分布式训练系列

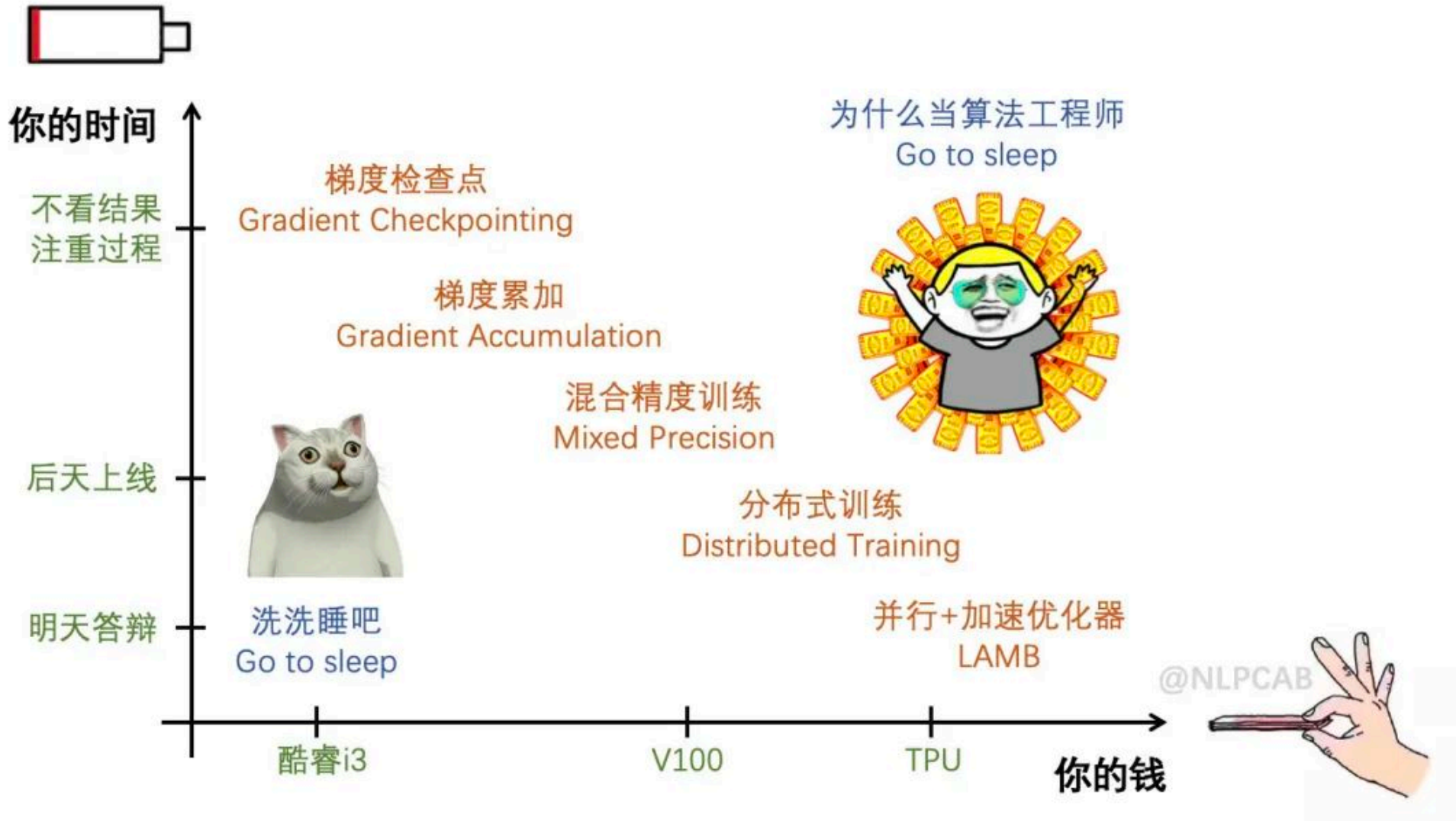
流水并行

ZOMI

关于本内容

1. 具体内容

- 大模型训练挑战
- AI框架的分布式
- AI集群架构
- AI集群通信
- 大模型算法
- **分布式并行算法：**数据并行 – 模型并行 – 流水并行
- 大模型混合并行
- 内存和计算优化
- 分布式并行总结

