昇腾长序列 + verl：
Qwen3-235B长序列RL训练系统优化

CANN大模型训推优化专家　曹靖宜
2025年12月20日

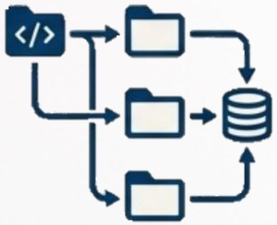
Content

目录

背景介绍与总览

01

1.1 LLM训练趋势：更长的上下文，解锁全新应用场景



- **Agent应用（e.g. 仓库级代码理解）**
- 让AI编程助手理解整个代码库，进行架构级重构和复杂错误修复

- **深度文档分析**
- 一次性处理数百页的法律合同、财务报告或医学记录等，无需切分文档

- **长时间多模态交互**
- 分析长达数小时的视频、音频内容，或处理包含数百张图片

模型	开发者	上下文长度
Llama4 Scout	Meta	10M
Llama4 Maverick	Meta	1M
Gemini 3 Pro	Google	1M
Gemini 2.0 Flash	Google	1M
Claude 4 Sonnet	Anthropic	1M
Grok4 Fast	xAI	2M
Qwen3-Coder	Alibaba	1M
DeepSeek-R1	DeepSeek	128K
GPT-5	OpenAI	400K

1.2 长序列RL挑战

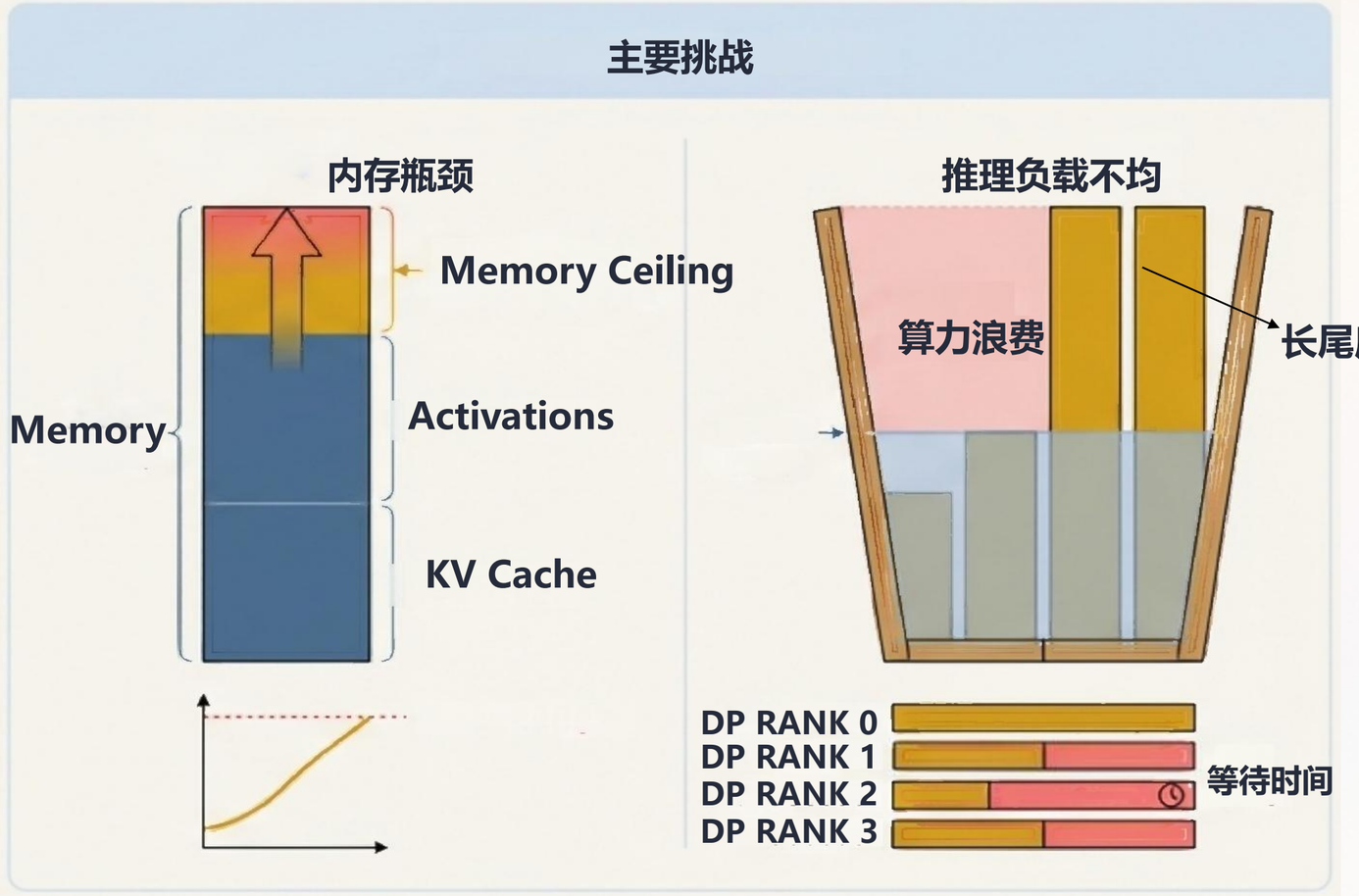
- 长序列能力是LLM 实用化的关键，但在RL场景下面临独特的工程瓶颈。
- 长序列推理和训练对系统效率和稳定性提出更高要求 。

内存瓶颈 Memory Bottleneck

KV Cache与激活值内存随序列长度线性增长，在32k上下长度下极易导致OOM

长尾分布不均 Long-tail Load Imbalance

少数长序列任务成为整个系统的“木桶短板”，导致大量算力闲置等待



1.3 基于verl框架的Qwen3-235B 32K长序列RL实践

- 开源地址：https://gitcode.com/cann/cann-recipes-train/blob/master/llm_rl/qwen3/README.md
- 框架：verl + MindSpeed + vLLM-Ascend

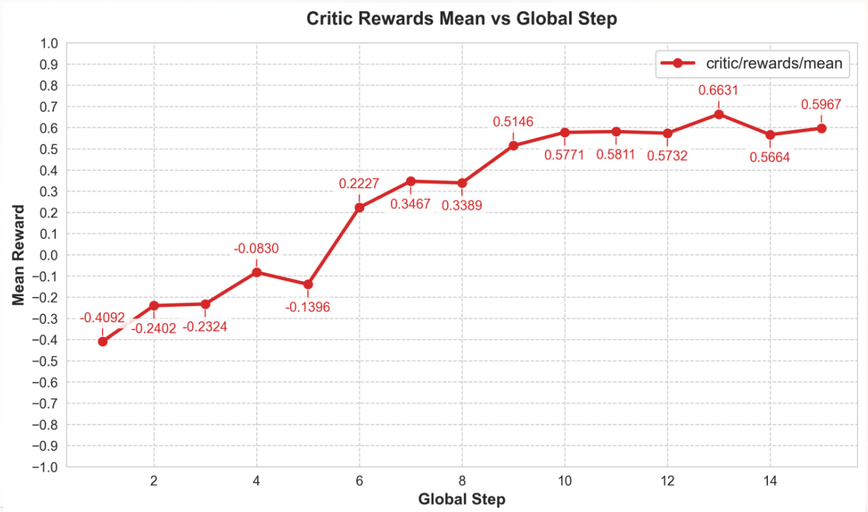
GRPO算法优化性能

训练吞吐
404TPS/卡

推理吞吐
234TPS/卡

系统吞吐
122TPS/卡

DAPO训练reward曲线



模型	RL算法	平均输入	平均输出	推理时间	训练时间	总时间	GBS	采样数	Rollout 部署	Actor/Ref 部署
Qwen3-235B-A22B	GRPO	73.7	7344.97	6638.49s	797.93s	7833.16s	512	16	128die **DP32TP4** **EP128**	128die **TP4PP4** **CP4EP32**

