ParaGraph: An application-simulator interface and toolkit for hardware-software co-design

Mikhail Isaev[1], Nic McDonald[2],
Jeff Young[1], Rich Vuduc[1]

[1]Georgia Institute of Technology, [2]Nvidia

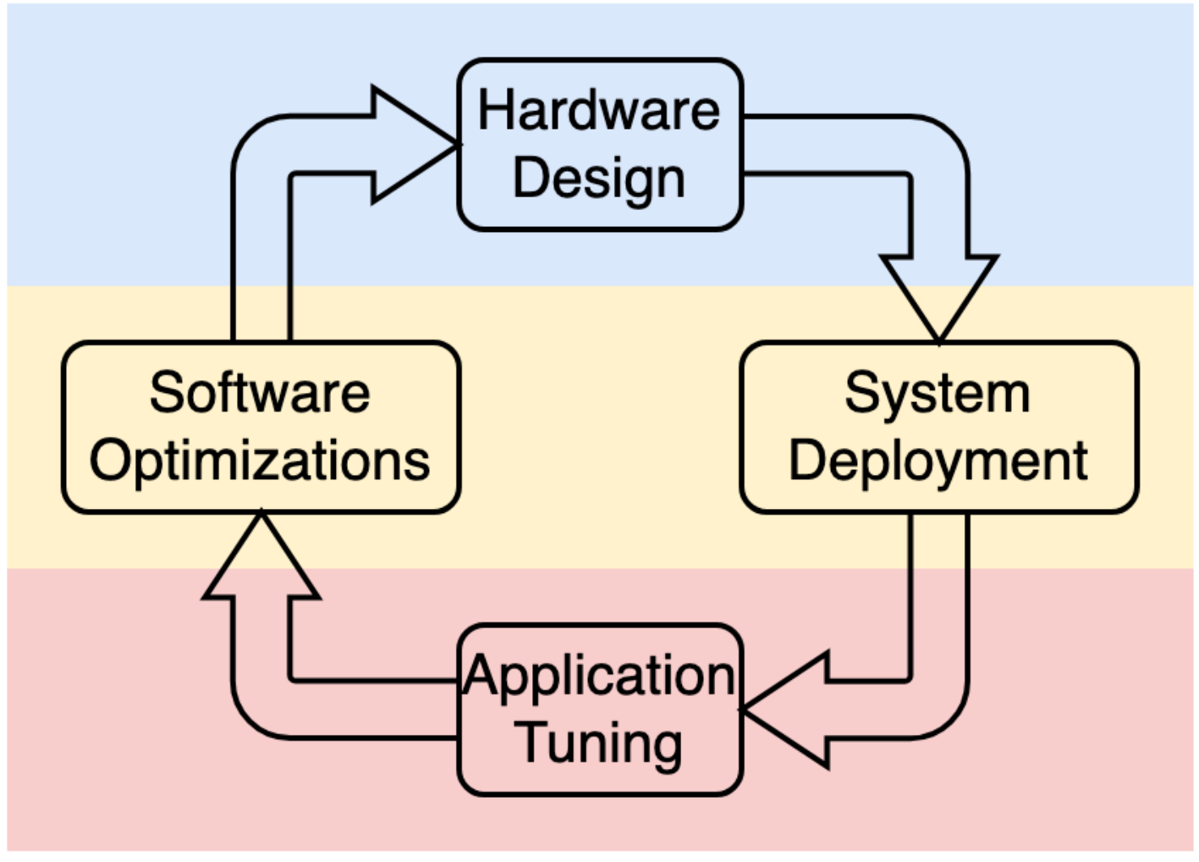
Research problem

System co-design crosses many domain boundaries

Most HW/SW co-design studies fall under one of the categories:

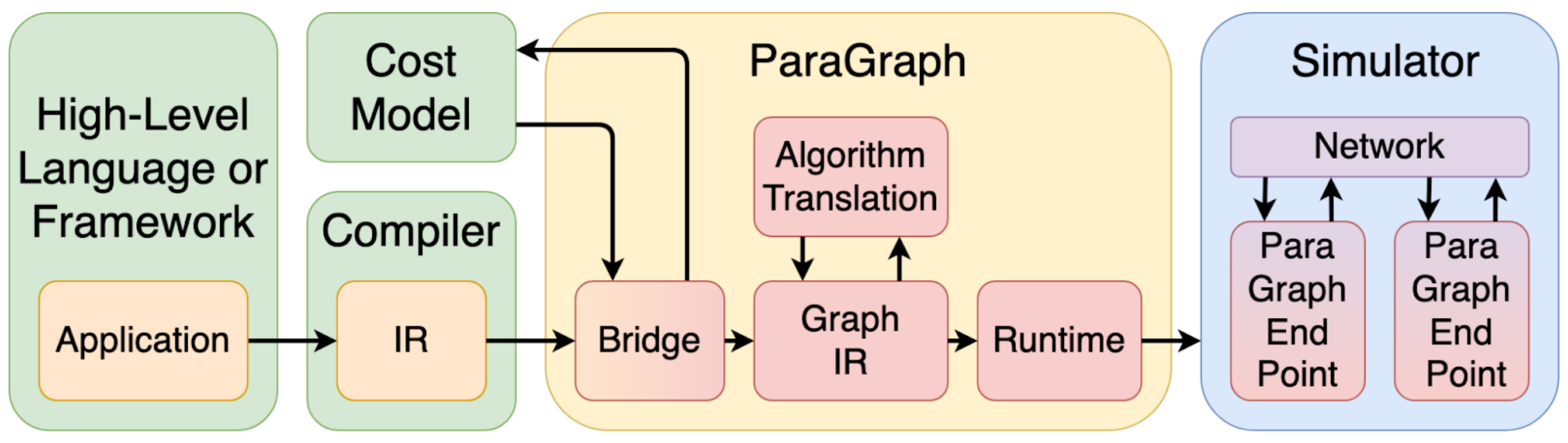
- Optimizing **software** on **existing hardware**;
- Designing **future hardware** systems with limited and **simplistic application** models.

