



# 智能计算系统

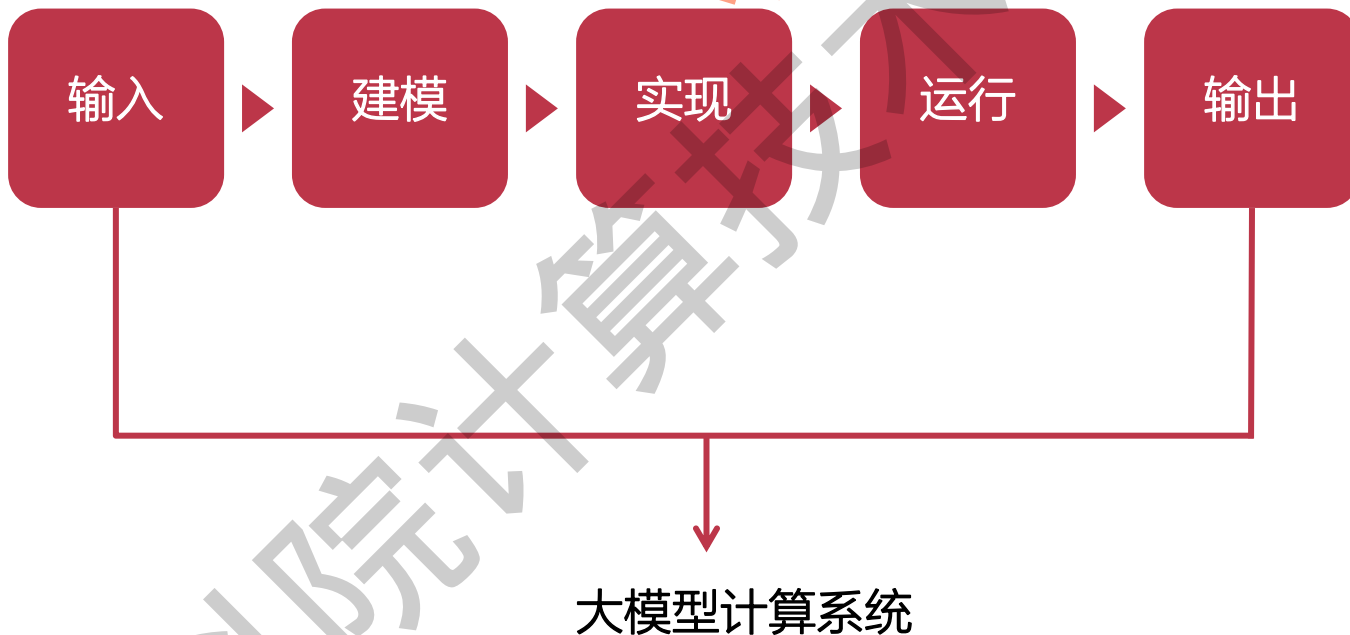
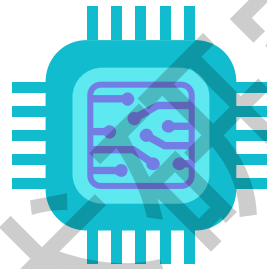
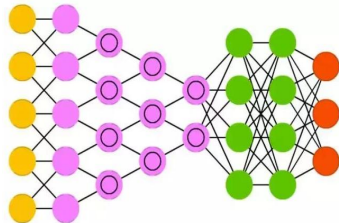
## 第九章 大模型计算系统

中国科学院计算技术研究所

李威 副研究员

liwei2017@ict.ac.cn

# 本章内容定位

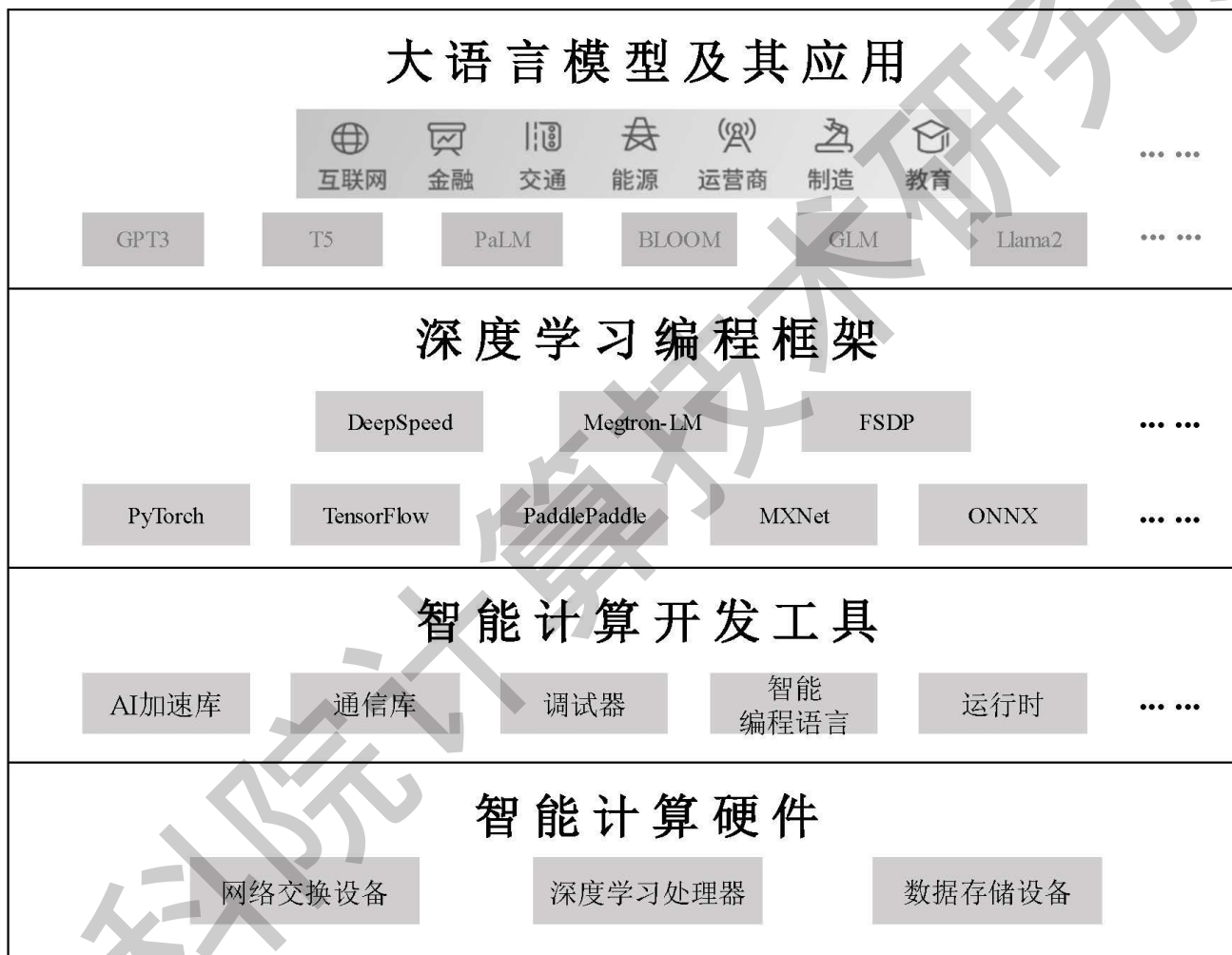


将前面各章介绍的智能算法、编程框架、芯片架构、编程语言等内容串联起来，使读者能真正融会贯通，从而全面地理解智能计算系统。

# 提纲

- ▶ 本章概述
- ▶ 大模型算法分析
- ▶ 大模型驱动范例：BLOOM
- ▶ 大模型系统软件
- ▶ 大模型基础硬件
- ▶ 本章小结

# 本章概述

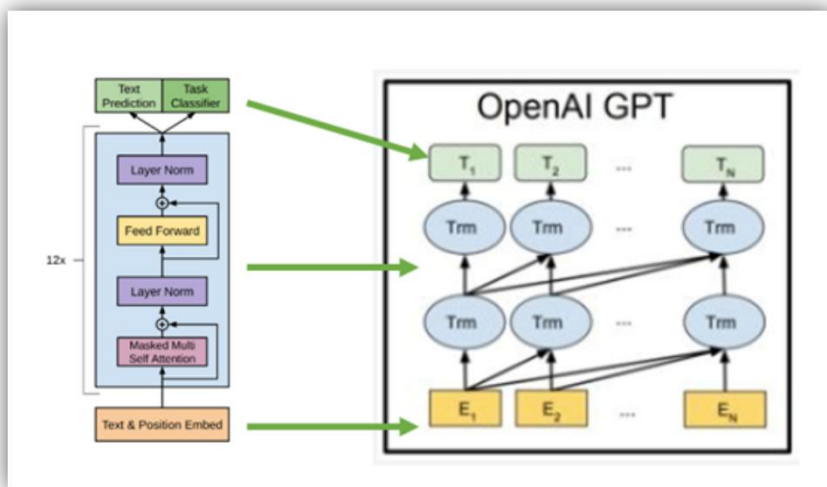


大模型计算系统的整体架构

# 提纲

- ▶ 本章概述
- ▶ 大模型算法分析
- ▶ 大模型驱动范例：BLOOM
- ▶ 大模型系统软件
- ▶ 大模型基础硬件
- ▶ 本章小结

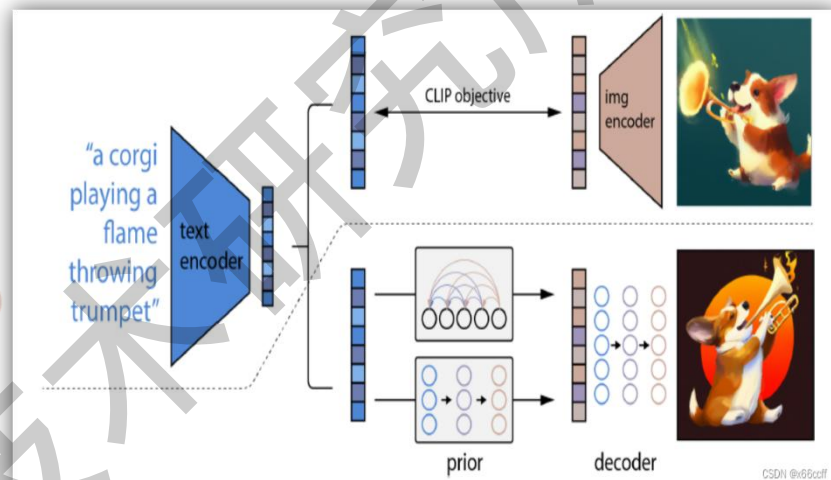
# 大语言模型和多模态大模型的比较



大语言模型

- 通过对自然语言进行建模得到概率模型来预测某个位置的词序列的概率
- 从早期的统计语言模型发展到如今最受关注的大语言模型
- 在各类自然语言理解与生成任务中表现出了强大的能力

