

PR #2334 完整报告

THUDM/slime

[release] bump to v0.3.2

合并时间: 2026-08-28 10:17

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2334>

执行摘要

- 一句话: 发布 v0.3.2, 新增 SGLang 确定性补丁与发布校验脚本
- 推荐动作: 建议负责构建与发布的同学精读 `check_release.py` 与 `build_conda.sh`: 前者用 AST/ 正则做交叉校验的思路简洁可靠, 可直接作为发布门禁; 后者集中体现了 conda/Docker 双路径对齐的维护成本。对于关心 FP8 推理的工程师, `sglang-deterministic.patch` 中 UE8M0 scale 与 SiLU 舍入的处理与近期 PR#2327 的 FP8 开关联动, 值得深入研究。该 PR 无需逐行审查补丁, 但应确认补丁与上游 commit 固定一致。

功能与动机

PR body 为空, 动机主要从提交与变更内容推断: v0.3.2 需要同步发布包的版本号, 并修复构建环境中的 cu13/cu129 依赖残留问题 (`build_conda.sh` 注释说明了这一点)。新增的 `sglang-deterministic.patch` 旨在让 SGLang 的 FP8 量化与 MoE 调度在 Blackwell 上具备确定性, 避免 EAGLE 采样与训练侧数值不一致; `megatron-sglang-aligned.patch` 则通过 FP32 残差和路径对齐 Megatron 与 SGLang 的归一化边界。新增 `check_release.py` 是为了让此类发布操作可被脚本校验, 降低漏改版本号或补丁失配的风险。

实现拆解

1. 同步发布版本号: `setup.py` 中 `setup(version="0.3.2")`、`docs/conf.py` 中 `__version__ = "0.3.2"`、`docker/version.txt` 同步到 v0.3.2, 保证 wheel、文档与镜像三处版本标识一致。
2. 新增 SGLang 确定性补丁: 添加约 1620 行的 `docker/patch/v0.5.15.post1/sglang-deterministic.patch`, 修改 SGLang 的 `per_token_group_quant_8bit_v2.cuh` 等内核, 将 SiLU 计算与 UE8M0 量化 scale 调整为与 SM90 `sgl_kernel` 一致的单次低精度舍入边界; 同时包含 DeepGEMM `num_sms` 配置、MoE dispatcher 创建逻辑调整, 以保证 Blackwell FP8 推理的数值可复现。
3. 新增 Megatron-SGLang 对齐补丁: 添加约 119 行的 `docker/patch/v0.5.15.post1/megatron-sglang-aligned.patch`, 修改 `megatron/core/transformer/transformer_layer.py`, 通过环境变量 `MEGATRON_USE_SGLANG_FUSED_RESIDUAL_RMS` 开关启用 FP32 残差和 + 原生 RMSNorm 路径, 使训练侧与 SGLang 的残差归一化边界对齐。
4. 更新 conda 构建脚本: `build_conda.sh` 将 `SGLANG_VERSION` 提升到 v0.5.15.post1, 新增 `FLASH_QLA_COMMIT` 环境变量并固定其安装提交; 在安装 SGLang 后强制恢复

cuda-python==12.9 以修复 cu13 运行时混入；在 patch 循环中追加
sglang-deterministic.patch。

5. 新增发布校验能力：新增 .claude/skills/release/SKILL.md 与 scripts/check_release.py，脚本用 AST 与正则解析 setup.py、docs/conf.py、docker/Dockerfile、docker/justfile、build_conda.sh、docker/README.md，交叉核对 slime 版本、SGLang 版本、MEGATRON_COMMIT/TMS_COMMIT/FLASH_QLA_COMMIT 以及补丁目录内容是否一致，并在 main() 中汇总错误。