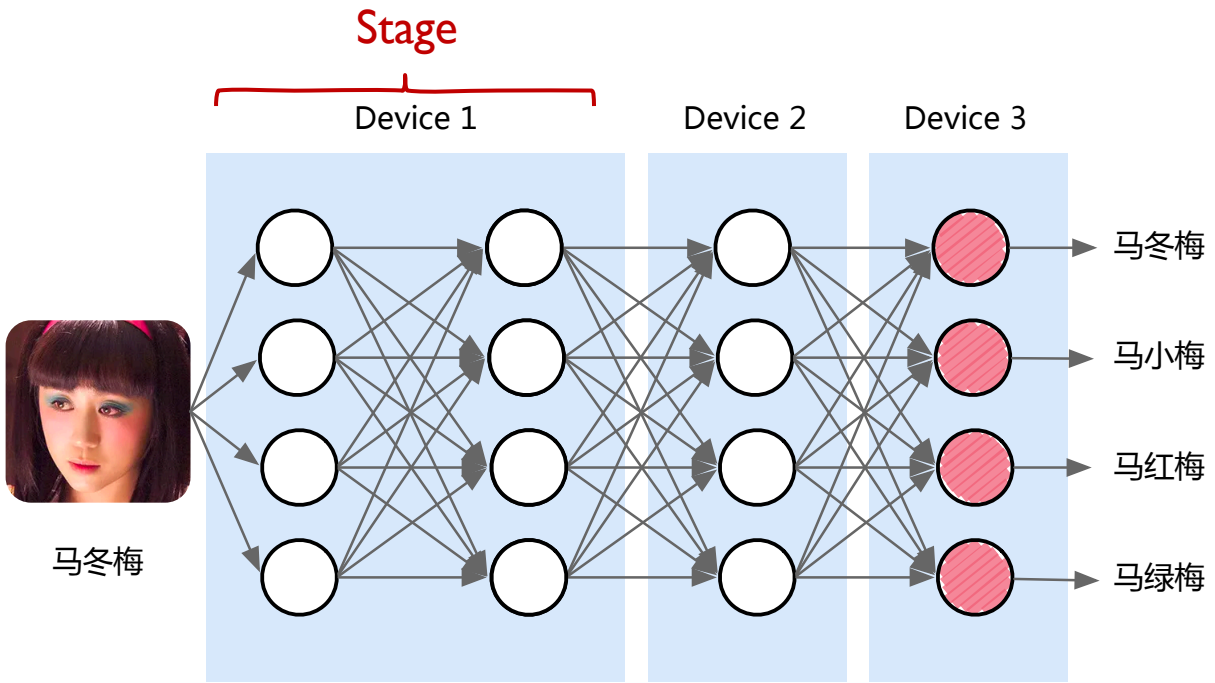


Model Parallelism, MP 模型并行

- Pipeline Parallelism 流水线并行
  - Naive PP
  - Gpipe
  - PipeDream
- Tensor Parallelism 张量并行

MP(I): Pipeline parallelism 流水线并行

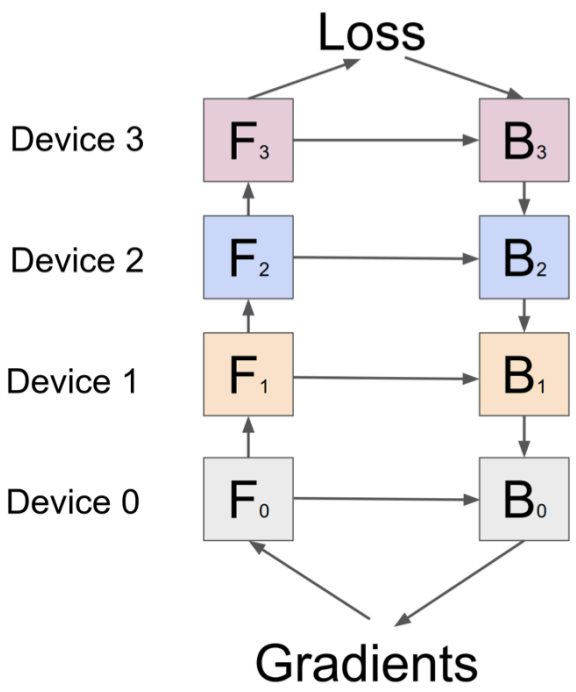
- Model divided layers into different devices, which we called pipeline parallelism
- 流水线并行：按模型layer层切分到不同设备，即层间并行

