

AlphaFold 3 推理优化实践与中山大学HPC课程建设

报告人：中山大学 黄烨



目录

1 优化场景分析

AlphaFold 3推理流程分析

应用热点分析

现有并行方法的不足

2 并行方案设计

整体方案分析

并行Pairformer

并行MSA与Template

通信Fusion

采样并行

3 CPU 特定优化

通信优化

其他优化

4 优化结果

目录

5 中山大学超算课程建设

课程与平台

6 中山大学超算队梯队建设经验分享

赛队历史

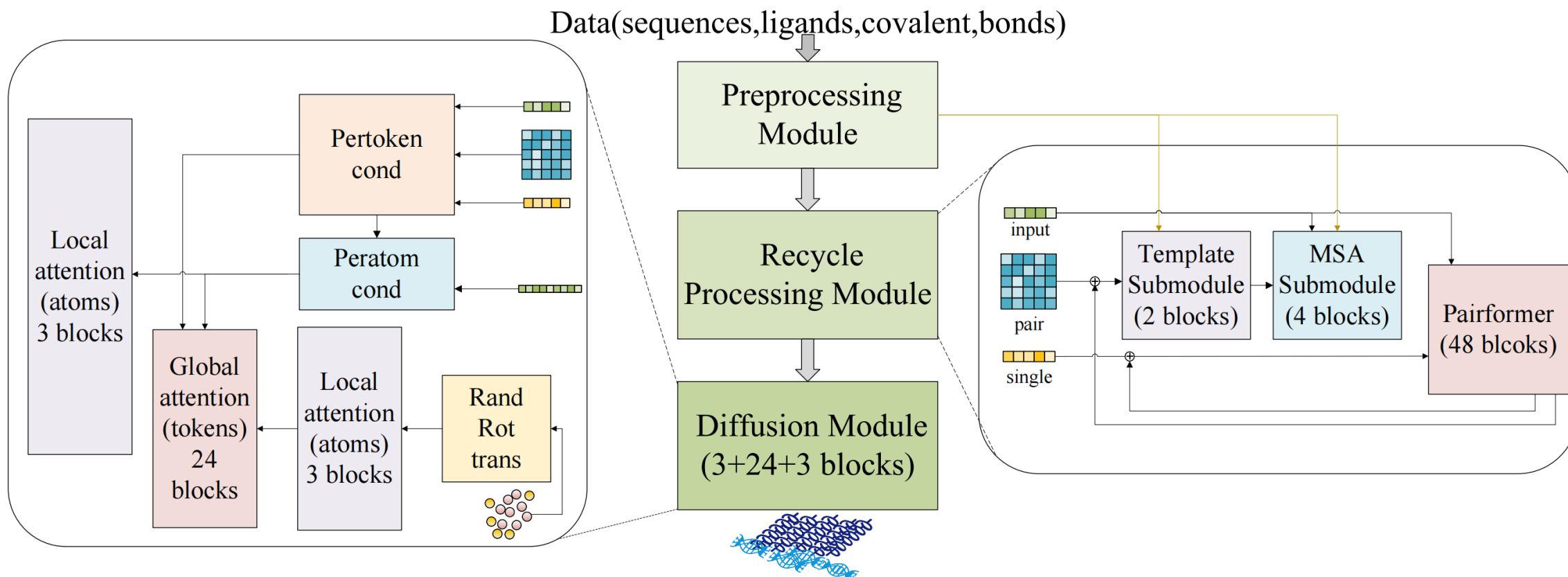
招新方案

赛队内培



相关背景/AlphaFold 3推理流程分析

AlphaFold 3的推理阶段一共分为前处理、循环处理、扩散生成三个部分。



相关背景/应用热点分析

Diffusion 与 Recycle 模块为主要瓶颈

AlphaFold 3的三个模块中，Diffusion部分占比约60%的用时，Recycle Processing Module的占比约40%，前处理部分用时可以忽略不计。由于Diffusion模块和Recycle Processing模块均在总推理时长中有显著的时间占比（40%与60%），需要分别对两个模块设计加速方案。

