AI 芯片 – AI 计算体系

深度学习计算模式

ZOMI

Talk Overview

1. AI 计算体系

- 深度学习计算模式
- 计算体系与矩阵运算

2. AI 芯片基础

- 通用处理器 CPU
- 从数据看 CPU 计算
- 通用图形处理器 GPU
- AI专用处理器 NPU/TPU
- 计算体系架构的黄金10年

Talk Overview

I. 深度学习计算模式

- The History – AI 的发展和范式
- Models Architecture – 经典模型结构
- Quantization and Pruning – 模型量化与剪枝
- Efficient Models – 轻量化网络模型
- Models Parallel – 大模型分布式并行

轻量化网络模型设计

经典的轻量级模型

CNN 系列

1. SqueezeNet 系列（2016）
2. ShuffleNet 系列（2017）
3. MobileNet 系列（2017）
4. ESPnet 系列（2018）
5. FBNet系列（2018）
6. EfficientNet 系列（2019）
7. GhostNet 系列（2019）

Transformer 系列

1. MobileViT （2021）
2. Mobile-Former（2021）
3. EfficientFormer（2022）


推理系统模型小型化
推理参数介绍
01 什么是FLOPS/MACC?
如何计算模型参数量?

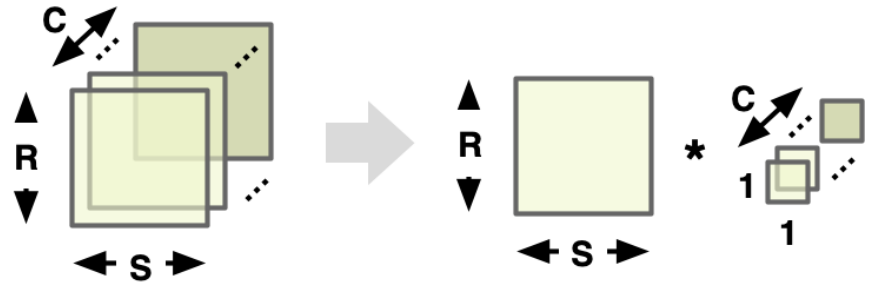

推理系统模型小型化
CNN模型小型化
02(下) CNN小型化算法解读
ESPNet/GhostNet


推理系统模型小型化
CNN模型小型化
02(下) CNN小型化算法解读
Shuffle/Mobilenet

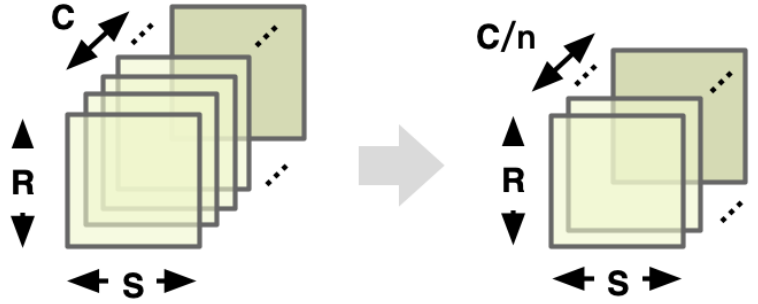

推理系统-模型小型化 03
Transformer
模型小型化

轻量化网络 Efficient DNN Models

- **设计轻量化网络模型主要2个方法：**
  - 通过改变网络模型不同层的layer shape或者卷积方式
  - 通过 Neural Architecture Search(NAS) 来搜索更轻量化的网络模型



Decompose large filters into smaller filters



Reduce number of channels before large filter convolution

轻量化网络 Efficient DNN Models

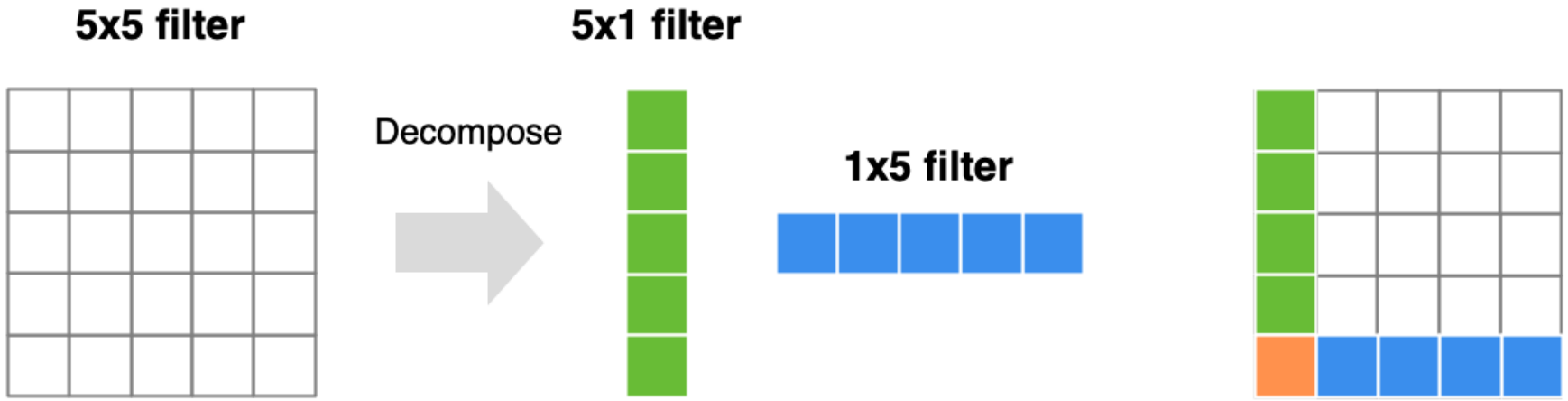
- **设计轻量化网络模型主要2个方法：**
  - 通过改变网络模型不同层的layer shape或者卷积方式
  - 通过 Neural Architecture Search(NAS) 来搜索更轻量化的网络模型

Models	Year	Accuracy	Layers	Weights	MACs
AlexNet	2012	80.4%	8	61M	724M
MobileNet	2017	89.5%	28	4M	569M

