In-Network Aggregation for Shared Machine Learning Clusters

Abstraction

- Two main component of PANAMA:
    - in-network hardware accelerator
    - lightweight load-balancing and congestion control protocol
- Method to evaluate the system
    - FPGA-based prototyped
    - large scale simulation
- Outperformed result
    - decrease the training time
    - provides benefit to non-aggregation flow (because the resources occupied by aggregation flow are released)

Introduction

- Motivation for in-network aggregation
    - Cause the current aggregation takes places in the end-host, rather inside the network switch. This design gives a great pressure to the endhost, while also increase the times to exchange the gradient if use the Ring-AllReduce. If the gradients can be aggregated inside the switch, the communication bottleneck of data-parallel might be solved.
    - The limitation of the current work.
        - focus on small number of switches
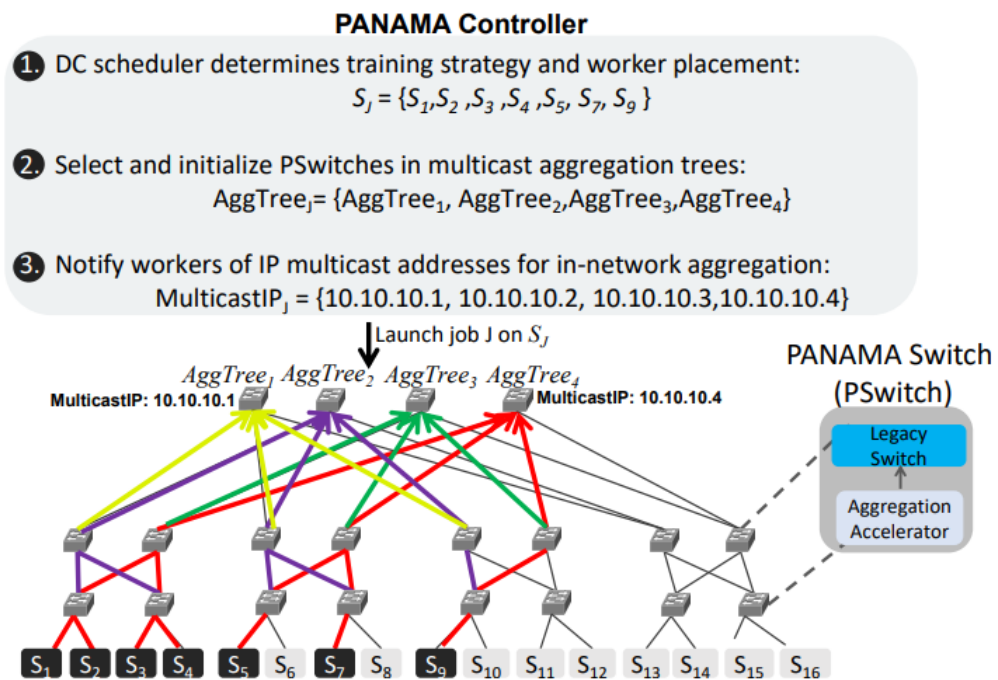   - focus on single ML jobs (not large scale DNN)

        However, with the increasing variety of Neural network, the cost and resource waste are increased. Many jobs are shifted to use the shared clusters, because the shared clusters can reduce the cost and resource waste. (To be honest, I don't understand the logic of this statement. Maybe I will read more related background regarding this)
        - The practical in-network aggregation method is questionable to manage the network resources at large scale

Overview of PANAMA

- An in-network aggregation framework for shared environment with a heterogenous set of workloads.
- Two components:
    - An aggregation accelerator which support multiple active training jobs concurrently.
        - Interesting point: the in-network accelerator is decoupled from the switch.
    - A load balancing mechanism in conjunction with a light weight congestion control protocol.
        - **the existing congestion control in data center assume a point to point connection between servers, rather than a tree based.**

- Division of the aggregation flow and non-aggregation flow.
    - The gradient exchange/transfer flow of the data parallel is called the aggregation flow. The others are called non-aggregation flow.
    - Why do this paper distinguish this two kinds of flows?
        - First, the in-network gradient aggregation's object is the aggregation flow. Therefore, it is very likely that it will bring different impacts on these two kinds of flows.
        - Second, if the non-aggregation flow can be recognized, these packets can avoid go through the process of aggregation, therefore decrease the time used and increase the throughput.
    - How to distinguish this two kinds of flows?
        - Since it use the PANAMA protocol instead of using ITP protocol, the headers in the PANAMA aggregation packet can contain the message of whether is the the aggregation flow or non-aggregation flow. If it is the non-aggregation flow, it can be directly sent from the in port of the PANAMA Switch to the out port of the PANAMA switch.
- Worker placement
    - The data center scheduler determines the optimal distributed training strategy and instantiates the job on a set of worker nodes. The figure below is illustrative.

PANAMA Controller

1. DC scheduler determines training strategy and worker placement:
$$S_J = \{S_1, S_2, S_3, S_4, S_5, S_7, S_9\}$$

2. Select and initialize PSwitches in multicast aggregation trees:
$$AggTree_J = \{AggTree_1, AggTree_2, AggTree_3, AggTree_4\}$$

3. Notify workers of IP multicast addresses for in-network aggregation:
$$MulticastIP_J = \{10.10.10.1, 10.10.10.2, 10.10.10.3, 10.10.10.4\}$$

Launch job J on S_J

$AggTree_1$ $AggTree_2$ $AggTree_3$ $AggTree_4$

MulticastIP: 10.10.10.1   MulticastIP: 10.10.10.4

PANAMA Switch (PSwitch)

Legacy Switch

Aggregation Accelerator

S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_{10} S_{11} S_{12} S_{13} S_{14} S_{15} S_{16}

Figure 3: High-level workflow of PANAMA.

- Network design
    - Challenge 1: load balance of the aggregation traffic
        - use Round Robin to balance the packets among the aggregation tree(avoid placing all the workers within one tree).
    - Challenge 2: fair sharing the network bandwidth between aggregation and non-aggregation flow.
        - As mentioned, use the packet header parser. Then deal with the packets differently.
        - control logic. (congestion control logic)