PR #34587 完整报告

sgl-project/sglang

[Docs] Add Qwen3.8 cookbook

合并时间：2026-08-12 23:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34587

执行摘要

本 PR 为 Qwen 迄今最大的开源模型 Qwen3.8-2.4T-A95B 新增配置驱动的 cookbook 页面，覆盖 NVIDIA（H200/B200/B300/GB300）与 AMD（MI300X/MI350X/MI355X）7 平台共 18 个启动配方，并将投机解码的 NEXTN 与 DSpark 策略纳入其中。除了新增页面（+1305 行），还做了两处刻意且必要的配套改动：扩展 Playground 引擎使 PD 角色可携带专属 flags/env（_playground.jsx +18），以及修复 cookbook 脚手架模板的 router 端口硬编码缺陷。全 PR 不触碰模型或推理代码，属于纯文档与文档工具链变更，但因其部署矩阵的规模和架构特殊性（GDN 线性注意力主导），对用户部署决策有直接且重要的指导价值。

功能与动机

PR body 的 Motivation 指出：Qwen3.8-2.4T-A95B 没有 cookbook 页面，而它是 Qwen 最大的开源权重模型（2.4T 总参 / 95B 激活，92 层由 23 组『3 × (Gated DeltaNet → MoE) → 1 × (Gated Attention → MoE)』构成），部署方式与已有页面覆盖的稠密注意力模型完全不同：

- 三分之二的层是线性注意力，限制并发的不是 KV cache，而是 GDN 循环状态池；
- 2.4T 参数下 FP8 无法装入页面中任何平台的单节点，拓扑受权重体积决定；
- 推理不可关闭，必须显式携带 reasoning parser。

因此该页面不是对既有模板的简单复制，而是针对混合线性注意力架构重新设计的部署指南。

实现拆解

1. 新增模型配置数据（核心） docs/src/snippets/configs/Qwen/qwen3.8.jsx（+939）是整页数据源：定义 4 种量化（BF16/FP8/NVFP4/MXFP4）× 4 策略（Low Latency/Balanced/High Throughput/DSpark）× 4 档节点数的部署矩阵、各平台 Docker 镜像（含 ROCm 7.00/7.20 区分）、多节点 IB/NCCL 绑定提示，以及 Playground 各卡片能力——Attention 并行（TP 到 64、DP-Attention 到 64）、MoE backend（DeepEP/MegaMoE/FlashInfer MXFP4/Marlin）与 WideEP（EP 到 64）、解析器（qwen3 / qwen3_coder）、投机解码（EAGLE/NGRAM/DSpark）、PD 分发（Mooncake/NiXL）与层级 KV cache。关键设计是 DSpark 选项的 disable 条件与 _handle_dspark（python/sglang/srt/arg_groups/speculative_hook.py）的硬约束一一对应。
2. 扩展 Playground PD Disagg 引擎 docs/src/snippets/_playground.jsx（+18）允许 modes[] 条目声明角色专属 flags / env（如 prefill 的 --load-balance-method、decode 的

轮询间隔）。新增符号 modeMeta 在角色分支内查找当前角色，先按 flag 头剥离 base cell 同名 flag 再追加，env 按值去重。由于 applyAllDeltas 每次 render 从 baseFlags 重新 seed，该改动对现有 8 个配置是 no-op。

3. 新增cookbook页面并注册docs/cookbook/autoregressive/Qwen/Qwen3.8.mdx（+315）承载架构介绍、配置要点（GDN 状态槽比例 1/3/5、PP 关闭 overlap scheduler 时为 4；NEXTN 48 并发默认；SM90 与 SM100 线性注意力后端差异）并嵌入 Deployment/Playground 组件。docs/docs.json 注册页面，intro.mdx 卡片链接转向 Qwen3.8，Qwen3.6.mdx 移除 tag: NEW。
4. 修复脚手架模板端口硬编码 .claude/skills/cookbook-add-model/templates/config.jsx.tmpl 原先硬编码 router 的 prefill/decode 端口 30000/30001，而引擎 decode 实际监听 30100；改为 {{PREFILL_PORT}} / {{DECODE_PORT}} / {{ROUTER_PORT}} 占位符，并在 authoring-reference.md 中明确『字面端口不会跟随引擎』的告诫。
5. 测试与校验配套 无模型 / 内核代码改动，无对应测试文件；作者声明 mint validate 与 mint broken-links 均通过；benchmark 以空桩形式发布（页面显示 pending），避免未经验证的性能数字上页面，同时 benchmarkCommands 提供复现命令供用户自行测量。