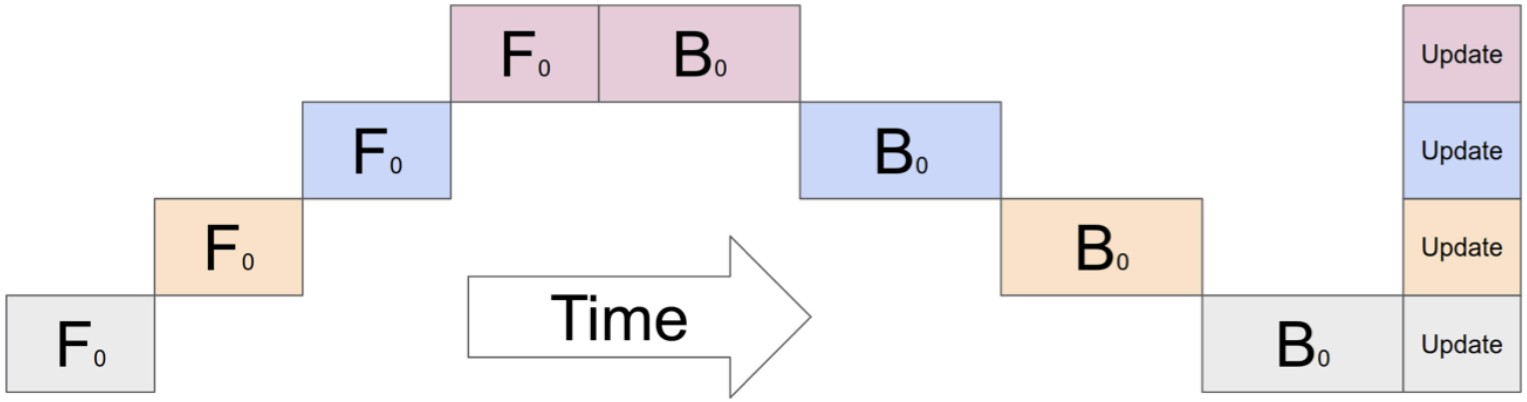


流水线并行（层间并行）

PP(I): Naïve Pipeline parallelism 朴素流水线并行

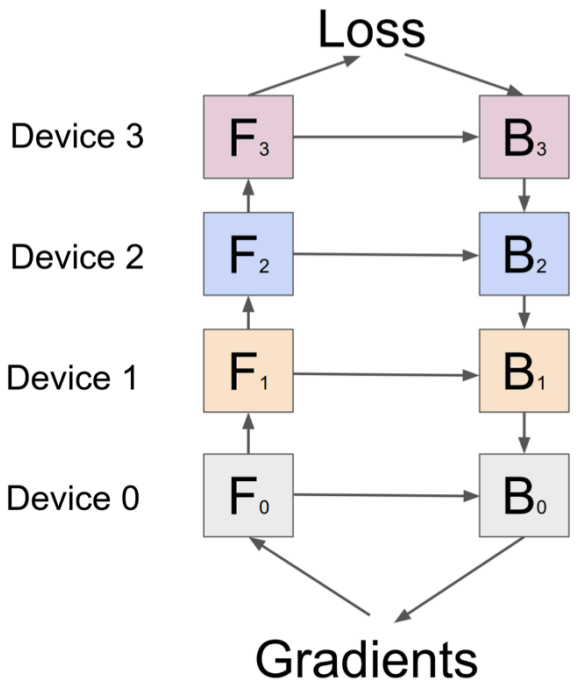
- Naive pipeline parallelism: Leads to severe under-utilization due to the sequential dependency of the network.
- 朴素流水线并行：同一时刻只有一个设备进行计算，其余设备处于空闲状态，计算设备利用率通常较低

