Speedy Distro Porting via the `cpu` Command

Daniel Maslowski

Introduction

Thank you, it’s good to be back!

Hi, I’m Daniel!

- professional app and web developer
- been to openSUSE Conference many times
- hacking on firmware and operating systems
- started to write code in Go and Rust

Thank you, it’s good to be back!

Hi, I’m Daniel!

- professional app and web developer
- been to openSUSE Conference many times
- hacking on firmware and operating systems
- started to write code in Go and Rust

Remember? In 2019, we talked about open source firmware!

Thank you, it’s good to be back!

Hi, I’m Daniel!

- professional app and web developer
- been to openSUSE Conference many times
- hacking on firmware and operating systems
- started to write code in Go and Rust

Remember? In 2019, we talked about open source firmware!

Agenda

- Distributing an OS
- Porting Firmware
- Speeding Things up

Distributing an OS

Building Software

Building Software

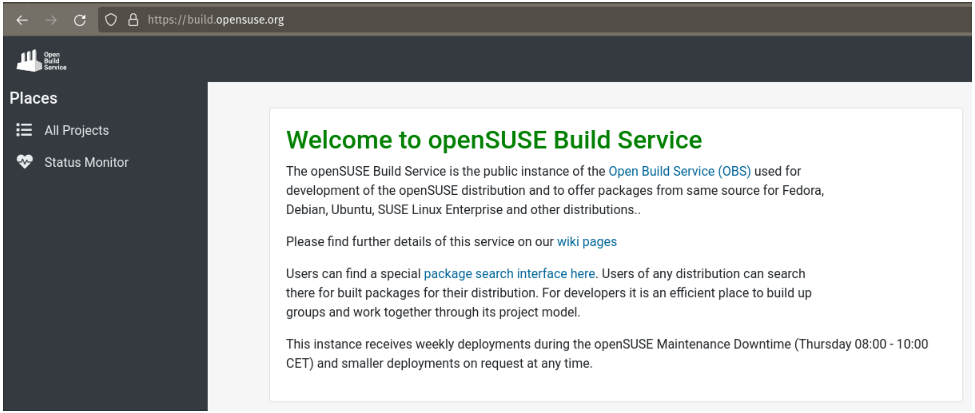
For distribution, software needs to be built, by the distro or the end user.

Building Software

For distribution, software needs to be built, by the distro or the end user.

Building software requires toolchains, meeting assumptions, patching.

Hello OBS!



Welcome to openSUSE Build Service

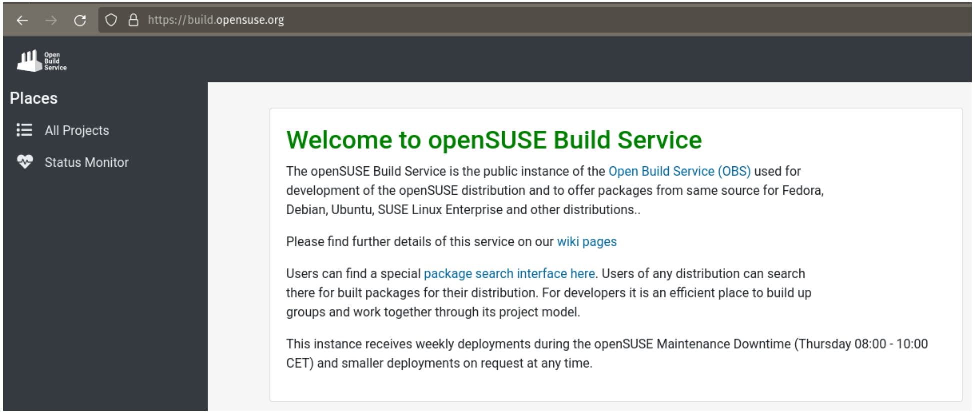
The openSUSE Build Service is the public instance of the Open Build Service (OBS) used for development of the openSUSE distribution and to offer packages from same source for Fedora, Debian, Ubuntu, SUSE Linux Enterprise and other distributions..

Please find further details of this service on our wiki pages

Users can find a special package search interface here. Users of any distribution can search there for built packages for their distribution. For developers it is an efficient place to build up groups and work together through its project model.

This instance receives weekly deployments during the openSUSE Maintenance Downtime (Thursday 08:00 - 10:00 CET) and smaller deployments on request at any time.