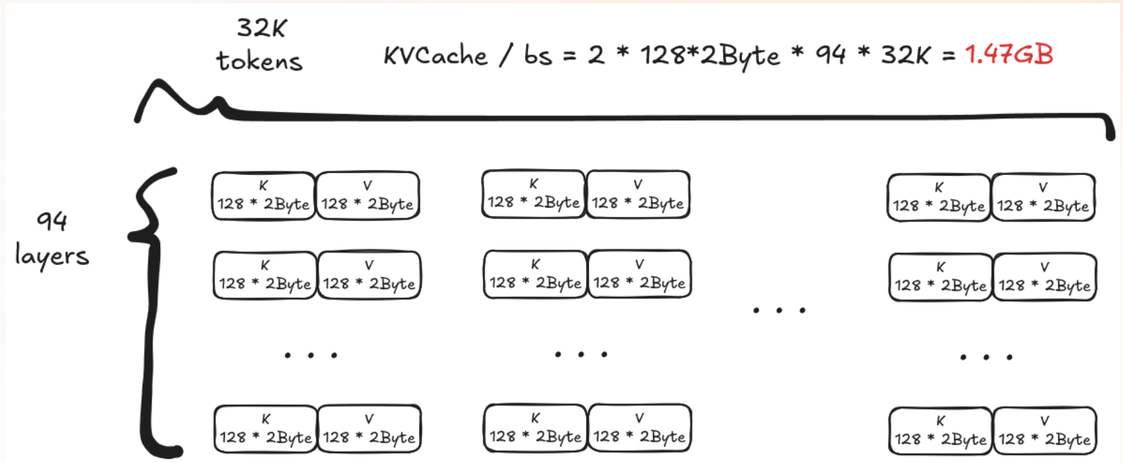
1.4 verl优化实践总览

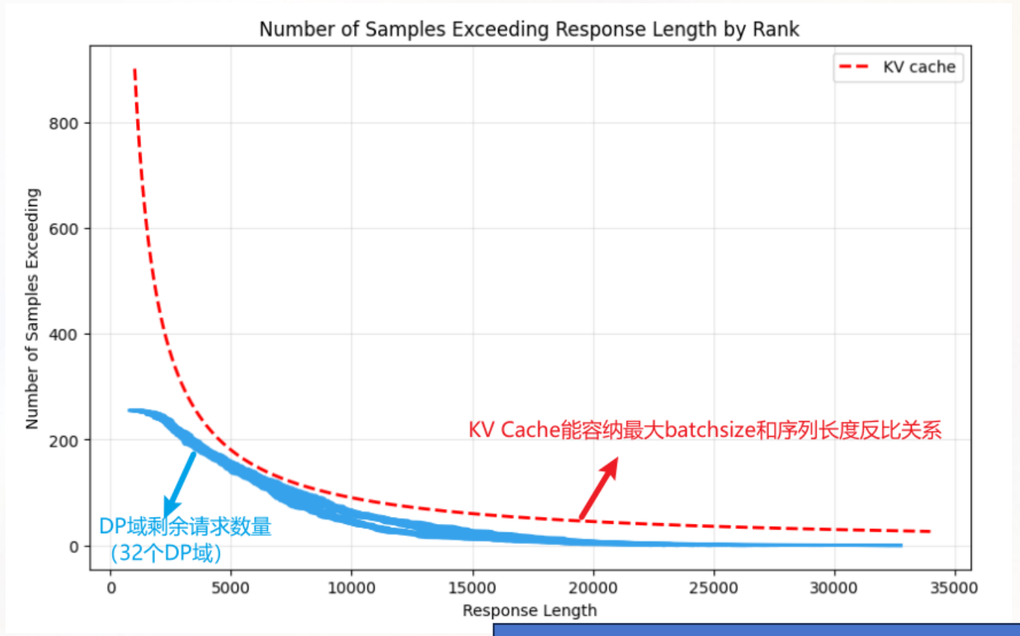
计算与调度优化

02

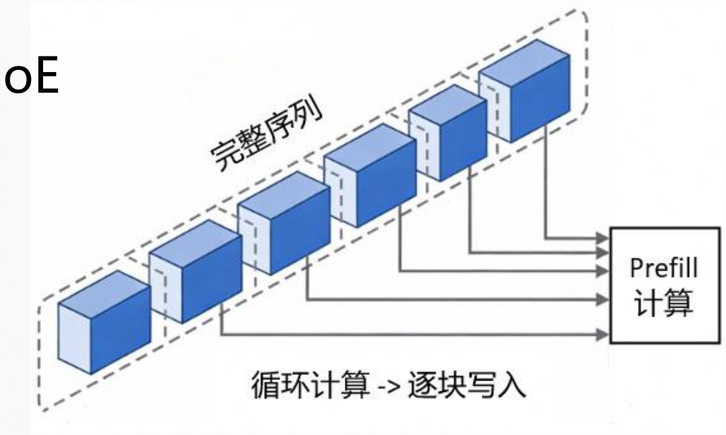
2.1 内存占用分析与部署策略选择

- **问题：**
  - 单序列32k占用KV Cache高达1.47GB
  - 理论上的单卡并发上限仅为20~30
- **优化方案：**
  1. 采用TP4切分，降低单卡权重占比（约7.09GB）
  2. Chunk MoE，降低Prefill阶段峰值内存
  3. 基于实际数据长度分布，设置更大的单卡并发数，batchsize=256





Chunk MoE

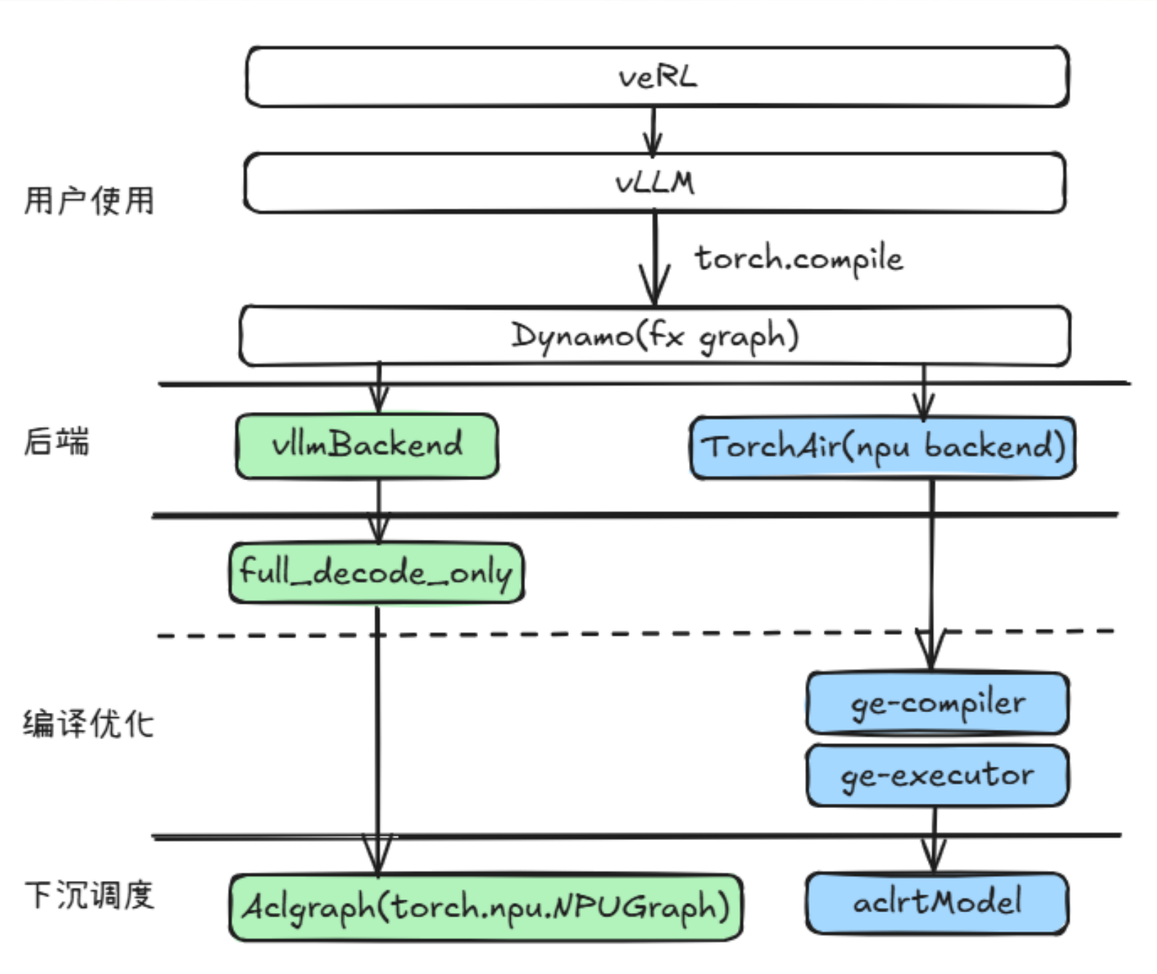


https://gitcode.com/cann

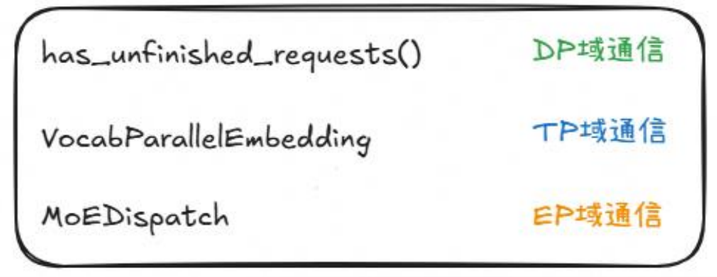
此处预留直播画面LOGO
不放置内容

2.2 计算&调度优化

- 使能图模式来减少CPU侧的调度开销



- Host Bound 导致通信延迟
  - 多个域之间的通信延迟源于CPU端
  - 可能由自动垃圾回收和线程竞争导致

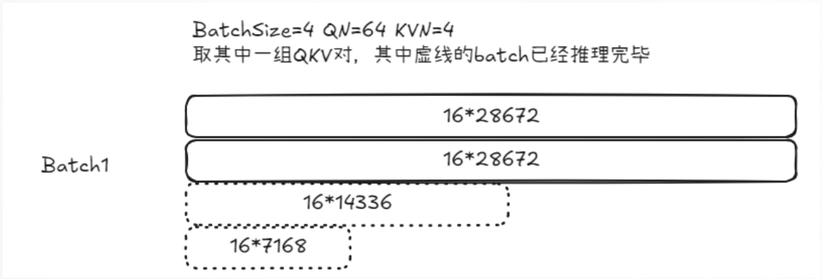


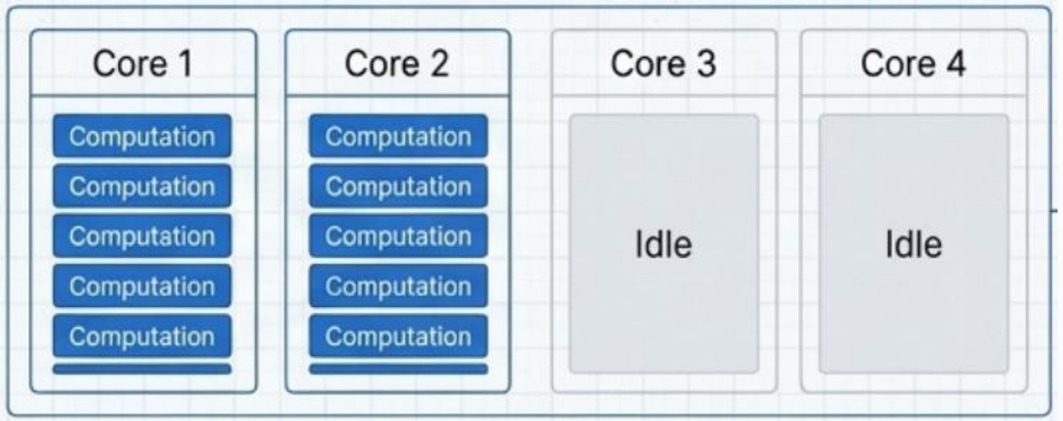
- 优化方案
  1. 在decode step期间临时禁用垃圾回收
  2. 绑核确保NUMA亲和性

