PR #33850 完整报告

sgl-project/sglang

[diffusion] retire released warmup and decoder flags

合并时间：2026-08-06 23:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33850

执行摘要

- 一句话：退役旧预热与 decoder_tp 参数，统一为 warmup_mode
- 推荐动作：值得精读 server_args.py 的 _reject_retired_args 与 _adjust_warmup，这是“参数退役”的可借鉴范式：先双轨兼容 → 到期后硬拒绝并给出替代提示，避免长期维护两套配置语义。建议关注两点：一是 release notes 中明确标注 breaking change；二是观察外部用户迁移反馈，必要时提供一段临时兼容告警而非直接报错。整体属于高信号的安全清理型重构。

功能与动机

PR body 明确指出：“The deprecated flags have passed their supported migration window and retaining them obscures the real configuration model.” 即 --warmup/--server-warmup/--decoder-tp 的兼容迁移窗口已结束，继续保留会掩盖真实的配置模型（warmup_mode 与 decoder_sp），因此选择直接移除并统一所有调用点。

实现拆解

变更以 python/sglang/multimodal_gen/runtime/server_args/server_args.py 为入口，按以下 5 步完成：

1. 删除退役参数的定义与 CLI 入口：在 ServerArgs 数据类中删除 warmup、server_warmup、decoder_tp 三个字段；在 server_args/disagg.py 中删除 --decoder-tp 参数；在 add_cli_args 中删除 --warmup 与 --server-warmup 两个参数定义。这样 CLI 用户再传旧参数会直接得到 argparse 的 unrecognized arguments 错误。
2. 新增硬校验 _reject_retired_args：在 from_dict 开头调用该静态方法，维护“退役参数 → 替代写法”的映射（decoder_tp → decoder_sp、warmup → warmup_mode=request/off、server_warmup → warmup_mode=server），命中即抛 ValueError。同时删除 _adjust_disagg_parallelism_aliases，不再做静默别名转换。
3. 简化 _adjust_warmup：删除从两个布尔字段推导 warmup_mode 的旧逻辑，改为只围绕 warmup_mode 做归一化：非法值校验 → enable_torch_compile 默认 server → 显式 warmup_resolutions 触发 request → BCG 强制 server → 非 monolithic 角色把 server 降级为 request → 最终 None 默认 off。
4. 切换运行时查询点：所有读取 server_args.warmup / server_args.server_warmup 的地方改为读取 warmup_mode。包括 server_warmup.py 的 should_run_server_warmup、

should_run_explicit_client_warmup、process_received_reqs_with_req_based_warmup；launch_server.py 的 disagg 角色 role_overrides；http_server.py 的 lifespan 预热任务启动条件；denoising.py 的 _maybe_offload_during_compile；diffusion_generator.py 的 _log_summary；benchmark 脚本 bench_diffusion_denoise.py 的 build_sglang_cmd 与参数对齐校验。disagg 角色中 warmup=True + server_warmup=False 的组合被等价替换为 warmup_mode="request"（encoder 角色）和 warmup_mode="off"（其他角色），语义保持一致。

5. 测试与文档配套：test_server_args.py 重写 TestWarmupModeNormalization（去掉 legacy 布尔分支测试，新增 BCG 强制 server 的用例），并新增 test_retired_warmup_config_is_rejected、test_retired_warmup_kwargs_are_rejected 两个硬校验测试；test_cfg_parallel_warmup.py 将 fixture 从布尔字段改为 warmup_mode；gen_perf_baselines.py、test_pi05_e2e.py、gpu_cases.py 同步；.claude/skills 下的 SKILL.md、benchmark-and-profile.md 以及 docs/docs/sglang-diffusion/下多处mdx文档同步替换为--warmup-mode/--decoder-sp。

关键文件：

- python/sglang/multimodal_gen/runtime/server_args/server_args.py（模块 服务参数；类别 source；类型 core-logic；符号 _adjust_disagg_parallelism_aliases, _reject_retired_args）：核心变更文件：删除退役字段与 CLI 参数，新增 _reject_retired_args 硬校验，并重构 _adjust_warmup 为纯 warmup_mode 推导。
- python/sglang/multimodal_gen/test/unit/test_server_args.py（模块 参数测试；类别 test；类型 test-coverage；符号 test_serve_cli_preserves_explicit_warmup_false, test_serve_cli_preserves_explicit_warmup_mode_off, test_serve_cli_preserves_config_warmup_false, test_serve_cli_preserves_config_warmup_mode_off）：测试配套：重写 TestWarmupModeNormalization 并新增退役参数拒绝测试，验证 from_dict/from_kwargs 对新契约的强制行为。
- python/sglang/multimodal_gen/runtime/launch_server.py（模块 启动器；类别 source；类型 core-logic）：disagg 池启动器：把各角色 role_overrides 中的 warmup/server_warmup 布尔组合替换为 warmup_mode，保证分解部署语义不变。
- python/sglang/multimodal_gen/runtime/server_warmup.py（模块 预热逻辑；类别 source；类型 core-logic）：预热运行时：should_run_server_warmup、should_run_explicit_client_warmup 与 request 预热插入路径全部改为按 warmup_mode 判断，是查询点迁移的关键文件。
- python/sglang/multimodal_gen/runtime/server_args/disagg.py（模块 分解参数；类别 source；类型 configuration）：删除 --decoder-tp CLI 参数定义，decoder 并行只保留 --decoder-sp，避免歧义。
- 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）：benchmark 脚本联动：构建 CLI 命令时从 --warmup 改为 --warmup-mode request，夜间对齐校验同步更新跳过列表，避免基准测试触发退役参数报错。

