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

Faster inter-core communication

Baseline
Emulate communication in software
- Flexible, memory-mapped queues
- Easy exploration
- High overhead

Xqueue extension
Hardware queue push and pop instructions
- Single-instruction queue access
- Fixed number of available queues

Queue-linked Register
Hardware for autonomous queue management
- No communication overhead
- Small setup cost

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

Results
Implemented & evaluated on MemPool

- Memory (1024 Banks, 1MB)
- Interconnect (≤5 cycles latency)

matmul
4.8x speedup

2dconv
22.7x speedup

Shared-memory

Hybrid

Matrix Multiplication
15% faster than shared-memory

64% MAC unit utilization

2D Convolution
18% faster than shared-memory

77% MAC unit utilization

< open-source on GitHub!

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