On a RISC-V Lightweight Manycore for Operating Systems Research

Pedro Henrique Penna[1,2]

Advisors: Jean-François Méhaut[1] and Henrique Freitas[2]

Co-Advisors: Márcio Castro[3] and François Broquedis[4]

[1]Université Grenoble Alpes (UGA)

[2]Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

[3]Universidade Federal de Santa Catarina (UFSC)

[4]Institut National Polytechnique de Grenoble (Grenoble INP)

- Thousands of Lightweight Cores
  - MIMD workloads
  - Massive thread-level parallelism
  - Low power consumption
- Distributed Memory Architecture
  - Performance scalability
  - Communication predictability
- On-Chip Heterogeneity
  - Adaptability to computing demands
  - High energy efficiency
- Rich On-Chip Interconnects
  - Quality of Service (QoS)
  - Asynchronous communications

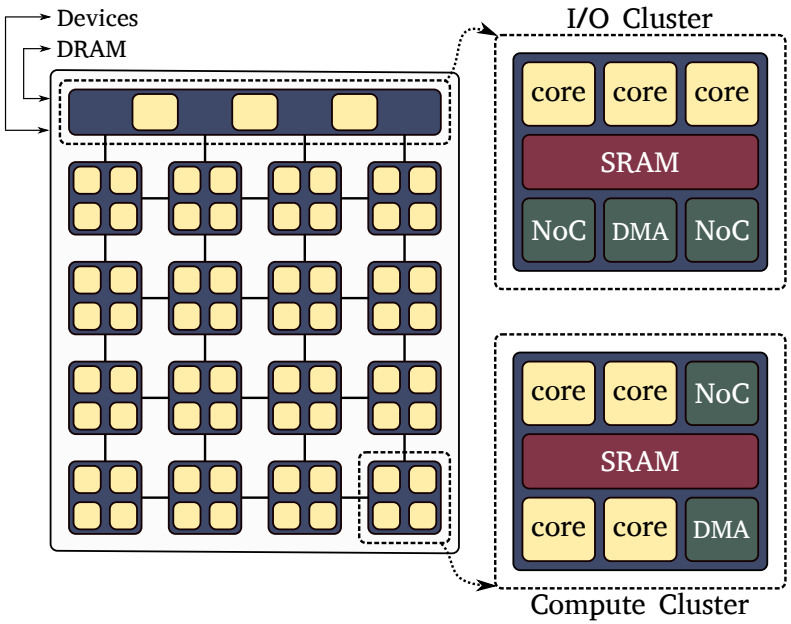

Figure: LW manycore with 67 cores.

- Currently used in embedded computing, critical systems and networking
  - What about domains with **multi-application** requirements?

- High Density Circuit Integration
  - Heat dissipation
  - Dark silicon
- Distributed Memory Architecture
  - Small local memories
  - Challenging software design
- On-Chip Heterogeneity
  - Thread scheduling
  - Data placement
- Rich On-Chip Interconnects
  - Network congestion
  - Security checking

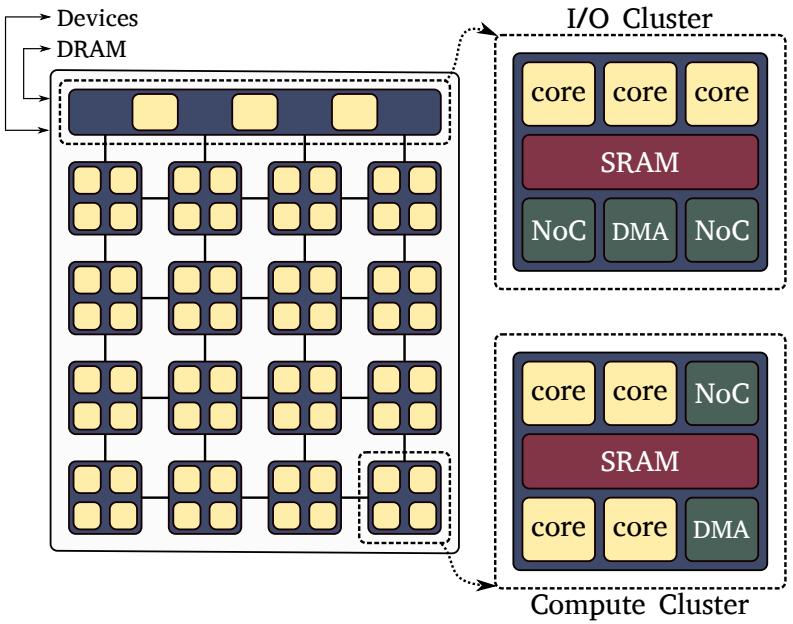

Figure: LW manycore with 67 cores.

Performance vs Programmability vs Portability

- **An Operating System (OS) Bridges Software Challenges**
  - Expose rich abstractions and APIs
  - Multiplex access to resources
  - Provide and ensure security
- **How about Using Commodity Kernels?**
  - Ex: Linux, FreeBSD, Windows...
  - Pros: automatically support tons of software
  - Cons: memory footprint is too large to fit in lightweight manycores
- **Is It Possible to Design OSes for LW Manycores Like We Do for Multicores?**
  - Symmetric kernel design leads to cache interference (Wentzlaff and Agarwal 2009)
  - Poor fine-grain lock scalability (Amdahl's Law)
  - Increasingly diverse hardware (Baumann et al. 2009)
  - Multiple non-coherent physical address spaces (Dinechin et al. 2013)

No, we need another approach!

