A Scalable Benchmarking Infrastructure for Distributed Deep Learning

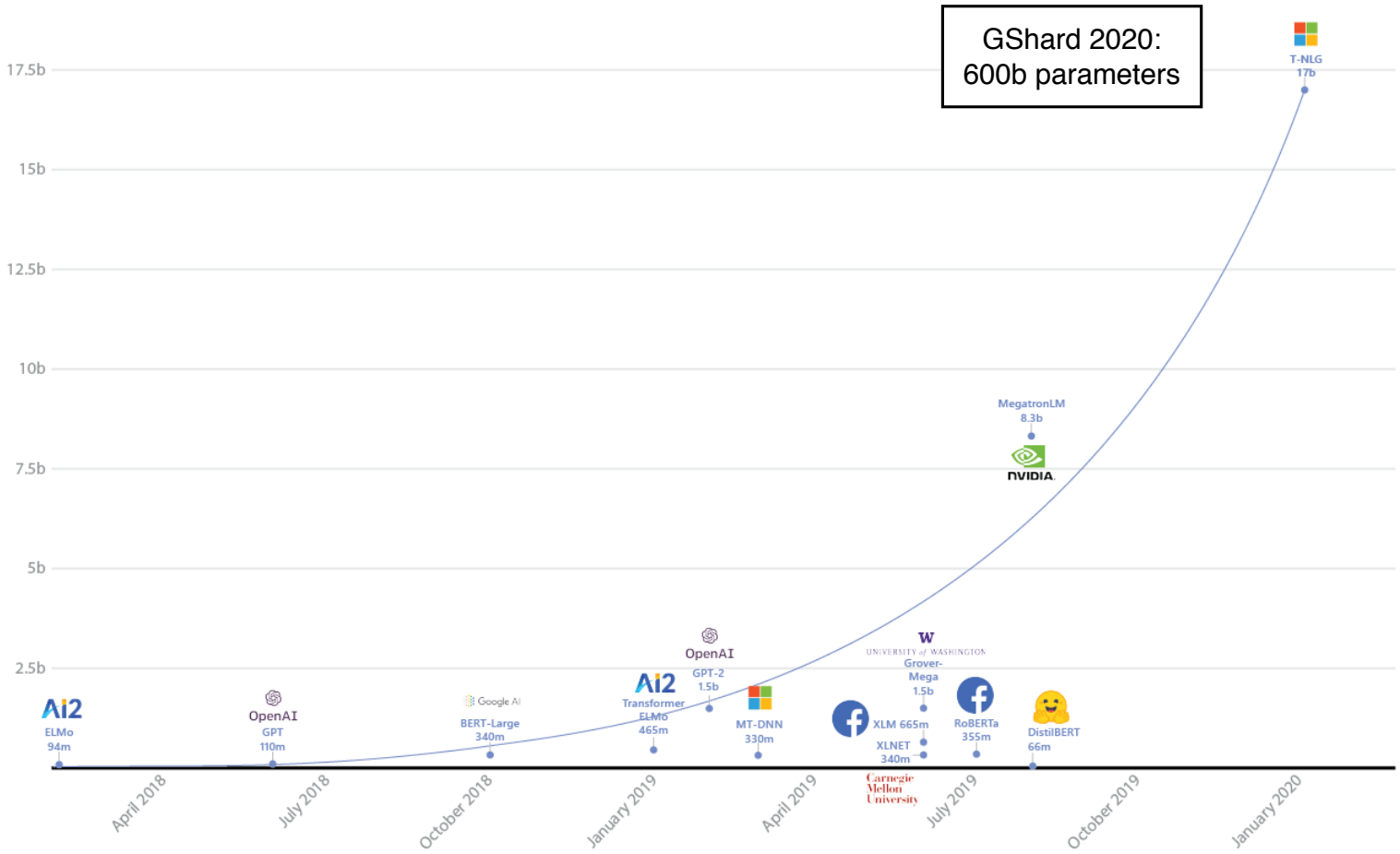
Matthijs Jansen

VU Amsterdam
m.s.jansen@vu.nl

February 2021

Model Size Explosion

Improve accuracy by using larger models