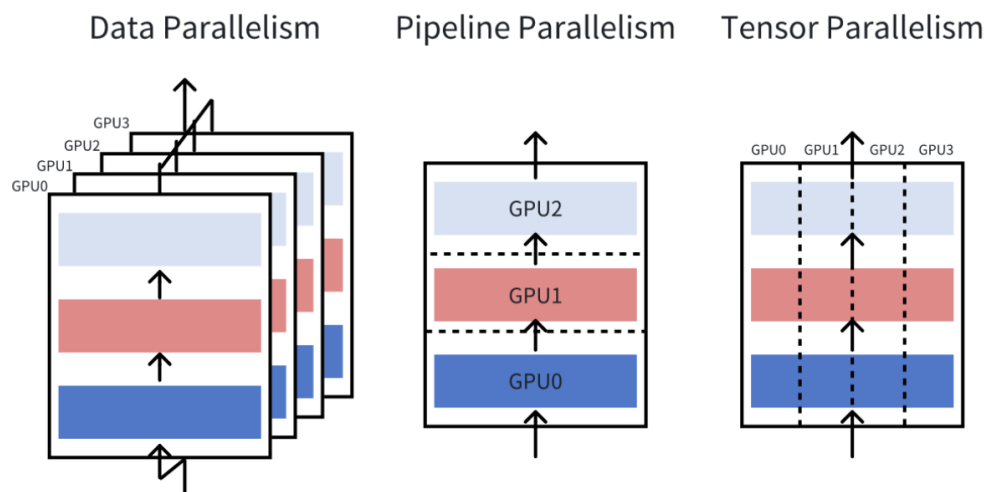
高激活值占比计算模式驱动定制化并行优化

AlphaFold 3的所有处理流程基本都是针对二维的残基对表示进行的，其空间占比随残基数量二次方增长。这带来了低模型权重占比，高激活值占比的独特特性，与现有的Transformer类大语言模型的高模型权重占比，低激活值占比存在显著区别。需要设计特定的并行优化方法。

相关背景/现有分布式并行方法的不足

由于AlphaFold 3低模型权重占比，高激活值占比的独特特性，现有的针对Transformer类模型的分布式并行方法（如DP，TP，PP），均不能高效地使用在AlphaFold 3模型上。

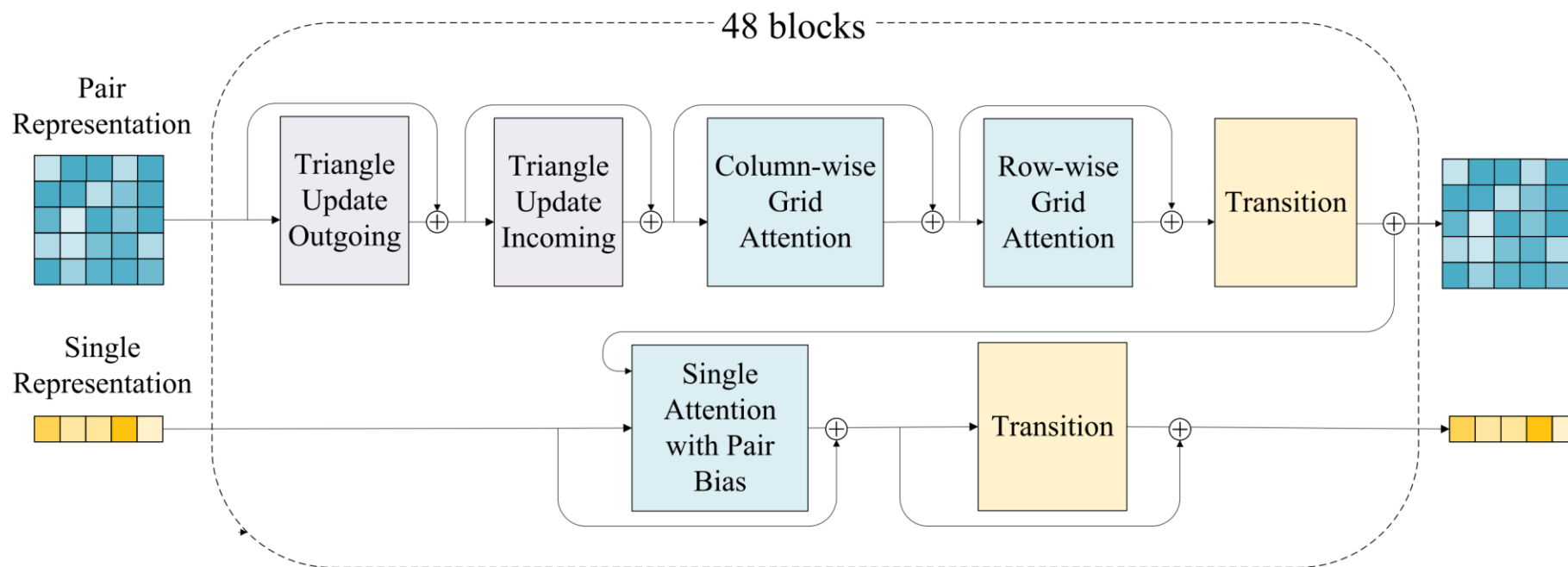
- 数据并行（DP）：需要将输入切分成完整且互不相关的块（通常是多样本切分），然而AlphaFold 3推理针对单一不可分的残基对。
- 张量并行（TP）：张量并行针对模型权重进行切分，每一个分布式实例持有部分权重与全量激活，与AlphaFold 3低模型权重占比，高激活值占比的特性并不适配，且attention阶段TP对head进行切分，扩展性受限。
- 流水并行（PP）：流水并行同样对权重分层切分，与AlphaFold 3并行特性不够适配。



并行方案设计/整体方案分析

Recycle Processing Module内主要分为Pairformer, Template Embedding和MSA Embedding三个部分
其中主要的组成部分是Pairformer, 其并行方法可以外推到Template Embedding和MSA Embedding
Pairformer内主要需要进行并行设计的为Triangle Update和Grid Attention两个模块

