

PR #38861 完整报告

sgl-project/sglang

Make the remaining DeepSeek-V4.1 NVIDIA cells start

合并时间: 2026-09-11 02:04

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/38861>

执行摘要

本 PR 只改一个文件 `docs/src/snippets/configs/deepseek-ai/deepseek-v4_1.jsx`, 为 DeepSeek-V4.1 cookbook 中剩余的 NVIDIA cell 补上启动必需的两类 flag:
`--cuda-graph-max-bs-decode 64` (GB300 / B200 Low-Latency 与 H200 x2) 与 `--mem-fraction-static 0.8` (仅 H200 x2), 同时把 H200 x2、B200 x2 四个 cell 从 `in-progress` 提升为 `verified`。动机是 #38839 的遗留疑问被证伪: 失败既不是 B300 特有, 也不是镜像 /PYTHONPATH 的 artifact, 而是所有 DSpark cell 都在 `decode_cuda_graph_runner` 捕获 decode CUDA graph 时 OOM, 服务在启动阶段就永不 ready。改动不触运行时代码, 风险面集中在「文档承诺的可复现性」与「64 只是能启动、不是调优值」这两点。

功能与动机

38839 在给 B300 Low-Latency 加 cap 时留了一句 caveat: B300 是在 stock 镜像上把 #38798 树挂到 PYTHONPATH 测的, 而非页面自带的 `lmsysorg/sglang:dev-dsv41`, 因此无法判断失败是 B300 特有还是该 setup 的产物。本 PR 在页面镜像上把其余平台跑完, 给出结论:

It is neither. Running the other three platforms on the page's own image settles it: every DSpark cell fails to start, and on H200 both cells do.

失败形态统一是 `Exception: Capture cuda graph failed: CUDA out of memory`。——派生的 `--cuda-graph-max-bs-decode` 在捕获 decode 图时放不下, 属于启动失败而非负载下才出现的问题。作者选择修而不是写文档, 理由有两条:

- GB300 Low-Latency cell 已被页面标为 `verified` 且是参考平台, 徽标今天不可复现;
- MI350X Low-Latency cell 早已带等价的 `--cuda-graph-max-bs 64`——ROCm 作者先撞上并加了上限, 只有 NVIDIA 侧漏了。

实现拆解

1. 规则上移到文件头注释: 原来每个 cell 各自的注释被合并成一段 header, 声明「每个 DSpark cell 都要限制 `--cuda-graph-max-bs-decode`: 派生的 batch 列表在捕获 DSpark

decode 图时放不下，所有 NVIDIA 平台都一样」，并补充 MI350X 已有等价 `--cuda-graph-max-bs`、H200 需要「上限 + 更低显存占比」、另外三个 High-Throughput cell 不带这两项也能启动。顺带修掉 #38839 把 GB300 的 `mem-fraction` 注释误写成 `verify graphs` 的问题，让读者读一处即可解释所有差异。

2. GB300 Low-Latency cell 补 `cap: match: { hw: "gb300", strategy: "low-latency" }` 的 `flags` 中，在 `--speculative-dspark-block-size 5` 之后插入 `--cuda-graph-max-bs-decode 64`。该 cell 实测「5.82 GiB wanted, 2.15 GiB free of 276.62」OOM，是页面参考平台，修复优先级最高。
3. B200 Low-Latency cell 补 `cap: match: { hw: "b200", strategy: "low-latency" }` 只加 `--cuda-graph-max-bs-decode 64`（已有 `--mem-fraction-static 0.8`），对应 4x B200 上 6.00 GiB 需求 vs 1.48 GiB 可用。
4. H200 两个 cell 补「`cap` + 显存占比」：Low-Latency 与 High-Throughput 都新增 `--cuda-graph-max-bs-decode 64` 和 `--mem-fraction-static 0.8`。原因被量得很清楚——只加 `cap` 后仍有 1.89 GiB 可用、小于 2.00 GiB 的分配请求而继续 OOM；两个 flag 同时生效后捕获时有 19.3 GiB 余量。
5. 状态与注释收敛：H200 x2、B200 x2 的 `verificationStatus: "in-progress"` 改为 `verified: true`；B200 区块删掉「`verification round open`」，B300 区块删掉「比 GB300 多一个内存 knob」的差异化描述——现在所有 DSpark cell 都要 `cap`，该差异不再成立。MI350X 按 PR 说明未测、未改。
6. 配套与校验：无运行时源码、无单测、无部署脚本变化；静态校验依赖 `node docs/scripts/check_cookbook_configs.mjs`（PR 勾选项显示 `cookbook config check: OK`）。CI 中 PR Test (Base) 通过、PR Test (Extra) 失败、AMD 侧无对应 run，无法据此判断配方可启动性。