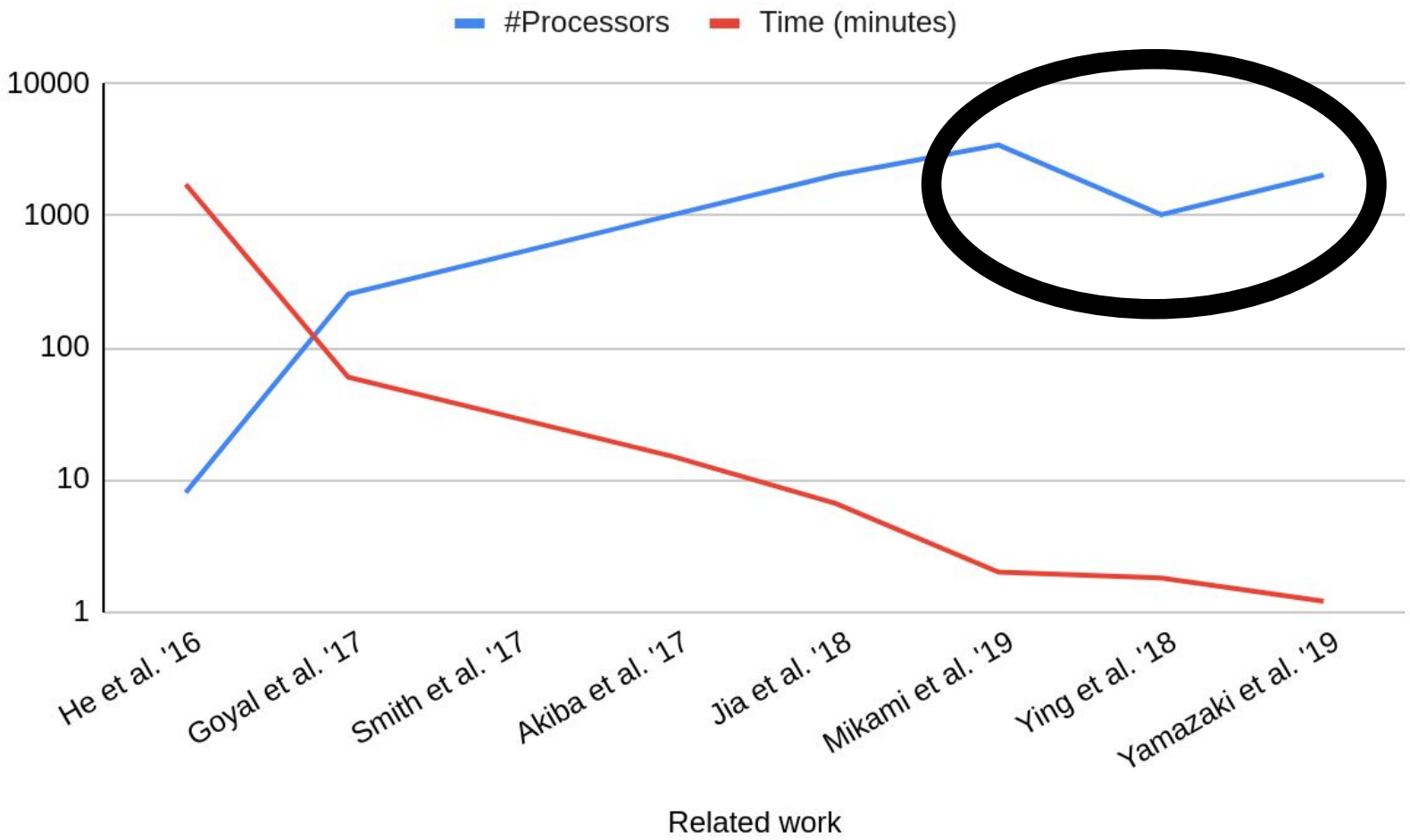


But these models do not fit on a single device anymore

[1] C Rosset. "Turing-nlg: A 17-billion-parameter language model by microsoft". In: *Microsoft Blog* (2019)