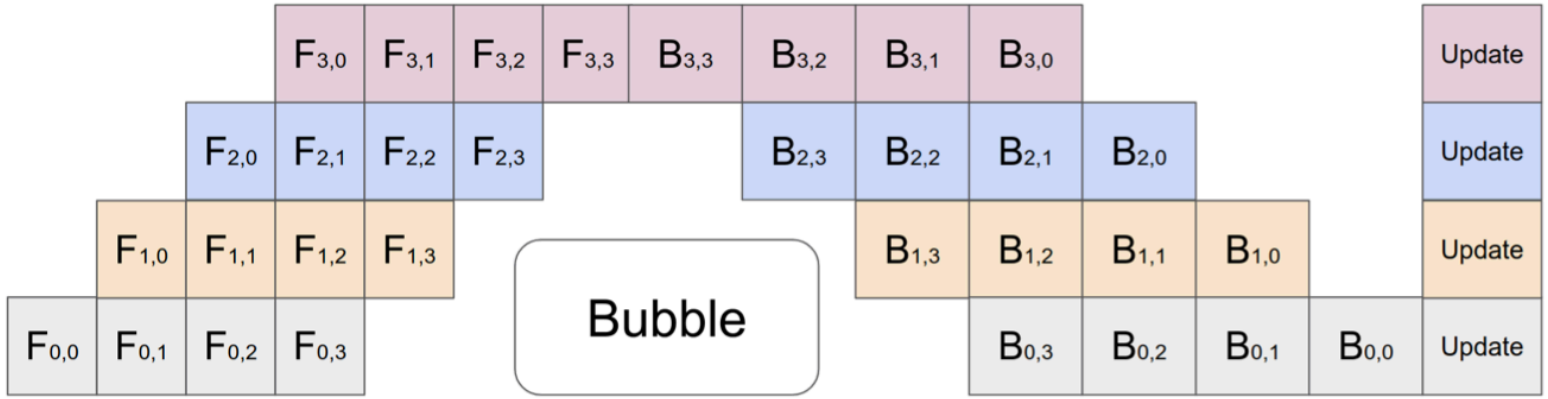




PP(I): Mini-batch Pipeline parallelism 小批次流水线并行

- **Mini-batch pipeline parallelism:** divides the input mini-batch into smaller micro-batches, enabling different accelerators to work on different micro-batches simultaneously.
- **小批次流水线并行**：将朴素流水线并行的 batch 再进行切分，减小设备间空闲状态的时间，可以显著提升流水线并行设备利用率。





Gpipe

1. Partition Stage
2. Micro-Batch & Pipeline
3. Re-Materialization

GPipe: Easy Scaling with Micro-Batch Pipeline Parallelism

Yanping Huang
huangyp@google.com

Youlong Cheng
ylc@google.com

Ankur Bapna
ankurbpn@google.com

Orhan Firat
orhanf@google.com

Mia Xu Chen
miachen@google.com

Dehao Chen
dehao@google.com

HyoukJoong Lee
hyouklee@google.com

Jiquan Ngiam
jngiam@google.com

Quoc V. Le
qvl@google.com

Yonghui Wu
yonghui@google.com

Zhifeng Chen
zhifengc@google.com

Gpipe

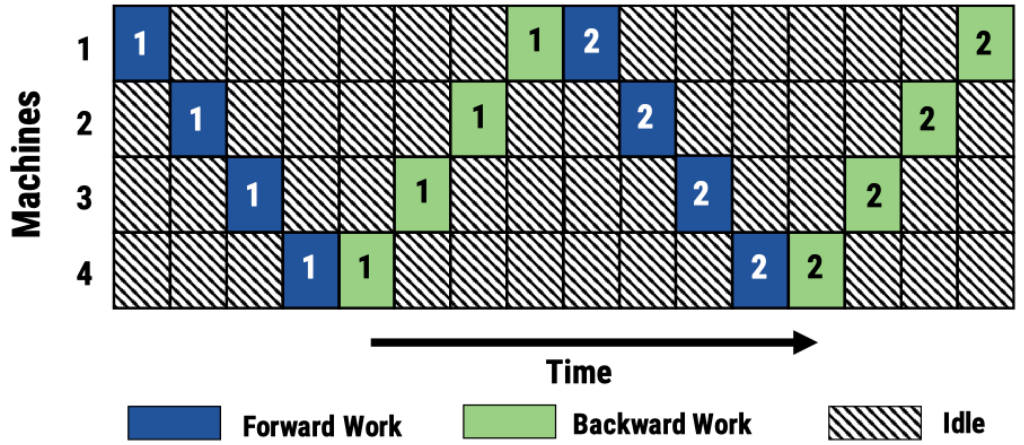
对于Transformer-L模型，在使用128块GPU的情况下，模型最大达到937.9GB，使用了128块GPU。

Table 1: Maximum model size of AmoebaNet supported by GPipe under different scenarios. Naive-1 refers to the sequential version without GPipe. Pipeline-k means k partitions with GPipe on k accelerators. AmoebaNet-D (L, D): AmoebaNet model with L normal cell layers and filter size D . Transformer-L: Transformer model with L layers, 2048 model and 8192 hidden dimensions. Each model parameter needs 12 bytes since we applied RMSProp during training.