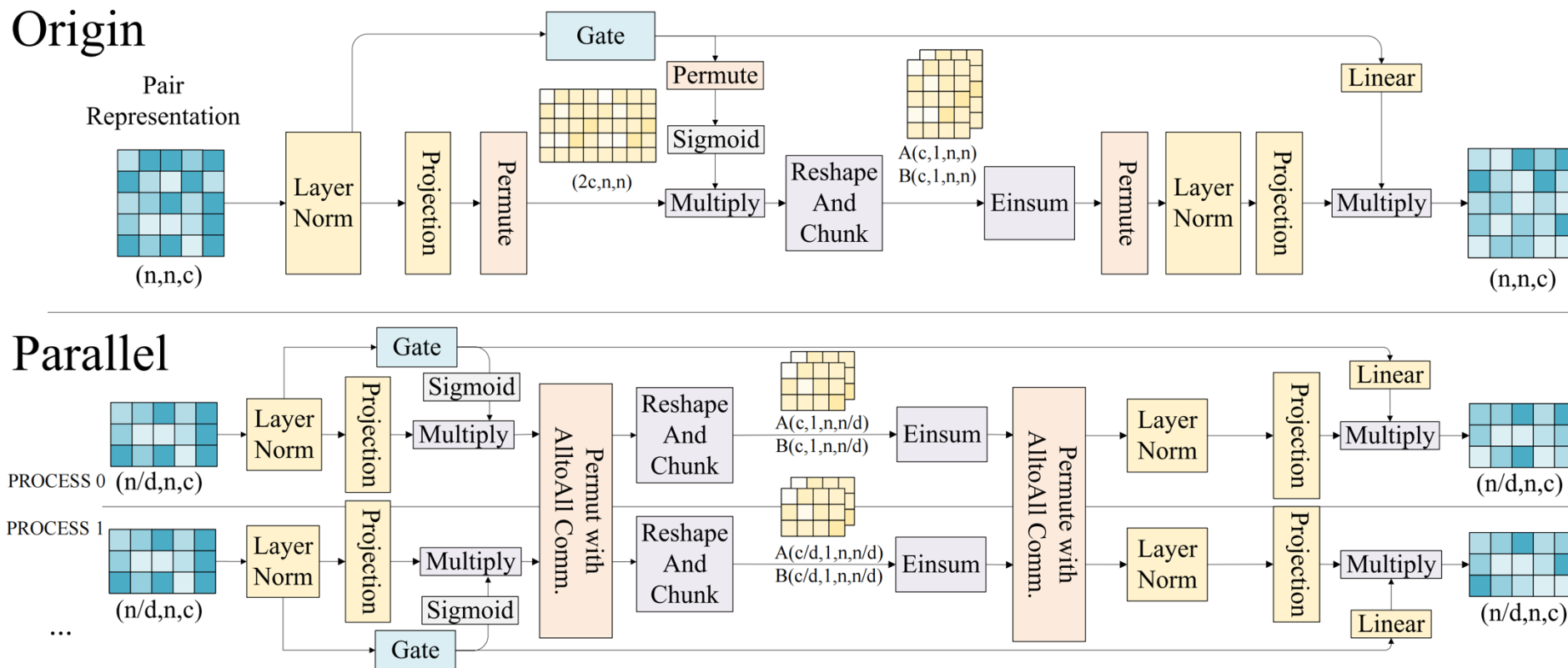
针对Diffusion部分, 由于比赛均对多样本进行采样, 我们采用采样并行的方法进行多机并行



并行方案设计/Parallel Triangle Update

Triangle Update属于是最复杂的一个模块，为了并行Triangle Update，我们主要沿着**不参与计算的维度**对残基对表示进行切分，并在该维度变化时进行all-to-all同步。

