MemPool Flavors: Between Versatility and Specialization in a RISC-V Manycore Cluster

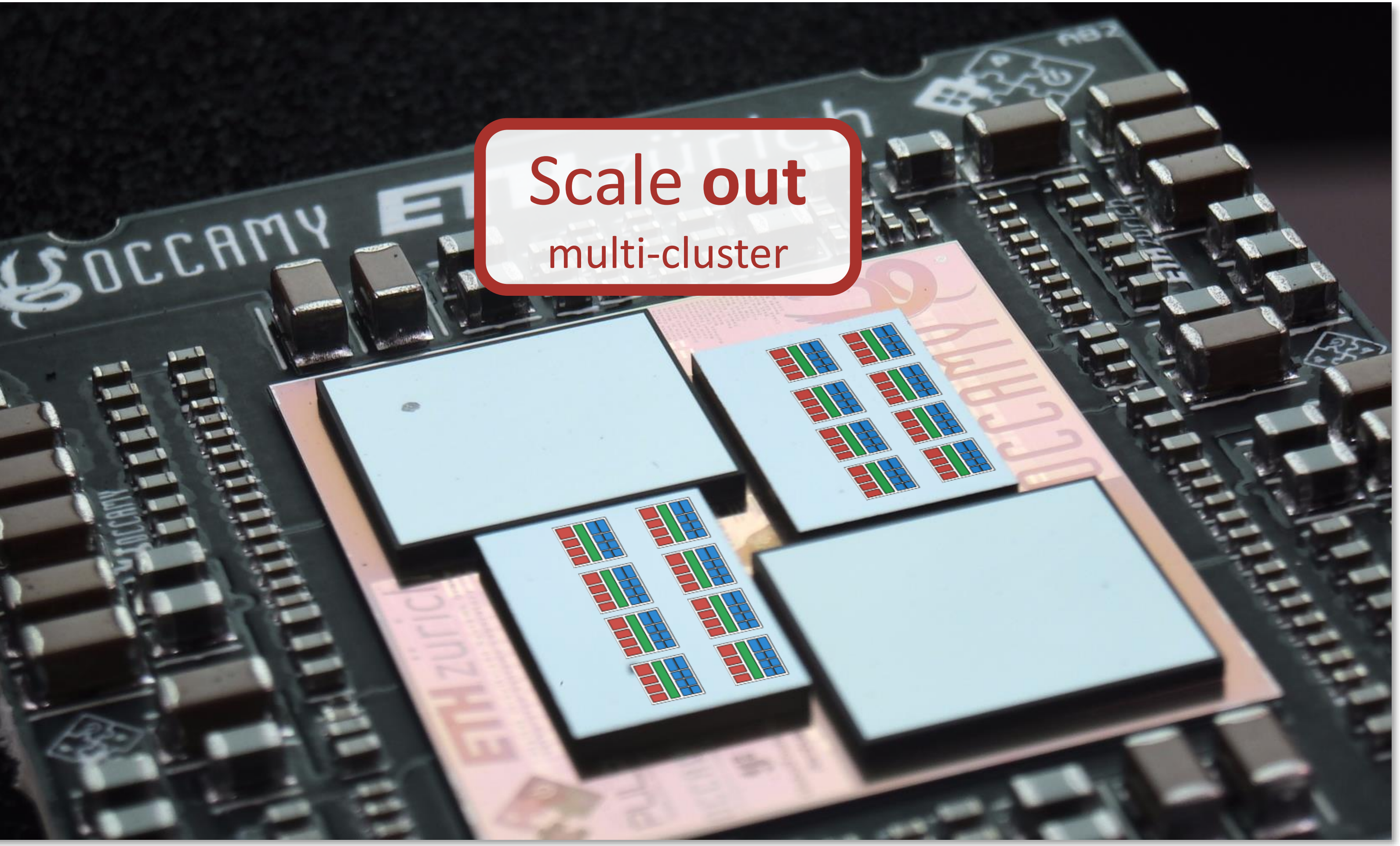
Integrated Systems Laboratory (ETH Zürich)

Sergio Mazzola smazzola@iis.ee.ethz.ch
Yichao Zhang
Marco Bertuletti
Diyou Shen
Luca Benini

PULP Platform
Open Source Hardware, the way it should be!

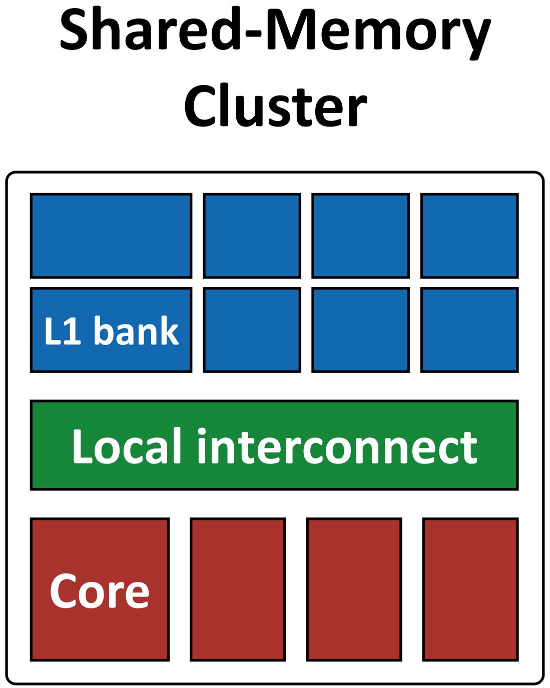
pulp-platform.org
@pulp_platform
company/pulp-platform
youtube.com/pulp_platform

