ZeRO-Offload: Democratizing Billion-Scale Model Training

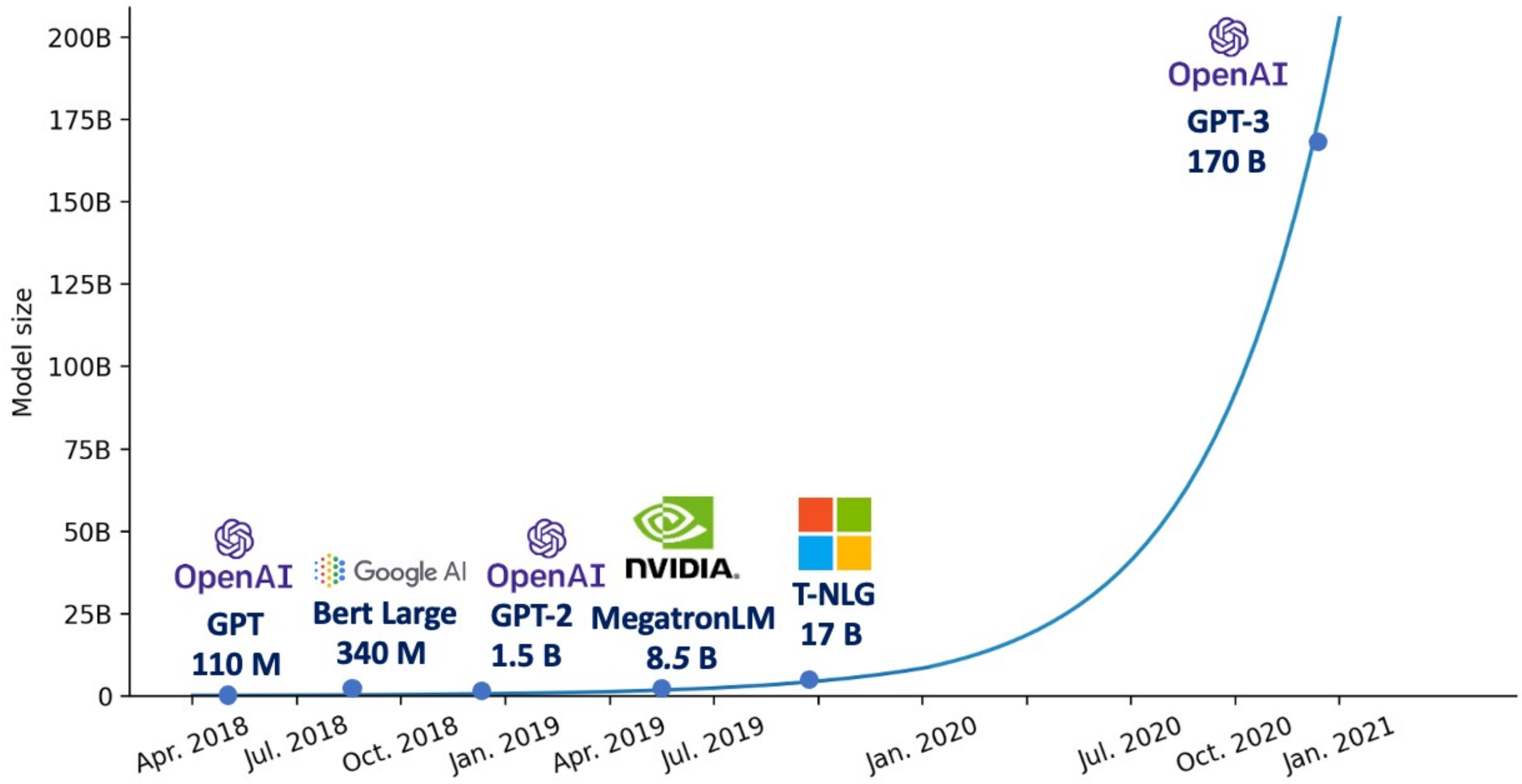
Jie Ren[*] Samyam Rajbhandari[†] Reza Yazdani Aminabadi[†] Olatunji Ruwase[†]

Shuangyan Yang[*] Minjia Zhang[†] Dong Li[*] Yuxiong He[†]

[*] University of California, Merced

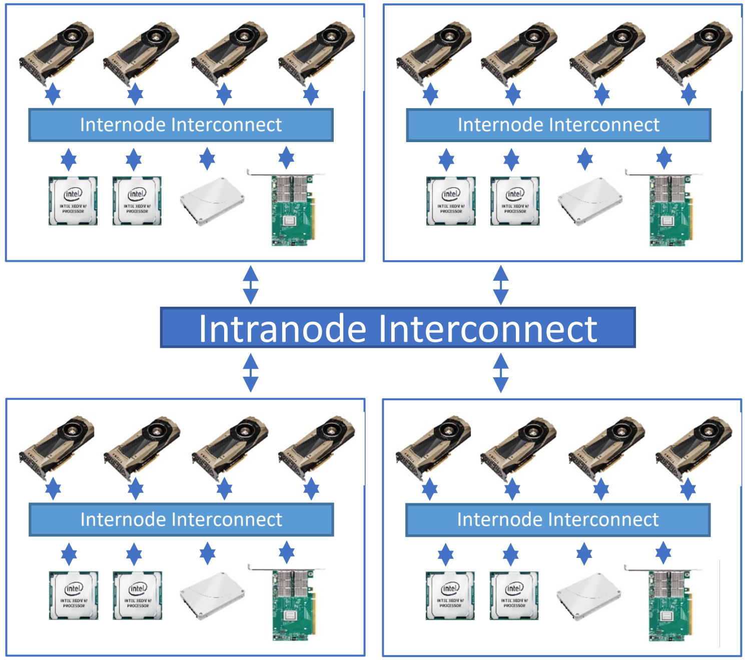
[†] Microsoft

The Size of Deep Learning Model is Increasing Quickly



Billon-Scale Model Training - Scale Out Large Model Training

- Model parallelism (Megatron-LM)
  - Partition the model states vertically across multiple GPUs.
- Pipeline parallelism (PipeDream, SOSP’19)
  - Partition the model states horizontally across layers.
- ZeRO: Zero Redundancy Optimizer (ZeRO, SC’20)
  - Split the training batch across multiple GPUs without model states duplication.