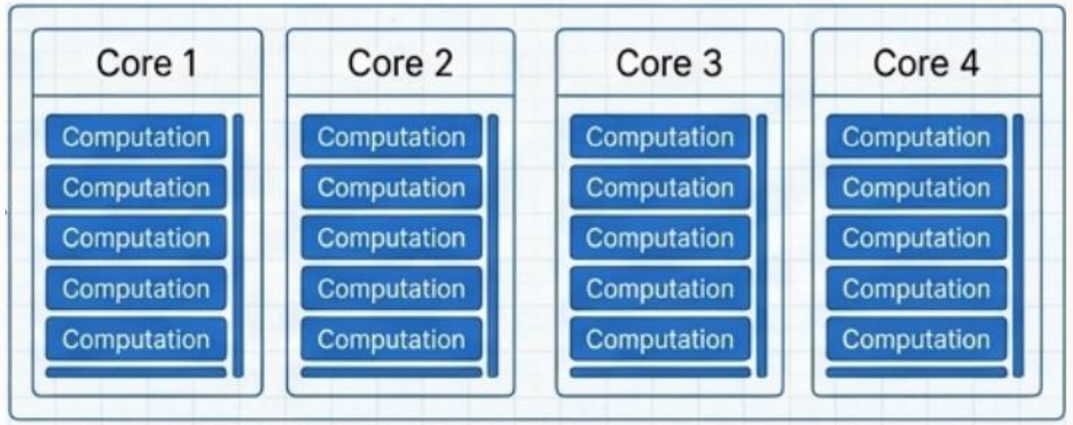
	单步decode forward 平均时间 (ms)	推理时间 (s)	收益
baseline	137.64	4851.98	-
关闭gc+添加绑核	124.66	4192.09	13.6%

2.3 FA算子负载均衡优化

- **优化方案：**通过**确定最优核数、优化块分配策略**，确保各核上的总计算开销基本均衡







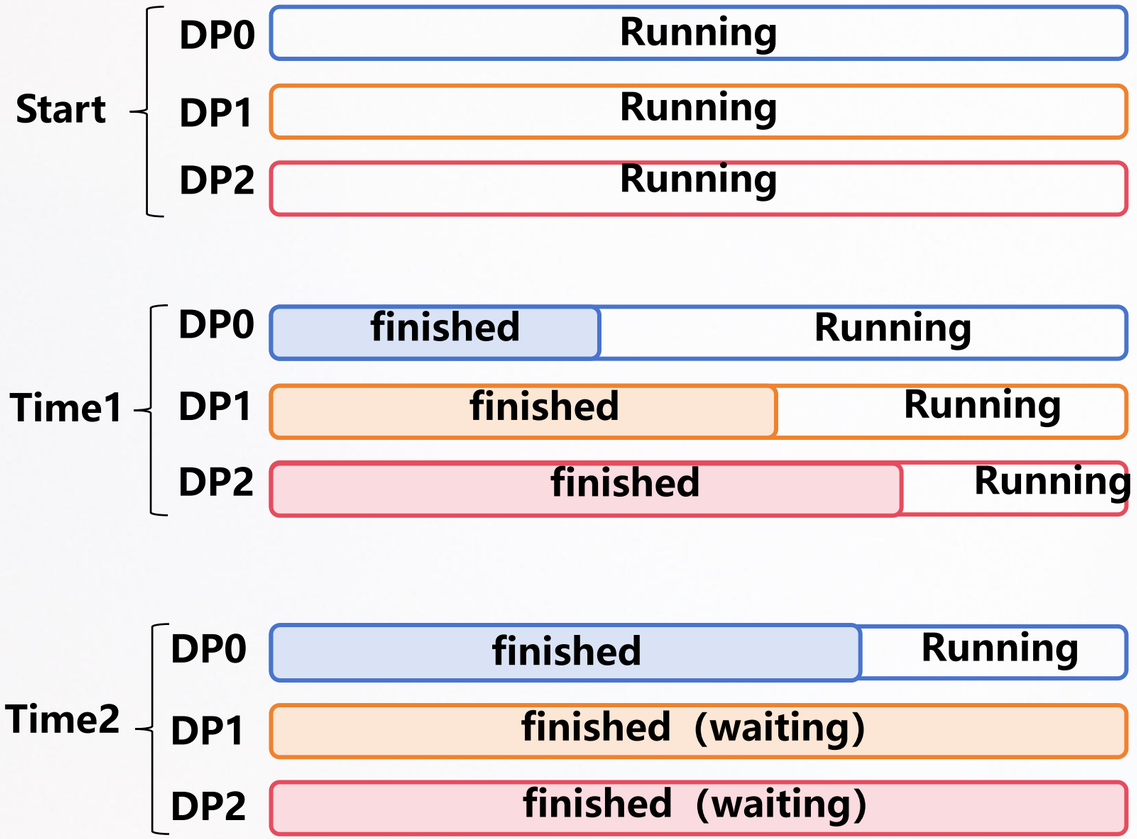
推理时间 4919s -> 4153s，减少15.6%↓　系统吞吐 105TPS -> 117TPS，提升11.4%↑

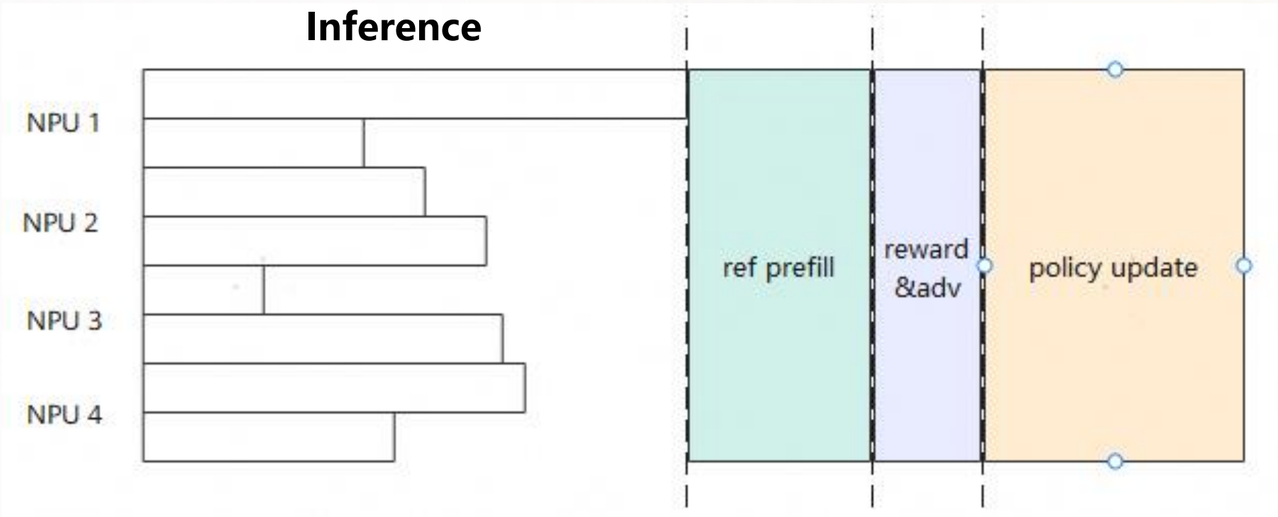
长序列负载均衡

03

长尾序列造成大规模计算资源浪费

- 长序列推理生成的Response长度存在显著的长尾分布，所有数据并行（DP）分组都必须等待最慢的分组完成。



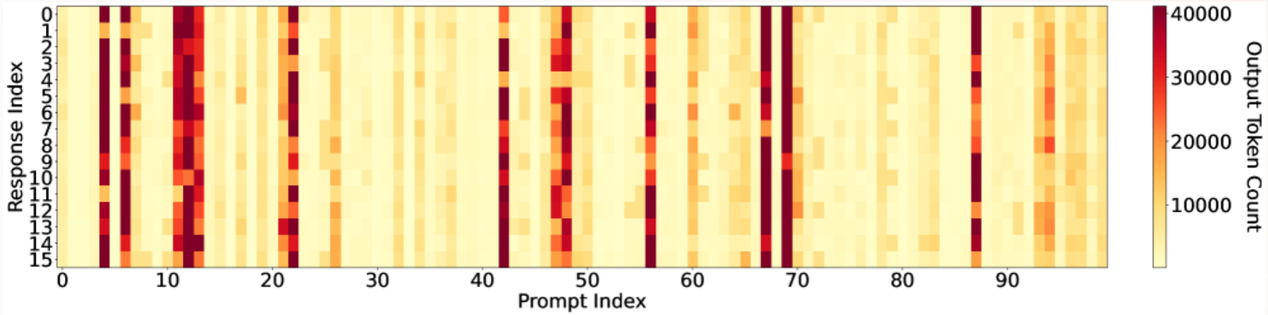


- 任务耗时不均：少数生成极长序列的任务会成为性能短板
- 算力资源浪费：短序列任务的计算节点完成后，进入长时间的闲置等待状态

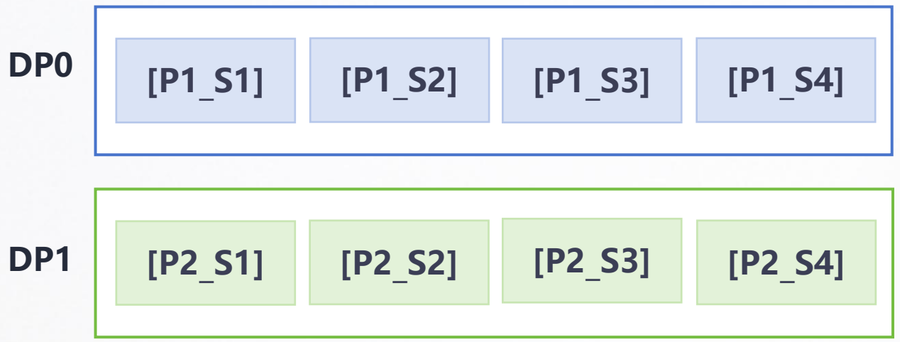
3.1 输入数据重排序 Data Balance

同一prompt的多个输出的长度有相关性 => 对**推理数据集进行均衡**可以缓解推理阶段长尾问题

- **优化方案：**
  - 将prompt 按采样数n复制后，改为间隔排布，使每个DP 组处理的 prompt 组合更加多样化和均衡；
  - 训练前重排回原始顺序。



