PIPEDREAM

- Varun Batra

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

AGENDA

- Why PipeDream?
- Pipeline Parallelism
  - Partitioning
  - Scheduling
  - Learning
- Implementation
- Experimentation

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

THE JOURNEY SO FAR….

- Distbelief and Adam – Using Commodity Machines
- TensorFlow – Generalization and giving user the power to code
- Problem - Time and Resource consumption. Imagine billions of parameters in a word imbedding/ image processing task.

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

THE JOURNEY SO FAR….

- Solution – Parallelism! 10 points to Gryffindor!

- Naïve parallelism can be detrimental, as quality matters and also can blow up computation or communication overheads down the road.

- Time per pass can decrease, but number of passes increase! Accuracy/Convergence impacted.

- Total Time = Time per epoch * Number of epoc for a given accuracy.

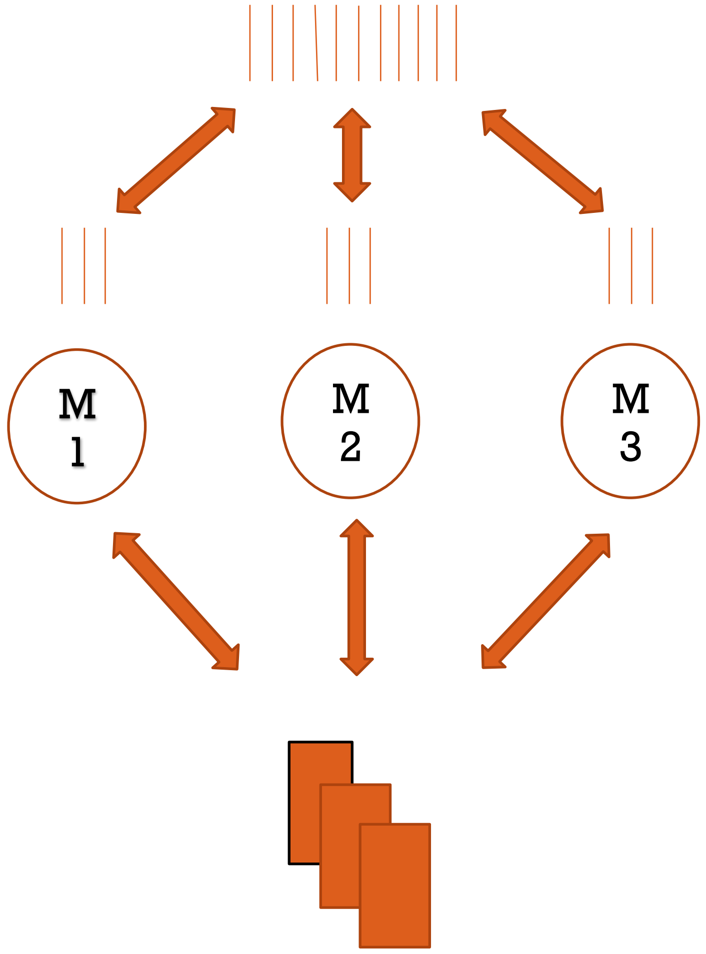
WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

WHAT IS A MINIBATCH?

