ETH zürich

MemPool Meets Systolic: Flexible Systolic Computation in a Large Shared-Memory Processor Cluster

Samuel Riedel, Gua Hao Khov, Sergio Mazzola, Matheus Cavalcante, Renzo Andri, and Luca Benini
Integrated Systems Laboratory (IIS), ETH Zürich, Switzerland

High performance + high flexibility?

Systolic Architectures

+ Highly efficient for specific workloads
− Very rigid execution scheme, target-specific

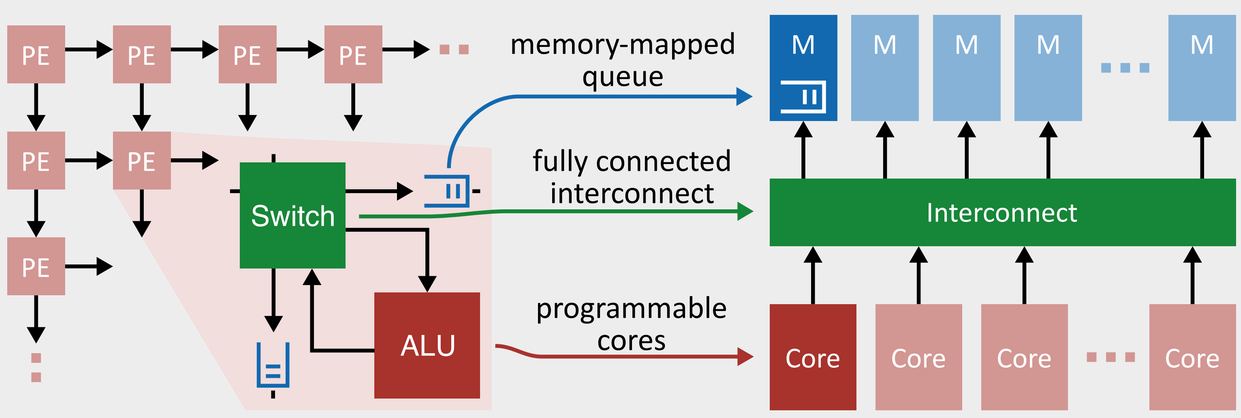
Shared-memory Clusters

+ General-purpose processing (flexibility, programmability)
− Communication overhead lowers throughput

Combine the best of both worlds

Hybrid architecture

Systolic array ⟷ Shared-memory cluster



Hybrid Architecture: *Efficient systolic execution on shared-memory systems*

- Get the performance of a systolic array for suitable workloads
- Keep the flexibility of a shared-memory system

Faster inter-core communication

Baseline

Emulate communication in **software**

+ Flexible, memory-mapped queues
+ Easy exploration
- High overhead

```
c = 0;
for (i=0; i<N; i++) {
  a = queue_pop(qa_in);
  b = queue_pop(qb_in);
  c += a * b;
  queue_push(a, qa_out);
  queue_push(b, qb_out);
}
```

Xqueue extension

Hardware queue **push** and **pop** instructions

+ Single-instruction queue access
- Fixed number of available queues

```
c = 0;
for (i=0; i<N; i++) {
  a = __builtin_pop(qa_in);
  b = __builtin_pop(qb_in);
  c += a * b;
  __builtin_push(a, qa_out);
  __builtin_push(b, qb_out);
}
```

Queue-linked Register

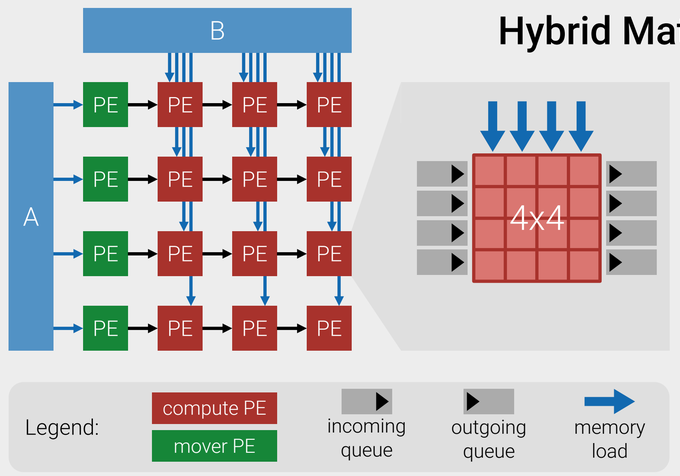
Hardware for autonomous queue management

+ No communication overhead
- Small setup cost

```
c = 0;
setup_qlr(a, qa);
setup_qlr(b, qb);
for (i=0; i<N; i++) {
  c += a * b;
}
```

Hybrid systolic-shared-memory algorithms

- *Hybrid architecture:* allows exploring **new systolic topologies**
- *Hybrid algorithms:* use the flexibility of general-purpose shared-memory architecture to boost the performance of systolic algorithms even further



Hybrid Matrix Multiplication

Mover PEs

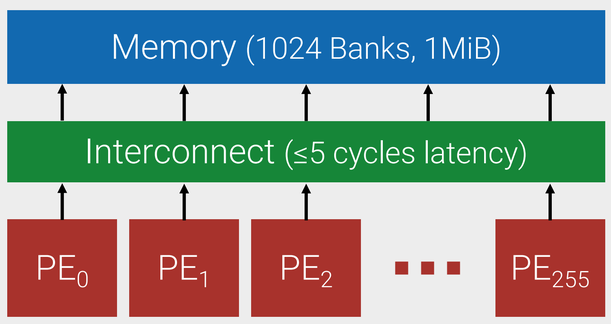
- Load A (memory)
- Push to the systolic array