docs/src/snippets/configs/Qwen/qwen3.8.jsx

整页部署配置的核心数据源（+939 行），定义 7 平台 × 4 量化 × 4 策略的 18 个 launch recipe，以及 Playground 各卡片能力、DSpark 约束建模、PD 角色 flag 声明。

```
// ----- Card: "Speculative Decoding" -----
// DSpark 是训练好的草稿模型，替换 NEXTN 作为独立 strategy 芯片，
// 而不是吞吐 / 延迟曲线上的另一个档位（不同解码器，不是不同操作点）。
speculative: {
  options: [
    { id: "current", label: "Inherited from base" },
    { id: "off", label: "Off (greedy)" },
    { id: "mtp", label: "EAGLE / MTP",
      flags: ["--speculative-algorithm EAGLE", "--speculative-num-steps 3",
              "--speculative-eagle-topk 1", "--speculative-num-draft-tokens 4"] },
    // _handle_dspark（python/sglang/srt/arg_groups/speculative_hook.py）
    // 在约束不满足时直接 raise 而不是降级：pp_size 必须为 1，
    // DP-Attention 下还要求 --enable-dp-lm-head、--moe-a2a-backend none
    // 且无 context parallel。因此下面的 disable 条件与 hook 的硬约束
    // 一一对应，保证 Playground 不会生成注定启动失败的命令。
    { id: "dspark", label: "DSpark",
      flags: ["--speculative-algorithm DSPARK",
              "--speculative-draft-model-path RadixArk/Qwen3.8-Max-DSpark"],
      disable: [
        { when: { dpAttnOn: [true] },
          reason: "DSpark with DP-Attention additionally requires " +
                  "--enable-dp-lm-head, the built-in TP MoE " +
                  "(--moe-a2a-backend none) and no context parallel." },
        { when: { hw: ["h200", "mi300x"] },
          reason: "DSpark requires pp_size == 1 and this recipe is " +
                  "pipelined (TP x PP across nodes)." },
```

```
        { when: { hw: ["b200", "b300"], quant: ["fp8"] },
          reason: "DSpark requires pp_size == 1 and the B200/B300 FP8 " +
                  "recipes are TP8 x PP2." },
      ] },
    { id: "ngram", label: "NGRAM",
      flags: ["--speculative-algorithm NGRAM",
              "--speculative-num-draft-tokens 16",
              "--speculative-ngram-max-bfs-breadth 10"],
      disable: { dpAttnOn: [true] } },
  ],
},
```

docs/src/snippets/_playground.jsx

共享渲染引擎的唯一源码改动（+18），让 modes[] 条目可携带角色专属 flags/env，是本次唯一影响所有既有 cookbook 页的变更点。

```
// 确定当前 PD 角色：config 声明了 modes[] 时以 Playground 选中值为准，
// 否则回落到 Deploy 面板上的 pdMode（该角色在 Deploy 面板选择）。
const mode = (fc.modes || []).length ? value.mode : ((sel && sel.pdMode) || "off");

if (mode === "prefill" || mode === "decode") {
  const backend = value.transferBackend || (backends[0] || {}).id || "mooncake";
  const adds = [
    `--disaggregation-mode ${mode}`,
    `--disaggregation-transfer-backend ${backend}`,
  ];
  // 回填 bootstrap 与 IB device flag（顺序与 base cell 保持一致）。
  if (bootstrapPort) adds.push(`--disaggregation-bootstrap-port ${bootstrapPort}`);
  if (value.ibDevice && value.ibDevice !== "auto") {
    adds.push(`--disaggregation-ib-device ${value.ibDevice}`);
  }

  // 角色专属 flags：modes[] 条目可声明只属于 prefill / decode 的 flags
  // （如 prefill 的 --load-balance-method、decode 的轮询间隔）。先按
  // flag 头剥离 base cell 中同名 flag，让角色专属值胜出而不是重复输出。
  // 该剥离只发生在角色分支内：applyAllDeltas 每次 render 从 baseFlags
  // 重新 seed，角色关闭时不存在残留 base flag 需要清理。
  const modeMeta = (fc.modes || []).find((m) => m.id === mode);
  if (modeMeta && modeMeta.flags && modeMeta.flags.length) {
    flags = h.stripFlagsByFirstToken(
      flags, modeMeta.flags.map((f) => f.split(/[\s=]/)[0]));
    adds.push(...modeMeta.flags);
  }

  // 单机无需 --dist-init-addr：prefill / decode 按 PD_PORTS 从各自
  // --port 推导 ZMQ 端口（间隔 100），范围不重叠；多机仍由渲染器
  // 注入跨节点 --dist-init-addr。
  flags = h.insertBeforeTail(flags, adds);
```

```
    // 角色专属 serving 端口，使 router 的 prefill / decode 目标对齐。
    const servePort = PD_PORTS[mode].serve;
    flags = flags.map((f) =>
      f.split(/[\s=]/)[0] === "--port" ? `--port ${servePort}` : f);

    // 后端 env 按 envWhen 硬件门控追加，保留 base cell 原有 env。
    const meta = backends.find((b) => b.id === backend);
    if (meta && meta.env && meta.env.length) {
      const gate = meta.envWhen;
      const ok = !gate || Object.keys(gate).every(
        (k) => (gate[k] || []).includes(sel[k]));
      if (ok) env = [...env, ...meta.env.filter((e) => !env.includes(e))];
    }
    // 角色专属 env 同样按值去重；apply 是纯函数、每次从 baseEnv 重建，
    // 不会跨 render 残留上一个角色的 env。
    if (modeMeta && modeMeta.env && modeMeta.env.length) {
      env = [...env, ...modeMeta.env.filter((e) => !env.includes(e))];
    }
  }
  return { flags, env };
```