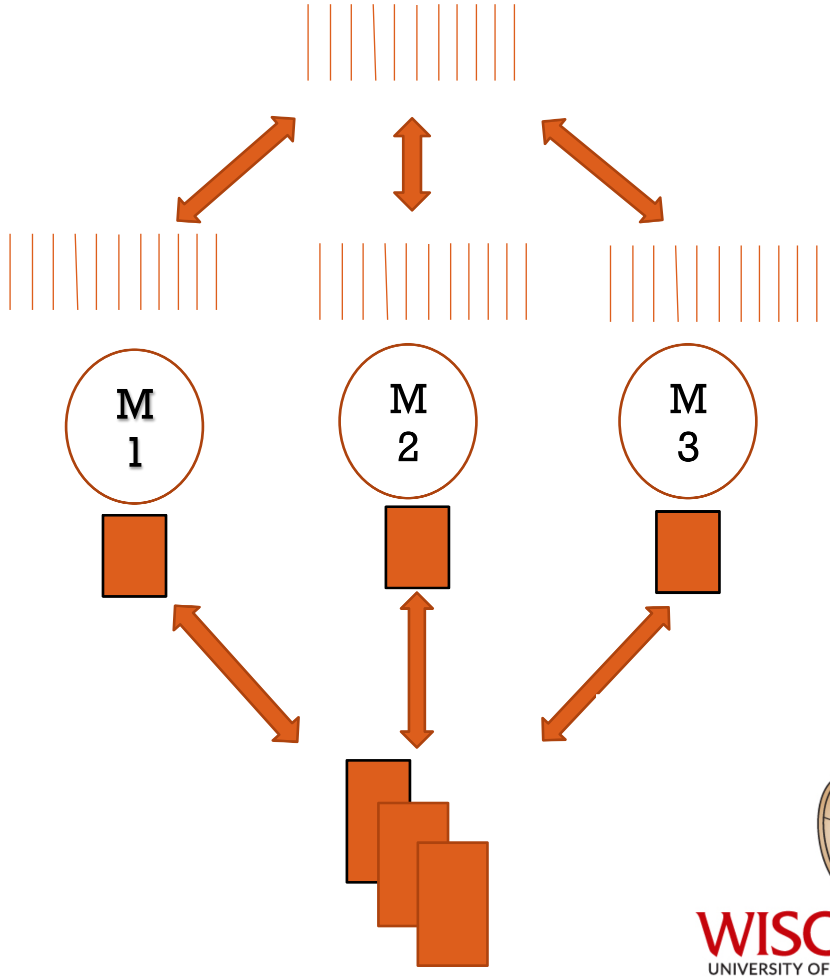
- Training contains multiple epochs over the entire data.
- In each epoch, model trains over all the inputs in the dataset using steps.
- In each step, the current model makes a prediction from a small set of training samples called minibatch. This process is called forward pass.
- Minibatch fed to layer 1, each layer computes a function using learned parameters and passes to next layer. The final output class prediction is compared to actual value and the error is propagated back in a Backward Pass to update the weights.

