

CS 131: Containerization Support Language

Jonathan Woong
804205763

Abstract

This report intends to explore language alternatives to Docker, which is implemented in Go. The possible alternative languages are Java, OCaml, and Coconut (Python). The intent of implementing an alternative Docker, DockAlt, is to be able to run any software on any platform using a Linux container. The key features to examine are portability, performance, and maintainability.

1 Introduction

Docker is a software containerization platform written in Go that allows for any piece of software to always run the same, regardless of its environment. Docker achieves this by wrapping the software in a complete filesystem that contains all of the software's dependencies¹.

A few of Docker's key features include:

- being lightweight: containers on a single machine share the same OS kernel, start instantly, and use less RAM
- being open: containers are based on open standards, allowing them to run on all major Linux distributions and on Windows
- being secure: containers isolate applications from one another and from the underlying infrastructure while providing a layer of protection for the application¹

A few of Go's key features include:

- reduction of code bloat using a declare-and-initialize construct
- removal of type hierarchy using duck typing
- orthogonality: methods can be implemented for any type, structures represent data while interfaces represent abstraction²

- ability to link with GCC-compiled C programs³
- goroutines that multiplex independently executing functions (coroutines) onto a set of threads instead of traditional threading
- removal of exceptions in an effort to improve readability of source code⁴

2 Java

Java is an object oriented language that runs inside of a Java Virtual Machine (JVM). In this way it is inherently secure and portable. However, its interpreted strategy leaves it with generally slower performance than compiled languages⁵.

One benefit of implementing DockAlt using Java is safety in types. Virtually any language that does not implement duck typing will ensure that developers understand the exact behavior of caller and callee while allowing for easier compile time debugging. This feature, however, goes against Docker's duck typing and ultimately means that the system would be slower if implemented in Java.

Another benefit of Java is that the JVM can run on many platforms without the need to tailor to any specific one. This is a key feature of Docker that can be carried over in Java.

One downfall of using Java is that DockAlt would be much more bloated compared to Go and would have longer source code. This can harm readability and maintainability if developers in the future intend to read or modify the source code.

3 OCaml

OCaml is a functional programming language that emphasizes application safety and speed. The language is generally considered simple and easy to both learn and teach, making it viable for quick prototyping.

One benefit of Ocaml that can be carried over to DockAlt is the type inference system, which mirrors Go's duck typing. This leads to better performance over a Java alternative and cleaner looking code. The downfall of this language feature is increased difficulty in debugging and maintainability in the future.

Another benefit of Ocaml is its ability to interoperate with C code. This increases the portability of an OCaml DockAlt since Linux is written in C, although Ocaml is not as system-independent as Java.

Another benefit of OCaml is its evaluation regime, which is strict but also allows lazy evaluation when required. This allows for OCaml to manipulate potentially infinite data streams, which is necessary to operate a system like Docker.⁵

4 Coconut

Coconut is a functional programming language built on Python. Coconut code can only extend upon Python code. The core purpose of Coconut is to bring functional programming features to Python, such as pattern-matching and lazy evaluation⁷.

One benefit of Coconut (Python) is its code simplicity and ease of quick implementation. This makes prototyping quick and easy compared to verbose languages like C++ or Java. This will make implementing DockAlt easier in Coconut than Java.

One downfall of Python is its slow performance. This disagrees with the Docker philosophy of high performance.

Python's duck typing is also a feature of Go, which can either be beneficial or detrimental. Its benefit is being able to more accurately mirror Docker's behavior, but its downfall is an increased difficulty in debugging and source code understanding.

References

1: <https://www.docker.com/what-docker#/VM>

2: <https://golang.org/doc/faq#principles>

3: <https://golang.org/doc/faq#Usage>

4: <https://golang.org/doc/faq#Design>

5: <http://www.javatpoint.com/features-of-java>

6: <https://ocaml.org/learn/description.html>

7: <https://coconut.readthedocs.io/en/master/FAQ.html>