关键文件：

- .claude/skills/release/scripts/check_release.py（模块 发布校验；类别 source；类型 dependency-wiring；符号 _setup_version, _assigned_string, _shell_exports, _docker_args）：新增的发布一致性校验脚本，用 AST/ 正则解析版本号与补丁目录，是本次发布流程自动化的核心。
- docker/patch/v0.5.15.post1/sglang-deterministic.patch（模块 运行时补丁；类别 infra；类型 infrastructure；符号 _log_softmax_kernel, dump_current_tensors, _dump_hook, inner_dump_hook）：本次发布的核心功能补丁，约 1620 行，修改 SGLang 量化与 MoE 调度内核，保证 FP8 推理确定性。
- docker/patch/v0.5.15.post1/megatron-sglang-aligned.patch（模块 对齐补丁；类别 infra；类型 infrastructure；符号 get_transformer_layer_offset, sharded_state_dict）：新增 Megatron 侧对齐补丁，通过环境变量开关启用 FP32 残差和 RMSNorm 路径，使 Megatron 与 SGLang 数值边界一致。
- build_conda.sh（模块 构建脚本；类别 other；类型 core-logic）：更新 conda 构建流程：提升 SGLang 版本、新增 FLASH_QLA_COMMIT 固定、修复 cuda-python 版本回退、加入新补丁。
- setup.py（模块 打包配置；类别 source；类型 core-logic）：package 版本号 0.3.1 → 0.3.2，是 wheel 发布的基础配置。
- docs/conf.py（模块 文档配置；类别 source；类型 core-logic）：文档版本号与 setup.py 保持一致，避免 Sphinx 文档版本错乱。
- docker/version.txt（模块 镜像版本；类别 docs；类型 documentation）：镜像版本标记同步到 v0.3.2。
- .claude/skills/release/SKILL.md（模块 发布技能；类别 docs；类型 documentation）：描述 Claude release 技能流程，配合 check_release.py 形成完整的发布操作手册。

关键符号：main, _setup_version, _assigned_string, _shell_exports, _docker_args, _loop_items, _use_sglang_fused_residual_rmsnorm, _sglang_native_rmsnorm_from_fp32_sum

关键源码片段

[.claude/skills/release/scripts/check_release.py](#)

新增的发布一致性校验脚本，用 AST/ 正则解析版本号与补丁目录，是本次发布流程自动化的核心。

```
def main() -> int:
```

```

parser = argparse.ArgumentParser()
parser.add_argument("--repo", type=Path, default=Path.cwd())
parser.add_argument("--expected-version")
args = parser.parse_args()

repo = args.repo.resolve()
errors: list[str] = []

# 校验 slime 自身版本: setup.py 与 docs/conf.py 必须一致,
# 并可通过 --expected-version 约束本次发布的期望版本。
setup_version = _setup_version(repo / "setup.py")
docs_version = _assigned_string(repo / "docs/conf.py", "__version__")
if setup_version != docs_version:
    errors.append(f"setup.py={setup_version} but docs/conf.py={docs_version}")
if args.expected_version and setup_version != args.expected_version:
    errors.append(f"release version is {setup_version}, expected {args.expected_version}")

# Docker 镜像与 conda 构建的 SGLang 版本与依赖 commit 必须对齐,
# 避免两条安装路径运行时行为不一致。
docker_text = (repo / "docker/Dockerfile").read_text()
conda_text = (repo / "build_conda.sh").read_text()
docker_args = _docker_args(docker_text)
conda_exports = _shell_exports(conda_text)

# 镜像 tag 形如 v0.5.15.post1-cu129, 去掉 -cu 后缀后与 conda 版本比对
image_tag = docker_args.get("SGLANG_IMAGE_TAG", "")
docker_sglang_version = re.sub(r"-cu\d+$", "", image_tag)
conda_sglang_version = conda_exports.get("SGLANG_VERSION", "")
if docker_sglang_version != conda_sglang_version:
    errors.append(
        f"Docker SGLang={docker_sglang_version or '<missing>'}, "
        f"conda SGLang={conda_sglang_version or '<missing>'}"
    )

# Megatron、TMS、FlashQLA 等固定提交也必须两侧一致
for pin in ("MEGATRON_COMMIT", "TMS_COMMIT", "FLASH_QLA_COMMIT"):
    if docker_args.get(pin) != conda_exports.get(pin):
        errors.append(
            f"{pin}: Docker={docker_args.get(pin, '<missing>')}, "
            f"conda={conda_exports.get(pin, '<missing>')}"
        )

# 后续代码继续校验 docker/README.md 的 stable 版本、docker/justfile
# 的镜像 tag, 以及 docker/patch/<version> 与 docker/patch/latest 的补丁一致性。