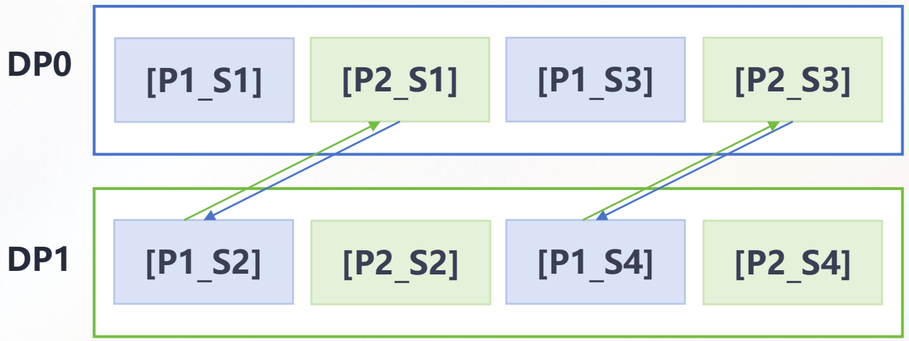
默认排布（Adjacent）



首DP空闲时间占比

64.84%

Data Rebalance排布（Interleaved）

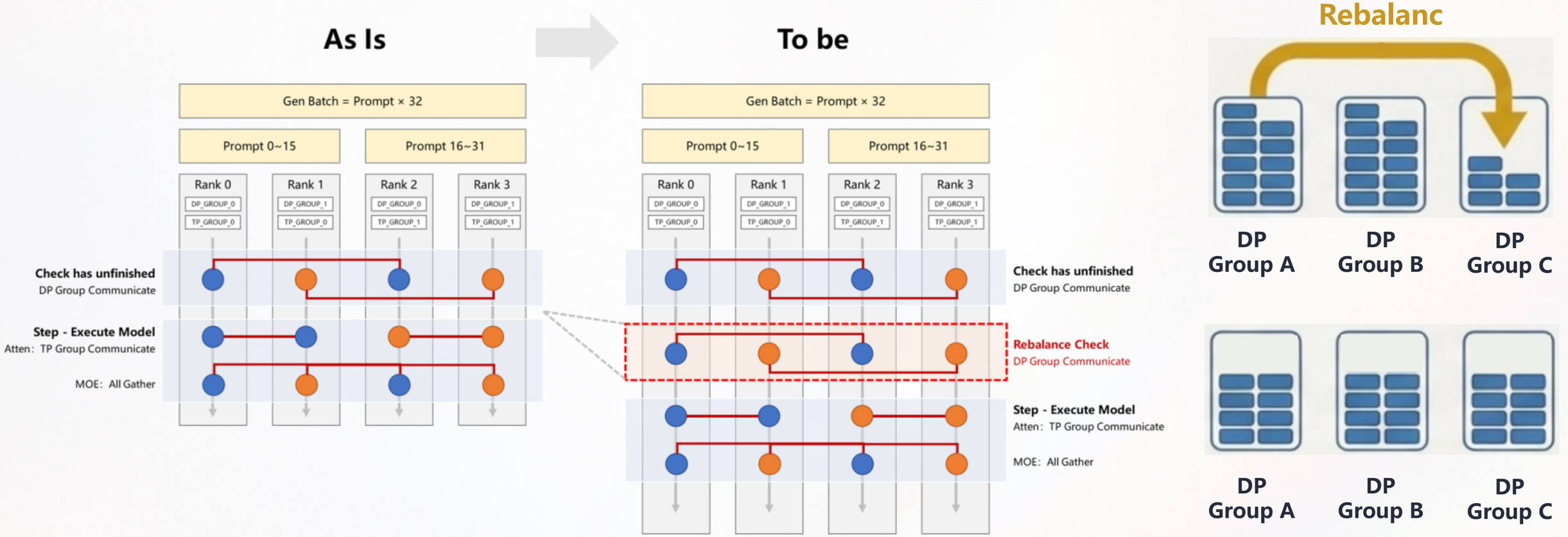


首DP空闲时间占比

24.83%

3.2 Rollout Rebalance

- **基于动态感知的推理时负载均衡**：在推理过程中动态感知全局负载，当检测到DP组间负载不均满足特定条件时，主动触发Rebalance（重均衡）调度，将任务在不同节点间迁移，从而恢复负载均衡，提升系统整体的吞吐和资源利用率。

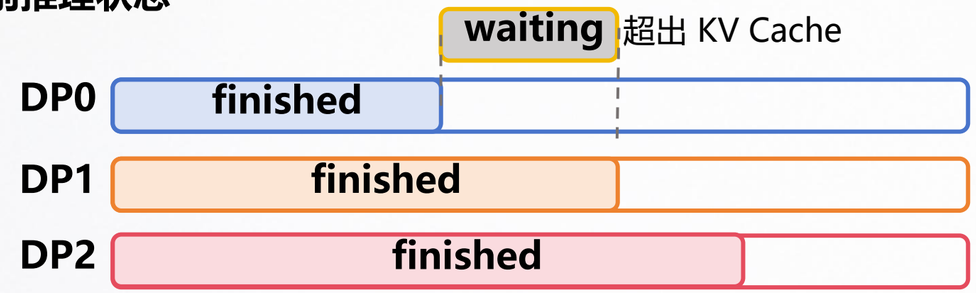


3.2 Rollout Rebalance

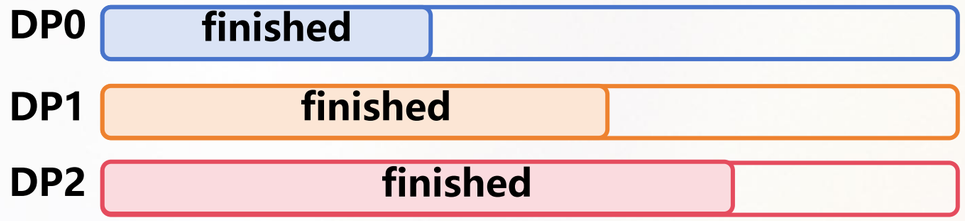
- **两阶段均衡调度策略**

1. 排队请求调度：优先将高负载节点的等待任务迁移至空闲节点，最大化资源利用率
2. 运行时请求调度：全局协同进行任务迁移，使所有DP能共同“降档”到更小的BatchSize，提升计算效率。