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

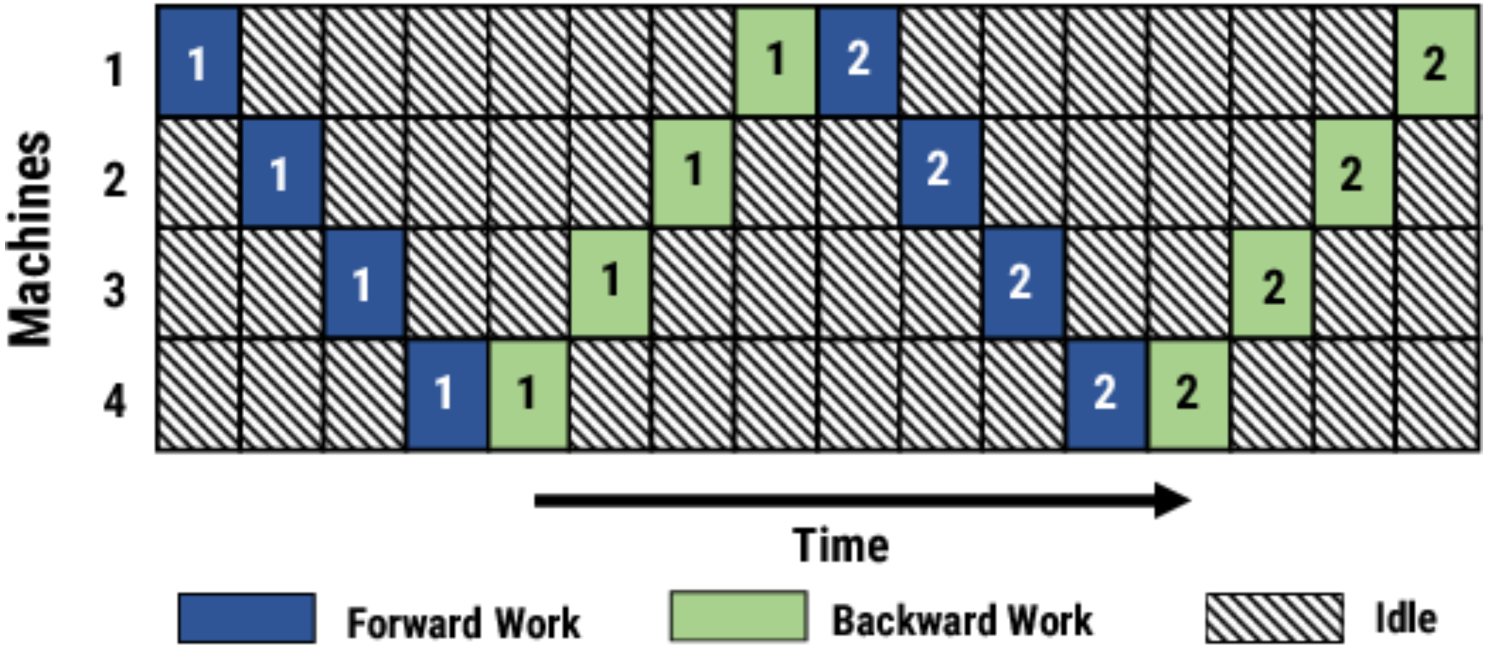
MODEL PARALLELISM



DATA PARALLELISM



MODEL PARALLELISM IMPACT

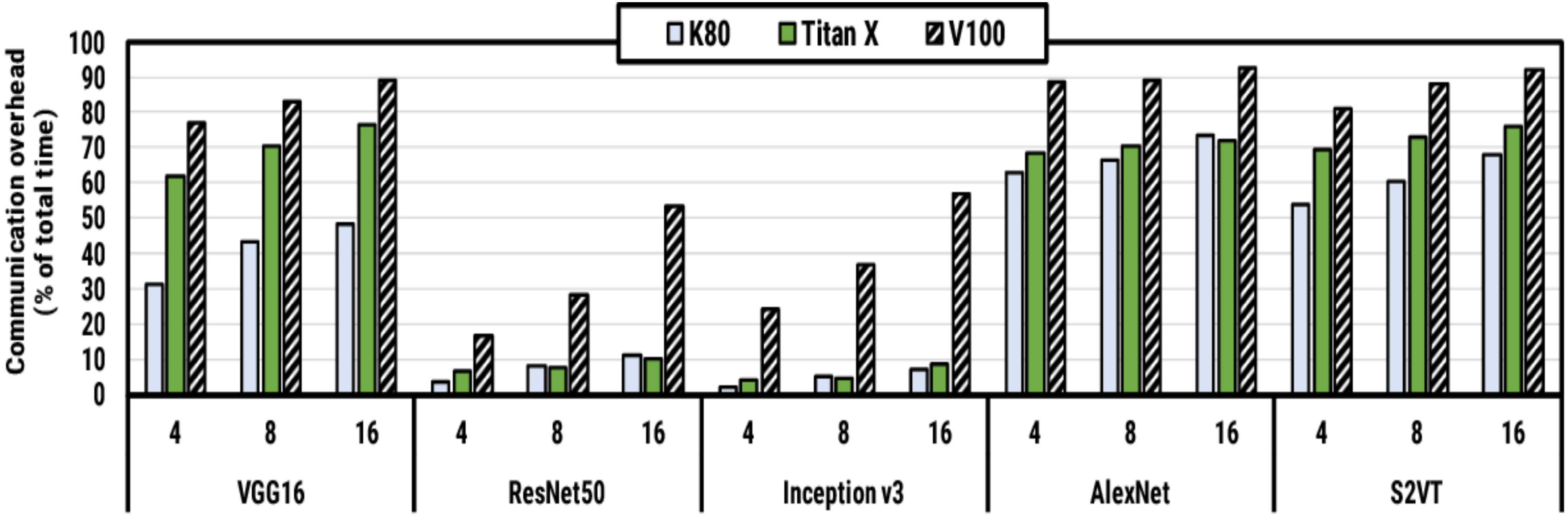


- Under-Utilization
- Unknown Model Splitting Technique

