We had 64-bit, yes. What about second 64-bit?

Mathieu Bacou[†], Adam Chader[†], Chandana Deshpande[‡], Christian Fabre[⋆], César Fuguet[⋆], Pierre Michaud[⋆⋆], Arthur Perais[‡], Frédéric Pétrot[‡], Gaël Thomas[†], Eduardo Tomasi[‡⋆]

[†]Télécom Sud Paris, IP Paris [‡]Univ. Grenoble Alpes, CNRS, Grenoble INP[1], TIMA [⋆]Univ. Grenoble Alpes, CEA, List [⋆⋆]Inria, Univ. Rennes, CNRS, IRISA

[1]Institute of Engineering Univ. Grenoble Alpes

Overview of ANR Project Maplurinum (ANR-21-CE25-0016)

The Big Picture

- Current OSes struggle to provide efficient abstractions for increasingly heterogeneous hardware (accelerators)
- Scalability issues of hardware and software to a rack-scale computing model where multiple blades share main memory
- Advent of load/store accessible backup store (e.g., NVM)

The Goal

- Rethink the operating system as intercommunicating satellites managing active hardware accelerators
- Future-proof it by implementing 128-bit flat addressing
  → Rely on an open-source RISC-V extension, RV128
  → Impacts the whole stack from OS to micro-architecture

Operating System & Software

Multi-kernel: constellation of active satellites

- Scalability and uniformity over hardware heterogeneity
  → Make the kernel a distributed system: satellite kernels
  → Satellites run on RV128 monitor cores appended to accelerators: hardware devices become active

Unified address space

- Allow direct access of applications to the data plane: loads and stores to unified memory space
  → User processes get memory grants from remote satellites

Architecture & Microarchitecture

128-bit Architecture

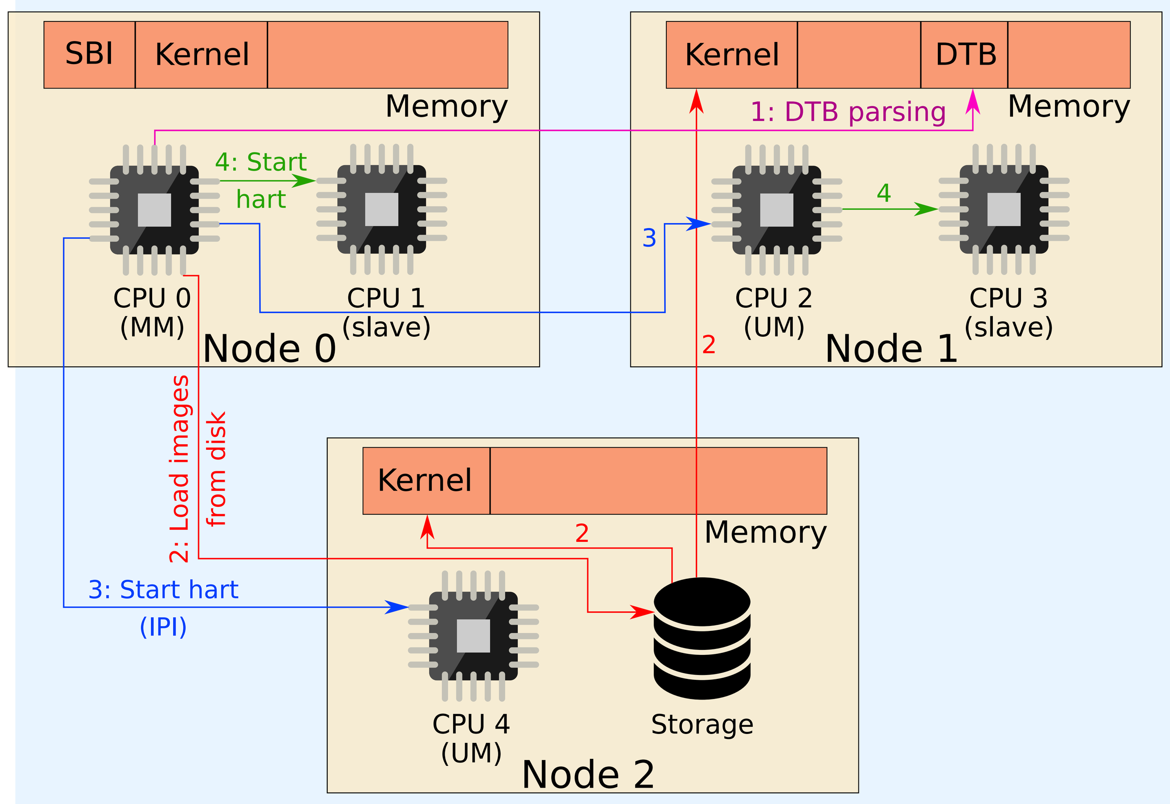
- RV128 extension as a common denominator: All agents (CPUs, GPUs, TPUs, FPGAs) are RV128-capable
  → Satellite kernels can run anywhere