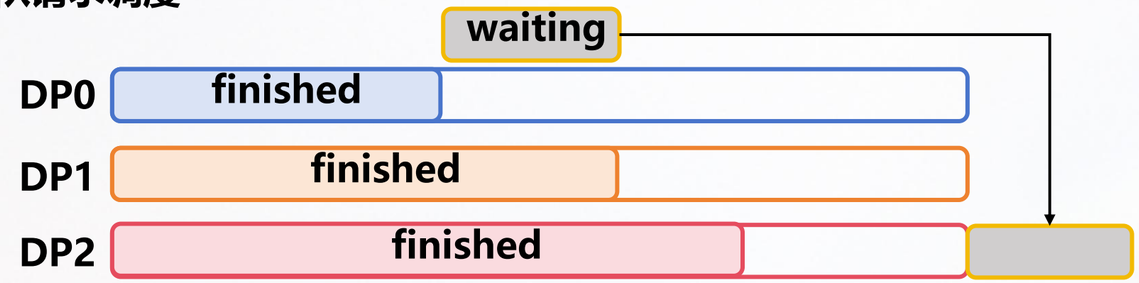
当前推理状态



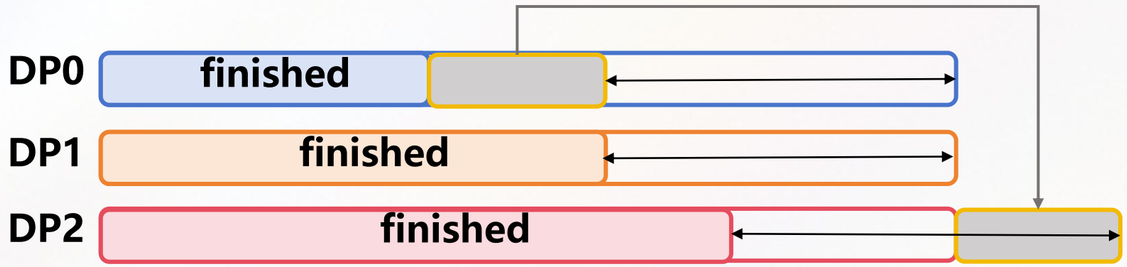
当前推理状态



排队请求调度

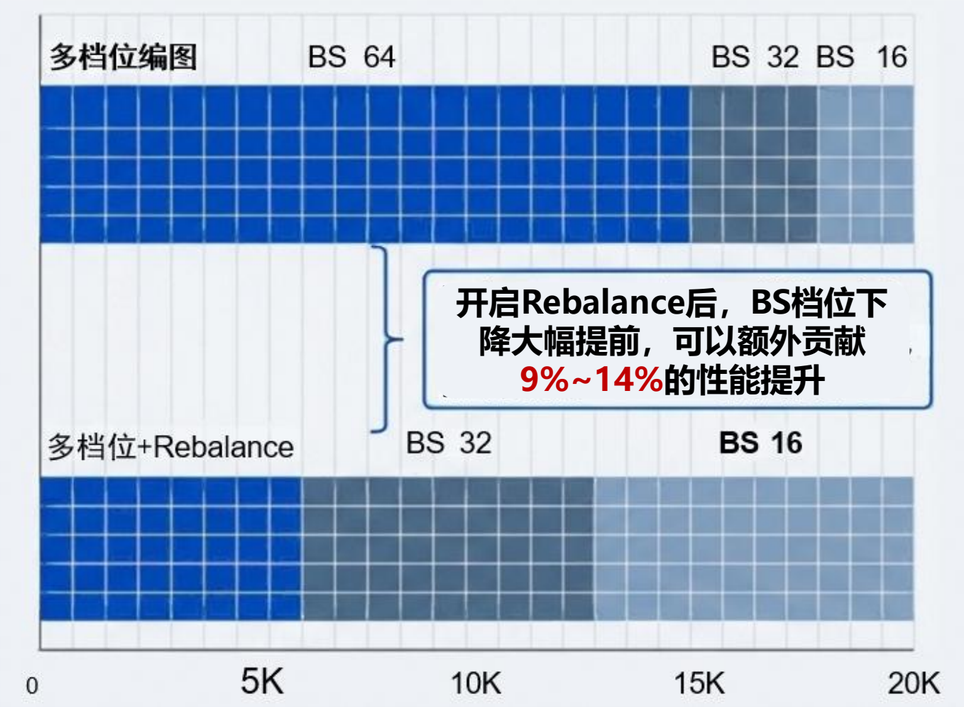


运行时请求调度

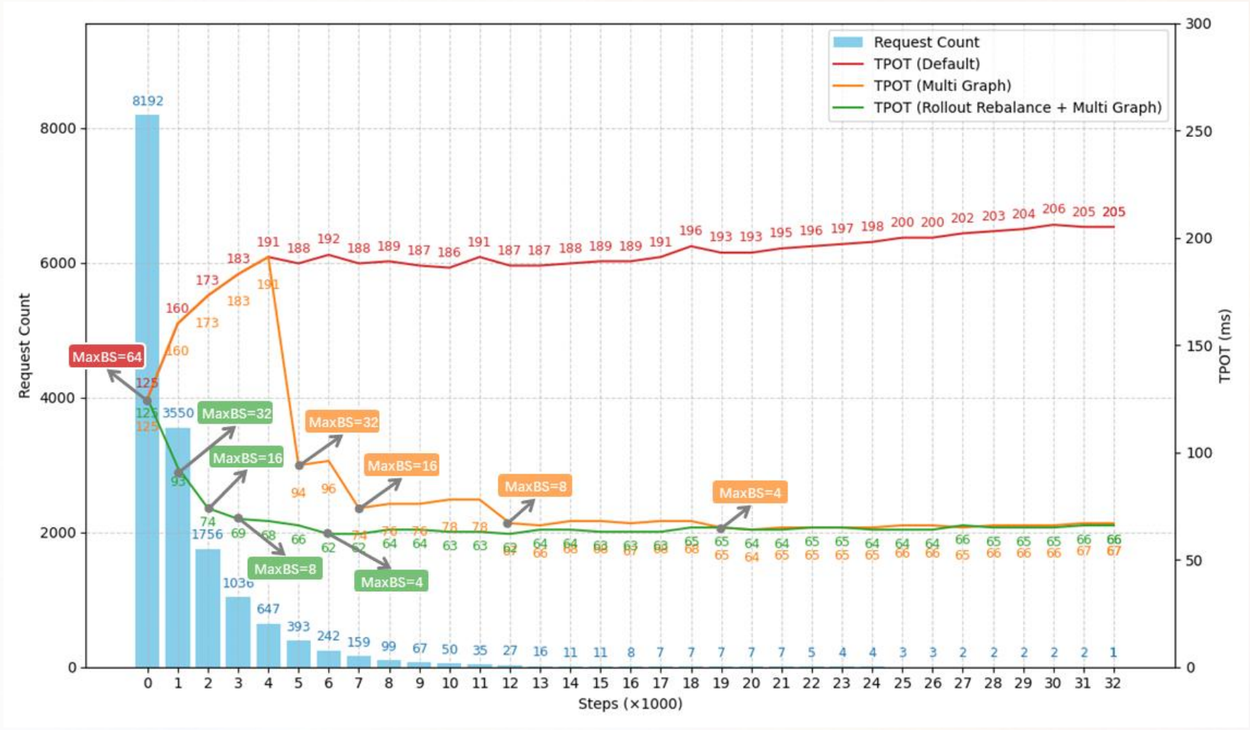


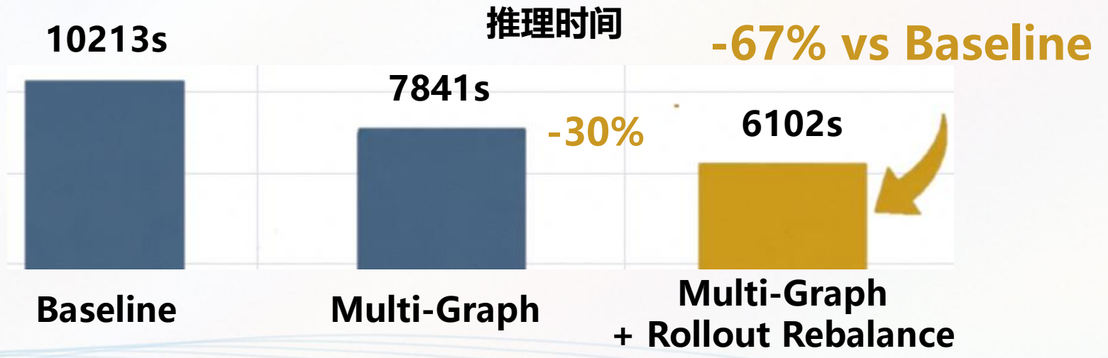
3.2 Rollout Rebalance

模拟实验



DeepSeek V3实验

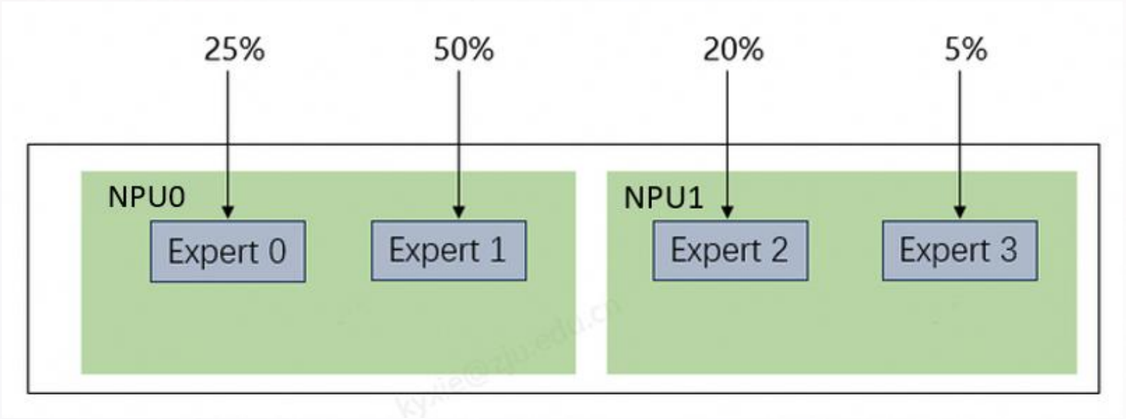




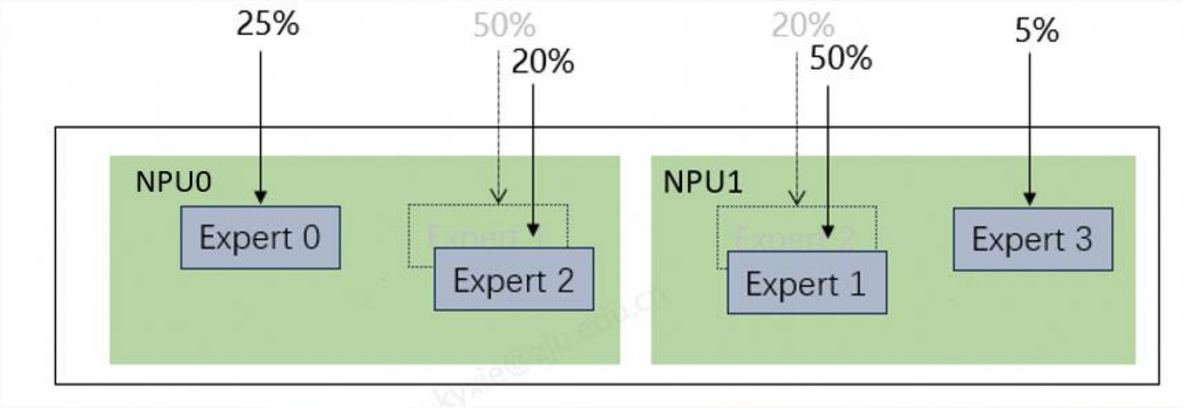
- Qwen3 235B 实测在多档位编图优化的基础上，Rollout Rebalance进一步带来**20~25%**的性能提升，相比基线提升**67%**。

3.3 Expert Parallelism Load Balance

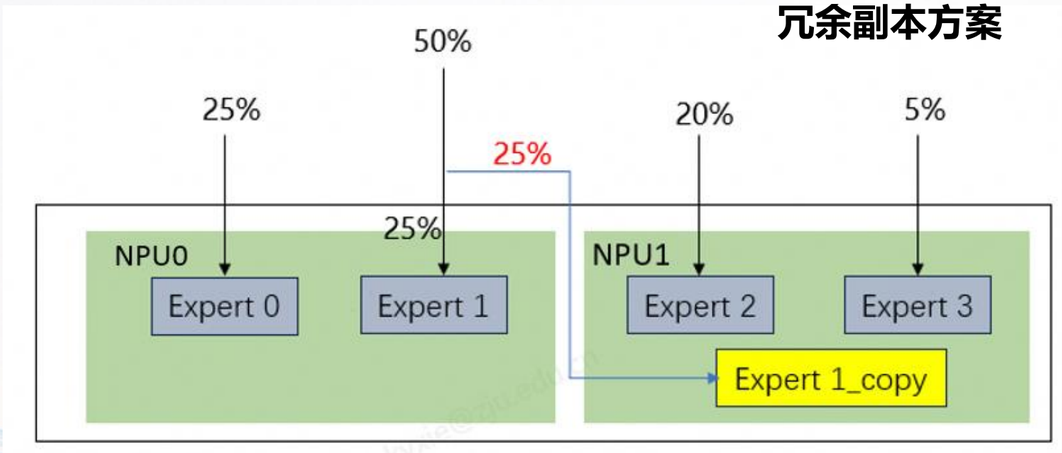
- 分层与全局两种模式负载均衡调度
- 实时负载估计与动态调整



全局重排序方案



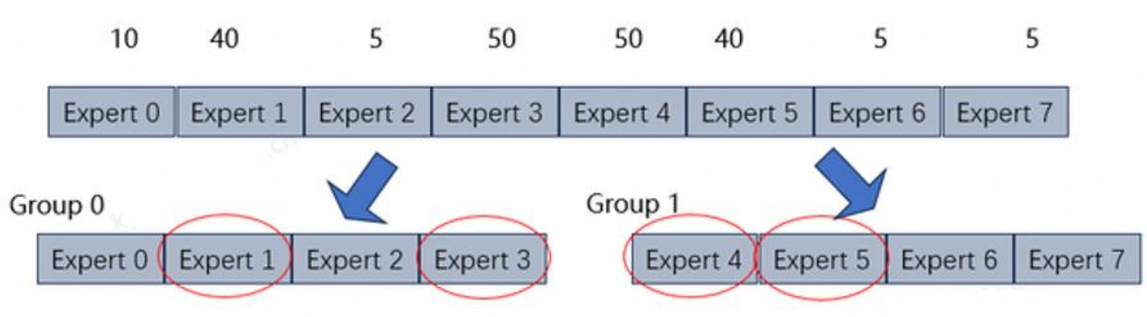
冗余副本方案



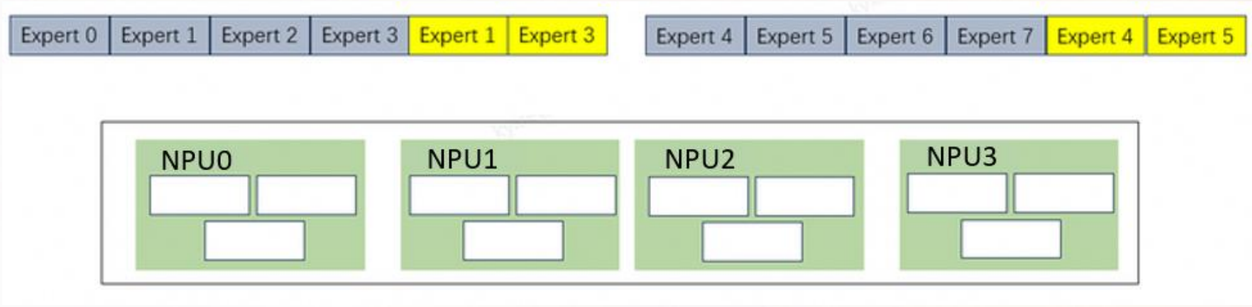
- **全局重排序方案**：能够让流量更加均匀，不需要额外内存，但需要调整全部的专家
- **冗余副本方案**：避免全局调整，但需要消耗额外内存空间

3.3 Expert Parallelism Load Balance

步骤1：将专家分为2组，并找到每组中热度最高的两个专家

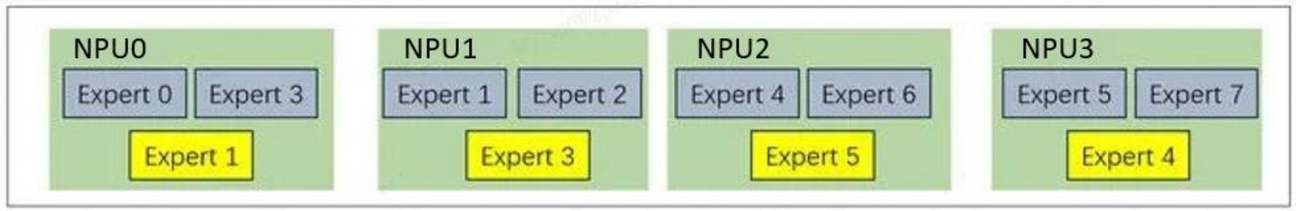


步骤2：复制每组中热度最高专家副本



- 构造极端负载不均衡场景，将选择的专家都固定为0-15号专家，此时rank0、rank1有较高负载，其余卡负载为0

步骤3：按照负载由高到低对专家进行排序，贪心装箱，使卡间负载均衡

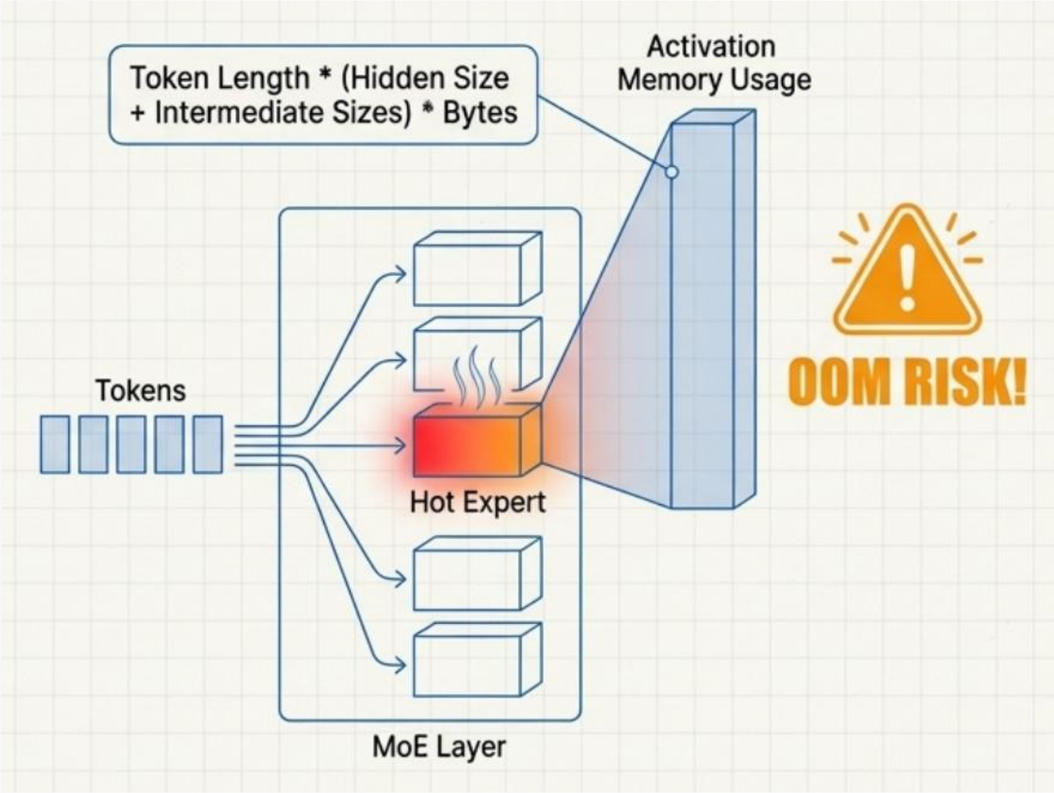


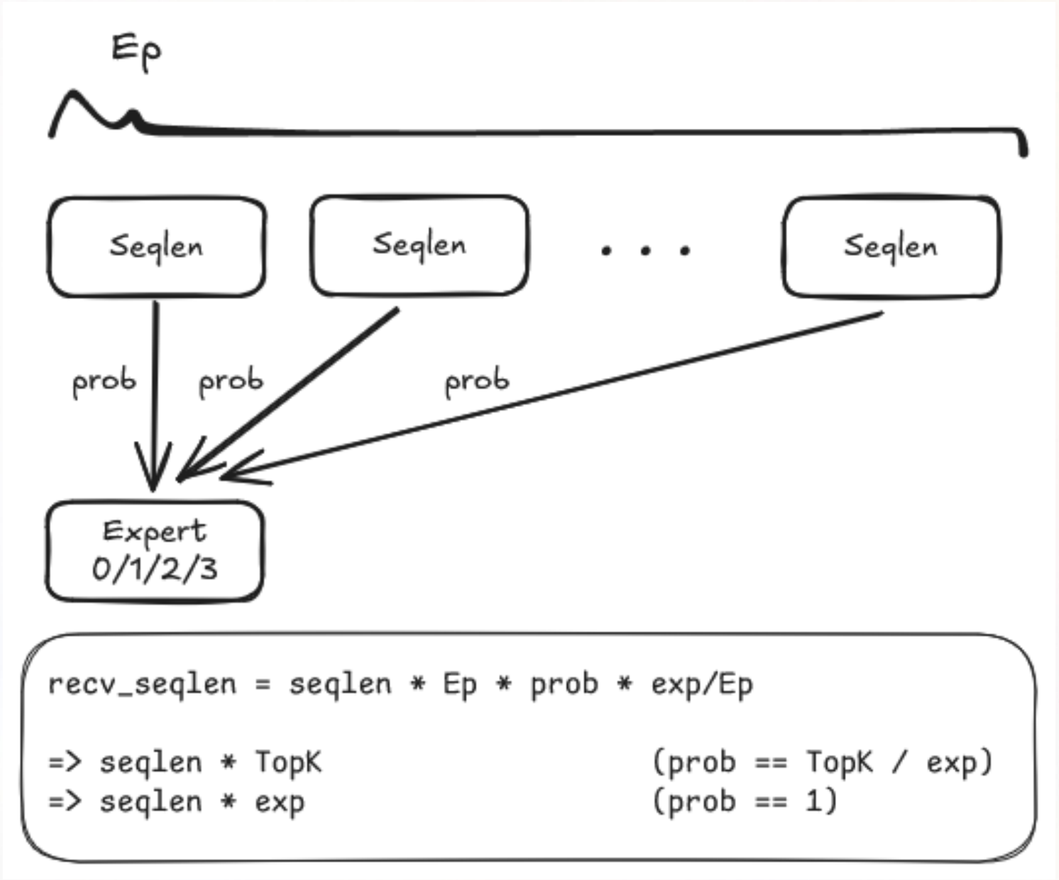
训练优化

04

4.1 内存占用分析和模型切分策略选择

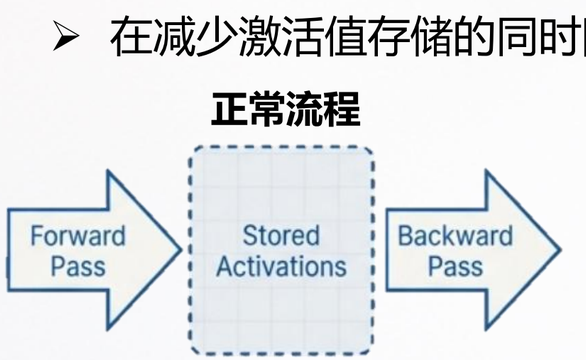
- 训练阶段动态内存与序列长度成正比，在长序列场景下激活内存往往成为显存瓶颈
- MoE架构下负载不均将会导致单层激活内存急剧上升

