POWER9 Support in coreboot

OpenPOWER Summit NA 2020

Michał Żygowski

3MDEB

Agenda

- whoami?
- Who is involved?
- Why coreboot?
- Use cases
- Development roadmap
- Future plans
- Dasharo
- Validation with RTE
- Discussion
- Q&A

whoami?

Michał Żygowski
Firmware Engineer

- @_miczyg_
- michal.zygowski@3mdeb.com
- linkedin.com/in/miczyg
- facebook.com/miczyg1395

- Braswell SoC, PC Engines and Protectli maintainer in coreboot
- interested in:
  - advanced hardware and firmware features
  - coreboot
  - security solutions

Firmware is class of software

Firmware is class of software that helps in low-level initialization and control of your hardware

- **OEMs and ODMs** depend on firmware to provide secure, compatible and high-performant environment for customers OSes and applications
- **Cloud providers** depend on firmware to use advanced platform security and provide trusted and hardened VMs in their infrastructure
- **Service providers** depend on firmware to efficiently leverage security, runtime and management capabilities in their solutions
- **AI/ML/Analytics companies** depend on firmware to fully utilize underlying hardware and tune it to particular workload

Why coreboot?

Initialize the necessary hardware as fast as possible and boot to Linux

This is the goal of both OpenPOWER firmware and coreboot. So why not use it? coreboot always was a good replacement or alternative for the platform firmware.

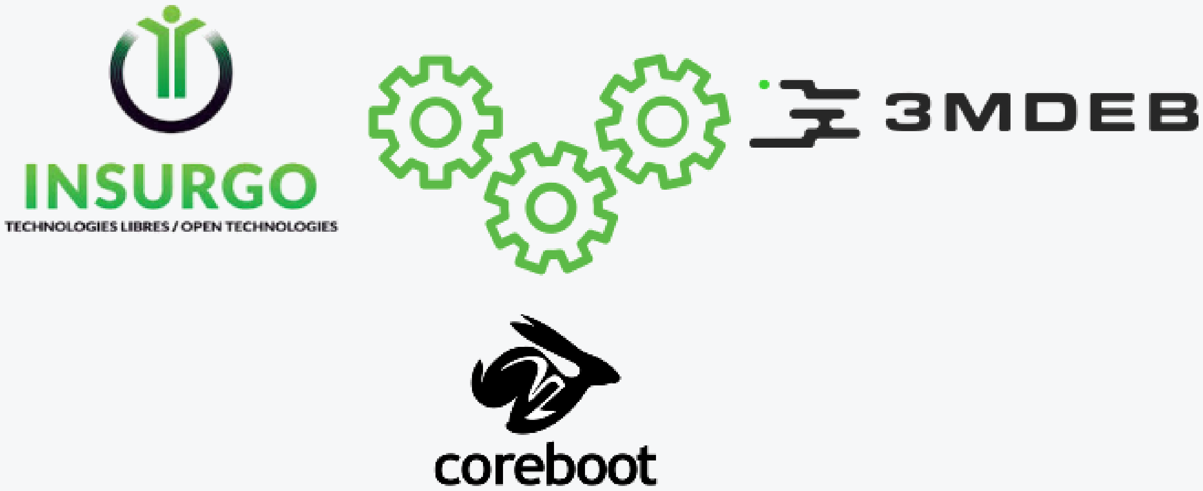
Who uses coreboot?

- Google (Chromebooks) https://chromium.googlesource.com/chromiumos/third_party/coreboot/
- Siemens (industrial devices)
- Tesla Motors https://github.com/teslamotors/coreboot
- Supermicro servers
- Various Lenovo laptops
- PC Engines (network appliance devices)

Why coreboot?

- coreboot is well recognized brand for open source firmware
- boot speed (coreboot should be faster, see "Goals" here: https://wiki.raptorcs.com/wiki/Coreboot/ToDo)
- programming language - coreboot C vs. OpenPOWER firmware C++
- one firmware for all platforms and architectures
- both projects seem to use Gerrit for code review, but coreboot has an public instance, while POWER uses internal one (thus it is not clear how to contribute)
- it is easier to find coreboot developers than OpenPOWER firmware developers