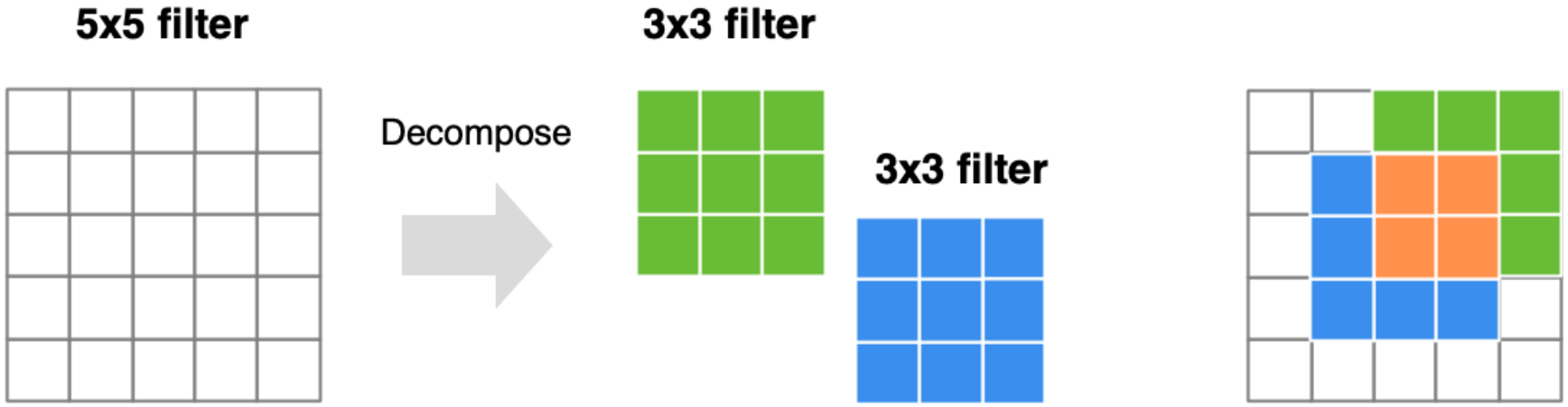
减少空间大小 Reduce Spatial Size: Stacked Filter