- Kernels
  - Run (self-consciously) on each cluster
  - Provide minimum abstractions
  - Ensure policies and security
- System Servers
  - Run on top of kernels at user-level
  - Provide traditional abstractions
  - Collaboratively implement subsystems
- Runtime Libraries
  - Run alongside with user-applications
  - Interface with system servers
  - Expose standard APIs (i.e., POSIX)

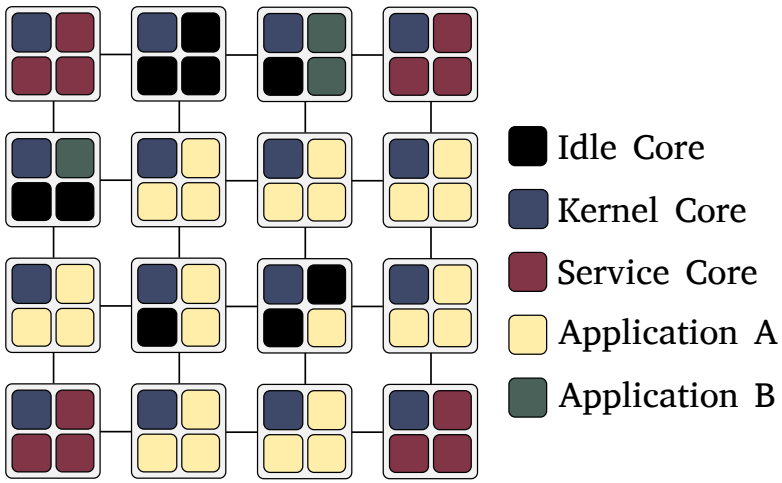

Figure: The multikernel OS structure.

- **The Nanvix Operating System**
  - Joint research between UGA, PUC Minas, UFSC and Grenoble INP
  - Multikernel designed from scratch to lightweight manycores
  - Supports multiple ISAs: RISC-V, OpenRISC, x86 and Bostan (Kalray MPPA-256)

https://github.com/nanvix

- Two-Level Privilege Mode
  - Resource protection
  - Bare-bones for security
- Virtual Memory Support
  - Physical memory multiplexing
  - Address space expansion and protection
  - Must have to provide process abstraction
- Atomic Instructions
  - Intra-cluster thread synchronization
  - Required in multicore clusters
- Fast Interrupt/Exception Forwarding
  - User-Level interrupt/exception handling
  - Essential for fast microkernel support
- Fine-Grain Interrupt Hooking Control
  - Interrupt priority scheme
  - Low-latency in inter-cluster communication

- Three-Level Privilege Mode (Machine, Supervisor and User)
- 48-bit Address Space Support
- Atomic Instructions
- Interrupt/Exception Delegation
- Flexible and Extensible ISA
- Rich Interrupt System
- Multiple open-source implementations
- Trending architecture and active community

Gem5

- Full architectural simulation
- Too slow for system design and development
- Partial support for RISC-V
- Time-consuming and hard to change

QEMU

- Fast processor emulation
- Rich system debugging support
- Adequate for medium-size configurations
- Support for RV32GC, RV64GC, Spike and SiFive
- Misses distributed configuration model

PULP Cluster

- 1+8 RI5CY cores
- 4 KB of L1 I-Cache
- 256 KB of L1 SPM
- 256 KB of L2 SPM
- DMA controller

RI5CY Core

- Low energy consumption
- 4-stage 32-bit pipeline
- I M F C extensions
- Partial M and U modes
- No virtual memory
- No atomic instructions