UNIVERSITY OF WISCONSIN–MADISON

DATA PARALLELISM IMPACT

As number of workers increase, the communication overhead increases.



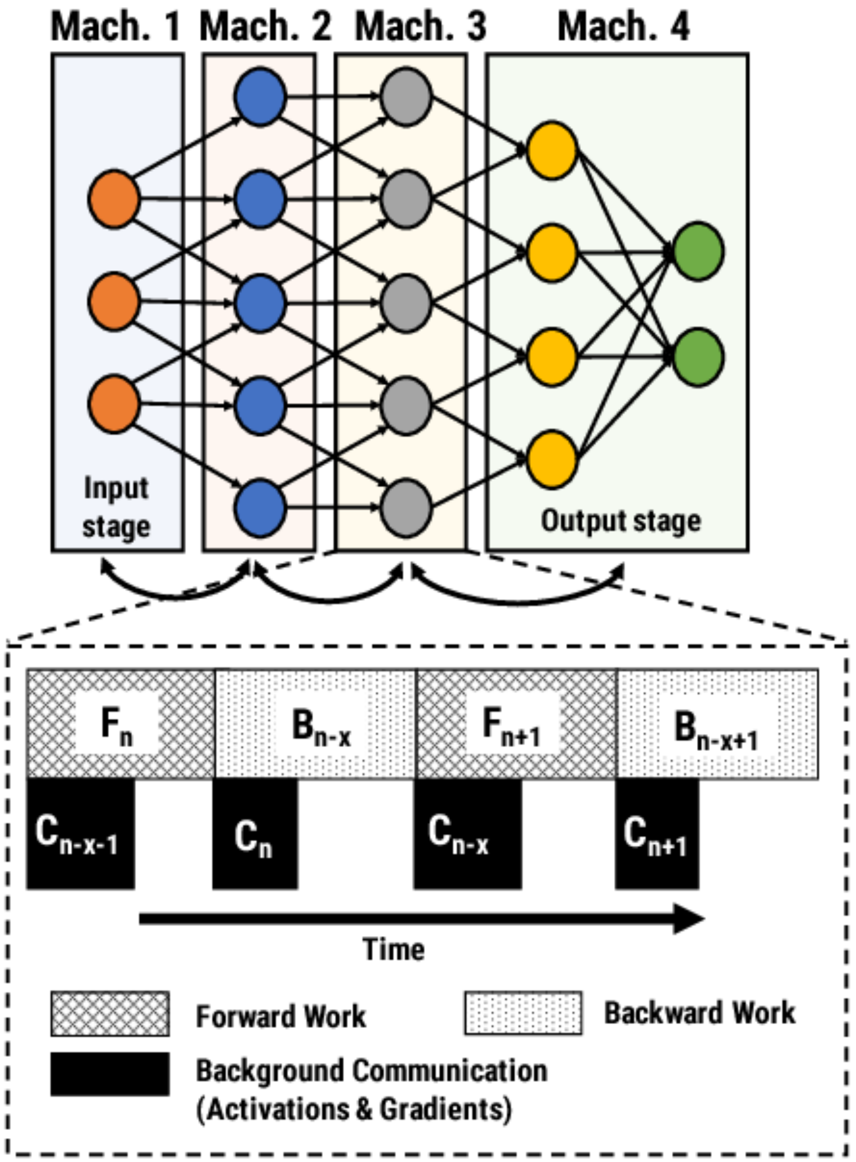
RACE BETWEEN COMMUNICATION V/S COMPUTATION

SOLUTION OF SOLUTIONS?