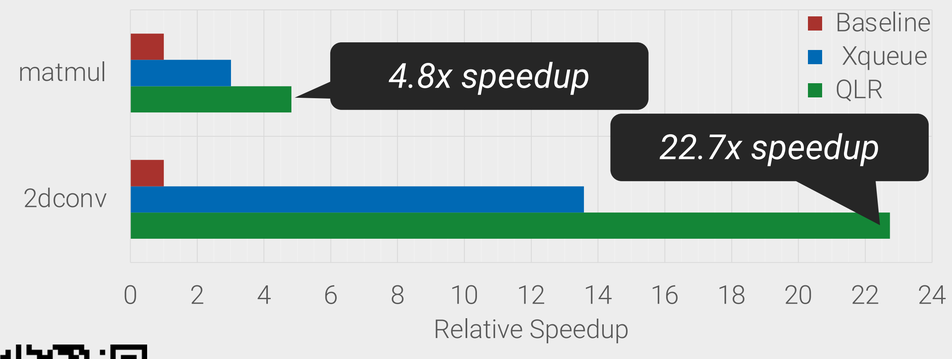
Compute PEs

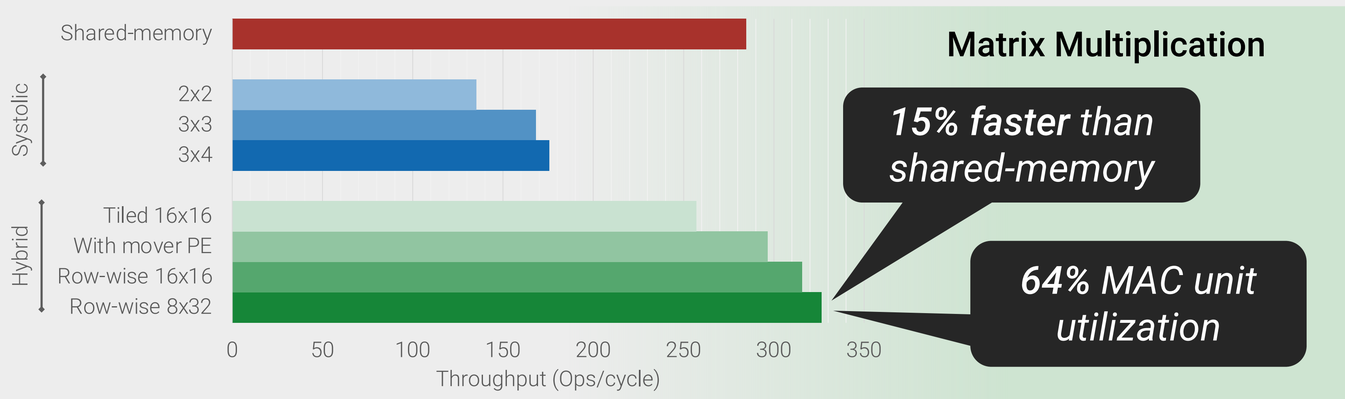
- Get A (systolic)
- Load B (memory)

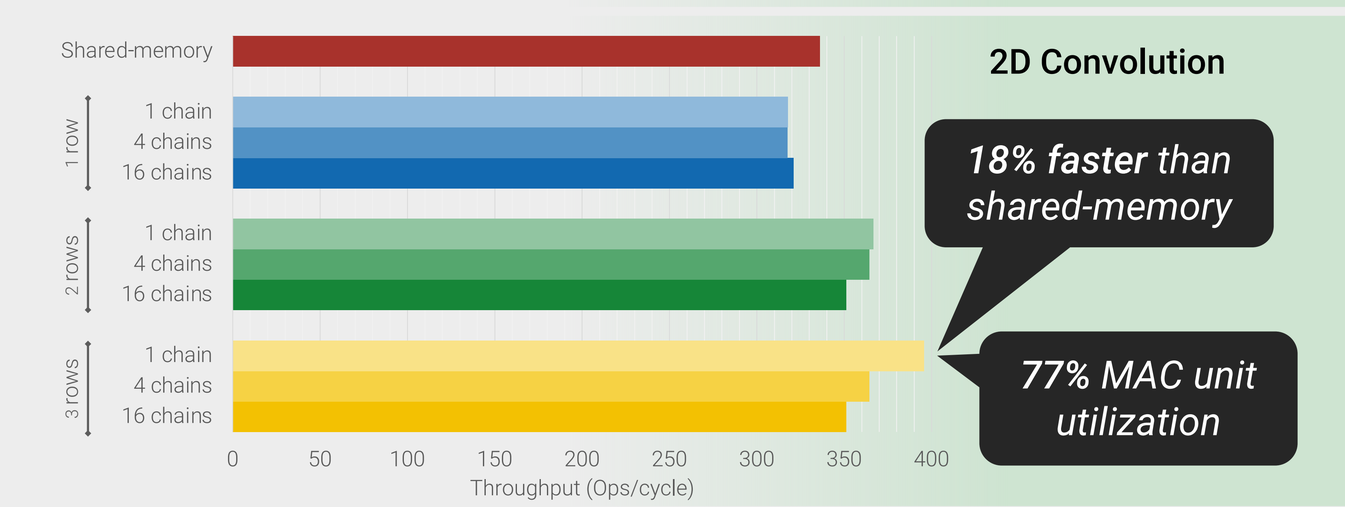
Results

Implemented & evaluated on **MemPool**









◁ open-source on GitHub!

sriedel@iis.ee.ethz.ch pulp-platform.org @pulp_platform