LinuxBoot and coreboot at Google

Ron Minnich, Staff Software Engineer, Google
Current state

- LinuxBoot effort started by Google Jan 2017
  - Linux kernel + u-root initramfs in FLASH
- Deployed at scale at Google by December 2020
- PoR for future servers and “servers”
- Deployed at other hyperscalars and companies too numerous to name
- Facebook/Intel coreboot partnership across 3 chipsets presented at OCP
- Google has made coreboot Plan of Record for future systems of any kind
  - Has been Plan of Record for chromebooks for 10 years
- coreboot@ByteDance: https://inf.news/en/tech/73e123f55bf760adcd2d7d3fd6e91f82.html
- Multiparty NDA coreboot efforts on various chipsets from various vendors
How we create LinuxBoot on UEFI

- Open source potentially open source, delivered as a blob
- Closed

- SEC/PEI is about 10%

- Security SEC
  - Pre-EFI PEI
  - Drivers DXE
  - Boot Device Select BDS
  - Transient System Load TSL
  - Run Time RT

- Intrinsic

- Dispatcher
  - Boot Manager
    - OS-Absent App
    - Transient OS Env
    - Transient OS Boot Loader
    - UEFI shell
    - Final OS Boot Loader (e.g. grub)
    - Runtime kernel

- OS-present App (who loads this?)

OPEN POSSIBILITIES.
LinuxBoot one UEFI: reduce & replace

You can see why we originally called it NERF:
Non-Extensible Reduced Firmware

You can see why we originally called it NERF:
Non-Extensible Reduced Firmware
LinuxBoot on UEFI results

- LinuxBoot in UEFI systems deployed in aerospace, cloud, mfg in 2017
- Deployed in at least one hyperscalar in 2018
- Deployed in APAC companies 2019
- Deployed at scale in Google 2020
- Deployed at ByteDance in 2021
- Works across x86 and ARM (see: LBBR)
- Can use same LinuxBoot binary on Intel and AMD
- LinuxBoot allows us to “reduce and replace” much of UEFI
- coreboot allows us to “remove and replace” the rest of UEFI
coreboot

- Started in 1999 as LinuxBIOS at Los Alamos National Labs (LANL)
- Literally that -- Linux as your BIOS
- Two parts:
  - Small startup code, purposely design to do as little as possible -- “Let Linux Do It”
  - On one chipset I got it down to 200 bytes
  - Linux kernel
- Used in several Top 10 supercomputers in early 2000s
- German gov’t sponsored laptop support 2006-2008
- Used in x86 chromebooks since 2011 and ARM chromebooks since 2014
- Tesla picked it up in (??) 2017?
Google’s path to coreboot

- Google UEFI modifications
  - Proprietary BIOS with modifications to chipset blobs
- Chipset vendor blobs
  - Restrictive licensing
    - Almost no common code

  - Linux kernel burned into flash
    - Drivers reused from as needed
      - About 50% common code across a single architecture

- Linux Kernel Drivers
- U-root userspace
- Reduced proprietary BIOS
- Chipset vendor blobs

- EDK2 (partly open source)
- Chipset vendor blobs

- More minimal init provided by CPU vendor
  - More minimal init provided by CPU vendor
  - About 50% common code across all architectures

- Coreboot
  - Mostly open source
    - All open standards
    - Community can do more
      - Common platform across NPIs
      - Higher confidence in security
      - About 90% common code across all architectures
ByteDance path to coreboot – parallel evolution!
How coreboot improves security

- Statically linked
  - critical to security analysis
- Open source code base
  - *NOT* “many eyes” -- that’s known to fail
  - “Many highly skilled eyes” -- that’s known to work
- Twenty years of hardening in the open community
- Imported into Coverity over ten years ago
- Hardened for Chrome OS (consumer devices) for 11 years
How coreboot improves quality

- Compact codebase
- Large set of contributors
- 90% of code is common to all boards
  - “Fix one, fix all” is a common experience
- Full build for 252 mainboards for each commit
- One codebase for 252 mainboards, 5 architectures
How coreboot helped one project at Google

- Board had 228 DXEs (UEFI drivers)
- We removed 122
- Some removals seemed to work, had adverse impact later
  - Nobody from chipset vendor, board vendor, or Google could ever be sure
- We had a real struggle to free up even a few MiB
- UEFI was getting to be difficult to work with
Moved to coreboot and Intel FSP

● Freed up 10 MiB instantly
● Reduced number of fingers in firmware pie
● Allowed us to really understand what was going on
● enhanced security
● faster boot
Conclusion

● LinuxBoot started out as a UEFI project
● Allowed us to remove a lot of UEFI
  ○ Led companies to question: “do we need UEFI?”
● Companies started to push hard for coreboot
● And here we are today:
  ○ Intel SPR: facebook, Intel, ByteDance, WiWynn, … Google
  ○ AMD Rome: Google, Datacom Electronics GmbH, others (not sure I can name them)
● Surprise: making a project explicitly UEFI+LinuxBoot led to coreboot+LinuxBoot
Call to Action

• open system firmware slack: https://slack.osfw.dev/
• coreboot is available for OCP DeltaLake server based on Intel Xeon Scalable processor
  ○ see the OCP Github for open system firmware (OSF)
  ○ https://github.com/opencomputeproject/OpenSystemFirmware
    ■ /tree/master/Wiwynn/deltalake
  ○ With more to come in 2022!
• linuxboot.org
Thank you!