



Contribution ID: 169

Type: **not specified**

RISC-V Vector: Current Status and Next?

Monday, 13 November 2023 12:10 (15 minutes)

In this talk we are going to briefly share the status of Vector extension support and focus our discussion on the use of Vector in the kernel-mode. We will do it by reviewing others arch approaches and seeking if there is anything we may carry or improve as risc-v.

Most architectures provide SIMD instruction set to improve throughput of some operations. However, the use of SIMD instructions in their kernel mode is often restricted due to latency considerations for extra state-keeping. For example, it is not uncommon to see an arch that disables preemption when using kernel-mode SIMD. Also, msot ban the use of SIMD in interrupt context.

Among these architectures, seldom provide SIMD-optimized common sub-routines (mem/str ops). PowerPC provides vsx/vmx-optimized common routines with a precondition and a side-effect. First, it cannot leverage those routines in interrupt context. And, it must disable kernel preemption while using these subroutines. Meanwhile, though the same side effect applies to x86, it provides `irq_fpu_usable()` to allow some level of SIMD uses in interrupt context.

On risc-v, should we follow the path of powerPC? If yes, how do we decide when to use it? Vendors may have varies performance characteristic of V and using SIMD for small inputs may not gain. Should we do runtime detection for this or just enable V whenever the hardware supports?

Further, should risc-v take a step forward, by enabling preemption for its kernel-mode SIMD [1]? Supporting kernel preemption during Vector execution may enable us to widely use Vector in kernel thread or syscalls while remain same level of responsiveness. Historically, the reason for disabling kernel preemption while using kernel-mode SIMD is the per-cpu variable consideration [2]. However, the per-cpu FPU cache is phasing out on x86 and is not present on risc-v's approach. So, it might be a good timing for discussing if such support is a good idea now.

The talk will cover the following topics:

- The current status of Vector extension support
- Basic idea of supporting kernel-mode Vector
- The use of Vector optimized sub-routines for us and other arch
- Should we support running kernel-mode Vector with preemption?

1 <https://lore.kernel.org/all/20230721112855.1006-1-andy.chiu@sifive.com/>

2 https://yarchive.net/comp/linux/kernel_fp.html

Primary author: CHIU, Tao

Presenter: CHIU, Tao

Session Classification: RISC-V MC

Track Classification: LPC Microconference: RISC-V MC