Large workloads → hundreds of **cores** + big **memory**

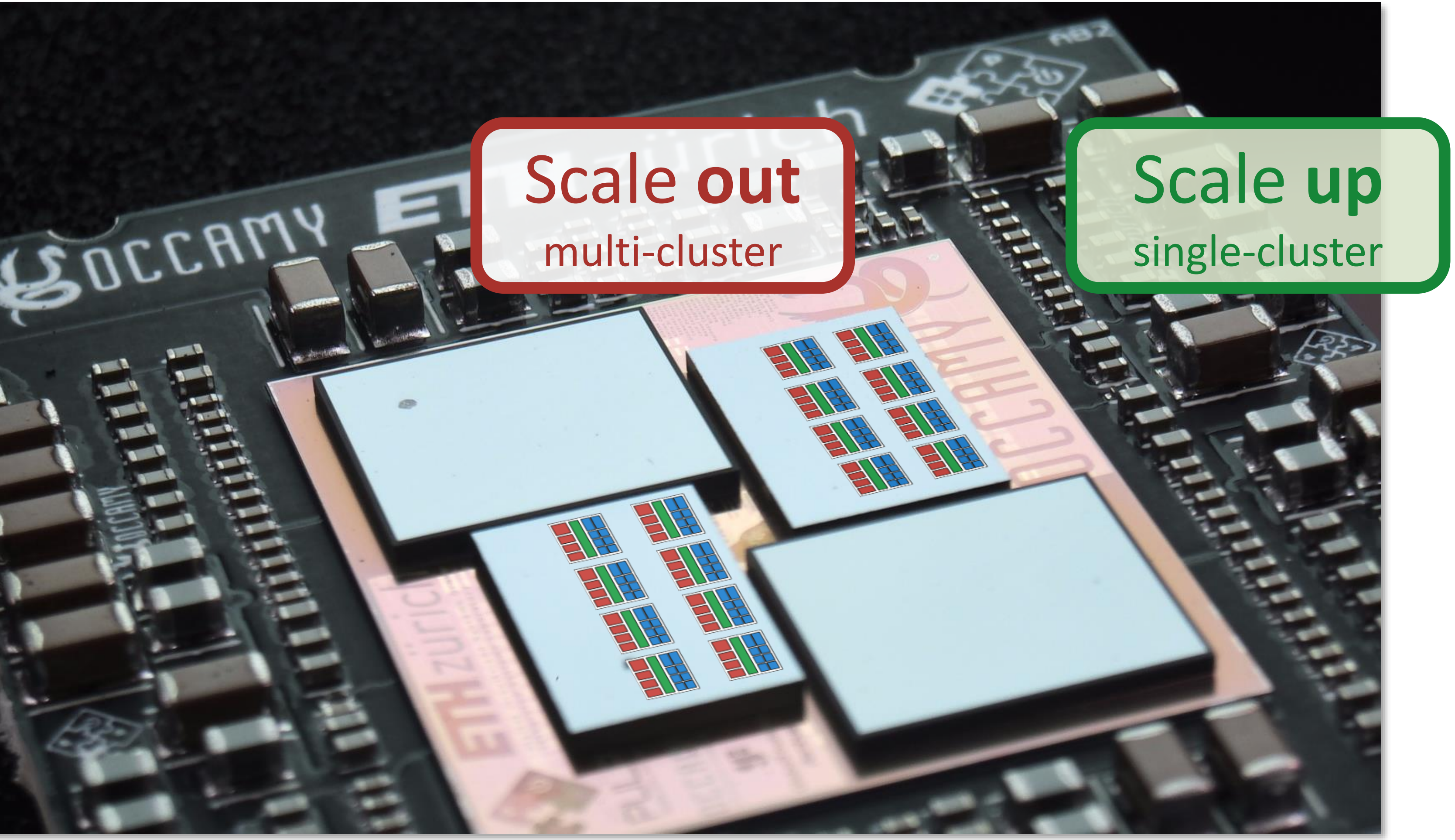


Scale **out**

multi-cluster



Large workloads → hundreds of **cores** + big **memory**

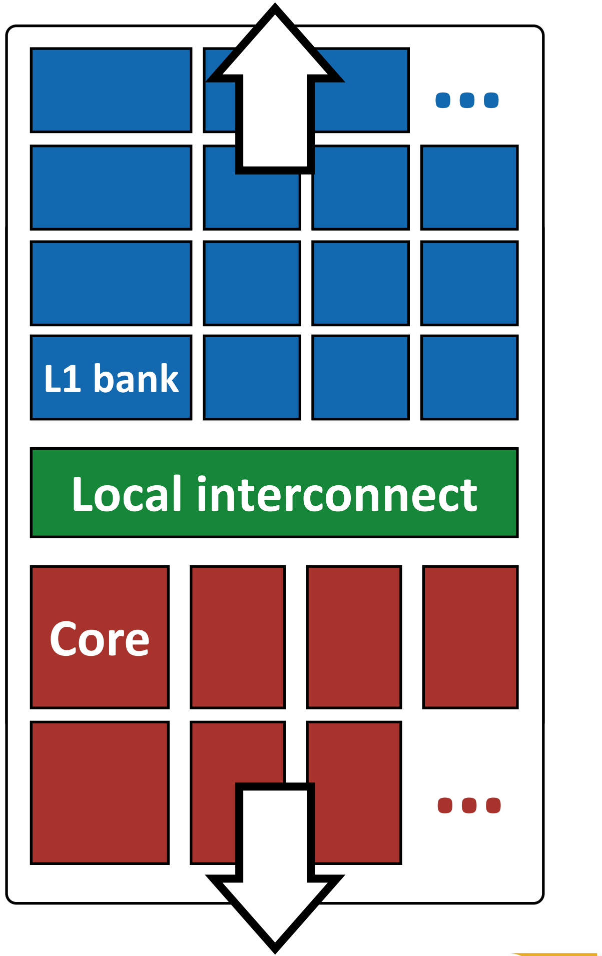


Scale **out**

multi-cluster

Scale **up**

single-cluster

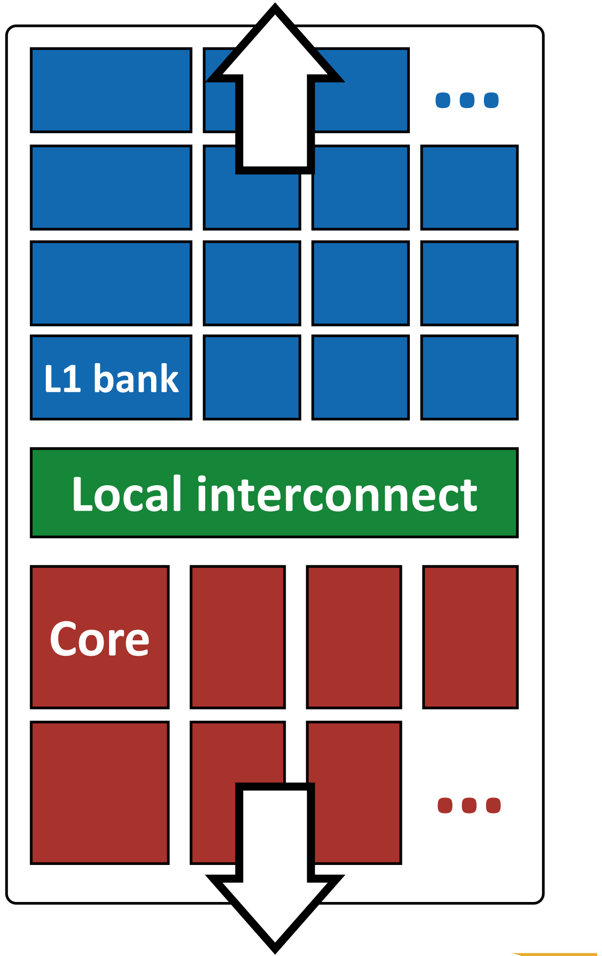


Large workloads → hundreds of **cores** + big **memory**

- **Reduce overhead of data chunks split/transfer/merge**
- **High compute utilization**
- **Easy to program**

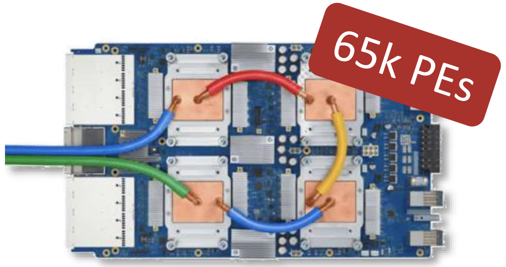
- **Low-latency memory access**
- **Physically-feasible interconnect**

Scale **up**
single-cluster



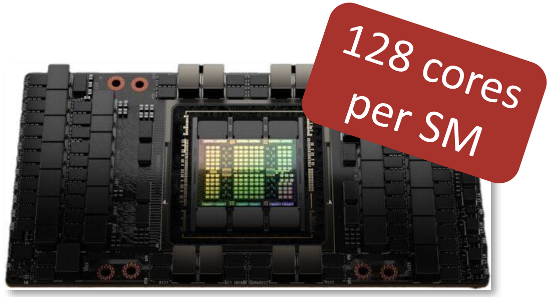
That’s difficult! How do we **scale up**?