```

build_conda.sh

更新 conda 构建流程: 提升 SGLang 版本、新增 FLASH_QLA_COMMIT 固定、修复 cuda-python 版本回退、加入新补丁。

```
# 安装 SGLang 后，SGLang 会顺带升级 cuda-python 到 >=13, 与 CUDA 12.9
# 运行时冲突；这里强制恢复到 12.9, 修复 site-packages/nvidia 目录混装问题。
pip install --force-reinstall "cuda-python==12.9"

# $patch_dir 在此之前已指向 docker/patch/$PATCH_VERSION;
# 逐项应用 slime 维护的 SGLang 补丁。新增的 sglang-deterministic.patch
# 固定 FP8 量化的 scale 计算与 SiLU 舍入行为，保证 Blackwell 上的 EAGLE
# 采样与训练侧数值对齐，从而让 rollout 结果可复现。
for patch_name in sglang.patch sglang-top_p.patch sglang-release_hicache.patch sglang-pull_
weights.patch sglang-deterministic.patch; do
    patch_path="$patch_dir/${patch_name}"
    if [ ! -f "$patch_path" ]; then
        echo "patch not found: $patch_path"
        exit 1
    fi
    git apply "$patch_path"
done
```

评论区精华

该 PR 没有 review 评论，合并过程无公开讨论。从补丁内容看，设计重点集中在数值对齐策略：例如 sglang-deterministic.patch 中 UE8M0 量化采用四舍五入的 power-of-2 scale 以避免与下游 GEMM dequant 的 scale 不匹配（补丁注释称可避免约 14 倍误差放大）；megatron-sglang-aligned.patch 通过环境变量开关引入 FP32 残差和，避免改动默认训练路径。这些决策未在 PR 内展开讨论，但实现上做了条件开关与明确注释。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 外部源码补丁风险：docker/patch/v0.5.15.post1/ 下的补丁直接修改上游 SGLang/Megatron 源码，若后续升级 SGLang 版本，这些补丁可能无法干净应用，阻碍依赖升级。
2. 构建环境修复依赖强制重装：build_conda.sh 中 pip install --force-reinstall "cuda-python==12.9" 和 torch 系 cu129 wheel 的强制重装，依赖 PyPI 与 PyTorch 索引可用性，若 wheel 变动可能导致构建失败；同时 site-packages/nvidia 混装修复属于精细操作，仍有回归可能。
3. 版本号多处同步：slime 版本同时出现在 setup.py、docs/conf.py、docker/version.txt，以及 Docker 镜像 tag 中，任何遗漏都会造成发布产物不一致；虽有 check_release.py 校验，但该脚本本身没有对应自动化测试（本 PR 未新增测试文件）。
4. 补丁影响面：sglang-deterministic.patch 改动 FP8 量化与 MoE 调度路径，若运行环境非 SM90/Blackwell，或 --force-fp8-ue8m0-scale 等开关未按预期组合，可能引入数值不一致或性能回退。- 影响：对用户：安装 v0.3.2 的 Docker/conda 环境会打入 SGLang 与 Megatron 补丁，FP8 量化和 MoE 推理行为发生变化，默认数值确定性提升，但可能略微影响吞吐；同时依赖被固定在特定 commit，升级路径受限。对系统：构建

流程更复杂，conda 环境需要额外修复步骤（cuda-python 回退，补丁应用），但发布校验脚本降低了版本错配风险。对团队：发布操作开始向 Claude 技能化演进，后续发版可通过 `check_release.py` 进行预检，减少人工检查成本。 - 风险标记：外部源码补丁，构建脚本变更，版本号多处同步，新增脚本无测试覆盖

关联脉络

- PR #2327 feat: allow forcing UE8M0 FP8 scales: `sglang-deterministic.patch` 中大量修改 UE8M0 量化 scale 计算，与 PR#2327 新增的 `--force-fp8-ue8m0-scale` 开关属于同一 FP8 量化功能线，本 PR 的补丁可视为 SGLang 侧的配套对齐实现。
- PR #2330 [docs] fix out-dated doc: PR#2330 引入了文档一致性测试，覆盖 `docs/conf.py` 的版本号来源；本 PR 同步更新 `docs/conf.py` 的 version，两者共同维护文档与发布版本的同步。