Net For AI

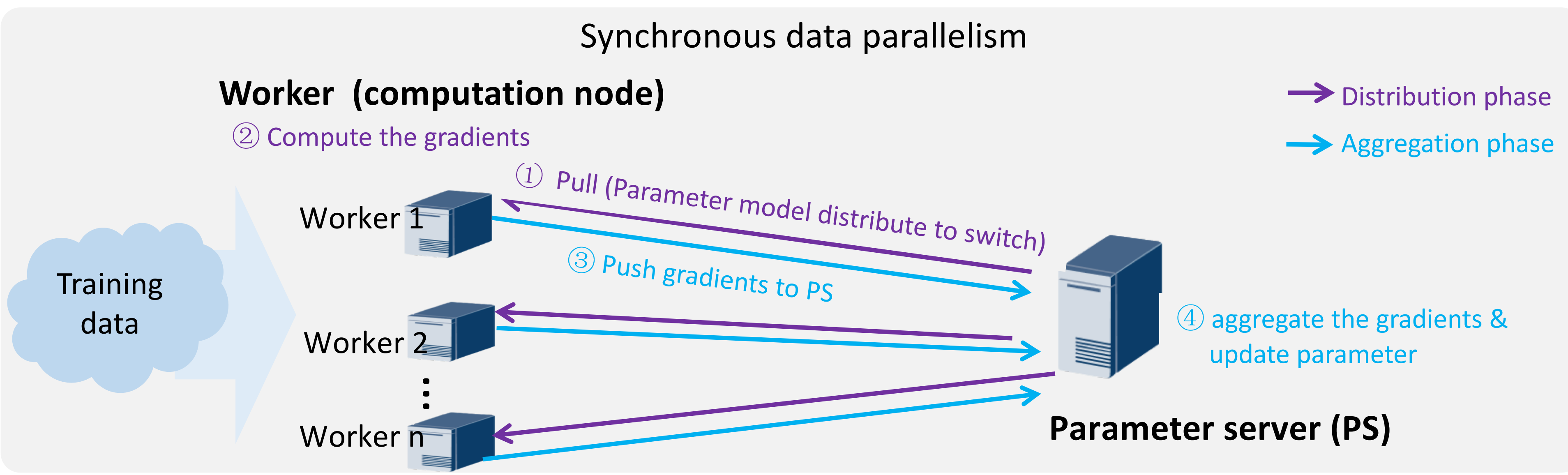
Rachel Chen, Jeffrey He, Qiaoling Wang

IETF 103, Bangkok

AI/ML in Networking: background

- **The Parameter Server (PS) architecture has been widely adopted by many ML systems.**
  - Two kinds of nodes: PS (maintains the global parameter), worker (computation node)

- **Two communication phases during each training iteration generate heavy traffic.**
  - Distribution phase: parameters are distributed from the PS to the workers (①), which then execute a local training algorithm (②).
  - Aggregation phase: each worker sends the local gradients to the PS (③). Then PS aggregates the gradients and generates a new set of parameters to feed into the distribution phase (④).



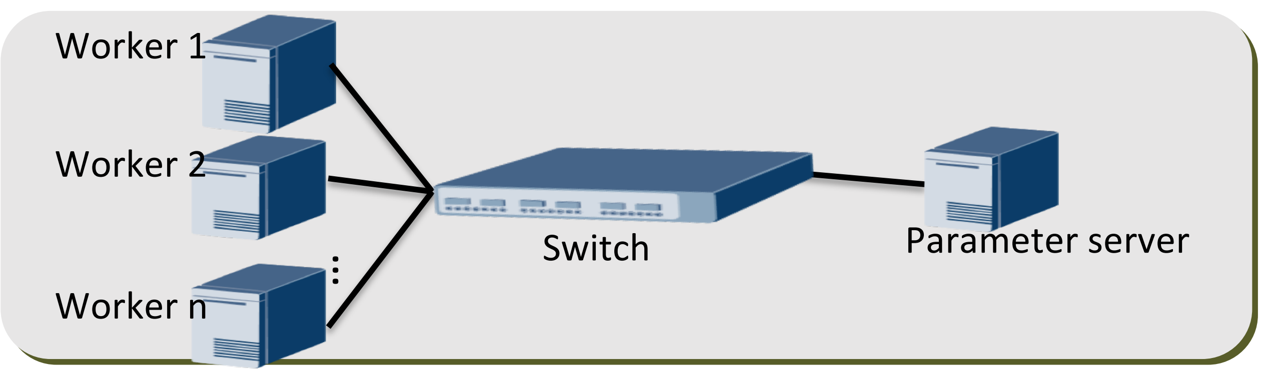
There is great potential to improve the E2E ML training performance

- The model parameters are relatively large (typically hundreds of MBs).
- Data transmission between PS & workers takes the majority of the iteration time.

