GOLANG ~ Problem, Characteristics and Good at

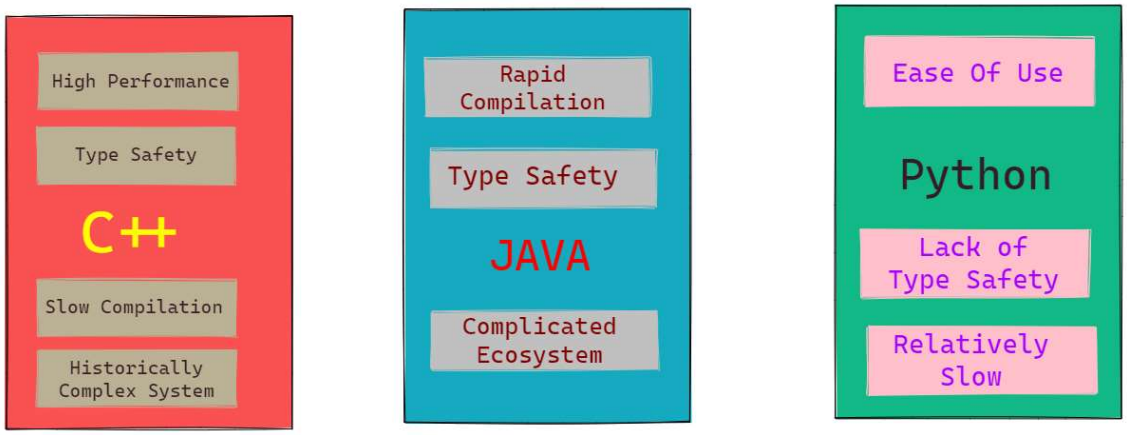
1.1 The problem

To understand what go is good at we really need to understand the problems that go is trying to solve, In order to understand the problem go is trying to solve we need to understand the context that the go language was founded in.

Go is actually created by small team of very talented engineers within the google organization, and at that time google has primary 3 languages to develop the application i.e. C++, JAVA, and Python.

When we look at these languages and try to map those on the problem spaces that google was addressing, we have some advantages that each language brought and we have some disadvantages.

For example :-



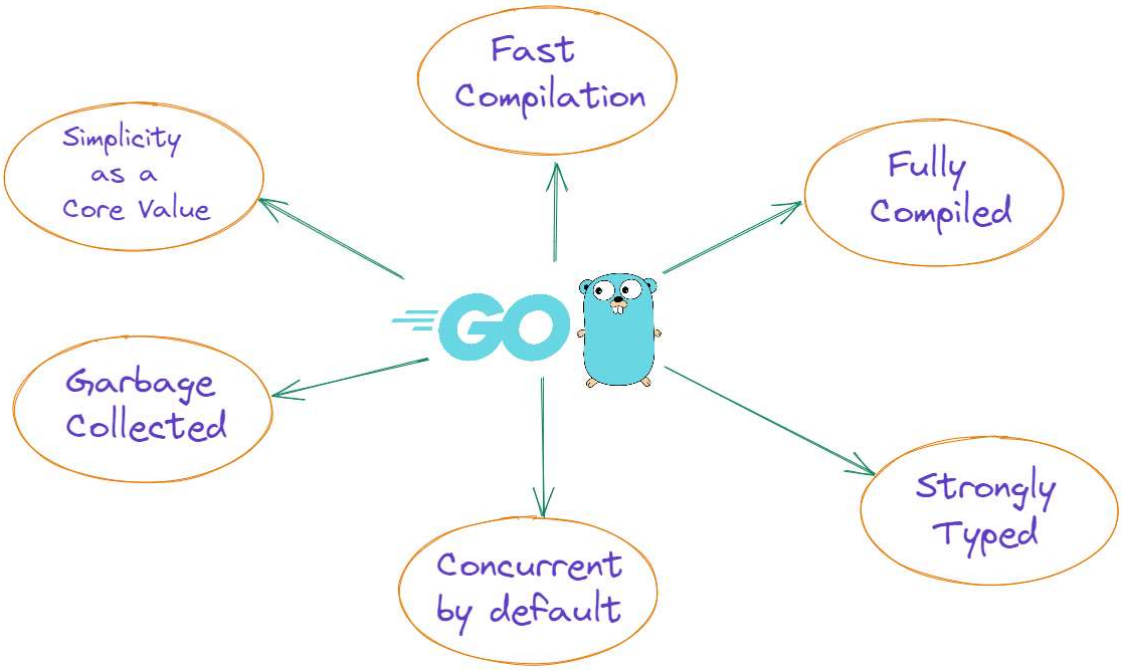
With C++ we can typically get High performance and type safety that makes our application more maintainable. However with C++ we can get Slow compilation and Historically complex syntax as disadvantage.

If we move to JAVA we se that some of the concerned of C++ have been addressed, JAVA typically has Rapid compilation and Type safety. But with JAVA we can get Complicated ecosystem there are many different ways to solve the same problem and some of those solution does not play nicely together so you tend to building custom tool chains for each JAVA application that you build which means that every engineer that moves on to your project needs to understand that specific set of tools that are being used in that application.