Distributed Deep Learning

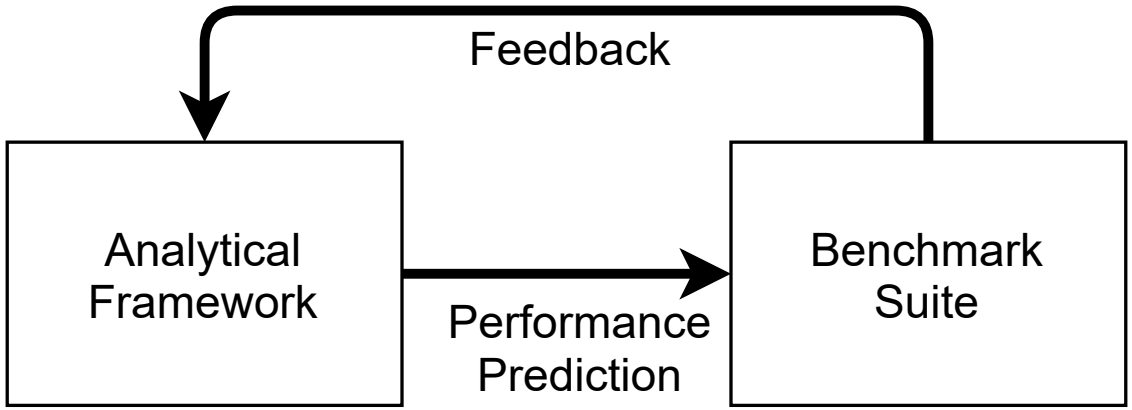
DDL improves performance and allows the use of large models
But more processors $\neq$ better performance per se



How to better understand performance?

Contributions

- DDLBench: Benchmark suite for distributed deep learning
- Accompanied by an analytical framework
- In-depth empirical analysis using DDLBench