  (eg. 10GE, VGG-16, transmission time is significantly longer than calculation time in a single iteration.)
- For in net aggregation, the programmable switch only needs to do sums.

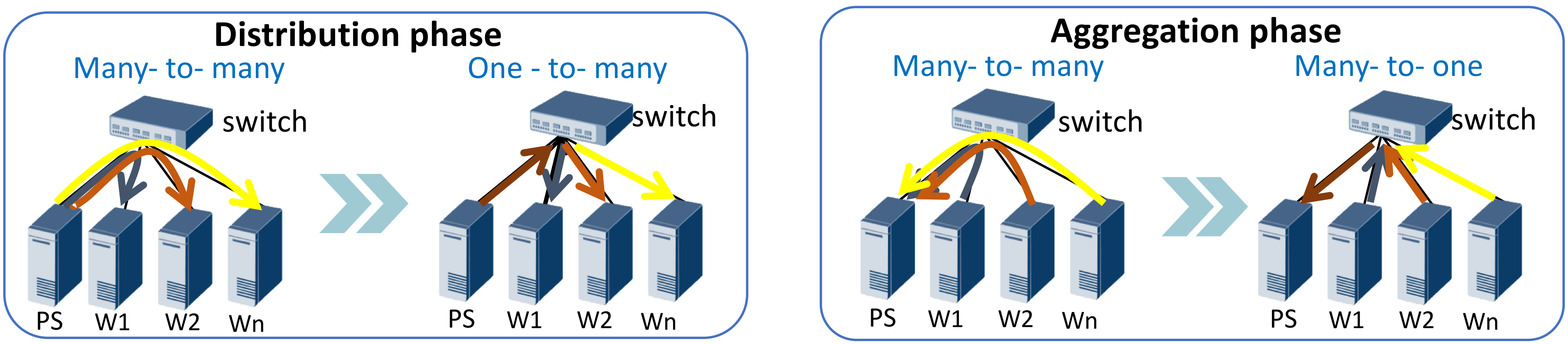
DNN model	Parameter Quantity(MB)	Calculation time (ms)	Single iteration (ms)	Theoretical Transmission (ms)	
			10 Gbps test results	10Gbps	40Gbps
Inception-v3	106	1035.7	1700	1017.6	254.4
Resnet-152	230	650.9	2781	2208	552
VGG-16	528	285.2	7114	5068.8	1267.2

Test condition: 1 PS, 6 workers, batch size=32

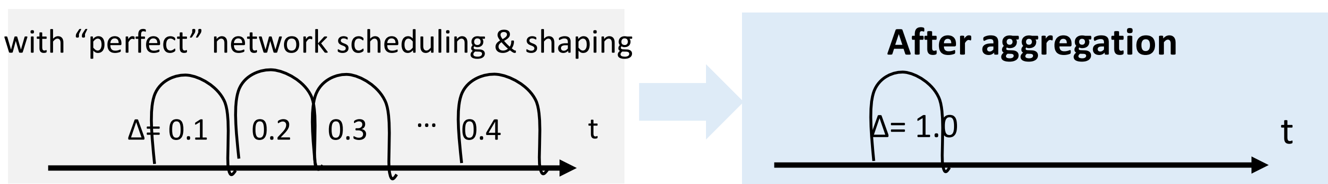


In-net computing could bring overall ML performance improvement

- **ML training can be greatly accelerated by traffic reduction (the traffic pattern changes).**



Traffic reduction: from switch to PS (aggregation)



- **Recent advances on in-net computing research show optimistic results.**
  - The average overlap is around 35% and 64.5% for SGD and Adam applications [SAPIO]. (A high overlap means that aggregating local updates of each worker in the network can significantly reduce the traffic load.)
  - By adding in-net computing and multicast, the VGG-16 training performance can be greatly enhanced [CHANG].