VS

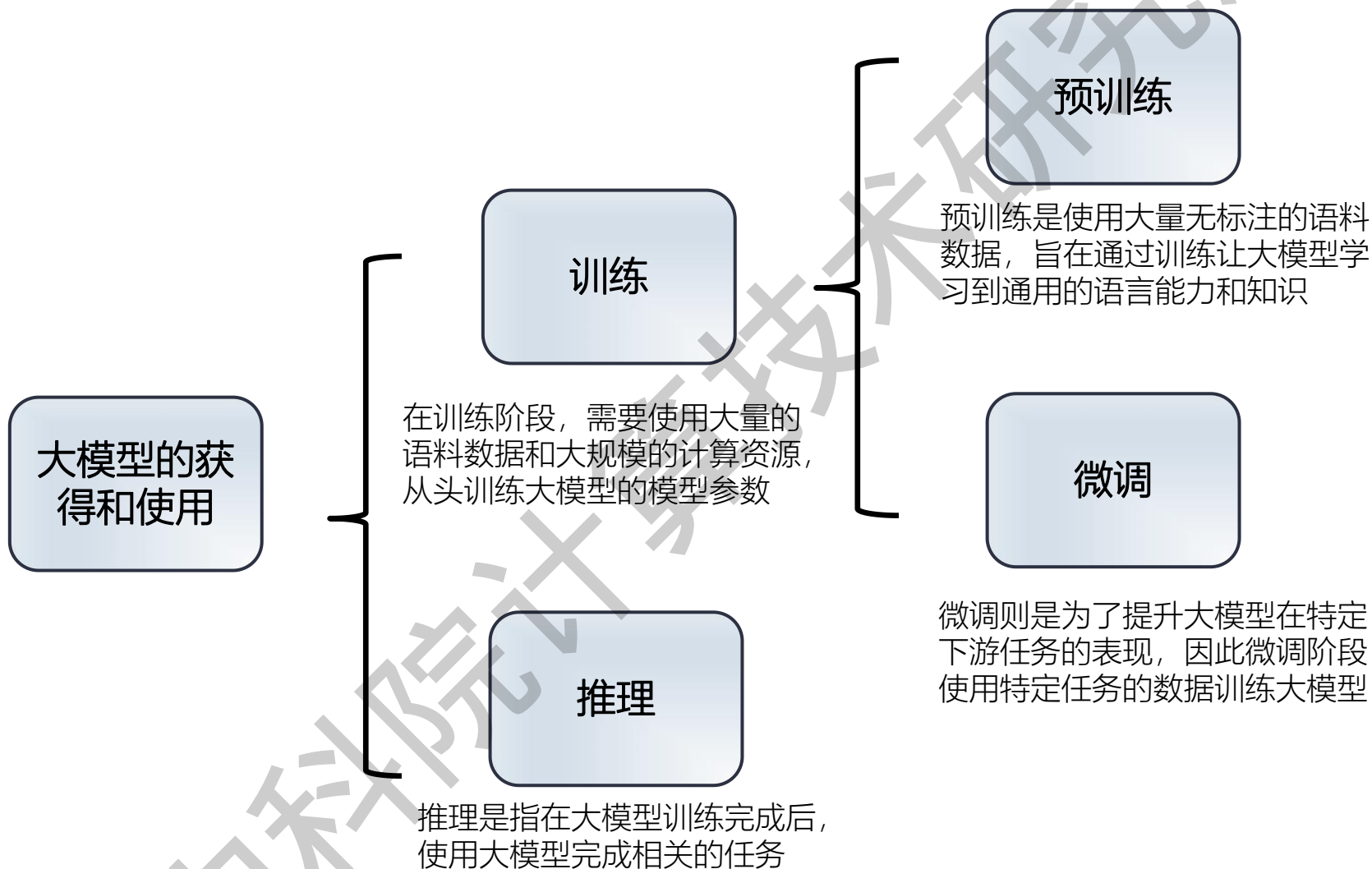


多模态大模型

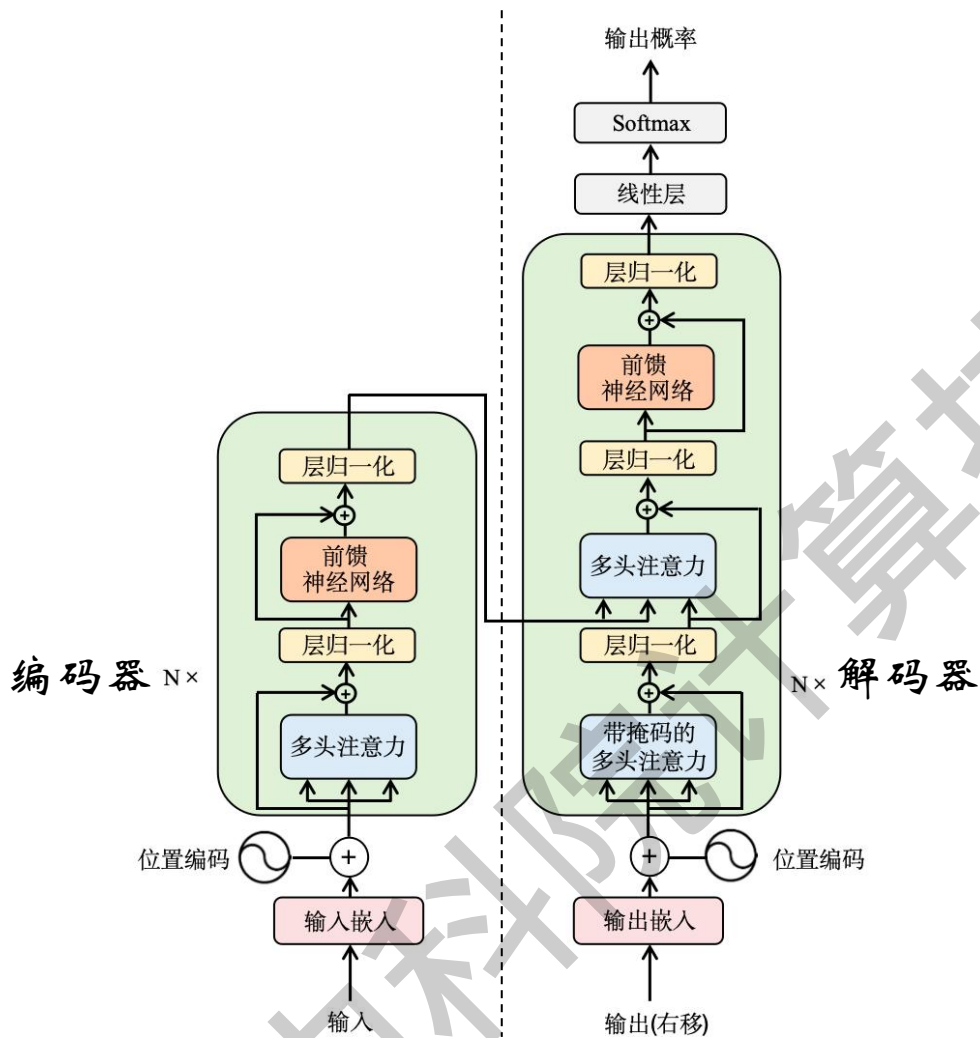
- 在文本数据的基础上将图像、语音等多种模态数据也引入大模型
- 重点是将多种模态数据对齐后进行信息交互和融合
- 将语言对应的文本数据作为标杆、将多种模态数据与文本数据进行对齐，是实现多模态大模型的高效而实用的手段