Hello OBS!


https://build.opensuse.org

Welcome to openSUSE Build Service

The openSUSE Build Service is the public instance of the Open Build Service (OBS) used for development of the openSUSE distribution and to offer packages from same source for Fedora, Debian, Ubuntu, SUSE Linux Enterprise and other distributions..

Please find further details of this service on our wiki pages

Users can find a special package search interface here. Users of any distribution can search there for built packages for their distribution. For developers it is an efficient place to build up groups and work together through its project model.

This instance receives weekly deployments during the openSUSE Maintenance Downtime (Thursday 08:00 - 10:00 CET) and smaller deployments on request at any time.

Building as a Service - BaaS

Processors and Architectures

Processors and Architectures

Software on lower levels involves platform specifics.

[1] https://azrael.digipen.edu/~mmead/www/Courses/CS180/OSOverview.html

Processors and Architectures

Software on lower levels involves platform specifics.

The kernel is commonly the lowest level part of an operating system[1].

[1] https://azrael.digipen.edu/~mmead/www/Courses/CS180/OSOverview.html

Processors and Architectures

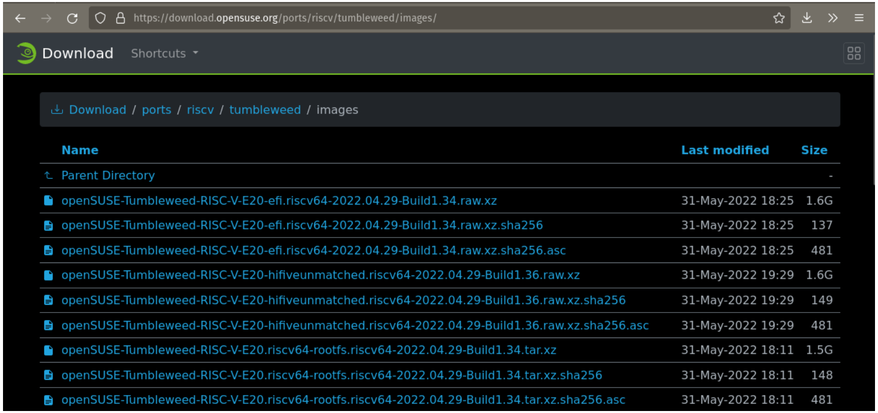
Software on lower levels involves platform specifics.

The kernel is commonly the lowest level part of an operating system[1].

One kernel, one distro image per architecture.

[1] https://azrael.digipen.edu/~mmead/www/Courses/CS180/OSOverview.html

Wait, what happened?


https://download.opensuse.org/ports/riscv/tumbleweed/images/

Download / ports / riscv / tumbleweed / images

Name	Last modified	Size
Parent Directory		-
openSUSE-Tumbleweed-RISC-V-E20-efi.riscv64-2022.04.29-Build1.34.raw.xz	31-May-2022 18:25	1.6G
openSUSE-Tumbleweed-RISC-V-E20-efi.riscv64-2022.04.29-Build1.34.raw.xz.sha256	31-May-2022 18:25	137
openSUSE-Tumbleweed-RISC-V-E20-efi.riscv64-2022.04.29-Build1.34.raw.xz.sha256.asc	31-May-2022 18:25	481
openSUSE-Tumbleweed-RISC-V-E20-hifiveunmatched.riscv64-2022.04.29-Build1.36.raw.xz	31-May-2022 19:29	1.6G
openSUSE-Tumbleweed-RISC-V-E20-hifiveunmatched.riscv64-2022.04.29-Build1.36.raw.xz.sha256	31-May-2022 19:29	149
openSUSE-Tumbleweed-RISC-V-E20-hifiveunmatched.riscv64-2022.04.29-Build1.36.raw.xz.sha256.asc	31-May-2022 19:29	481
openSUSE-Tumbleweed-RISC-V-E20.riscv64-rootfs.riscv64-2022.04.29-Build1.34.tar.xz	31-May-2022 18:11	1.5G
openSUSE-Tumbleweed-RISC-V-E20.riscv64-rootfs.riscv64-2022.04.29-Build1.34.tar.xz.sha256	31-May-2022 18:11	148
openSUSE-Tumbleweed-RISC-V-E20.riscv64-rootfs.riscv64-2022.04.29-Build1.34.tar.xz.sha256.asc	31-May-2022 18:11	481

Revisting Assumptions

Revisting Assumptions

Many chip vendors and multiple architectures imply fragmentation.