Benchmark Suite

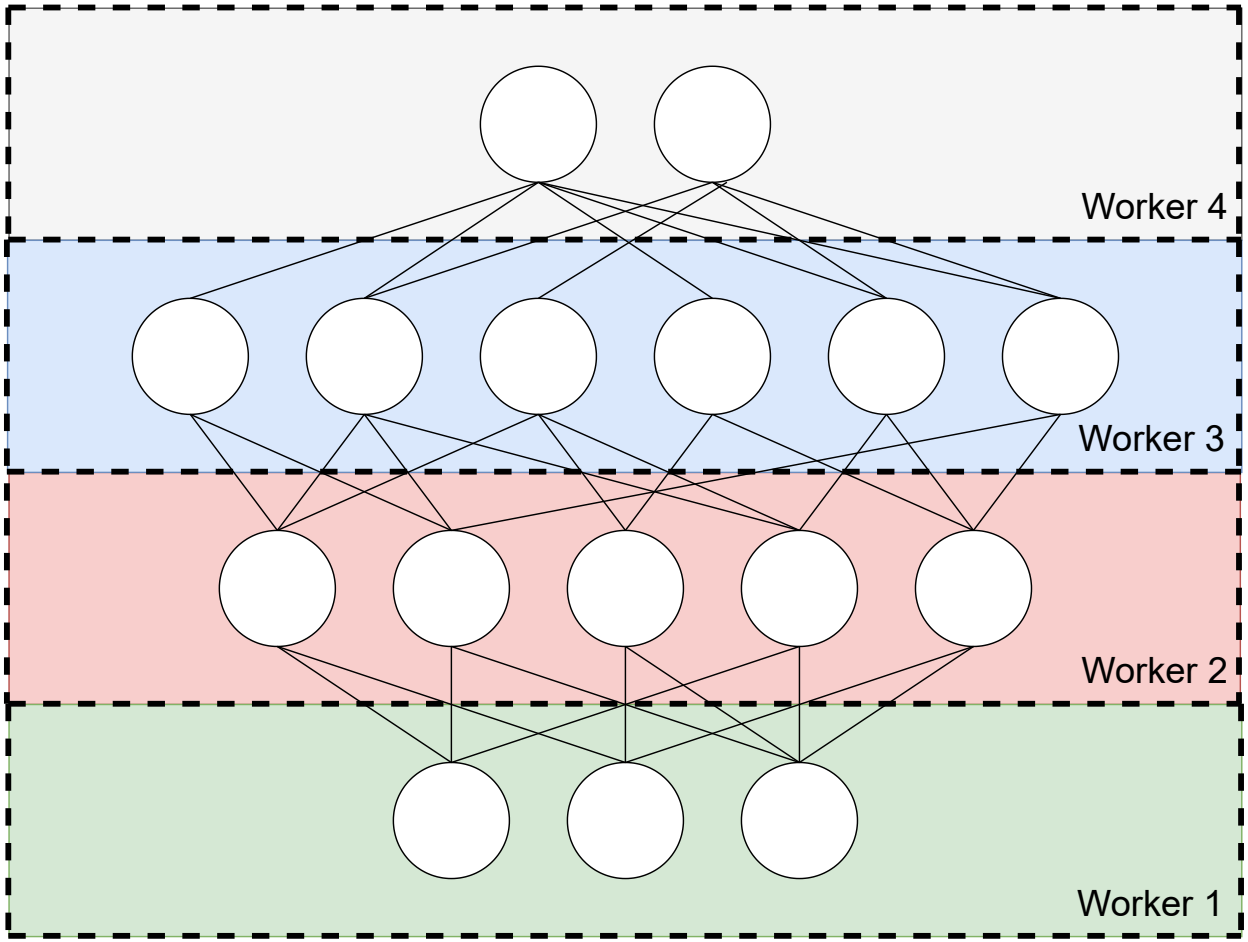
Datasets: #images, size of each image, classes, ...
Neural networks: #layers, layer type, ...

Table: A comparison of different distribution models for machine learning.

Model	*TF*	**PyTorch**	*CPU*	*GPU*	*Data*	*Model*	*Pipeline*
tf.distribute	X		X	X	X		
tf.Mesh	X		X	X	X	X	
PipeDream		X		X		X	X
(torch)GPipe	X	X		X		X	X
Horovod	X	X	X	X	X	X	
torch.distributed		X	X	X	X	X	