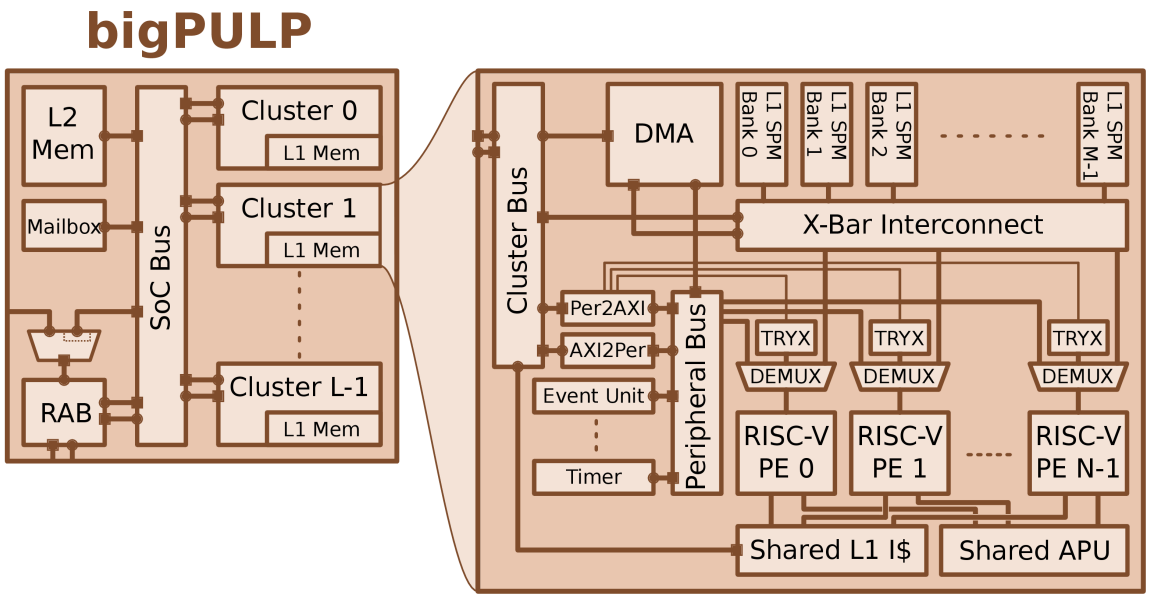

Figure: bigPULP architecture overview.

OpenPiton

- Tiled Configuration
- Mesh NoC topology
- Cache-coherence system
- Directory coherence

Ariane Core

- High performance
- 6-stage 64-bit pipeline
- I M A C extensions
- M, S and U modes
- Remote GDB support

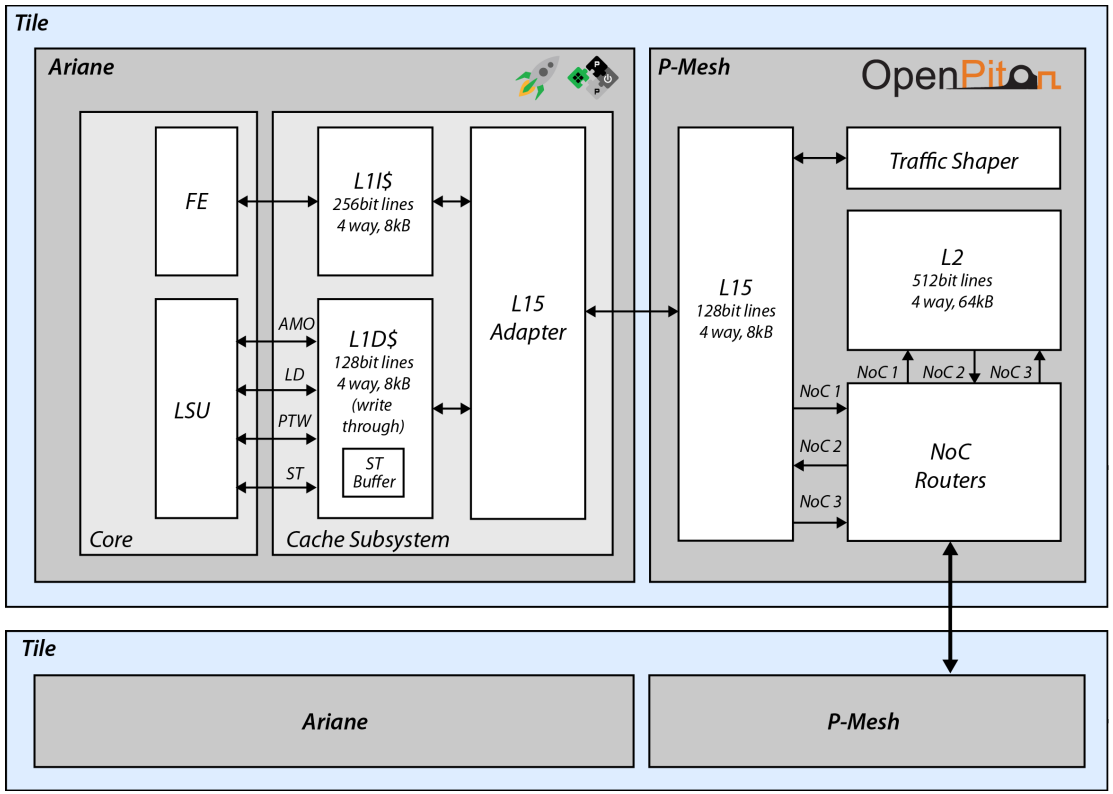

Figure: OpenPiton+Ariane architecture overview.

- Design is to large to emulate a manycore configuration
  - 2x2 single hart system in Xilinx VC707 ($ 3,495)

Conclusions

Recap

- Lightweight Manycores
  - Distributed memory configuration
  - Rich on-chip interconnect
- Operating System for Lightweight Manycores
  - Enable multi-applications to be deployed alongside
  - Expose rich abstractions and APIs
  - Multiplex hardware resources fairly and safely

What Is next?

- Virtual RISC-V Manycore Platform
  - QEMU-based platform emulation
  - Virtual interconnect with network interfaces
  - Patch RISC-V platforms with network devices
- RISC-V Manycore FPGA Emulation
  - Configuration on OpenPiton+Ariane
  - Lighter Ariane cores (remove hardware PTW, branch prediction, cache system)