scalable


65k PEs

https://cloud.google.com/tpu

Google TPU
- Custom accelerator
- Very specialized

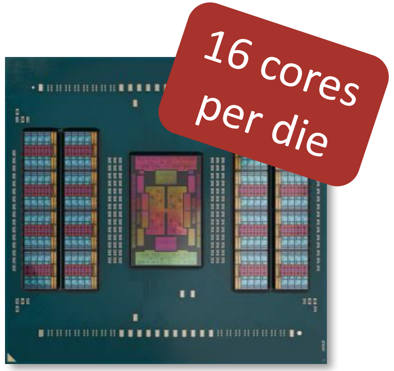

128 cores per SM

https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/

NVIDIA Hopper GPU
- SIMT
- Complex mem hierarchy
- Flexible

256+ cores

MemPool


16 cores per die

AMD EPYC CPU
- General-purpose
- Doesn’t scale

https://www.amd.com/en/products/processors/server/epyc/9005-series.html

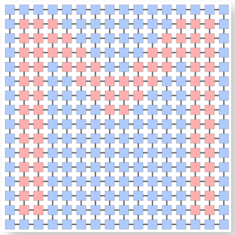
versatile

MemPool **Flavors**

MemPool

Vectorial MemPool

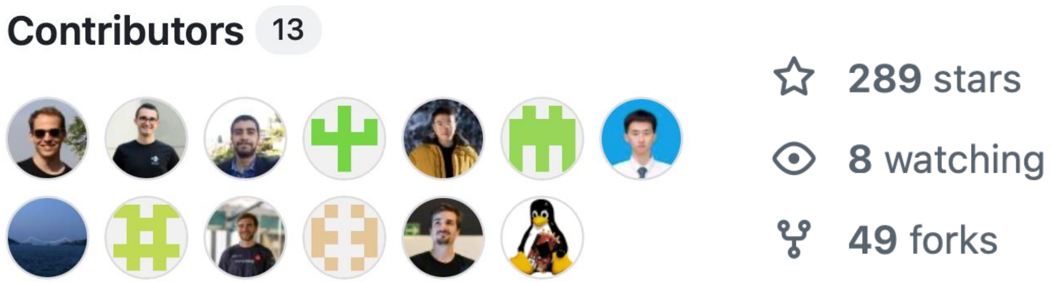
ITA MemPool


Systolic MemPool

TeraPool

NoC TeraPool

CachePool


Contributors 13

289 stars

8 watching

49 forks

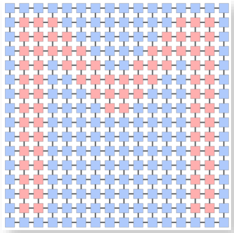
github.com/pulp-platform/mempool

MemPool **Flavors**

MemPool

Vectorial MemPool

ITA MemPool



Systolic MemPool

TeraPool

NoC TeraPool

CachePool

More than an architecture: **MemPool Ecosystem**

SOFTWARE

- Wide software **kernel library**
- Bare-metal **runtime**, OpenMP, Halide
- **GCC** and **LLVM** toolchain support
- Support for GVSOC and Banshee **platform emulators**

HARDWARE

BACKEND

- Mature backend flow in many **modern technologies**
- **2 tapeouts**

MINPOOL
by
Domenic Wüthrich, Manos Frouzakis,
Matheus Cavalcante and Samuel Riedel
PULP

MinPool (2021)
16 cores, 200 MHz
TSMC65, 2.4mm x 2.4mm

Heartstream (2024)
64 cores, 720 MHz
GF12, 2.5mm x 2mm

More than an architecture: **MemPool Ecosystem**

SOFTWARE

- Wide software **kernel library**
- Bare-metal **runtime**, OpenMP, Halide
- **GCC** and **LLVM** toolchain support
- Support for GVSOC and Banshee **platform emulators**

HARDWARE

BACKEND

- Mature backend flow in many **modern technologies**
- **2 tapeouts**

MINPOOL
by
Domenic Wüthrich, Manos Frouzakis, Matheus Cavalcante and Samuel Riedel
PULP

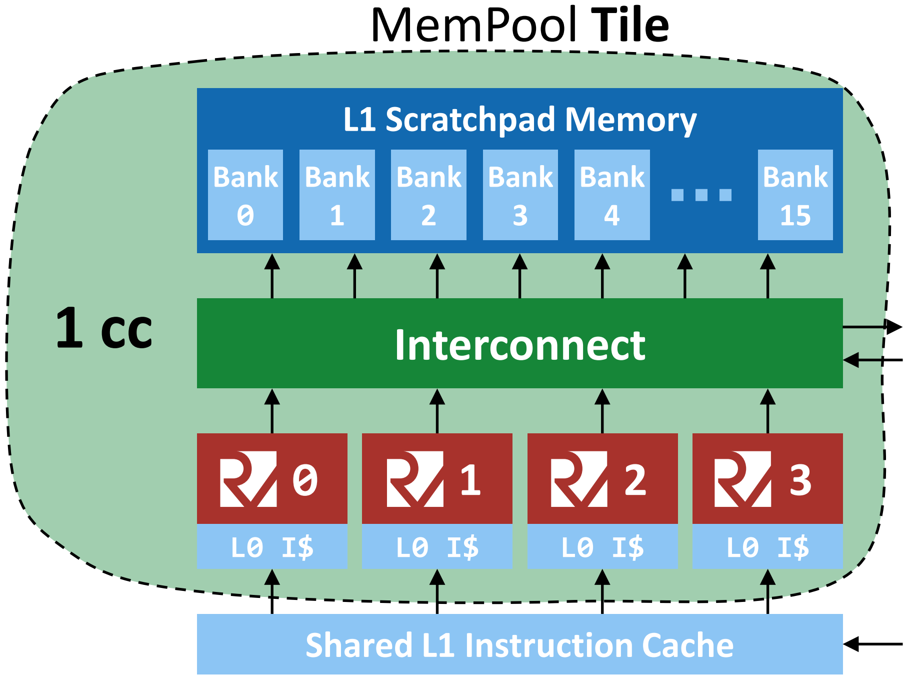
MinPool (2021)
16 cores, 200 MHz
TSMC65, 2.4mm x 2.4mm

Heartstream (2024)
64 cores, 720 MHz
GF12, 2.5mm x 2mm

Baseline MemPool: The hierarchical architecture

MemPool Tile

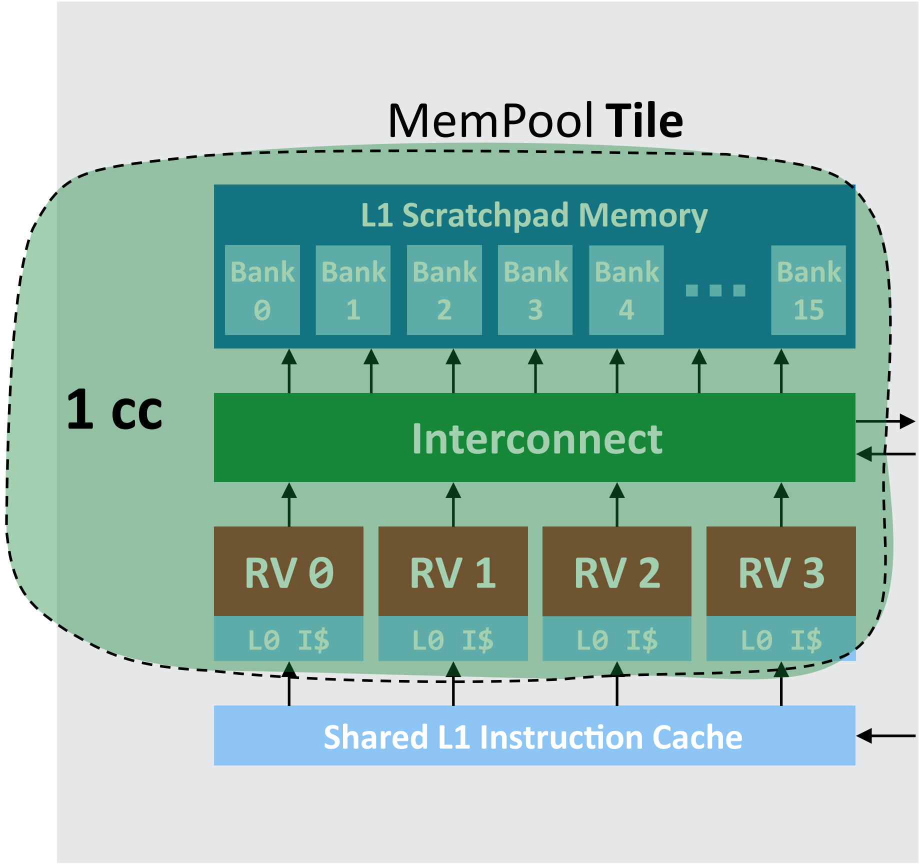
- **Four 32-bit Snitch cores**
  - Extensible, open-source RISC-V ISA
  - Independent instruction flow
  - **Latency-tolerant**
- **16 memory banks**
- **1-cycle latency**