分布式场景下的Permute需要进行**阻塞通信**，代价很高，因此我们还把原版代码中的**3个Permute操作**优化成了**2个Permute操作**。

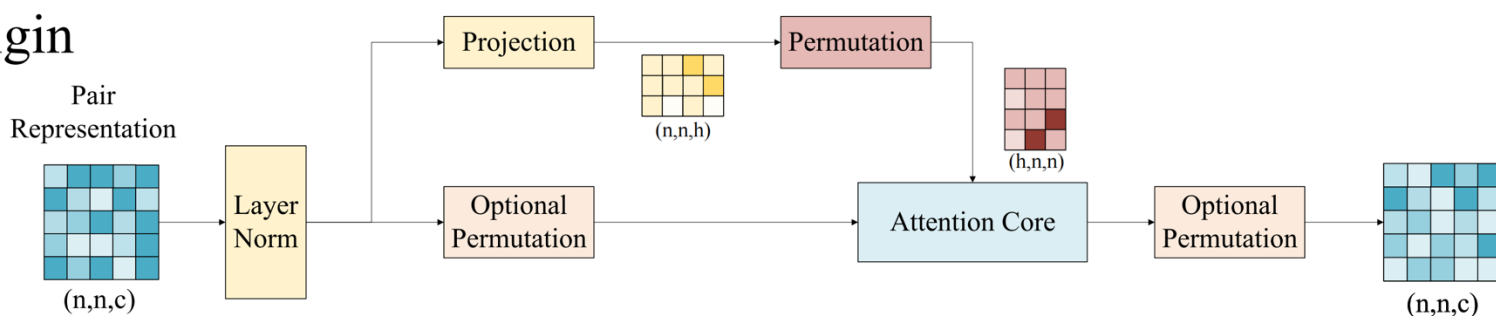


并行方案设计/Parallel Grid Attention

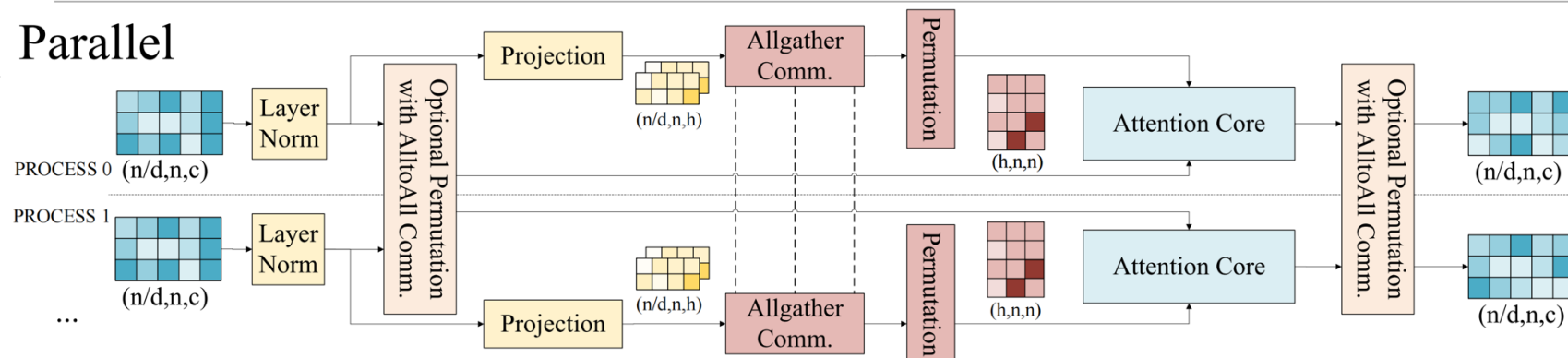
Grid Attention是对Transformer在一维序列上大获成功的注意力机制的扩展，将其应用到了二维的残基对表示上，我们可以将其视为Batch Self-Attention with Bias.

在分布式的场景下，天然的想法就是沿着Batch层进行切分，并在需要Bias信息时使用All-Gather获取全局Bias内容

Origin



Parallel

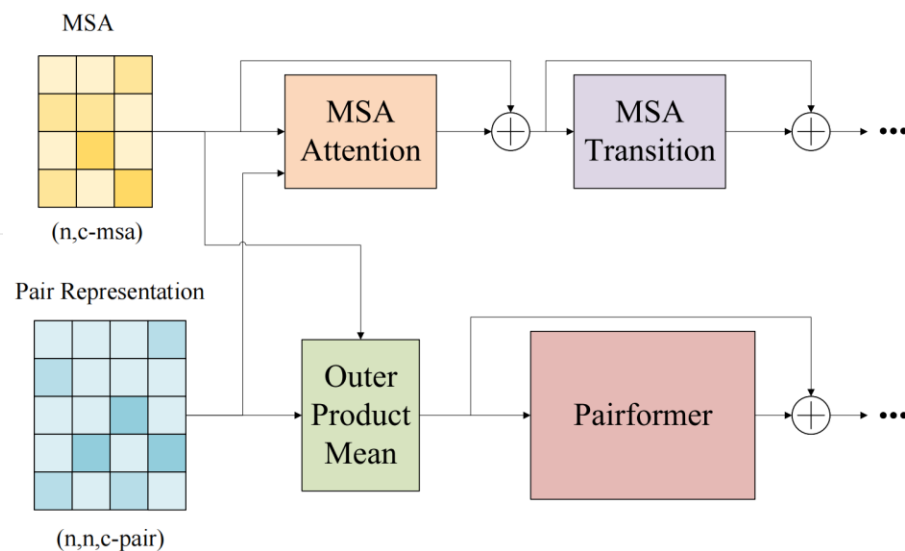


并行方案设计/Parallel MSA&Template Module

这两个模块相对简单，主要是在Pairformer的基础上嵌入MSA信息和模板信息，嵌入方式也是使用矩阵乘、Attention和Layernorm等方式的混合，基本将Pairformer的并行方法进行外推即可。