- PipeDream
- Pipeline Parallelism = MP + DP + Pipelining

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

PIPELINE PARALLELISM



- Entire Model broken into Stages
- Each Stage mapped to a Machine that performs both backward and forward pass
- Multiple minibatches inserted together to make use of all machines.

PIPELINE PARALLELISM

- Benefits over Data Parallelism :
  - Pipelining communicates less
    - output of layer much smaller than parameter size
  - Pipelining overlaps computation and communication
    - forward and backward pass has a lot of communication and computation overlap for subsequent minibatches, so, better hardware efficiency.

CHALLENGES HANDLED

- Automatic Partitioning
- Scheduling
- Effective Learning

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

AUTOMATIC PARTITIONING

Goals

1. Each Stage performs roughly same amount of work
2. Inter-stage data communication is minimum

AUTOMATIC PARTITIONING

- Profiling : Dry run the model on a single machine to estimate for each layer :
  - Total Forward and Backward Computation time.
  - Size of output activation and input gradients.
  - Size of parameters

AUTOMATIC PARTITIONING

- Partitioning Algorithm :
  - Computes :
    - Partitioning of layers into stages
    - Replication Factor for each stage
    - Minibatches to keep pipeline busy

  - Goal is Minimize the Overall Time in the Pipeline System
    ie. Minimizing the time for the slowest stage.

$$T(i \to j, m) = \frac{1}{m} \max\left(\sum_{l=i}^{j} T_l, \sum_{l=i}^{j} W_l^m\right)$$

- Let T(i → j, m) denote the time taken by a single stage spanning layers i through j, replicated over m machines.
- Let A(j, m) denote the time taken by the slowest stage between layers 1 and j using m machines.
- Goal – Find A(N, M), and the corresponding partitioning where N is the number of layers and M is the number of Machines.

1. $$A(j, m) = T(1 \to j, m)$$

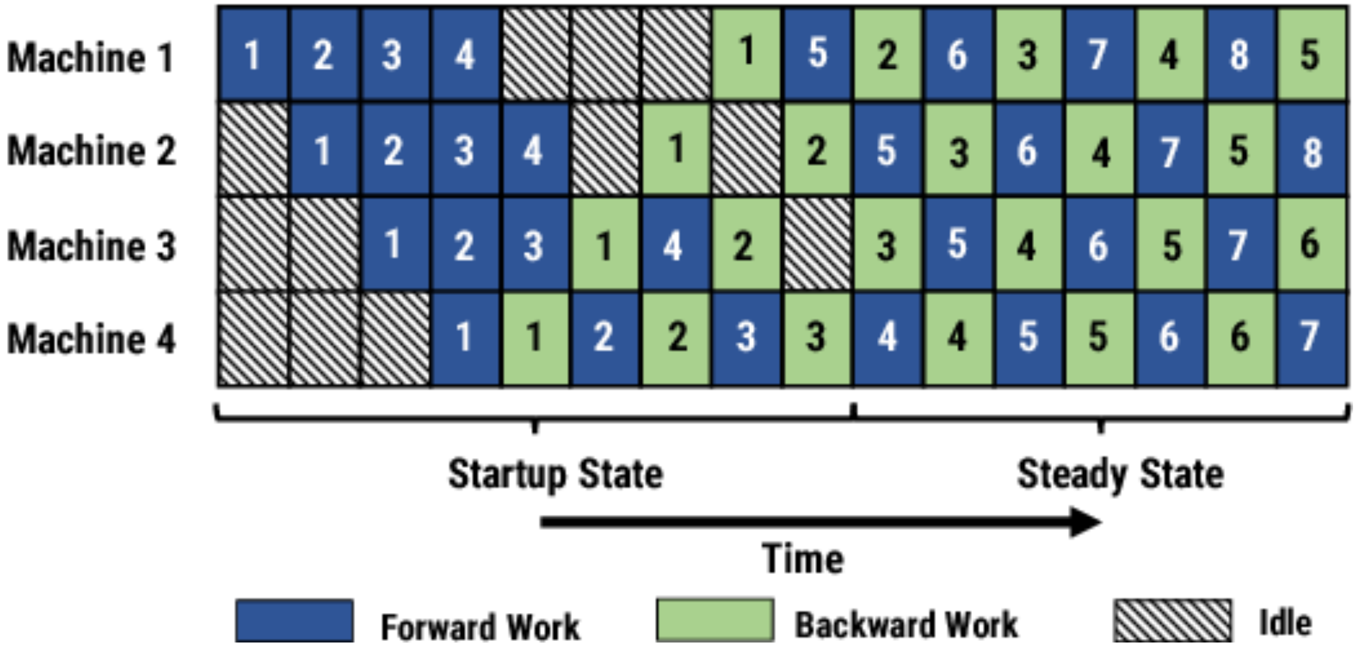
2. $$A(j, m) = \min_{1 \le i < j} \min_{1 \le m' < m} \max \begin{cases} A(i, m - m') \\ 2 \cdot C_i \\ T(i + 1 \to j, m') \end{cases}$$