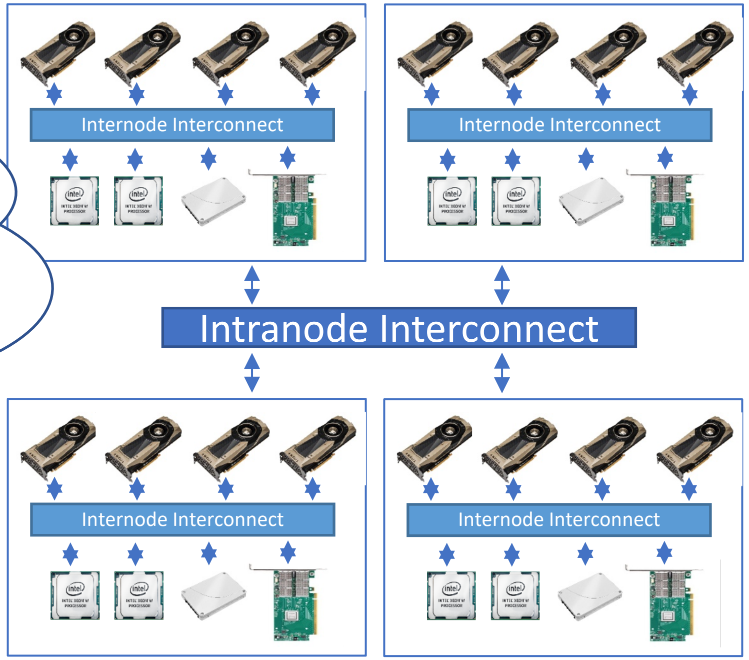
Distributed GPU Cluster

Billon-Scale Model Training - Scale Out Large Model Training

- Model parallelism (Megatron-LM)
    - Partition the model states vertically across multiple GPUs

Require having enough GPU devices

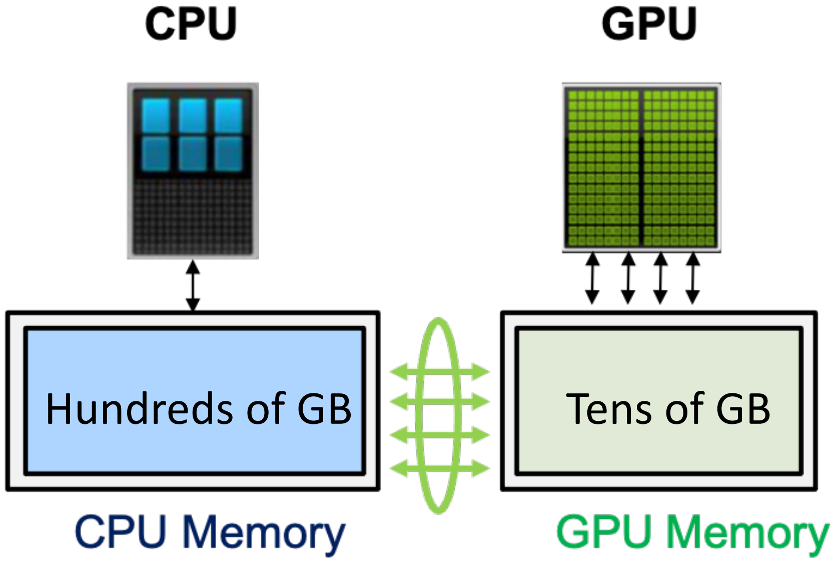
- ZeRO: Zero Redundancy Optimizer (ZeRO, SC'19)
    - Split the training batch across multiple GPUs without model states duplication.



Distributed GPU Cluster

Billon-Scale Model Training - Scale Up Large Model Training

- Heterogeneous DL training (SwapAdvisor, ASPLOS'20; Sentinel, HPCA'21; L2L)
  - Offload tensors from GPU memory to CPU memory when tensors are not used in computation.
  - Prefetch tensors from CPU memory to GPU memory before computation happens.



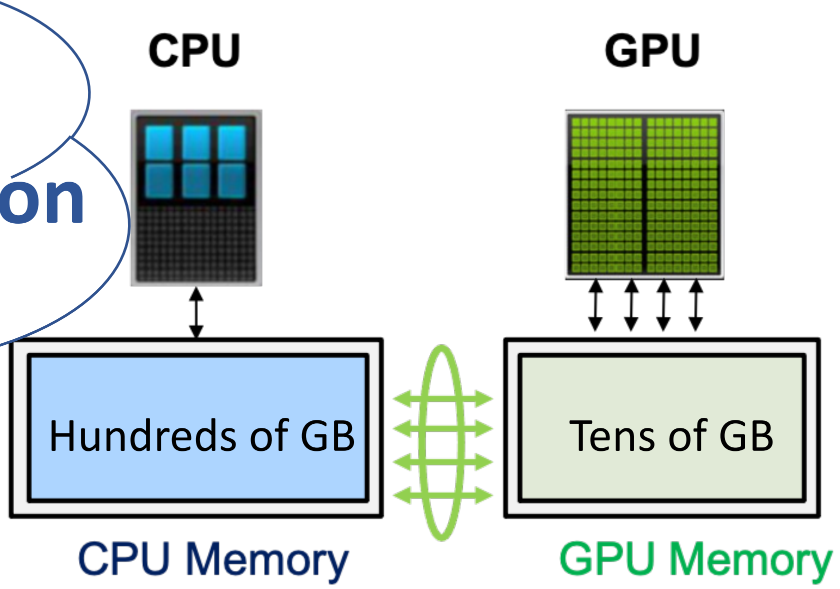
Billon-Scale Model Training - Scale Up Large Model Training

- Heterogeneous DL trainin Ad
  A

Only use CPU memory but not CPU computation

Designed for a single GPU

  - Tensor swapping sh verlap with computation as much as possible.