Lack of infrastructure to model both future **hardware AND applications** with appropriate fidelity



System development cycle crosses **hardware**, **software**, and **application** domains

ParaGraph - our take on HW/SW co-design toolchain



ParaGraph goal – a real software model for hardware people and future hardware model for software people

ParaGraph for simulators is what LLVM is for real hardware

ParaGraph Workflow

(a) Pseudocode

```
import tensorflow as tf
input, target = load_data(64)
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(1024,)),
  tf.keras.layers.Dense(10)
])
loss_fn = tf.keras.losses.MeanAbsoluteError()
model.compile(loss=loss_fn)
model.fit(input, target)
```

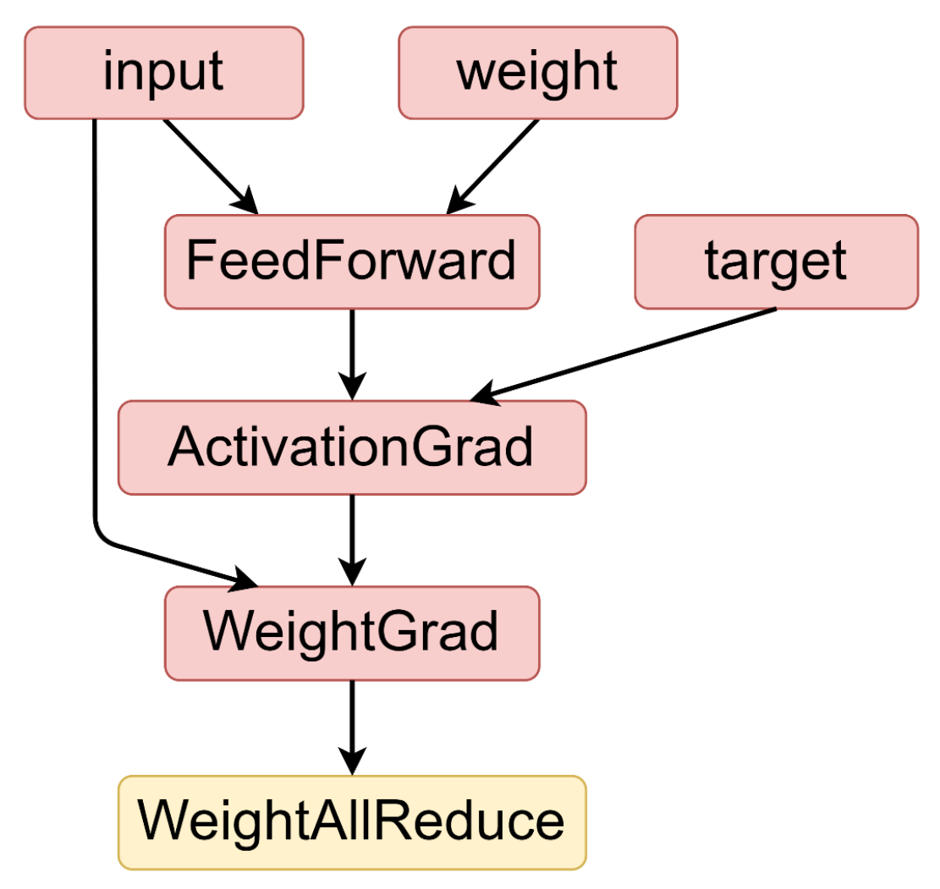
(b) HLO IR

```
%input.1 = f32[64,1024]{0,1} parameter(0)
%weight.2 = f32[1024,10]{1,0} parameter(1)
%target.3 = f32[64,10]{1,0} parameter(2)
%FeedForward.4 = f32[64,10]{1,0} dot(f32[64,1024]{0,1} %input.1, f32[1024,10]{1,0} %weight.2),
        lhs_contracting_dims={0}, rhs_contracting_dims={1}
%ActivationGrad.5 = f32[64,10]{1,0} add(f32[64,10]{1,0} %FeedForward.4, f32[64,10]{1,0} %target.3)
%WeightGrad.6 = f32[1024,10]{1,0} dot(f32[64,1024]{0,1} %input.1, f32[64,10]{1,0} %ActivationGrad.5),
        lhs_contracting_dims={0}, rhs_contracting_dims={0}
%WeightAllReduce.7 = f32[1024,10]{1,0} all-reduce(f32[1024,10]{1,0} %WeightGrad.6),
        replica_groups={{0,1,2}}
```

