

# 面向中小规模昇腾算力的大模型高性能推理实践

唐适之 清程极智联合创始人

<https://gitcode.com/cann>

清程极智 Qingcheng.AI

**CANN**

# 大小皆功：不同部署规模应对不同挑战

- 针对 DeepSeek-R1 等 MoE 模型的性能优化是持续的研究热点
- 社区在超节点等大规模部署场景下有许多优秀成果，但中小规模部署也代表了大量实际需求
- 针对不同规模的部署形态，需要差异化的优化策略
  - 百卡规模 → PD 分离、以 DP+EP 为主的混合并行、专家负载均衡
  - 32 卡规模 → PD 混布、以 DP+EP 为主的混合并行、专家负载均衡
  - 16 卡规模 → PD 混布、TP+PP 并行
  - 单机规模 → PD 混布、TP 并行、极致量化
- 清程极智针对 A2 单机/中等规模集群进行了针对性开发优化
  - 面向 A2 4 机 32 卡：针对中等规模部署 DeepSeek-R1 的优化
  - 面向 A2 单机 8 卡：基于 FP4 权重量化实现单机推理 DeepSeek-R1

# 目录

Part 1: 面向 A2 单机 8 卡: 基于 FP4 权重量化实现单机推理 DeepSeek-R1

Part 2: 面向 A2 4 机 32 卡: 针对中等规模部署 DeepSeek-R1 的优化

# A2 单机 8 卡问题分析：显存占用

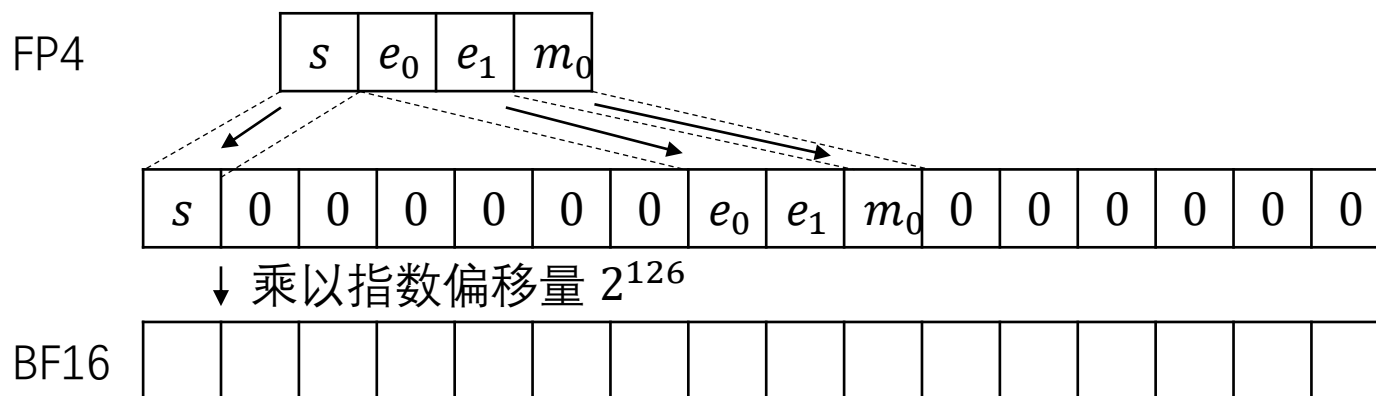
- 单机低门槛大模型部署存在广泛需求
- DeepSeek-R1 参数量 671B，A2 单机 8 卡仅有 512GB 显存
- A2 单机 8 卡部署 DeepSeek 需要 4 位及以下的极致量化
- FP4 量化是量化算法的最新进展之一，精度较 INT4 更优

# A2 单机 8 卡问题分析：访存 vs 计算

- 小规模推理计算密度低：单个参数被加载后，与其进行计算的请求数有限
- 性能受限于**访存**而非计算：尽管 910B 无硬件 FP4，但软件 FP4 同样合适
- 硬件 INT8 ( INT8 W8A8 ) :
  - 硬件计算 INT8 矩阵乘
  - **访存量小**，计算量小
- 软件 FP4 ( FP4 W4A16 ) :
  - 软件将 FP4 动态转换为 BF16，然后硬件计算 BF16 矩阵乘
  - **访存量更小**，计算量大
- 本次分享内容：基于 CANN 生态实现软 FP4 矩阵乘算子

# A2 单机 8 卡：软件 FP4 转换算法

- 使用两步计算完成 IEEE 754 窄浮点数转宽浮点数
  - 将符号位、指数位、位数尾置于正确位置
  - 利用浮点数乘法校正指数位的偏差
- 无需考虑规格数/非规格数的差别