ZeRO-Offload: Democratizing Billion-Scale Model Training

Efficiency

- Enable 13B-parameter model training on a single NVIDIA V100 GPU at 40 TFLOPS.

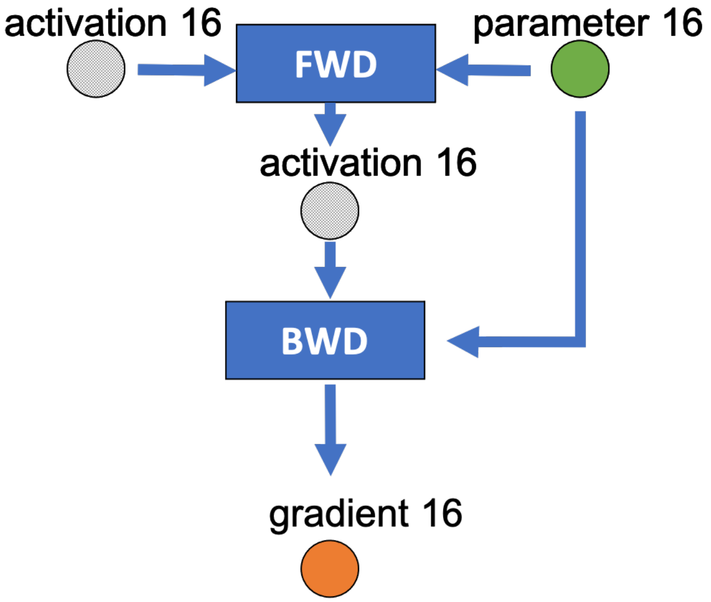
Scalability

- Achieve near perfect linear speedup with multiple GPUs.

Usability

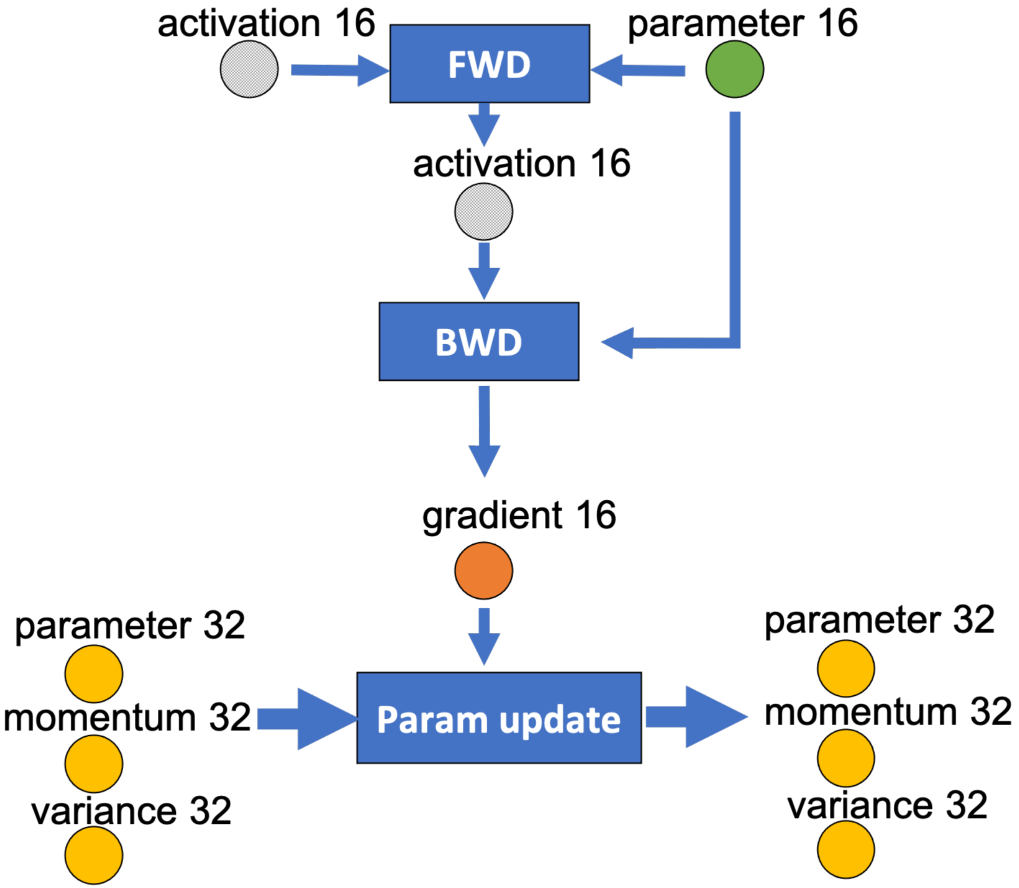
- Require no model refactoring.

