RIVETS: An Efficient Training and Inference Library for RISC-V with Snitch Extensions

Andrei Ivanov[1], Timo Schneider[1], Luca Benini[1,2], Torsten Hoefler[1]
[1]Department of Computer Science, ETH Zurich;
[2]Department of Information Technology and Electrical Engineering, ETH Zurich

1 Introduction

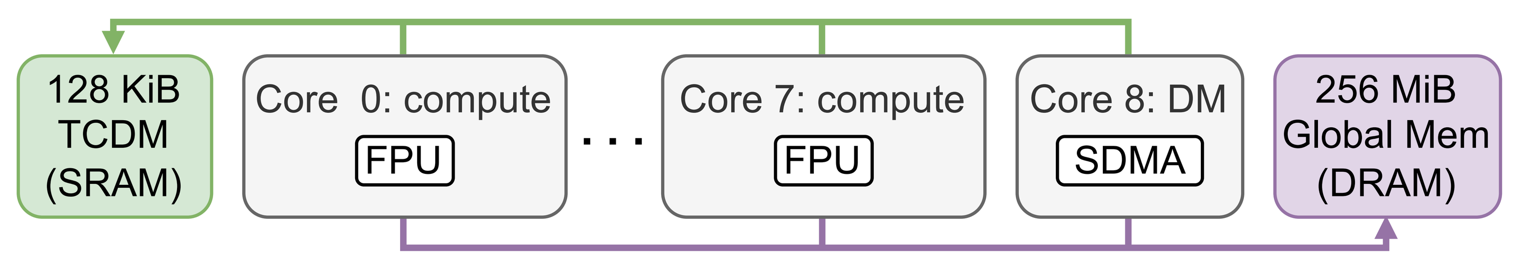
- There is rising interest in using RISC-V to do DL **training** [3]
- Library optimized for **floating-point** computations is needed
- Library should follow specifications of popular APIs for DL

cuDNN ONNX oneAPI

2 Existing works

	muRISCV-NN	PULP-NN	XNNPACK	OneDNN
Extensions	Vector "V" Packed "P"	Xpulp	Vector "V"	Vector "V"
Precision	Integer	Subbyte-quantized integer	Floating-point	Floating-point
Kernels	Softmax, Pooling, Conv, LSTM, SVD, ReLu, Sigmoid	Add, Pooling, Linear, MatMul	Sqrt, Sqr, Abs, Neg, HSwish, Clamp	Pooling

3 Target Platform: Snitch [4]



Overview of the Snitch cluster environment

- SSR: Stream Semantic Registers [5] → "register access acts as streamed memory access" fadd.d ft3, ft0, ft3 / fadd.d ft3, ft0, ft3
- SDMA: Snitch asynchronous data movement
- SmallFloat [6]: Support of fp8, fp16, fp32, fp64
- FREP: Floating-point repetition → "repeat N=1 instruction M=5 times" frep.o 5, 1, 0, 0 / fadd.d ft3, ft0, ft3
- TCDM: Tightly Coupled Data → fast scratchpad memory

4 Optimization example: LayerNorm

D = dst-src // distance between src and dst arrays in memory
P = beta-gamma

$$\mathrm{dst}(b,n) = \gamma(n) \cdot \frac{\mathrm{src}(b,n) - \mu(b)}{\sqrt{\sigma^2(b) + \varepsilon}} + \beta(n)$$

	SSR 0	SSR 1	SSR 2
shapes:	[B, 2, 2, N]	[B, N, 2]	[B, 2, N]
strides:	[N, D, 0, 1]	[N, 1, P]	[N, 0, 1]