这里以MSA Module为例：

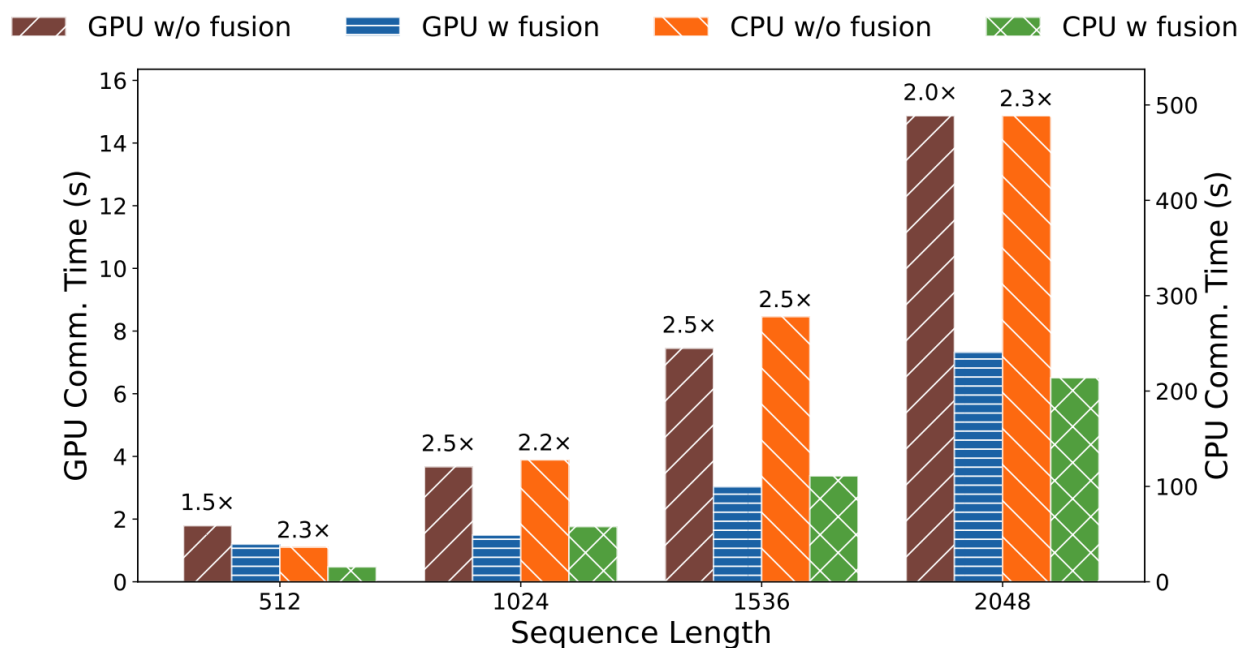
- MSA attention和MSA Transition都可以用类似的方法沿着不参与计算的维度进行切分
- 里面的Pairformer直接沿用前面的并行方法



并行方案设计/通信Fusion

在备赛前期，由于已并行模块较少，对于暂时未设计并行方式的模块，我们使用了All-Gather的方式对残基对表示切片进行聚合，但当前面提到的几个并行优化都做好了之后，Recycle Processing Module中很多地方实际上都是可以只对残基对表示进行处理而不需要聚合的，将模块进行fusion，减少All-Gather的数量能够省下来很多的通信时间。

在我们的测试中，能够带来的通信收益如下：



并行方案设计/采样并行

ASC初赛和决赛对于Diffusion阶段的采样数要求都是5，并且各样本之间的采样是完全独立的，可以进行无需通信的分布式并行，这也指导了我们最终决赛时的服务器数量选择，我们最终选择使用5台机器并不插显卡，追求最大的性能释放。

