics/blob/main/FinalSimulationResults.csv](https://github.com/hannah-r-graham/GoVsPython_MonteCarloStudy_Assignment4_MSDS460_business_analytics/blob/main/FinalSimulationResults.csv)

Operation (100,000 samples)	Python Average Time (seconds)	Go Average Time (seconds)	Percentage Difference (%) Go compared to Python	Fastest Language
Addition	7.673025131225586e-08	0.0	Null	Go
Multiplication	2.496790885925293e-07	0.0	Null	Go
Exponentiation	2.0070075988769532e-08	1.07722e-07	-436.7294078023284	Python
Square Root	5.742049217224121e-07	0.0	Null	Go
Matrix Multiplication (size = 10)	8.51142406463623e-07	9.32186e-07	-9.521743120884045	Python
Concatenate	9.195566177368164e-08	4.052e-08	55.93528531203816	Go
Search Substring	5.780410766601562e-07	2.65452e-07	54.07731202534152	Go
Regex Match	0.00024084230661392211	1.048991e-06	99.56444903109089	Go

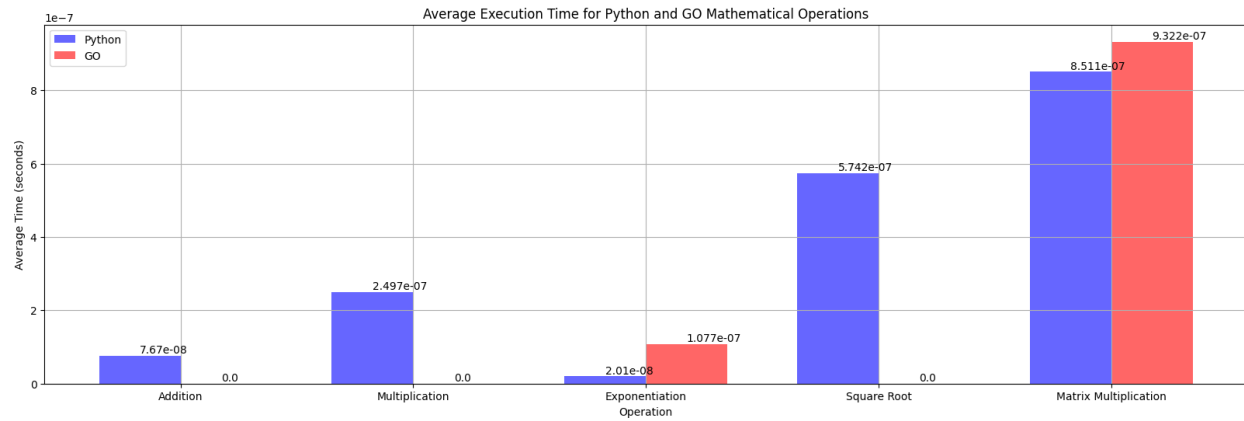


Figure 1: Bar chart demonstrating the average times to compute each mathematical test case for both Python (blue) and Go (red). As you can see, Go times was greater than Python only in 2 use cases.