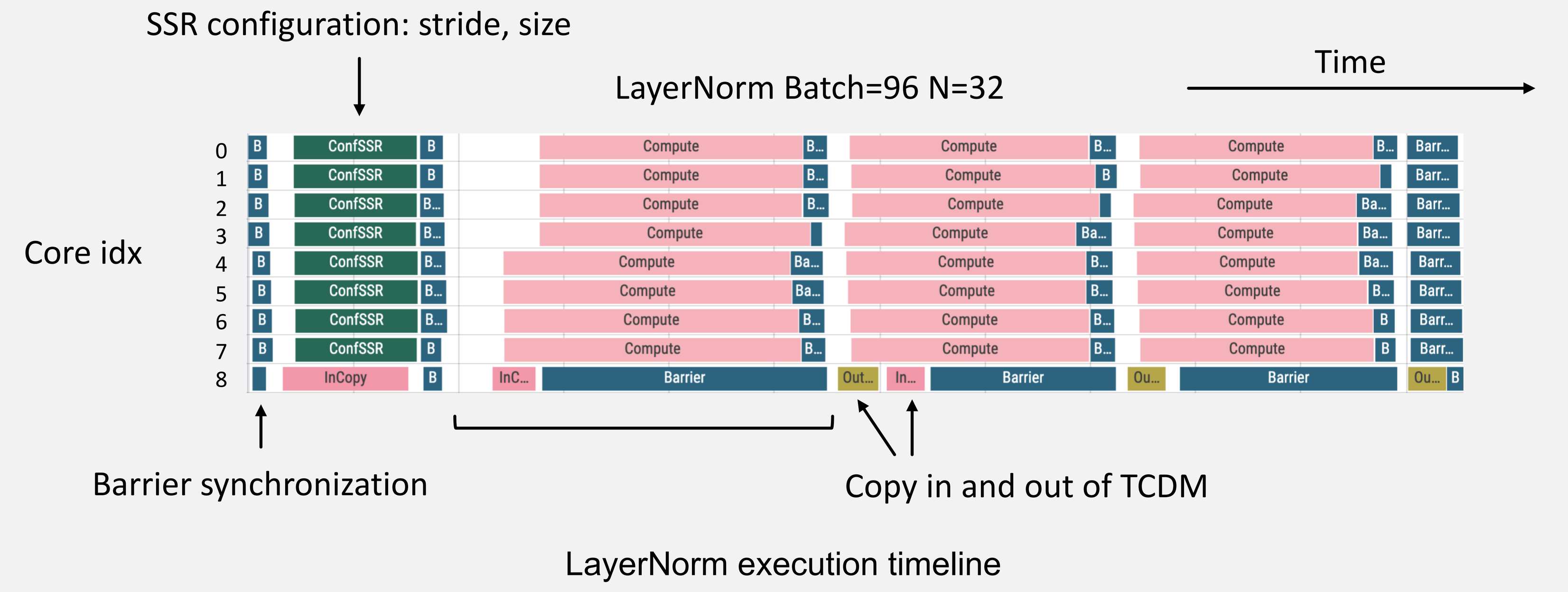
```
for (size_t b = 0; b < B; b++) {
    mu[b] = 0;
    for (size_t n = 0; n < N; n++) {
        mu[b] += src[b * N + n];
    }
    mu[b] /= N;
    sigma[b] = 0;
    for (size_t n = 0; n < N; n++) {
        dst[b * N + n] = src[b * N + n] - mu[b];
    }
    for (size_t n = 0; n < N; n++) {
        sigma[b] += SQR(dst[b * N + n]);
    }
    sigma[b] = 1.0 / SQRT(sigma[b] / (N - 1) + eps);
    for (size_t n = 0; n < N; n++) {
        dst[b * N + n] = gamma[n] * dst[b * N + n] * sigma[b] + beta[n];
    }
}
```

The use of SSR and FREP extensions to optimize LayerNorm performance

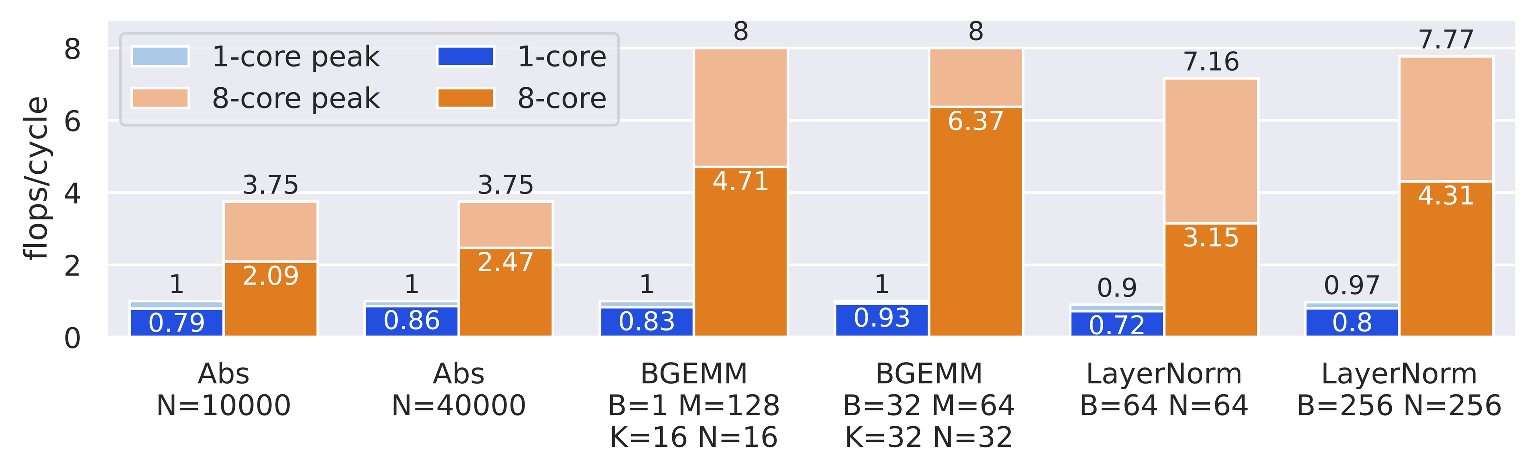
5 Operations per cycle in the Snitch core