有一个要注意的点是伪随机数生成器的问题，如果分布到多台机器且使用相同的随机种子，**需要给传入的随机种子加一点偏置**，不然采样出来的结果会是完全一样的。



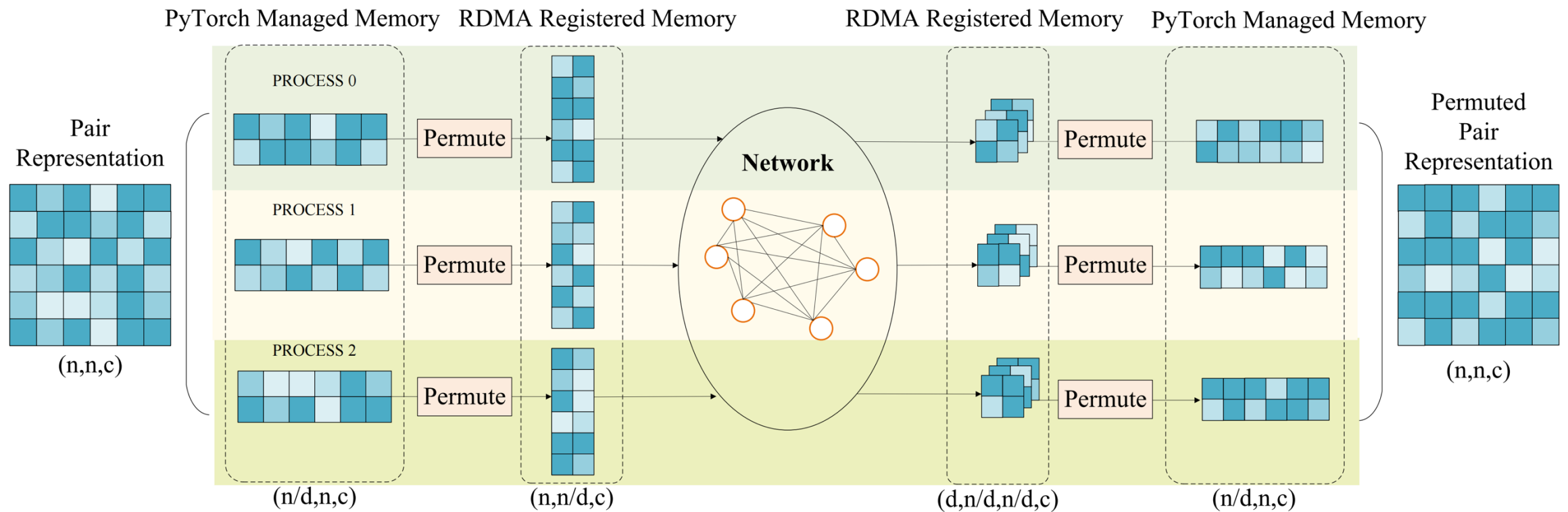
CPU特定优化/ 通信优化

Pairformer部分的并行我们一开始是使用torch.distributed实现的，在初赛GPU场景下使用NCCL后端表现良好，但是经过我们的测试后发现在CPU场景下，torch.distributed的表现非常一般。

- 如果使用GLOO后端只能跑到理论带宽的10%左右
- MPI后端在我们测试的时候会遇到inference mode和MPI不兼容的问题

这时候如果还使用torch.distributed进行通信，用时会变得不可接受。因此我们封装了MPI4PY库，在使用MPI进行通信基础上，进一步对 All-to-All 通信优化了RDMA写的首次地址注册问题，以尽可能利用通信带宽。这样才能跑到实测情况下比较可接受的通信速度。

CPU特定优化/ 通信优化



CPU特定优化/ 其他优化

Flash-Attention算子

由于CPU端无法使用xfold提供的高性能Triton Flash-attention 算子，一开始我们使用了朴素的torch实现在CPU上做替代。后来发现实际上mask和bias可以进行融合，融合后就可以使用PyTorch提供的较高性能的CPU SPDA实现。

异构进程分配+针对性功耗调控策略

在赛场服务器环境下，发现Recycle Processing部分适合2P48T的配置，Diffusion部分适合1P96T的配置，因此我们使用了异构的进程、线程分配方式，并设计了对应的功耗调控方法。

内存分配库优化

我们在测试后使用了Google的tcmalloc库替代了PyTorch原生的内存分配器，PyTorch的原生内存分配器非常不适合在CPU上使用。

实验结果

单链场景优化结果(部分)

8w4w_a_358_358	38.62s
9jcv_a_606_606	69.75s
8qft_a_1174_1174	201.54s

多链场景优化结果(部分)

8c4z_a_112_b_282_394	44.63s
8x0w_a_470_b_464_934	133.10s
8j7f_a..._e_20_1176	203.64s

Part 2 中山大学HPC课程建设与超算队梯队建设分享



Coursework of System in SYSU

- HPC Programming and Practicing
- Parallel Programming and Algorithm
- Computer Architecture
- Operating System
- Principle of Compiler
- Computer Networks
- Computer Organization
- Basics of C Programming
-

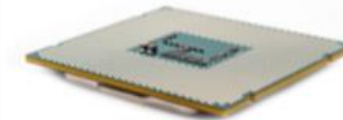


超级计算机原理与操作

中山大学杜云飞老师主讲

并行计算

👤 619 🔥 1.1 万 ⭐ 3 🍵 6



高级并程序设计

中山大学卢宇彤和杜云飞教授主讲, 超算专题

并行计算

👤 700 🔥 4,401 ⭐ 8 🍵 6

中山大學

SUN YAT-SEN UNIVERSITY

高性能计算程序设计基础-2021年秋

以“高性能计算程序设计基础 (2024) 课程邀请...

并行计算

大数据

👤 113 🔥 7,019 ⭐ 3 🍵 3

Lecture and Lab Platform

- Our HPC lecture is released on <https://easyhpc.net/course/212>
- 中山大学超算习堂
 - Email to huangd79@mail.sysu.edu.cn to get an invitation code
 - Include slides, Labs and reading materials
 - Cluster resource for MPI and OpenMP
- GPU resource is provided at ai.easyhpc.net (智算习堂)
 - NSCC-GZ support various GPUs
 - x86/ARM CPU, DCU
 - 3090, V100, K20 GPU



Online Lab Platform

- Integrated coding environment
- High quality online course materials
- Learning and practicing

The screenshot displays the Online Lab Platform interface, which is divided into two main sections: course materials and a software library.