GoogleNet

InceptionNet V3

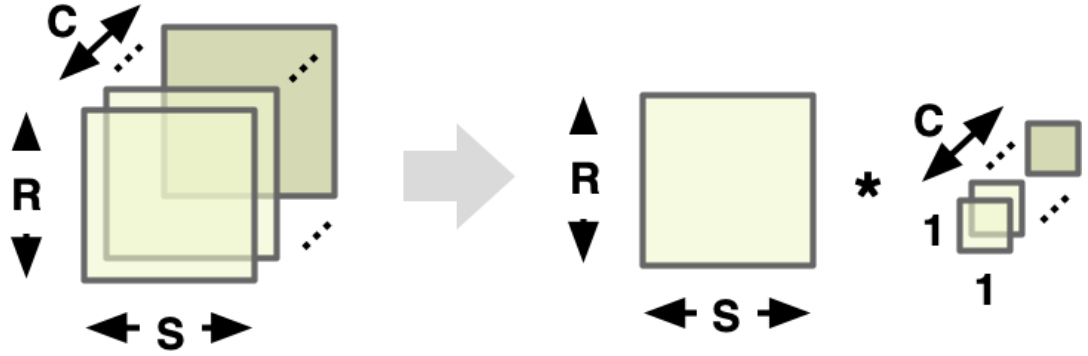


VGG

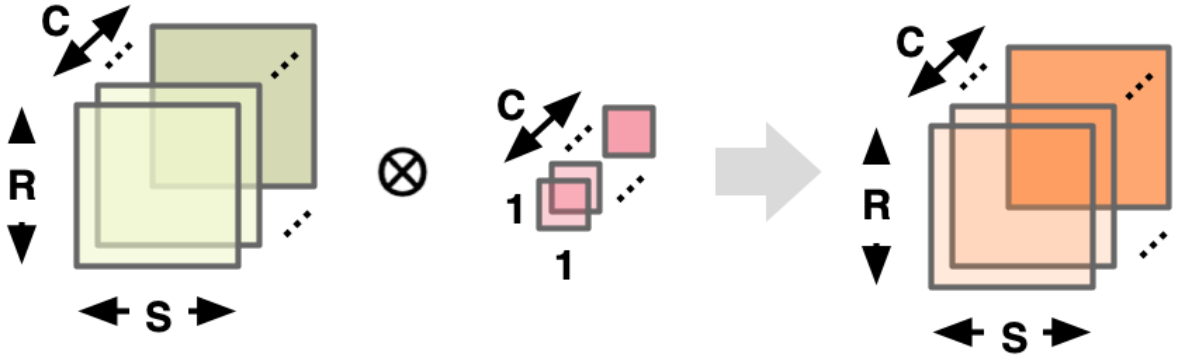


减少 Channels：Group of Filters/1x1 Convolution

Mix Information Across Groups
Pointwise(1x1) Convolution
MobileNetV1

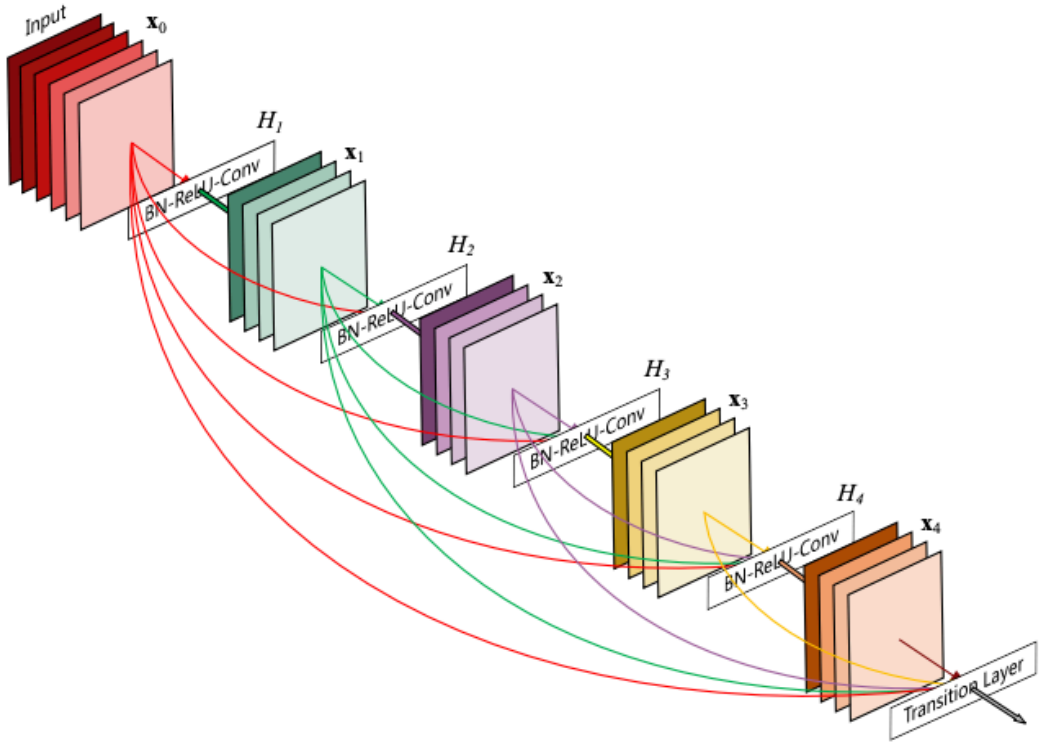


Use **1x1 filter** to summarize cross-channel information

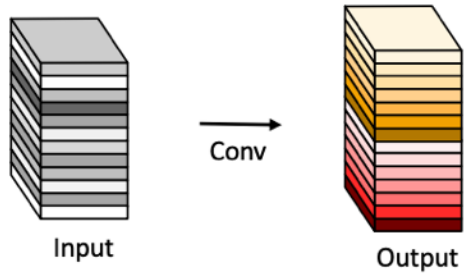


减少 Filters(M): Feature Map Reuse

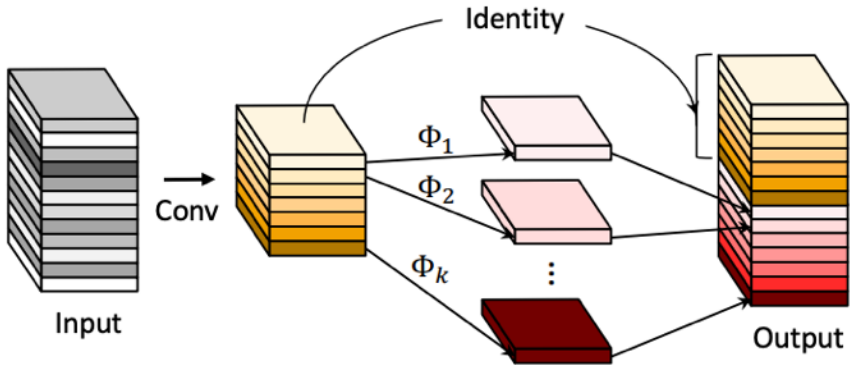
DenseNetV1 Reuses feature map
from multiple layers



GhostNet
Extract feature in feature maps



(a) The convolutional layer.



(b) The Ghost module.

AI 计算模式思考 (III)

卷积核方面 Convolution：

1. 大卷积核用多个小卷积核代替
2. 单一尺寸卷积核用多尺寸卷积核代替
3. 固定形状卷积核趋于使用可变形卷积核
4. 使用1×1卷积核 - bottleneck结构