Initialization. $A(1, m) := T(1 \to 1, m)$, where $T(.)$ is as defined above, and m is varied from 1 through M (the total number of machines). $A(i, 1) := T(1 \to i, 1)$, where i is varied from 1 through N (the total number of layers in the model).

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

SCHEDULING

Alternate between Forward and Backward Work – 1F1B



WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

EFFECTIVE LEARNING

- Mixing of Forward and Backward passes with different versions of parameters can lead to incorrect/slow learning.

- Weight Stashing – Maintaining multiple versions of weight for Forward and Backward pass in a stage. In Forward – Use latest version, in Backward – use the corresponding version

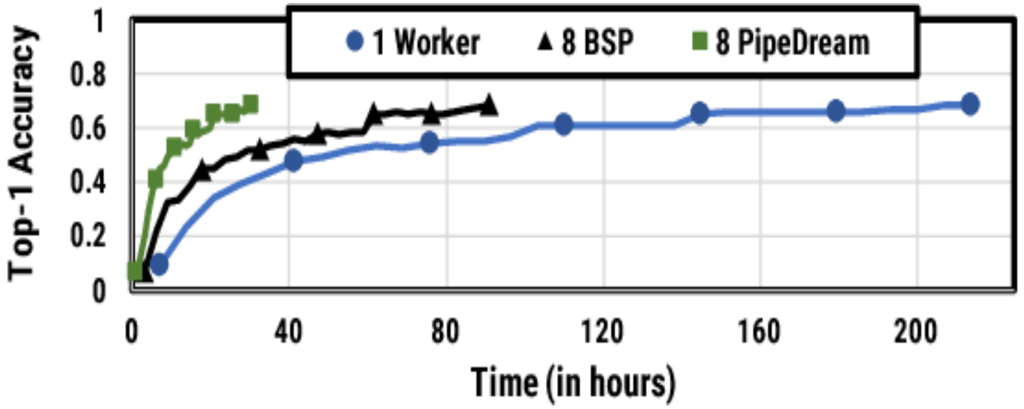
- Vertical Sync – After performing the backward pass of a minibatch using an older version, each stage applies latest updates to use new weights.