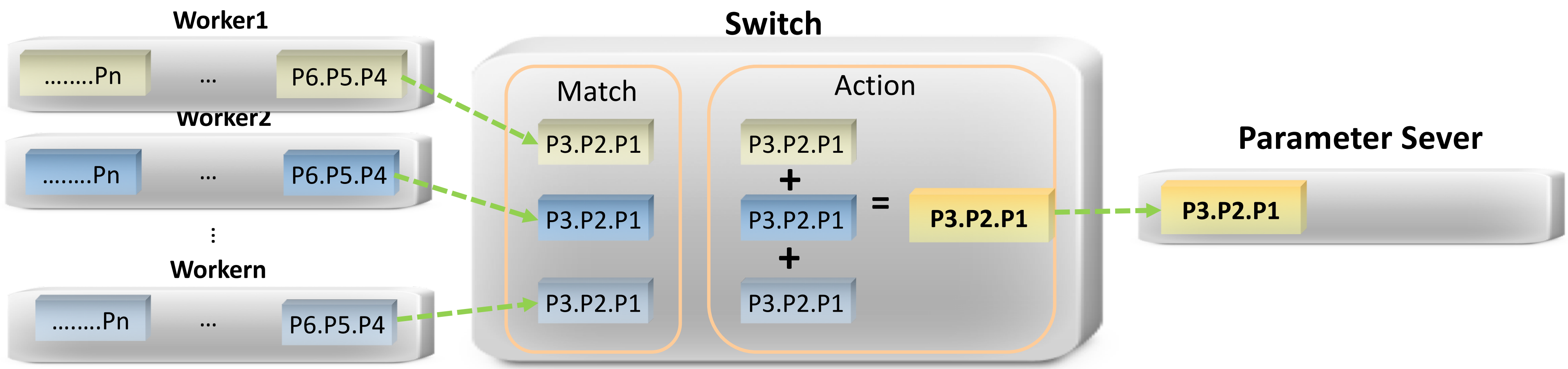
[SAPIO] Sapio et al., "In net computing is a dumb idea whose time has come", 2017.

[CHANG] Chang et al., "Network Evolution for DNN Trainings", 2018.

SGD: Stochastic Gradient Descent

Research considerations (1/3): Protocol

There are some gaps between the design goals and the current protocol and hardware limitations.



For Match-Action operation in the switch:

1. **Each packet should be of the same length,**
2. **And have an ID**

VS.

Parameter: hundreds of MB

TCP packet length can be variable

UDP packet maxi ~64K, hence the need to split each packet, to be labeled with an ID for M-A operation

① Current protocols can't easily meet the requirements of in-net computing

The depth of packet parsing: ~ hundreds of bytes

VS.

MTU of 1500 bytes

② There is a relationship between the depth of packet inspection & the performance

Research considerations (2/3): Increasing storage

- **The storage capacity of the switch may need to be increased to support ML intermediate data aggregation**
  - In the multi-machine distributed computing, the time for the backward propagation of each worker is variable. Each set of parameters may have a size of hundreds of MBs, while the buffer of current programmable switches is often limited.
  - The parameters sent from some workers may be cleared due to the FIFO principle, as a result of using opportunism.

DNN model	Parameter Quantity(MB)
Inception-v3	106
Resnet-152	230
VGG-16	528

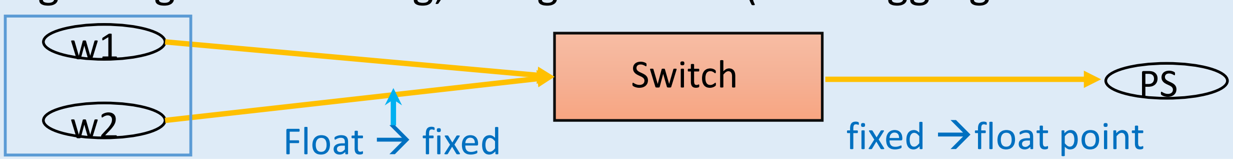
- **Open Issue: What is the impact of opportunism on the E2E performance gain? (the impact of buffer size?)**
  - Need to find out the "sweet spot" between the added complexity and the gain.

Research considerations (3/3): Fixed vs. floating point

- **Typically AI algorithms use floating-point computation. This raises the following questions:**
  - Is it necessary to introduce floating-point which current switches do not support ?
  - If changed to fixed-point calculation, will the AI training results be worse?

Our preliminary analysis show optimistic results (work in progress):

- If the value to be calculated is known, move the decimal point to do integer calculation and then convert it into floating points.
- Some existing research on fixed point based training show that the error is small and training process is not affected [Gupta] [Matthieu].
- We use floating point for training. Weight (1) is floating, and gradient (2) (to be aggregated in the switch) is fixed point.



(1) A weight in ANN is the importance of the feature (input) to the Neuron.
(2) Gradient: also called slope, describes how the network's error varies as weight is adjusted.

[Gupta] Deep Learning with Limited Numerical Precision (Fixed point) https://arxiv.org/abs/1502.02551)
[Matthieu] Training deep neural networks with low precision multiplications (Dynamic fixed point) arxiv.org/abs/1412.7024

Next steps

- An ID on COIN for ML
- Finding interested co-authors
- Updated presentation in Prague

Comments are welcome!

For further discussion/ comments

chenlijuan5@Huawei.com