Course Materials Section:

- Linux 基础入门:** A course titled "Linux 基础入门" (Linux Basics) with 1.1 million views and 362 likes. It includes a video thumbnail showing a Linux terminal.
- MPI 编程环境部署:** A course titled "MPI 编程环境部署" (MPI Programming Environment Deployment) with 9,250 views and 505 likes. It includes a video thumbnail showing a 3D molecular structure.
- 并行计算 - 结构·算法·编程:** A course titled "并行计算 - 结构·算法·编程" (Parallel Computing - Structure, Algorithm, Programming) with 8,565 views and 43 likes. It includes a video thumbnail showing a 3D molecular structure.
- GPU 并行计算和CUDA程序开发入门:** A course titled "GPU 并行计算和CUDA程序开发入门" (GPU Parallel Computing and CUDA Program Development Introduction) with 4,554 views and 6 likes. It includes a video thumbnail showing a 3D molecular structure.
- 天体物理概论:** A course titled "天体物理概论" (Introduction to Astrophysics) with 48 views and 0 likes. It includes a video thumbnail showing a 3D molecular structure.

Software Library Section:

- WebIDE:** A software environment for web-based development, available in CUDA 10.0 and 10.2 versions.
- Ubuntu 24.04 LTS:** A software environment for Ubuntu 24.04 LTS.
- Jupyter Notebook:** A software environment for Jupyter Notebook.
- PyTorch 2.6.0 Notebook:** A software environment for PyTorch 2.6.0 Notebook.
- SciPyStack Notebook:** A software environment for SciPyStack Notebook.

The interface also features a search bar, a navigation menu, and a footer with contact information.

HPC and AI Coding Environments

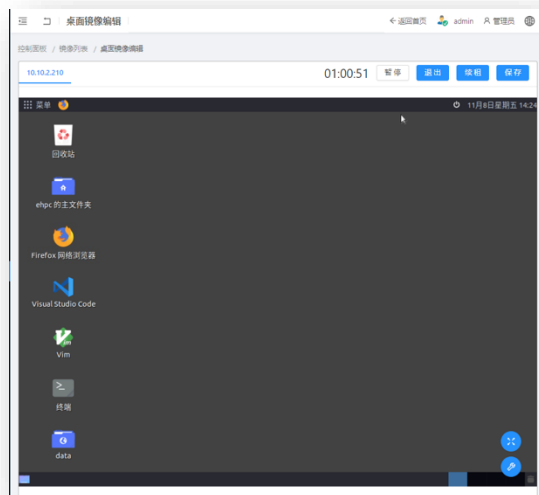
- Customized containers and resources for Lab

安装软件环境：如MPI、OpenMP、Pthreads、CUDA、Python

镜像编辑

节点配置

指定硬件配置：节点数、CPU核心数、内存大小、GPU类型、使用时长等



实践环境



资源引用

在线实训

在线课程

软件库

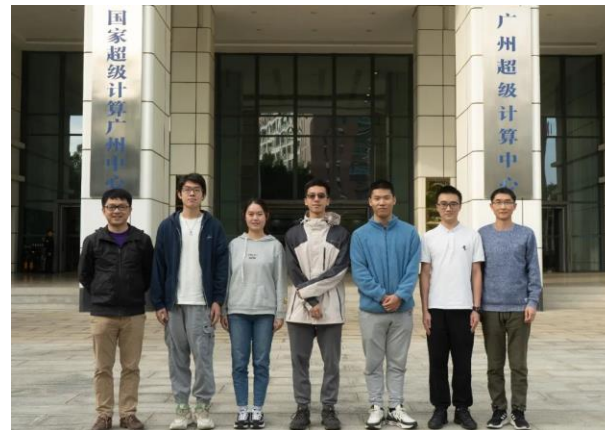
案例

ARM、x86 CPU集群

DCU、4090、K20、V100 GPUs

Student Cluster Club in SYSU

- Founded in 2012
- Base on NSCC-GZ and CSE SYSU
- Participate in various cluster and parallel competitions
- Cultivating expertise in HPC and AI



Advisors

- Zhiguang Chen (陈志广) :
 - Professor in SYSU and NSCC-GZ
 - Ph.D. obtained from NUDT
- Dan Huang (黄聃)
 - Associate Prof. in SYSU and NSCC-GZ
 - Previously worked in ORNL and NJIT
 - Ph.D. from University of Central Florida
- Guangnan Feng (丰光南)
 - Post-doc in SYSU and NSCC-GZ
 - Participated in ASC'18, ISC'19
 - Ph.D. obtained from SYSU with publishing papers in SC and ICS
 - Research area: HPC interconnect protocols, algorithms and implementations



Awards and Achievements

- Recent HPC awards
 - 2025 ASC, First Prize
 - 2024 ASC, silver prize
 - 2022-23 ASC, first prize
 - 2024/25 ISC, online Champion
 - 2023 ISC, 4th place
 - 2022 ISC, 2nd place
 - 2024 SC, 2nd place (IndySCC)
 - 2022 SC, Online Champion
 - 2021 SC, Champion (IndySCC)
- Parallel code optimization
 - 2024 PAC, 1st prize
 - 2022 PAC, 2nd prize
 - 2020 PAC Silver prize



Recruitment

- Assign **tasks** .
- Provide **tutorials** for beginners.
- **Track progress**.
- Encourage, and provide guidance.