平台 / cell	改前	本 PR 后
4x GB300 (aarch64) Low-Latency	OOM, 5.82 GiB wanted / 2.15 GiB free	加 <code>--cuda-graph-max-bs-decode 64</code> , starts
8x H200 Low-Latency	OOM, 2.00 / 0.96	加 <code>cap</code> + <code>--mem-fraction-static 0.8</code> , starts
8x H200 High-Throughput	OOM, 2.00 / 0.48	加 <code>cap</code> + <code>--mem-fraction-static 0.8</code> , starts
4x B200 Low-Latency	OOM, 6.00 / 1.48	加 <code>cap</code> , starts
4x B300 Low-Latency (#38839)	OOM, 5.82 / 1.25	已在 #38839 加 <code>cap</code>
GB300 / B200 / B300 High-Throughput	starts	刻意不加 <code>cap</code> , 不变

关键源码片段

docs/src/snippets/configs/deepseek-ai/deepseek-v4_1.jsx

唯一改动文件，承载 DeepSeek-V4.1 cookbook 的全部按硬件 / 策略分片的启动命令；本 PR 在此为 GB300、B200、H200 的 5 个 cell 补上 decode CUDA graph batch 上限与 H200 显存占比，并把 4 个 in-progress cell 提升为 verified，是「配方可复现性」这一承诺的直接载体。

```
// docs/src/snippets/configs/deepseek-ai/deepseek-v4_1.jsx
// 单文件 export const config 字面量 —— 不用展开 / 函数 / IIFE (Mintlify 在 hydration 时会重新求值)。
//
// 标记 verified 的 cell 都抄录自对应硬件上、用真实权重跑通的记录。
//
// 每个 DSpark cell 都要限制 --cuda-graph-max-bs-decode: 派生的 batch 列表
// 在捕获 DSpark decode 图时放不下，所有 NVIDIA 平台都一样。
// MI350X cell 早已带上等价的 --cuda-graph-max-bs; H200 是这里最紧的板卡 (140 GiB)，
// 两个 cell 需要该上限再加更低的显存占比；另外三个 High-Throughput cell 不带也能启动。
export const config = {
  // ... modelName / matchDims / modelNames / placeholders / options 保持不变 ...

  cells: [
    // ----- GB300: 4x GB300 (SM103)，TP4 + EP4。页面参考平台。 -----
    {
      match: { hw: "gb300", strategy: "low-latency" },
      nnodes: 1,
      verified: true, // 参考平台的徽标此前不可复现，本 PR 修复
      flags: [
        "--trust-remote-code",
        "--model-path {{MODEL_NAME}}",
        "--tp 4",
        "--ep-size 4",
        // 派生值在 128K 形状下捕获 verify 图会 OOM。
        "--mem-fraction-static 0.8",
        "--speculative-algorithm DSPARK",
        "--speculative-dspark-block-size 5",
        // 【本 PR 新增】Decode CUDA graph: 派生 batch 列表在这里放不下，
        // 实测 5.82 GiB wanted vs 2.15 GiB free (共 276.62 GiB)，服务永远不 ready。
        "--cuda-graph-max-bs-decode 64",
        "--reasoning-parser auto",
        "--tool-call-parser auto",
        "--host {{HOST_IP}}",
        "--port {{PORT}}",
      ],
    },
    // ----- H200: 8x H200，TP8 + EP8。Hopper 上没有 MXFP8 dense 路径。
    // 这里是显存最紧的板卡 (139.80 GiB)：只加 cap 仍只余 1.89 GiB，
    // 小于 2.00 GiB 的分配请求，所以两个 cell 还要把显存占比压到 0.8。 -----
    {
```

```

match: { hw: "h200", strategy: "low-latency" },
nnodes: 1,
verified: true, // 由 verificationStatus: "in-progress" 提升而来
flags: [
  "--trust-remote-code",
  "--model-path {{MODEL_NAME}}",
  "--tp 8",
  "--ep-size 8",
  // 【本 PR 新增】cap 之外再压低显存占比，两项一起才能在捕获时留下 19.3 GiB。
  "--mem-fraction-static 0.8",
  "--attention-backend dsv4",
  "--moe-runner-backend flashinfer_mxfp4",
  "--enable-decoder-swa-bounded-replay",
  // 【本 PR 新增】Decode CUDA graph: 派生 batch 列表在这里放不下。
  "--cuda-graph-max-bs-decode 64",
  "--reasoning-parser auto",
  "--tool-call-parser auto",
  "--host {{HOST_IP}}",
  "--port {{PORT}}",
],
},
// 同平台的 high-throughput cell 走完全相同的两项，只是把
// --enable-decoder-swa-bounded-replay 换成 --max-running-requests 256。
],
};