Revisting Assumptions

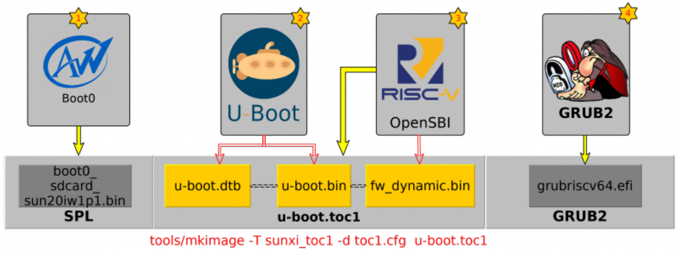
Many chip vendors and multiple architectures imply fragmentation.

What may an OS safely assume? (our core question)

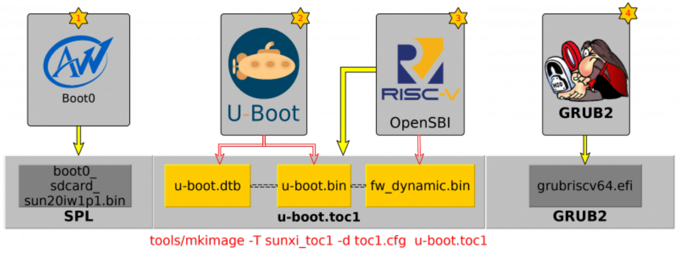
Porting Firmware

Hello RISC-V!

Hello RISC-V!

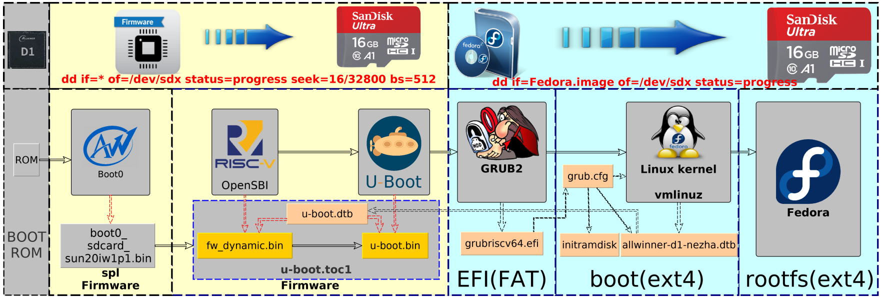

1
Boot0
2
U-Boot
3
RISC-V
OpenSBI
4
GRUB2
boot0_
sdcard_
sun20iw1p1.bin
SPL
u-boot.dtb
u-boot.bin
fw_dynamic.bin
u-boot.toc1
grubriscv64.efi
GRUB2
tools/mkimage -T sunxi_toc1 -d toc1.cfg u-boot.toc1

Hello RISC-V!



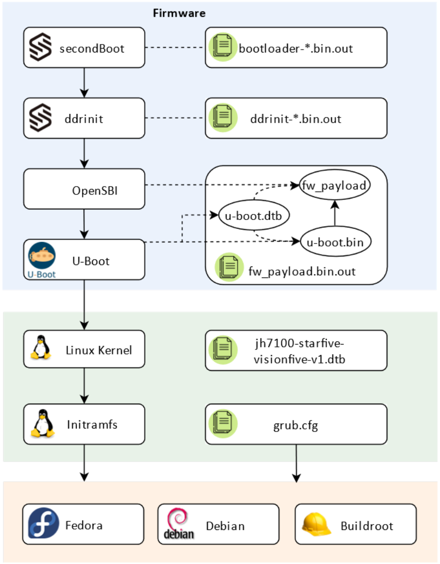
Operating System

Hello, RISC-V?



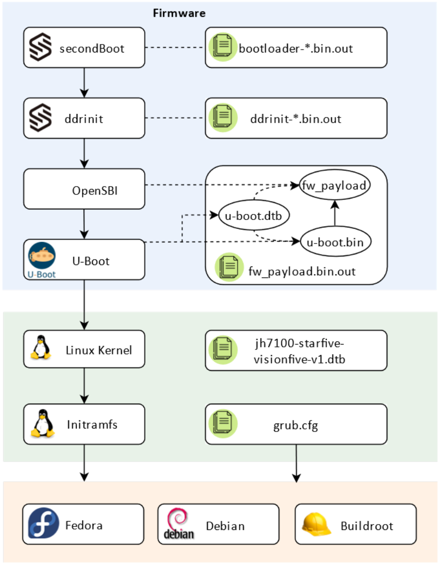
https://fedoraproject.org/wiki/Architectures/RISC-V/Allwinner