卷积层通道方面 Channel：

1. 标准卷积用depthwise卷积代替
2. 使用分组卷积
3. 分组卷积前使用 channel shuffle
4. 4. 通道加权计算

卷积层连接方面 Connection：

1. 使用skip connection，让模型更深
2. densely connection，融合其它层特征输出

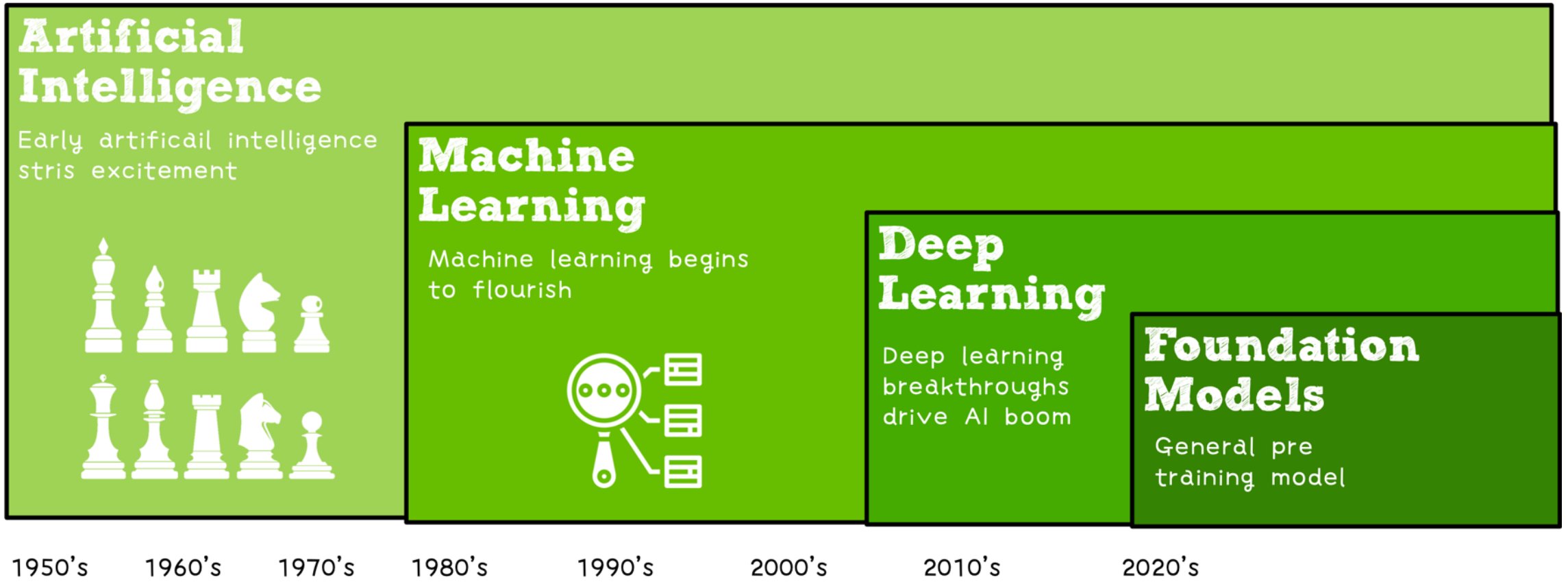
大模型
分布式并行


半小时精通
分布式系统
AI框架、通信&集群管理

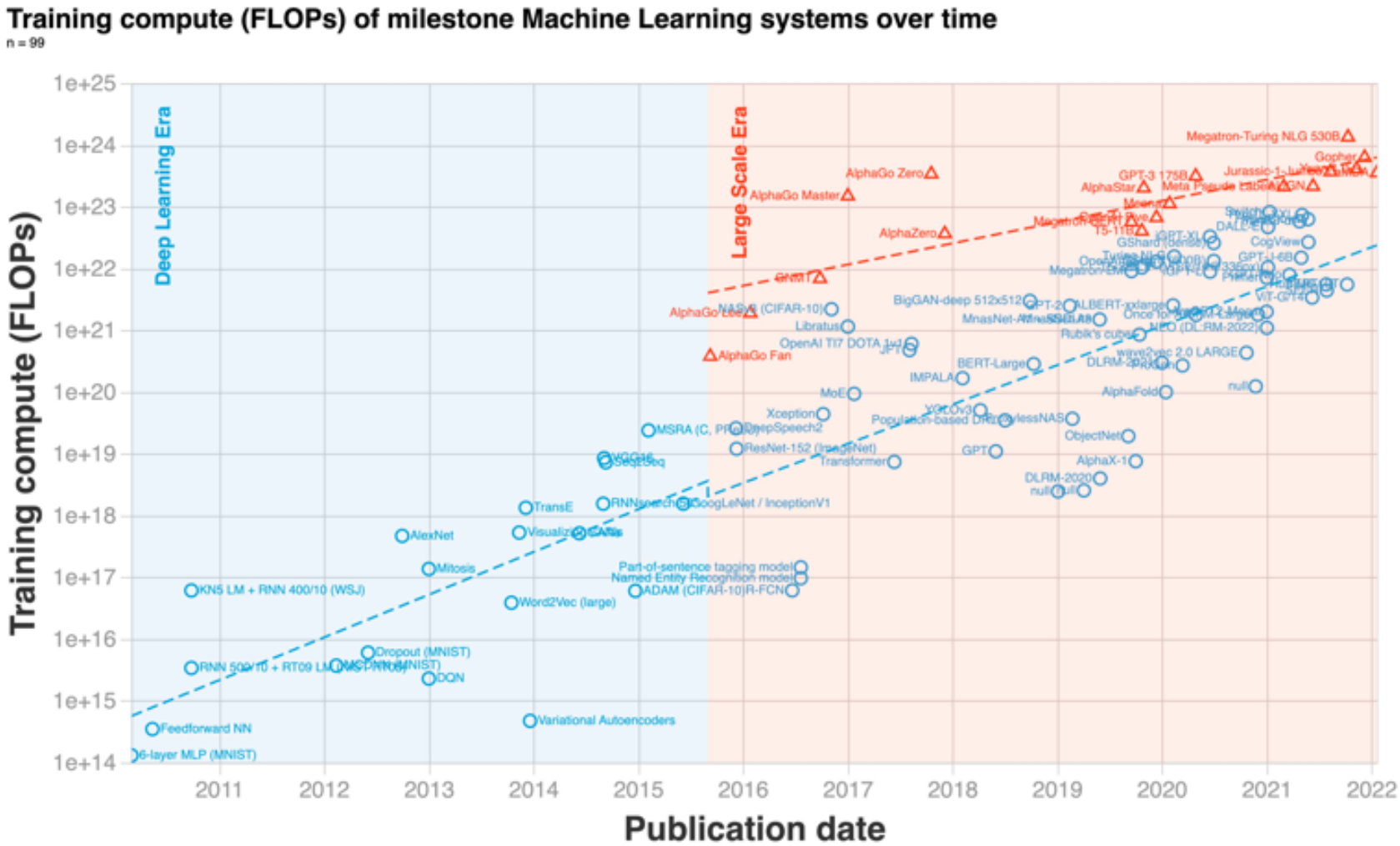

半小时精通
分布式算法
从十亿规模到万亿规模


半小时精通
分布式并行系列
从入门到精通

AI vs. Machine Learning vs. Deep Learning.

