Оптимизация нейронных сетей для запуска на ускорителях с использованием компиляторных оптимизаций

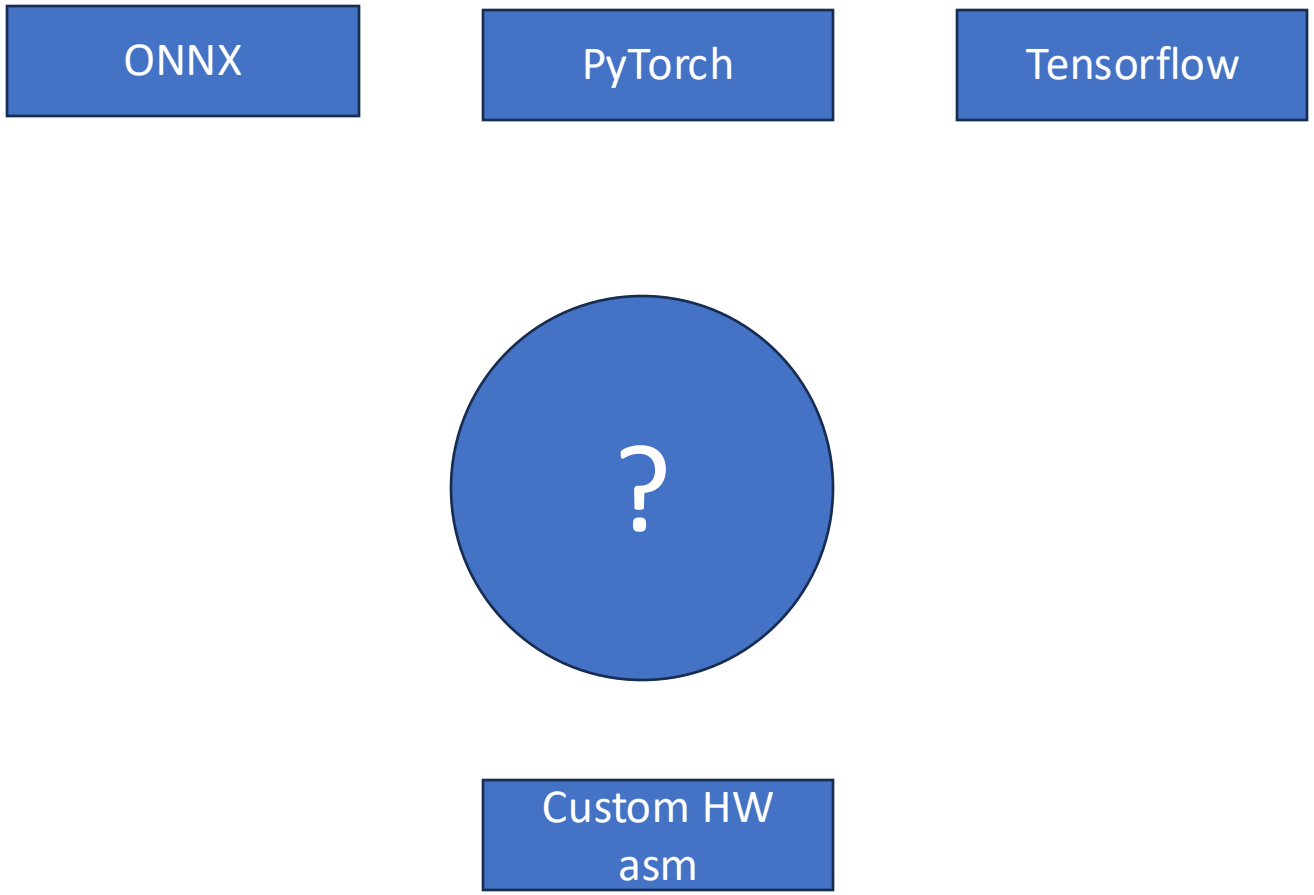
Оболенский Арсений Андреевич

Горшков Антон Валерьевич

Мееров Иосиф Борисович