IMPLEMENTATION

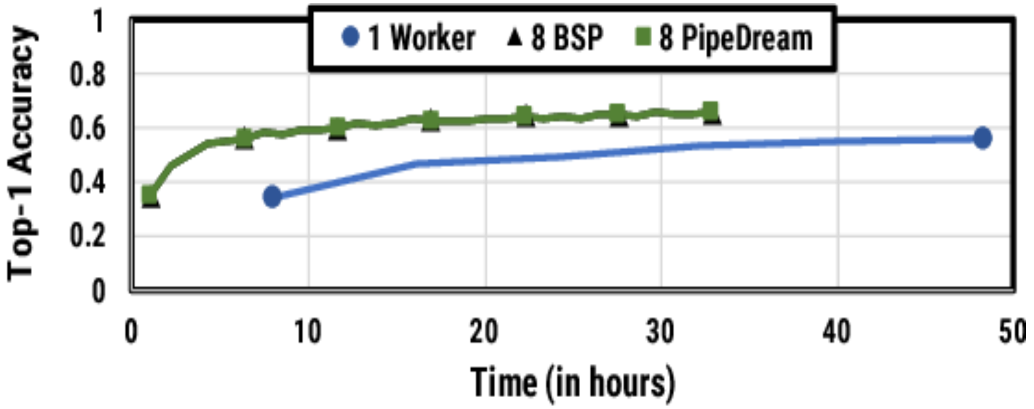
- Initialization Step
- Parameter State
- Intermediate State
- Checkpointing

EXPERIMENTATION

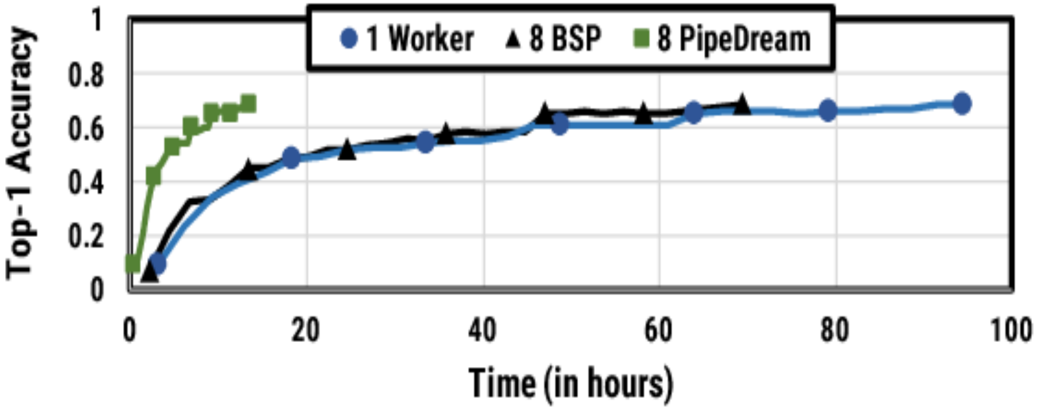
- Cluster A – Fast Network, Slow GPU
- Cluster B – Fast GPU, Slow Network