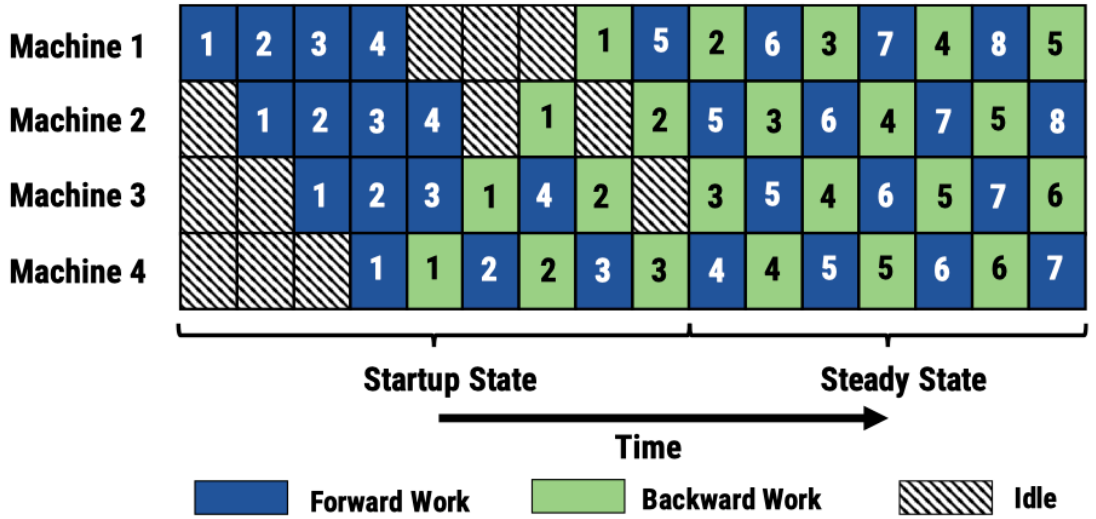
NVIDIA GPUs (8GB each)	Naive-1	Pipeline-1	Pipeline-2	Pipeline-4	Pipeline-8
AmoebaNet-D (L, D)	(18, 208)	(18, 416)	(18, 544)	(36, 544)	(72, 512)
# of Model Parameters	82M	318M	542M	1.05B	1.8B
Total Model Parameter Memory	1.05GB	3.8GB	6.45GB	12.53GB	24.62GB
Peak Activation Memory	6.26GB	3.46GB	8.11GB	15.21GB	26.24GB
Cloud TPUv3 (16GB each)	**Naive-1**	**Pipeline-1**	**Pipeline-8**	**Pipeline-32**	**Pipeline-128**
Transformer-L	3	13	103	415	1663
# of Model Parameters	282.2M	785.8M	5.3B	21.0B	83.9B
Total Model Parameter Memory	11.7G	8.8G	59.5G	235.1G	937.9G
Peak Activation Memory	3.15G	6.4G	50.9G	199.9G	796.1G

Pipeline Mode 并行的模式

F-then-B



1F1B



PipeDream

one-forward-one-backward-round-robin

PipeDream: Fast and Efficient Pipeline Parallel DNN Training

Aaron Harlap[†]* Deepak Narayanan[‡]*
Amar Phanishayee[⋆] Vivek Seshadri[⋆] Nikhil Devanur[⋆] Greg Ganger[†] Phil Gibbons[†]