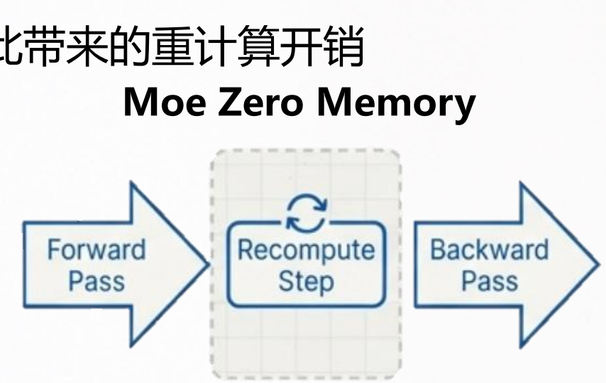


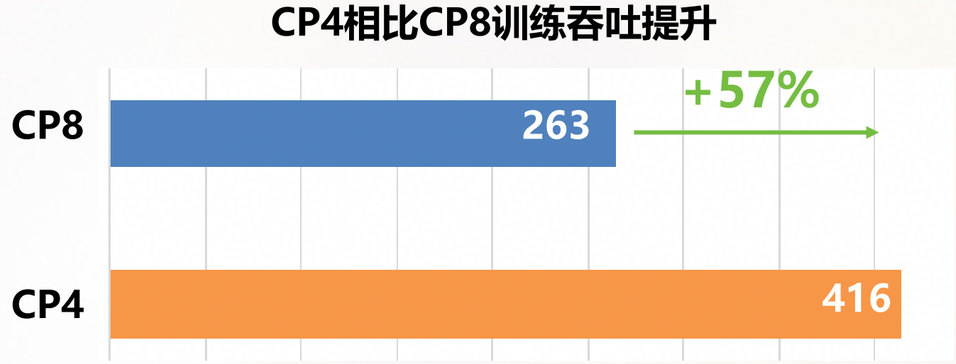


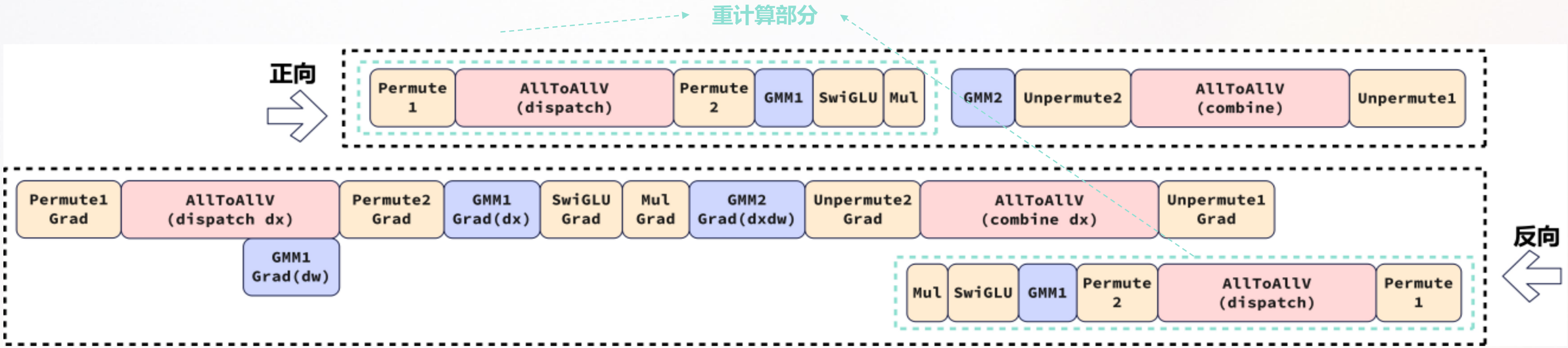
4.1 内存占用分析和模型切分策略选择

- MoE Zero Memory 核心思想：
  - 利用反向传播过程中原有的计算或通信，掩盖重计算过程中引入的通信和计算
  - 在减少激活值存储的同时降低因此带来的重计算开销









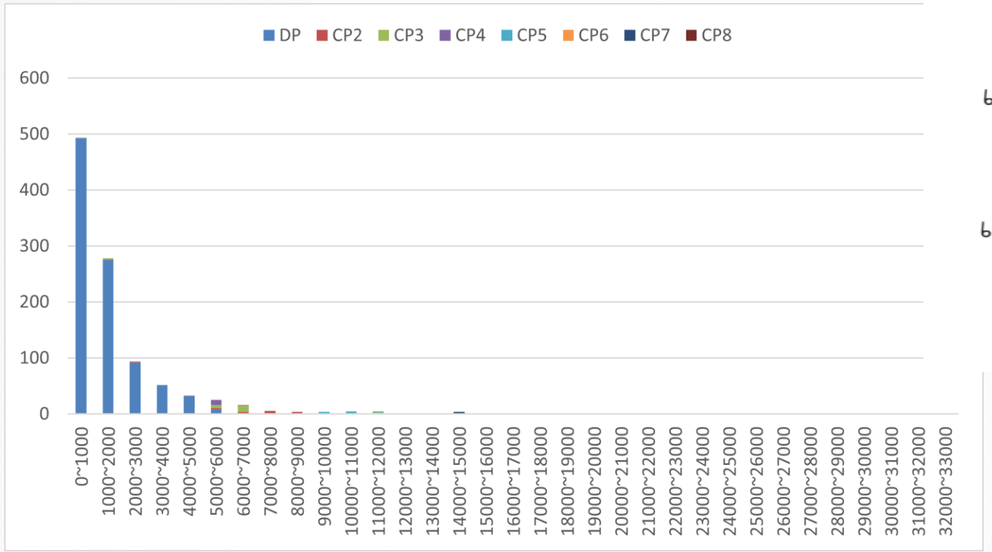
4.2 HDP(Hybrid Data Parallelism)混合数据并行

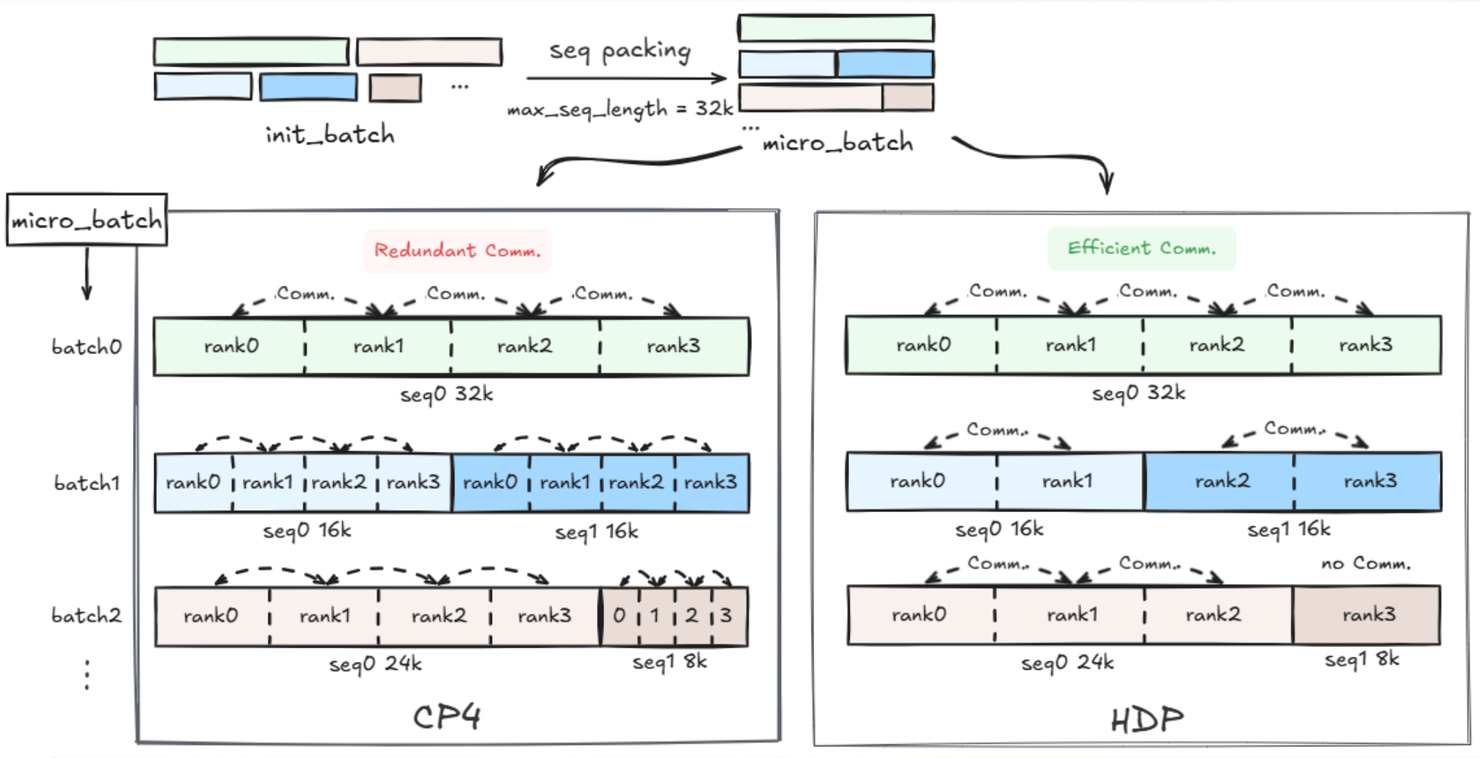
问题：当短序列与长序列一同被打包并进行CP处理时，短序列引入不必要的通信开销，造成计算资源浪费与训练效率下降

HDP核心思想：

- 融合DP和CP，根据序列长度动态选择最优并行
- 长序列：自动选择CP切分
- 短序列：划分到DP Group

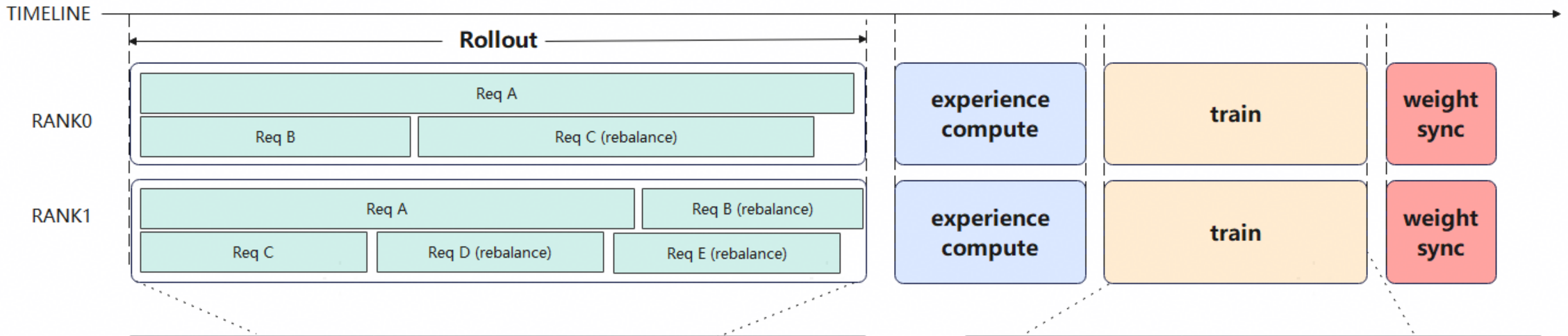
结论：收益与数据集长度分布强相关





HDP相比CP8，训练吞吐77.15提升到107.25，提升**39%**

总结



stage	优化方案
计算与调度	内存、切分策略
	图模式、HostBound优化
	FA算子负载均衡优化
负载均衡	Data Rebalance输入数据重排
	RolloutRebalance长尾负载不均优化
	Expert Parallelism Load Balance

stage	优化方案
训练	MindSpeed预训练优化复用
	内存、切分策略优化
	Hybrid Data Parallelism优化

未来的优化方向

近期探索

- **SAM 投机解码 (Speculative Adaptive Memory)：** 无需模型的投机策略，提升长序列RL推理吞吐。
- **KVP 切分：**将超长序列的KV Cache分布到多个NPU， 突破单卡显存瓶颈。

长远革新

- **新一代注意力机制：**线性注意力 (Linear Attention)、 稀疏注意力 (Sparse Attention) 等，从根本上解决二次复杂度问题。
- **下一代模型架构：**Qwen-Next 等新架构的预 训练优化，探索混合注意力机制的演进。

cann-recipes-train

- 本实践已经开源至 gitcode 代码仓：https://gitcode.com/cann/cann-recipes-train/blob/master/llm_rl/qwen3/README.md

RL后训练执行

在本样例代码根目录下启动Qwen3-235B-A22B的RL后训练。

```
# 请注意，以下bash启动脚本中的内容需要手动配置
# source脚本路径：  根据实际CANN安装目录调整
# MASTER_ADDR：     ray集群主节点的IP地址，每个节点的脚本配置一致
# SOCKET_IFNAME：   集群中各节点自己的网卡名，可通过ifconfig命令查看
# VLLM_DP_SIZE：    推理阶段DP配置，按推理模型切分和总卡数计算

bash ray_start_npu.sh TRAIN_SCRIPT ENV_SCRIPT
# 示例： bash ray_start_npu.sh ./internal/train_grpo_qwen3_235b_128die_random_init.sh ./internal/qwen3_235b_env.sh
# 如果不需要额外的环境变量配置，则不需要该参数，示例：bash ray_start_npu.sh ./internal/train_grpo_qwen3_32b_32die_true_weight.sh
```

可在 `ray_start_npu.sh` 启动训练时添加参数，实现随机权重训练GRPO算法、真实权重训练GRPO算法、真实权重训练DAPO算法，对应修改如下：

基础模型	训练	训练启动脚本	训练配置脚本	环境变量配置脚本
Qwen3-235B-A22B	随机权重训练GRPO算法	ray_start_npu.sh	./internal/train_grpo_qwen3_235b_128die_random_init.sh	./internal/qwen3_235b_env.sh
Qwen3-235B-A22B	真实权重训练GRPO算法	ray_start_npu.sh	./internal/train_grpo_qwen3_235b_128die_true_weight.sh	./internal/qwen3_235b_env.sh
Qwen3-235B-A22B	真实权重训练DAPO算法	ray_start_npu.sh	./internal/train_dapo_qwen3_235b_128die_true_weight.sh	./internal/qwen3_235b_env.sh
Qwen3-32B	真实权重训练GRPO算法	ray_start_npu.sh	./internal/train_grpo_qwen3_32b_32die_true_weight.sh	-
Qwen3-32B	真实权重训练DAPO算法	ray_start_npu.sh	./internal/train_dapo_qwen3_32b_32die_true_weight.sh	-

目录

- Qwen3系列模型 RL训练...
  - 概述
    - Qwen3-235B-A22B
    - Qwen3-32B
  - 硬件要求
  - 基于Dockerfile构建环境
  - 数据集准备
  - 模型权重准备
    - Qwen3-235B-A22B
    - Qwen3-32B
  - RL后训练执行
  - 附录
    - 文件说明
    - 手动准备环境

Preview Code 修改追溯 76.14 KB 历史 对比 源码 复制 IDE

Qwen3-235B 32K长序列RL训练优化实践

本文系统总结了基于verl框架与A3集群的Qwen3-235B模型32K长序列强化学习训练优化实践，详细介绍了对RL长序列场景下特有的**长尾序列负载不均、显存瓶颈**等问题进行的多项优化。通过文中所述技术，本实践成功将系统吞吐从个位数大幅度提升至122TPS/卡。同时，本文还与近期发布的Seer长序列RL训练系统中的相关优化思路进行了对比分析，揭示了技术路线上的共识与差异。

1. 前言

1.1 背景

长序列处理能力是推动大型语言模型走向实用化的关键。以具备32K上下文长度处理能力的Qwen3-235B模型为例，它能够有效应对长篇文章解析、复杂多轮对话和大规模代码编写等现实任务，在金融研报分析、法律条文解读以及科研文献理解等多个领域都展现出重要的应用价值。因此，如何高效处理训练与推理过程中的长序列数据，已成为影响模型部署效率的关键因素。

特别地，在强化学习场景中，模型不仅需要在超长上下文中进行复杂的思维链（CoT）推理与奖励训练，还面临**HBM内存消耗随序列长度线性增长、长尾序列分布导致负载不均等挑战**，这无疑对训练与推理系统的效率及稳定性提出了更高要求。

针对上述问题，本实践在前序工作基础上，沿用verl的前端，采用vLLM-Ascend作为推理引擎，MindSpeed作为训练引擎，在64卡A3集群上对Qwen3-235B模型进行了32K长序列GRPO算法的强化学习训练优化。本实践系统性地从**模型切分策略、长序列负载均衡、计算与通信效率**等多个维度入手，显著提升了长序列训练吞吐，最终实现122TPS/卡的系统性能。

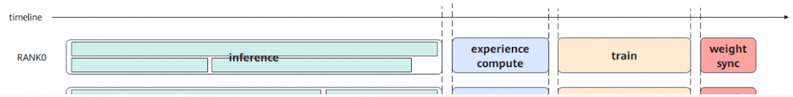
值得一提的是，近期月之暗面&清华发布的论文《Seer: Online Context Learning for Fast Synchronous LLM Reinforcement Learning》中，也提出了多项针对长序列强化学习的优化策略，与本实践的思路高度契合。本文也对其中的一部分策略进行了对比分析。

1.2 优化实践总览

本实践对verl开源代码进行了以下关键功能的适配与优化：

- 训练引擎适配：verl框架当前已经原生支持MindSpeed训练引擎，但在开启moe_alltoall_overlap后会出现专家权重加载异常问题，本实践针对该问题进行了修复与适配。
- 推理引擎优化：推理过程中的sleep模式依赖torch_npu的虚拟内存动态切换，目前采用前序工作的方法实现推理权重及kv_cache的加载与卸载。
- 框架适配：参考前序工作修改了verl框架，在GRPO这类on-policy训练算法中实现了old_log_prob免计算。

本实践针对长序列推理和训练中的性能瓶颈与内存占用，进行了系统性地优化，具体方法如下图所示：



目录

- Qwen3-235B 32K长序...
  - 1. 前言
    - 1.1 背景
    - 1.2 优化实践总览
    - 1.3 详细性能结果
      - 1.3.1 GRPO算法
      - 1.3.2 DAPO算法
  - 2. 推理优化
    - 2.1 内存占用分析与...
    - 2.2 计算与调度优化
      - 2.2.1 切分策略优...
        - 现象分析
        - 优化方案
        - 优化效果
      - 2.2.2 图模式路径...
        - 现象分析
        - 方案选择
        - 优化效果
      - 2.2.3 FA算子负...
        - 现象分析
        - 优化方案
        - 优化效果
      - 2.2.4 Decode阶...
        - 现象分析
        - 优化方案
        - 优化效果
    - 2.3 负载均衡优化
      - 2.3.1 RL rollout...

https://gitcode.com/cann/cann-recipes-train/blob/master/llm_rl/qwen3/README.md

https://gitcode.com/cann/cann-recipes-train/blob/master/docs/llm_rl/qwen3_235B_32k_longseq_rl_train_optimization.md

cann-recipes系列仓库

- cann-recipes希望通过提供拿来即用的算法模型样例，给到开发者最需要的指导：如何快速上手？如何复现业界SOTA模型？如何榨干NPU性能？
- 内容涵盖：
  - 全场景算法样例：覆盖大模型、多模态、空间智能、具身智能各领域的算法样例
  - 高性能模型复现：基于CANN深度优化的主流模型训练与推理脚本
  - 特性优秀实践：针对CANN新特性的使用指南与性能调优技巧
- 目前已开源四个仓如下，正在紧锣密鼓持续建设中，清程极智、中科院等生态开发者也在积极贡献

仓名	定位	典型样例
cann-recipes-infer	推理样例	DSv3.2/Kimi-K2-thinking 0-day支持发布及高性能优化 HunyuanVideo/Wan2.2发布
cann-recipes-train	训练样例	Qwen3-MoE 32K长序列/DS-R1 RL训练样例
cann-recipes-embodied-intelligence	具身智能样例	Pi0推理样例
cann-recipes-spatial-intelligence	空间智能样例	Hunyuan3D/VGGT推理

cann-recipes-sig小组

cann-recipes交流群

欢迎广大开发者体验并参与贡献，如有疑问可通过issue、SIG或者cann-recipes交流群联系我们！

Thanks!

访问CANN开源社区

关注昇腾CANN公众号