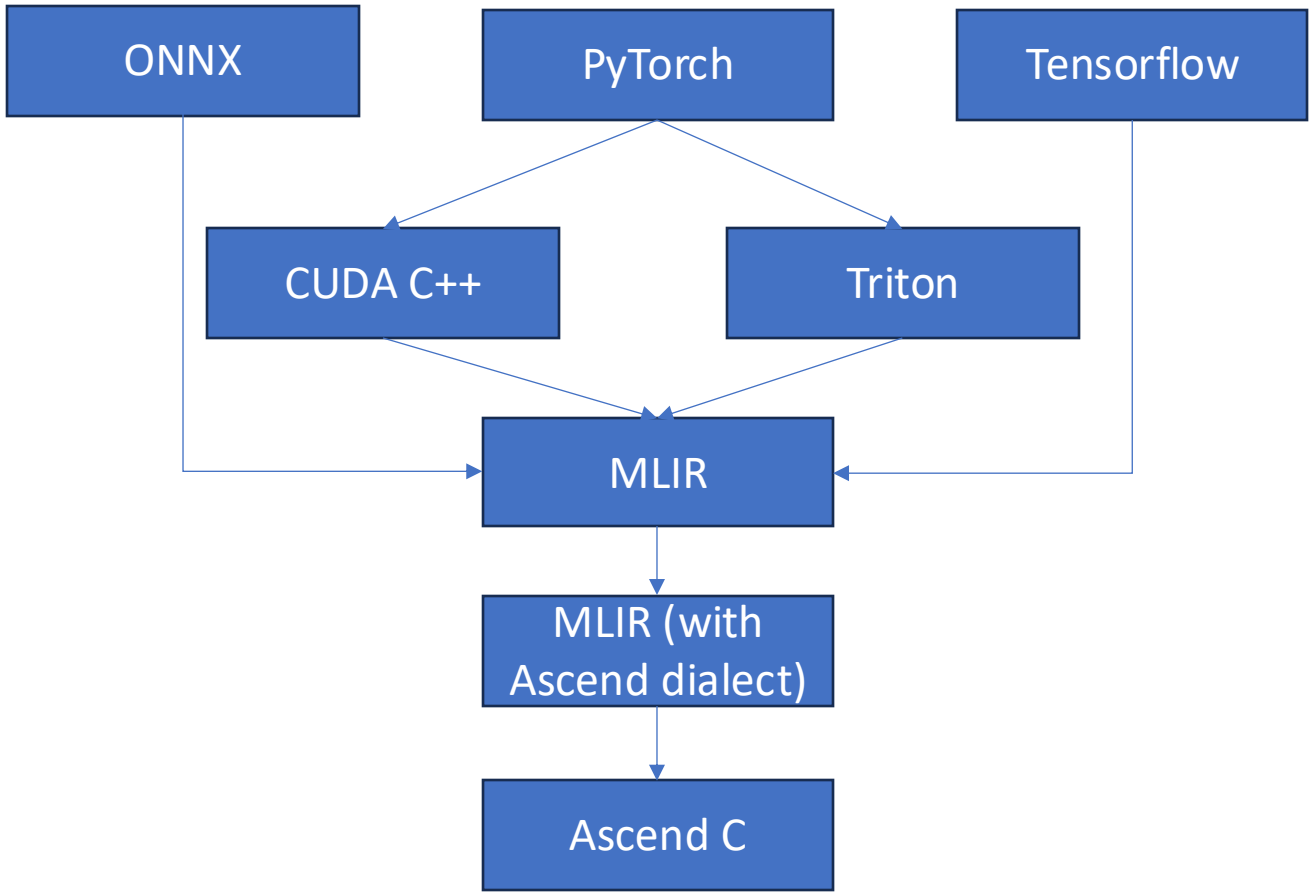
Motivation

- How to enable custom hardware?
- What approaches are suitable to convert higher level representation (NN graph, operators) to HW minimizing efforts?

