Homework 6. Containerization support languages

Motivation

Following up on the project, suppose you are trying to run a Twisted Places proxy herd on a large set of virtual machines. To keep costs low, you decide to standardize on Linux kernels and use operating system-level virtualization with Linux Containers (LXC). To manage your deployment, you'd like to use Docker.

Your boss is a bit worried about Docker, partly because it's so new, and partly because there is a single source for it—if the main version of Docker is buggy, you'll be toast. She wonders whether it'd be feasible to come up with a separate implementation for Docker, one written in a completely different programming language, so that your team will have the freedom to choose either the standard implementation or the alternative one (let's call it DockAlt), and switch to the alternative if problems crop up in the standard one.

Docker is implemented using the Go programming language, and its developers have explained why they chose Go in their talk Docker: an insider view.

Assignment

Review the insider-view talk, and consider three other languages that are plausible candidates for implementing DockAlt: (1) Java, (2) OCaml, and (3) a language taken from the following list.

- Coconut
- Dart
- Hack

Do some research on your three languages and support software as a potential platform for DockAlt. Your research should include an examination of the language and system documentation to help determine whether it would be an effective way to implement DockAlt.

Unlike the project, we are not expecting working prototypes, though prototypes are welcome.

Write an executive summary that compares the three alternate approaches to each other and to Docker. The summary should be in 10-point font or larger and should be at most three pages. You can put references and appendixes in later pages, if there's not enough room on four pages: the appendixes should contain any source code or diagrams. Your summary should focus on the technologies' effects on ease of use, flexibility, generality, performance, reliability; thie idea is to explore the most-important technical challenges in doing the proposed rewrite. The summary should be suitable for software executives, that is, for readers who have some expertise in software, particularly in managing software developers, but who are not experts in Java or OCaml or your chosen language. Please keep the resources for written reports and oral presentations in mind, particularly its rubrics and its advice for citations to sources that you consulted.

Submit

Submit a file `hw6.pdf` containing your summary. If you have prototype code, submit it as a compressed tarball `hw6.tar.gz` in the usual way; it should include a README file that contains instructions for how to build and run the prototype.

$Id: hw6.html,v 1.53 2016/11/22 23:22:40 eggert Exp $