# A2 单机 8 卡：软件 FP4 转换算法

- 利用整数位运算在算子加载权重时从 FP4 转换为 BF16

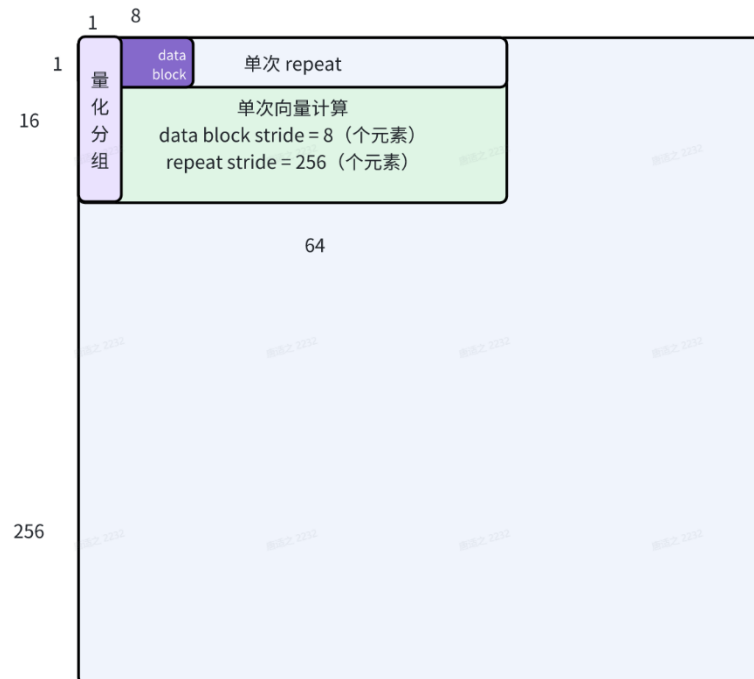


# A2 单机 8 卡：昇腾亲和的高效分组 scaling

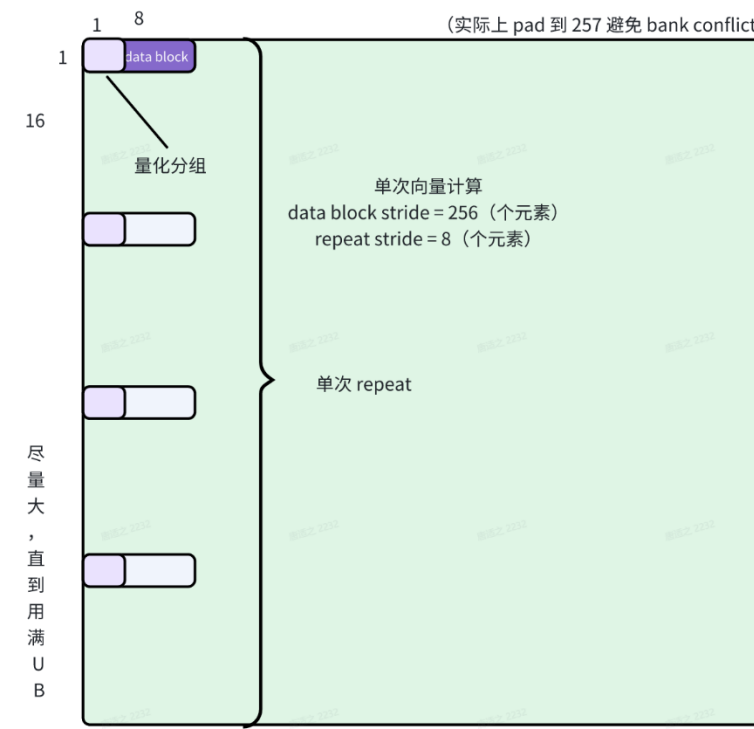
- 分组 scaling 已成为低位宽量化算法的主流选择
  - NVFP4 需要  $16 \times 1$  分组，每组乘以不同系数
- 所乘系数组内相同，组间不同，难以填充昇腾向量指令，朴素实现性能有限
- 通过数据排布优化分组 scaling
  - 昇腾向量指令可一次性处理  $\text{shape}=(r, 8, 8)$ 、 $\text{stride}=(x, y, 1)$  的高维数据
  - 通过预处理重排数据内存布局扩大每次处理的范围