大语言模型是多模态大模型的基础

# 大模型的获得和使用

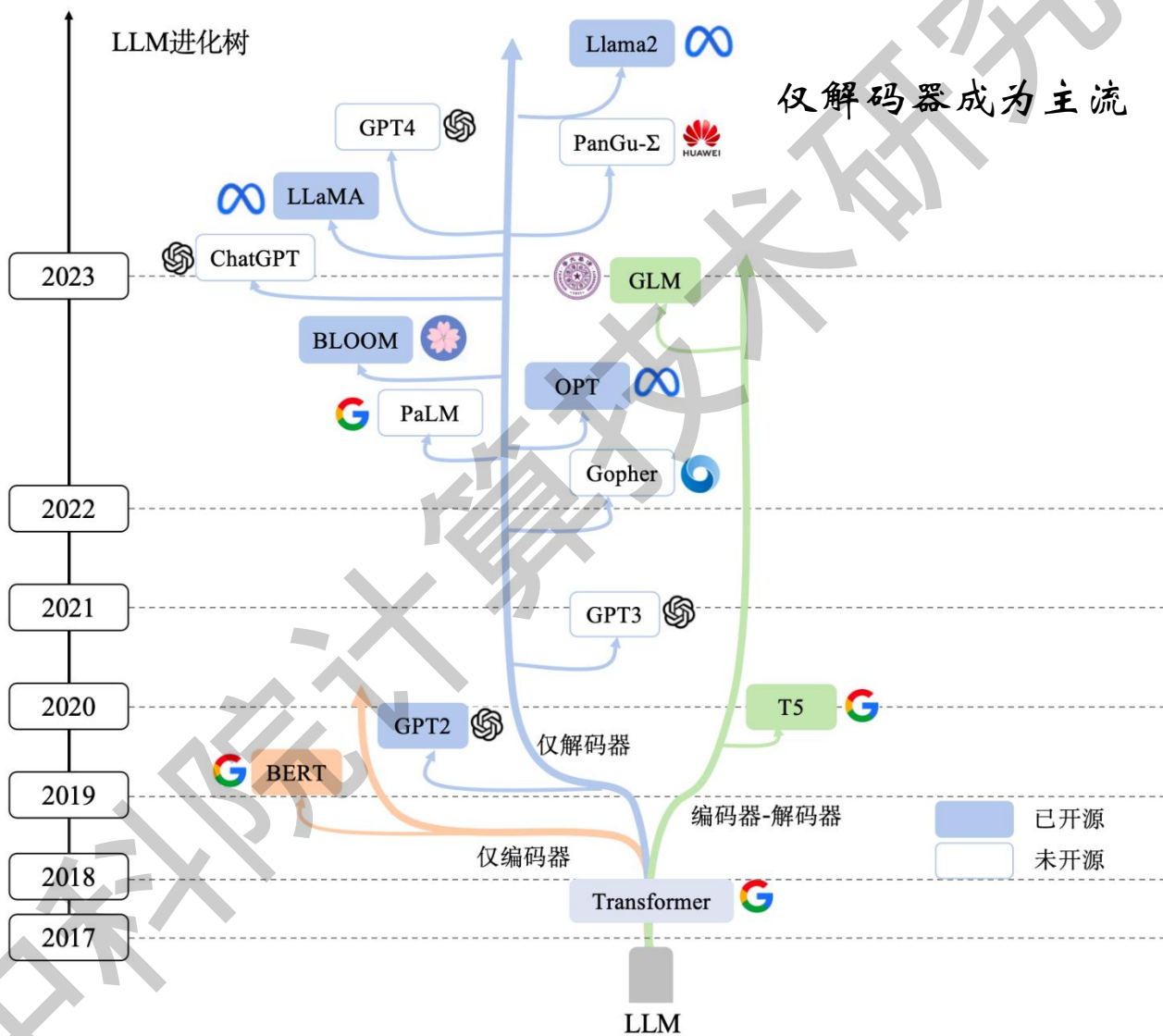


# 大模型算法分类



- ▶ 仅编码器架构
- ▶ 编码器-解码器架构
- ▶ 仅解码器架构

# 大模型算法发展历程



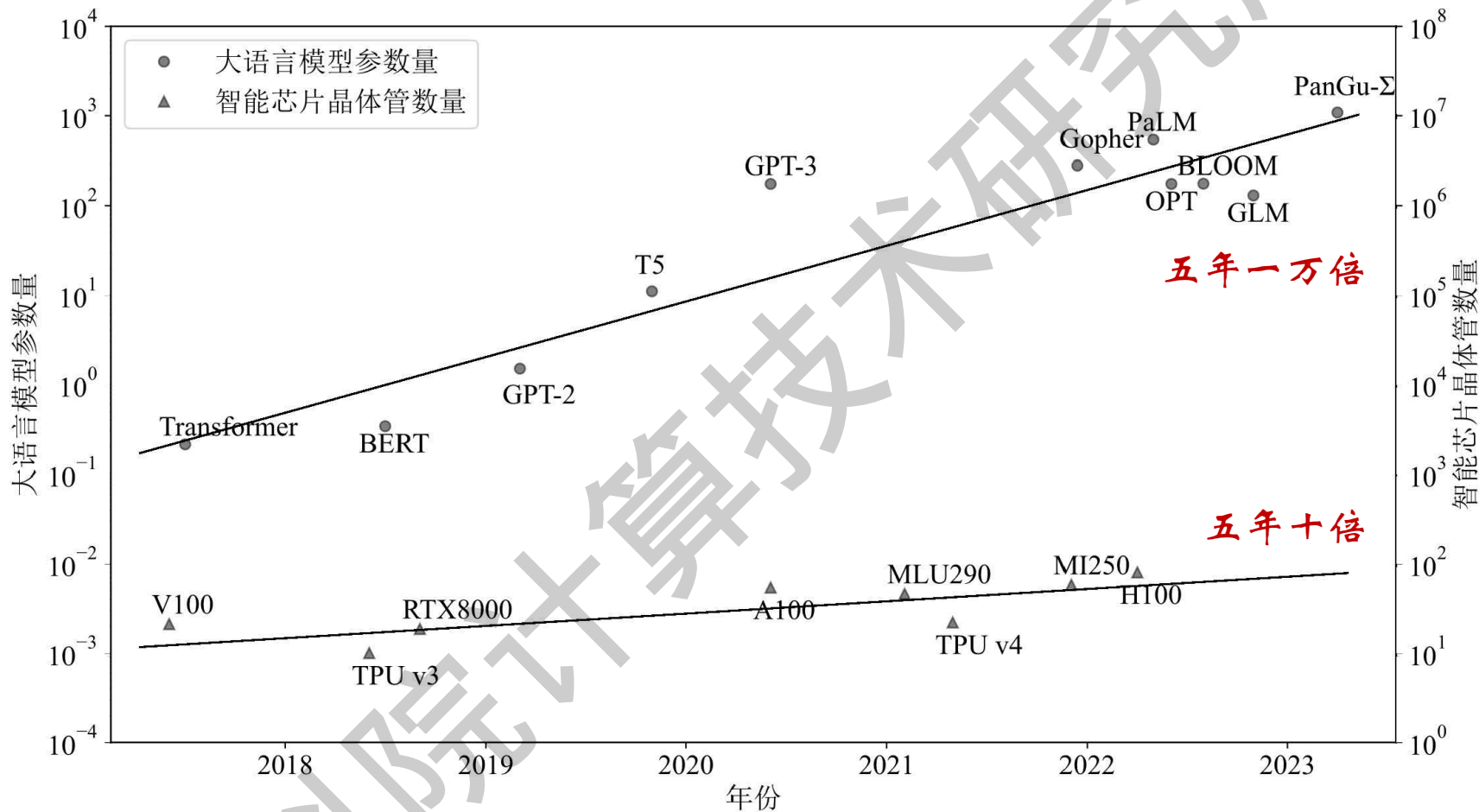
# 大模型算法计算需求

模型名称	发布机构	发布时间	开源	结构	参数量	预训练数据集规模	训练资源	训练时间
Transformer <sup>[45]</sup>	Google	2017.6	是	编码器-解码器	213M	15B 词元	8 P100	3.5 天
BERT <sup>[150]</sup>	Google	2018.10	是	仅编码器	340M	128B 词元	64 TPU	4 天
GPT-2 <sup>[363]</sup>	OpenAI	2019.2	是	仅解码器	1.5B	40 GB <sup>o</sup>	-	-
T5 <sup>[193]</sup>	Google	2019.10	是	编码器-解码器	11B	1T 词元	1024 TPU v3	-
GPT-3 <sup>[34]</sup>	OpenAI	2020.5	否	仅解码器	175B	300B 词元	-	-
Gopher <sup>[364]</sup>	DeepMind	2021.12	否	仅解码器	280B	300B 词元	4096 TPU v3	920 小时
PaLM <sup>[194]</sup>	Google	2022.4	否	仅解码器	540B	780B 词元	6144 TPU v4	-
OPT <sup>[365]</sup>	Meta AI	2022.5	是	仅解码器	175B	180B 词元	992 80G A100	-
BLOOM <sup>[216]</sup>	BigScience	2022.7	是	仅解码器	176B	366B 词元	384 80G A100	105 天
GLM <sup>[197]</sup>	清华大学	2022.10	是	编码器-解码器	130B	400B 词元	768 40G A100	60 天
ChatGPT <sup>[152]</sup>	OpenAI	2022.11	否	仅解码器	-	-	-	-
LLaMA <sup>[195]</sup>	Meta AI	2023.2	是	仅解码器	65B	1.4T 词元	2048 80G A100	21 天
GPT-4 <sup>[46]</sup>	OpenAI	2023.3	否	仅解码器	-	-	-	-
PanGu- $\Sigma$ <sup>[35]</sup>	华为	2023.3	否	仅解码器	1085B	329B 词元	512 Ascend 910	100 天
Llama2 <sup>[196]</sup>	Meta AI	2023.7	是	仅解码器	70B	2T 词元	2000 80G A100	36 天

DeepMind: 训练所需计算量正比于参数规模 $\times$ 数据集规模

- 模型随着其参数量和训练数据规模的不断发展, 其所需的训练资源越来越多, 意味着对智能计算系统的需求也越来越复杂。

# 大模型与智能计算芯片发展



● 需要从智能计算系统的软硬件层面进行系统的优化设计，实现高效的大模型计算系统。

# 提纲

- ▶ 本章概述
- ▶ 大模型算法分析
- ▶ 大模型驱动范例：BLOOM
- ▶ 大模型系统软件
- ▶ 大模型基础硬件
- ▶ 本章小结

# 大模型范例 vs 风格迁移范例

- B L O O M 全名为 BigScience Large Open-science Open-access Multilingual Language Model
- 是由BigScience研究团队于2022年7月推出的开源大模型

**BLOOM模型**

模型参数量：  
>1200倍

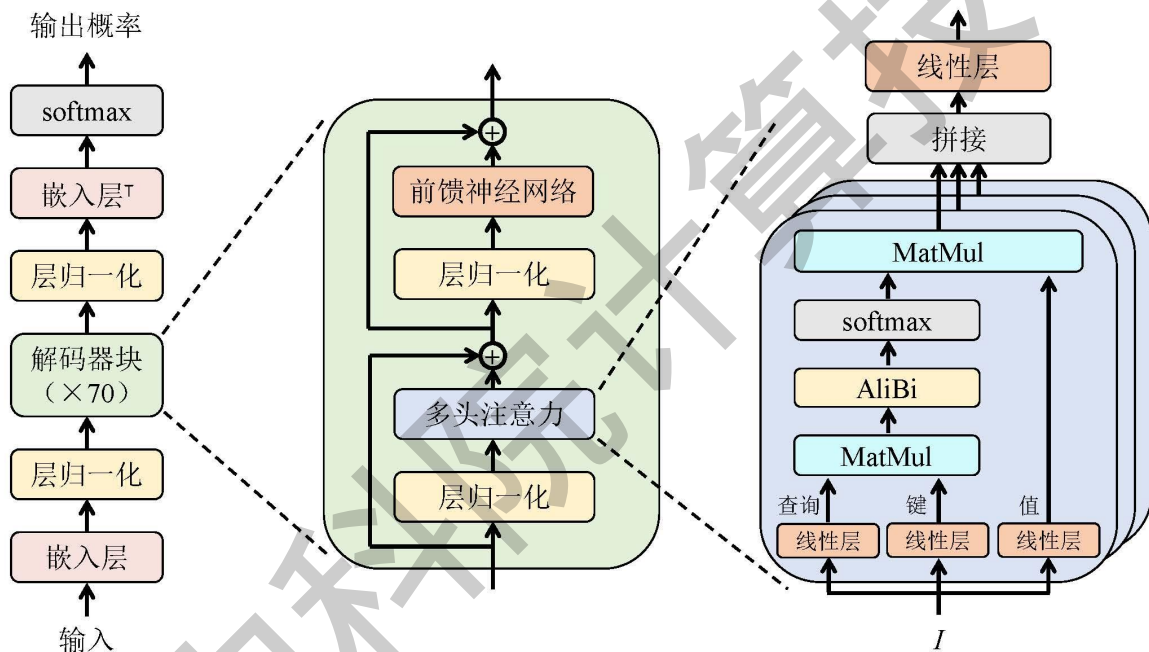
推理算力需求：  
>10000倍

风格迁移范例模型

大模型驱动范例与风格迁移驱动范例对比

# BLOOM-176B 模型结构

- BLOOM模型包含1760亿个参数（下称BLOOM-176B模型），使用ROOTS数据集训练。
- BLOOM-176B 模型主要包括 70个解码器块。
- ROOTS语料库是由BigScience研究团队提出的开源语料库，由498个数据集的组成，包含46种自然语言和13种编程语言在内共59种语言，总共1.61TB文本。
- 文本数据经过分词器进行分词后，可以转化为1660亿（166B）个词元用于BLOOM-176B模型的训练。



参数	含义	值
$N$	模型参数量	176B
$l$	解码器层数	70
$h$	总隐藏层维度	14336
$n$	多头注意力数目	112
$d$	注意力隐藏层维度	128
$s$	序列长度	2048
$B$	全局批量大小	2048
$b$	微批量大小	2

(a) BLOOM-176B 整体结构图

(b) 解码器块结构图

(c) 多头注意力结构图

# BLOOM-176B 运行平台

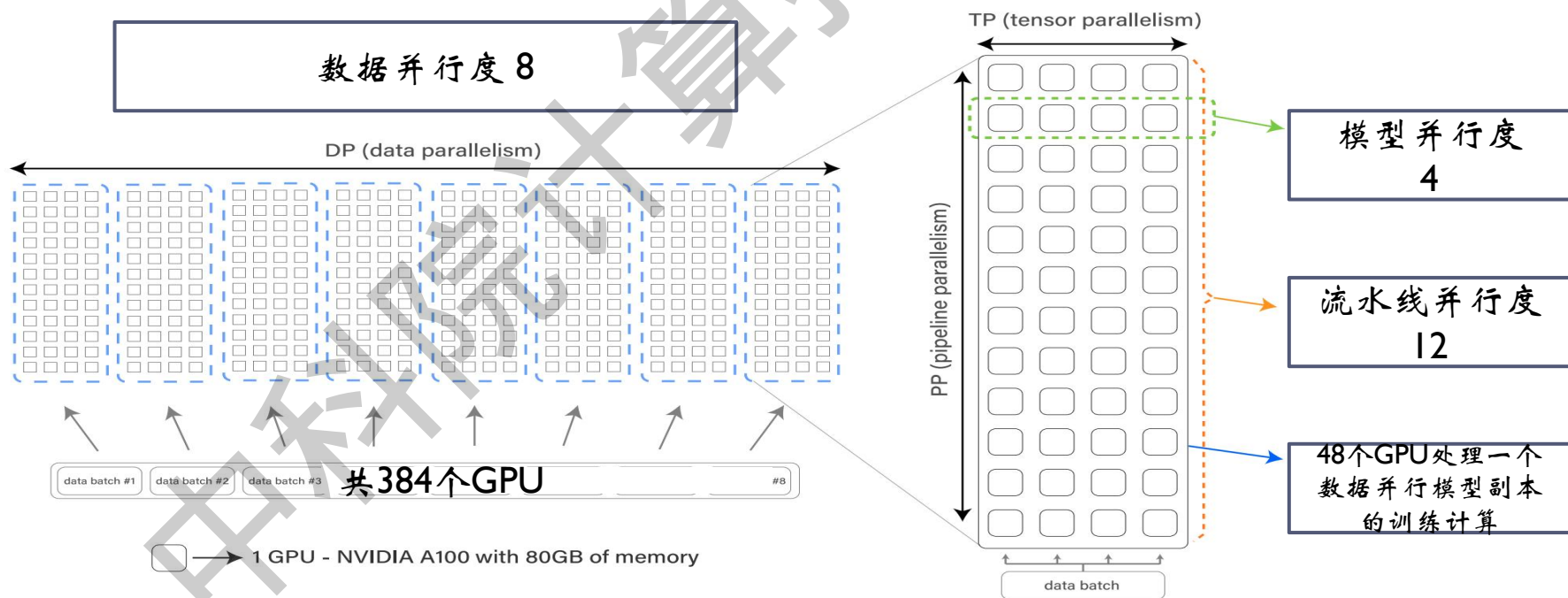
- ▶ 集群共含52个高性能计算节点。
- ▶ 实际运行：48个计算节点，即总计384 GPU<sub>s</sub>
- ▶ 热备节点：4个计算节点

项目	配置
通用处理器	64 核 (2 × AMD 霄龙 7543 处理器)
智能计算硬件	8 × 英伟达 A100 SXM4 80GB GPU <sub>s</sub>
主机端内存	512 GB DDR4
设备端内存	640 GB HBM2e
高速互联网络	4 × 英特尔 Omni-Path 100 Gbps (OPA)
共享存储系统	基于 SpectrumScale (GPFS) 的混合存储

Jean Zay 超级计算机集群单节点硬件信息

# BLOOM-176B 模型的训练过程

- ▶ 混合并行技术—数据并行、张量并行（算子内模型并行）、流水线并行
- ▶ 训练扩展到数百块GPU的同时保持高GPU利用率，加快训练速度
- ▶ BLOOM-176B模型包含70个解码器块，第1个阶段分配了1个嵌入层与5个解码器块，第12阶段分配了5个解码器块与1个嵌入层，其余阶段均分配6个解码器块