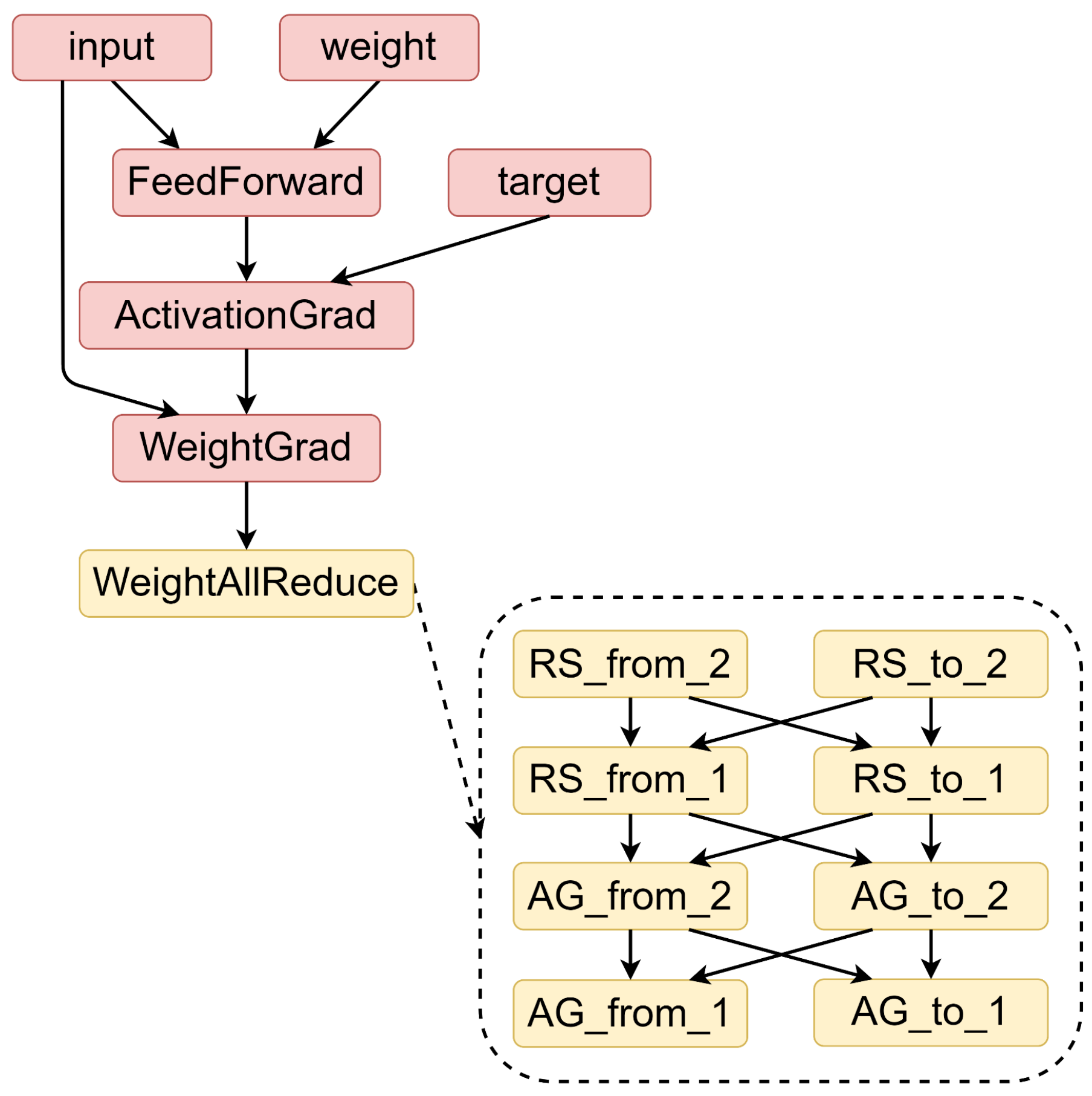
(c) ParaGraph IR

```
%input.1 = delay(), bytes_out=262144, seconds=0.250
%weight.2 = delay(), bytes_out=40960, seconds=0.039
%target.3 = delay(), bytes_out=2560, seconds=0.002
%FeedForward.4 = delay(%input.1, %weight.2), bytes_in=303104,
        bytes_out=2560, flops=1310720, seconds=1.25
%ActivationGrad.5 = delay(%FeedForward.4, %target.3), bytes_in=5120,
        bytes_out=2560, flops=640, seconds=0.007
%WeightGrad.6 = delay(%input.1, %ActivationGrad.5), bytes_in=264704,
        bytes_out=40960, flops=655360, seconds=0.625
%WeightAllReduce.7 = all-reduce(%WeightGrad.6), bytes_out=40960
        communication_groups={{0,1,2}}
```