<https://gitcode.com/cann>

清程极智 Qingcheng.AI



朴素向量实现



重排后的向量实现

# A2 单机 8 卡：纯 AIV 实现矩阵乘

- 软 FP4 矩阵乘由 FP4 转换与矩阵乘两步构成
  - AIV：能用于 FP4 转换与矩阵乘，性能低
  - AIC：只能用于矩阵乘，性能高
- 算子耗时分为计算耗时与访存耗时，小规模部署时访存耗时更为关键
  - 读取（访存）+ AIV 转换（计算）+ AIV-AIC 传递（访存）+ AIC 矩阵乘（计算）+ 写回（访存）：计算耗时稍低，访存耗时显著高
  - 读取（访存）+ AIV 转换（计算）+ AIV 矩阵乘（计算）+ 写回（访存）：计算耗时稍高，访存耗时显著低

## A2 单机 8 卡：实现效果与社区回馈

- 基于 Ascend C “工程化算子开发” 模板开发
- 参考 CANN 社区 cann-ops-adv 的矩阵乘 API
- 性能：

Batch size	1	2	4	8
Output TPS	17.76	26.95	55.97	87.00

DeepSeek R1 FP4 昇腾单机 8 卡测试性能（Input length 128, output length 1024）

- 算子已开源至 <https://github.com/QingCheng-AI/ascend-kernel>
- 可通过赤兔（<https://github.com/thu-pacman/chitu>）开源推理引擎使用

# 目录

Part 1: 面向 A2 单机 8 卡: 基于 FP4 权重量化实现单机推理 DeepSeek-R1

Part 2: 面向 A2 4 机 32 卡: 针对中等规模部署 DeepSeek-R1 的优化

# A2 4 机 32 卡：问题分析

- 适合 INT8 量化的 DeepSeek-R1
- 适合 PD 混布，不适合 PD 分离
  - A2 单机 8 卡总显存 512GB，DeepSeek-R1 INT8 量化模型约占 643GB
  - 若按机器切分 PD 分离，PD 分离只有“2 机 P 实例 + 2 机 D 实例”一种选择
  - PD 分离时 P 和 D 比例欠优
  - PD 分离时 P 或 D 节点各自受限于显存，并发容量有限
- PD 混布时，空余显存可用于如下优化
  - Attention DP + FFN EP：复制 attention 部分参数，减少通信
  - 专家负载均衡：复制热点专家、调整专家顺序，减少空转等待时间

# A2 4 机 32 卡：实现路径

- 基于社区推理引擎 **OmnInfer** ( <https://gitee.com/omni-ai/omniinfer> )
- **OmnInfer** 紧密跟踪 **CANN** 最新算子，充分发挥 **CANN** 的底座能力：
  - 例如 MLA 融合算子 `mha_prolog_v2`
  - 例如 DP+EP 通信算子 `torch_npu.npu_moe_distribute_dispatch/combine_v2`
- **OmnInfer** 已有深入调优，但社区已有代码主要面向大规模场景，必须使用 **PD** 分离
- 本次分享内容：基于 **OmnInfer** 新增面向中等规模场景的针对性改进
  - 新增 **PD** 混布请求服务/调度/任务逻辑
  - 泛化优化逻辑，支持 **Qwen3** 系列 **MoE** 模型

# A2 4 机 32 卡：实现效果和社区回馈

- 性能：Deepseek-R1 (int8) 平均每卡 output TPS 达到 303.67
  - 测试条件：global batch size 1024, input length 128, output length 1024
- 已将改进内容回馈至 OmnilInfer ( review 中 )
- 并将可运行实例开源至 CANN 社区 cann-recpies-infer 仓库 ( review 中 )

# CANN社区CANN be X计划

can be	can do	收获	社区支持
首席体验官	代码&文档的bug hunting	贡献榜单曝光、社区勋章	技术支持
赏金猎人	揭榜社区任务	任务奖品	奖品
城市主理人	创建同城CANN技术交流俱乐部，城市Meetup组织	社区勋章、组织能力提升，成为领域KOL	场地、活动奖品、技术专家资源
校园合伙人	创建校园级CANN社团、组织社区活动开展	社区勋章、组织能力提升，成为领域KOL	活动奖品、技术专家资源、场地等
布道师	CANN社区及项目推广，技术与经验分享	社区勋章、技术传播能力提升，成为领域KOL	技术支持、峰会/论坛邀请函
金牌讲师	基础课程建设与授课	社区勋章、技术传播能力提升，成为领域KOL	官方传播矩阵、技术支持
社区建筑师	代码贡献	社区勋章、贡献榜单、任务奖品	技术支持、奖品
赞助商	场地、活动基金等资源支持	社区公示	官方传播矩阵

# CANN Meetup

北京站

扫码反馈问卷，参与抽奖



\*抽奖活动仅限2025-11-15Meetup北京站线下活动