

# One Kernel Per Cluster: Multikernel for Heterogeneous RISC-V

**Cong Wang**, Founder and CEO, Multikernel Technologies, Inc.

**Yuning Liang**, Founder and CEO, Deep Computing

Linux Plumbers Conference 2026

# Sixteen Cores, Eight Dark

SpacemiT K3: 8 cores at VLEN 256, 8 AI cores at VLEN 1024. On mainline, the AI cores never come up.

```
if (riscv_v_vsize != this_vsize) {  
    WARN(1, "RISCV_ISA_V only supports one vlenb on SMP systems");  
    return -EOPNOTSUPP;  
}
```

`riscv_v_vsize` sizes every vector state buffer. Linux has one width for the whole machine.

# The Widths Will Not Converge

- **Different jobs:** latency vs. throughput. Widening the scalar cores would quadruple their vector state
- **No one can force agreement:** RVA23 requires at least 128 bits, nothing more
- **Physics:**  $32 \times 1024$  bits = 4 KB of live state; a 256-bit hart has 1 KB to receive it
- **Versions differ too:** SG2042 ships draft RVV 0.7.1, incompatible with 1.0

Migration of a vector task across clusters is dead for every design.

# Two Ways Out

**One kernel, patched**, as SpacemiT's downstream 6.18 does:

- Per-task `vlenb`, widest buffers, pin at first vector use
- Pays arm64's price: hotplug, `SCHED_DEADLINE`, affinity, cpusets
- hwcap is the intersection. Extensions on one cluster only disappear

**One kernel per cluster**, with multikernel:

- Upstream kernels spawned with `kexec`. Each owns its harts and devices
- Every kernel sees matching harts. Full width, honest hwcap, plain Linux

The hardware is already partitioned. So should be the software.

# Multikernel in One Slide

Many Linux kernels on one machine. No hypervisor. Each kernel owns its CPUs, memory, and devices.

- **Spawn:** `kexec` boots a new kernel on spare CPUs. The first keeps running
- **Describe:** a device tree assigns resources. Overlays change them at runtime
- **Talk:** vsock over IPIs and shared memory
- **Share:** software devices via virtio. Files via DAXFS

On K3: one kernel on the X100 harts, one on the A100 harts.

# Q1: Is K3 an Exception?

Mixed vector widths on one die. A one-off, or where RISC-V is heading?

- **The case for:** wide vectors only where needed save area and power
- **The case for:** AI cores share the ISA, so no separate NPU toolchain
- **The case against:** x86 and arm64 chose matching state. Intel fused off AVX-512 rather than mix
- **RISC-V is different:** core IP comes from many vendors. Nothing forces one width

**Does the economics favor heterogeneous widths, or will they fade?**

# Q2: Where Does the Argument Break?

Our position: when the hardware is already partitioned, so should be the software.

- **A patched kernel still partitions:** it only hides it, and the scalar cores pay
- **A JIT does not change the physics:** it changes when code is generated. Not how many bits fit
- **Most server vector code has no JIT:** glibc, OpenSSL, OpenBLAS
- **Multikernel needs no migration:** each workload starts on the cluster built for it

**Is there a workload where a single kernel is worth its price?**

# Q3: Spawning a Kernel on RISC-V

We `kexec` a second kernel onto some harts via SBI HSM. The first kernel keeps running.

- OpenSBI already has domains. Are they the right ownership model?
- Should the SBI spec define it, or stay out of OS partitioning?
- What happens to timers, IPIs, and SBI calls from harts a kernel does not own?

**Who on the firmware side wants to own this?**

# Q4: Interrupts Across Domains

With AIA, each hart has an IMSIC; wired interrupts go through an APLIC.

- Can one APLIC be partitioned between kernels, or does each domain need its own?
- We put devices on scalar harts. Does that work on every SoC?
- How does the vector domain receive IPC doorbells from the scalar domain?

**Is per-kernel IMSIC and APLIC ownership something the irqchip drivers should model?**

# Q5: Describing Heterogeneous Topology

Each kernel must see only its own harts, with their real ISA, VLEN, and errata.

- DT already has per-hart `riscv,isa-extensions` and `riscv,vlenb`. Enough for two clusters?
- Errata are patched for whichever hart booted first; per-cluster errata need a model
- ACPI RHCT: does it express any of this?

**Where should the cluster boundary be described: firmware, DT, or the spawning kernel?**

# Q6: hwprobe Per Domain

With one kernel per cluster, each `hwprobe` answers only for its own harts.

- Should a kernel report harts it does not own, so a scheduler above can place work?
- Is a cross-domain placement API userspace, or kernel?

**What should userspace see, and who decides where a workload runs?**

# Q7: Park, or Hand Off?

Today a hart with a mismatched VLEN is parked at boot. It never comes online.

- **Park**: safe and simple. Half of K3 becomes dead silicon
- **Online it in one kernel**: pays the price list from "Two Ways Out"
- **Hand off**: keep it offline, but leave it described for another kernel to claim
- The same choice applies to any hart Linux cannot use, not just VLEN mismatches

**Should mainline keep parking it, or hand it off?**

# Further Discussion

Contact: [cwang@multikernel.io](mailto:cwang@multikernel.io), [yuning.liang@deepcomputing.io](mailto:yuning.liang@deepcomputing.io)

Code: [github.com/multikernel/linux](https://github.com/multikernel/linux)

Mailing list: [multikernel@lists.linux.dev](mailto:multikernel@lists.linux.dev)