Baseline MemPool: The hierarchical architecture

MemPool Tile

- 4 cores, 16 banks, 1-cc latency



Baseline MemPool: The hierarchical architecture

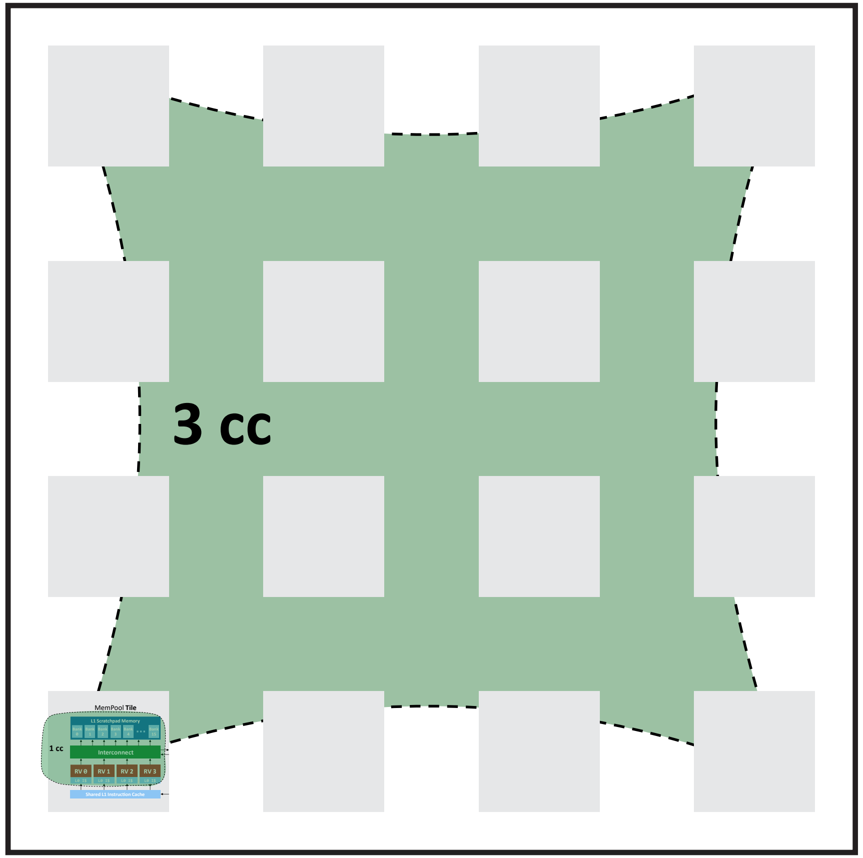
MemPool Tile

- 4 cores, 16 banks, 1-cc latency

MemPool Group

- **16 Tiles:** 64 cores, 256 banks, 3-cc latency

MemPool **Group**



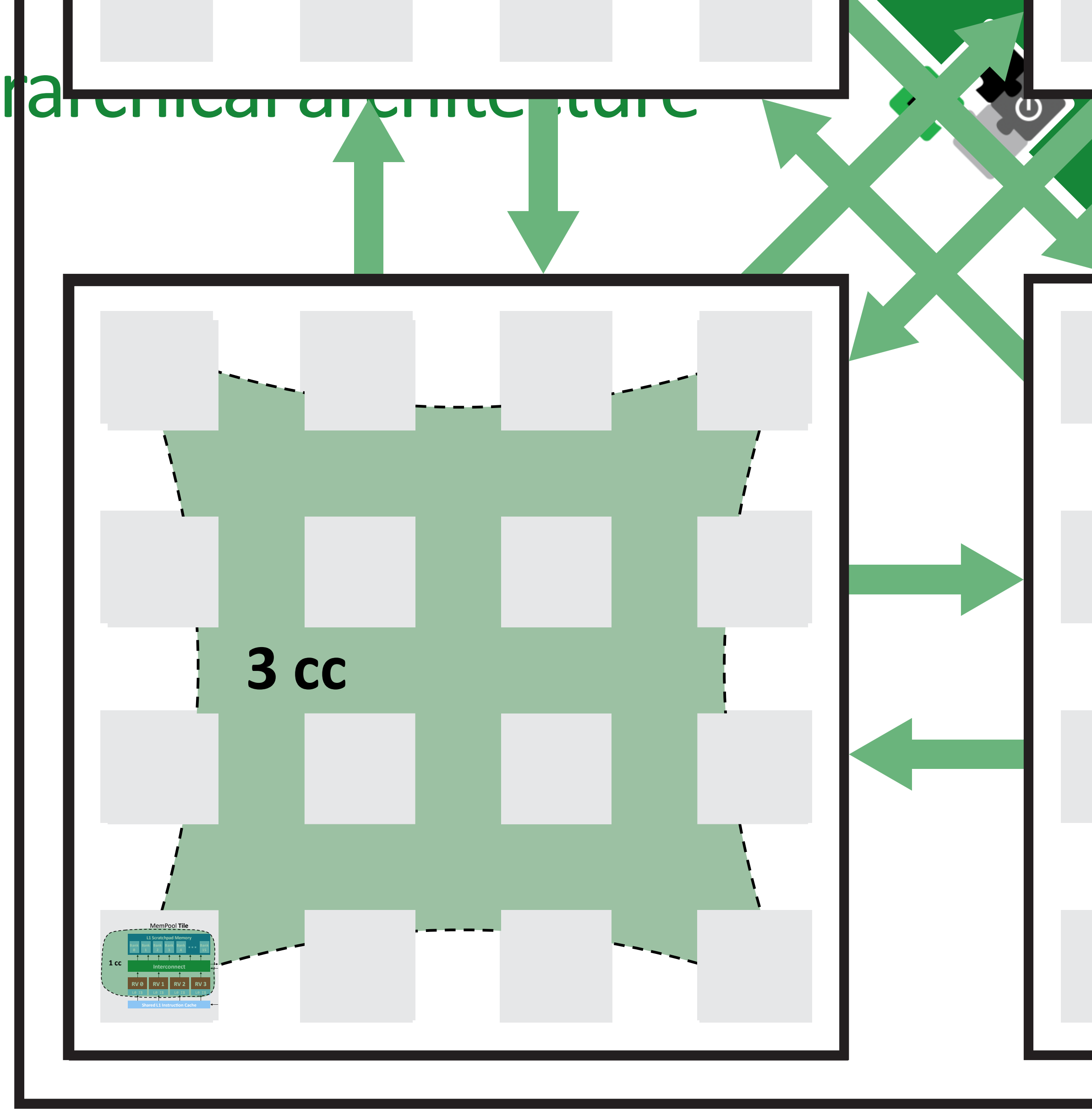
Baseline MemPool: The hierarchical architecture