评论区精华

该 PR 无独立 review 评论，两位 reviewer（JustinTong0323、wisclmy0611）均直接 APPROVED。最有价值的讨论来自作者在 PR body『Notes for reviewers』中主动澄清的设计决策：

> 『模板硬编码 30000 / 30001 而引擎 decode 服务在 30100，字面端口会让 router 打向 decode 进程从未监听的端口。』——这是本次唯一被顺带修复的真实缺陷，而非风格问题。
>
> 『dspark 是模型专属策略 ID，因为 DSpark 是另一套投机解码器，不是吞吐 / 延迟曲线上的不同操作点。』—— 解释了为什么它在策略矩阵中作为第四块芯片而非第四档。
>
> 『GB300 Balanced 和 High Throughput 档的 --moe-a2a-backend deepep_v2 上游尚不存在。』—— 明确标注了与上游发布的时序耦合。
>
> 『--kv-cache-dtype fp8_e4m3 是逐格携带而非页级默认；B300 NVFP4 和 GB300 BF16 刻意以模型精度服务 KV。』—— 避免一刀切默认值误导不同硬件组合。

风险与影响

风险：

1. 共享渲染引擎回归风险：_playground.jsx 被所有 cookbook 页面共用，本次 +18 改动无对应测试；作者声明对现有 8 个配置是 no-op 且逐一核验，但后续 config 结构变更若触发 modeMeta 分支，可能静默改变所有页面的命令输出。
2. GB300 配方依赖未发布 flag：--moe-a2a-backend deepep_v2 上游尚不存在，旧版本 SGLang 上直接运行 Balanced/High Throughput 命令会失败。
3. 基准数据全部 pending：18 个 benchmark cell 均为空桩，用户无法据此对比硬件 / 量化选型。

4. Docker 镜像 tag 依赖发布时序：lmsysorg/sglang:qwen38 与 ROCm 镜像 tag 为发布期 pin，镜像未同步推送时 docker pull 会失败。
5. 脚手架模板修复的传播范围：端口占位符改动影响后续所有新建 cookbook，但已按旧模板生成的页面需逐个确认是否仍携带 30001 字样。

影响：

- 用户：获得 Qwen3.8 的 day-0 部署配方，尤其是 GDN 状态池作为并发瓶颈的配置指引（状态槽比例、NEXTN 并发默认、SM90/SM100 差异）是此前页面从未覆盖的架构维度。
- 团队：脚手架 skill 的端口缺陷修复使后续所有模型页面生成受益；Playground 引擎能力扩展为 PD 角色专属 flag 提供了标准表达位置。
- 系统：纯文档变更，无推理路径、无运行时影响；_playground.jsx 的幂等 delta 设计保证对既有页面零行为变化。

关联脉络

- PR#34590（Qwen3.8-Max-DSpark 重命名为 Qwen3.8-2.4T-A95B-DSpark）与本 PR 同属 Qwen3.8 文档线：本页 DSpark 配方仍引用 RadixArk/Qwen3.8-Max-DSpark，后续需随重命名同步更新 repo id。
- PR#34379 / #34497展示了同一条 config 驱动 cookbook 工作流（MDX 页面 + snippets/configs 配置 + docs.json 注册）在 GLM、Cosmos 等模型上的持续复用，本 PR 还回馈了工作流本身——修复模板端口并扩展 Playground 契约。
- 结合近期多个 diffusion cookbook 与 AMD/NPU 支持 PR，可见 SGLang 正在把『配置驱动的部署指南』沉淀为标准化交付物，并让脚手架与渲染引擎随模型需求持续演化。