Why, RISC-V?



https://rvspace.org/en/Product/VisionFive/Technical_Documents/VisionFive_Single_Board_Computer_Quick_Start_Guide

Why, RISC-V?



https://rvspace.org/en/Product/VisionFive/Technical_Documents/VisionFive_Single_Board_Computer_Quick_Start_Guide

OREBOOT

We are working on a simpler oreboot port for the JH7100 / VisionFive. :)

Speeding Things up

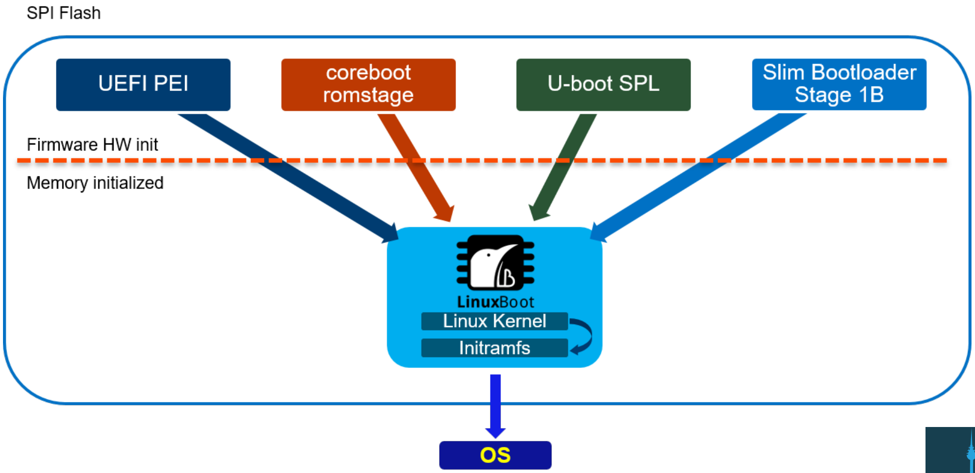
Offering LinuxBoot

Offering LinuxBoot

Take firmare for granted. Focus on the OS itself!

Offering LinuxBoot

Take firmare for granted. Focus on the OS itself!



https://www.linuxboot.org/

oreboot and LinuxBoot

oreboot and LinuxBoot

oreboot

oreboot initializes your hardware and executes a payload.

oreboot and LinuxBoot

oreboot

oreboot initializes your hardware and executes a payload.

LinuxBoot

LinuxBoot provides you with a Linux environment, including boot loaders.

LinuxBoot and `cpu`

LinuxBoot and `cpu`

LinuxBoot

Include `cpud`, the `cpu` daemon, in your LinuxBoot environment

LinuxBoot and `cpu`

Include `cpud`, the `cpu` daemon, in your LinuxBoot environment

Now do what you want and have a lot of fun!

Okay, `cpu` - but what is it?

Okay, `cpu` - but what is it?

Think of a little stub which lets you access and run anything on a remote that you bring from your host system.

Okay, `cpu` - but what is it?

Think of a little stub which lets you access and run anything on a remote that you bring from your host system.

Simple command

Use `cpu` instead of moving an SD card between host and test device!

Okay, `cpu` - but what is it?

Think of a little stub which lets you access and run anything on a remote that you bring from your host system.

Simple command

Use `cpu` instead of moving an SD card between host and test device!

```
cpu target-device ./program-to-test --with-some-args
```

Okay, `cpu` - but what is it?

Think of a little stub which lets you access and run anything on a remote that you bring from your host system.

Simple command

Use `cpu` instead of moving an SD card between host and test device!

```
cpu target-device ./program-to-test --with-some-args
```

Advanced (just as simple)

You can even `kexec` over `cpu`. :-)

Okay, `cpu` - but what is it?

Think of a little stub which lets you access and run anything on a remote that you bring from your host system.

Simple command

Use `cpu` instead of moving an SD card between host and test device!

```
cpu target-device ./program-to-test --with-some-args
```

Advanced (just as simple)

You can even `kexec` over `cpu`. :-)

```
cpu target-device ./kexec ./your-next-kernel
```

cpu DEMO

Leveraging `cpu` for Distro Testing

Leveraging `cpu` for Distro Testing

Use LinuxBoot to define a well-known system state. Attach to network.

Leveraging `cpu` for Distro Testing

Use LinuxBoot to define a well-known system state. Attach to network.

Power on the board, and check for readiness on the console.

