Containerization Support Languages Report

Thomas Lutton
University of California, Los Angeles
CS 131, Fall 2015

Abstract

Docker is an open-source that is used for building, shipping, and running distribute applications via Linux Containers (LXC). Docker has the potential to solve our deployment needs better than Twisted did, but our boss is cautious to use it because of its single implementation in the recently created language, Go. This paper examines the implementation of Docker that uses Go and also proposes three different languages that an alternate version of Docker, DockAlt, can be written in: Java, Python, and Groovy. A conclusion will be drawn as to which language and implementation shows the most promise for our desired application.

1 Introduction

In the course project, we implemented a simple proxy herd using Twisted and the Google Places API. Now, we are looking to keep costs low by replacing the virtual machine that Twisted used with Linux Containers which carry much less overhead since they operate at the kernel level. To do this, we are looking to evaluate Docker and determine if its Go implementation is appropriate for our purposes or if we should implement our own DockAlt in either Java, Python, or Groovy. We need to evaluate these other three languages and potential implementations because Docker is very new and is using a relatively new language as well. Because of this, it is possible that bugs may exist, or support for the language or Docker itself may drop off over time.

Docker utilizes Linux Containers which are individual operating-system level virtualization environments for running multiple isolated Linux systems on a single Linux control host. The Linux kernel allows for prioritization without the need for the starting of additional virtual machines. Essentially, applications believe they are running on a separate machine from one another when in actuality they are all sharing a single Linux host on the same machine. The benefit of LXC is that the user can avoid all of the downsides that come with the overhead of running multiple virtual machines, especially when the number of virtual machines would be very high.

2 Docker With Go

The Go programming language was developed by Google engineers in 2007. Docker was released as open source in 2013. The relative age of both of these technologies to more established languages like Java and Python is what initially causes apprehension.

Go is an imperative language that utilizes static compilation and contains an solid asynchronous primitives, low-level interfaces, an extensive standard library, and strong duck typing. With Go, developers can easily create binaries that are free of dependencies.

Go supports many different types and utilizes dynamic type inference and checking through duck typing. Go supports all of the stand types that we have come to expect from a newer, less-verbose language such as tuples, complex numbers, and pointers (even though pointer arithmetic is not allowed). Classes are also supported by Go in an interface style. In order for a class to be valid, it must implement the specified methods from the interface. Further, Go supports parallelism with goroutine and channel.

It should also be noted that since Go is a relatively new language, many programmers have yet to develop any bias for or against it like they would have with older languages like C++ or Java. This is a benefit to the Docker team because people may be more interested in trying Docker since they would have an unbiased opinion. This can also work in the opposite direction, that since Go doesn't have many followers, it could never gain traction. However, that doesn't seem to be the case since Go has gained much more popularity since Docker was released and became widely adopted.

3 DockAlt with Java

Java has been around since 1995 and since its inception has grown to one of the most widely used programming languages in the world. The intended application of Java allows developers to "write once, run anywhere." Since Java is a mature language and has such a large user base, it is a more stable and reliable choice of language over Go because Java has shown no signs of going away any time soon. With a plethora of built-in and third party libraries available, Java is much more established to meet any need that a programmer could have.

Technically speaking, Java is a compiled language like Go. Java is compiled down to bytecode and then run on a JVM. This allows Java programs to run on any Java compatible system with only one compilation. This also makes Java fast. It doesn't have to do any implicit typing. Java also supports parallel programming through its libraries. However, Java as a language is very different than Go. While the lack of type inference may be a performance gain, it sacrifices ease of use for the programmer. Java is very rigid compared to Go in this regard.

Java also does not carry support for LXC like Go does. So that means the premise of Docker would have to be modified to a more general solution, but there is no information that would tell how trivial or difficult that would be.

4 DockAlt with Python

On the surface, Python is very similar to Go so it presents itself as very good potential replacement to Go. Just like Go, Python utilizes duck typing which gives the programmer greater flexibility in development.

Python contains a very simple syntax and also contains many libraries like Java since it has been around since 1991. Where Python goes above and beyond Java in this regard is its direct support for LXC as an imported library. Utilizing this library would allow DockAlt to keep its LXC based implementation and also provide the benefits of having a stable language with a massive user base at its disposal.

Like Java and Go, Python also does garbage collection so it is memory-safe in all cases but some special examples of circular references. This fact, along with its ease of use and large number of libraries makes Python a worthy option at a reimplementation of Docker.

5 DockAlt with Groovy

Groovy is an object oriented language that was created in 2003 for the Java platform. However, it is more similar to Python in appearance and dynamically compiles to the JVM. Along with being a dynamic language, Groovy also has the capability to be statically compiled when prompted by the programmer. This is a nice feature that allows programmers to exploit the benefits of dynamic typing for the majority of their programming while still giving them the option to statically compile or specify types when they feel it is necessary.

Since Groovy was developed for the Java platform, it shares the advantages of Java, but also the disadvantages. Like Java, Groovy implements asynchronous methods well. However, the main disadvantage is that Java, and Groovy, do not support LXC natively which means that the programmer would have to create their own container implementation in order to keep Docker's design.

6 Conclusion

This report provided an overview of Docker which was implemented in the Go language and attempted to find a suitable replacement language if a programmer would want to implement his or her own DockAlt.

While Go seems like a suitable solution for Docker at this time, it does have its drawbacks as a language and Python seems to provide the best option if one were to re-implement Docker. Java and Groovy are great languages in their own right, but they do not offer LXC binding support like Python so implementing DockAlt could become very difficult, or at least much more difficult than it would be to just use Python. The performance benefits of Java over Python are more or less inconsequential in this application and Python is gaining speed as well. The option for dynamic or static typing in Groovy is a nice option to have, but it does not outweigh the downside that would be derived from having to find a replacement method to LXC. This makes Python the best choice from the three languages above for a DockAlt implementation .

7 References

Asay, Matt. "10 things you should know about Docker", http://www.techrepublic.com/blog/10-things/10-things-you-should-know-about-docker/.

"Docker and Go: Why Did We Decide to Write Docker in Go?" Docker and Go: Why Did We Decide to Write Docker in Go?, 07 Nov. 2013.

Graber, Stéphane. "Introducing the Python LXC API." Stephane Graber's Website, 12 Sept. 2012.

Groovy, *Groovy*, http://groovy.codehaus.org/

"Optional Typing in Groovy." Object Partners., 19 Aug. 2013.