Improvement to an Open Source Project Solo5

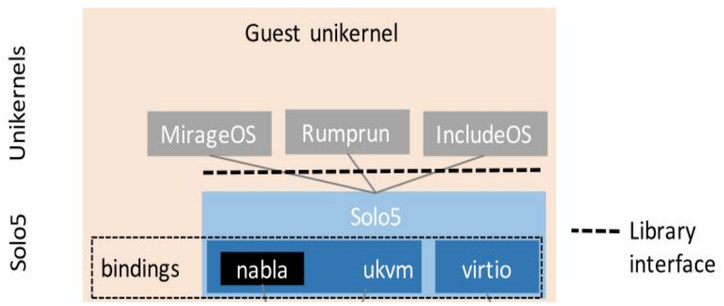
YanzhiLi yli16@earlham.edu)
Department of Computer Science,
Earlham College, Richmond, IN

Introduction

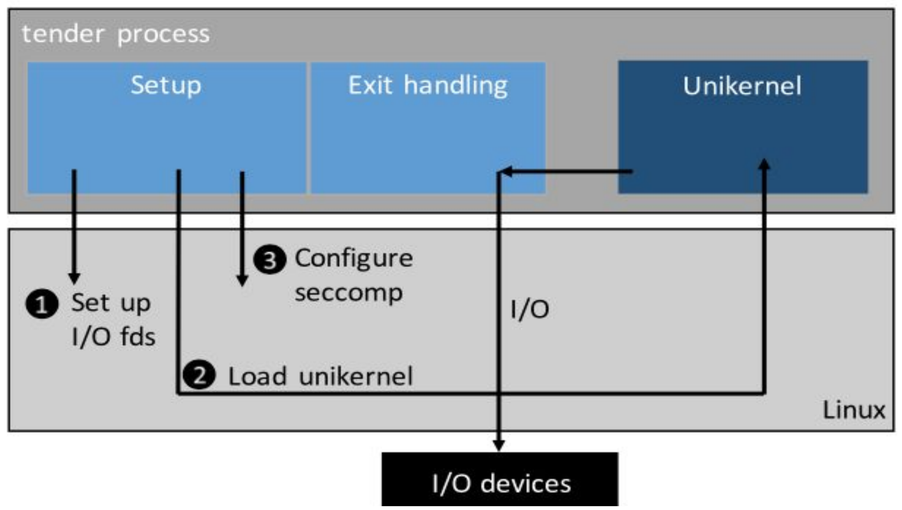
Unikernels have emerged as a fast, smalr and secure system that reuses VM isolation while also being light- weight by eliminating the general purpose OS from the VM.
Running unikernels as processes on diverse host operating systems and hypervisors is a practical way to take their advantage to run applications built using various unikernels on traditional platforms. (e.g., Linux)
Solo5 appeared as a general sandboxed execution environment that works as the interface between the host kernel and the unikernels. This study explored the Solo5 open source project and investigated three issues of the project.

Solo5 Overview

Solo5 is a sandboxed execution environment primarily intended for running applications built using various unikernels.
Solo5 introduces the concept of a tender, a process responsible for loading the unikernel code into its own address space.There are two primary tenders hvt (hardware virtualized tende) and spt (sandboxed process tender), which both were known as ukvm monitor.



Solo5 Structure



Solo5 Issues

This study investigated three open issues on the Solo5 project and solved two of them.

Enable the system to be built with GCC 10.x :
By modifying the configure script, the Solo5 system could now be built with GCC >9 on Linux.

Stop host system from executing Solo5 binaries directly on Linux;
By adding a PT_INTERP in binding-specific linker scripts, which points to a trailing slash /nonexistent/solo5/. The host kernel is not allowed to execute Solo5 binaries directly, which avoids the risks of exposing the host kernel and provide sensible error messages to users.

Add proper support for cross-compiling:
The ARM 64bit toolchain was used to test cross-compilation of Solo5. However, after careful investigation, it turned out that this issue is far beyond the scope of the project for two reasons. First, we could not find a way to override the existing toolchains used by the tender without crashing the compilation. Second, a comprehensive modification to the configure script and other files is too compliation to complete.

RESULTS & Future Work

A Debian virtual machine on VirtualBox was used to build the Solo5 system. With Opam, the unikernel MirageOS was loaded into Solo5. A hello-world program was built on MirageOS. This project successfully built application using an uniknerl implementation with the interface Solo5 on Linux. Solo5 has many limitations and requires long term efforts to develop. Solo5 has the potential of introducing unikernels to different platforms. The next step is to transform the experimental targets into productive targets. We will continue to contribute to the project and work on the open issues.

Acknowledgements

I would like to thank Prof.Xunfei for her help and advice for my project and Prof. Charlie for his suggestions on unikernels.

References

This poster is based on “Building MirageOS on Linux and Improvement to an Open Source Project Solo5”, Yanzhi Li, available at hhttps://portfolios.cs.earlham.edu/wp-content/uploads/2020/11/Yanzhi_paper.pdf