Leveraging `cpu` for Distro Testing

Use LinuxBoot to define a well-known system state. Attach to network.

Power on the board, and check for readiness on the console.

Deploy Linux with `cpud` as the `init` (*cpukernel*).

Leveraging `cpu` for Distro Testing

Use LinuxBoot to define a well-known system state. Attach to network.

Power on the board, and check for readiness on the console.

Deploy Linux with `cpud` as the `init` (*cpukernel*).

Now, build your distro. Use NFS root and `kexec` into your new kernel.

Leveraging `cpu` for Distro Testing

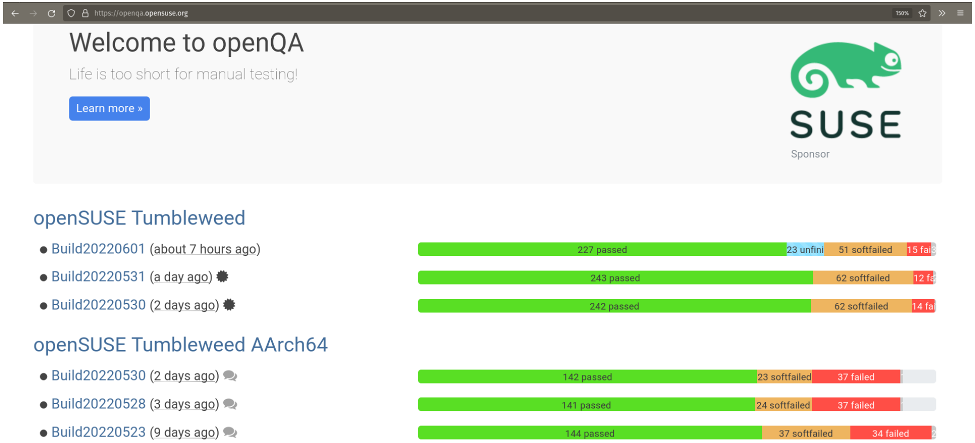
Use LinuxBoot to define a well-known system state. Attach to network.

Power on the board, and check for readiness on the console.

Deploy Linux with `cpud` as the `init` (*cpukernel*).

Now, build your distro. Use NFS root and `kexec` into your new kernel.
Does it boot? Yes -> yay! Nope -> you found a bug!

Hey openQA!



https://openqa.opensuse.org

Welcome to openQA

Life is too short for manual testing!

Learn more »

Sponsor

openSUSE Tumbleweed

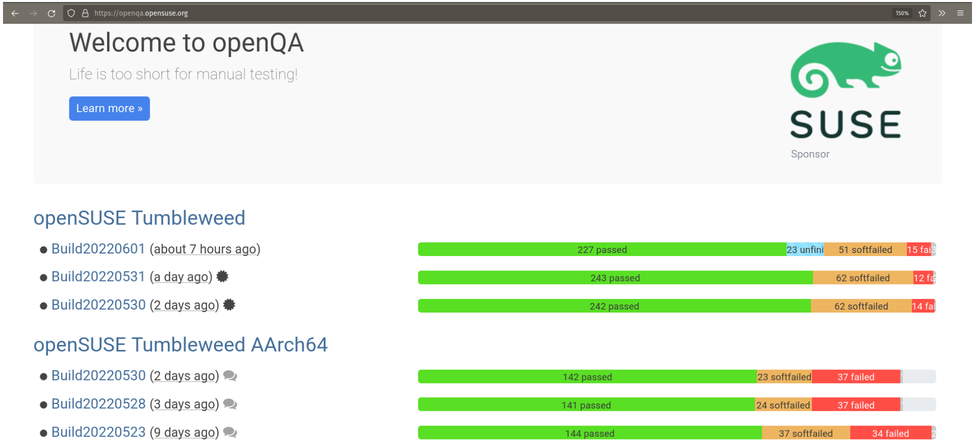
- Build20220601 (about 7 hours ago) — 227 passed | 23 unfini | 51 softfailed | 15 fai
- Build20220531 (a day ago) — 243 passed | 62 softfailed | 12 fa
- Build20220530 (2 days ago) — 242 passed | 62 softfailed | 14 fa

openSUSE Tumbleweed AArch64

- Build20220530 (2 days ago) — 142 passed | 23 softfailed | 37 failed
- Build20220528 (3 days ago) — 141 passed | 24 softfailed | 37 failed
- Build20220523 (9 days ago) — 144 passed | 37 softfailed | 34 failed

Life is too short for manual testing!