If we look at python we have another type of strengths, python has been historically recognized for very easy language to use, However python being a dynamically typed language lacks type safety so types can change over time which becomes little bit difficult to maintain application if those types aren't used consistently. Pythons is also relatively slow when compared to JAVA and C++ applications.

The problem Google was facing that non of these 3 languages were able to solve the google problem the best way possible. **So what was the solution to that then?** The solution was to create a new language called GO.

So what is the characteristics of the GO language?

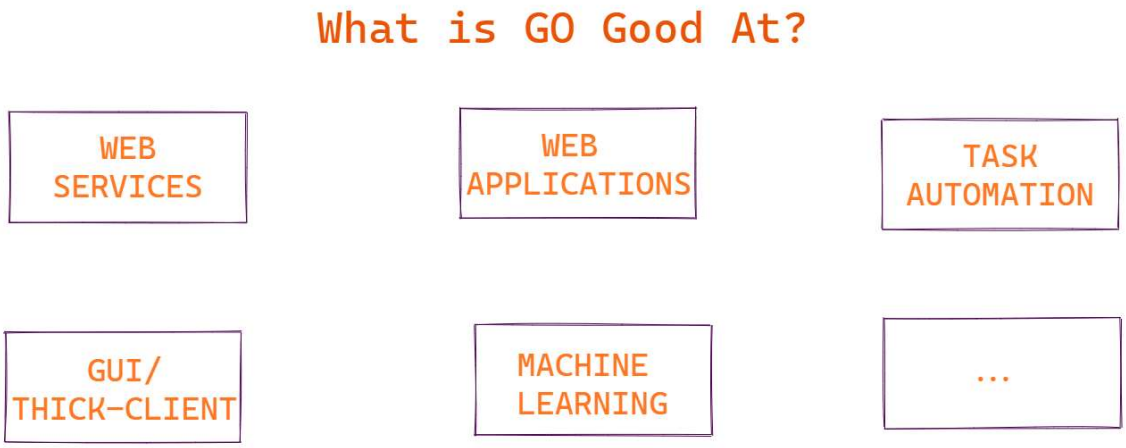


1. First characteristics that is Key to GO is **Fast compilation**
2. GO application are **fully compiled**.
3. GO is **strongly typed**
4. GO is **concurrent by default**
5. GO is also **Garbage collected** language
6. **Simplicity is a core value** of GO

1.2 What is GO good at?

So what is Go actually Good at?, Like most of other programming languages GO is a general purpose programming language, it means if you want to you can create any type of application you want with it. However, there are some types of application that Go is better then another.

So what type of application are created using GO?



1. **Web Services :-** Web service are primarily concerned with delivering data.

2. **Web application :-** Web application are primarily concerned with delivering web pages on web browser.

   - GO is really good at these two types because go is designed with concurrency in mind. So, web services and web applications are typically concerned with dealing with multiple transactions that are happening simultaneously well Go's built-in concurrency model easily allows to handle that at the same time.
   - Keep in mind, GO is designed within google, Google deals with network based applications all the time. GO standard library and it's support for network aware applications is very-very strong making GO very good choice for these type of applications.

3. **Task automation :-** GO is also commonly used in task automation, In this area Scripting languages like python, Java scripts or even bash scripts are used. So why GO is used for this? ~ It turns out that GO syntax is almost as light as many of the scripting languages, even though it's a

strongly typed language the compiler takes great strides to remove the ceremony of declaring those types from the engineer or developer. So writing GO applications that are performing automation task is often as simple or simpler then using scripting languages and you have the additional strong maintainability and clarity that a strong typing language provides for you.

4. **GUI/ Thick-client :-** You have an experiment to build GUI or thick-client applications using GO. There is an experimental library that provides all of the basic UI elements that are required to build applications that can run on multiple different platforms.

5. **Machine Learning :-** GO is also becoming very popular in machine learning space, a space very commonly occupied by python developers and that's again because the syntax is light enough and lot of the characteristics of the GO language are similar in many ways to the python based applications not the same but it is founded that GO has found very nice niche in machine learning space.

There are many more application types that are being explored with GO, Why? ~ GO is around for several years but relatively young by programming standards. So the community is still exploring and trying to understand what type of application GO is really Good at to solving and really good at helping to build and What type of application the GO is not good at.

If you look at in the community you will see wide variety of a experiments with the GO programming language with some being much more successful than others.

NOTE :- GO is extremely good at building Web services and Web application. It can do the other ones but really shines at these two.

1.3 Play with GO code

Now we have little bit of background about GO, where GO came from and the problems that GO is trying to solve. Let's see some GO code and play with it.

For that visit URL https://go.dev/play/ This is an online editor that supports the development of simple go applications. It's is a great resource if you are learning language itself, or you are exploring some aspects of the standard library. Lot of time you can just fire something very quickly in the playground and that helps you to really understand how to integrate some functionality in the application.