MemPool Tile

- 4 cores, 16 banks, 1-cc latency

MemPool Group

- **16 Tiles:** 64 cores, 256 banks, 3-cc latency



Baseline MemPool: The hierarchical architecture

MemPool Tile

- 4 cores, 16 banks, 1-cc latency

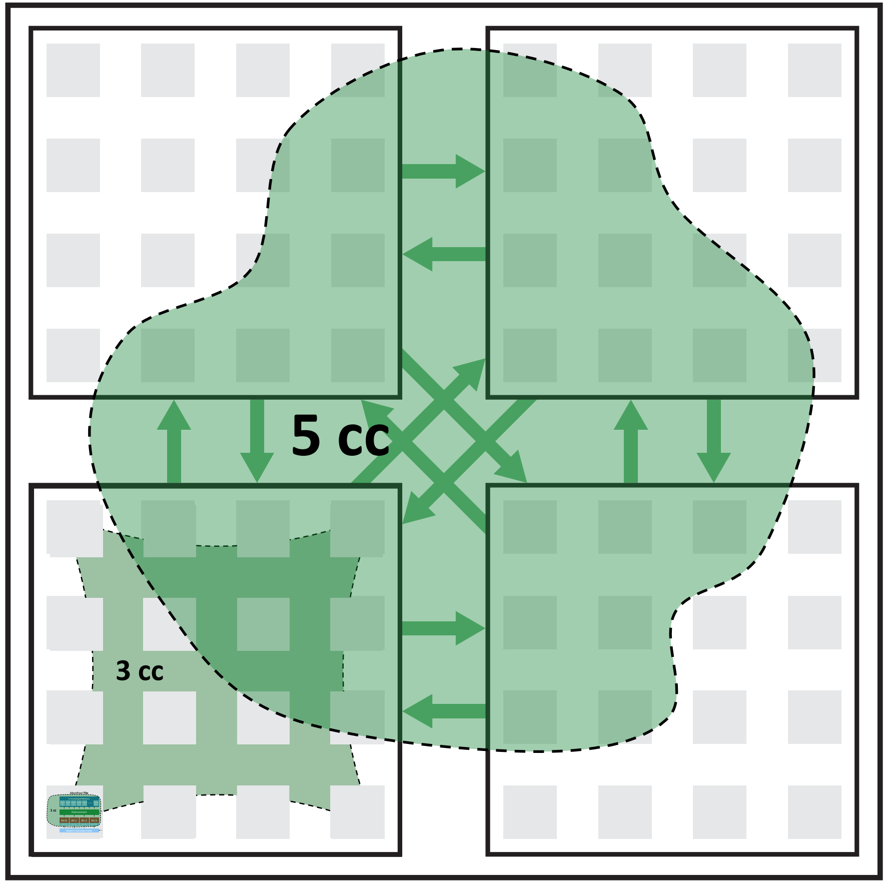
MemPool Group

- 16 Tiles: 64 cores, 256 banks, 3-cc latency

MemPool Cluster

- **4 Groups:** 256 cores, 1024 banks, 5-cc latency

MemPool **Cluster**



Baseline MemPool: The hierarchical architecture

MemPool Cluster

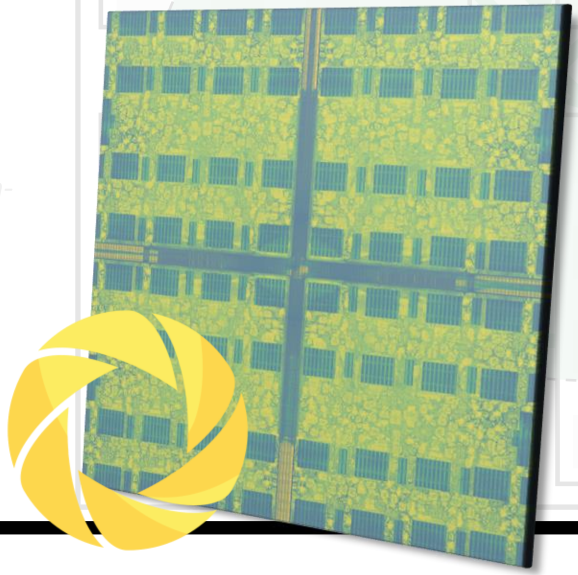
https://develope nvidia.com/blog/autodmp-opti design-with-ai-and-gpus/

- **Physically-aware design**

MemPool: A Scalable Manycore Architecture with a Low-Latency Shared L1 Memory

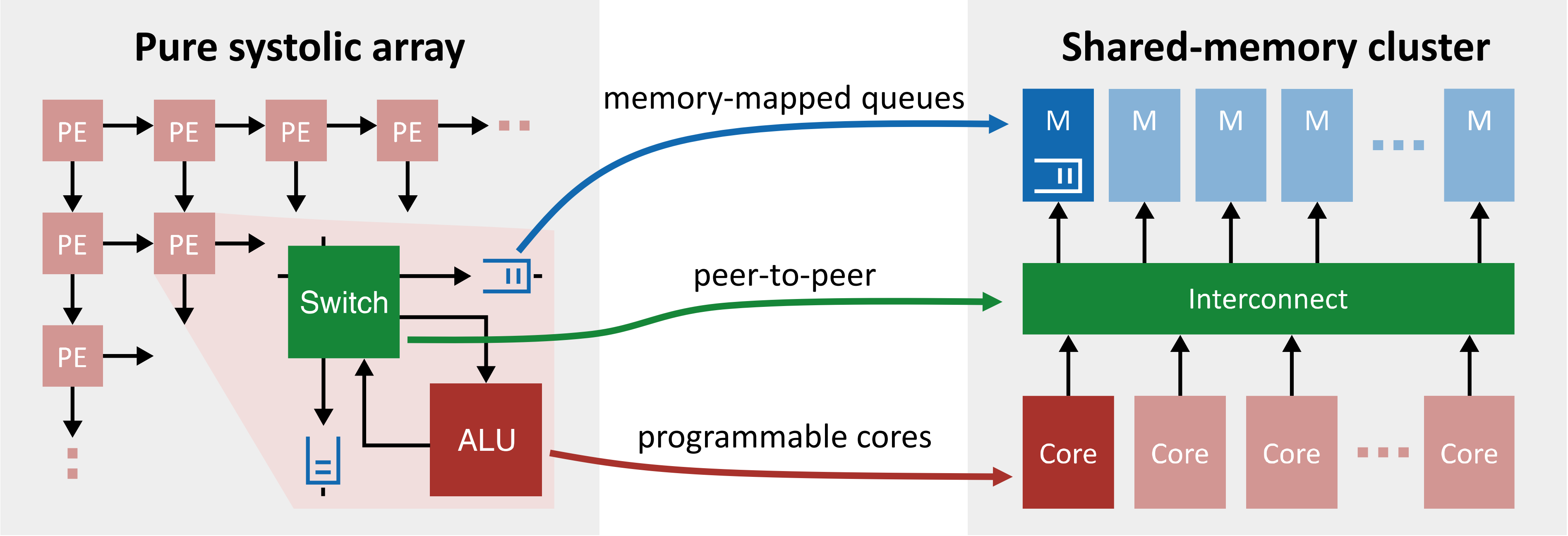
Samuel Riedel et al.