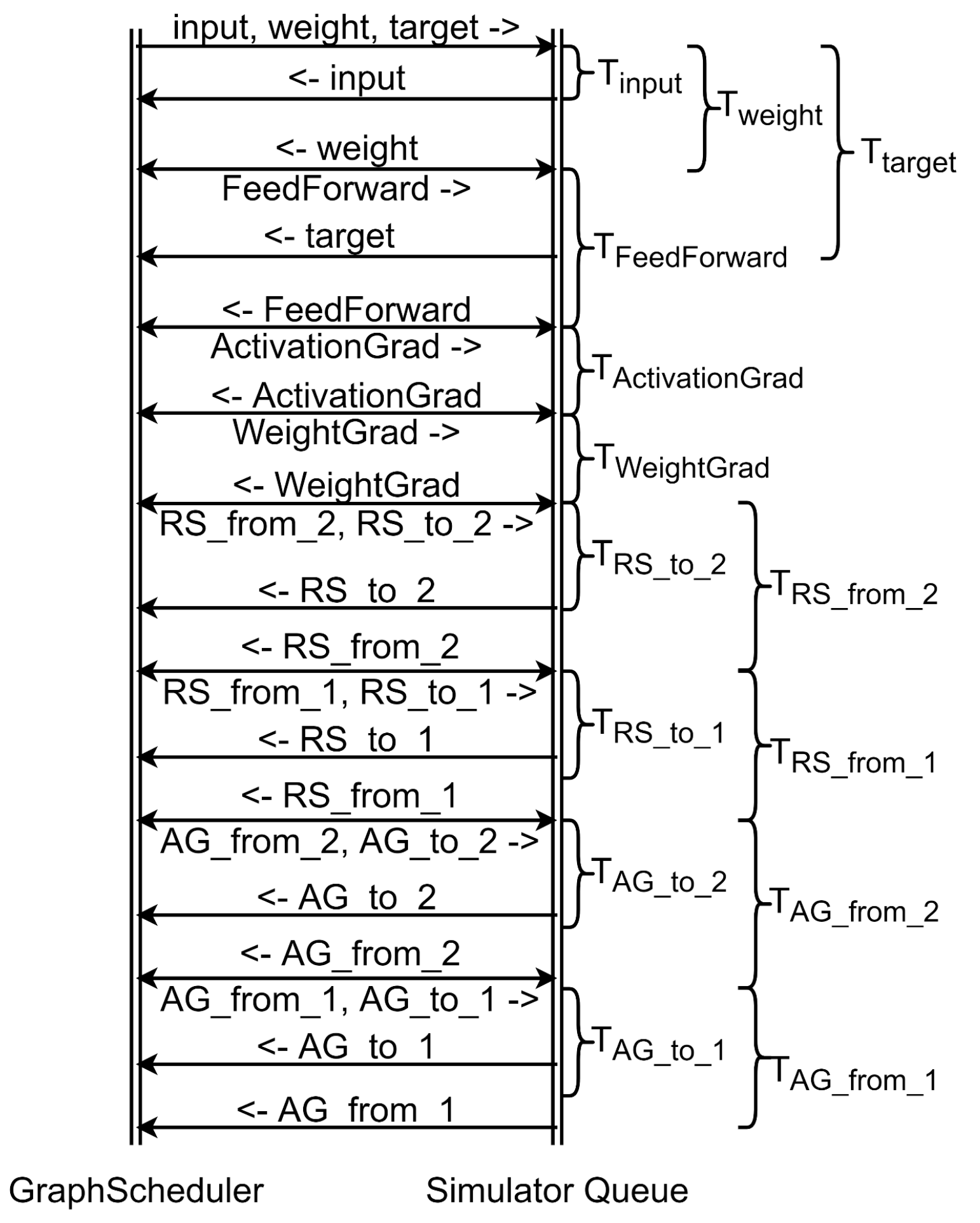
(d) ParaGraph/HLO graph



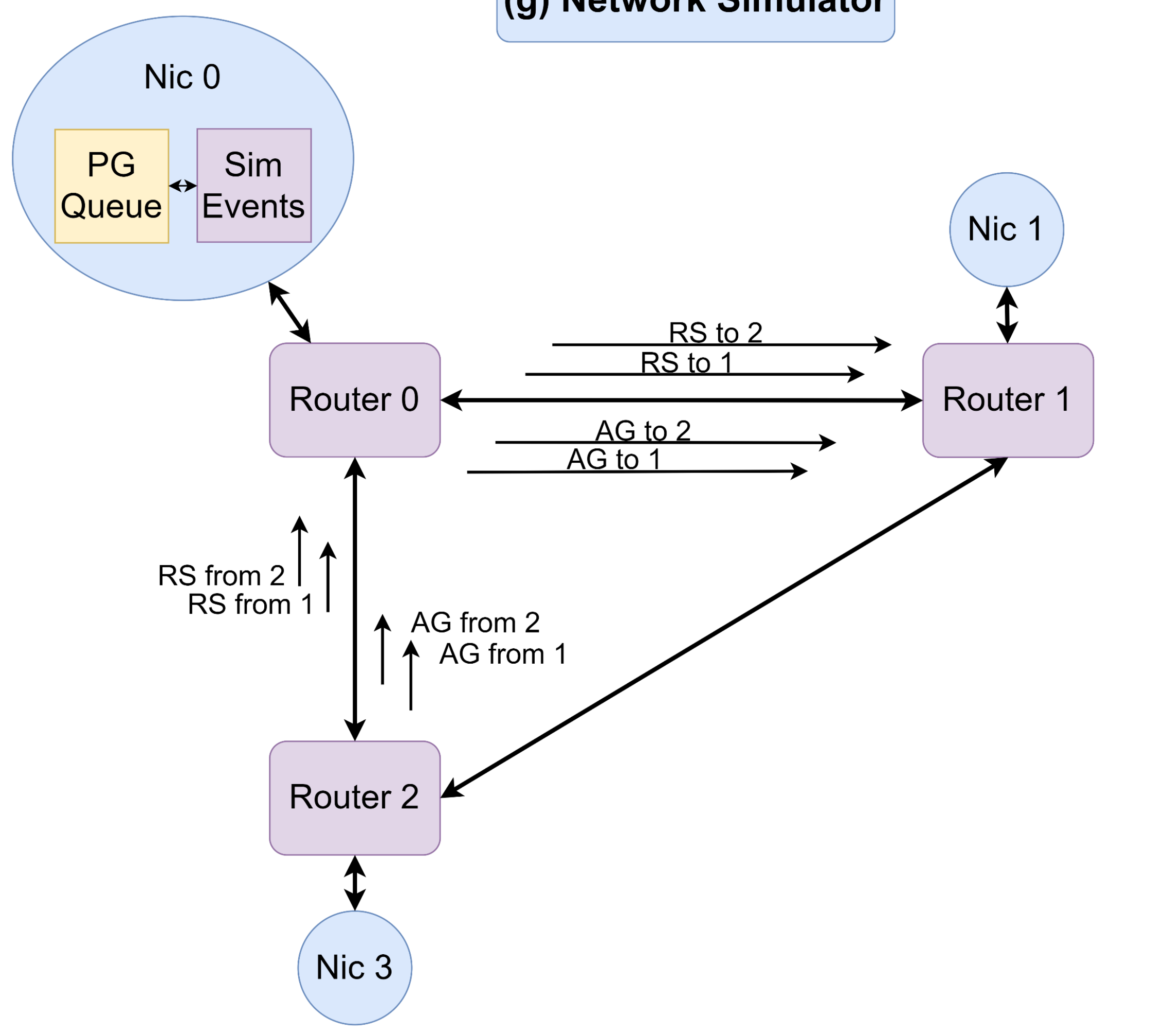
(e) ParaGraph graph after translation



