SwitchML
Scaling Distributed Machine Learning with In-Network Aggregation

Amedeo Sapio, Marco Canini, Chen-Yu Ho, Jacob Nelson, Panos Kalnis, Changhoon Kim, Arvind Krishnamurthy, Masoud Moshref, Dan Ports, Peter Richtarik
Innovation fueled by leaps in (costly) infrastructure:

**Clusters with hundreds of machines,**

*each with many HW accelerators (GPUs)*

Compute requirements **doubling every 3 months!**

Training models is still **very time-consuming**: days or even weeks!
Scaling Machine Learning

Make **efficient** use of combined resources at multiple worker nodes

Can the network be the ML accelerator?
Distributed ML (data-parallel)
Distributed ML (data-parallel)
Distributed ML (data-parallel)
Distributed ML (data-parallel)

~100 ms Worker 1

~100 ms Worker 2

100s of MBs to GBs in each iteration
→ ~800-1000ms
Distributed ML (data-parallel)

Worker 1

<Diagram>

Worker 2

<Diagram>

Problem:

Very intensive communication in all-to-all fashion! Network increasingly the bottleneck to training speed
Parameter Server (PS) and All-Reduce (ring)
A closer look at model synchronization

Worker 1
Worker 2
Worker 3
Worker 4

Distributed ML scales poorly due to communication costs

Switch

If only I could help...
Programmable data plane

100 Gbps line rate processing

6.5 Tbps
The network is the ML accelerator

Worker 1

Worker 2

Worker 3

Worker 4

Aggregate model updates in-network
SwitchML: Co-design ML and networking

**Challenges**
- Limited computation
- Limited storage
- No floating points
- Packet loss

**Design**
- Combined switch-host architecture
- Pool-based streaming aggregation
- Quantized integer operations
- Failure-recovery protocol
- In-switch RDMA implementation

**6.5 Tbps programmable data plane**
Streaming aggregation

Worker 1

Worker 2

\[ \begin{align*}
A_1 & \quad U_1 \\
1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 & \quad 6 \\
\end{align*} \]

\[ \begin{align*}
A_2 & \quad U_2 \\
1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 & \quad 6 \\
\end{align*} \]

~100’s of MB

~100’s of KB
Streaming aggregation

Worker 1

Worker 2

~100’s of MB

~100’s of KB
Streaming aggregation

Worker 1

Worker 2

~100’s of MB

~100’s of KB
Streaming aggregation

Worker 1

Worker 2

~100’s of MB

~100’s of KB
Streaming aggregation

Worker 1

Worker 2

Pool

Switch

~100’s of MB

~100’s of KB

A₁

U₁

A₂

U₂

1 2 3 4 5 6

1 2 3 4 5 6
Streaming aggregation

Worker 1

Worker 2

~100’s of MB

~100’s of KB
Streaming aggregation

Worker 1

Worker 2

Check the paper for fault tolerance mechanism!

Pool

Switch

~100’s of KB
Combined switch-host architecture

Worker 1: Quantization & failure recovery
Worker 2: Quantization & failure recovery
Worker 3: Quantization & failure recovery
Worker 4: Quantization & failure recovery

Switch: Fixed-point aggregation
Combined switch-host architecture

Block quantization

Scaling factors aggregation
Scaling factors aggregation
Scaling factors aggregation
Scaling factors aggregation

Fixed-point aggregation
Fixed-point aggregation
Fixed-point aggregation
Fixed-point aggregation

RTT

Worker 1  Worker 2  Worker 3  Worker 4
Quantization allows training to similar accuracy in a similar number of iterations as an unquantized network.
How large a packet can a switch process?

256B of payload per packet $\rightarrow$ 75.7% network efficiency
How large a packet can a switch process?

1024B of payload per packet → 92.6% network efficiency
A glimpse of the evaluation

Check the paper for an extensive evaluation!
Implementation and evaluation

• Switch program written for Intel Tofino
• End-host C++ library providing a familiar all-reduce API
• Integrated with ML frameworks
  - PyTorch
  - TensorFlow
  - HOROVOD
• Standard ML benchmarks
• Microbenchmarks for aggregation performance
How much faster is SwitchML?

SwitchML provides a speedup in training throughput up to 2.27x on 100Gbps networks. Speedup is higher with faster GPUs that reduce the computation/communication ratio.
How does SwitchML scale with the number of workers?

SwitchML performance does not depend on the number of workers.
Summary

- Use **in-network aggregation** to synchronize model parameters updates
  - Reduce network traffic volume and latency

- SwitchML speeds up training up to 2.27x with real-world DNN benchmarks

- Aggregation time does not depend on the number of workers

github.com/p4lang/p4app-switchML