IEEE Transactions on Computers, 2023



Systolic MemPool: A hybrid architecture

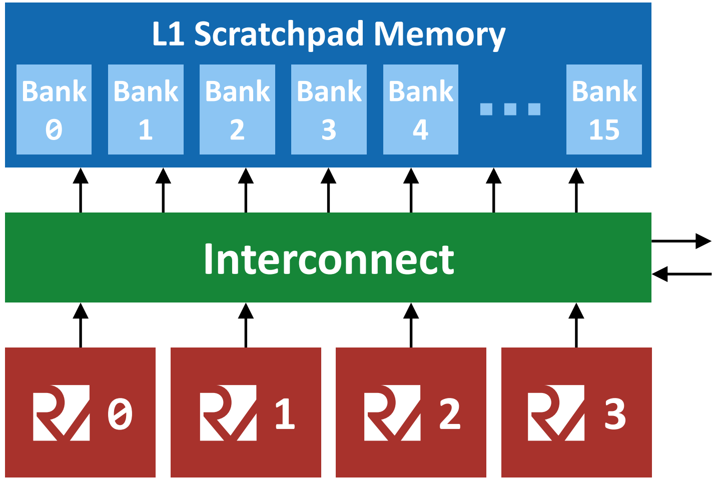
- **Efficient systolic dataflow in shared-memory**
  - Leverage **regular dataflow** of systolic workloads
  - Keep the **flexibility** of a shared-memory system



Systolic MemPool: An ISA for systolic queues

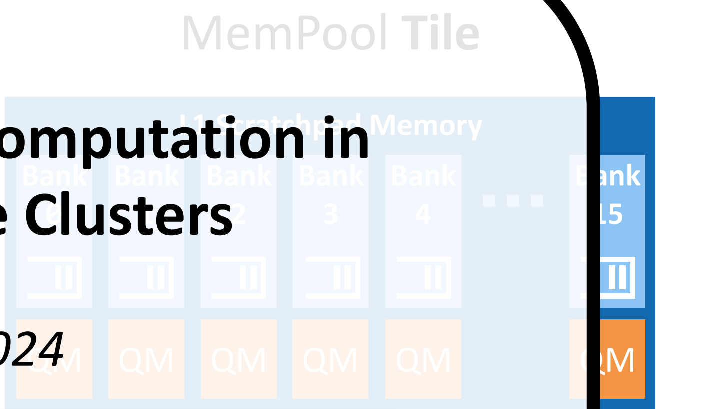
- **Low-overhead ISA extensions**

MemPool **Tile**



Systolic MemPool: An ISA for systolic queues

- **Low-overhead ISA extensions**
  - Queue Manager (QM) in hardware
    - 1-instruction access to queues
  - Autonomous access to queues
    - Elides loads & stores: **autonomous dataflow**
- **Any systolic topology**
- **Reconfigurable at ru**



Enabling Efficient Hybrid Systolic Computation in Shared-L1-Memory Manycore Clusters

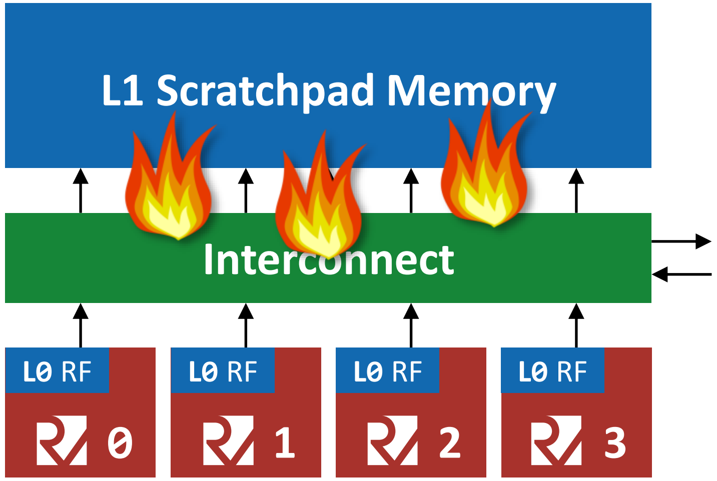
Sergio Mazzola et al.

IEEE Transactions on VLSI, 2024

Vectorial MemPool: Decreasing L1 bandwidth

- **Intuition**
  - Von Neumann bottleneck 🔥
  - Trade off **larger L0** size for **lower L1 bandwidth**
  - Higher data reuse **closer** to functional units
- **Data L0 = core's register file**
  - Scalar architecture: *L0 size not a knob*
  - **Vectorial** architecture: *VRF flexible by design*

MemPool **Tile**



Vectorial MemPool: Exploit DLP with vectors

- **Spatz**
  - Exploit **data-level parallelism** (SIMD) with a flock of **short-vector machines**
  - Based on **RVV ISA**
  - Multiple, lane-dedicated memory ports
  - Optimized sparse accesses (gather/scatter)
- **64 cores**, 256 vect FP

MemPool Tile

L1 Scratchpad Memory

Interconnect

Vectorial RF

vFPU

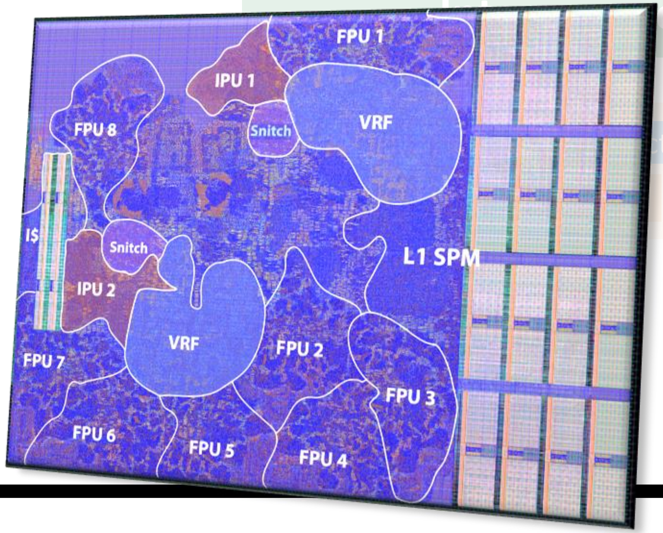
vFPU

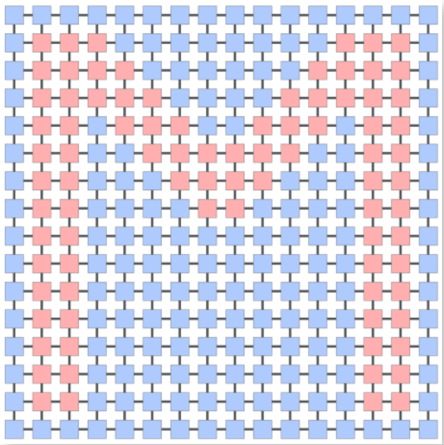
Spatz

Spatz: Clustering Compact RISC-V-Based Vector Units to Maximize Computing Efficiency

Matteo Perotti et al.

IEEE TCAD, 2025





Flavor Tasting

aka, the result section…

Matmul again… what flavor to pick?

- **Baseline MemPool**
- **Systolic MemPool**
  - 256 cores
  - 256 FPUs & Int DSP units
- **Vectorial MemPool**
  - 64 cores
  - 64 Vector Units (= 256 FPUs)

Utilization of compute units

[32-bit floating-point **matmul**]

✓ high compute intensity

✓ optimized for all MemPool flavors

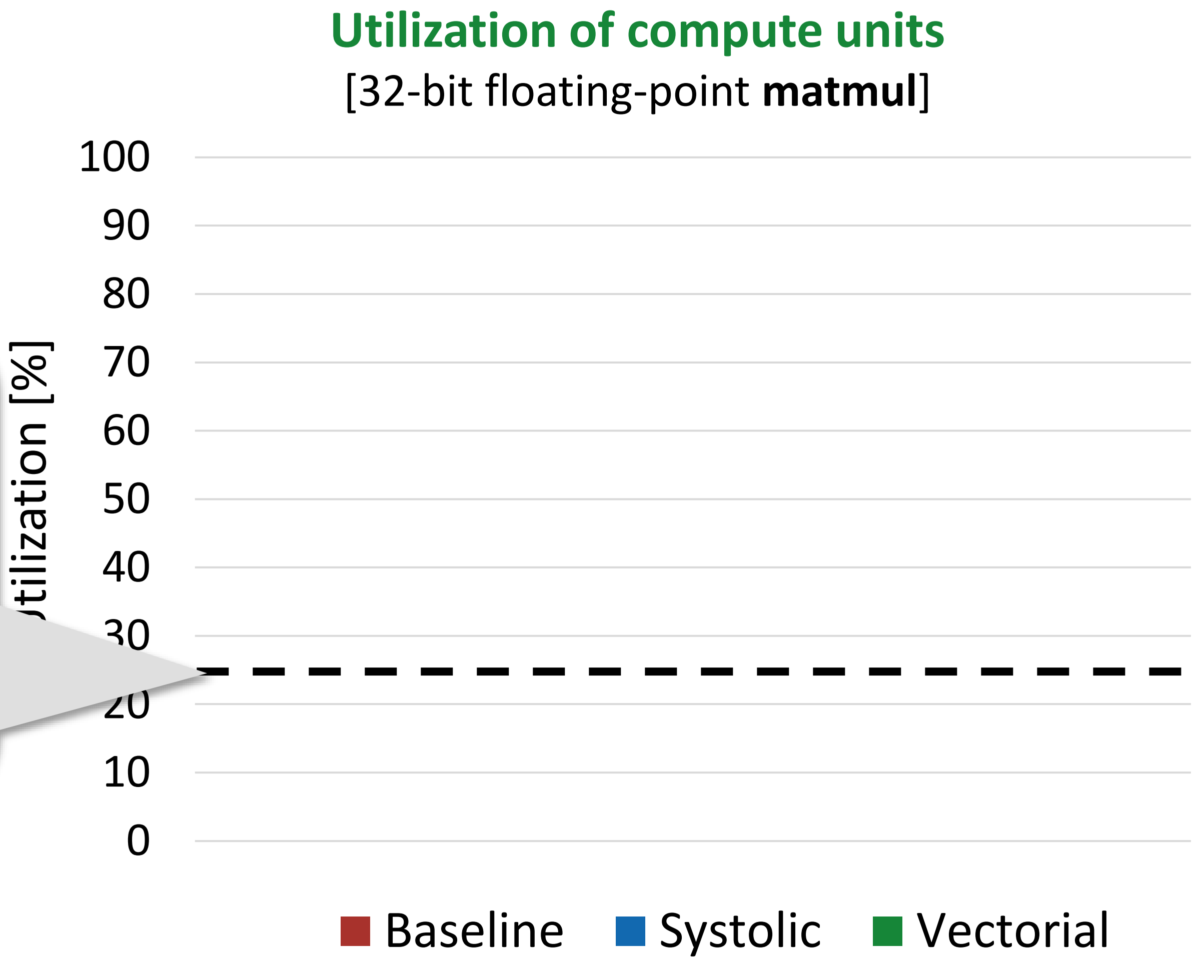
✓ key kernel for ML, DSP, …

Matmul again… what flavor to pick?

- **Baseline MemPool**
- **Systolic MemPool**
    - 256 cores

```
load f1, A[0][0]
load f2, B[0][0]
mac f3, f1, f2
```

256 FPUs → 256 MACs/cycle
But here: ~**33% utilization!**

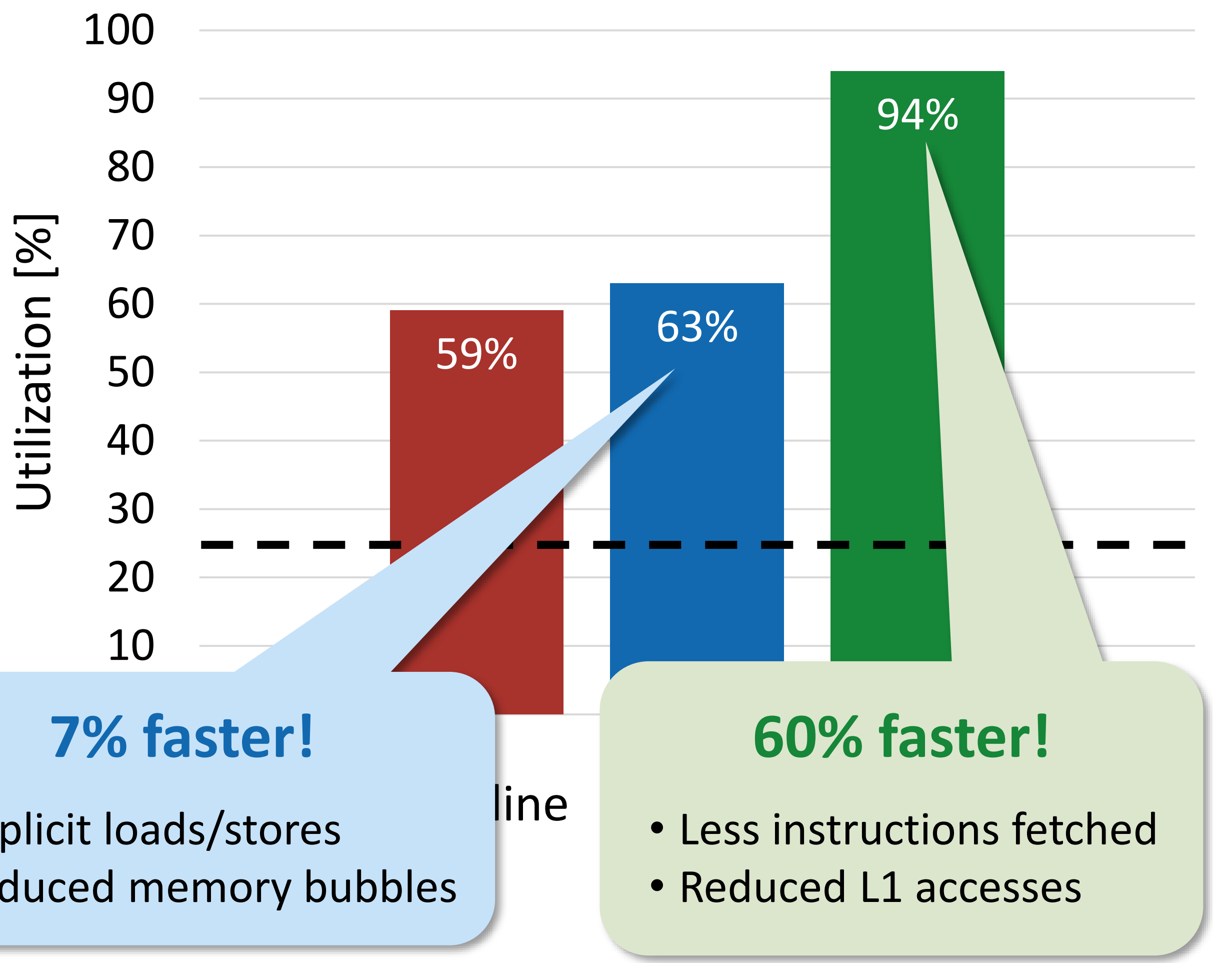


Matmul again... what flavor to pick?

- **Baseline MemPool**
- **Systolic MemPool**
  - 256 cores
  - 256 FPUs & Int DSP units
- **Vectorial MemPool**
  - 64 cores
  - 64 Vector Units (= 256 FPUs)

Utilization of compute units

[32-bit floating-point **matmul**]



7% faster!

