Microsoft DeepSpeed
[Zero: Memory Optimizations Toward Training Trillion Parameter Models]

CSE 5194.01
Autumn ‘20

Quentin Anthony
The Ohio State University
E-mail: anthony.301@osu.edu
Outline

• Introduction

• ZeRO Design

• DeepSpeed Results
Introduction: The State of Massive DL Models

- Computer vision models are no longer the largest!
- Just like for CNNs, increasing DL NLP accuracy requires (much) deeper models
- GPT-3 (175b) and T-NLG (17b) are the largest [designed for text generation]
- The memory limit of a single V100 GPU for NLP tasks is ~1 billion (depends on the batch size, DL framework, etc)
- Every emerging DL NLP model requires parallel training
Introduction: Why not use model parallelism?

- Model parallelism is great for a single node, but the communication volume is challenging for multi-node.

<table>
<thead>
<tr>
<th></th>
<th>Data Parallelism</th>
<th>Model Parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Efficiency</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td>Compute/Communication</td>
<td>Good</td>
<td>Poor</td>
</tr>
</tbody>
</table>

- Instead, why not bridge the gap between model and data parallelism?
  - Instead of replicating model states (parameters, gradients, and optimizer state) in data parallelism, partition them.
Outline

• Introduction

• ZeRO Design

• DeepSpeed Results
ZeRO Design: Overview

- ZeRO has three main optimization stages
  1. Optimizer State Partitioning ($P_{os}$): 4x memory reduction, same communication volume
  2. Add Gradient Partitioning ($P_{os+g}$): 8x memory reduction, same communication volume
  3. Add Parameter Partitioning ($P_{os+g+p}$): $(N_p)x$ memory reduction, 50% increase in communication volume

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Psi$</td>
<td># Params</td>
</tr>
<tr>
<td>$K$</td>
<td>Memory multiplier</td>
</tr>
<tr>
<td>$N_d$</td>
<td># CPUs/GPUs</td>
</tr>
</tbody>
</table>

- Old limit: GPU memory
- New limit: Aggregate memory
ZeRO Design: Details
ZeRO Design: What else?

- ZeRO also reduces the memory footprint of **activation** memory and **fragmented** memory.
ZeRO Design: What else?

- DeepSpeed also offers highly optimized transformer kernels along two lines:
  1. Alleviate memory access overhead
     - **Problem:** Transformer networks call many CUDA producer-consumer kernels, which leads to many read/write requests to memory
     - **Solution:** Use kernel fusion to reduce data movement, and distribute data processing among CUDA cores.
  2. Use transformer’s invertible operators to save memory and support large batch sizes
     - An operator is **invertible** if its backward pass is independent of the inputs and can be formulated based only on the outputs
     - Simply drop the inputs to reduce activation memory

- DeepSpeed also supports some miscellaneous designs for small improvements
  - i.e. asynchronous I/O, use of the pre-layer norm for higher learning rates, and reduction of redundant calculations at the output layer
Aside: ZeRO Implications and MP

• Instead of being limited by the **device** memory, we are now limited by the **aggregate** memory

• E.g. You want to train a trillion-parameter model on 1024 GPUs with 16 GB memory each
  – With 16-bit precision, model+optimizer = ~16 TB of memory
  – We can fit this into 1024 GPUs with ZeRO: $\frac{16 \text{ TB}}{1024 \text{ GPUs}} = 16 \frac{\text{GB}}{\text{GPU}}$

• Since ZeRO removes the DP memory limit, do we still need MP?
  – There are still models and data samples (e.g. pathology, astronomy, etc) that don’t fit inside GPU memory **even with ZeRO**
  – We can use basic pipeline MP with ZeRO for these cases, which provides ($\sim N_p \times N_m$) memory reduction, where $N_p$ and $N_m$ are the degree of data and model parallelism, respectively
Outline

• Introduction

• ZeRO Design

• DeepSpeed Results
DeepSpeed Results

System capability to efficiently train models with 170B parameters

- Largest Models
  - Zero-1: 10
  - Zero-2: 170

Model sizing (billion parameters)

Superlinear speedup of 60B parameter model with DeepSpeed

- Total Performance (TFlop)
  - 1024: 64
  - 2048: 128
  - 4096: 256
  - 8192: 512
  - 16384: 1024

Number of GPUs

- Performance/GPU: Observed Performance (TFlop)/Perfect Linear Scalability (TFlop)

Model Scale
- 170B parameter
- Order-of-magnitude bigger

Speed
- Up to 10X faster

Scalability
- Superlinear speedup

Usability
- Up to 13B without model parallelism

Up to 10x higher throughput for 100B parameter models

ZeRO Speedup

- PyTorch DP: 1.4
- Zero-1: 6
- Zero-2: 13

Powering 13B model training without model parallelism

Model sizing (billion parameters)
DeepSpeed Results

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>Number of V100 GPUs</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DGX-2</td>
<td>16</td>
<td>33 hr 13 min</td>
</tr>
<tr>
<td>4 DGX-2</td>
<td>64</td>
<td>8 hr 41 min</td>
</tr>
<tr>
<td>16 DGX-2</td>
<td>256</td>
<td>144 min</td>
</tr>
<tr>
<td>64 DGX-2</td>
<td>1,024</td>
<td>44 min</td>
</tr>
</tbody>
</table>

- DeepSpeed now holds the record for the fastest pre-training of BERT-Large using 1,024 V100 GPUs
- Previous record by NVIDIA: 47 min
- **Aside**: This is largely due to DeepSpeed’s transformer kernels
Conclusions

• ZeRO partitions the model and optimizer state to lessen the parameter bottleneck from GPU memory to the total aggregate GPU memory

• ZeRO targets HPC systems

• DeepSpeed allows the efficient training of massive CNN and NLP models
Thank You!

anthony.301@osu.edu

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

The MVAPICH2 Project
http://mvapich.cse.ohio-state.edu/

The High-Performance Deep Learning Project
http://hidl.cse.ohio-state.edu/