HW6 Report

Xiao Yan, *UCLA*

Abstract

This reports talk about the potential possibility of implementation of an alternative version of Docker, DockAlt, with three available languages including Java, OCaml and Coconut. Docker is written in GO. In this report, I will describe the key points of GO language and compare it to other three languages.

Docker and Go

Docker provides an easy and convenient platform for software developers to work under different environments. For most of the case, the software development/management need to be done under certain platforms, with platform-specific libraries, settings, etc. Most of the case it is difficult to share software or information across platforms due to the deeper design of these platforms and commercial reasons related[1].

One solution is a virtual machine. That one physical machine can contain two or more set of OS kernels, each has their own space, although the guest OS has to operate on top of host OS. This is one solution to solve this problem, however, having multiple kernels running on the same machine is most of the time delaying the speed of computing and may cause other issues in memory management or system I/O.

Docker provides a different solution to this. It embraces the basic idea of a virtual machine, but contains only the essential part that can be used in software development. All the essential parts are contained in the Docker distribution package: code, runtime, system tools and system libraries. This guarantees that the software will always run the same regardless of the environment.

Based on the Docker documentation, Docker has three main advantages:

LIGHTWEIGHT. Since Docker is essentially a container running on a machine and sharing the kernel with the OS, the memory usage is smaller.

OPEN. Docker is compatible with different components running on Linux as well as Windows system, which enables the multi-platform development.

SECURE. As a whole containers in Docker is one complete piece containing all the necessary development tools, so communications with other software will be minimum. That adds additional security benefits.

The Docker development team choose the GO language to implement Docker. According to the introduction, the benefits of using GO language could be[2,3]:

GO uses static compilation. Each function in GO generally contains all the library so usually it does not need the installation of other accessory software packages (Certain external packages can be included).

GO uses a very good asynchronous process writing so I/O will influence less of the overall continuation speed. GO also utilized duck typing method so it requires minimal knowledge or rules of the input and it can greatly increase the flexibility of the language programming

GO has already a functional set of packages that can help the functional workflow.

Now I will discuss the potential benefits and drawbacks of using other programming language to implement the DockAlt.

Java

Java is one of the most popular languages and it has a couple of very useful features that may be used in the DockAlt implementation[4,5,6]:

Java is concurrent, class-based and object-oriented. The object-oriented feature of Java is very useful in the implementation. In Java, everything including the main function is an object bundled with all the associated variables and functions. Objects communicate through each other with interfaces. This concept in design means that we can easily load different

parts of code into isolated packages. We can establish different software as long as we have very good interfaces and connect them together efficiently, which is the key concept of Docker.

The memory management of Java introduces the unique concept of Java virtual machine (JVM). It enables a computer to run a Java program. JVM translates java code into a bytecode instructions. Then the JVM translates the bytecode into the machine code, and by using this system, it allows the code to show the same results regardless of the platforms it runs on. That way it is very convenient to share resources and codes across platforms.

Java, unlike GO, because of its popularity, has quite a few very efficient and user-friendly IDEs. This greatly frees users from remembering the functions or checking the format of the code.

However, Java is a very verbose code. It takes time to be familiar with and transit from other languages to Java. Besides, right now Java lacks certain internal packages that support the development of LXCs.

OCaml

OCaml is another objected oriented program also less used. It uses the static type-checking to help eliminate problems at runtime. However, that also increase the programmers' work to ensure the rightness of the code[7,8].

Unlike other programming languages, OCaml focuses on improving the performance. The type-checking mechanism ensures that runtime type mismatch is impossible and thus obviates the runtime type and safety checks that harm the performance of the dynamic type checking languages.

OCaml is also a functional programming languages. To increase the efficiency, OCaml has a optimizing complier that can maximize the performance of the resulting machine code. Since OCaml is functional based and everything is written in function, it has a very similar sytle that GO uses. So, the transition from GO to OCaml will be relatively easier compared to other languages.

However, OCaml has its own limitations. OCaml is not a very popular language. That means there will be less experienced worker using OCaml. And there will be less available package around and developers will need to establish a lot of additional libraries. That increases time and cost.

Secondly, the grammer in OCaml is loosely based on verbal laguages compared to other popular language such as python, C or Java. That increases the difficulties of undering other's code.

And finally, there are currently no very good IDEs for OCaml. That again increases the difficulties of using OCaml as a wide-spread programming language.

Coconut

Coconut is a programming language and it compiles to Python. That gives additional advantage to Coconut that every Coconut instructions are essentially Python code. Coconut expands Python and adds modern function programming elements into the Python language. Coconut transform Python into a functional language[9].

Since Coconut compiles to Python, so essentially it has the same benefits that Python will have. Python uses a strong Duck Typing method and it provides with very high flexibility of the operations on the variables. That fits very well with the goal of the Docker, which is to help the development of multi-platform software. One reasonable assumption is that the developer may not know very well on the grammars of all the languages. So a Duck typing based language is very useful.

Python has become a more and more popular language nowadays. According to TIOBE, Python has become the fifth most popular language and it is kept moving up. There are multiple packages that helps Python dealing with large amount of data. As a matter of fact, Python is the leading choice for big-data related analysis. With development of the big-data concept, number of packages for Python has been escalating. Since Coconut is using Python, it will be easier to utilize packages written in Python. Even if it is not implemented in Coconut yet, the development will be easy by modifying the interface of Python package.

The third advantage is that, Coconut has a syntax that is very close to modern language.

That makes Coconut very easy to learn and master. However, there are not so many well-known IDEs for Coconut so that will be a disadvantage.

Conclusion

In general, it is feasible to write a different version of Docker, DockAlt, with the three languages we choose. Of this three, we believe Coconut will be the most convenient one due to its internal nature.

Reference

[1] Docker website: https://www.docker.com/
[2] GO documentation: https://golang.org/
[3] Developer talk on GO and Docker: http://www.slideshare.net/jpetazzo/docker-and-go-why-did-we-decide-to-write-docker-in-go
[4] Java wiki page: https://en.wikipedia.org/wiki/Java_(programming_language)
[5] Java documentation: https://www.java.com/en/
[6] JVM wiki page: https://en.wikipedia.org/wiki/Java_virtual_machine
[7] OCaml wiki page: https://en.wikipedia.org/wiki/OCaml
[8] OCaml documentation: http://www.ocaml.org/
[9] Coconut documentation: http://coconut-lang.org/