(f) ParaGraph API



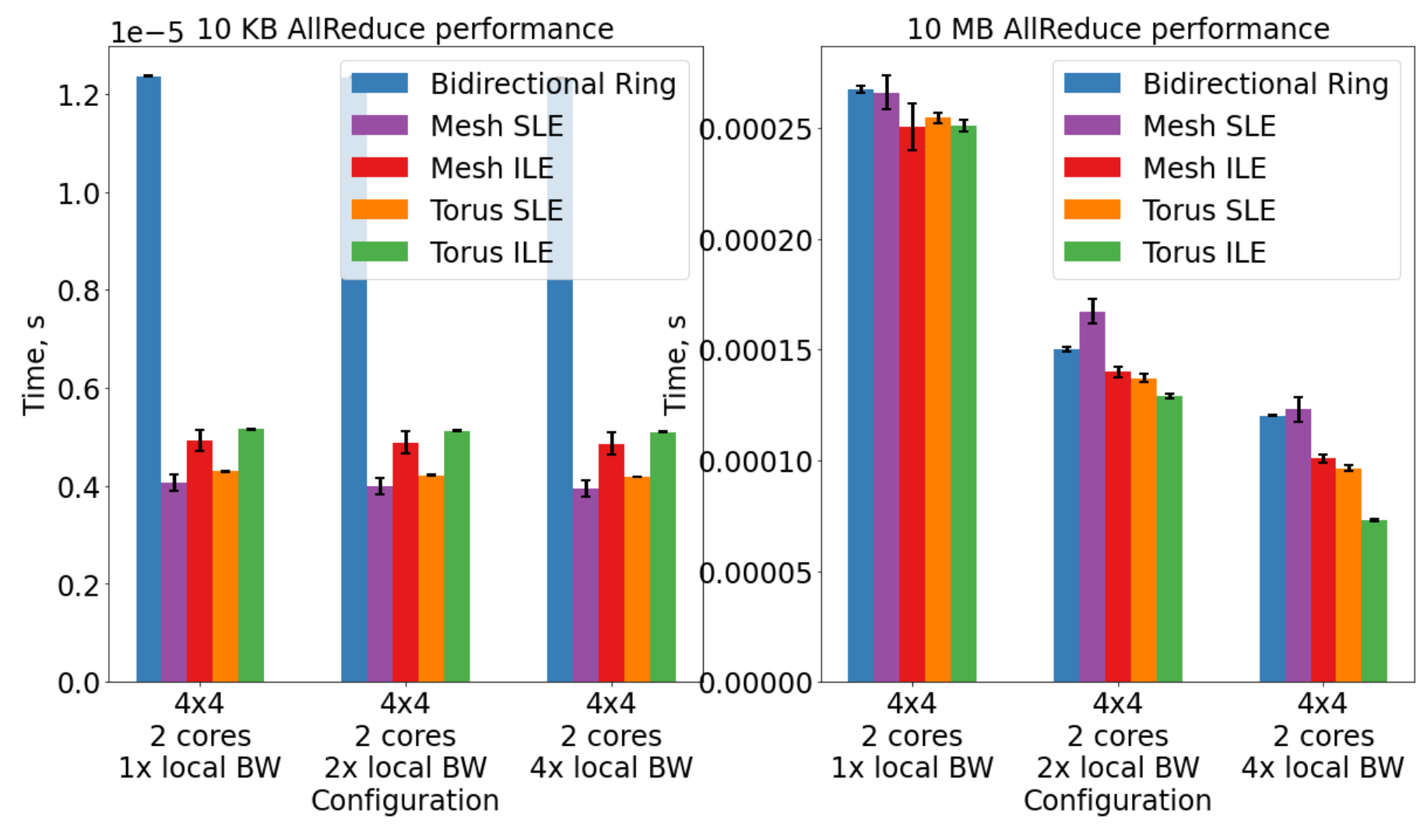
(g) Network Simulator

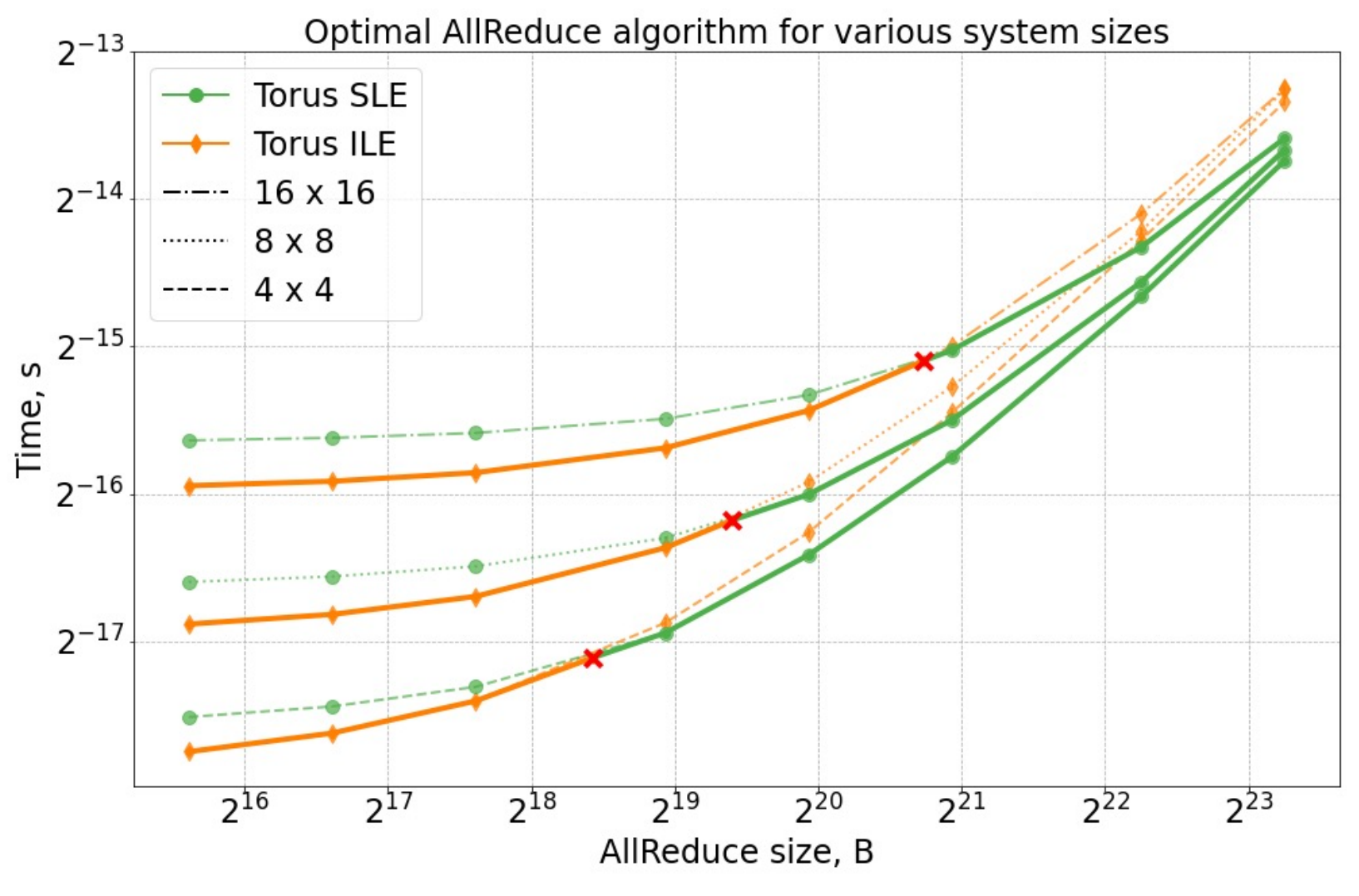


Case study 1: co-design on unified HW/SW search space

ParaGraph allows SW engineers to model system-level SW, such as communication libraries, before system deployment