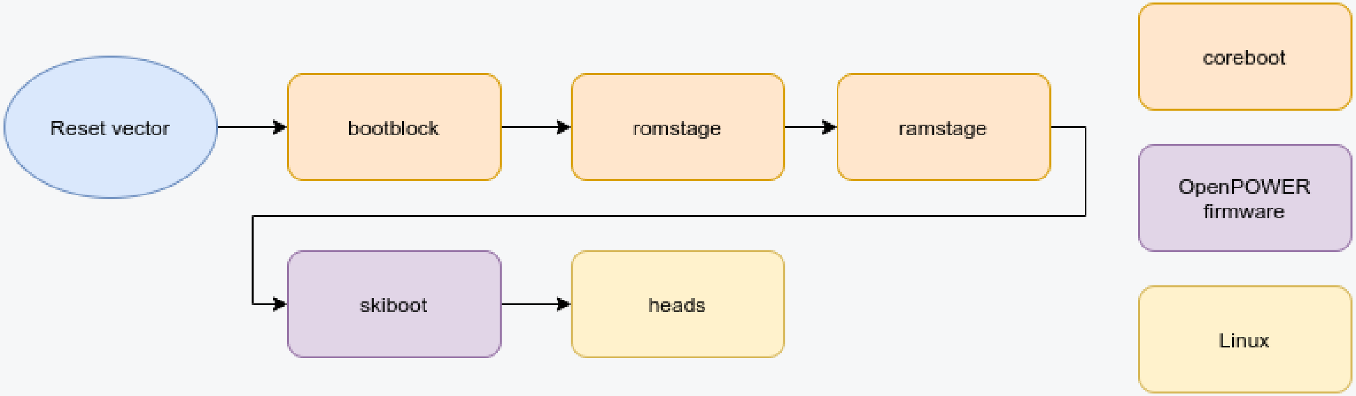
Who is involved?



3mdeb loves coreboot

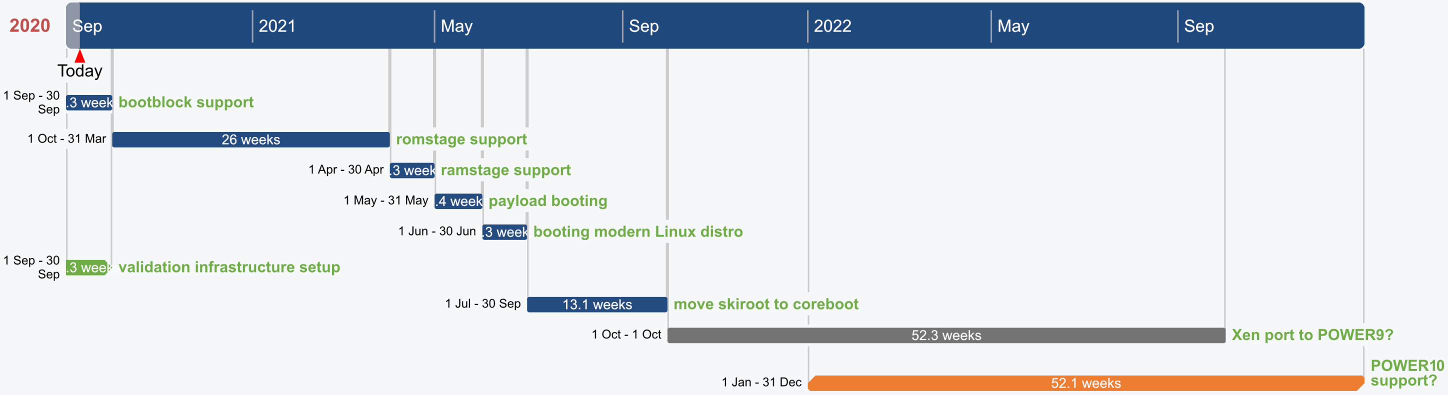
- strong presence in boot firmware for network appliance and firewall devices
- SRTM and DRTM firmware development
- TPM2.0 enabling
- IOMMU and other advanced hardware features enabling
- low level validation integration (BITS, CHIPSEC)
- Open Source Firmware training and workshops
- GRUB and QubesOS minisummit organizers
- active in Open Source Firmware community for 5+ years

