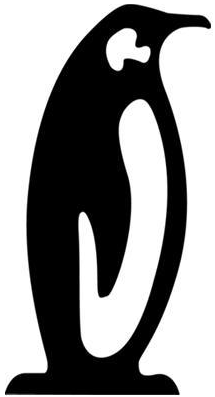


東京 2025 LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025

Preparing RISC-V Linux for RVA23

東京 2025
LINUX
PLUMBERS CONFERENCE
TOKYO, JAPAN / DEC. 11-13, 2025

Why should the kernel add support for RVA23?

- Standardization around a set of configs for performant hardware
- Enable broad-sweeping optimizations for RVA23 hardware
- Improve developer experience around extension support and discovery

RFC available here: https://lore.kernel.org/linux-riscv/20251210-profiles-v1-0-315a6ff2ca5a@gmail.com/T/#t

Standardization around a set of configs for performant hardware

- Introduce a new defconfig “rva23_defconfig” that is focused on server platforms
- This will be based on a new rva23 config, and be portable to all rva23 compatible platforms
- “defconfig” will continue to be used to build a kernel that can run on any platform

RVA23

Profiles: An attempt to wrangle the inherent fragmentation of the riscv ISA

- RVA23 targets server-class and application processors
- Notably requires:
  - Vector
  - Hypervisor
  - May-be-ops
  - Performance counter overflows

Example isa string for rva23 compatible hardware

rv64mafdcbvh_zicsr_zicntr_zihpm_ziccif_ziccrse_ziccrse_ziccamoa_zicclsm_za64rs_zihintpause_zic64b_zicbom_zicbop_zicboz_zfhmin_zkt_zihintntl_zicond_zimop_zcmop_zcb_zfa_zawrs_supm_svade_ssccptr_sstvecd_sstvala_sscounterenw_svpbmt_svinval_svnapot_sstc_sscofpmf_ssnpm_ssu64xl_sstateen_shcounterenw_shvstvala_shtvala_shvstvecd_shvsatpa_shgatpa

Enable broad-sweeping optimizations for RVA23 hardware

- Split extensions into “enabled”, “supported”, and “disabled”
- RVA23 config will select “enabled” for all mandatory extensions
- These extensions can then be added to the compiler’s march
- Focus performance optimizations on a kernel where extensions are compiled into it rather than discovered at runtime

riscv_has_extension_* optimization

```
if (riscv_has_extension_likely(EXT)) {
   alternative
} else {
   fallback
}
```

	Enabled	Supported	Disabled
Discovered in isa string	alternative	nop alternative j end no fallback end:	fallback
Not discovered in isa string	alternative	j no alternative j end no: fallback end:	fallback

Eliminate hand-coded assembly routines

Example: do_csum_no_alignment()

Standard kernel optimizations

```
srl     a4,a5,0x20
sll     a3,a5,0x20
add     a4,a4,a3
add     a5,a5,a4
srl     a4,a5,0x20
ld      ra,8(sp)
ld      s0,0(sp)
srl     a5,a5,0x30
sllw    a0,a4,0x10
addw    a0,a0,a5
addw    a0,a0,a4
srlw    a0,a0,0x10
add     sp,sp,16
```

Current hand optimizations

```
ld      ra,8(sp)
ld      s0,0(sp)
ror     a5,a0,0x20
add     a0,a0,a5
srl     a0,a0,0x20
rorw    a5,a0,0x10
addw    a0,a0,a5
srl     a0,a0,0x10
sext.w  a0,a0
```

Manual sign extension

Set zbb as "enabled"

```
ror     a4,a5,0x20
add     a5,a5,a4
ld      ra,8(sp)
ld      s0,0(sp)
sra     a5,a5,0x20
rorw    a0,a5,0x10
addw    a0,a0,a5
srlw    a0,a0,0x10
```

Compiler is now able to keep track of when to sign extend at caller

(Also: The riscv compiler has gotten a lot better over the years and can properly optimize more sequences now)

Improve developer experience around extension support and discovery

Globally enabled extensions simplify writing optimized code

- When code is guarded by CONFIG_RISCV_ISA_EXT and CONFIG_TOOLCHAIN_HAS_EXT, the extension is available to the compiler so ".option arch" can be dropped

What happens when trying to boot the rva23 kernel on non-compliant hardware?

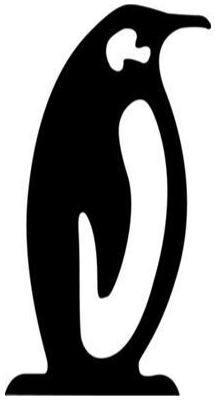
It will crash.

What happens when trying to boot the rva23 kernel on non-compliant hardware and it doesn't crash?

- The hardware supports enough instructions that any that are not supported are not being emitted by the kernel or the compiler
- The isa string could be checked and a warning thrown with which extensions are expected but not found

Should we add support for “rva23” in the isa string?

- The isa string is *really* long
- The developer only needs to add on additional extensions their platform supports
- Requires changes everywhere (QEMU, OpenSBI, applications reading /proc/cpu, etc...)



東京 2025
LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025