With ParaGraph we navigate landscape of SW and HW parameters simultaneously





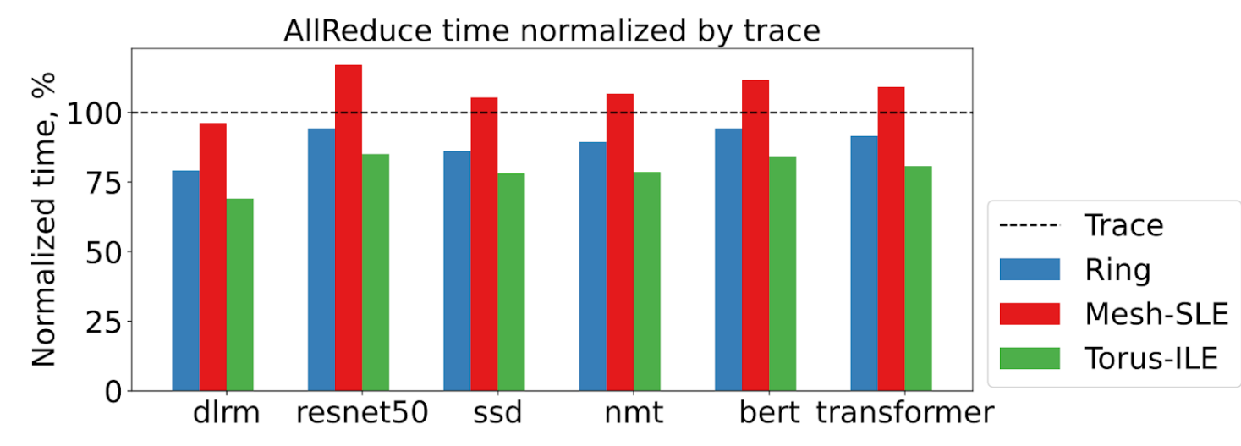
Case study 2: model accuracy analysis

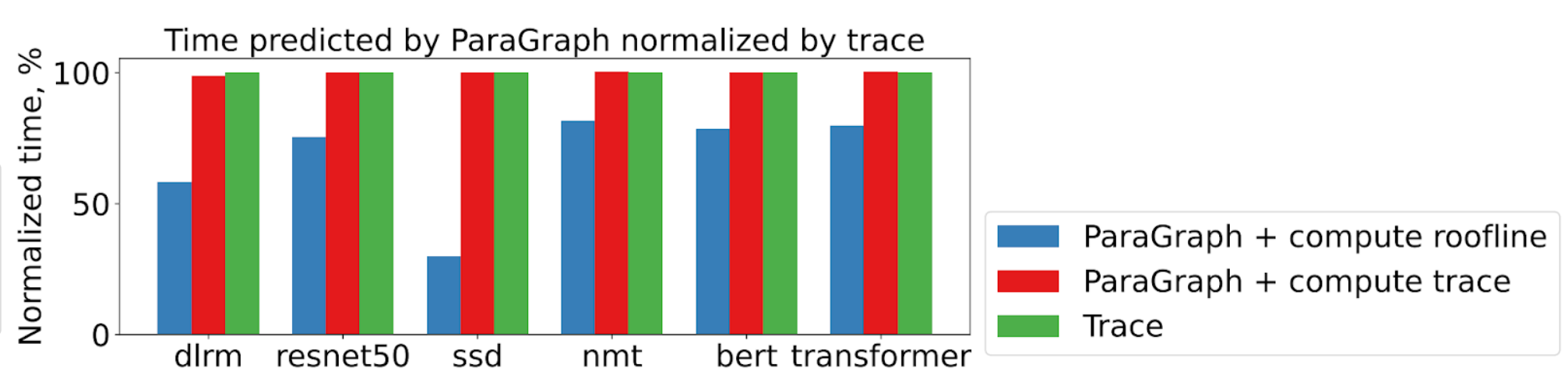
ParaGraph is validated against MLPerf training trace from 64-cores TPU v3 system