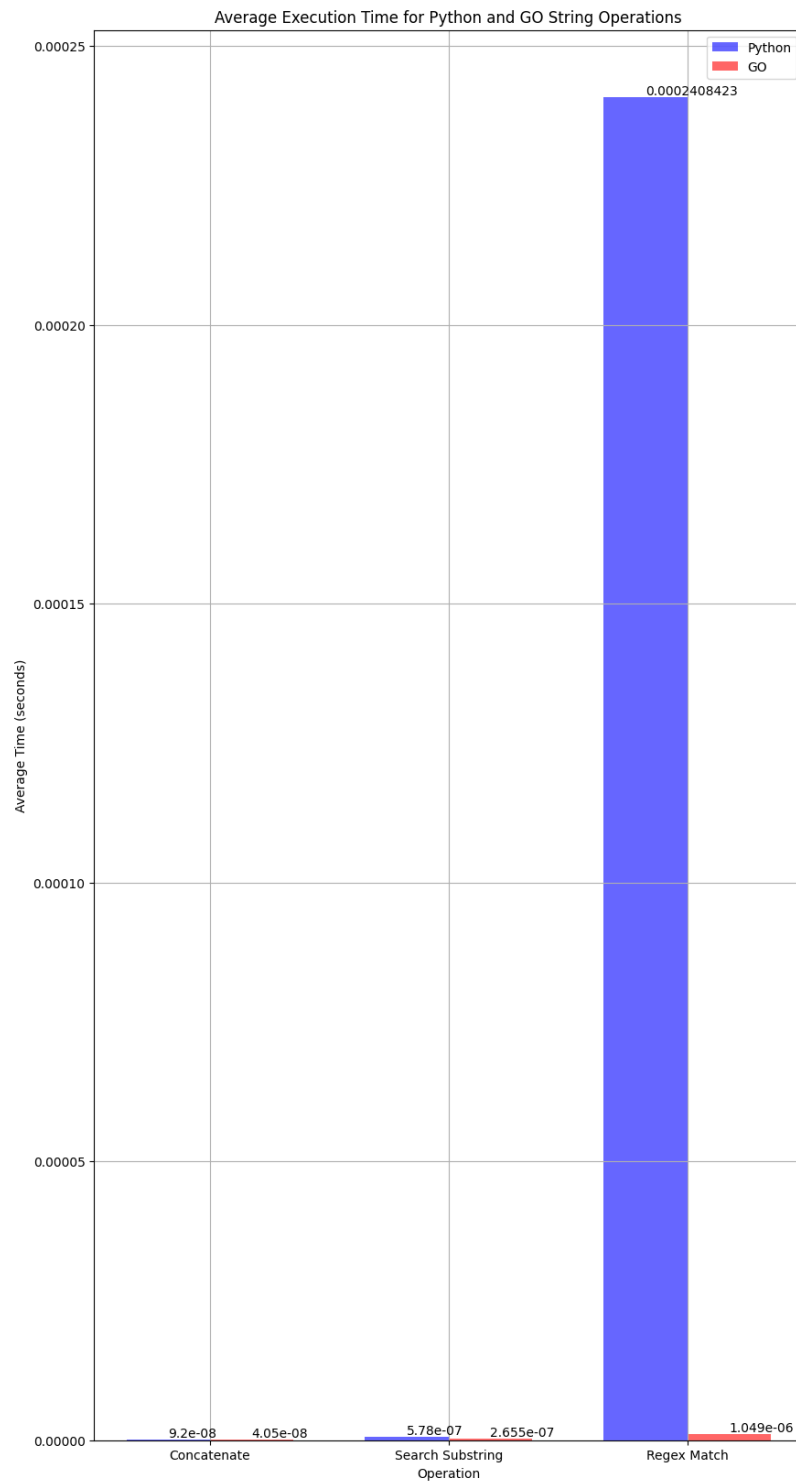


Figure 2: Bar chart displaying average processing times for string manipulations for Python (blue) and Go (red). Go had such short processing times it is difficult to even see them on the chart.



## Conclusions

When simulating 100,000 samples, Go outperformed Python in six out of the eight tests. Go outperformed Python in every string test and in three out of five of the mathematical tests. Python outperformed Go in the exponentiation calculation and the Matrix multiplication test case.

Interestingly Go had exactly zero seconds for the addition, multiplication, and square root use cases. Of course, it's not possible for the calculation to be zero so how could it show up as zero? After experimenting with sample sizes, and all of them came back with 0 for the time, I am hypothesizing that the calculation for addition, multiplication, and square root is so fast that the timing package in Go simply could not start the timer quick enough before the calculation ended.

While Go is faster than Python in 75% of cases, Go did require more functions and code to achieve the same things Python has built in ability for. For example, with the matrix multiplication, Go required a separate function where I checked that the matrix sizes were the same and if not, throw an error. Python has this functionality built in. While Python may be slower than Go, Python does have more ability built in to handle various errors that Go does not have those considerations included. Not having more “junk” on the back end is probably what supports Go being so fast. However, in a production environment with more complicated tests and scenarios, manual coding additions in Go that are automatic in Python could grow frustrating, especially if you are a newer developer to Go or Python.

Considering the applications and integrations of Python are endless, more testing is needed to understand if Go can also apply in all environments as often as Python does, as well as how Go handles more complicated tasks such as Machine Learning Algorithms, One hot encoding, and more. Go is the clear winner with simple computational tasks, but its application and abilities in real-life solutions leaves much to be explored.

## References

Bibikov, Ilya. "Go vs. Python: Which programming language is Better for a Project?" Ruby Roid Labs Blog September 19, 2024. Date Accessed: November 15, 2024.

<https://rubyroidlabs.com/blog/2024/09/go-vs-python/>

GitHub Staff "Octoverse: AI leads Python to top language as the number of global developers surges" GitHub Blog October 29, 2024. Date Accessed: November 15, 2024.

<https://github.blog/news-insights/octoverse/octoverse-2024/>

"Go (programming language)" Date Accessed: November 15, 2024.

[https://en.wikipedia.org/wiki/Go\\_\(programming\\_language\)#::~text=Go%20was%20designed%20at%20Google%20%E2%80%80in%202007%E2%80%81%20to%20improve](https://en.wikipedia.org/wiki/Go_(programming_language)#::~text=Go%20was%20designed%20at%20Google%20%E2%80%80in%202007%E2%80%81%20to%20improve)

"Python (programming language)" Date Accessed: November 15, 2024.

[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

Team Exaloop. "Go vs. Python Performance: A Head-to-Head Comparison" Exaloop Blog July 15, 2024. Date Accessed: November 15, 2024. <https://exaloop.io/blog/go-vs-python-performance/>

# Appendix A