Hey openQA!



https://openqa.opensuse.org

Welcome to openQA

Life is too short for manual testing!

Learn more »

SUSE

Sponsor

openSUSE Tumbleweed

- Build20220601 (about 7 hours ago) — 227 passed, 23 unfini, 51 softfailed, 15 fai
- Build20220531 (a day ago) — 243 passed, 62 softfailed, 12 fa
- Build20220530 (2 days ago) — 242 passed, 62 softfailed, 14 fa

openSUSE Tumbleweed AArch64

- Build20220530 (2 days ago) — 142 passed, 23 softfailed, 37 failed
- Build20220528 (3 days ago) — 141 passed, 24 softfailed, 37 failed
- Build20220523 (9 days ago) — 144 passed, 37 softfailed, 34 failed

Life is too short for manual testing!

Where is RISC-V? Let’s make it happen! :-)

Testing Strategies and Setup

Testing Strategies and Setup

- test different build setup variations (e.g., cmdline args)
- assert on serial console and video output
- reset when done with each case (hard reset)

Testing Strategies and Setup

- test different build setup variations (e.g., cmdline args)
- assert on serial console and video output
- reset when done with each case (hard reset)

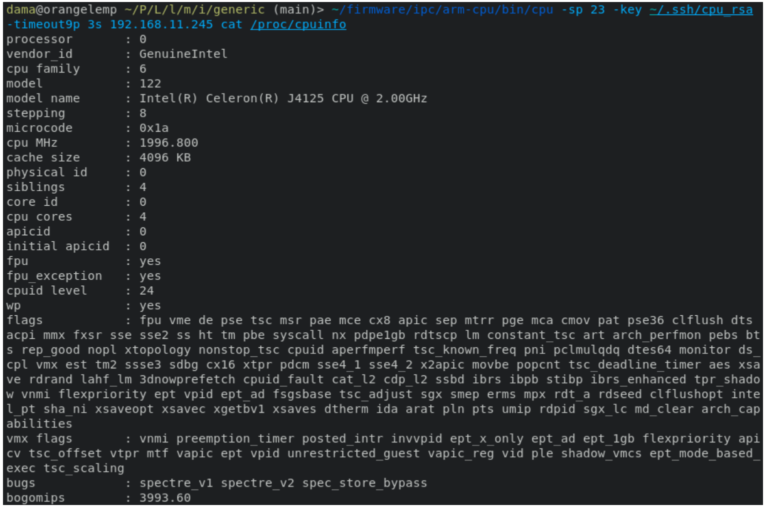
Requirements

- second piece of hardware for monitoring, reset, instrumentation
  - HDMI capture, USB-HDMI, VNC etc
  - connect to UART
  - hook up GPIO to reset
  - e.g., Raspberry Pi, 3mdeb RTE, DIY…
- some glue logic in build service (CI)

Try it out!

Join our workshop at 17:00 in Seminarraum 2

Preparation: Install either Docker or QEMU plus (optionally) Go 1.18.



```
dama@orangelemp ~/P/L/l/m/i/generic (main)> ~/firmware/ipc/arm-cpu/bin/cpu -sp 23 -key ~/.ssh/cpu_rsa
-timeout9p 3s 192.168.11.245 cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 122
model name      : Intel(R) Celeron(R) J4125 CPU @ 2.00GHz
stepping        : 8
microcode       : 0x1a
cpu MHz         : 1996.800
cache size      : 4096 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 4
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 24
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bt
s rep_good nopl xtopology nonstop_tsc cpuid aperfmperf tsc_known_freq pni pclmulqdq dtes64 monitor ds_
cpl vmx est tm2 ssse3 sdbg cx16 xtpr pdcm sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsa
ve rdrand lahf_lm 3dnowprefetch cpuid_fault cat_l2 cdp_l2 ssbd ibrs ibpb stibp ibrs_enhanced tpr_shado
w vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust sgx smep erms mpx rdt_a rdseed clflushopt inte
l_pt sha_ni xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts umip rdpid sgx_lc md_clear arch_cap
abilities
vmx flags       : vnmi preemption_timer posted_intr invvpid ept_x_only ept_ad ept_1gb flexpriority api
cv tsc_offset vtpr mtf vapic ept vpid unrestricted_guest vapic_reg vid ple shadow_vmcs ept_mode_based_
exec tsc_scaling
bugs            : spectre_v1 spectre_v2 spec_store_bypass
bogomips        : 3993.60
```