Mixed Precision Training



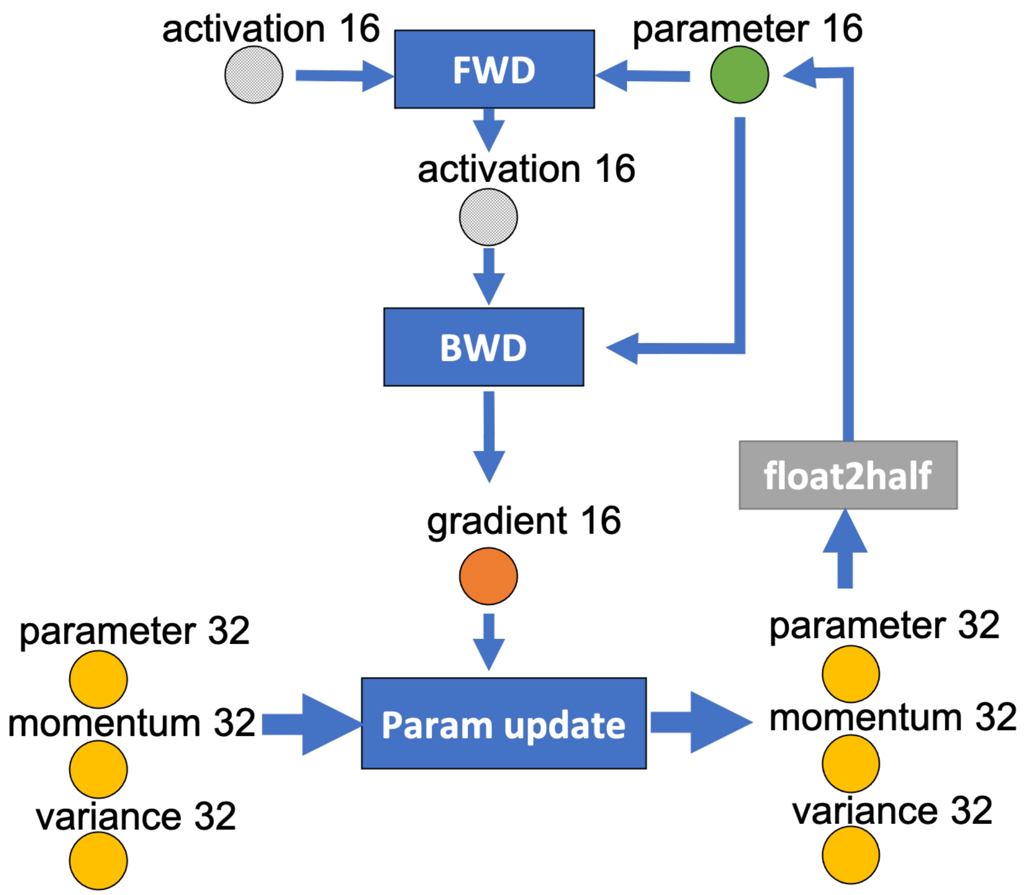
Mixed precision training iteration for a layer.

Mixed Precision Training



Mixed precision training iteration for a layer.

Mixed Precision Training



Mixed precision training iteration for a layer.

Offload Strategy

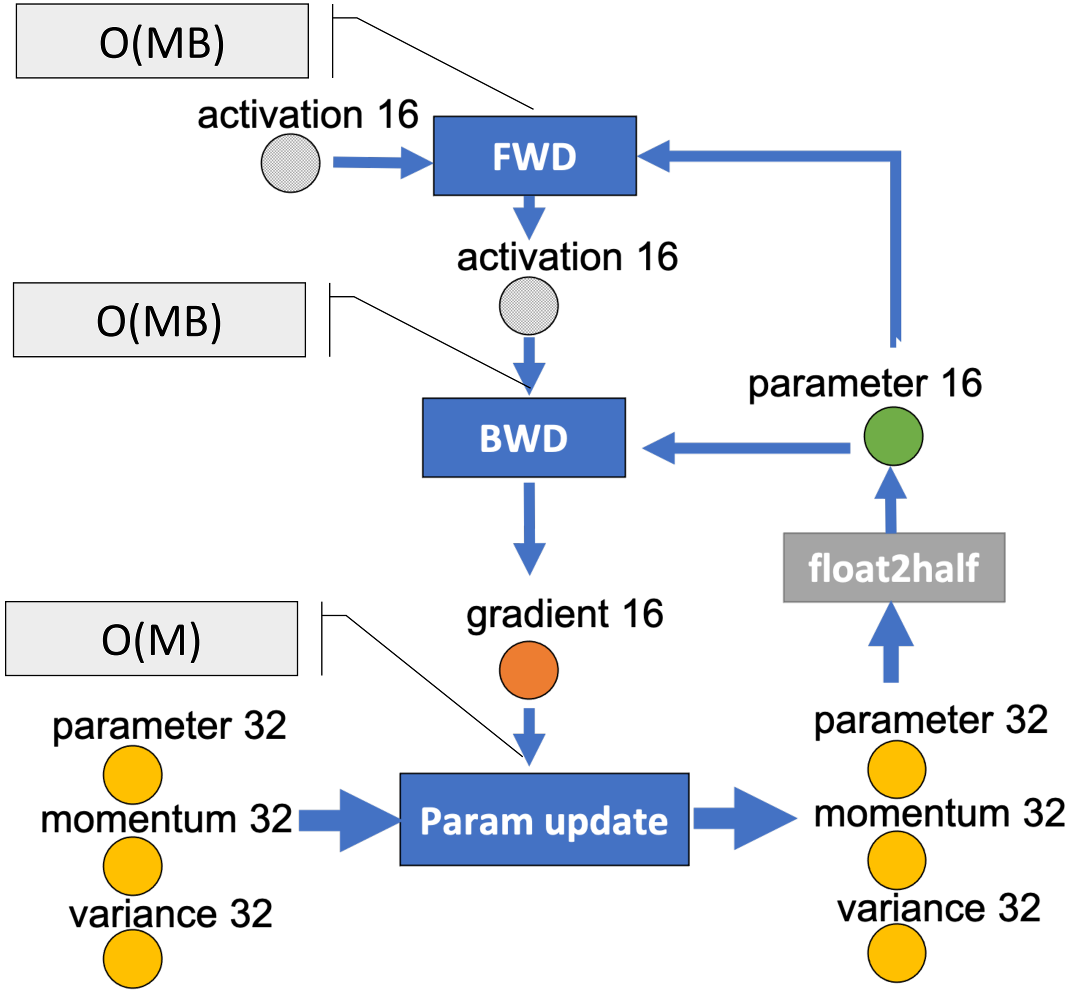
- ZeRO-Offload partitions the dataflow graph with:

  i. Few computation on CPU

  ii. Minimization of communication volume

  iii. Maximization of memory saving while achieving minimum communication volume

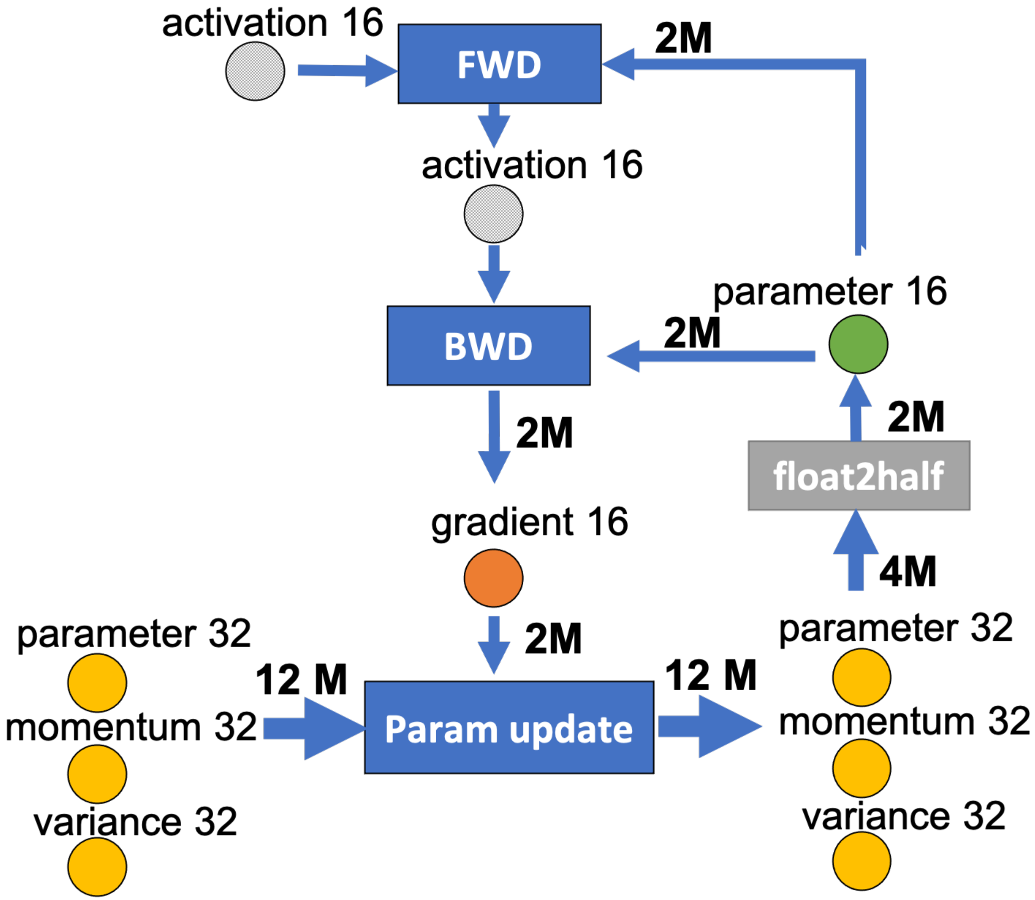
Limiting CPU Computation



Computational Complexity
O (MB)

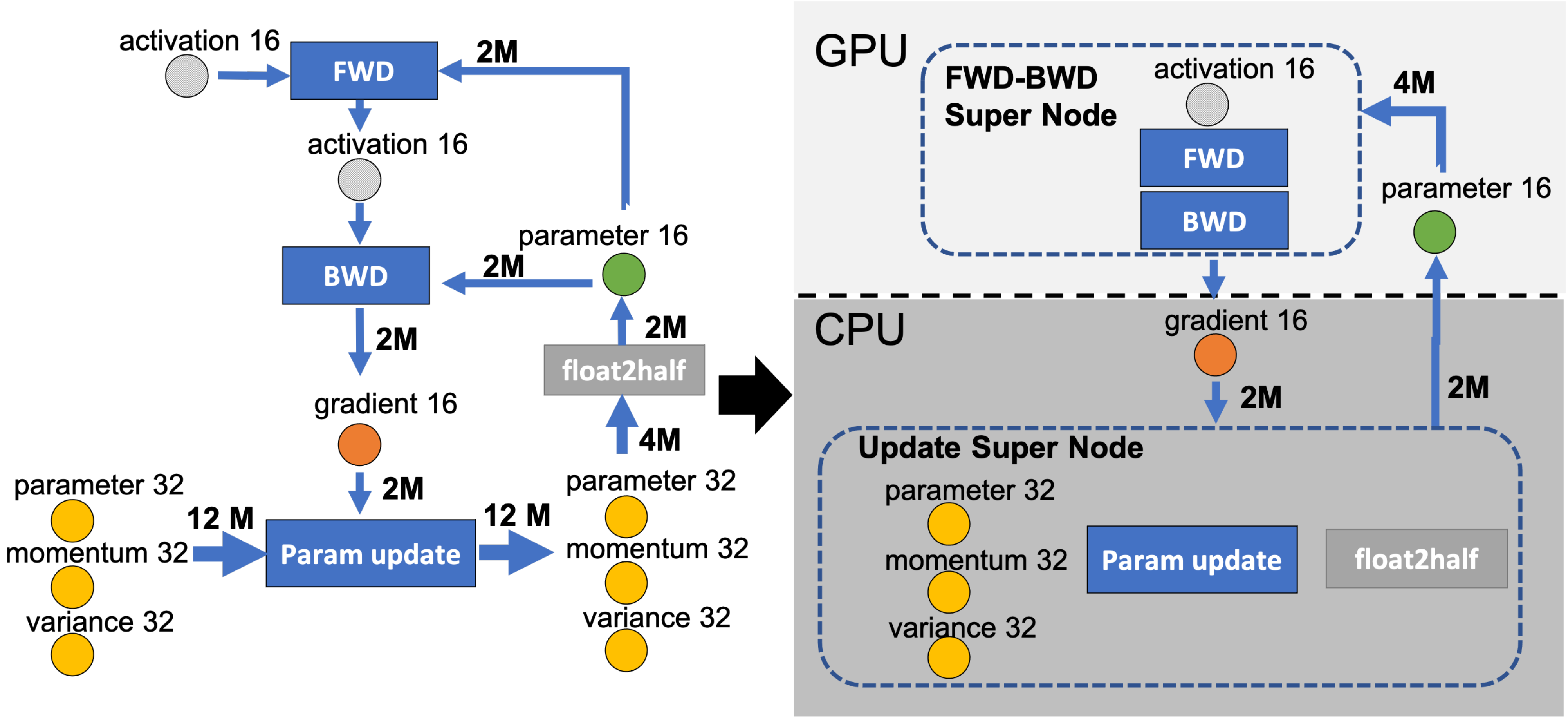
The dataflow of fully connected neural networks with M parameters.