ParaGraph and TPU run the same HLO code

Network DES model matches the trace

Compute roofline allows modeling longer application run

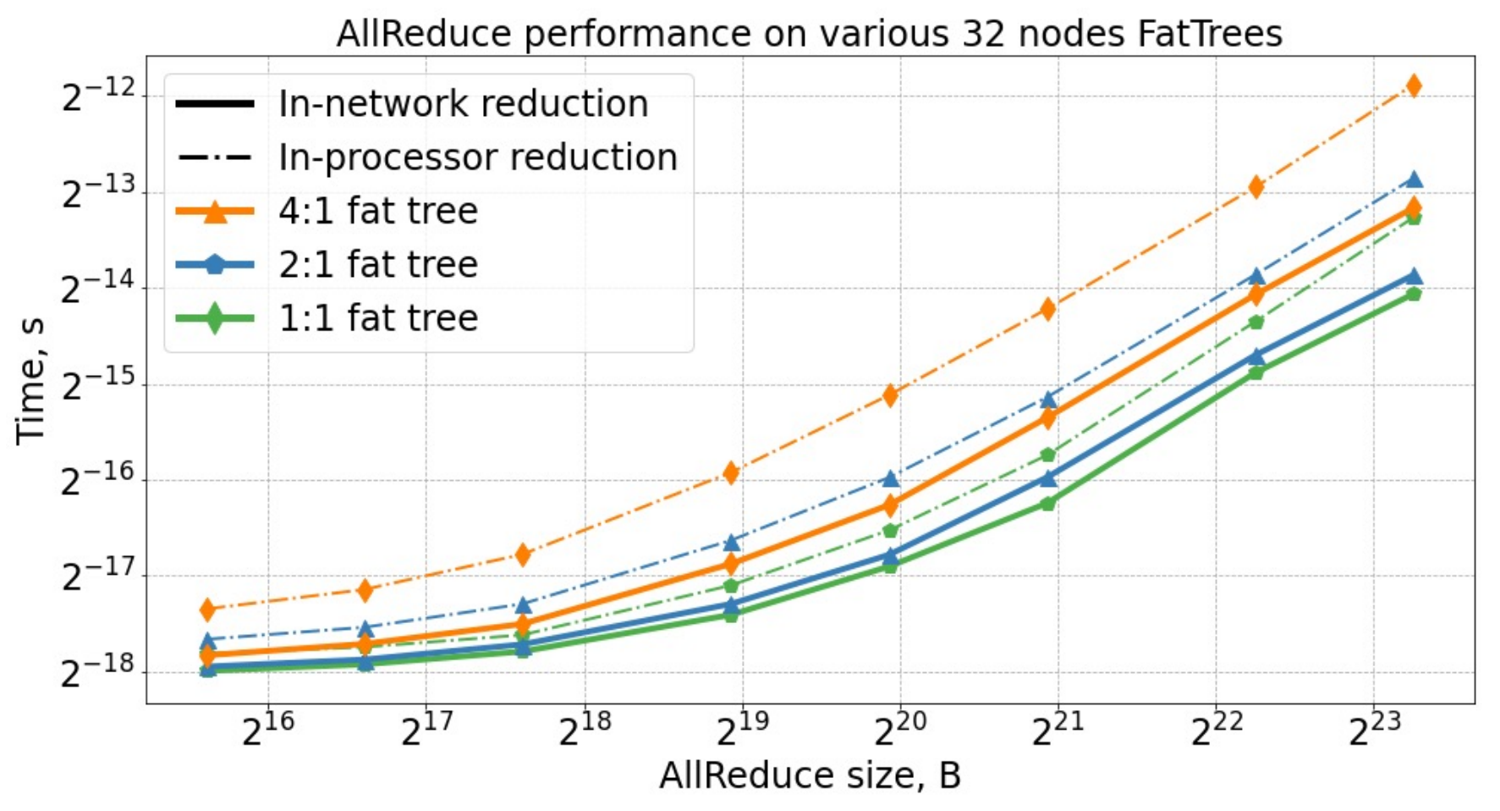


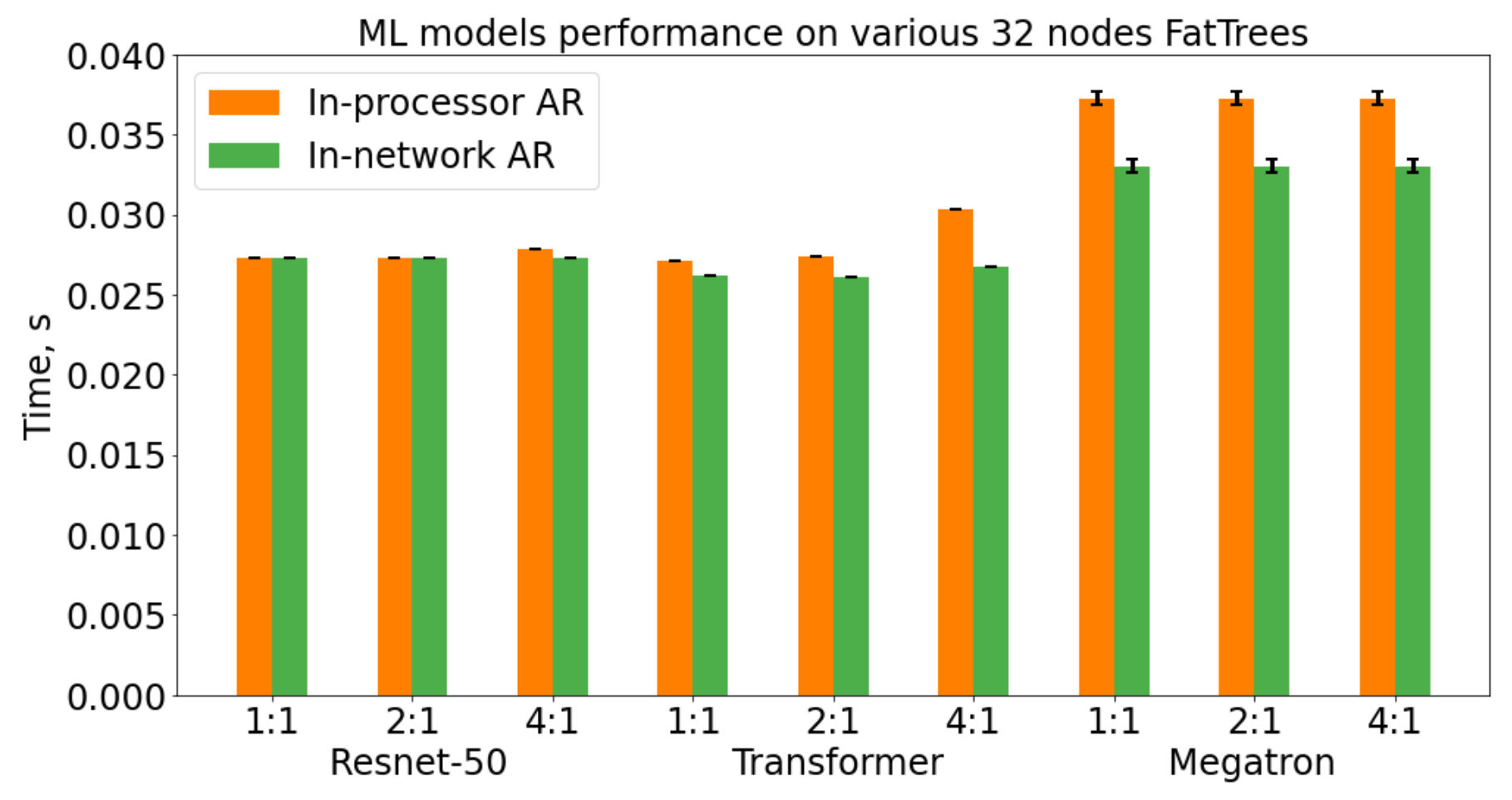


Case study 3: in-network reduction analysis

ParaGraph provides actual applications to future hardware engineers

ParaGraph helps assessing actual performance benefits of future hardware, and discovering potential performance bottlenecks, for example, performance with fat tree tapering





Conclusion

ParaGraph is a versatile co-design tool that effectively decouples application from hardware modeling, mutually beneficial to hardware and software engineers

ParaGraph supports various modeling workflows spanning several frontends (TF, JAX), backends (SuperSim, n10), and approaches (trace-driven, execution-driven, motif)

Find us on https://github.com/paragraph-sim

Acknowledgement

Portions of this work were performed during internships at Google and Nvidia.