128-bit General Purpose Microarchitecture

- Naively: Double datapath width (bypass, registers, functional units)
- Dennard scaling and Moore's Law not there to absorb the change anymore: Need to limit hardware cost of RV128

Machinæ pluribus unum – One Machine out of Many

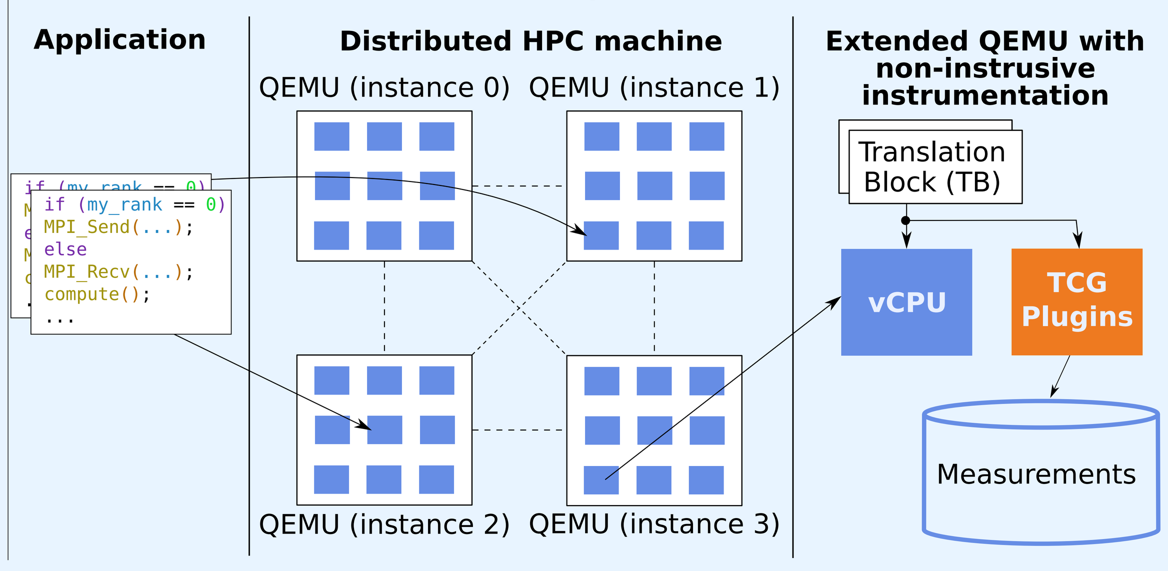
Operating System



- Simulation of a rack-scale machine: boot satellite kernels on NUMA nodes
- Experiments on porting userspace applications to a unified address space
  - Java VM with a remote garbage collector
  - Hardware virtualization for efficient userspace control of unified address space

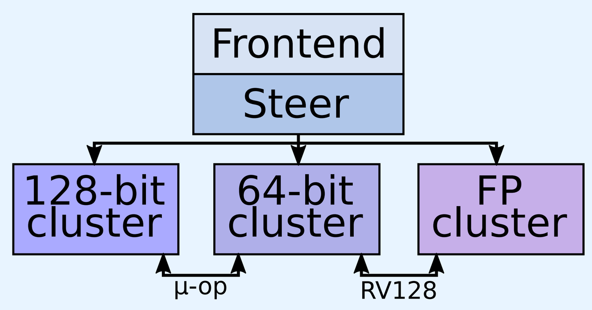
Memory Hierarchy

- NUMA effects are exacerbated in heterogeneous multi-socket, multi-board HPC computing systems.
- 128-bits shall ease the programming of large scale systems, but NUMA effects must be considered.
- Software and hardware mechanisms are being analyzed to hide this latency.
- We developed a QEMU-based simulator for distributed large scale machines.



Microarchitecture

- Compile an existing C program to RV128: About 40% of the instructions still operate on 32/64-bit
  → 128-bit operations will mostly be address generation slices



- Divide & Conquer: 128-bit cluster for addresses, 64-bit cluster for arithmetic
  → Push complex 128-bit operators (e.g., mul, div) to SW
  → Compress addresses (PRF, TLBs tags/data, cache tags), reduces area

Perspectives

- Discover the minimal ISA and adequate software interfaces for the satellite kernels
- Establish the user-kernel interfaces for efficient distributed computing through the unified address space
- Revisit basic operating system concepts for machine-wide, unified 128 bit address space: process, memory mapping policies, etc.
- Identify hardware requirements for adequate support of a machine-wide, distributed 128 bit address space
- Propose OS-driven hardware mechanisms that replicate data between nodes and manage coherency to reduce NUMA latency