PR #26384 完整报告

sgl-project/sglang

[Docs] GLM-4.7 cookbook: add NVIDIA Blackwell (B200, GB200) + NVFP4 sections

合并时间：2026-06-02 11:47

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26384

执行摘要

该 PR 为 GLM-4.7 的 cookbook 文档和部署命令生成器添加了 NVIDIA Blackwell (B200/GB200) GPU 和 NVFP4 4-bit 量化的一级支持，同时修复了现有文档中无效的 --reasoning-parser glm47 参数错误。通过引入硬件兼容性矩阵（SUPPORT），实现了部署配置的自动约束与 fallback，降低了用户配置出错风险。

功能与动机

PR 作者指出当前 GLM-4.7 cookbook 仅以 AMD GPU 为一等目标，但 nvidia/GLM-4.7-NVFP4 权重已是 Blackwell 上推荐的服务方式；同时现有文档使用的 --reasoning-parser glm47 在 v0.5.12 中仅对 tool-call 注册，推理侧仅 glm45 有效，导致用户直接复制命令会报 argparse 错误。

实现拆解

1. 更新 cookbook MDX 文档（GLM-4.7.mdx） - 调整 Model Introduction，加入 NVFP4 权重和 Blackwell 硬件说明。 - 新增 §3.2 硬件×权重类型兼容性表格（B200/GB200/H200/AMD 各芯片支持的量化及 TP 范围）。 - 添加 Docker 镜像表，按硬件平台区分不同镜像。 - 新增 Throughput 4K/1K 基准测试场景并嵌入 B200/GB200 的实测数据。 - 修复所有出现 --reasoning-parser glm47 的引用为 glm45。
2. 重构部署命令生成器 JSX 组件（glm-47-deployment.jsx） - 在硬件选项中添加 B200（默认）、GB200、H200，AMD 芯片调整为非默认。 - 量化选项新增 NVFP4（默认），FP8/BF16 保持可选。 - 新增“Number of GPUs”选择器（2/4/8，默认 4）。 - 定义 SUPPORT 常量作为硬件兼容性单一事实源，通过 quantSupported、allowedTps、firstSupportedQuant 辅助函数驱动 UI 控件的有效范围，当用户选择不支持的量化或 TP 时静默 fallback 到最近的有效值。
3. 修复命令生成逻辑 - 根据硬件和量化类型从 SUPPORT 矩阵获取允许的 TP 值列表，并选取合适的 TP 大小。 - 当启用 DP 时，将 GPU 总数合理分配给 TP 和 DP（例如 dp=2，tp=总 GPU 数 /2），避免 TP×DP > GPU 数的无效配置。 - 当选择 Blackwell + NVFP4 时自动添加 --mem-fraction-static 0.85，避免 CUDA graph 捕获 OOM。 - AMD 路径保持已验证的固定 TP 命令形状，不做功能性变动。
4. 添加验证数据 - 在 cookbook 中插入 GSM8K 准确度测试结果（B200 0.946，GB200 0.951）和 Throughput 性能数据，供用户参考。

docs_new/src/snippets/autoregressive/glm-47-deployment.jsx

部署命令生成器的核心组件，通过 SUPPORT 矩阵动态约束硬件、量化和 TP 组合，并实现自动 fallback，是本次变更的程序逻辑核心。

```
// 硬件支持矩阵 — 单一事实源，定义各硬件支持的量化类型及对应的 TP 容量
// hardware -> quantization -> allowed TP list
const SUPPORT = {
  b200: { nvfp4: [2, 4, 8], fp8: [4, 8], bf16: [8] },
  gb200: { nvfp4: [2, 4], fp8: [4] },
  h200: { fp8: [8], bf16: [8] },
  mi300x: { fp8: [2, 4, 8], bf16: [4, 8] },
  mi325x: { fp8: [2, 4, 8], bf16: [4, 8] },
  mi355x: { fp8: [2, 4, 8], bf16: [4, 8] },
};

// 检查给定硬件 (hw) 是否支持某量化 (q)
const quantSupported = (hw, q) => Boolean(SUPPORT[hw] && SUPPORT[hw][q]);

// 获取给定硬件 + 量化下允许的 TP 值列表
const allowedTps = (hw, q) => (SUPPORT[hw] && SUPPORT[hw][q]) || [];

// 为硬件找到第一个支持的量化类型（用于 fallback）
const firstSupportedQuant = (hw) => Object.keys(SUPPORT[hw] || {})[0] || 'fp8';

const handleRadioChange = (optionName, value) => {
  setValues(prev => {
    const next = { ...prev, [optionName]: value };
    // 当硬件或量化发生变更时，确保选择组合仍在支持矩阵内
    if (optionName === 'hardware' || optionName === 'quantization') {
      // 若当前量化不支持，自动 fallback 到该硬件下第一个支持的量化
      if (!quantSupported(next.hardware, next.quantization)) {
        next.quantization = firstSupportedQuant(next.hardware);
      }
      // 若当前 GPU 数量不在该组合的允许 TP 列表中，调整成有效值
      const tps = allowedTps(next.hardware, next.quantization);
      if (tps.length && !tps.includes(parseInt(next.gpus, 10))) {
        next.gpus = String(tps.includes(4) ? 4 : tps[0]);
      }
    }
    return next;
  });
};
```

评论区精华

在 Review 中，gemini-code-assist[bot] 提出了四项关键反馈：

- DP/TP 配置错误（高优先级）：当启用 Data Parallelism（DP）时，原实现将 --tp 和 --dp 均设为总 GPU 数，导致需要 gpus×gpus 个物理 GPU 的不合理配置。建议将物理 GPU 在 TP 和 DP 之间分割（如 dp=2，tp= 总 GPU 数 /dp）。后续提交已采纳该建议，重构了 DP/TP 分配逻辑。

- 文档推理解析器名错误（中优先级）：§1 中一处文字引用 glm47 推理解析器应改为 glm45。该处已修正。
- 冗余条件检查（中优先级）：AMD 代码块中已通过 if (isAMD) 进行分支，内部额外判断 ! isNvidiaBlackwell 冗余，建议移除。最终提交已清理该冗余检查。

所有讨论均已解决，最终版本获得了维护者（zijiexia）的批准。

风险与影响

- 风险：SUPPORT 矩阵若遗漏某些硬件 + 量化组合可能导致生成无效命令；AMD 路径的 DP/TP 分割逻辑重写可能引入回归；基准测试结果随时间可能过时；推理解析器名称依赖版本注册表。
- 影响：Blackwell 用户获得明确部署指南，AMD 用户受益于 DP/TP 修复；纯文档和前端变更，无运行时影响；团队需在新增硬件或量化时同步更新矩阵。

关联脉络

该 PR 是 GLM-4.7 文档的增量更新，与此前 AMD 专用文档方向互补，并与 SGLang 持续支持的 Blackwell 硬件路线一致。无直接关联的其他 PR。