Func. block	Operation	Peak ops/cycle	latency [cycles]
ADDMUL	fma, add, mul	1	4
DIVSQRT	sqrt, div	0.05	22
COMP	min, max, abs	1	1
SDMA	byte transfer	60	166

6 Evaluation

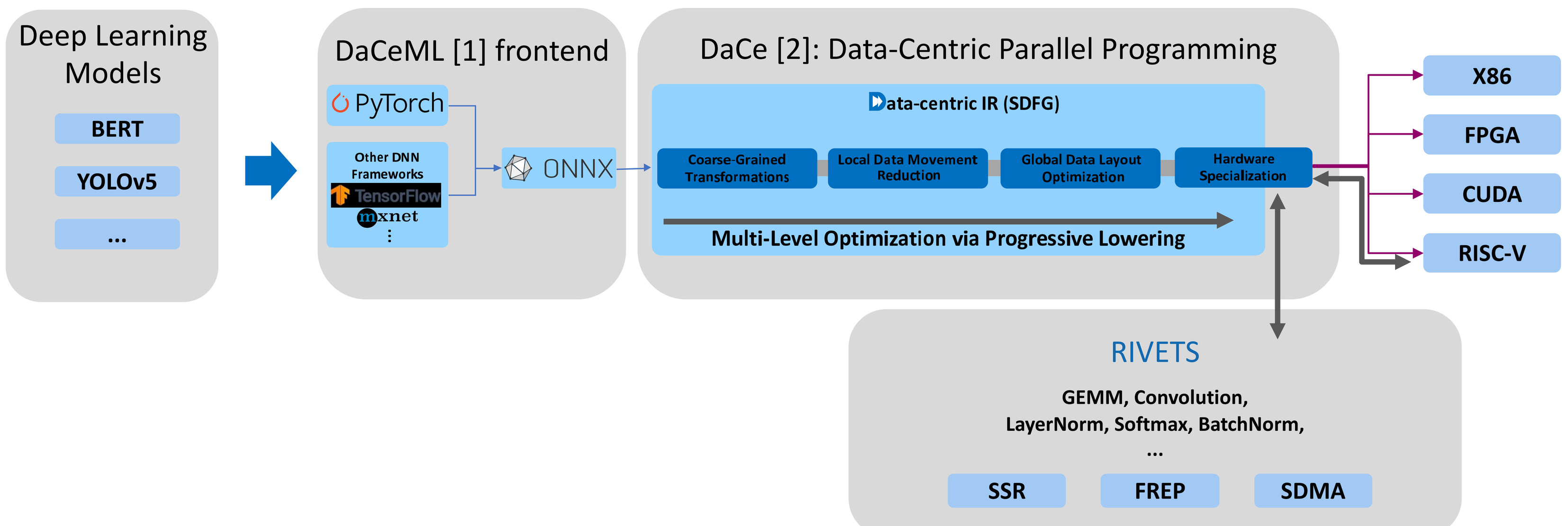


LayerNorm execution timeline



Performance of kernels on Snitch platform.

7 End-to-end model support



References

1. Oliver Rausch, Tal Ben-Nun, Nikoli Dryden, Andrei Ivanov, Shigang Li, and Torsten Hoefler. "A data-centric optimization framework for machine learning." ICS '22
2. Tal Ben-Nun, Johannes de Fine Licht, Alexandros N. Ziogas, Timo Schneider, and Torsten Hoefle. "Stateful dataflow multigraphs: A data-centric model for performance portability on heterogeneous architectures." SC '19
3. A. Garofalo et al., "DARKSIDE: A Heterogeneous RISC-V Compute Cluster for Extreme-Edge On-Chip DNN Inference and Training," in IEEE Open Journal of the Solid-State Circuits Society, vol. 2, pp. 231-243, 2022.
4. F. Zaruba, F. Schuiki, T. Hoefler and L. Benini, "Snitch: A Tiny Pseudo Dual-Issue Processor for Area and Energy Efficient Execution of Floating-Point Intensive Workloads," in *IEEE Transactions on Computers*, vol. 70, no. 11, pp. 1845-1860, 1 Nov. 2021.
5. F. Schuiki, F. Zaruba, T. Hoefler and L. Benini, "Stream Semantic Registers: A Lightweight RISC-V ISA Extension Achieving Full Compute Utilization in Single-Issue Cores," in *IEEE Transactions on Computers*, vol. 70, no. 2, pp. 212-227, 1 Feb. 2021.
6. G. Tagliavini, S. Mach, D. Rossi, A. Marongiu and L. Benini, "Design and Evaluation of SmallFloat SIMD extensions to the RISC-V ISA," *DATE '19*