```

评论区精华

- 唯一评审来自 @wisclmy0611，状态 APPROVED，正文为空，review_comments_count = 0；作者 @zijiexia 同时是合并者，单 commit 直接进入 main。评审注意力被 PR 描述的实测表承接，而非逐行 diff。
- 技术交锋实际发生在 PR 描述里，#38839 的 caveat 被当作待验证假设处理：

It is neither. Running the other three platforms on the page's own image settles it: every DSpark cell fails to start, and on H200 both cells do.

The GB300 Low-Latency cell is already marked **verified** and is the page's reference platform, so the badge is not reproducible today. The MI350X Low-Latency cell already carries the equivalent **--cuda-graph-max-bs 64** — the ROCm author hit this and capped it. Only the NVIDIA cells were missing it.

- 取值取舍被作者自述清楚，也因此没有引出反对意见：

64 is a startup-feasibility value rather than a tuned one; it is ample for the interactive batch sizes these recipes target, and the three High-Throughput cells that start without a cap are deliberately left uncapped.

- 数据来源差异也被显式披露：B300 行取自 #38839 的 stock 镜像 + PYTHONPATH 挂 #38798 树，作者以「included for completeness」标注，避免读者混用不同测量条件。

风险与影响

- 64 不是调优值：把 decode CUDA graph 的最大 batch 从派生值（B300 为 256）压到 64，超出部分走非 graph 路径，较大并发下可能有吞吐损失；文档面向交互场景，但读者若把它当高并发配方会与预期不符。
- verified 徽标缺少回归保护：仓库唯一自动校验是 docs/scripts/check_cookbook_configs.mjs（静态 / 结构检查），无法发现启动期 OOM 回归。本 PR 一次性把 4 个 cell 提升为 verified，而 #38839 已经出现过一次徽标不可复现，说明这类漂移是真实存在的。
- 结论强绑定特定镜像与权重：GB300 / B200 / H200 均跑在 lmsysorg/sglang:dev-dsv41 上，B300 行来自另一套 setup，MI350X 未测；镜像或默认 batch 列表变化时余量（H200 捕获时 19.3 GiB）可能不再成立。
- H200 显存被两头收紧：--mem-fraction-static 0.8 会压缩 KV 池，文档只承诺「starts, serves」，长上下文或高并发下的表现没有数据。
- 无运行时风险：不涉及 decode_cuda_graph_runner 等源码，不存在服务端行为回归或兼容性问题。
- CI 信号不完整：本次 commit 的 PR Test (Extra) 失败、AMD 未跑；这些失败难以由纯文档改动触发，但也意味着没有任何自动测试为「配方能启动」这一结论背书。

对用户而言，8 个 NVIDIA cell 从 5 个起不来变为全部可启动，复制即用；对团队而言，NVIDIA 侧与 ROCm 侧（MI350X 早已带 cap）终于一致，「为什么带这些 knob」的理由集中写在文件头，降低后续维护成本。

关联脉络

- #38802 建立了 DeepSeek-V4.1 Flash cookbook 与本配置文件，定义「按硬件 × 策略分 cell、用 verified / verificationStatus 表达可信度」的结构；本 PR 是在这套结构上补齐 NVIDIA 侧启动条件。
- #38839 是直接前置：修 §3.1 推理示例并只给 B300 Low-Latency 加 cap，留下「B300 特有 vs setup artifact」的 caveat；本 PR 用同一套实测表格式收尾该 caveat，把 cap 推广到 GB300 / B200 / H200，并统一注释口径。
- #38844 是同一文件的另一条演进线：往 options 里加 HiCache L2 开关，与本 PR 往 cells[].flags 里加显存 / 图捕获 knob 分处配置的两层，共同说明这个 Playground 配置文件已成为 DeepSeek-V4.1 部署经验（含 HiCache、DSpark、显存调参）的主要沉淀点。
- 更宏观地看，从 #31470（FlashInfer MegaMOE）、#30575（AMD 稀疏 MLA）、#37306（Rust TreeCore 外部 linker）到本 PR，仓库近期在推理内核与缓存树上持续演进，而 cookbook 侧的这几次修补反映的是同一件事：硬件特性的多样化（DSpark、MXFP8、不同显存容量）让「一条命令跑通」越来越依赖按平台收敛的启动预算，文档需要与运行时默认值保持同步才能维持可信度。