Use cases



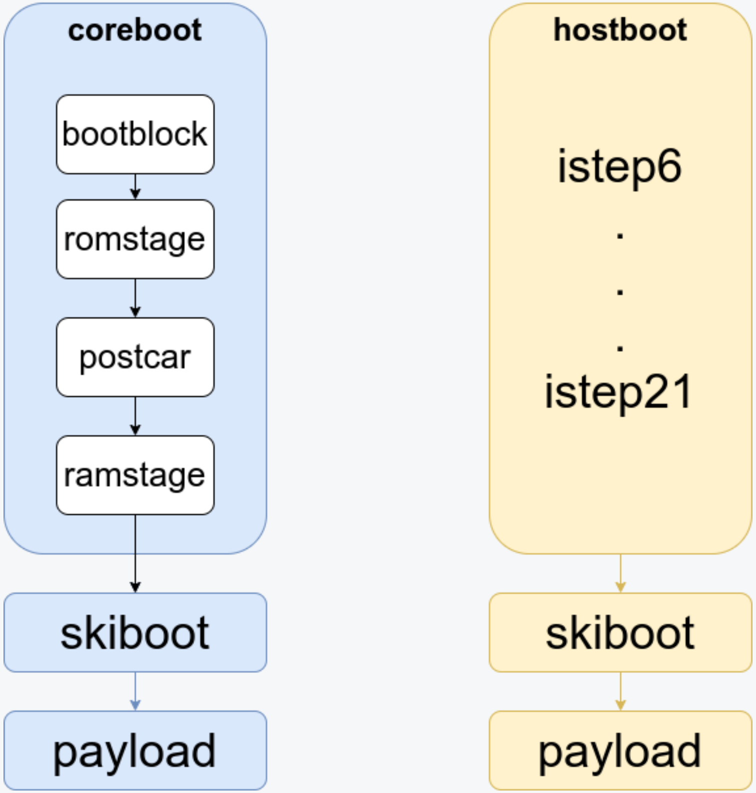
- coreboot + skiboot + heads
  - simple replacement of hostboot with coreboot
  - faster boot time
  - heads security model

Development roadmap



- The port will cover the hostboot part only (for now).
- How it fits into the coreboot booting and device model?
- Mapping isteps to coreboot stages

Development roadmap



coreboot

bootblock

romstage

postcar

ramstage

skiboot

payload

hostboot

istep6

.

.

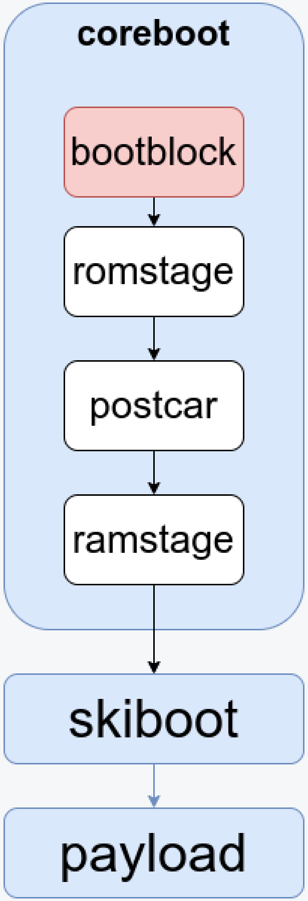
.

istep21

skiboot

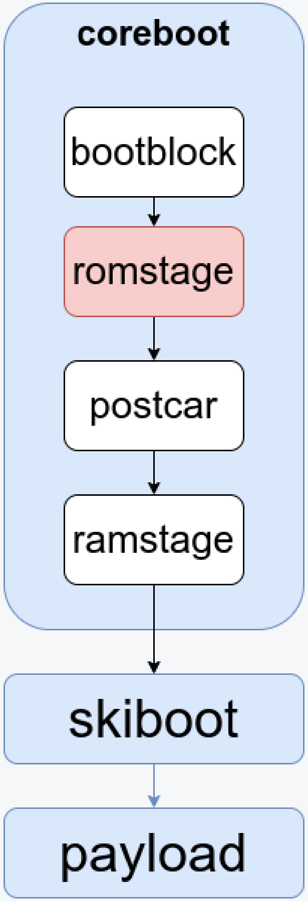
payload

Development roadmap



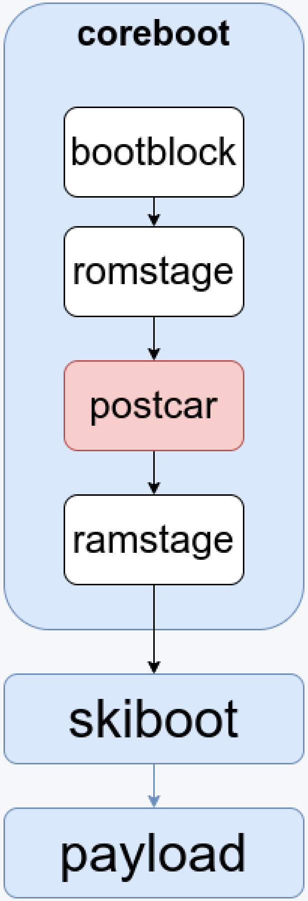
- x86 boot phase case:
  - setup temporary memory in cache
  - setup stack to run C code
  - perform very basic hardware initialization
  - setup debugging interface
- POWER9 boot phase case:
  - istep 6
  - handle the state left by SBE
  - setup registers to run C code
  - setup debugging interface