Minimizing Communication Volume



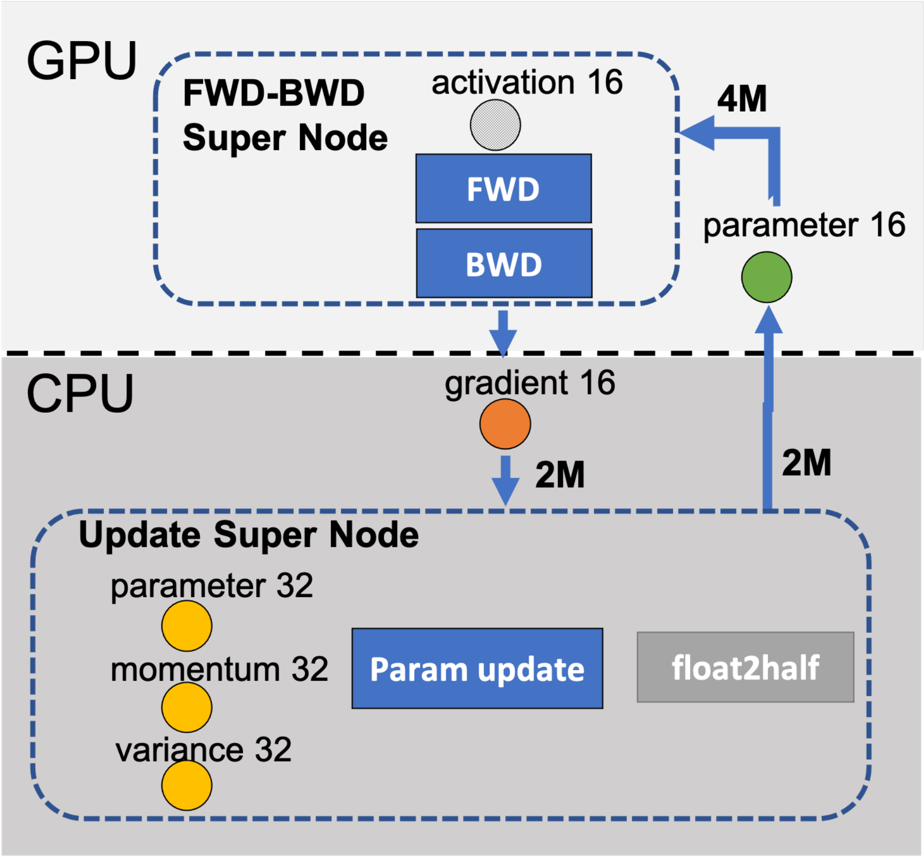
The dataflow of fully connected neural networks with M parameters.

ZeRO-Offload Enables Large Model Training by Offloading Data and Compute to CPU



Offloading fp16 gradients and updating super node on CPU

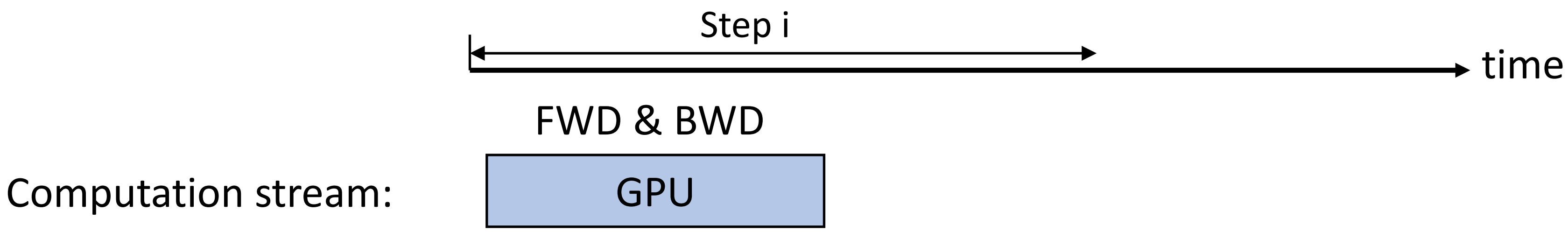
Unique Optimal Offload Strategy



FWD-BWD	param16	gradient16	Update	Memory	Reduction
GPU	GPU	GPU	GPU	16M	1x(baseline)
GPU	GPU	CPU	GPU	14M	1.14x
GPU	GPU	GPU	CPU	4M	4x
GPU	GPU	CPU	CPU	4M	8x