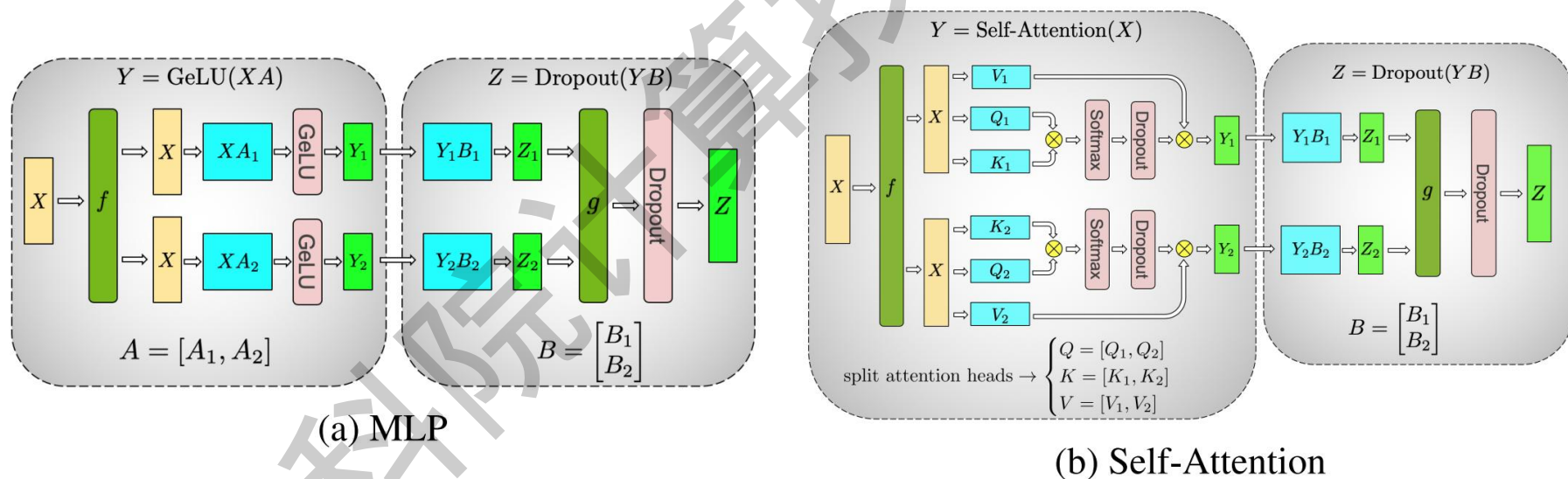


# BLOOM-176B 模型的训练过程

## 张量并行

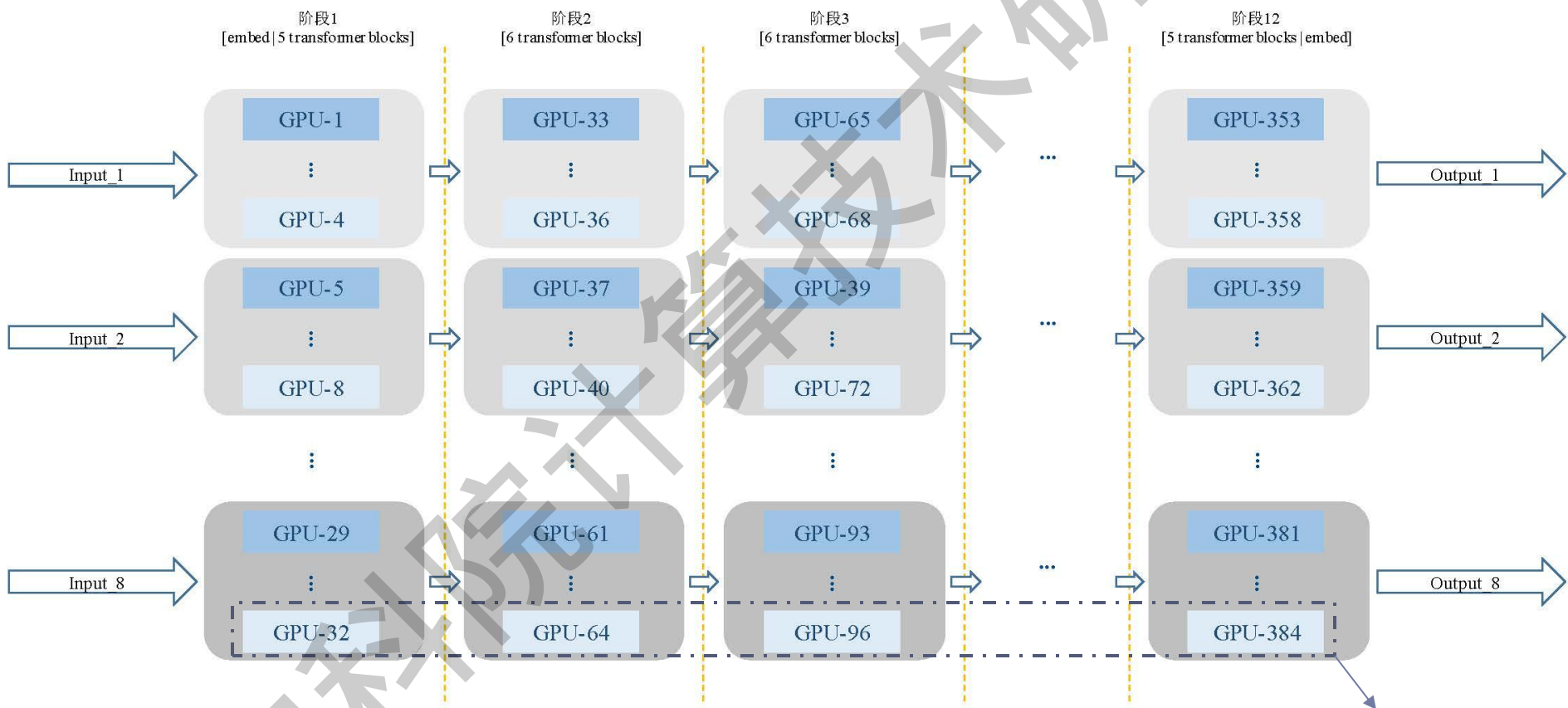
MLP: 矩阵按照行和列切分

Self-Attention: 按照Head切分和按照行切分



# BLOOM-176B 模型的训练过程

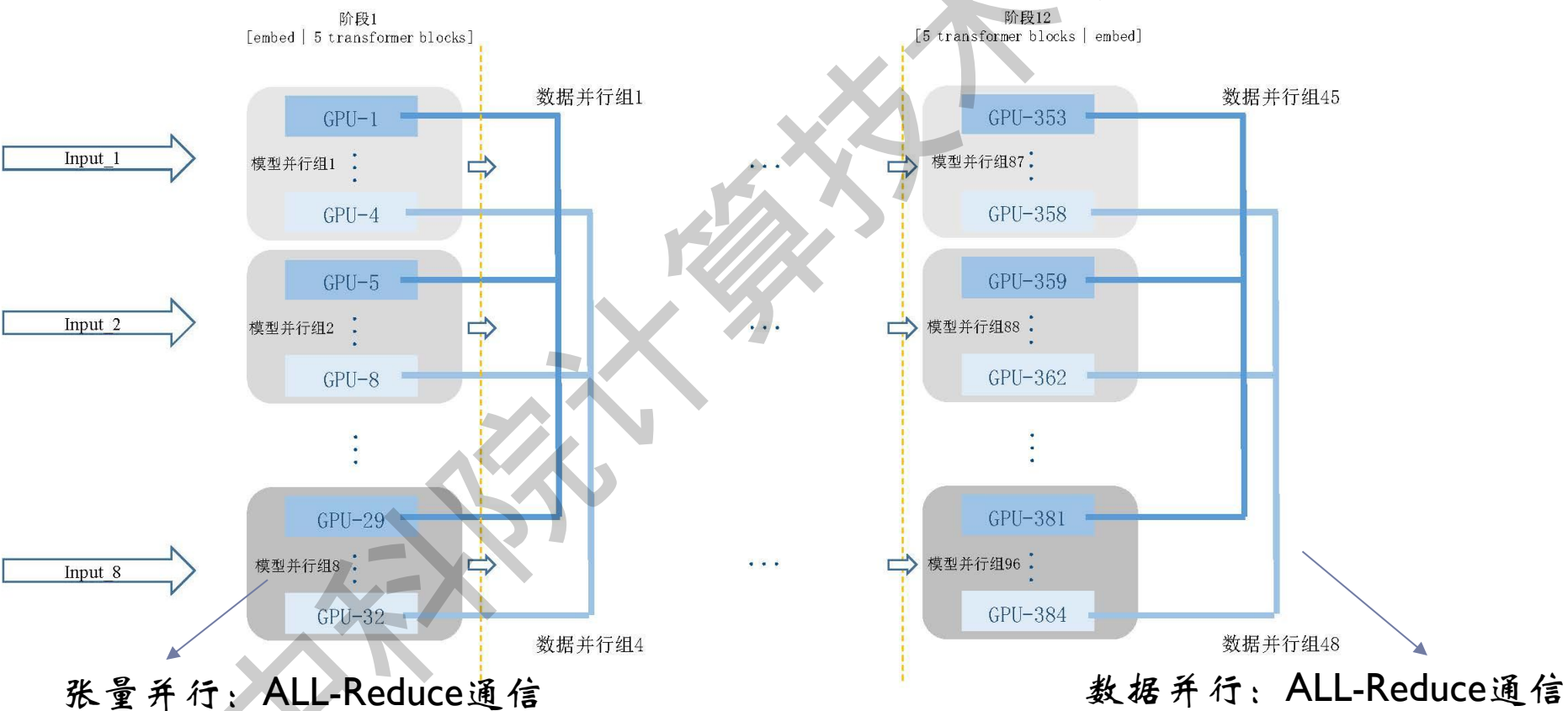
## 数据并行与流水线并行



流水线并行组: P2P通信

# BLOOM-176B 模型的训练过程

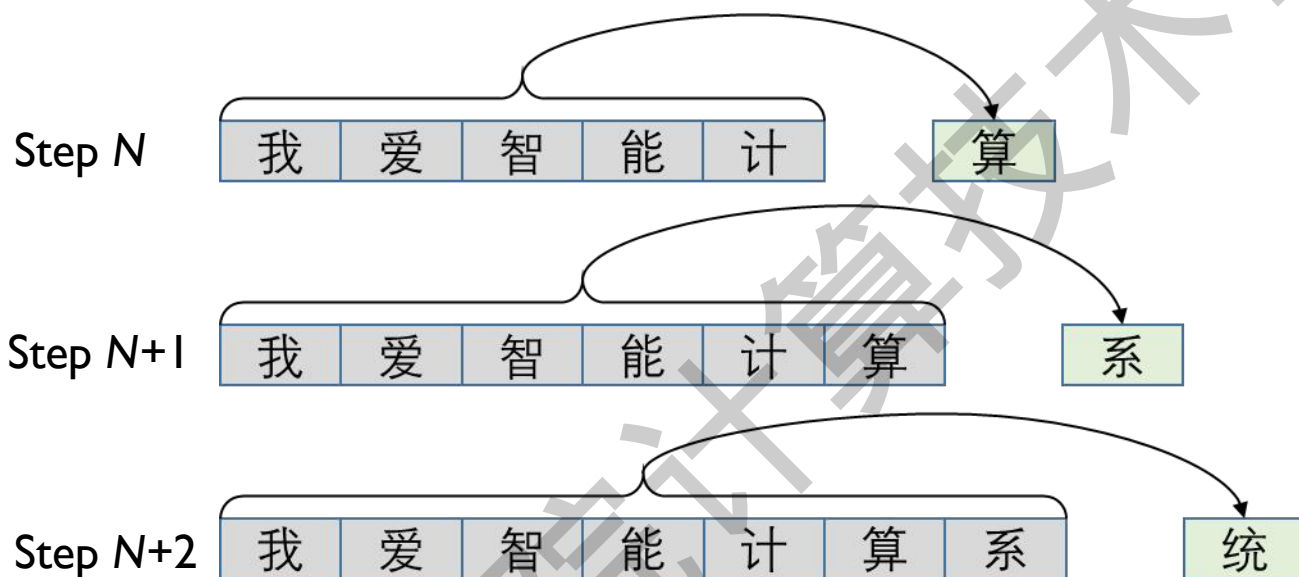
- 并行组：每一个组内的智能处理器之间会通过通信库实现数据通信，是实际通信时的操作单元。
- 进一步可以分为数据并行组（data parallel group）、张量模型并行组（tensor model parallel group）和流水线模型并行组（pipeline model parallel group）



# BLOOM-176B 模型的推理过程

## 自回归推理

- 由于缺乏正确的参考序列，在生成新词元时依赖于之前结果，模型必须依赖于自身在前面的输出来生成下一个词元

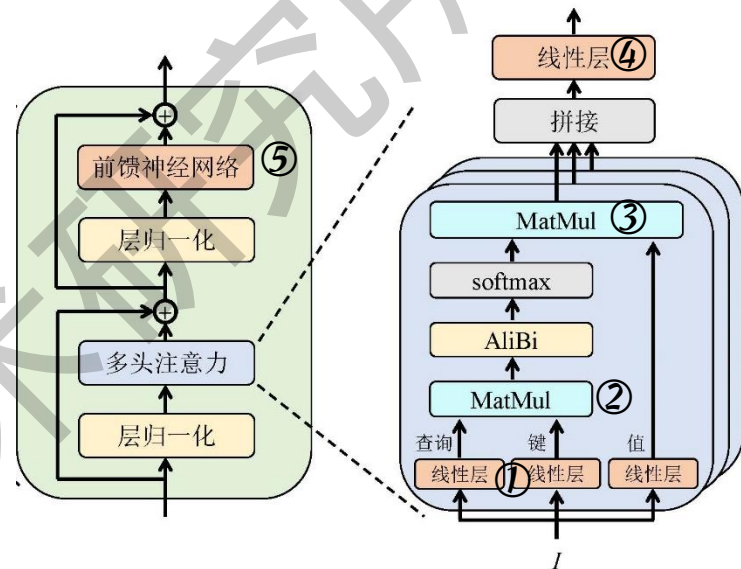


语言模型根据  
输入句子的一部分文本  
来预测下一个词

- 硬件：1个计算节点
- 并行计算策略：张量并行或者流水线并行皆可

# 计算分析

对于一个解码器块而言，正向传播时的浮点运算主要分为 5 个部分



参数	含义	分解	Input1	Input2	Output	运算量	运算密度
$h$	总隐藏层维度	$IW_Q$	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
$n$	多头注意力数目	$IW_K$ ①	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
$d$	注意力隐藏层维度	$IW_V$	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
$s$	序列长度	$Q, K, V$ 转置	$(b, s, h)$ reshape- $\rightarrow$ $(b, s, n, d)$ transpose- $\rightarrow$ $(b, n, s, d)$				
$b$	微批量大小	$Q \times K^T$ Attention ②	$(b, n, s, d)$	$(b, n, s, d)$	$(b, n, s, s)$	$2bs^2h$	$1/(\frac{1}{s} + \frac{1}{2d})$
		TEMP $\times V$ ③	$(b, n, s, s)$	$(b, n, s, d)$	$(b, n, s, d)$	$2bs^2h$	$1/(\frac{1}{s} + \frac{1}{2d})$
		转置与拼接	$(b, n, s, d)$ transpose- $\rightarrow$ $(b, s, n, d)$ reshape- $\rightarrow$ $(b, s, h)$				
		线性层 ④	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
		前馈神经网络-part1 ⑤	$(b, s, h)$	$(h, 4h)$	$(b, s, 4h)$	$8bsh^2$	$1/(\frac{5}{8h} + \frac{1}{2bs})$
		前馈神经网络-part2	$(b, s, 4h)$	$(4h, h)$	$(b, s, h)$	$8bsh^2$	$1/(\frac{5}{8h} + \frac{1}{2bs})$

按照运算密度归类：

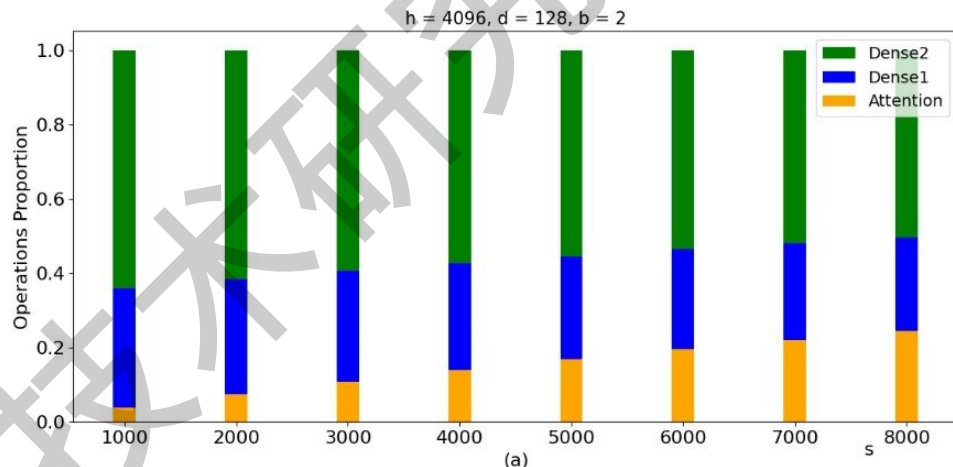
①④：Dense1

②③：Attention

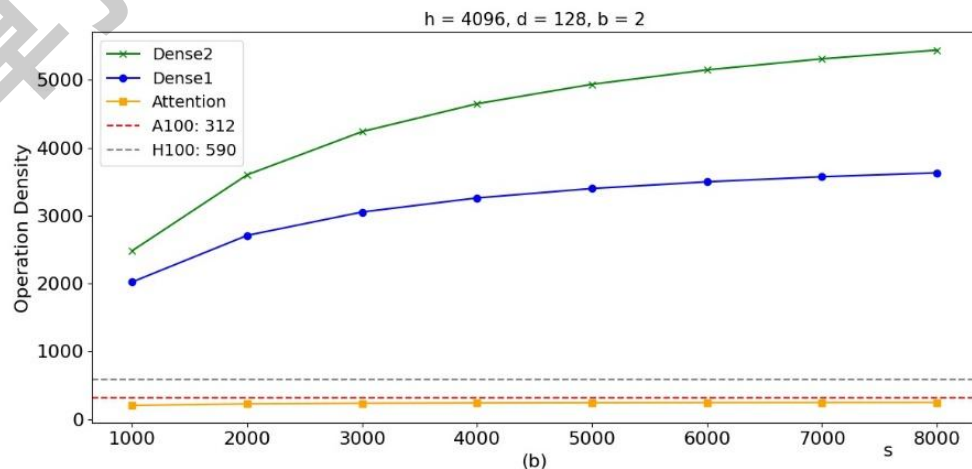
⑤：Dense2

# 计算分析

▶ 随着序列变长，注意力计算量占总比逐渐提升



▶ 但注意力的运算密度始终低于智能处理器的运算密度



多头注意力的运算可能成为大模型训练计算中的一个瓶颈。

# 计算分析

▶ 总浮点计算量:

反向传播计算量=正向传播计算量 \* 2。

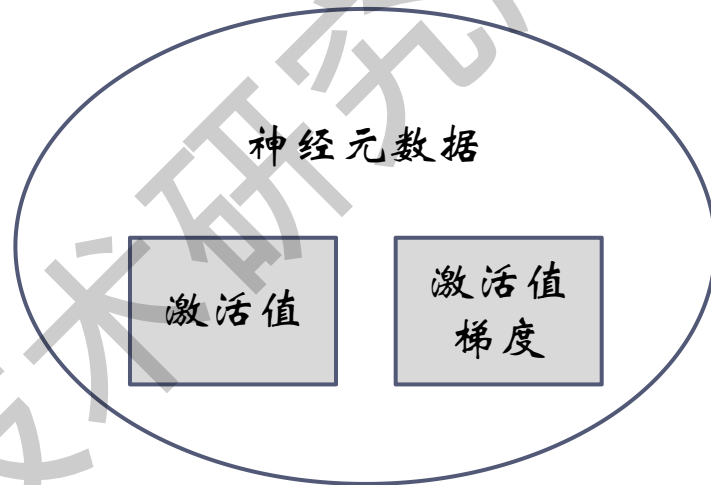
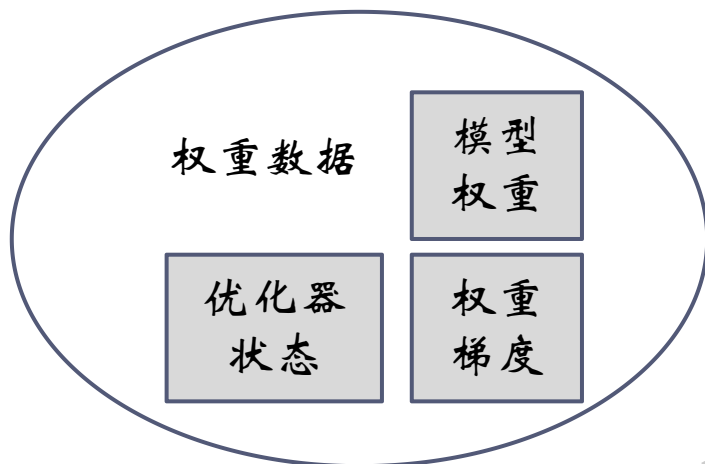
阶段	计算量
正向传播	$24blsh^2 + 4bls^2h$
反向传播	$48blsh^2 + 8bls^2h$
合计	$72blsh^2 + 12bls^2h$ (即 $72blsn^2d^2 + 12bls^2nd$ )

表 9.6 BLOOM-176B 模型训练一个批量的运算量

微批量  $b=2$ ，全局批量  $B = 16$  时，需 **34.8P FLOPs**

忽略存储容量，一块 A100 需要算数十分钟

# 存储分析



AdamW 优化器  
FP32 计算精度

张量	存储空间 (字节)	BLOOM-176B 模型训练需求
模型权重	$N * 4$	705 GB
优化器状态	$N * 12$	2115 GB
权重梯度	$N * 4$	705 GB
小计	$N * 20$	3525 GB

大模型训练时存储空间需求 (已忽略神经元数据的需求)

- 包括优化器模型权重、优化器动量、优化器方差
- 仅梯度更新时使用

3525GB 需要至少 **45** 个 80GB 的智能处理器!

实际应用中, 还需要存放神经元数据, 这进一步增加了对存储空间的需求

# 通信分析

并行策略	通信内容	通信方式	总通信量
数据并行	权重梯度	多对多规约	高
张量并行	算子分区的中间结果	多对多规约	非常高
流水线并行	激活值和激活值梯度	点对点	中

## 混行并行带来的通信内容与通信量

- 除了通信数据量大以外，大模型训练的通信还具有以下特点：
- 通信次数多，无论数据并行、张量并行、流水线并行，均会产生必要的通信和同步；
  - 通信分布不均匀，由于模型的前向和反向传播时的算子依赖关系，某些层可能需要等待其他层完成后才能通信，导致通信在时间上不均匀。

# 提纲

- ▶ 本章概述
- ▶ 大模型算法分析
- ▶ 大模型驱动范例：BLOOM
- ▶ 大模型系统软件
- ▶ 大模型基础硬件
- ▶ 本章小结

# 为什么采用大模型系统软件

- 传统的深度学习系统软件已经难以满足大模型的特殊需求。
- 大模型系统软件的出现是为了解决模型并行化、存储管理、通信优化等。
- 大模型系统软件更加注重资源利用的高效性、分布式计算的优化、以及模型的可扩展性。
- 大模型系统软件还需要考虑如何在有限的硬件资源上实现有效训练。

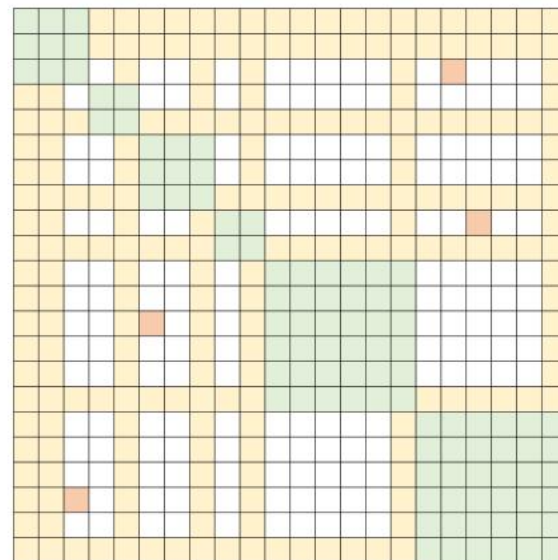


# 训练场景中计算相关优化

## 稀疏注意力机制:

- 通常情况下当前词与相邻若干词存在关联，与很远的词关联较弱
- 所有词计算自注意力 -> 信息冗余 -> 注意力存在**稀疏性**
- 稀疏注意力机制在原本全局注意力的基础上，额外引入了局部注意力和随机注意力的概念
- 通过基于块的稀疏运算，将原始注意力机制的计算需求降低几个数量级。
- 通过稀疏注意力机制优化，DeepSpeed可以用6倍的加速比执行10倍长的输入序列，优化效果显著

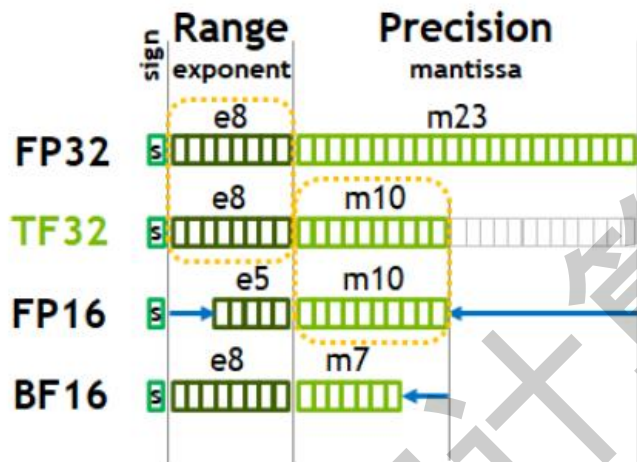
分解	Input1	Input2	Output	运算量	运算密度
$IW_Q$	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
$IW_K$	① $(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
$IW_V$	<b>长序列时，Attention的计算量显著增加</b>				
$Q, K, V$ 转置					
$Q \times K^T$	② $(b, n, s, d)$	$(b, n, s, d)$	$(b, n, s, s)$	$2bs^2h$	$1/(\frac{1}{s} + \frac{1}{2d})$
Attention	③ $(b, n, s, s)$	$(b, n, s, d)$	$(b, n, s, d)$	$2bs^2h$	$1/(\frac{1}{s} + \frac{1}{2d})$
TEMP $\times V$					
转置与拼接	$(b, n, s, d)$ transpose-> $(b, s, n, d)$ reshape-> $(b, s, h)$				
线性层	④ $(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
前馈神经网络-part1	$(b, s, h)$	$(h, 4h)$	$(b, s, 4h)$	$8bsh^2$	$1/(\frac{5}{8h} + \frac{1}{2bs})$
前馈神经网络-part2	$(b, s, 4h)$	$(4h, h)$	$(b, s, h)$	$8bsh^2$	$1/(\frac{5}{8h} + \frac{1}{2bs})$



黄色、绿色和橙色分别表示全局注意力、局部注意力和随机注意力

# 训练场景中计算相关优化

**专用数据类型**：除了传统的单精度浮点数据类型（FP32）和半精度浮点数据类型（FP16）之外，各类智能硬件还设计了专用数据类型，在基于混合精度训练的大模型训练过程中广泛使用。



使用 TF32 代替 FP32 可以几乎不降低精度的情况下，提升运算速度

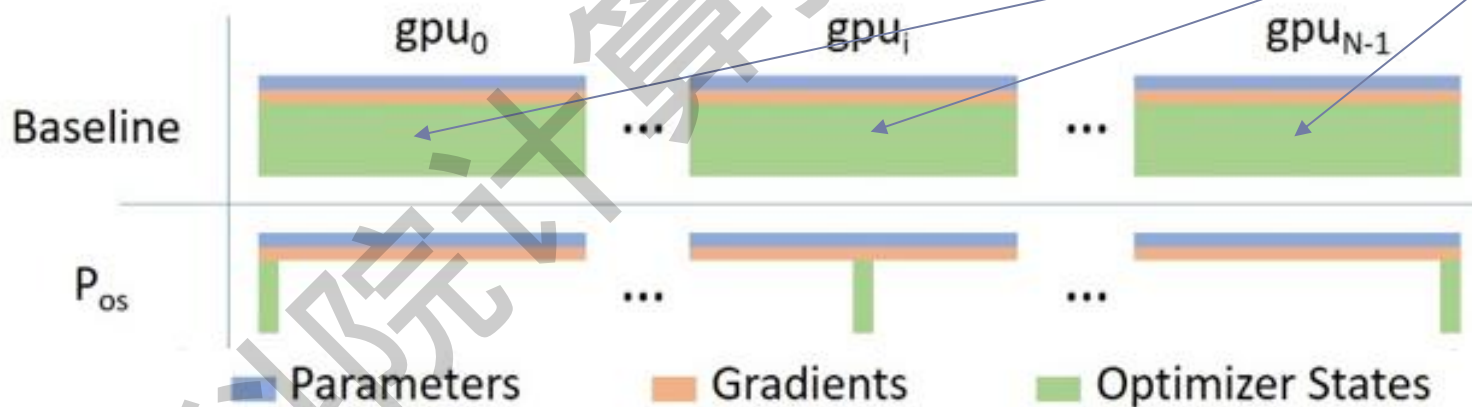
数据类型	浮点算力
FP32	19.5 TFLOPS
TF32	156 TFLOPS (8x FP32)
BF16/FP16	312 TFLOPS (16x FP32)

# 训练场景中存储相关优化——ZeRO系列存储优化

## ZeRO（零冗余优化器）一级优化

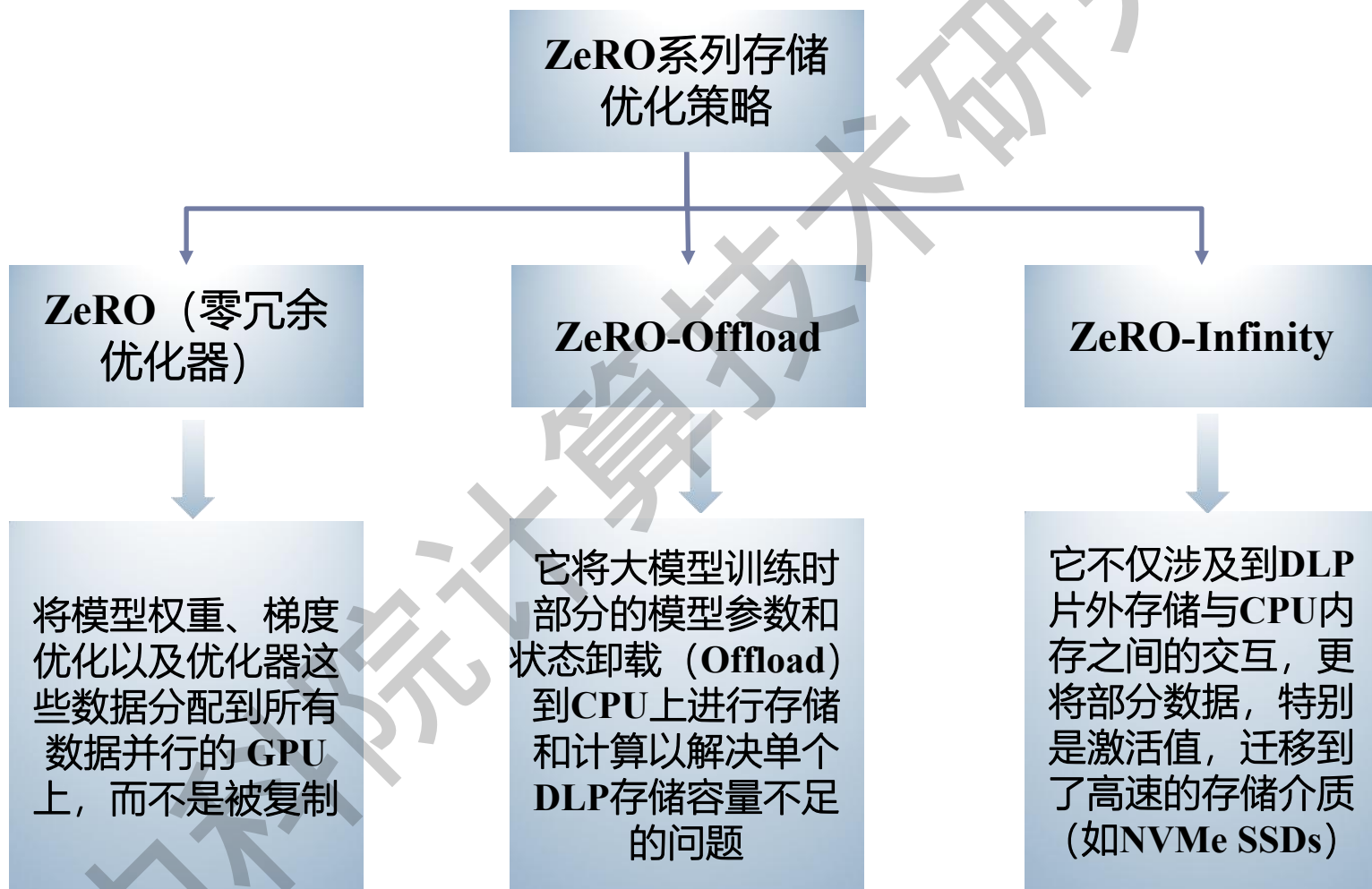
张量	存储空间（字节）	BLOOM-176B 模型训练需求
模型权重	$N * 4$	705 GB
优化器状态	$N * 12$	2115 GB
权重梯度	$N * 4$	705 GB
小计	$N * 20$	3525 GB

- 1, 优化器状态最多
- 2, 数据并行训练存在多个副本

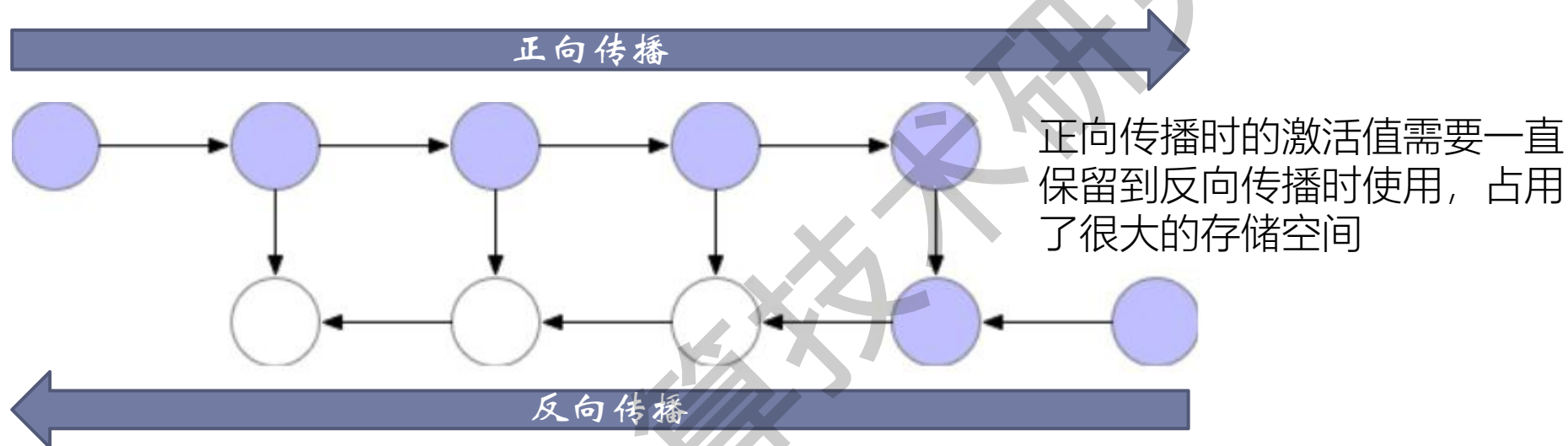


优化器状态被**分块并分配**到所有数据并行的 GPU 上，而不是被**复制**，并在训练过程中使用基于 all-gather/broadcast 的通信集合即时重建

# 训练场景中存储相关优化——ZeRO系列存储优化



# 训练场景中存储相关优化——重计算优化



- 重计算优化 (recomputation) 指的是在正向传播时不保存所有层的激活值，而是仅保留部分层的计算结果作为检查点 (checkpoint)，然后在反向传播时再根据检查点重新计算所需的激活值。

**计算换存储，计算增加约30%-40%**

- 选择性重计算 (selective activation recomputation)，通过对Transformer层内部计算量和存储量的量化分析，选择性的将中间层的激活值保留或舍弃，最终能够在引入可忽略不计的计算量的前提下，将激活值的存储使用减少5倍。

# 训练场景中存储相关优化——注意力机制融合优化

长序列时，Attention的计算中间结果存储需求显著增加，因此较长的上下文长度会引发了较大的访存量，进而影响了整体训练的性能。

分解	Input1	Input2	Output	运算量	运算密度
$IW_Q$	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
$IW_K$	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
$IW_V$	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
Q, K, V 转置	$(b, s, h)$ reshape $\rightarrow$ $(b, s, n, d)$ transpose $\rightarrow$ $(b, n, s, d)$				
$Q \times K^T$	$(b, n, s, d)$	$(b, n, s, d)$	$(b, n, s, s)$	$2bs^2h$	$1/(\frac{1}{s} + \frac{1}{2d})$
Attention TEMP $\times V$	$(b, n, s, s)$	$(b, n, s, d)$	$(b, n, s, d)$	$2bs^2h$	$1/(\frac{1}{s} + \frac{1}{2d})$
转置与拼接	$(b, n, s, d)$ transpose $\rightarrow$ $(b, s, n, d)$ reshape $\rightarrow$ $(b, s, h)$				
线性层	$(b, s, h)$	$(h, h)$	$(b, s, h)$	$2bsh^2$	$1/(\frac{1}{h} + \frac{1}{2bs})$
前馈神经网络-part1	$(b, s, h)$	$(h, 4h)$	$(b, s, 4h)$	$8bsh^2$	$1/(\frac{5}{8h} + \frac{1}{2bs})$
前馈神经网络-part2	$(b, s, 4h)$	$(4h, h)$	$(b, s, h)$	$8bsh^2$	$1/(\frac{5}{8h} + \frac{1}{2bs})$

# 训练场景中存储相关优化——注意力机制融合优化

注意力机制融合优化可以对带有softmax的矩阵乘法进行分块和融合，从而避免了 $O(s^2)$ 的片外访存。

---

## 算法 9.1 Attention 融合算法

---

- 1: **for**  $j \leftarrow 1$  to  $M$  **do**
  - 2:   将  $K_j$  和  $V_j$  从片外存储空间拷贝到片上存储空间 最高3倍的性能提升
  - 3:   **for**  $i \leftarrow 1$  to  $N$  **do**
  - 4:     将  $Q_i$ ,  $O_i$ ,  $l_i$  和  $m_i$  从片外存储空间拷贝到片上存储空间
  - 5:     计算  $S_{ij} = Q_i K_j^T$
  - 6:     计算  $m_{ij} = \text{rowmax}(S_{ij})$ ,  $P_{ij} = e^{S_{ij} - m_{ij}}$ ,  $l_{ij} = \text{rowsum}(P_{ij})$
  - 7:     更新  $m_i^{\text{new}} = \max(m_i, m_{ij})$ ,  $l_i^{\text{new}} = e^{m_i - m_i^{\text{new}}} l_i + e^{m_{ij} - m_i^{\text{new}}} l_{ij}$
  - 8:     将  $O_i \leftarrow \text{diag}(l_i^{\text{new}})^{-1} (\text{diag}(l_i) e^{m_i - m_i^{\text{new}}} O_i + e^{m_{ij} - m_i^{\text{new}}} P_{ij} V_j)$  拷贝到片外存储空间
  - 9:     将  $l_i \leftarrow l_i^{\text{new}}$  和  $m_i \leftarrow m_i^{\text{new}}$  拷贝到片外存储空间
  - 10:   **end for**
  - 11: **end for**
-

# 训练场景中通信相关优化

- 通信优化旨在减少数据传输量、提高通信效率和减少通信与计算的竞争。
- 典型的如DeepSpeed中专为大模型训练引入的1-bit Adam算法优化。通过1-bit Adam算法优化，DeepSpeed可以在保持模型精度的同时，最大减少5倍的通信量，并获得最高3.3倍的训练性能提升。
- 具体来说，1-bit Adam在每个训练步骤中首先计算出梯度的均值和方差，然后使用这些统计数据将梯度量化为1位，从而将原始的32位梯度值被压缩为1位，减少了通信的数据量。此外，1-bit Adam还采用了累积误差修正机制，确保量化过程中的误差不会累积。

# 推理场景中计算相关优化——批处理优化

T1	T2	T3	T4	T5	T6	T7	T8
S1	S1	S1	END				
S2	S2	S2	S2	S2	END		
S3	S3	S3	S3	S3	S3	S3	END
S4	S4	S4	S4	END			

(a)

T1	T2	T3	T4	T5	T6	T7	T8
S1	S1	S1	END	S5	S5	S5	S5
S2	S2	S2	S2	S2	END	S6	S6
S3	S3	S3	S3	S3	S3	S3	END
S4	S4	S4	S4	END	S7	S7	S7

(b)

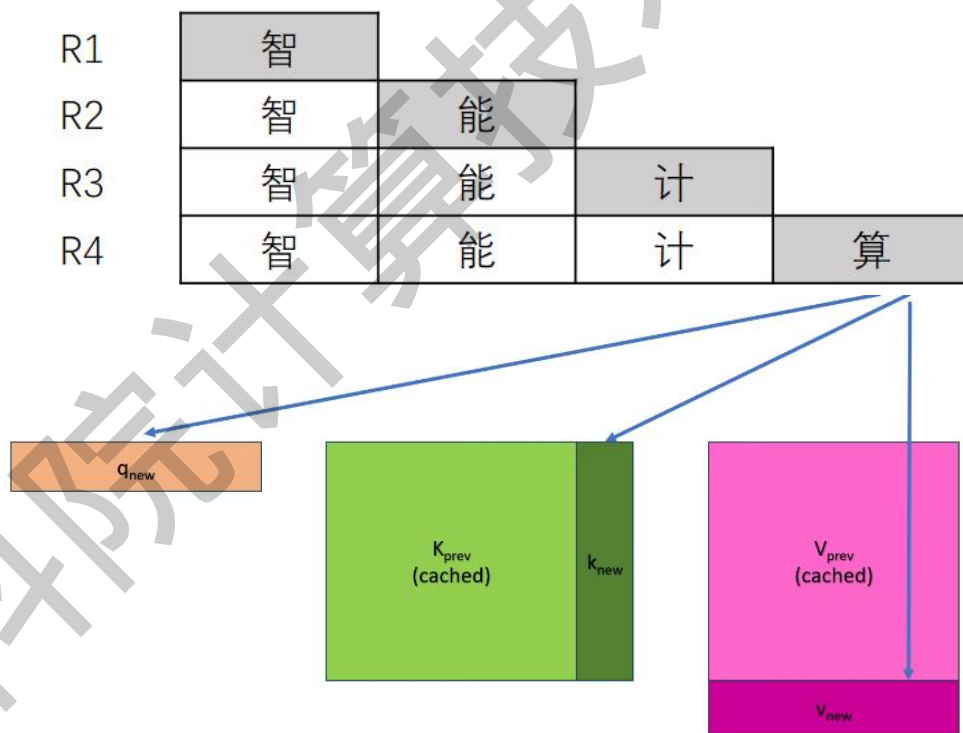
大模型批处理优化。(a) 多个任务直接批处理 (b) 连续批处理方法

**(a) 多个任务直接批处理：**该方法静态地设计一个最长的序列长度，若有任务提前结束（即输出“END”），则其需要等待同一个批量中所有任务都完成后才能结束，因此会由于负载不均衡导致整体的吞吐较低。

**(b) 连续批处理方法：**该方法动态地对任务进行批处理。当一个任务提前结束时，其会动态地选择一个新的任务进行处理，其中选择策略对于最终的吞吐率有很大的影响。一种常见的选择策略是先到先服务策略，该策略选择最近到达的任务进行调度。

# 推理场景中计算相关优化——键值缓存优化

键值缓存 (KV cache) 优化指在处理一个序列时，通过缓存过去的生成结果以避免重复计算的方法，从而减少大模型推理的计算量。



# 推理场景中存储相关优化——键值缓存分页优化

- 前述KV cache优化中，由于碎片化和过度保守的分配策略，可能导致60%到80%的存储浪费。
- 针对KV cache的分页优化借鉴了操作系统中的分页思想，通过分页的方法提高系统对存储的利用率。
- 将每个序列的KV cache划分为块，每个块包含固定数目tokens的键（Key）和值（Value），采用非连续的存储分配方案，其中块内数据连续，则可以将空间浪费率降低至5.5%

我	爱	语	文				
我	爱	数	理	化			
我	爱	智	能	计	算	系	统

(a)

块1	块2	块3	块4	块5	块6	块7	块8	块9
我	语	我	数	化	我	智	计	系
爱	文	爱	理		爱	能	算	统

(b)

KV cache 存储空间分配。(a) 无优化情况。(b) 分页优化。

# 推理场景中存储相关优化——量化优化

- ▶ 32位存储下，大模型的模型权重和激活值将占据大量的存储空间。



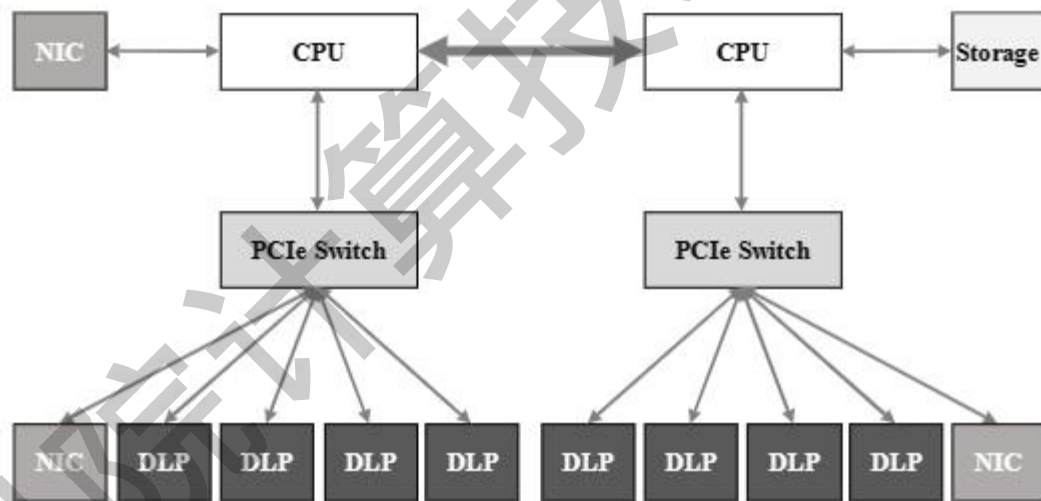
- ▶ 大模型量化的难度体现在激活值量化上，因为激活张量在通道维度上存在少量（约0.1%）的异常值
- ▶ 如果都使用一个缩放系数对整个张量进行量化，则会导致取值较小的通道有严重的精度损失，导致整体精度较差。
- ▶ 解决思路：仅权重量化、混合精度分解以及量化难度转移

# 提纲

- ▶ 本章概述
- ▶ 大模型算法分析
- ▶ 大模型驱动范例：BLOOM
- ▶ 大模型系统软件
- ▶ 大模型基础硬件
- ▶ 本章小结

# 大模型计算节点——计算节点的拓扑结构

单个大模型计算节点主要包括若干 CPU 构成的控制单元、主机端存储单元和若干 DLP 板卡构成的计算单元。

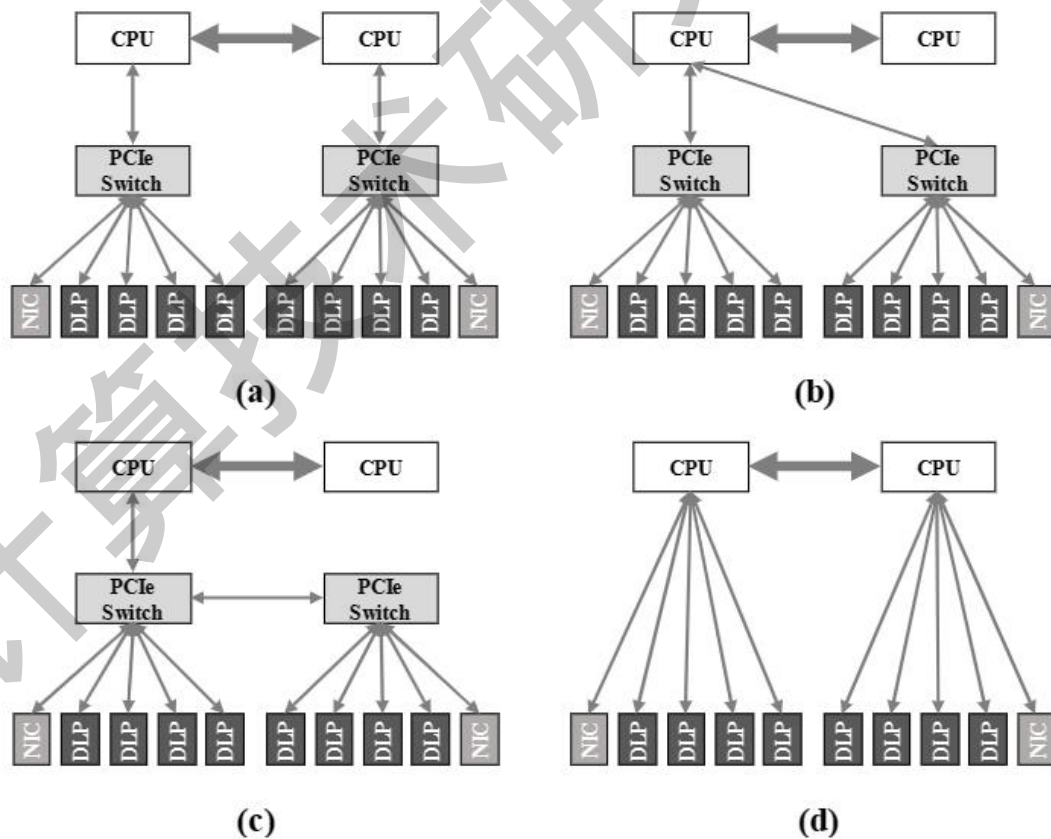


以深度学习处理器为计算核心的大模型计算节点

# 大模型计算节点——计算节点的拓扑结构

不同拓扑结构主要影响的是：

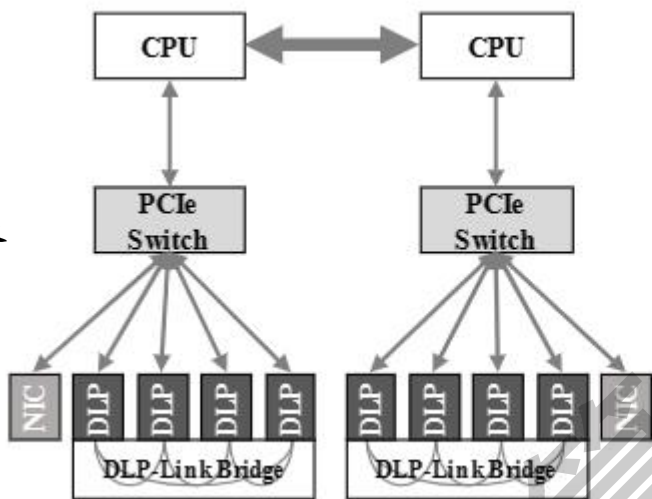
- 处理器与DLP板卡之间的**总通信带宽**，
- DLP板卡之间互相通信的**带宽**，
- DLP板卡之间互相通信的**延迟**。



