PR #2228 完整报告

THUDM/slime

[docker] upgrade sglang to v0.5.15.post1

合并时间：2026-07-23 16:26

原文链接：http://prhub.com.cn/THUDM/slime/pull/2228

执行摘要

- 一句话：SGLang 升级至 v0.5.15.post1，同步更新 Docker 构建与 patch
- 推荐动作：本 PR 适合所有使用 slime Docker 镜像的团队精读，尤其是 arguments.py 中的参数兼容模式和 sglang_engine.py 的默认值覆盖策略值得在设计迁移时借鉴。建议 merge 前运行完整 CI（包括 offload、disagg 测试组合），并确认无回归。

功能与动机

升级 SGLang 以获取上游最新功能、性能优化和 bugfix。从新增测试 test_memory_saver_disables_default_breakable_prefill_cuda_graph 可以看出，新版本引入了 cuda_graph_backend_prefill 字段，在启用 memory saver 时自动禁用 breakable prefill CUDA graph 以节省显存，这是升级的主要收益之一。此外，大幅简化的 patch 表明 slime 团队在持续跟进上游，减少维护成本。

实现拆解

1. Dockerfile 依赖更新：调整 docker/Dockerfile 和 docker/version.txt，将 SGLang 版本锁定为 v0.5.15.post1，并同步更新关键 Python 依赖版本（如 torch、transformers 等），确保构建一致性。
2. SGLang 补丁重生成与裁剪：重新生成四个核心 patch 文件（sglang.patch、sglang-release_hicache.patch、sglang-top_p.patch、sglang-pull_weights.patch）。由于上游已合入部分定制功能（如 release_memory_occupation、resume_memory_occupation 的树缓存管理），patch 内容大幅减少：sglang.patch 从 638 行减至 148 行，sglang-release_hicache.patch 从 430 行减至 247 行，精简了重复适配。
3. 参数兼容性处理：在 slime/backends/sglang_utils/arguments.py 的 validate_args 中，为新旧参数名（sglang_dp_size vs sglang_data_parallel_size 等）建立双向映射，确保用户以任何方式传入的参数都能正确读取，避免因 SGLang 升级丢失参数。
4. SGLang 引擎运行时适配：在 slime/backends/sglang_utils/sglang_engine.py 中，当启用 memory saver（enable_memory_saver）且用户未显式指定 sglang_cuda_graph_backend_prefill 时，自动将其设为 "disabled"，防止 CUDA graph 在 offload 场景下导致显存压力。

5. 测试配套：新增 test_memory_saver_disables_default_breakable_prefill_cuda_graph 验证上述默认行为，同时调整 test_qwen3_0.6B_parallel_check.py 和 test_qwen3_4B_external_pd.py 中的参数，适配新版本字段名变化。

关键文件：

- docker/patch/latest/sglang.patch（模块 SGLang 补丁；类别 test；类型 test-coverage；符号 release_memory_occupation, SchedulerDisaggregationDecodeMixin, resume_memory_occupation, SchedulerDisaggregationPrefillMixin）：核心 patch，净减少 342 行，说明大量定制已被上游吸收，是升级的关键证据。
- slime/backends/sglang_utils/arguments.py（模块 参数解析；类别 source；类型 core-logic；符号 validate_args）：参数兼容核心逻辑，确保新老版本参数名正确映射，直接影响运行时行为。
- tests/utils/test_sglang_config.py（模块 测试覆盖；类别 test；类型 test-coverage；符号 test_memory_saver_disables_default_breakable_prefill_cuda_graph, CurrentServerArgs, LegacyServerArgs）：新增测试验证新版 cuda_graph_backend_prefill 默认行为，是升级正确性的重要保障。
- docker/patch/latest/sglang-release_hicache.patch（模块 HiCache 补丁；类别 test；类型 test-coverage；符号 init_weight_updater, init_kv_buffer, get_ksize_per_token, DeepSeekV4PagedHostPool）：与 sglang.patch 类似，大幅简化，体现上游对 HiCache 功能的直接支持。
- slime/backends/sglang_utils/sglang_engine.py（模块 引擎适配；类别 source；类型 core-logic；符号 _compute_server_args）：运行时核心文件，新增 cuda_graph_backend_prefill 默认值控制逻辑。
- docker/Dockerfile（模块 Docker 构建；类别 infra；类型 infrastructure）：构建基础设施文件，变更依赖安装步骤以匹配新版本。

关键符号：validate_args, _compute_server_args, test_memory_saver_disables_default_breakable_prefill_cuda_graph, release_memory_occupation, resume_memory_occupation

关键源码片段

slime/backends/sglang_utils/arguments.py

参数兼容核心逻辑，确保新老版本参数名正确映射，直接影响运行时行为。

```
# slime/backends/sglang_utils/arguments.py

def validate_args(args):
    # Older SGLang versions stored these CLI aliases under their long names,
    # while newer versions use the short ServerArgs field names as argparse dests.
    # Keep both attributes available for user code, preferring the newer names
    # when a namespace happens to contain both.
    for current_name, legacy_name in (
        ("sglang_dp_size", "sglang_data_parallel_size"),
        ("sglang_pp_size", "sglang_pipeline_parallel_size"),
```

```
        ("sglang_ep_size", "sglang_expert_parallel_size"),
    ):
        value = getattr(args, current_name) if hasattr(args, current_name) else getattr(args,
        legacy_name)
        setattr(args, current_name, value)
        setattr(args, legacy_name, value)
```

tests/utils/test_sglang_config.py

新增测试验证新版 cuda_graph_backend_prefill 默认行为，是升级正确性的重要保障。

```
# tests/utils/test_sglang_config.py

def test_memory_saver_disables_default_breakable_prefill_cuda_graph(self, monkeypatch):
    from slime.backends.sglang_utils import sglang_engine

    @dataclass
    class CurrentServerArgs:
        enable_memory_saver: bool = False
        cuda_graph_backend_prefill: str | None = None

    @dataclass
    class LegacyServerArgs:
        enable_memory_saver: bool = False

    args = Namespace(
        hf_checkpoint="/tmp/hf",
        seed=1,
        offload_rollout=True,
        rollout_num_gpus_per_engine=1,
        num_gpus_per_node=8,
        sglang_pp_size=1, sglang_dp_size=1, sglang_ep_size=1,
        use_rollout_routing_replay=False, fp16=False,
    )
    compute_kwargs = {
        "rank": 0,
        "dist_init_addr": "127.0.0.1:12345",
        "nccl_port": 12346,
        "host": "127.0.0.1",
        "port": 30000,
        "base_gpu_id": 0,
    }

    # When ServerArgs has enable_memory_saver and no explicit setting,
    # cuda_graph_backend_prefill should default to "disabled".
    monkeypatch.setattr(sglang_engine, "ServerArgs", CurrentServerArgs)
    kwargs, _ = sglang_engine._compute_server_args(args, **compute_kwargs)
    assert kwargs["cuda_graph_backend_prefill"] == "disabled"

    # User explicit setting overrides default.
```

```
args.sglang_cuda_graph_backend_prefill = "full"
kwargs, _ = sglang_engine._compute_server_args(args, **compute_kwargs)
assert kwargs["cuda_graph_backend_prefill"] == "full"

# LegacyServerArgs lacks cuda_graph_backend_prefill field, key should not appear.
monkeypatch.setattr(sglang_engine, "ServerArgs", LegacyServerArgs)
kwargs, _ = sglang_engine._compute_server_args(args, **compute_kwargs)
assert "cuda_graph_backend_prefill" not in kwargs
```

评论区精华

本 PR 无公开 review 评论。设计决策可从变更内容中推断：patch 大幅删减表明 slime 的若干定制（如 release_memory_occupation 等）已被 SGLang 主线采纳，团队选择直接移除冗余补丁；参数兼容采用双向设置（新旧属性同时写入），平衡了平滑迁移与代码清晰。

- 无公开讨论 (other): 无需额外变更，PR 已被合并。

风险与影响

- 风险：
  - 依赖版本兼容风险：新版本 SGLang 可能改变了某些内部接口（如 ServerArgs 字段），patch 的删除若遗漏关键适配可能导致运行时错误。需关注 sglang-engine 的 _compute_server_args 是否全面覆盖新版本差异。
  - 参数兼容逻辑缺陷：validate_args 中的兼容循环假设新旧名至少存在一个；若二者都不存在则会触发 AttributeError。此外，若用户同时设置新旧名但值不同，最终以 current_name 为准，可能违反用户预期。
  - 定制功能回归风险：被删除的 patch（如 release_memory_occupation 的树缓存释放）如果未完全集成到上游，可能在某些配置下导致内存泄漏或训练 hang。建议在完整 CI 套件中重点覆盖 offload 场景。
  - 测试覆盖有限：仅新增一项关于 cuda_graph_backend_prefill 的单元测试，其他行为变化（如 update_weight_version 接口）缺少回归验证。
  - 影响：用户：使用官方 Docker 镜像的开发者将自动获得升级后的 SGLang，需确保配置文件中使用新版参数短名（如 sglang_dp_size 而非 sglang_data_parallel_size），否则旧名仍能工作（兼容层提供）。系统：Docker 镜像体积可能变化；构建时间因依赖更新而有所延长。CI 流水线需同步更新基础镜像。团队：后续维护 patch 的难度降低（补丁更小），但需要持续跟踪上游变动。本次升级为后续引入 SGLang 新功能（如 PD disaggregation 优化）铺平道路。
- 风险标记：依赖升级版本兼容风险，补丁重构可能导致定制遗漏，参数兼容逻辑未全覆盖，新版本行为变化缺少回归测试

关联脉络

- PR #2178 [docker] Update dependencies: 同属 Docker 依赖升级系列，本次在此基础上升级 SGLang 版本。

- PR #2172 [docker] Update training side dependencies: 之前的 Docker 训练依赖升级，为本次 SGLang 升级奠定基础。
- PR #2173 [docker] Update SGLang patch for PD R3 routed experts: 之前更新 SGLang patch 以支持 routed experts，本次进一步升级 SGLang 并简化 patch。
- PR #2169 Merging profiling info into router: 重构了 sglang 补丁结构，为本 PR 的 patch 简化提供了上下文。
- PR #2181 [3/n] Disaggregated rollout: engine-side /pull_weights: 引入了 pull_weights 补丁，本次升级中该补丁也得到兼容性更新。