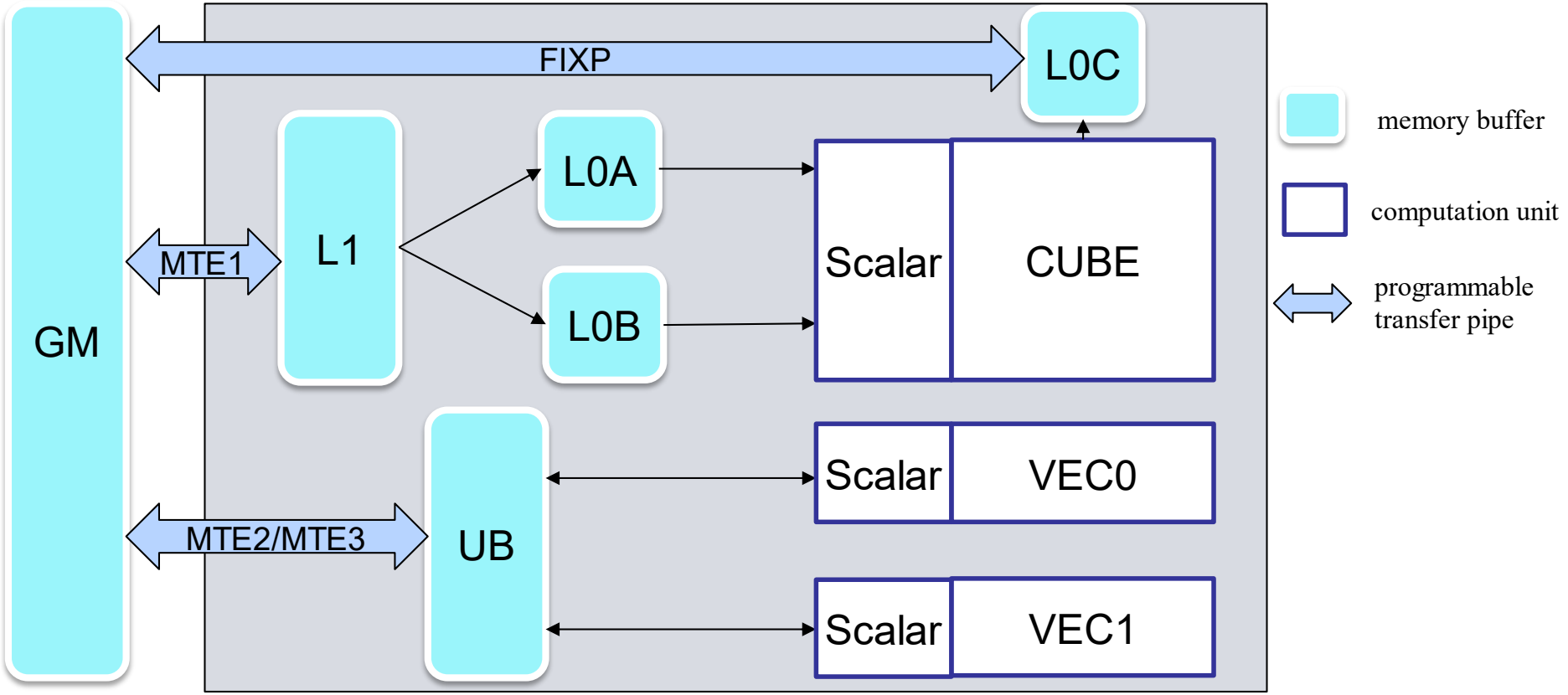


Current work

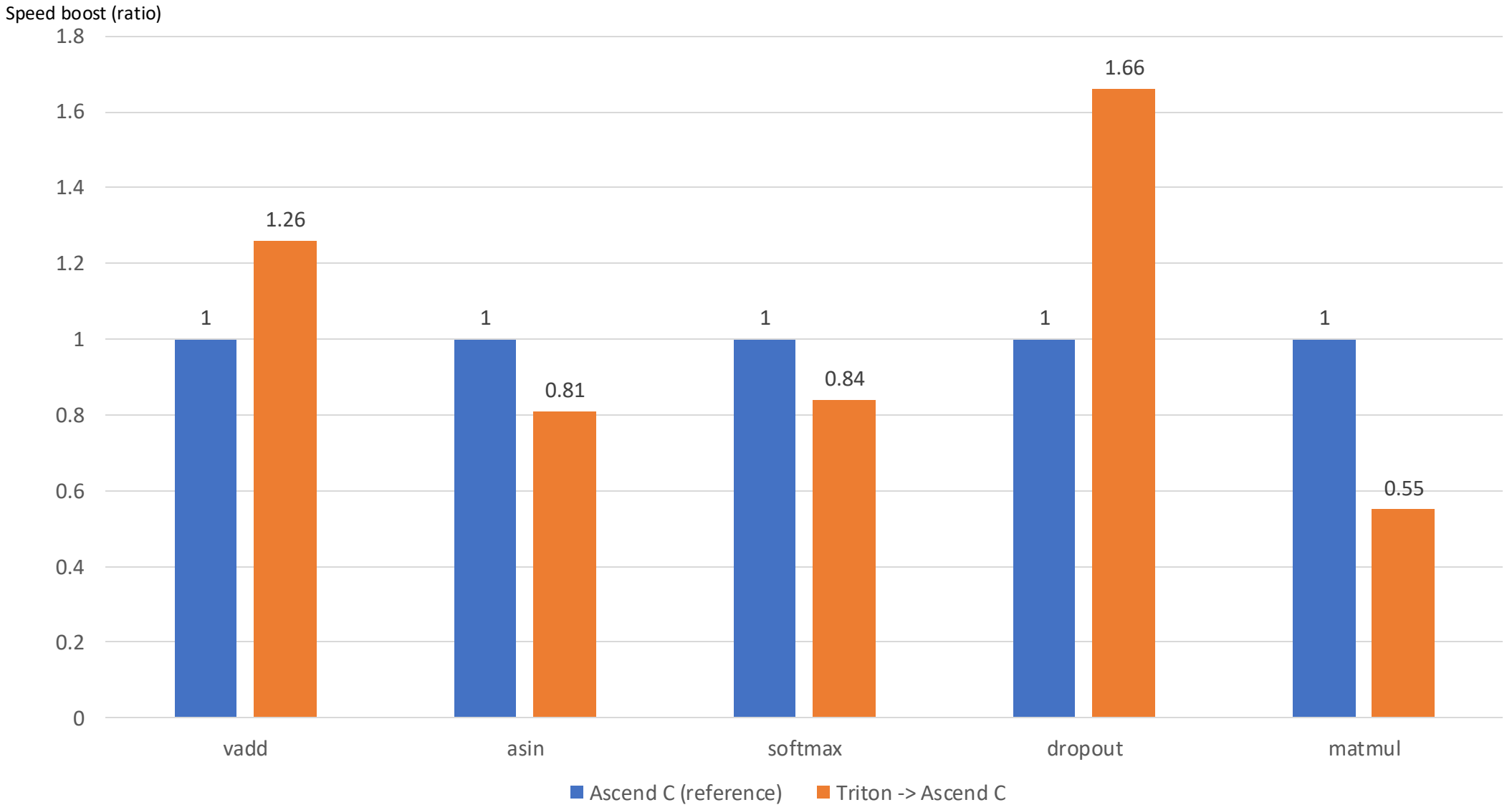
- Chosen (more or less) optimal FE to lower the code
- Chosen approach: MLIR based compiler
- Scoped, analyzed and imple,emted solution for lowering to sophisticated HW platform