Development roadmap



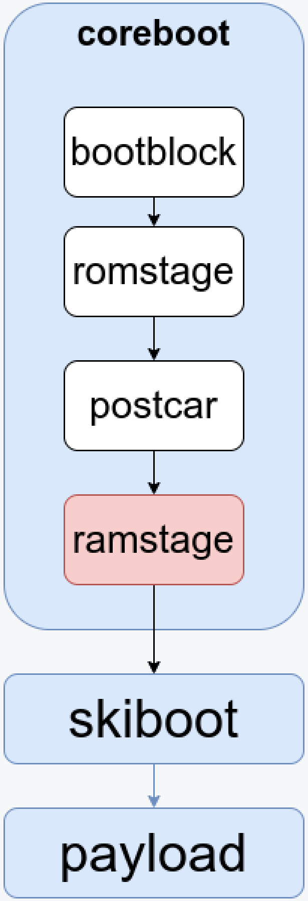
- x86 boot phase case:
  - perform necessary hardware initialization to train memory
  - setup and train main memory
  - setup cbmem
- POWER9 boot phase case:
  - isteps 7-14
  - configure Nest
  - configure Fabric
  - perform XBUS training
  - perform SMP initialization
  - perform DDR4 training
  - setup cbmem

Development roadmap



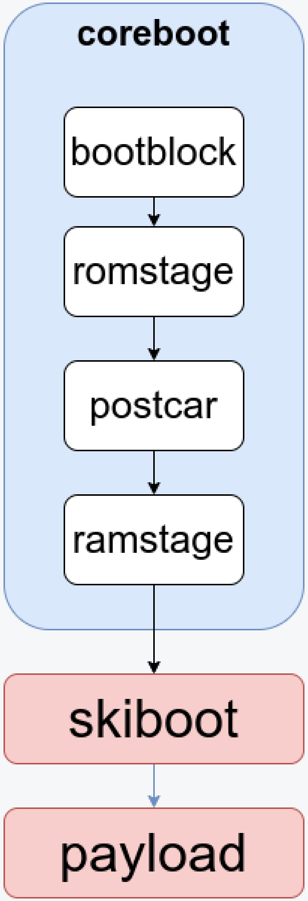
- x86 boot phase case:
  - tear down the temporary memory environment in cache
  - prepare to run code from DRAM
- POWER9 boot phase case:
  - rather nothing to do in particular

Development roadmap



- x86 boot phase case:
  - initialize devices
  - construct SMBIOS and ACPI tables
  - finalize hardware initialization by locking registers
  - check for OS resume and resume if needed
  - load and execute payload
- POWER9 boot phase case:
  - isteps 15-21
  - initialize devices
  - construct devicetree for OS consumption
  - load and execute payload (skiboot)

Development roadmap



- x86 boot phase case:
    - one of target applications (SeaBIOS, GRUB, UEFI payload, Linux kernel, simple ELF) which can boot another OS/application
- POWER9 boot phase case:
    - skiboot as a 1st stage payload
    - skiboot performs the rest of hardware initialization, exposes OPAL
    - 2nd stage payload: petitboot or equivalent kexec based payload (e.g. heads) to load target OS

Future plans

- **port skiboot** related hardware initialization **to coreboot**
- **Xen** port to **POWER9** https://github.com/QubesOS/qubes-issues/issues/4318
- **POWER10 support ~2022**
- **support** firmware update via **fwupd/LVFS** for OpenPOWER based machines
- offer **specialized solutions** for OpenPOWER **with Dasharo** boot firmware technology
- **boost knowledge** about coreboot and OpenPOWER with **trainings** to **widen the group of specialists** and **attract world with POWER** architecture

Dasharo reflects castling

This unique move in chess provides various opportunities for a wise chess master. It opens the rook's position, but most of all: **protects the king**. That's what essentially Dasharo does. It provides security in a way that may look simple, but its complexity is enormous - just as the impact.