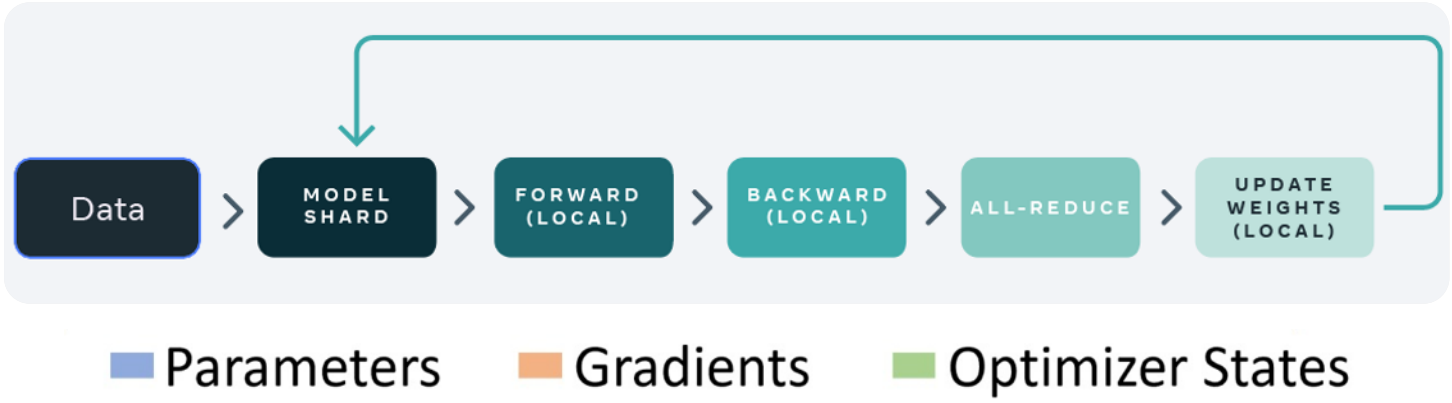


大模型的时代来临



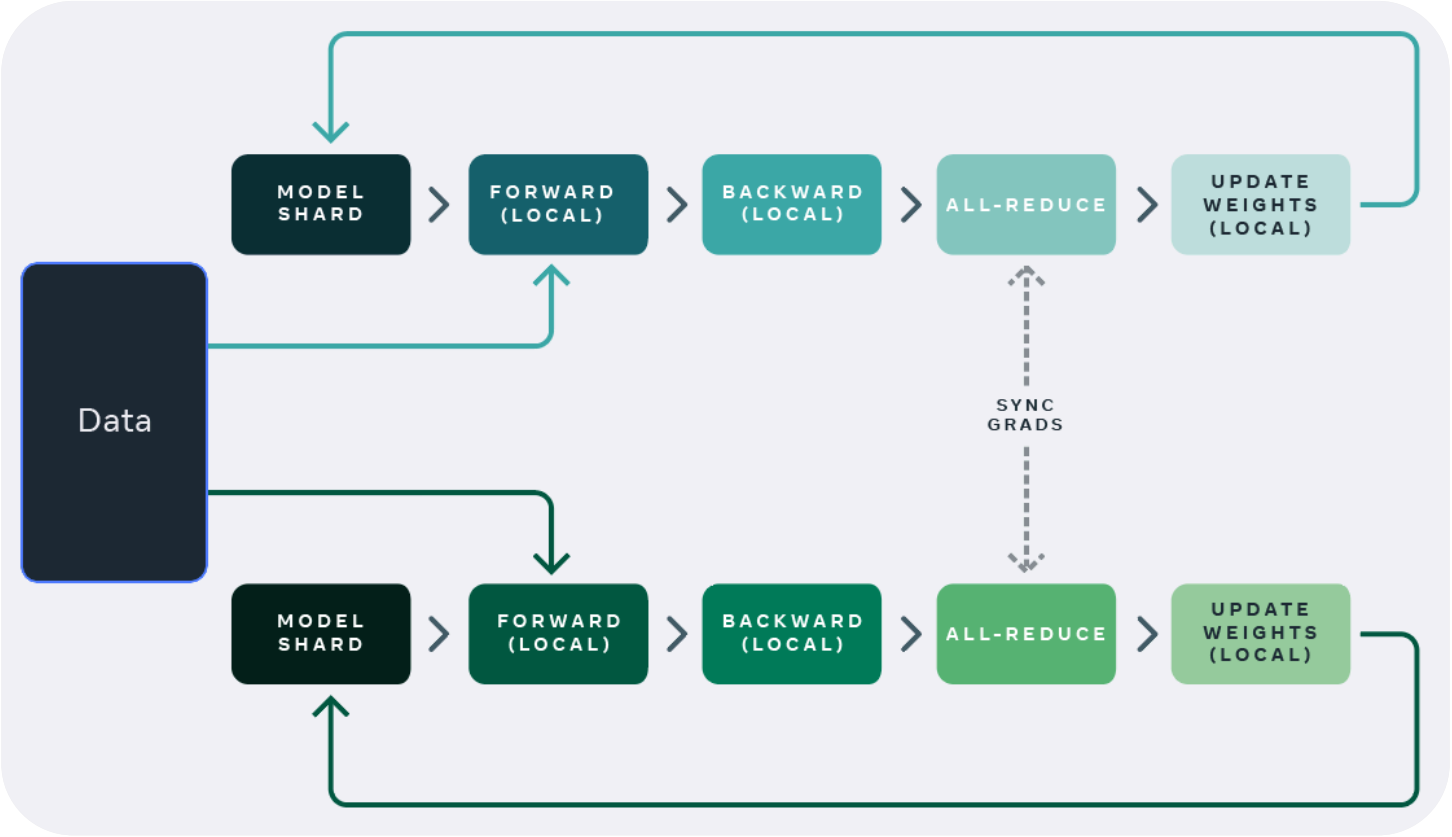
Data parallelism

1. Data parallelism, DP
2. Distribution Data Parallel, DDP
3. Fully Sharded Data Parallel, FSDP



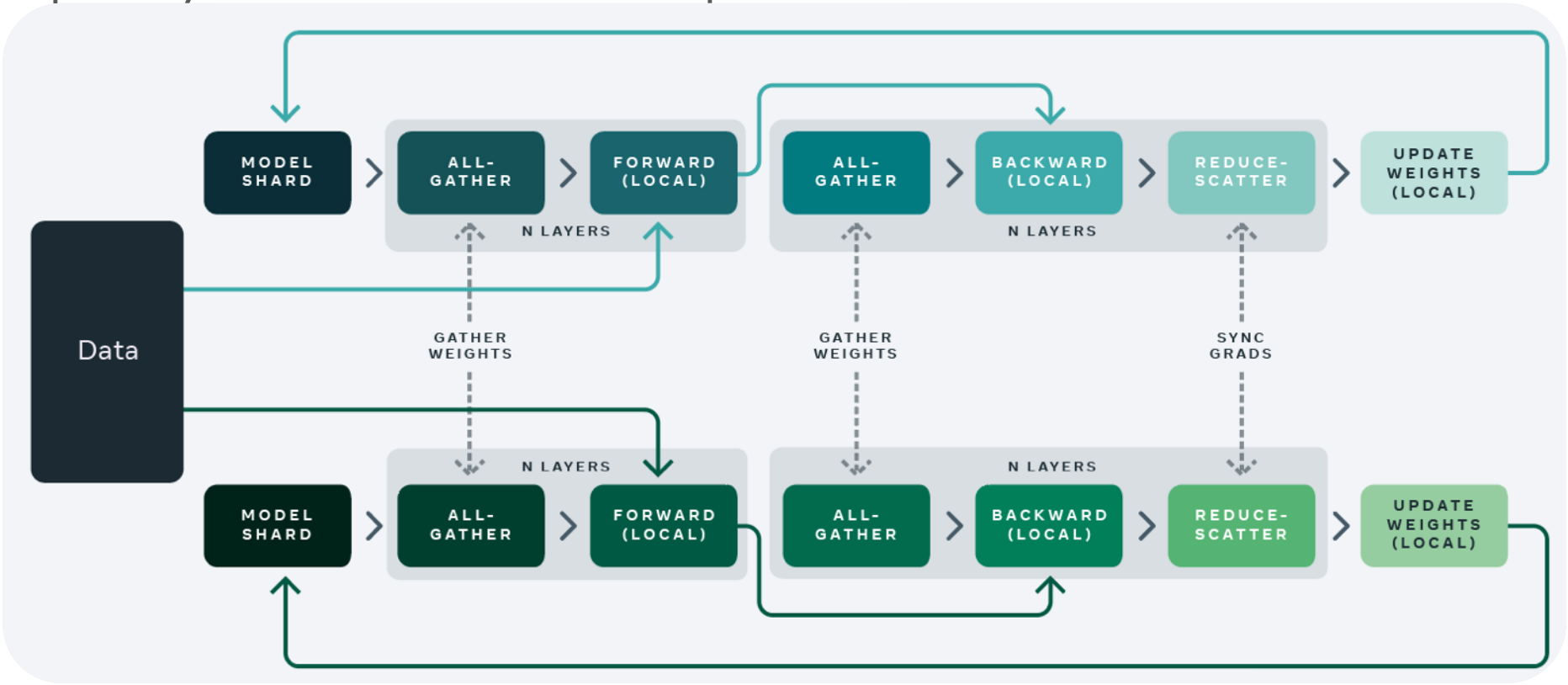
Data parallelism, DP

- Data Parallel automatically splits training data and sends model jobs to multiple GPUs. After each model completed, Data Parallel will Accumulate Gradients.

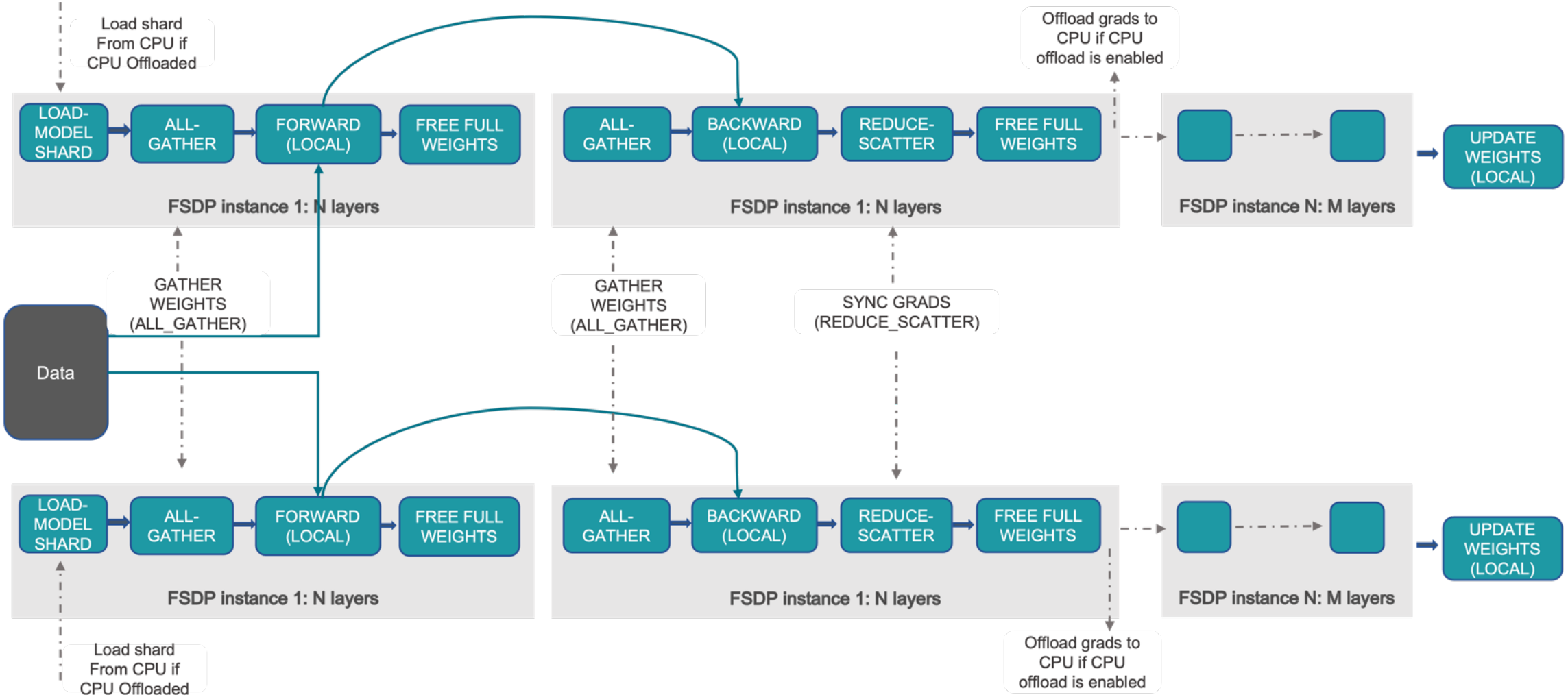


Fully Sharded Data Parallel, FSDP

- FSDP shards all of model's parameters, gradients and optimizer states across data-parallel workers and can optionally offload the sharded model parameters to CPUs.



Fully Sharded Data Parallel, FSDP



Megatron-LM 语言大模型

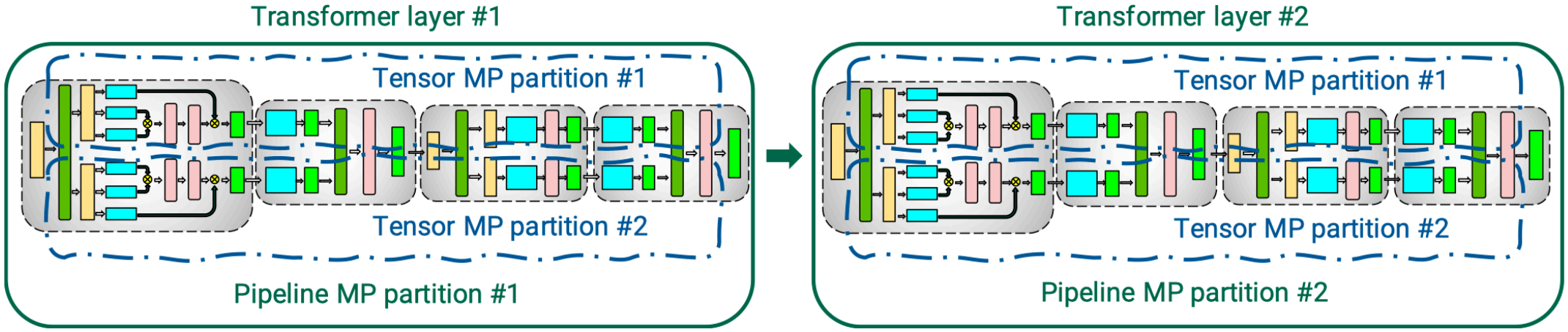
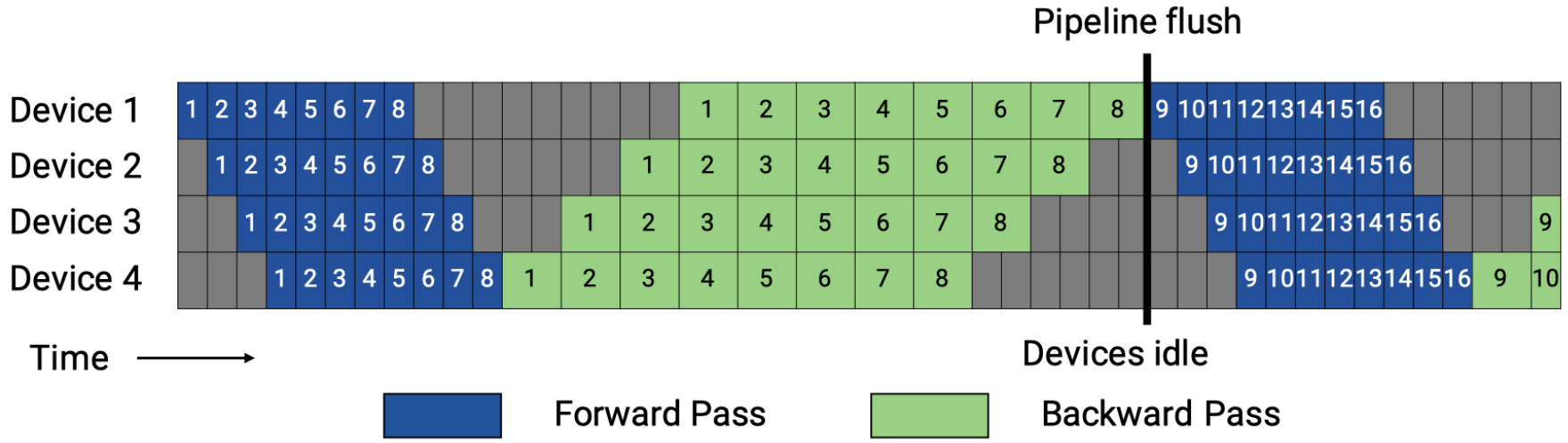
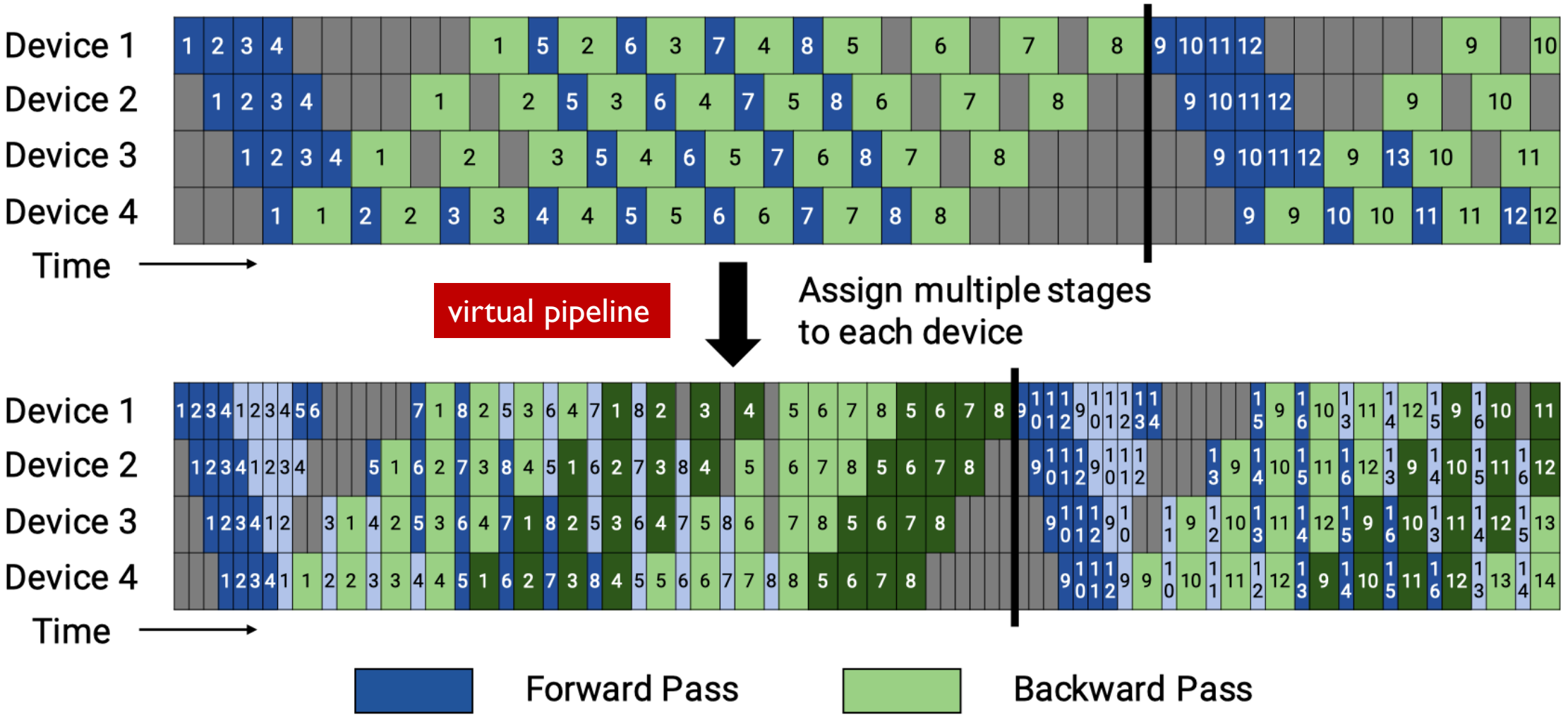


Figure 2: Combination of tensor and pipeline model parallelism (MP) used in this work for transformer-based models.



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.

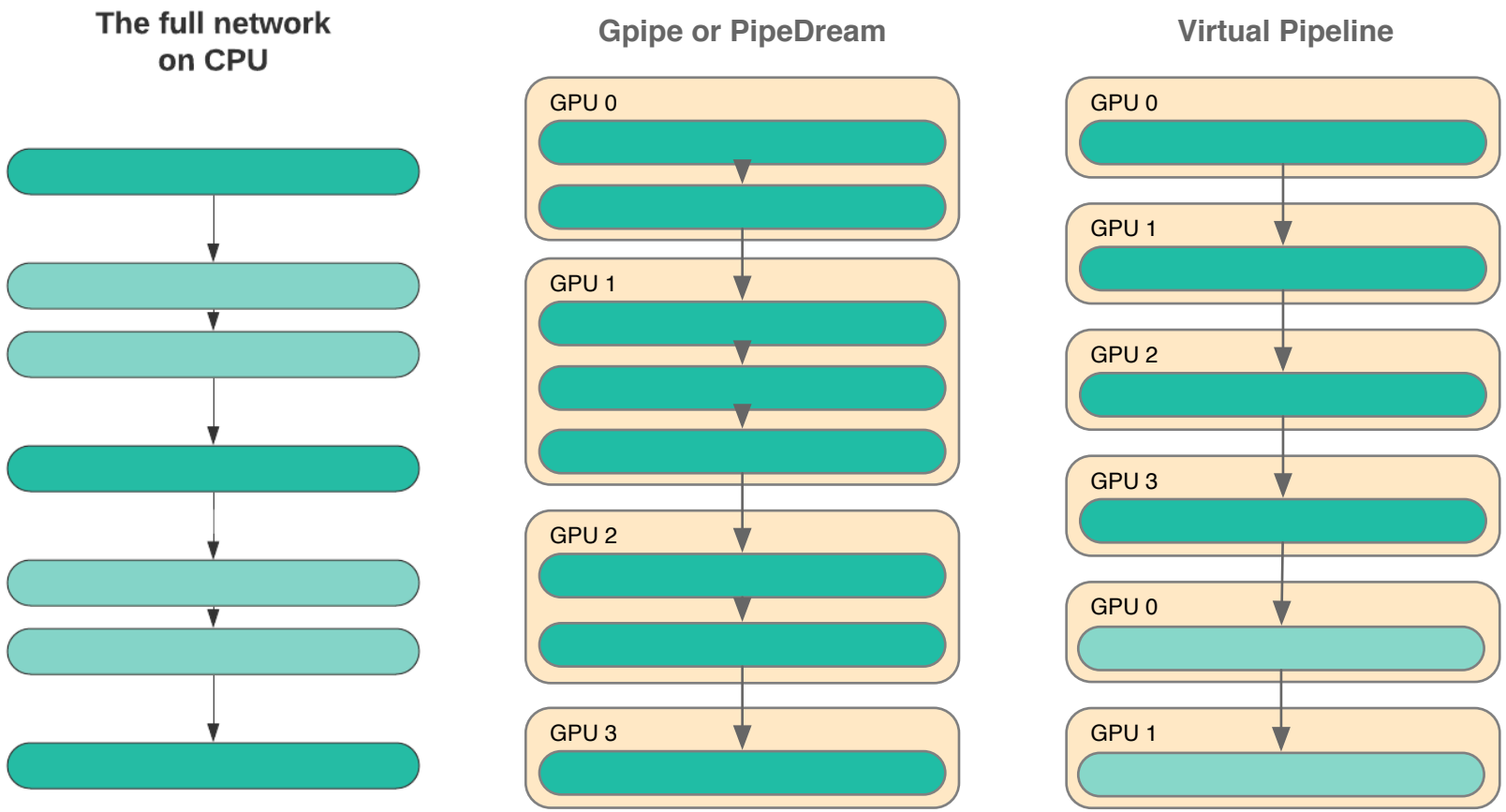
Megatron-LM 语言大模型



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.

Megatron-LM 语言大模型

在 device 数量不变的情况下，分出更多的 pipeline stage，以更多的通信量，换取空泡比率降低



Megatron-LM 语言大模型

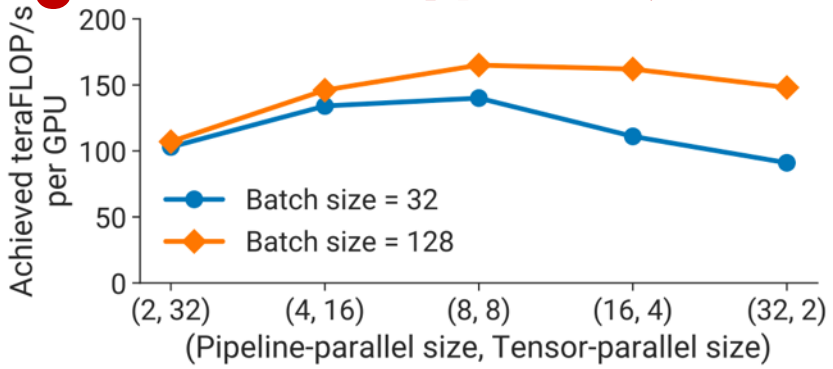


Figure 13: Throughput per GPU of various parallel configurations that combine pipeline and tensor model parallelism using a GPT model with 162.2 billion parameters and 64 A100 GPUs.

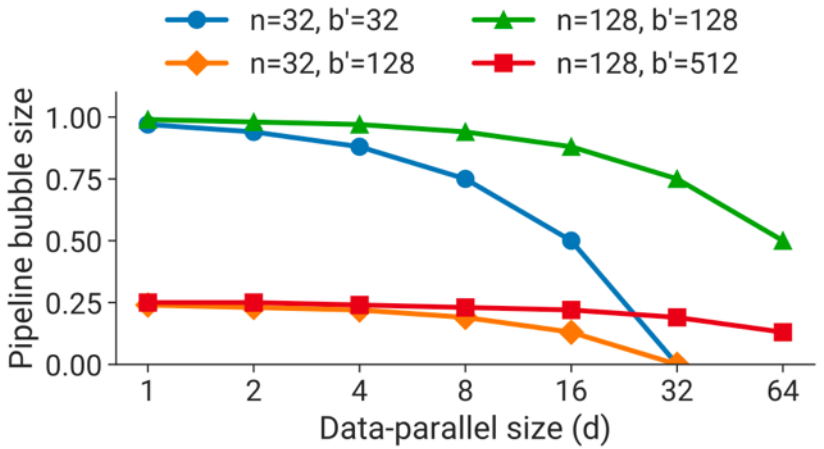


Figure 6: Fraction of time spent idling due to pipeline flush (pipeline bubble size) versus data-parallel size (d), for different numbers of GPUs (n) and ratio of batch size to microbatch size ($b' = B/b$).

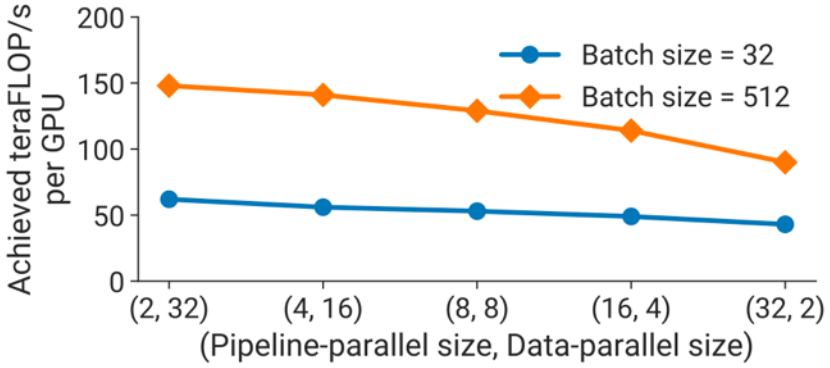


Figure 14: Throughput per GPU of various parallel configurations that combine data and pipeline model parallelism using a GPT model with 5.9 billion parameters, three different batch sizes, microbatch size of 1, and 64 A100 GPUs.

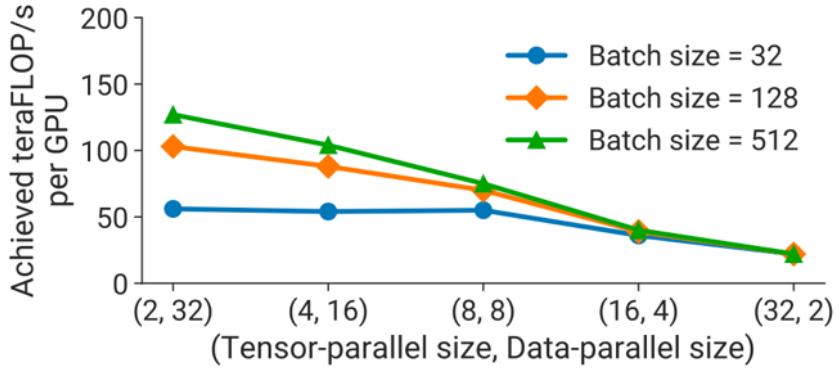


Figure 15: Throughput per GPU of various parallel configurations that combine data and tensor model parallelism using a GPT model with 5.9 billion parameters, three different batch sizes, microbatch size of 1, and 64 A100 GPUs.

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.

AI 计算模式思考 (IV)

1. **芯片间互连技术，提供 X00GB/s 带宽**
   1. 支持CPU+GPU 双架构，为大规模 AI 和HPC异构平台提供高速带宽
   2. 使能 NPU 同构芯片内超快片间互连技术
2. **专用高速 Transformer 引擎**
   - 大模型以Transformer为基础结构进行堆叠，高速的Transformer计算
   - 更低比特Transformer模块，并支持MoE构建万亿大模型

AI 计算模式思考 Summary

1. **网络模型结构支持 Architecture**
   - 支持高维的张量存储与计算
   - 神经网络模型的计算逻辑
2. **模型压缩(剪枝&量化) Model Compress**
   - 提供不同的 bit 位数
   - 利用硬件提供稀疏计算
3. **轻量化网络模型 Model Slim**
   - 复杂卷积计算（小型卷积核，e.g. 1x1 Conv）
   - 复用卷积核内存信息（Reuse Convolution）
4. **大模型分布式并行 Foundation Model**
   - 大内存容量、高速互联带宽
   - 专用大模型DSA IP模块，提供低比特快速计算

引用

1. https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks
2. https://machine-learning.paperspace.com/wiki/activation-function
3. https://developer.nvidia.com/blog/accelerating-ai-training-with-tf32-tensor-cores/