Ascend910B architecture

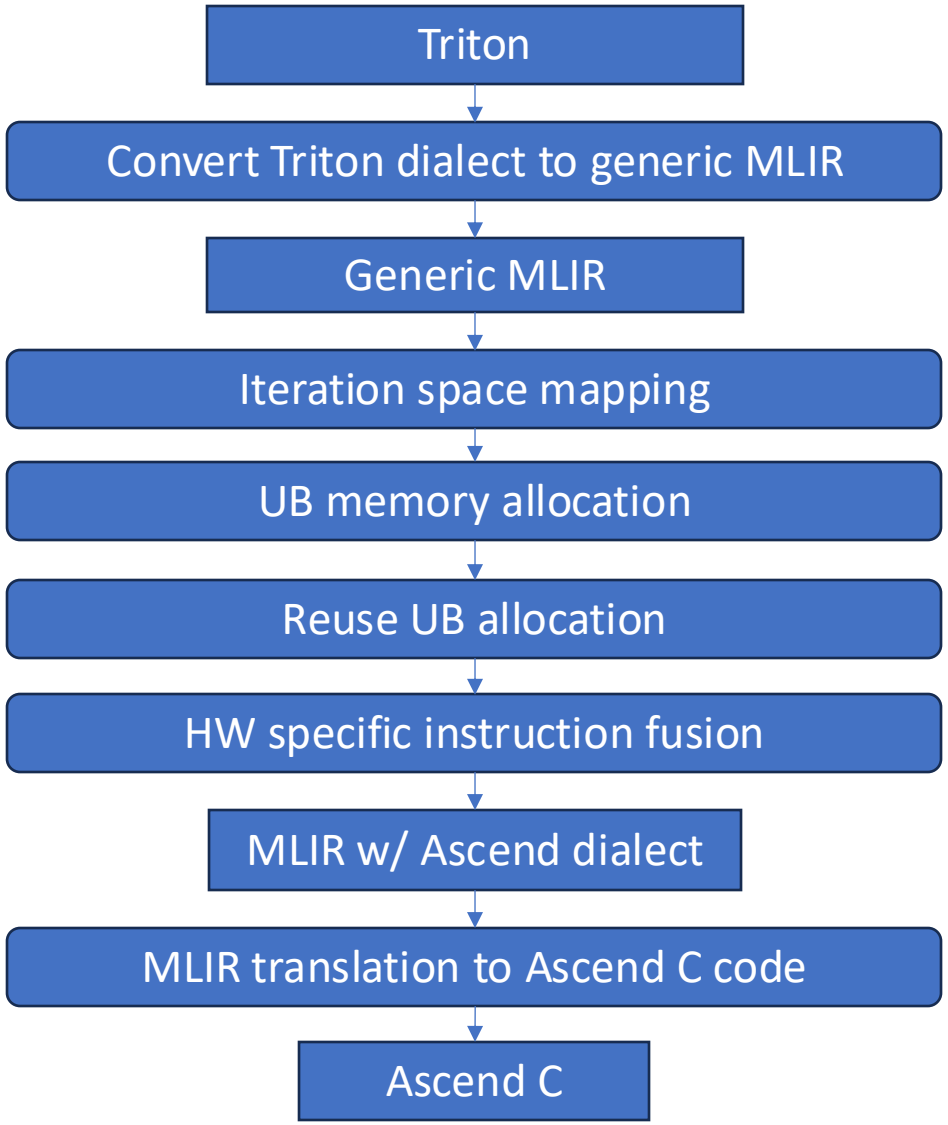


Results


Speed boost (ratio)
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1
1.26
1
0.81
1
0.84
1
1.66
1
0.55
vadd
asin
softmax
dropout
matmul
Ascend C (reference)
Triton -> Ascend C

MLIR pass pipeline

- Ascend platform enabling requires to solve a number of challenges:
  - Convert to SPMD SW architecture
  - UB memory allocation and allocation optimization
  - Scalar/vector/cube core calculations split
  - No publicly available assembler exposed
- Pass pipeline proposed, implemented and results obtained



Next plans

- Continue investigation in the field of different frontends
  - Possible options: PyTorch (or other ML frameworks compatible with MLIR)
- Finalize (productize) the software and publish it to the open source
- Continue investigating options of lowering to CPU architecture (ARM) using the same infrastructure