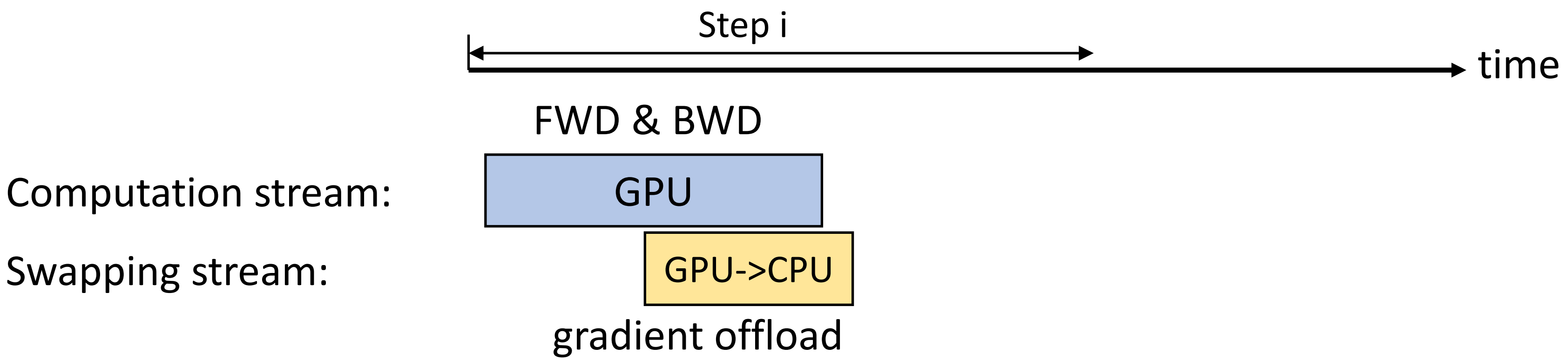
Memory saving for offload strategies that minimize communication volume compared to the baseline.

ZeRO-Offload Single GPU Schedule



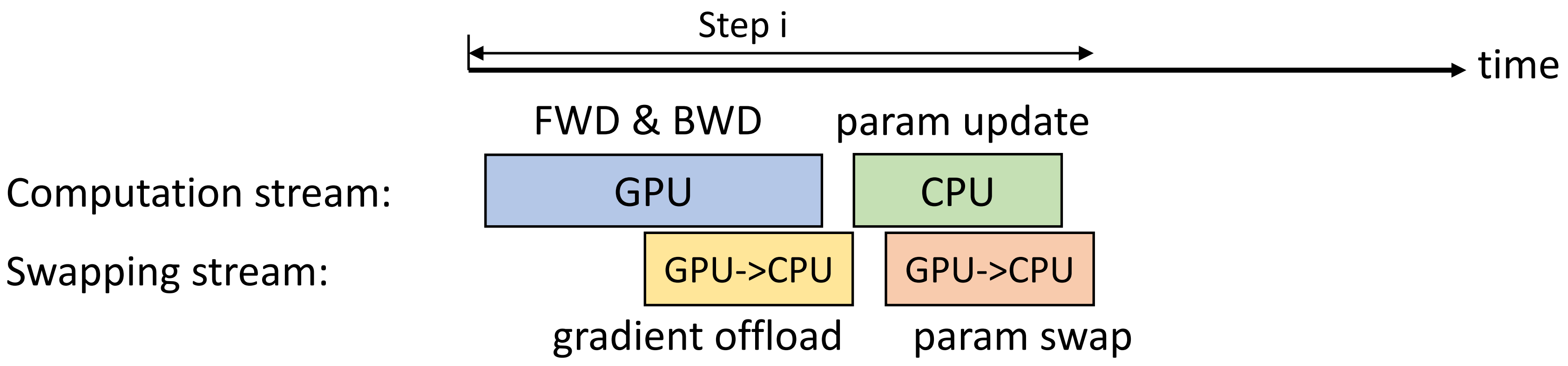
ZeRO-Offload training process on a single GPU.

ZeRO-Offload Single GPU Schedule



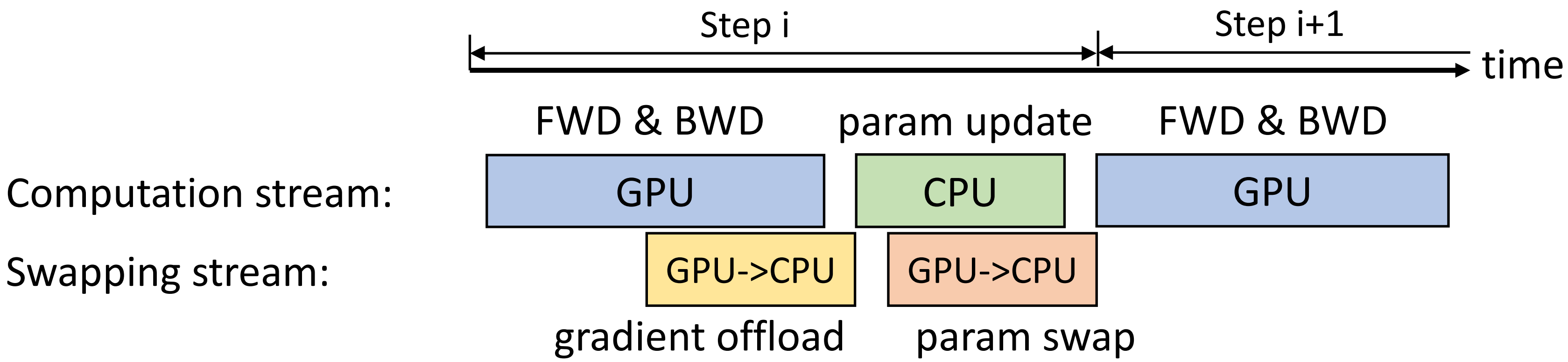
ZeRO-Offload training process on a single GPU.