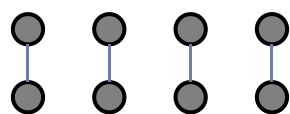
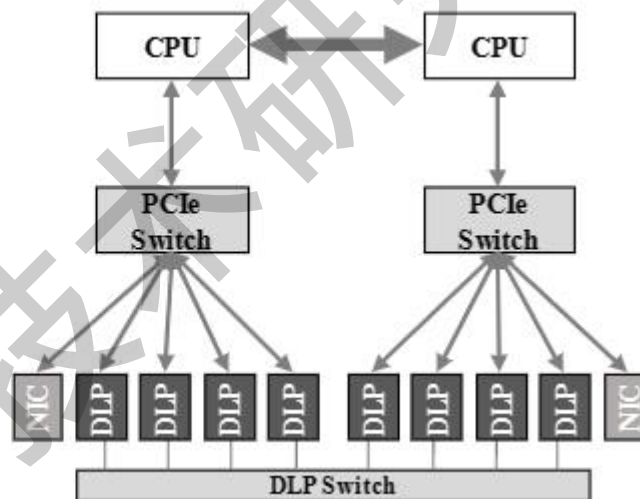
服务器的不同拓扑对比：(a) 平衡型；(b) 通用型；(c) 级联型；(d) 直连型

# 大模型计算节点——智能处理器的互联

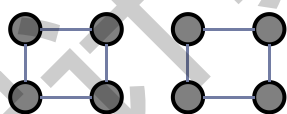
DLP桥接器  
环状拓扑



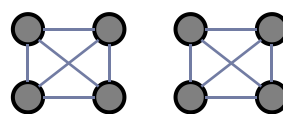
DLP交换机



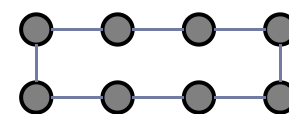
a) 4 rings



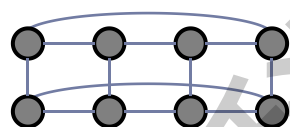
b) 2 rings



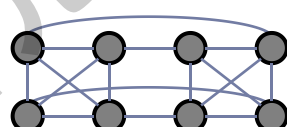
c) 2 fully-connected



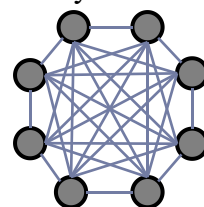
d) ring



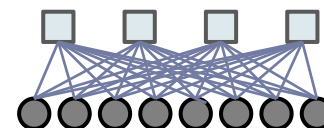
e) torus



f) hypercube



g) fully-connected

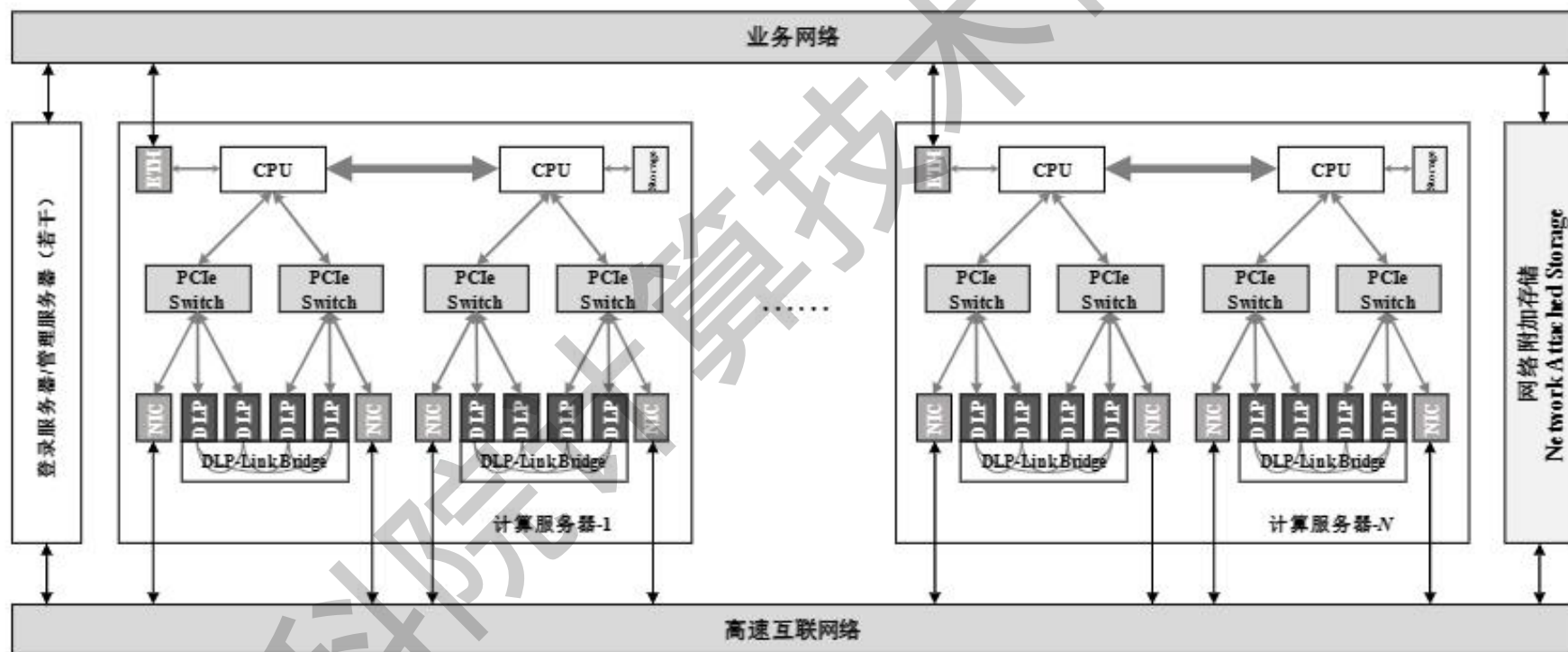


h) switch network

● Accelerator  
— Link  
□ Switch

# 大模型计算集群——计算集群的系统结构

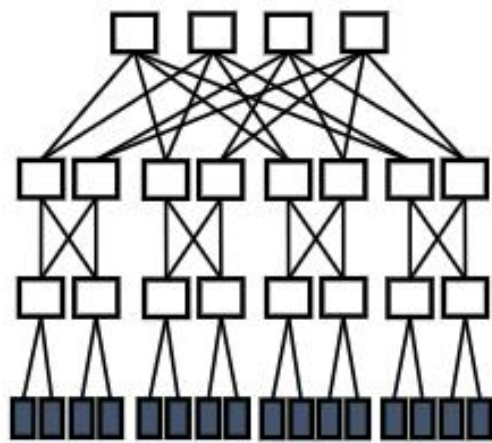
多机多卡集群配置大量的计算节点，配置若干登录管理节点进行集群管理工作，同时为了确保节点之间对数据的统一访问以及高速通讯，集群还应该配置统一的网络数据存储和多套互联通信网络。



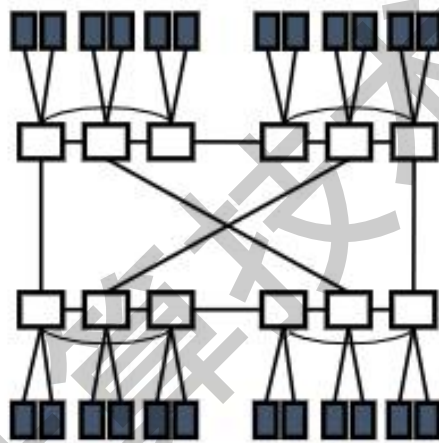
以大量智能计算节点为核心构成的多机多卡集群

# 大模型计算集群——计算集群的网络拓扑

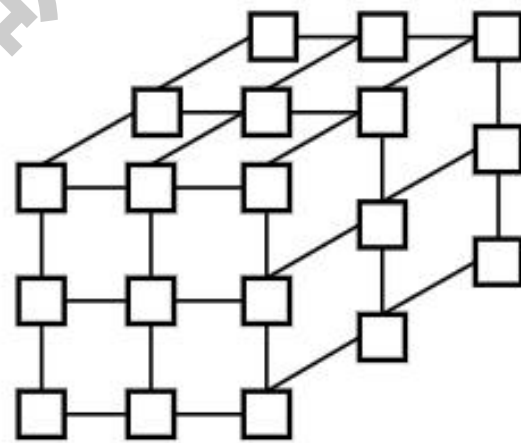
三种常见的网络拓扑结构：胖树（Fat tree）、Dragonfly 和 3D。



Fat tree



Dragonfly



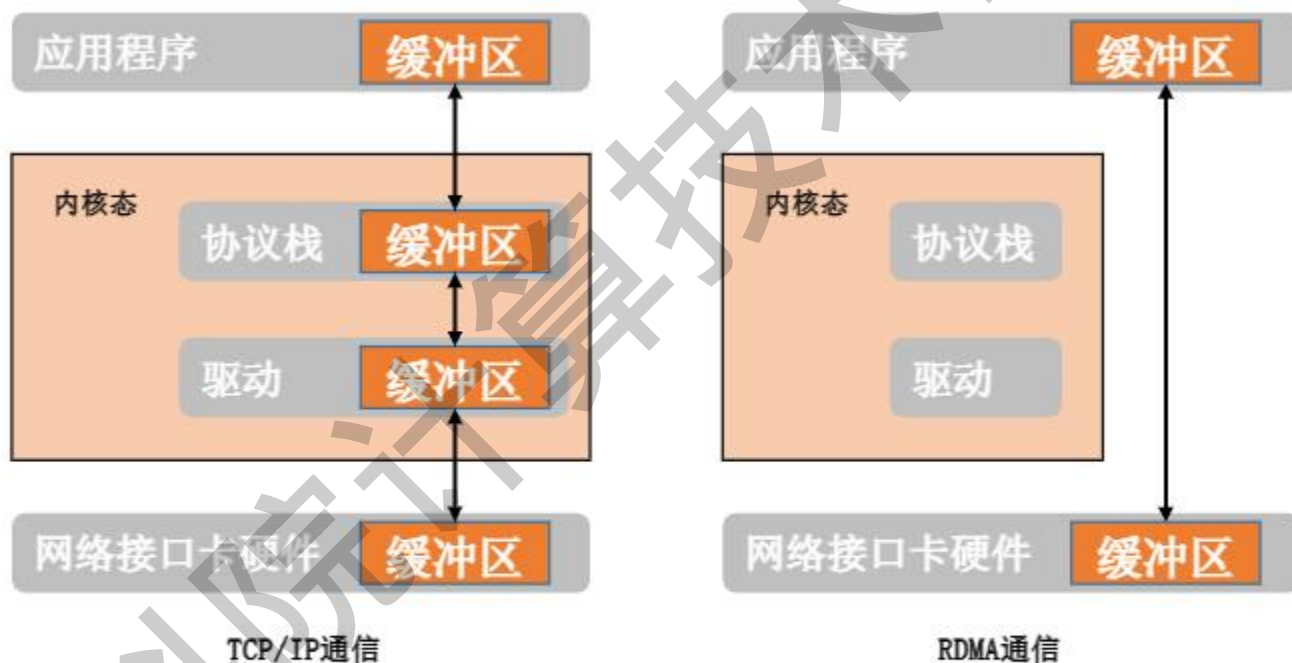
3D

网络拓扑对比

- 本地带宽指的是物理上相邻智能处理器的通信带宽，而全局带宽则用于提供整个系统的跨节点连接
- 对于大模型训练而言，张量模型并行带来的通信开销最大，因此应该将张量模型并行的范围控制在服务器本地，然后使用流水线并行来跨服务器扩展更大的网络模型。

# 大模型计算集群——计算集群的网络传输

较于TCP/IP, RDMA 零拷贝 (zero copy) 减少用户空间和内核空间中来回复制数据的开销, 内核旁路 (kernel bypass) 减少了软件调用的开销, 因此RDMA具有高吞吐、低延迟和低CPU开销的特点。



RDMA 相比较于 TCP/IP, 零拷贝减少用户空间和内核空间中来回复制数据的开销, 内核旁路减少了软件调用的开销

# 提纲

- ▶ 本章概述
- ▶ 大模型算法分析
- ▶ 大模型驱动范例：BLOOM
- ▶ 大模型系统软件
- ▶ 大模型基础硬件
- ▶ 本章小结



谢谢大家!

中科院计算技术研究所