[⋆]*Microsoft Research* [†]*Carnegie Mellon University* [‡]*Stanford University*

Abstract

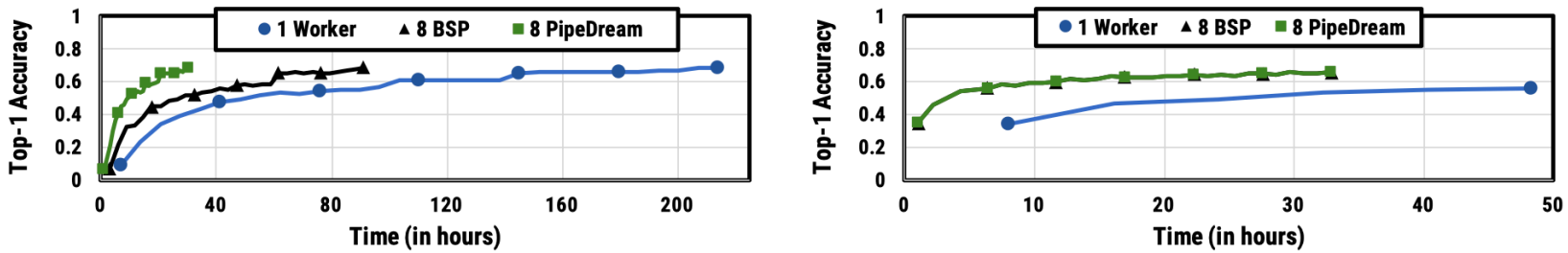
PipeDream is a Deep Neural Network (DNN) training system for GPUs that parallelizes computation by pipelining execution across multiple machines. Its *pipeline parallel* computing model avoids the slowdowns faced by data-parallel training when large models and/or limited network bandwidth induce high communication-to-computation ratios. PipeDream reduces communication by up to 95% for large DNNs relative to data-parallel training, and allows perfect overlap of communication and computation. PipeDream keeps all available GPUs productive by systematically partitioning DNN layers among them to balance work and minimize communication, versions model parameters for backward pass correctness, and schedules the forward and backward passes of different inputs in round-robin fashion to optimize

reflects updates across all inputs. The amount of data communicated per aggregation is proportional to the size of the model. Although data-parallel training works well with some popular models that have high computation-to-communication ratios, two important trends threaten its efficacy. First, growing model sizes increase per-aggregation communication. Indeed, some widely-used models are large enough that the communication overheads already eclipse computation time, limiting scaling and dominating total training time (e.g., up to 85% of training time for VGG-16 [36]). Second, rapid increases in GPU compute capacity further shift the bottleneck of training towards communication across models. Our results show these effects quantitatively (Figure 1) for three generations of NVIDIA GPUs (Kepler, Pascal, and Volta), across five different DNN models.

PipeDream

DNN Model	# Machines (Cluster)	BSP speedup over 1 machine	PipeDream Config	PipeDream speedup over 1 machine	PipeDream speedup over BSP	PipeDream communication reduction over BSP
VGG16	4 (A)	1.47×	2-1-1	3.14×	2.13×	90%
	8 (A)	2.35×	7-1	7.04×	2.99×	95%
	16 (A)	3.28×	9-5-1-1	9.86×	3.00×	91%
	8 (B)	1.36×	7-1	6.98×	5.12×	95%
Inception-v3	8 (A)	7.66×	8	7.66×	1.00×	0%
	8 (B)	4.74×	7-1	6.88×	1.45×	47%
S2VT	4 (A)	1.10×	2-1-1	3.34×	3.01×	95%

Table 1: Summary of results comparing PipeDream with data-parallel configurations (BSP) when training models to their advertised final accuracy. "PipeDream config" represents the configuration generated by our partinioning algorithm—e.g., "2-1-1" is a configuration in which the model is split into three stages with the first stage replicated across 2 machines.



(a) VGG16 **(b)** Inception-v3

Figure 10: Accuracy vs. time for VGG16 and Inception-v3 with 8 machines on Cluster-A

Summary 总结

1. 模型并行分为张量并行和流水线并行，张量并行主要层内并行、流水线主要层间并行，一般来说机内使用张量并行，机间使用数据并行；
2. 流水线并行作为模型并行中的一部分，一般不会单独使用，而是通过混合张量并行、数据并行等方式共同进行的；
3. 流水线并行从初始化的 F-then-B 模式逐渐发展到 1F1B，从 mini-batch 到 micro-batch 每次 batch 粒度更细；

Inference

I. https://zhuanlan.zhihu.com/p/450854172 全网最全-超大模型+分布式训练架构和经典论文

II. Huang, Yanping, et al. "Gpipe: Efficient training of giant neural networks using pipeline parallelism." Advances in neural information processing systems 32 (2019).

III. Harlap, Aaron, et al. "Pipedream: Fast and efficient pipeline parallel dnn training." arXiv preprint arXiv:1806.03377 (2018).

IV. Narayanan, Deepak, et al. "PipeDream: generalized pipeline parallelism for DNN training." Proceedings of the 27th ACM Symposium on Operating Systems Principles. 2019.

V. Narayanan, Deepak, et al. "Efficient large-scale language model training on gpu clusters using megatron-lm." Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2021.