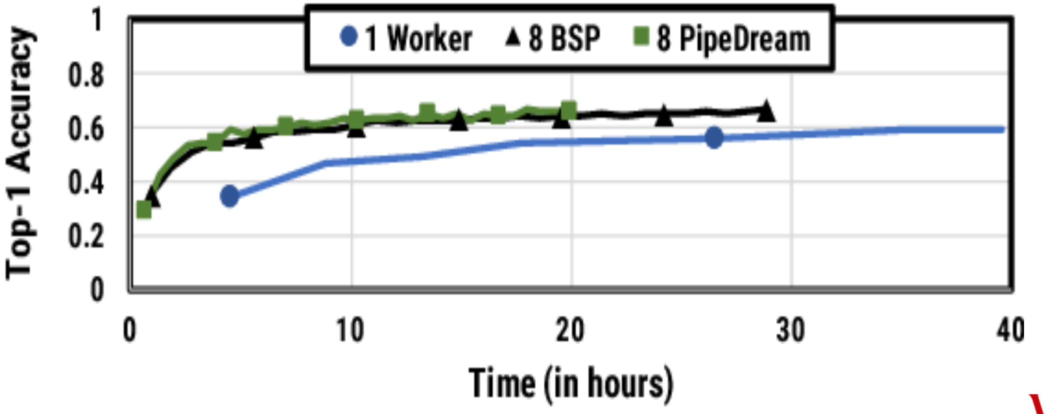


(a) VGG16



(b) Inception-v3





WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

EXPERIMENTATION

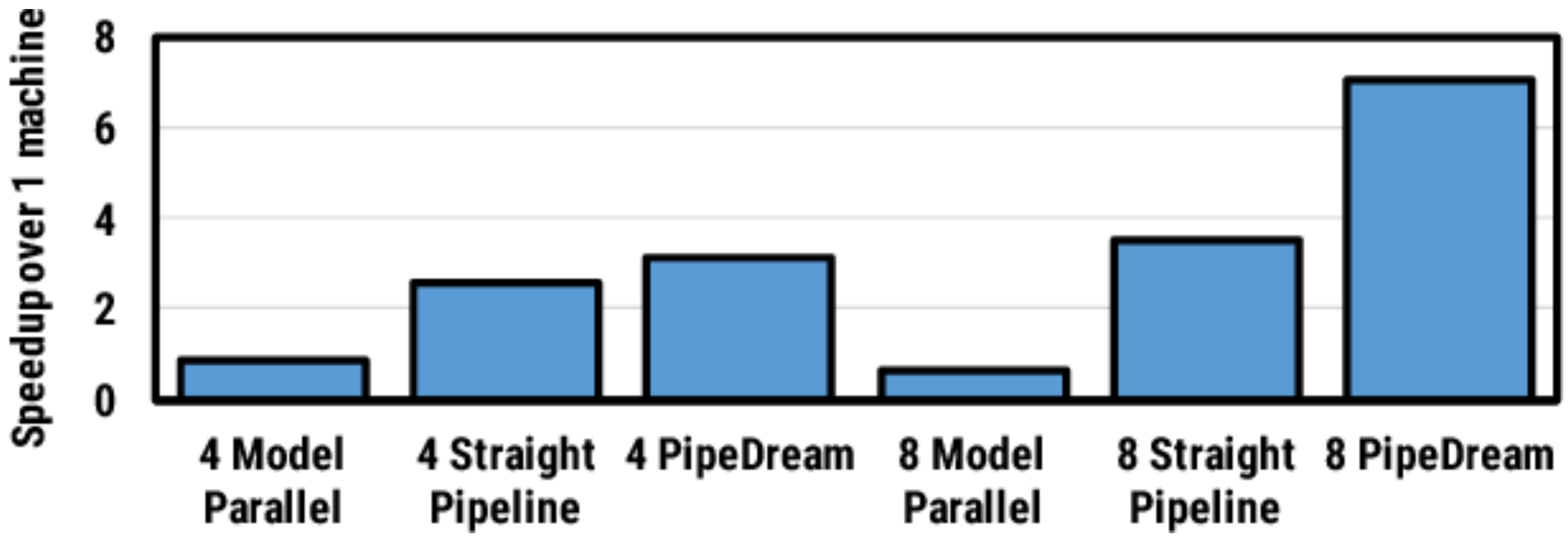
PIPELINE VS DATA PARALLELISM

DNN Model	# Machines (Cluster)	BSP speedup over 1 machine	PipeDream speedup over 1 machine	PipeDream speedup over BSP	PipeDream communication reduction over BSP
VGG16	4 (A)	1.47×	3.14×	2.13×	90%
	8 (A)	2.35×	7.04×	2.99×	95%
	16 (A)	3.28×	9.86×	3.00×	91%
	8 (B)	1.36×	6.98×	5.12×	95%
Inception-v3	8 (A)	7.66×	7.66×	1.00×	0%
	8 (B)	4.74×	6.88×	1.45×	47%
S2VT	4 (A)	1.10×	3.34×	3.01×	95%

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

EXPERIMENTATION

MODEL VS PIPELINE VS PIPEDREAM



WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

THANK YOU!

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON