ZeRO: Memory Optimizations Toward Training Trillion Parameter Models

Background

- DNN model computation involves difference types of data:
- Activation
- Parameter
- Gradient
- Optimizer

Background

- Different parallelisms partitions different types of data

	Partitioned Data Type	Benefits	Drawbacks
Data Parallelism	Activation	Low communication volume	High memory volume
Model Parallelism	Parameter Gradient Optimizer	Low memory volume	High communication volume Small compute granularity
Pipeline Parallelism	Parameter Gradient Optimizer	Low communication volume	Imbalance memory volume Bubbles Latency cannot scale down

Background

- As models become larger, existing parallelisms are ineffective
- Data parallelism: size of data that cannot be partitioned exceeds the GPU memory
- Model parallelism: large models requires many GPUs and fine-grained partitioning, causing excessive communication overhead
- Pipeline parallelism: each layer is still too large; and with deeper layers, the devices handling the first few layers must hold more gradients/optimizers for the ongoing batches.

Challenge

- Reduce communication overhead
- Partition all types of data to save memory in a balanced way
- Preserve large operator size per device for TensorCore utilization

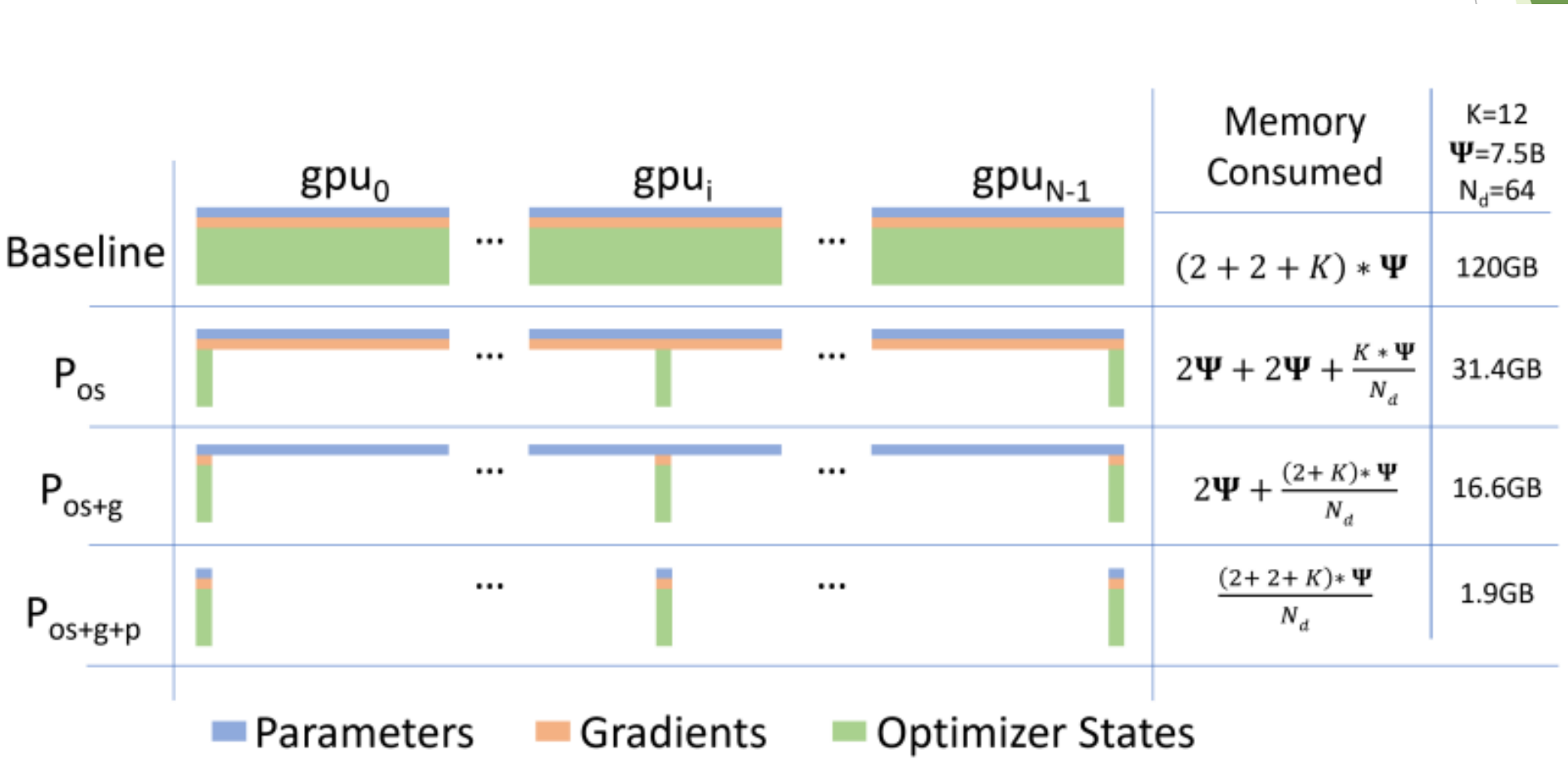
Observation

- In data/model parallelism, all types of data may be duplicated.
- In DNN model executions, most data stays idle when other layers are be computed.

- Thus, this data can be scatted into multiple devices and gathered when they are demanded

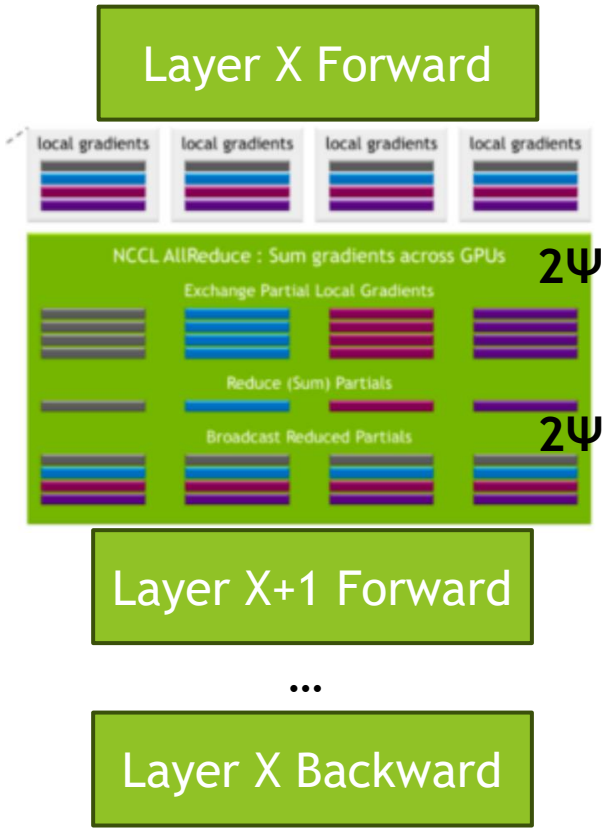
ZeRO-DP

Ψ : model size (number of parameters)
K : memory multiplier of optimizer states,
Nd : DP degree (number of devices)
2 : size of FP16



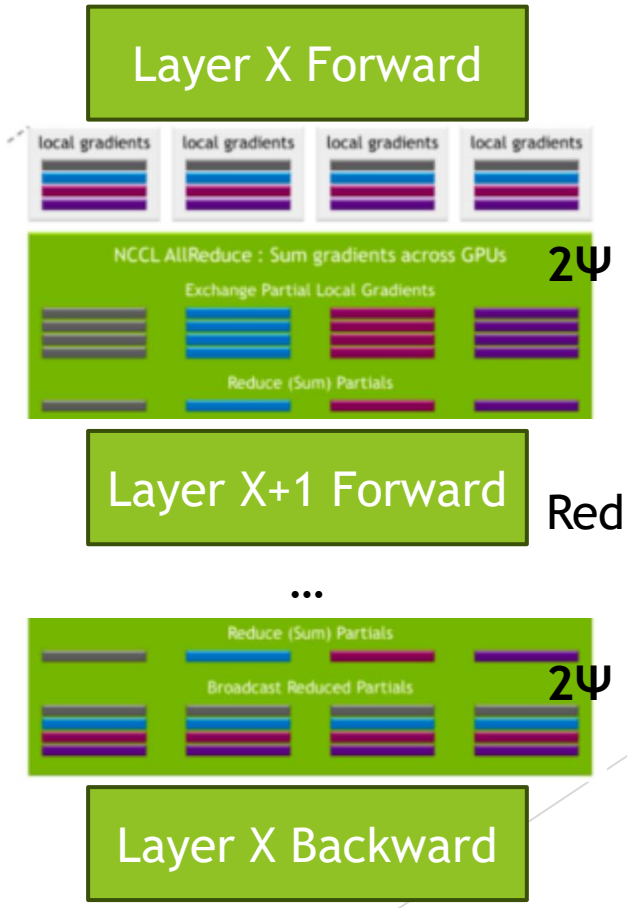
ZeRO-DP (Grad / OS)

Ψ : model size (number of parameters)
K : memory multiplier of optimizer states,
Nd : DP degree (number of devices)
2 : size of FP16



2Ψ Communication

2Ψ Communication



2Ψ Communication

Reduced memory footprint

2Ψ Communication

ZeRO-R / DP (Para)

Ψ : model size (number of parameters)
K : memory multiplier of optimizer states,
Nd : DP degree (number of devices)
2 : size of FP16

Layer X Forward

2*4Ψ Communication
(Megatron-LM + Zero-R)

Layer X Re-Comp

2*4Ψ Communication
(Megatron-LM + Zero-R)

…

Layer X Backward

2*4Ψ Communication
(Megatron-LM + Zero-R)

Layer X Forward

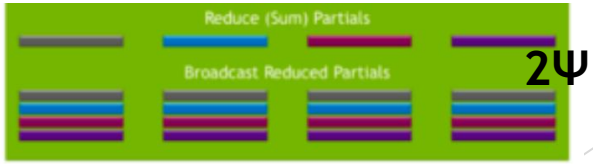
2*4Ψ Communication
(Megatron-LM + Zero-R)

Reduced memory footprint

Layer X Re-Comp

2*4Ψ Communication
(Megatron-LM + Zero-R)

…



2Ψ Communication

Layer X Backward

2*4Ψ Communication
(Megatron-LM + Zero-R)

Data Management

- Defragmentation
  - Short lived memory: discarded activation
  - Long lived memory: checkpointed activation, model states
  - Placing short-lived and long-lived memory together causes fragmentation
  - Solution: reserve contiguous memory for long lived memory
- Buffer size
  - To enable efficient reduction, Nvidia fuses all parameters into one buffer
  - Buffer of a large model can be too large for GPU memory
  - Solution: partition the buffer into fixed-sized buffers, which still have high efficiency in reduction

Evaluation

- Performance scales with number of GPUs