Modern Programming Language Go

Aliaksandr Hladkou
Dr. Alvin Grissom II
CS-374A: Principles of Programming Languages
Ursinus Colloge

1 Introduction

The main purpose of the project is to highlight all the facets of the programming language Go, compare with other languages such as Java or C# and contrast in sense of code writing complexity. In addition, another part of the project is to provide examples to demonstrate the performance of Go versus the other languages by using algorithms.

2 History

2.1 Releasing Date

Go is compiled programming language designed at Google by Robert Griesemer, Robe Pike, and Ken Thompson. They started to develop a new language on September 21, 2007. In the beginning of the 2008, Ken started to work on exploring their ideas and theories by making a small compiler which generated C code as the output. Go became a public open source project on November 10, 2009.

2.2 Purpose of Creating Go

The purpose of creating a new language was simple: most production software was written using C++, Java, or Phyton. Those language are from 1990's single threaded environment. However, even most of those languages supports multi-threading, but they still offered a little help to program multiprocessors efficiently. Therefore, when developers worked on Go, concurrency was kept in mind because multi-core processor were already available.

3 Advantages of Go

Go is a relatively new programming language in the computer world. Therefore, developers kept in mind many aspects when they worked on creating a new

programming language. There were clear considerations to include when creating Go:

> "We decided to take a step back and think about what major issues were going to dominate software engineering in the years ahead as technology developed, and how a new language might help address them. For instance, the rise of multicore CPUs argued that a language should provide first-class support for some sort of concurrency or parallelism. And to make resource management tractable in a large concurrent program, garbage collection, or at least some sort of safe automatic memory management was required."

3.1 Multi-threading

Most programming languages support multi-threading. However, Go becomes more beneficial in terms of concurrent execution, threading-locking and deadlocks. As an example, creating threads in Java is less memory efficient rather than creating goroutines in Go because every thread consumes approximately 1MB of the memory while goroutines consume about 2KB memory. Therefore, creating thousands of threads in Java will cause shut down, when the RAM memory will be full. In contrast, it is possible in Go to create millions of goroutines at any time. In addition, communications between threads are very difficult.
Furthermore, there are additional benefits of gorontines comparing to threads:

- Goroutines have a faster startup time than threads
- Goroutines use channels to communicate between themselves while communications between threads are very difficult

3.2 No Virtual Machine

Go is compiled language which means that it does not require any Virtual Machine. As an example, Java or C# will require VM in order to translate human readable code into machine readable code i.e. into binary. Like C++, Go compiles into binaries directly which helps to increase the performance.

3.3 Cross-Platform

Portability is one of the most important advantages of Go, since it is a compiled language. Comparing to other programming languages, Go does not have many dependencies and handle them at build time rather than at run time. This feature makes Go more friendly toward developers and makes the cross-compiling very simple.

3.4 Embedded Garbage Collector

Go has embedded garbage collection. Many variables, local to a function as an example, disappears when the function exits. And if using programming

languages such as C, it's up to developers to free the memory associated with those variables which is no longer needed. In C, a developer would need to use function $free(str)$; Therefore, using Go developers don't have to think when they have to free those variables. Garbage collection adds overhead, but it also eliminates a number of devastating bugs.

3.5 Code is easy to maintain

Programming in Go is different then programming other languages such as Java or C#. The major part to point out is that Go does not include many features of OOP languages:

- No classes
- No inheritance
- No constructors
- No annotations
- No generics
- No exceptions

All the above makes code cleaner and easier to maintain. Also, it makes code easy to understand for the other developers working with the code.
In addition, Go is strict about importing packages. Unused imports can slow compilation. Therefore, it will not compile if a program imports a package but do not use it.

3.6 Go is Easy to Learn

Go is a relatively simple language with a relatively simple standard library. For instance, experienced developers would start coding within first two days. One of the purposes of Go is simplicity. Therefore, this programming language has rigidly fixed style code, a developer would spend less time writing a code rather than reading it. Also, the code written by many developers will look very similar. Another example, Go has less specifications than most other programming languages.

3.7 Single Exe

Go complies everything into single file. Since this file does not have any dependencies, it can be very beneficial for developers when they often have to work with dependencies too. However, it is important to assume that it will load everything into RAM at once which may cause long start up time.

4 Difference Between Languages

Creating a simple table will help to illustrate the difference between programming languages.

