Embedded Linux Conference Europe

Building a Network Operating System using Linux and Yocto

John Mehaffey

aruba
a Hewlett Packard Enterprise company

THE LINUX FOUNDATION

Introduction

ArubaOS-CX (Code name Halon) is a Network Operating System (NOS) based on Openswitch.

It uses Linux as its base (kernel and userland), and Yocto as the build system.

This talk will discuss the challenges of combining the three open source technologies.

Linux Kernel

aruba
a Hewlett Packard
Enterprise company

Kernel Size

The Linux kernel is extremely configurable, the trick is finding the correct combination of features and tables that accomplish the work needed.

- Start small, work up
- Use Yocto layers to add platform dependent features
- Aggressively trim dependent kernels (e.g. kexec)
- Use busybox, add real packages only when necessary
- Share RAM for functions needed at different times
  - e.g. packet buffers, core dumps

Kernel Updates

Kernel updates are one of the most important and most intrusive functions of Devops. Openswitch started with kernel 4.4, and Halon migrated to all of the LTS kernels available in Yocto along the way.

Kernel Updates (continued)

Kernel updates provide critical bug fixes, CVE mitigations, and new features, but also provide a convenient excuse for problems.

Ex: upgrading from 4.4 to 4.9 provided 18 (!) bugs blamed on the kernel. 9 were actually code/design flaws exposed by the upgrade, only 2 were actual kernel issues.

Kernel Updates (continued)

Management perception is guided by initial reports, and resulted in questions about why are we upgrading if it breaks our code.

Lots of time spent explaining perception vs reality, but it set the stage for future upgrades, which went more smoothly.

Yocto

aruba
a Hewlett Packard
Enterprise company

Yocto Layers

Yocto layers are used extensively in Halon:

- Poky layers: for the basics
- Openembedded layers: per-subsystem
- Meta-foss: opensource packages
- Halon-common: boot loader and OS
- Halon-distro: for all Halon platforms
- platform-dependent

Yocto Upgrades

Yocto upgrades are a lot harder to justify than kernel upgrades.

- Management always wants to delay painful issues.
- Must be persistent. Make sure you are ready with upgrades before major branches, to go in before things settle.

Yocto Upgrades

aruba
a Hewlett Packard
Enterprise company

Subsystem upgrades

aruba
a Hewlett Packard
Enterprise company

Subsystem Upgrades

Upgrades of large subsystems (ASIC SDKs, Metaswitch, etc.) have similar issues to other upgrades mentioned already

- Use merge commits
- Make sure you save the SHA before/after
- Useful to have images before/after for triage/comparison

Build

aruba
a Hewlett Packard Enterprise company

Build Issues

Halon uses:

- 527 poky packages
- 122 foss packages
- 276 kernel modules
- 518 custom packages
- 2 architectures/8 variants

Build Issues

Build times varied widely throughout development, between 10 minutes and 4 hours.
Many experiments were tried to reduce build times.
Biggest remaining issue is DEPENDS chains that cause extensive rebuilds.

PCI

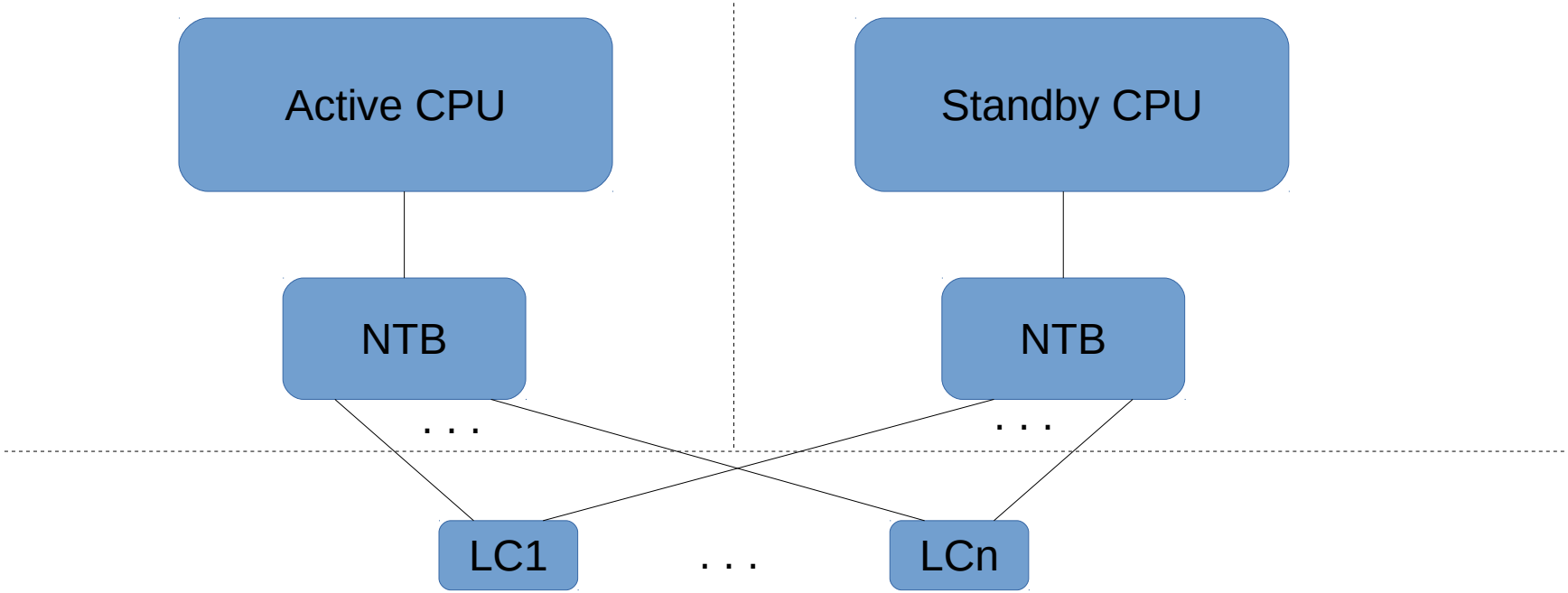
aruba
a Hewlett Packard
Enterprise company

PCI Issues

PCI turned out to be one of our biggest engineering issues, due to HA requirements.

The architecture requires the modular line cards to remain up (passing traffic) during CPU failover events.

PCI Architecture



PCI Issues

- LCs have CPUs running transport layer. TL would starve and timeout during periods of high DMA traffic on PCI links.
  - Needed fair share algorithm.
- PCI drivers would miss interrupts and lock up.
  - Use polling
- Active/Standby failover would cause other issues
  - Reset links on failover. Changed to EoP. KISS!

Embedded Linux Conference Europe