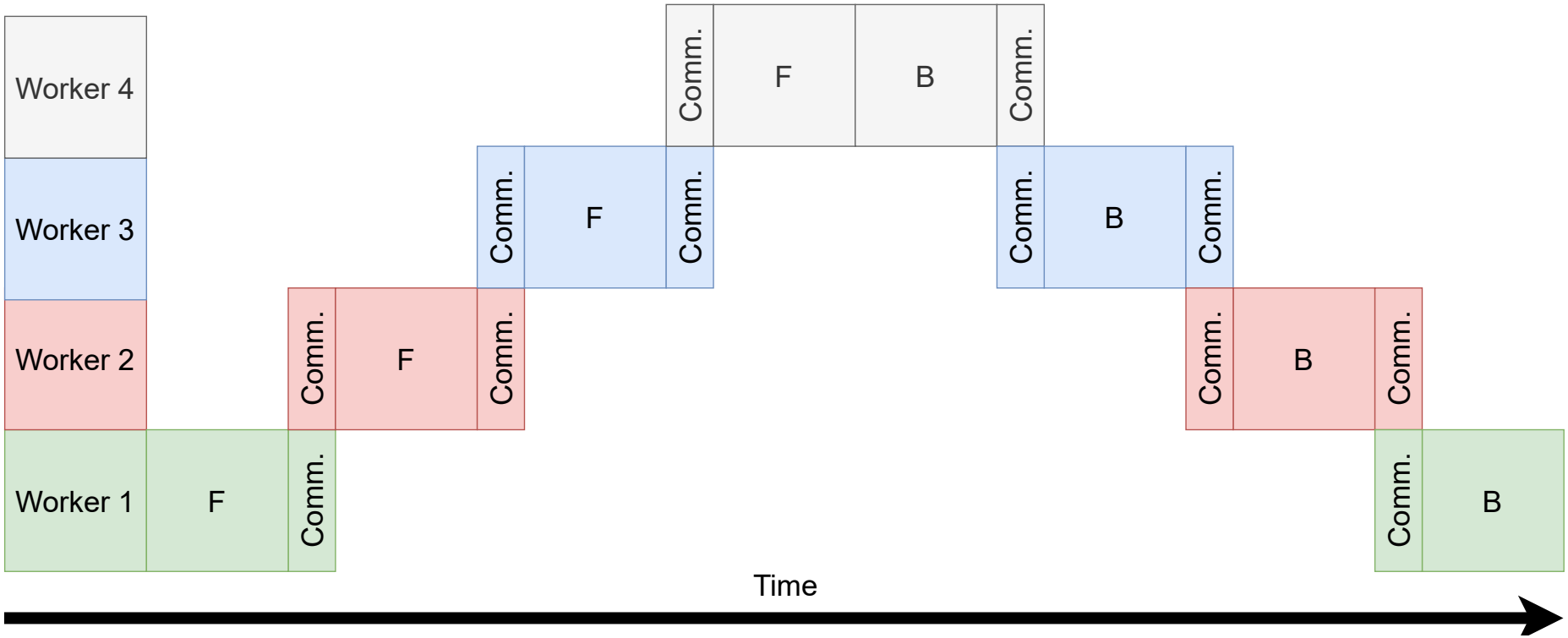
Model Parallelism

Distribute model over multiple workers



Execution Pipeline

Sequential execution due to dependency between workers

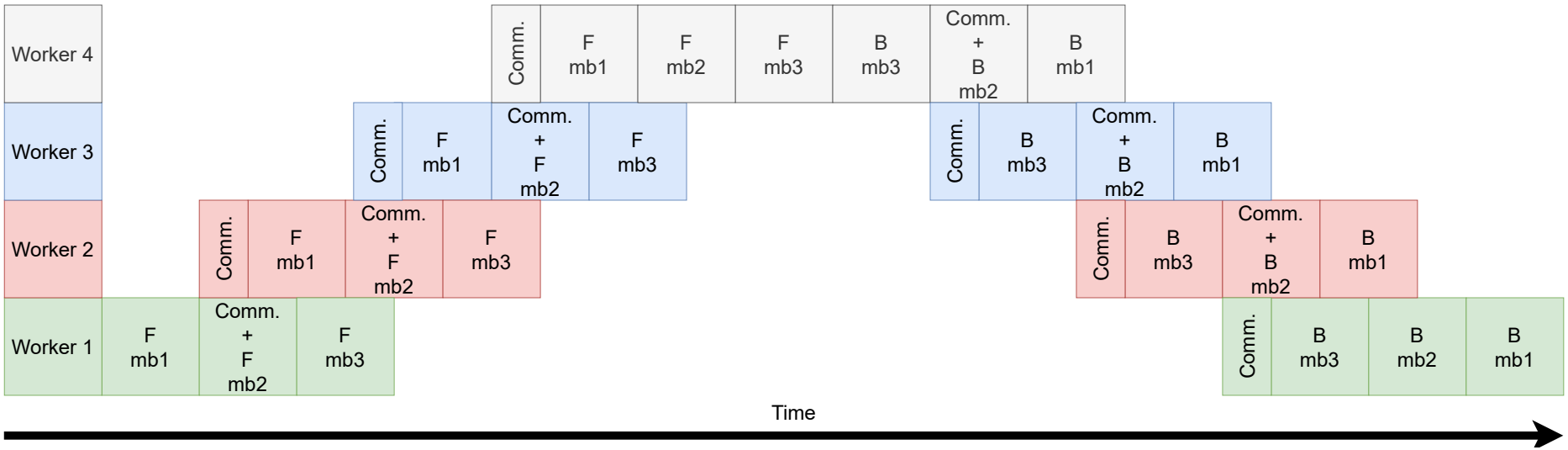


How to improve the performance?