	Java	C#	C++	Go
Multi-threading	Simple	Simple	Complex	Simple
Garbage Collector	Yes	Yes	No	Yes
Virtual Machine	Require	Require	No	No
Learning	Simple	Simple	Difficult	Simple
Specification (pages)	780	500	1300	85

5 Using Go Worldwide

Nowadays, Go is used by many companies in different countries. Therefore, being more specific, here's some of most famous companies: Google, Apple, Docker, Dropbox, SoundCloud, Facebook, Twitter, BBC. Adobe.
In addition, the current century is the time of the Internet: it is developing very fast. And Go is one of the best programming languages for writing network application. Usually, it helps servers to get their work much faster than using other languages. Also, Go was originated for system programming and it can be used to write a system code.

6 Go in Practice

Go is very popular language for supporting web-application. However, there are other choices to check performance of this language.
Let's begin with an example of using Go to implement parallel merge sort using channels:

```
//mergesort.go
func mergeSort(data []int, r chan []int) {
    if len(data) == 1 {
        r <- data
        return
    }

    leftChan := make(chan []int)
    rightChan := make(chan []int)
    middle := len(data)/2

    go mergeSort(data[:middle], leftChan)
    go mergeSort(data[middle:], rightChan)
```

```
    left := <-leftChan
    right := <-rightChan

    close(leftChan)
    close(rightChan)
    r <- merge(left, right)
    return
}
```

In contrast, the implementation of parallel merge sort in Java using ForkJoin:

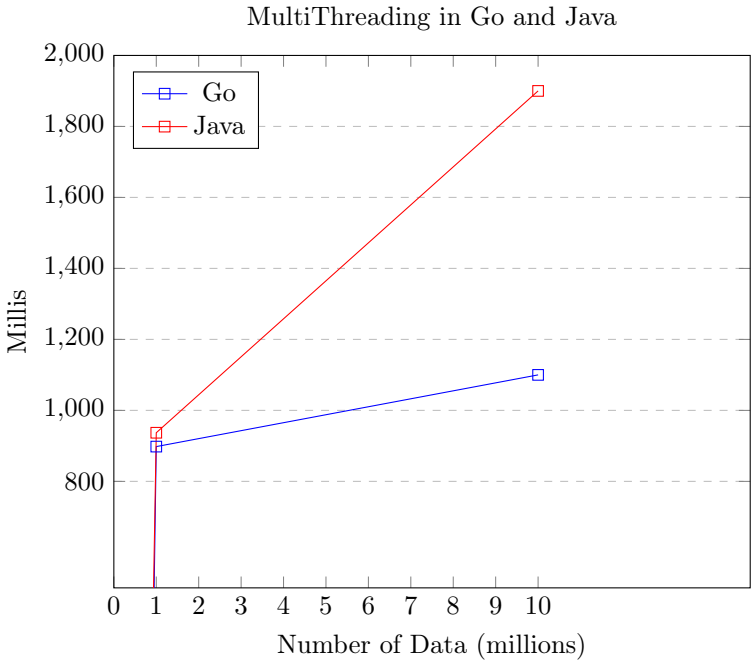
```
//MyMergeSort.java
public static void mergeSort(int[] a, int n) {
        if (n < 2) {
            return;
        }
        int mid = n / 2;
        int[] l = new int[mid];
        int[] r = new int[n - mid];

        for (int i = 0; i < mid; i++) {
            l[i] = a[i];
        }
        for (int i = mid; i < n; i++) {
            r[i - mid] = a[i];
        }

        mergeSort(l, mid);
        mergeSort(r, n - mid);

        merge(a, l, r, mid, n - mid);
    }
```

Now, let's take a look at the result:



7 Issues And Difficulties Using Go

After using programming languages such as Java or C#, there are several points to bring that a programmer should keep in mind. As mentioned before, Go is strict about importing packages which means that it will not compile any program if there exist unused packages. Also, it is important to memorize that all declared variables must be used. The same as with unused imports, Go will not allow to have unused variables.
For instance, Go will not run program because there is unused variable:

```
//Author: Aliaksandr Hladkou
//Date: 10/05/2019
//Go basics

package main

import (
   "fmt"
)

func main() {
   text := "Hello, world"
   num1, num2 := 5, 10
```

```
   fmt.Println("The sum is: ", add(num1, num2))
}

func add(a, b int) int {
   return a + b
}
```

In addition, there is even less visible issue that exists in Go. It is necessary to use *rand.Seed*() function in order to have different set of random values when executing the program over and over again. Otherwise, it will keep using the same values.

References

[1] Karl Seguin, "The Little Go Book." https://github.com/karlseguin/the-little-go-book

[2] Keval Patel, "Why should you learn Go," *Medium*, 2017, Jan 8. https://medium.com/@kevalpatel2106/why-should-

[3] Johannes Liebermann, "Why Golang Is Great for Portable Apps," codeburst.io, https://codeburst.io/why-golang-is-great-for-portable-apps-94cf1236f481