ZeRO-Offload Single GPU Schedule



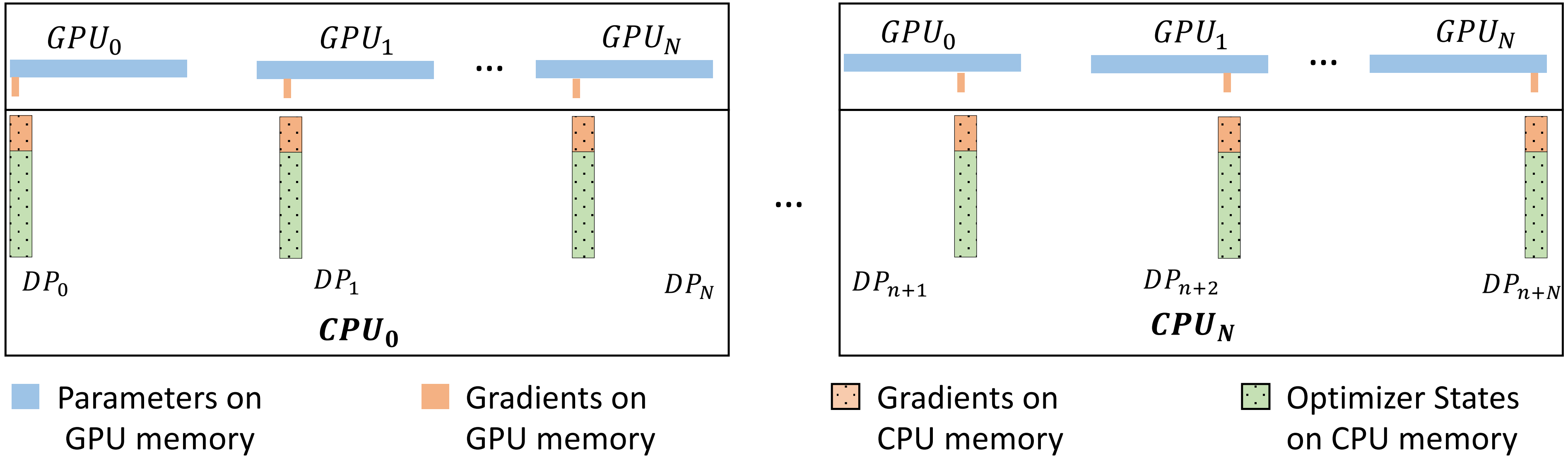
ZeRO-Offload training process on a single GPU.

ZeRO-Offload Single GPU Schedule



ZeRO-Offload training process on a single GPU.

ZeRO-Offload Multi-GPUs Schedule



Partitioning based on ZeRO[*] before offloading

[*] ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. SC'20

Optimized CPU Execution

- Highly parallelized CPU optimizer implementation

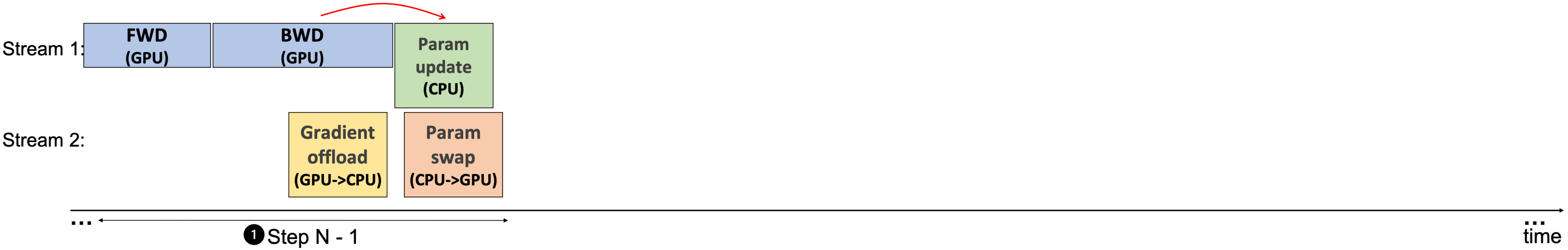
1) SIMD vector instruction for fully exploiting the hardware parallelism supported on CPU architectures.

2) Loop unrolling to increase instruction level parallelism.

3) OMP multithreading for effective utilization of multiple cores and threads on the CPU in parallel.

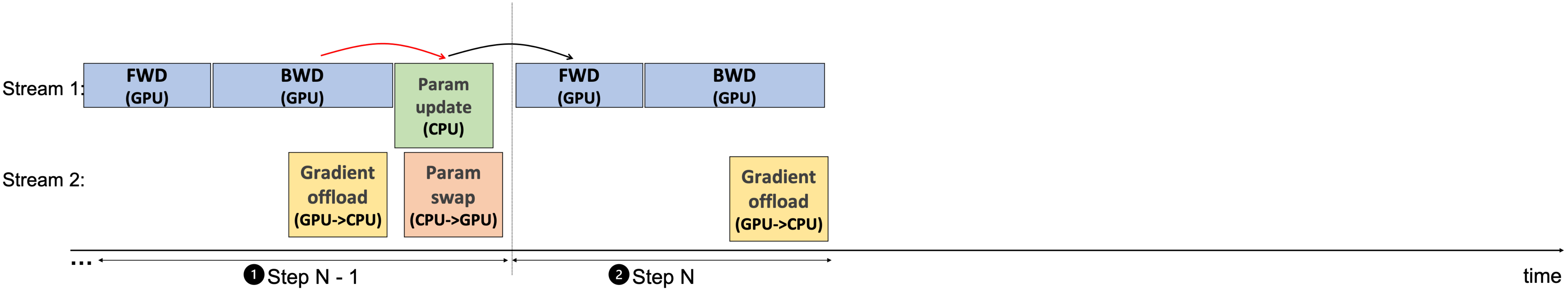
Optimized CPU Execution

- One-Step delayed parameter update



Optimized CPU Execution

- One-Step delayed parameter update


Stream 1:
FWD (GPU)
BWD (GPU)
Param update (CPU)
FWD (GPU)
BWD (GPU)
Stream 2:
Gradient offload (GPU->CPU)
Param swap (CPU->GPU)
Gradient offload (GPU->CPU)
...
❶ Step N - 1
❷ Step N
time