TorchGPipe

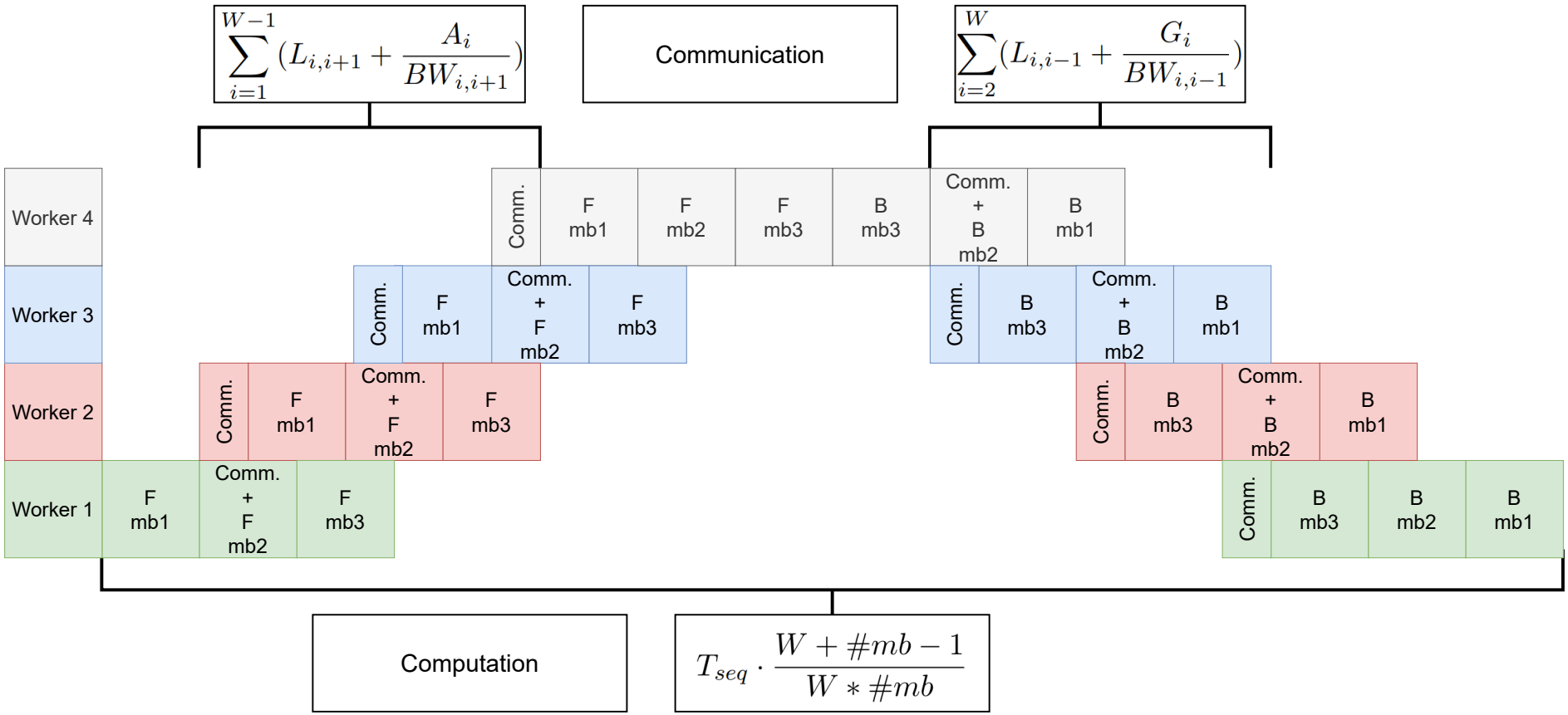
Split a batch up into multiple smaller micro-batches

No dependency between micro-batches, execute them in parallel



Performance Model

What are the communication patterns?
What are the computation patterns?
What scalability do we expect?



