Containerization support languages

Chen Chen

INTRODUCTION

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings. Containers isolate software from its surroundings, for example differences between development and staging environments and help reduce conflicts between teams running different software on the same infrastructure. Containers virtualize the operating system instead of hardware, containers are more portable and efficient. [1] Also multiple containers can run on the same time. Compared with virtual machine, container needs less RAM resources to deploy and execute. Docker is the company driving the container movement and the only container platform provider to address every application across the hybrid cloud. This report will force on the programming languages sides effect to built a Docker.

I. JAVA

Introduction

Java is a C-like, imperative computer programming language which forced on object-orient and perform same reuslt on different environments or machine. Java virtual machine is the tool Java used to achieve the goal that perform same result regradless the computer architecture. Java virtual machine provides interpretation for most of code and compilation for the code which used multiple times. Though Java is a C-like language, it has automatic memory management. Java implemented with static scope checking like other languages. Currently, lots of Java libraries and modules supports multiple threads software framework.

Advantages

Java's syntax derives from C and C++, and also has the some same attribute with C/C++. Java is a strong type language and has static type checking. This attribute attribute brings prevents typing error in compilation stage. This attribute brings easiness for programmers to write and debug code when developing Docker. With the support of Java virtual machine, compiled Java code can be run on different architecture computer system without recompilation, which means Java is portable and match with one of attribute of container. Also because the executation of compiled Java code used both interpretation and compilation, Java can be considered as a efficient programming language and also match with the one of attribute of container. Also static scope checking meets lots of programmers' logic while coding, which prevents bug from scope.

Disadvantages

Because Docker need the access from the OS layer, but Java virtual machine prevents programmers to access OS layer, this attribute brings disadvantage on Java. Also beacuse virtual machine needs lots of RAM resources to boot, and installation of Java virtual machine is necessary for Java's code to run, this attribute is incompatible with the design of container which needs small RAM resources to boot because container onle includes binary codes and libraries.

II. OCAML

Introduction

Ocaml is functional programming language derived from ML also implemented with object-orient features. Ocaml is a strong type language and has static type checking system. Ocaml also implemented with some features like patteren-matching and tail recursion optimization.

Advantages

Native code compiler is available from many platforms. Portability is achieved through native code generation support for major architectures: IA-32, X86-64(AMD64),Power, SPARC, ARM, and ARM64. [2] This attribute match with the one of feature of container. OCaml's optimizing compiler employs static program analysis methods to optimize value boxing and closure allocation, helping to maximize the performance of the resulting code even if it makes extensive use of functional programming constructs. Xavier Leroy has stated that ”OCaml delivers at least 50% of the performance of a decent C compiler”. [3] Also beacuse of libraries in Ocaml, which gives Ocaml the ability to access lower layer OS informations. The static type checking system help programmers eliminate bugs and problem at runtime. Unlike Java, Ocaml doesn't need to write down the type name, which is the feature Docker needs. Similar with Java, Ocaml can help programmers to catch bugs at compilation type. This attribute gives programmers convention to write down code without cumbersome type name, and programmers obtains the benefits from static type checking.

Disadvantages

Though Ocaml compiler can compile fast machine code, Ocaml has poor multiple threads and cores support, which means poor performance on nowadays computer. Ocaml's poor support on polymorphism, which brings inconvenient on large project like Docker. Third, Ocaml is heavy based on functional paradigm while programmin. Compared to most of programming language, Ocaml is freaks. It will cost lots of time and resources to change from currently popular programming thinking mode from Ocaml's thinking mode.

Becasue of this issue, it is hard to recruit programmers who capable of such working.

III. Rust

Introduction

Rust is a systems programming language, supporting functional and imperative-procedural paradigm. [4] Rust has C-like syntax. Rust doesn't use automatic garbage collection system, instead similar with C++ Rust use ”resources acquistion is intialzation” to manage memory and other resources. Unlike C/C++, Rust doesn't permit null pointers and dangling pointers, which prevent dangling pointers and other undefined behavior. Rust also is a strong type language and has static type checking.

Advantages

Beacsue the goal of Rust is to be a language for highly concurrent and highly safe for system, [5] this attribute match perfectly match with one of the goal of Docker. Like Java, static type checking system help programmers debug at compile time. Like C++, Rust takes zero-cost abstractions as one of its core principles: none of Rusts abstractions impose a global performance penalty, nor is there overhead from any runtime system in the traditional sense. [6] This design gives Rust the ability to run on machine as run as C/C++. Becasue of the support of LLVM, Rust does cross-compilation, which allow Rust code to run on different platform.

Disadvantages

Rust is not a popular programming language compared to C++/Java. The Ecosystem is not flourished as C++/Java, which means company probably need to develop software framework for Docker. Compared to C++/Java, the time cost to develop is much longer. Second, programming in Rust is extreme verbose, even much compilcated than C++, because Rust compiler need specifications on every action in order to compile to machine code. Mixed with the first disadvantage, it extend the development time even longer. It probably need decades to develop a mature software framework, beacuse Rust compared to other programming is a young programming language. Third, long compilation time because of verbose syntax incompatible with current software developing such as continuous integration. The compilation time for large project like Docker probably takes more than one day for programmers to get the test result, which also extend the development time.

References

[1] “What is a container,” Sep 2017. [Online]. Available: https://www.docker.com/what-container

[2] Ocaml, “ocaml/ocaml,” Nov 2017. [Online]. Available: https://github.com/ocaml/ocaml/tree/trunk/asmcomp

[3] X. Leroy, “Caml weekly news.” [Online]. Available: https://lwn.net/Articles/19378/

[4] “The rust programming language.” [Online]. Available: https://www.rust-lang.org/

[5] “Interview on rust, a systems programming language developed by mozilla.” [Online]. Available: https://www.infoq.com/news/2012/08/Interview-Rust

[6] “Frequently asked questions.” [Online]. Available: https://www.rust-lang.org/en-US/faq.htmlhow-fast-is-rust