使用长序列训练 LLM 需要占用大量内存。为了在不大幅度降低训练速度的同时训练长序列的 LLMs，人们提出了很多方法来缓解内存压力。

然而这些方法存在三个个局限性。
（1）目前的方法在缩放序列长度和模型大小方面不够理想，GPU 内存和通信带宽可以进一步有效利用，以支持更长的序列长度和更大的模型尺寸。
（2）现有方法在训练性能方面不够高效。许多方法通过激活重计算或在设备间分配数据来减少所需的内存，这带来了不可避免的计算或通信开销。例如，DeepSpeed Ulysses 需要通过网络在不同服务器之间频繁通信，而网络带宽通常较低。
（3）现有的方法只支持有限的并行维度或者强依赖于模型和硬件的配置，导致次优的训练 performance。

我们研究了现有的四种提供序列并行的算法 MTP，MSP，FSP，DSP。

ISP 建模：

Search Space Analysis

Two-GPU Example

pattern of Using MTP, MSP, FSP

Comm model

1. 通信扩展因子 todo?
   i. **扩展因子的定义**：
   - 扩展因子是用来衡量分布式训练系统通信性能的一个比例。
   - 它由单个训练工作器（worker）执行预设数据量的通信任务所需的执行时间 T_1 与系统在有 N 个工作器处理相同训练任务时的执行时间 T_N 之比来定义。

   ii. **扩展因子的计算**：
   - 扩展因子的计算公式为：扩展因子 = $\frac{T_1}{T_N \times N}$。
   - 例如，如果一个给定的训练任务，1个工作器需要9小时完成，而8个工作器只需1.25小时完成，则8个工作器的系统扩展因子为 $\frac{9}{1.25 \times 8} = 0.9$。
2. Comm. & Comp. Modeling
   i. Comm.
   - 旧算法使用环形方法，其中每个进程的数据被发送到一个虚拟的进程环中。
   - 在第一步中，每个进程 i 向进程 $i+1$ 发送数据并从进程 $i-1$ 接收数据（有环绕）。
   - 从第二步开始，每个进程 i 向进程 $i+1$ 转发它从进程 $i-1$ 在上一步中接收到的数据。
   - 如果 p 是进程的数量，整个算法需要 $p-1$ 步。如果 n 是每个进程要收集的总数据量，则在每一步中每个进程发送和接收 n/p 数据。
   - $T_{\text{ring}} = (p-1)\alpha + \frac{(p-1)n\beta}{p}$。
   - $T_{\text{tree}} = \log(p)\alpha + \frac{(p-1)n\beta}{p}$

Model states Comm. modeling.

P_{para},P_{grads},P_{os}: we use P_x define the number of GPUs partitioning the member x
n: number of GPU in world size
M: number of parameters in model
Dt: datatype of parameters/gradient/optimizer states stored
BW: bandwidth, when $P_x > 8$, BW=IB network,else BW=Nvlink
Gn: number of gradient accumulation steps. It equal to $Gn = \frac{GlobalBatchSize}{n \times MicroBatchSize}$

1. $T_{\text{Comm}}(P_{para}, ISP, allgather) = 2[(P_{para}-1)\alpha + \frac{(P_{para}-1)MDt}{P_{para}BW}\beta]$。
2. $T_{\text{Comm}}(P_{grad}, ISP, reducescatter) = Gn[(P_{grad}-1)\alpha + \frac{(P_{grad}-1)MDt}{P_{grad}BW}\beta]$。
3. $T_{\text{Comm}}(P_{grad}, ISP, broadcast) = 3[(P_{os}-1)\alpha + \frac{(P_{os}-1)MDt}{P_{os}BW}\beta]$。

All2All Comm. modeling.

all2all communication occur before and after the computation of attention related to sequence parallism. b,s,h present MicroBatchSize,SquenceLength and HiddenSize repsectively.

all2all is a point to point communication, inter and intra node communication are independent. Therefore, when $SP > 8$,we should model the inter and intra node communication seperately corelated to the specific communication traffic.

IF SP<8:
$T_{\text{Comm}}(SP, ISP, all2all) = (SP-1)\alpha + \frac{4(SP-1)bshDt}{SP \times BW}\beta$。

IF SP>8: TODO

Model states Comp. modeling.

1. Compute attention block:
   Calculate Q, K, V: $3 \times [b,s,h] \times [h,h] = 6bsh^2$
   QK^T matrix multiplication: $[b,a,s,h] \times [b,a,h] = 2bas^2$
   Score dot V: $[b,a,s,h] \times [b,a,h] = 2bas^2$
   Post attention: $[b,s,h] \times [h,h] = 2bsh^2$
2. Compute mlp block:
   First linear layer: $\times [b,s,h] \times [h,4h] = 8bsh^2$
   Second linear layer: $\times [b,s,4h] \times [4h,h] = 8bsh^2$

$T_{\text{Comp}}(SP, b, C_{qkv}, Gemm) = Gemm(6bh^2 \times \frac{s}{SP})$。
$T_{\text{Comp}}(SP, b, C_{qkT+ScoreV}, FlashAttn) = FlashAttn(4bh^2 \times \frac{s}{SP})$。
$T_{\text{Comp}}(SP, b, C_{PostAttn}, Gemm) = Gemm(2bh^2 \times \frac{s}{SP})$。
$T_{\text{Comp}}(SP, b, C_{L1}, Gemm) = Gemm(8bh^2 \times \frac{s}{SP})$。
$T_{\text{Comp}}(SP, b, C_{L2}, Gemm) = Gemm(8bh^2 \times \frac{s}{SP})$。

Overlap Modeling

$T_{\text{Comm}}(Total) = T_{\text{Comm}}(P_{para}, ISP, allgather) + T_{\text{Comm}}(P_{grad}, ISP, reducescatter) + T_{\text{Comm}}(P_{grad}, ISP, broadcast) + T_{\text{Comm}}(SP, ISP, all2all)$。

$T_{\text{Comp}}(Total) = T_{\text{Comp}}(SP, b, C_{qkv}, Gemm) + T_{\text{Comp}}(SP, b, C_{qkT+ScoreV}, FlashAttn) + T_{\text{Comp}}(SP, b, C_{PostAttn}, Gemm) + T_{\text{Comp}}(SP, b, C_{L1}, Gemm) + T_{\text{Comp}}(SP, b, C_{L2}, Gemm)$。

$T_{\text{layer}} = \max(T_{\text{Comm}}(Total), T_{\text{Comp}}(Total))$。