CS263 Project

Github repository: https://github.com/avanitanna/CS263-project

Team: Avani Tanna (avani@ucsb.edu) , Satyam Avasthi (satyam@ucsb.edu)

Photos:

Overview:

In this project, we would like to learn a new programming language - Go! Go is a statically type language designed at Google. Golang is high up on the list of in-demand languages today, deemed very useful for programming scalable servers and large software systems. We would like to evaluate its learning curve and performance. We will compare it with Python (v3), a dynamically typed language, and in doing so, highlight the interesting features of the language. Python is another popular language that offers us plenty of opportunities to compare the learning curves and analyze differences in the design of the two that yields a performance boost for one in the empirical evaluation. To this end, it would help us upskill ourselves whilst understanding important similarities, differences and practices when it comes to programming language implementations as well as runtime systems.

Project plan and components:

Survey piece:
We plan on referring to the official documentation, books, code, tutorials, popular blogs and content from the developer community of Golang and Python. This would help us understand the programming language structure, implementations, and the recommended programming practices along with popular applications. By conducting the survey, we also hope to gain insight into the learning curve of the language.

Coding piece:
We plan on designing a set of experiments to dig deeper into the programming language implementations and runtime systems. We intend to solve and implement the same set of problems using both programming languages to understand the functionalities parallely.

Empirical evaluation:
We would like to design our experiments strategically such that we can highlight important features of the language as well as show similarities and differences, for instance, coding efficiency, concurrency, compilation, runtime environments, performance analysis including memory footprint, execution time, etc.