Publicity

- Open to **all majors**.
- Online and offline channels.
- Welcome students with **passion, dedication, and humility**.

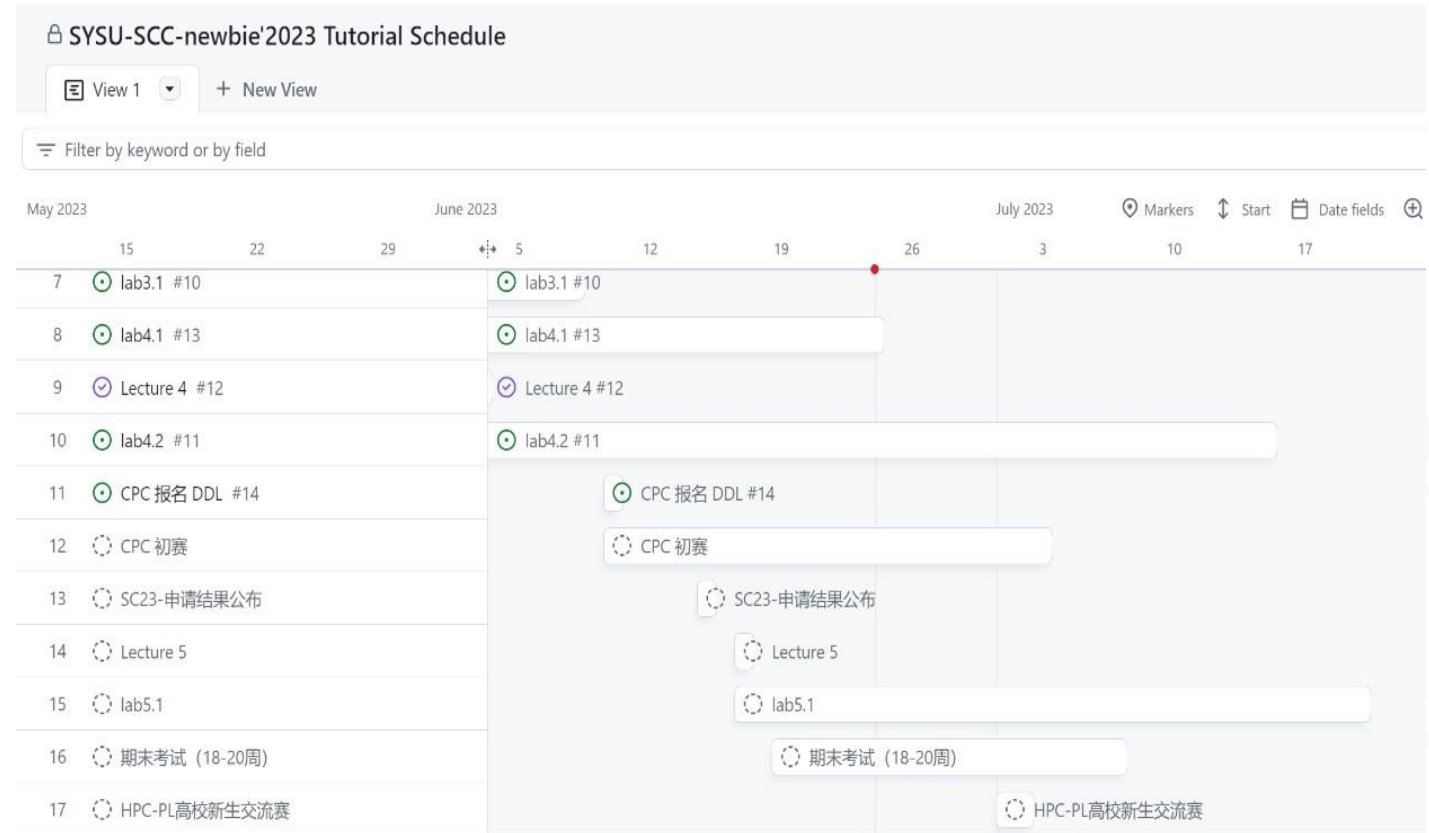
Preliminary

Final Interview

- **Question** task completion.
- **Stress** interviews.
- Learn About Candidates' **Personality**

Internal Training

- Invite seniors to make **tech talks** every week.
- Leverage **academic forums** at NSCC-GZ to broaden horizons.



Tutorial Schedule for Newbies

Daily Training

- Group Competitions.
- Time Management.
- Review and Summary.



Start from Server

Thank you

