PR #34143 完整报告

sgl-project/sglang

docs(diffusion): refresh skills for latest runtime

合并时间：2026-08-09 10:30

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34143

执行摘要

- 一句话：刷新 diffusion 技能文档与基准脚本，对齐最新运行时
- 推荐动作：该 PR 值得快速浏览，重点看 bench_diffusion_denoise.py 两处逻辑变更和 existing-fast-paths.md 的 fast-path 清单——它们直接反映了近期 diffusion 内核优化（FLUX/GLM/SANA LayerNorm+modulate、Wan causal VAE 等）的落地状态。若你负责 diffusion 性能分析或维护 agent 技能，建议以此 PR 为基线核对本地技能版本；若只是普通用户，了解 SGLANG_DIFFUSION_SYNC_STAGE_PROFILING 的语义即可，不需要精读。

功能与动机

PR body 明确说明：技能文档早于近期合并的 diffusion 模型、部署控制、量化 checkpoint 和内核快路径。具体问题是 --validate-nightly-alignment 因 comparison_configs.json 仍包含对比驱动的 --warmup 标志而失败（原生 CLI 已迁移到 --warmup-mode）。保持技能与实现一致，避免陈旧命令、重复优化、误导性的 ModelOpt 后端建议和错误的阶段级 benchmark 归因。

实现拆解

本 PR 以文档刷新为主，附带一个基准脚本的小幅逻辑调整，整体分四步：

1. 刷新 benchmark/profile 技能文档（sglang-diffusion-benchmark-profile/SKILL.md、benchmark-and-profile.md、existing-fast-paths.md）：补充 SGLANG_DIFFUSION_SYNC_STAGE_PROFILING 环境变量的使用说明，明确 stage 级计时必须同步，否则异步 GPU 工作可能泄漏到后续阶段导致 DecodingStage 虚高 2-3 倍；在 existing-fast-paths.md 中扩展了最新融合内核清单（modulate_scale_shift、fused_ln_modulate、native_bf16_rmsnorm、wan_causal_cache、VAE quality gate 等）及对应验证测试文件。
2. 调整基准脚本逻辑（scripts/bench_diffusion_denoise.py）：在 _expected_nightly_cli_args 中把 --warmup 与 --warmup-mode 一并排除在 preset 漂移校验之外，因为对比驱动仍使用旧标志；在 run_benchmark_once 中通过 env.setdefault("SGLANG_DIFFUSION_SYNC_STAGE_PROFILING", "1") 默认开启阶段同步，除非调用方显式设 0 退出。

3. 更新 ModelOpt 量化技能（sglang-diffusion-modelopt-quant/SKILL.md）：补充 FP4 GEMM 后端默认值（FlashInfer TensorRT-LLM）、NVFP4 支持家族扩展（Qwen Image 系列）、已发布 ModelOpt checkpoint 仓库数量与命名规则，并增加完整 Diffusers repo 的 --model-path 用法示例。
4. 更新性能与加模型技能（sglang-diffusion-performance/SKILL.md、sglang-diffusion-add-model/SKILL.md）：性能表新增 Performance Mode、Breakable CUDA Graph、Cross-node SP、质量门控、TeaCache/Spectrum/Progressive Resolution、Causal KV-Cache 量化等条目，并明确各选项的相互排斥关系与限制；add-model 技能补充 Krea-2、LingBot Video MoE 30B 的流水线实现位置和 native task-contract 差异说明。

测试方面没有新增测试文件，但脚本的 --validate-nightly-alignment 已通过全部 11 个映射 nightly preset 的验证；文档提及的新路径均通过源文件存在性检查。

关键文件：

- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/scripts/bench_diffusion_denoise.py（模块 基准脚本；类别 source；类型 core-logic；符号 _expected_nightly_cli_args, run_benchmark_once）：唯一源码变更文件，修复 nightly preset 对齐校验误报并默认启用 stage 同步，直接影响 benchmark 结果可信度。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/existing-fast-paths.md（模块 技能文档；类别 docs；类型 documentation）：大幅更新 diffusion 已实现内核与 fast-path 清单，是技能中最重要的参考资料，直接影响后续优化工作的起点判断。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-modelopt-quant/SKILL.md（模块 技能文档；类别 docs；类型 documentation）：更新 ModelOpt 量化技能，修正 FP4 后端默认值描述并扩充 NVFP4 支持家族，避免误导性的量化建议。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-performance/SKILL.md（模块 技能文档；类别 docs；类型 documentation）：性能调优技能新增多个运行时特性的使用说明和约束，是用户选择优化手段的关键入口。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/benchmark-and-profile.md（模块 技能文档；类别 docs；类型 documentation）：benchmark 工作流文档补充 stage 同步计时的使用要求和结果记录规范，与脚本默认值变更配套。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-add-model/SKILL.md（模块 技能文档；类别 docs；类型 documentation）：补充新增模型家族（Krea-2、LingBot Video MoE）的流水线位置和 native task-contract 说明，帮助加模型时定位代码。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/SKILL.md（模块 技能文档；类别 docs；类型 documentation）：benchmark/profile 技能主文档同步更新 stage sync 说明和 fast-path 检查清单，保持与脚本行为一致。

关键符号：_expected_nightly_cli_args, run_benchmark_once, validate_nightly_alignment

关键源码片段

python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/scripts/bench_diffusion_denoise.py

唯一源码变更文件，修复 nightly preset 对齐校验误报并默认启用 stage 同步，直接影响 benchmark 结果可信度。

```
def _expected_nightly_cli_args(case: dict) -> dict[str, str]:
    expected = {
        "width": str(case["width"]),
        "height": str(case["height"]),
    }

    for key, flag in (
        ("num_frames", "num-frames"),
        ("fps", "fps"),
        ("num_inference_steps", "num-inference-steps"),
        ("guidance_scale", "guidance-scale"),
    ):
        if key in case:
            expected[flag] = str(case[key])

    if case.get("num_gpus", 1) > 1:
        expected["num-gpus"] = str(case["num_gpus"])

    serve_args = shlex.split(case["frameworks"]["sglang"].get("serve_args", ""))
    parsed_serve_args = _parse_cli_args(serve_args)
    for flag, value in parsed_serve_args.items():
        # Nightly 对比驱动仍持有旧版 --warmup 开关；warmup-mode 迁移后它不再是
        # sglang generate 的合法参数，因此校验 preset 漂移时两种写法都要跳过。
        if flag in {"enable-torch-compile", "warmup", "warmup-mode"}:
            continue
        expected[flag] = _normalize_cli_value(value)

    return expected
```

```
def run_benchmark_once(
    model_key: str,
    label: str,
    output_dir: Path,
    warmup: bool = True,
    torch_compile: bool = True,
) -> dict:
    """Run a single benchmark pass and return results dict."""
    perf_path = output_dir / f"{model_key}_{label}.json"

    cmd = build_sglang_cmd(
        model_key,
        perf_dump_path=str(perf_path),
        warmup=warmup,
        torch_compile=torch_compile,
```

```
        artifact_dir=output_dir,
    )

    env = os.environ.copy()
    env.setdefault("FLASHINFER_DISABLE_VERSION_CHECK", "1")
    # perf dump 会被当作分阶段 denoise 测量来消费。在阶段边界排空设备队列，
    # 避免异步 denoise 工作泄漏到后续阶段（最明显的是 DecodingStage）。
    # 调用方环境里显式设 0 仍可退出该行为，用于纯端到端实验。
    env.setdefault("SGLANG_DIFFUSION_SYNC_STAGE_PROFILING", "1")
    cfg = MODELS[model_key]
    for key, value in cfg.get("env", {}).items():
        env.setdefault(key, str(value))
    if env.get("HF_TOKEN") and not env.get("HUGGINGFACE_HUB_TOKEN"):
        env["HUGGINGFACE_HUB_TOKEN"] = env["HF_TOKEN"]
```

评论区精华

本 PR 没有收到实质性的 review 评论或讨论线程（review_comments_count 为 0，comments 仅有作者触发的 /tag-and-rerun-ci 命令）。PR body 中说明了验证过程，包括 bench_diffusion_denoise.py --validate-nightly-alignment 全部通过、ruff check 通过等；也提到本地 macOS 环境因 Transformers 版本过旧且 SciPy/NumPy ABI 不匹配，无法收集 diffusion CPU 测试。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险集中在基准脚本的默认行为变更：
  - bench_diffusion_denoise.py 默认设置 SGLANG_DIFFUSION_SYNC_STAGE_PROFILING=1，会改变 perf dump 的 stage 计时口径。若用户用旧脚本生成的未同步结果与新结果对比，可能得出错误的性能结论；文档已要求成对对比时保持该设置一致，但存量脚本用户可能忽略。
  - _expected_nightly_cli_args 将 --warmup 排除出漂移校验，虽然修复了误报，但也意味着若对比驱动未来真的需要支持 --warmup-mode 而脚本未同步更新，校验可能继续通过而掩盖问题。
  - 文档中大量新增的路径、参数、内核清单若与 main 分支后续演进脱节，会再次产生陈旧文档风险；PR 只做了静态存在性检查，未做运行级验证。
  - 影响：对用户的影响：使用 Claude skills 的开发者（尤其是通过 agent 驱动的 diffusion 工作流）会获得与当前 main 一致的命令、参数和瓶颈定位清单，避免基于过时信息做重复优化或误判。对系统影响：脚本默认启用 stage 同步后，benchmark 输出的 DecodingStage 等阶段指标更可信，但历史数据的可对比性下降。对团队影响：这是一次低风险的文档维护，但包含一个可观测性相关的默认值变更，需要同步到 CI 或文档入口，避免工具链前后不一致。
  - 风险标记：默认启用 stage 同步影响 benchmark 可比性，文档与实现需持续对齐

关联脉络

- PR #34124 [diffusion] perf_logger: SYNC_STAGE_PROFILING must drain the GPU queue for stage records too (fixes 2-3x inflated DecodingStage readings): 本 PR 中脚本默认启用 SGLANG_DIFFUSION_SYNC_STAGE_PROFILING=1 正是为了吸收该修复带来的行为变化，两者构成前后端配套。
- PR #34085 [diffusion] Clean up kernels and shared fast paths: existing-fast-paths.md 中新增的共享快路径清单与该 PR 的内核清理和共享路径整理直接相关。
- PR #34126 [diffusion] FLUX.1: route the adaLN LN+modulate sites through the bit-exact fused LayerNorm+modulate kernel: fast-path 文档新增的 fused_layernorm_modulate 条目对应此 PR 引入的 FLUX 融合内核。
- PR #34125 [diffusion] Bit-exact data-movement elimination for the Wan causal VAE decoder: existing-fast-paths.md 新增的 wan_causal_cache 数据搬运内核条目对应此 PR 的 Wan VAE 优化。
- PR #34015 [diffusion] Sana: bit-exact fused aten LayerNorm+modulate under BCG: fast-path 文档新增的 Sana LayerNorm+modulate 条目对应此 PR 的 Sana 融合内核。