

NetServ JUNOS Port Project

Krishnan Rajeswar (kr2428) with
Jae Woo Lee

Learning Curve - NetServ

- NetServ Architecture (reading related white papers)
- NetServ Code with Jae
- JNI interface used to invoke JVM from native code.

Learning Curve – JUNOS Architecture

- Setup the Juniper M7i router at the CRF lab.
- Online and Live Training for JUNOS SDK development.
- Documentation

JVM on JUNOS

- The key requirement for porting NetServ
- invoke.c – Sample program to test JNI interface
- On RE or Services Plane?

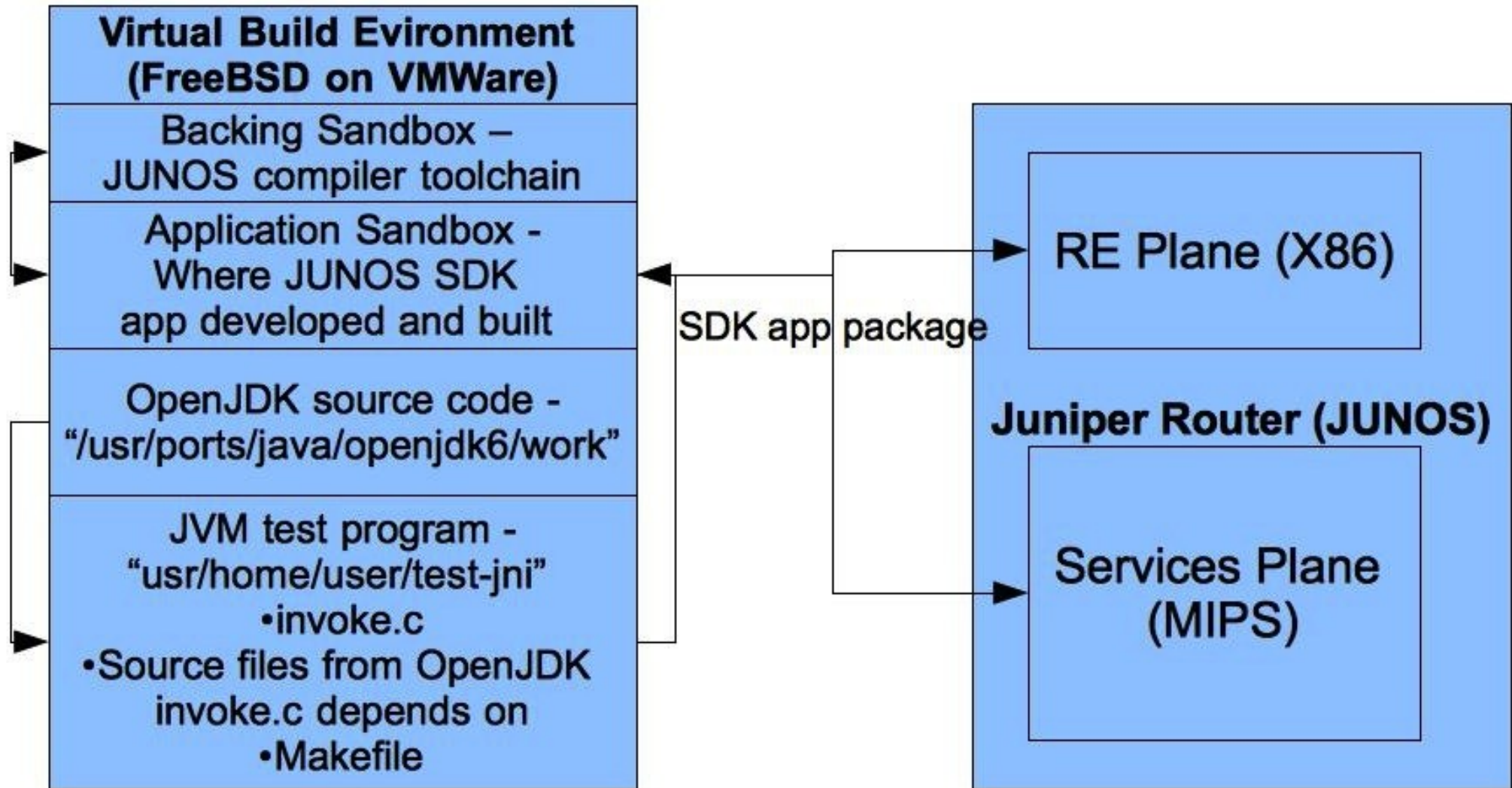
libjvm.so

- To invoke JVM from native code, libjvm.so is the required library.
- Tried to include the existing libjvm.so as an external library in a JUNOS build – Does not work as expected.
- The solution – to re-build the library from its source

libjvm.so from source

- Considered all options available – Native JDK, Linux JDK, Diablo JDK and OpenJDK
- OpenJDK only viable option
- Complex build process

Thousand Words



Contribution and Future Work

- Resolved thousands of dependencies and found 500+ source files to build invoke.c successfully.
- Comprehensive documentation.
- Compiling invoke.c using JUNOS tool chain. On success deploy it as an RE application
- OpenJDK MIPS support
- Juniper Virtual Engine Environment

What I learnt?

- How to approach a fairly complex problem
- Understanding makefiles and build process of an application.
- Resolving tricky dependency issues.