Dasharo is a new BIOS firmware technology that brings solutions to the problems of ownership, performance, security and compatibility, allowing to create secure and efficient images that can be fully customizable to your product.

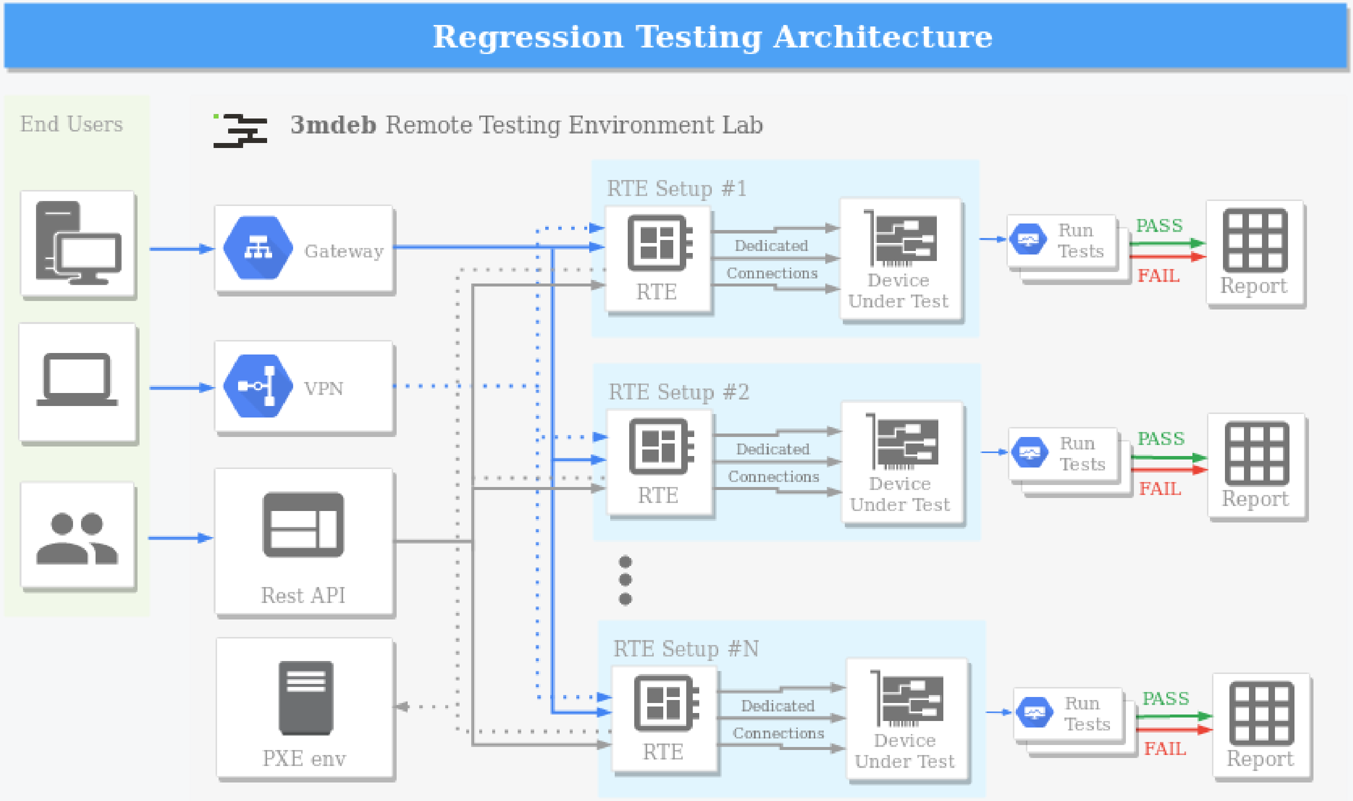
Dasharo Modularity

Security
Performance
DASHARO
Compatiliby
Marketing

Four modules of Dasharo:

- **Security Module** - features that make your hardware trustworthy
- **Performance Module** - hardware performance optimization features
- **Compatibility Module** - maintenance features
- **Marketing Module** - brand awareness and customer value features

Validation with RTE



Discussions

- What issues do you see in our plan?
- How could OpenPOWER benefit more from coreboot?
- Do you see any problem with current OpenPOWER firmware?
- Is there any need for ISV providing constant support in the firmware field?

Q&A