Results: Performance Scaling with Multiple Workers

- Data parallelism performs best
- Model parallelism uses least amount of memory
- Performance is not always as expected

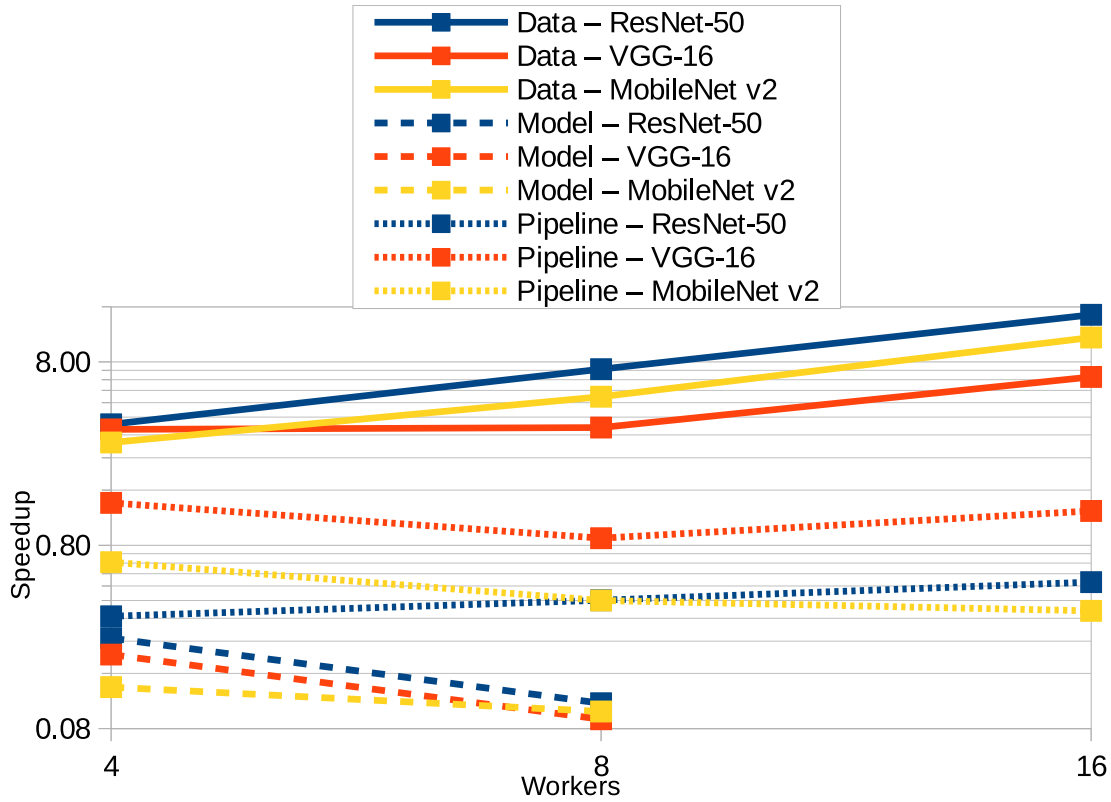


Figure: Data, Model and Pipeline parallelism

Conclusion

- DDLBench can capture the complex and dynamic behaviour of DDL applications
- Performance is not always as expected, better analytical models are needed
- v0.1: 4 datasets, 6 neural networks, 3 distribution models
- Designed with diversity and extensibility in mind
- Open-source: https://github.com/sara-nl/DDLBench