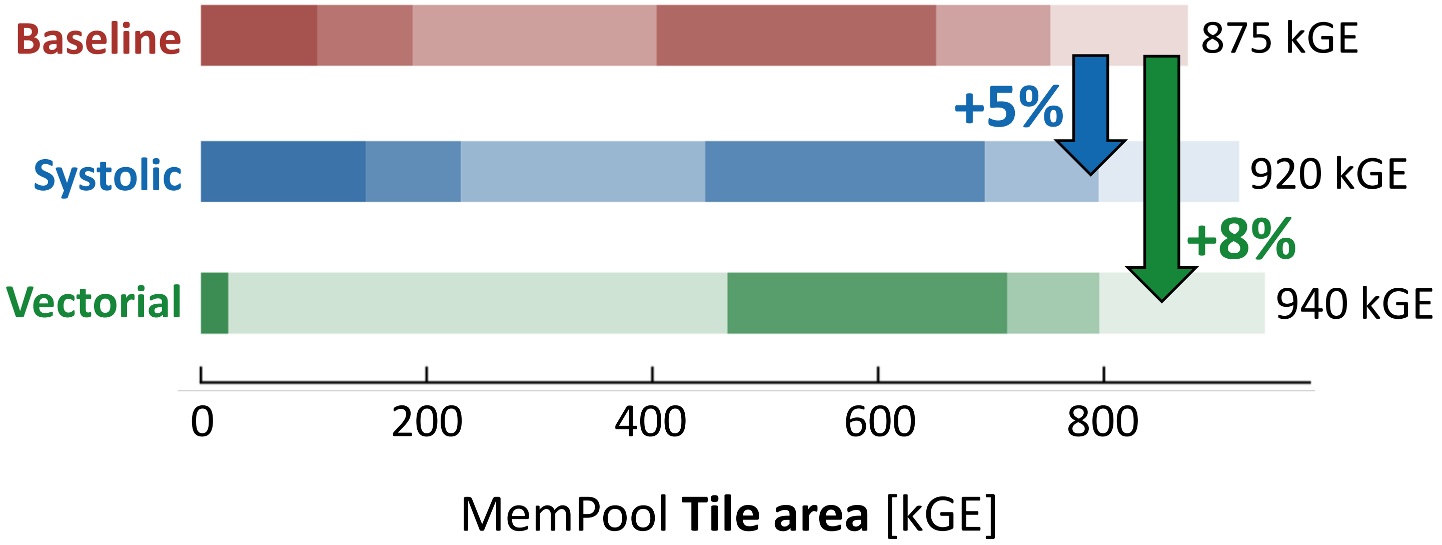
- Implicit loads/stores
- Reduced memory bubbles

60% faster!

- Less instructions fetched
- Reduced L1 accesses

...It depends on what you want

- **Baseline MemPool**
- **Systolic MemPool**
  - 256 cores
  - 256 FPUs & Int DSP units
- **Vectorial MemPool**
  - 64 cores
  - 64 Vector Units (= 256 FPUs)

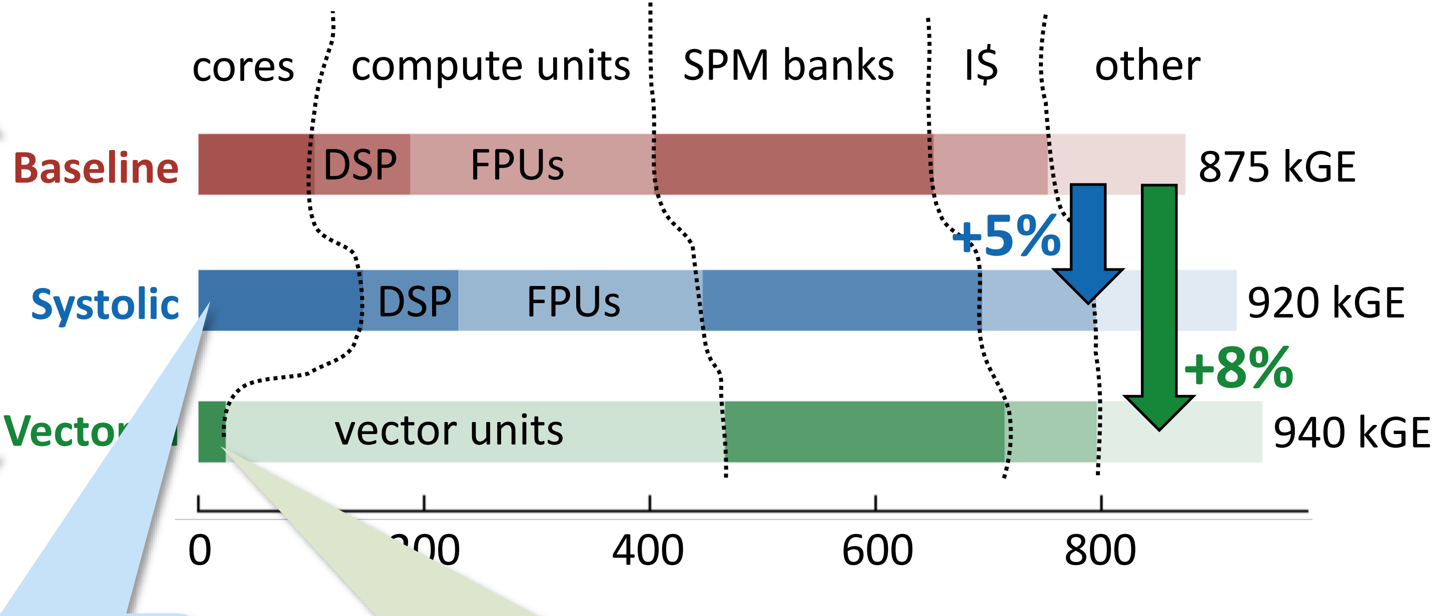


Technology & target
GF 12nm FinFET
800 MHz, worst-case corner

...It depends on what you want

- **Baseline MemPool**
- **Systolic MemPool**
  - 256 cores
  - 256 FPUs & Int DSP units
- **Vectorial MemPool**
  - 64 cores
  - 64 Vector Units (= 256 FPUs)

flexibility
specialization



Technology & target
GF 12nm FinFET
800 MHz, worst-case corner

- Extra logic FIFOs, queues control

- **1 core per Tile only**
- Extra logic for control (vect load/store, shuffle, ...)

MemPool: An open-source, RISC-V research platform

- **Scaled-up cluster covering a large trade-off space**
  - From classic shared-memory cluster…
  - …to exotic hybrid architectures
  - but always with **versatility** and **programmability**
- **Lively ecosystem since 2020**
- **Multiple tapeouts**

github.com/pulp-platform/mempool

M. Cavalcante et al., "**MemPool: A shared-L1 memory many-core cluster with a low-latency interconnect**," in 2021 Des. Autom. Test Eur. Conf. Exhib. Grenoble, France: IEEE, Mar. 2020

E. Beyne et al., "**3D SoC integration, beyond 2.5D chiplets**," 2021 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2021

M. Cavalcante et al., "**MemPool-3D: Boosting Performance and Efficiency of Shared-L1 Memory Many-Core Clusters with 3D Integration**," 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 2022

A. Agnesina et al., "**Hier-3D: A Hierarchical Physical Design Methodology for Face-to-Face-Bonded 3D Ics,**" in Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '22). 2022

M. Cavalcante et al., "**Spatz: A Compact Vector Processing Unit for High-Performance and Energy-Efficient Shared-L1 Clusters**," 2022 IEEE/ACM International Conference On Computer Aided Design (ICCAD), San Diego, CA, USA, 2022

S. Venkateswarlu et al., "**Thermal Performance Analysis of Mempool RISC-V Multicore SoC**," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Nov. 2022