关键符号：_adjust_warmup, _reject_retired_args, _adjust_disagg_parallelism_aliases, should_run_server_warmup, should_run_explicit_client_warmup, should_run_synthetic_server_warmup, process_received_reqs_with_req_based_warmup

关键源码片段

python/sglang/multimodal_gen/runtime/server_args/server_args.py

核心变更文件：删除退役字段与 CLI 参数，新增 _reject_retired_args 硬校验，并重构 _adjust_warmup 为纯 warmup_mode 推导。

```
def _adjust_warmup(self):
    """将 `warmup_mode` 归一化为唯一的预热控制信号。

    取值来自 `WARMUP_MODES`（"off" / "server" / "request"）；
    运行时特征（torch.compile、BCG、disagg 角色）只能在这里推导或修正该值，
    不再维护派生的布尔字段 `warmup` / `server_warmup`。
    """
    if self.warmup_mode is not None and self.warmup_mode not in WARMUP_MODES:
        raise ValueError(
            f"Invalid --warmup-mode {self.warmup_mode!r}; "
            f"expected one of {WARMUP_MODES}."
        )

    # torch.compile 希望启动期完成编译，否则首个真实请求会承担编译延迟。
    if self.enable_torch_compile and self.warmup_mode is None:
        self.warmup_mode = "server"
        logger.info(
            "Automatically enabled server warmup for torch.compile so first "
            "real requests do not pay compile latency. Set --warmup-mode off "
            "to disable this behavior."
        )

    # 显式声明分辨率时至少需要 request 级预热；若用户已选 server 则保留。
    if self.warmup_resolutions is not None and self.warmup_mode in (None, "off"):
        self.warmup_mode = "request"

    # BCG 需要在启动期用合成请求捕获全部 CUDA Graph，因此强制 server 模式。
    if self.enable_breakable_cuda_graph and self.disagg_role == RoleType.MONOLITHIC:
        self.warmup_mode = "server"

    # 解耦角色不承载 HTTP 启动请求，server 预热降级为首个请求预热。
    if self.disagg_role != RoleType.MONOLITHIC and self.warmup_mode == "server":
        self.warmup_mode = "request"

    if self.warmup_mode is None:
        self.warmup_mode = "off"

@staticmethod
def _reject_retired_args(kwargs: dict[str, Any]) -> None:
```

```
    """在 `from_dict` / `from_kwargs` 入口拒绝已退役的兼容参数。

    旧参数迁移窗口已过，继续接受只会掩盖真实的配置模型；这里直接抛错，
    并给出替代写法，让外部调用方快速迁移到 `warmup_mode` 与 `decoder_sp`。
    """
    retired = {
        "decoder_tp": "decoder_sp for decoder/VAE parallel decode",
        "warmup": "warmup_mode=request or warmup_mode=off",
        "server_warmup": "warmup_mode=server",
    }
    for arg, replacement in retired.items():
        if arg in kwargs:
            raise ValueError(
                f"{arg} is retired; use {replacement} instead."
            )
```

python/sglang/multimodal_gen/test/unit/test_server_args.py

测试配套：重写 TestWarmupModeNormalization 并新增退役参数拒绝测试，验证 from_dict/from_kwargs 对新契约的强制行为。

```
class TestWarmupModeNormalization(unittest.TestCase):
    """`_adjust_warmup` 现在只解析规范的 `warmup_mode`，不再有 legacy 布尔分支。"""

    def _resolve(
        self,
        *,
        warmup_mode=None,
        warmup_resolutions=None,
        enable_torch_compile=False,
        enable_breakable_cuda_graph=False,
        disagg_role=None,
    ):
        from sglang.multimodal_gen.runtime.disaggregation.roles import RoleType

        sa = ServerArgs.__new__(ServerArgs)
        sa.warmup_mode = warmup_mode
        sa.warmup_resolutions = warmup_resolutions
        sa.enable_torch_compile = enable_torch_compile
        sa.enable_breakable_cuda_graph = enable_breakable_cuda_graph
        sa.disagg_role = RoleType.MONOLITHIC if disagg_role is None else disagg_role
        sa._adjust_warmup()
        return sa

    def test_breakable_cuda_graph_forces_server_warmup(self):
        # BCG 需要启动期捕获全部 CUDA Graph，必须走 server 合成预热。
        sa = self._resolve(enable_breakable_cuda_graph=True)
        self.assertEqual(sa.warmup_mode, "server")

    def test_disagg_role_disables_server_warmup(self):
```

```
        # 解耦角色不承载 HTTP 启动请求，server 预热降级为 request。
        from sglang.multimodal_gen.runtime.disaggregation.roles import RoleType

        sa = self._resolve(
            warmup_mode="server",
            disagg_role=RoleType.DENOISER,
        )
        self.assertEqual(sa.warmup_mode, "request")
```

python/sglang/multimodal_gen/runtime/launch_server.py

disagg 池启动器：把各角色 role_overrides 中的 warmup/server_warmup 布尔组合替换为 warmup_mode，保证分解部署语义不变。

```
    # 为每个 disagg 角色实例构造覆盖参数；并行度全部由 get_role_parallelism 统一推导。
    role_overrides = {
        "disagg_role": role_type,
        "disagg_mode": True,
        "pool_work_endpoint": work_eps[inst_idx],
        "pool_result_endpoint": result_ep,
        "num_gpus": num_role_gpus,
        # 仅 encoder 角色执行预热（request 模式，等价于旧的 warmup=True + server_warmup=
        False）；
        # denoiser / decoder 角色在分解池中不预热。
        "warmup_mode": "request" if role_type == RoleType.ENCODER else "off",
        "scheduler_port": find_port(port_cursor),
        "master_port": find_port(port_cursor + 100),
    }

    base_dict.update(role_overrides)
    base_dict.pop("pipeline_config", None)
    role_args = ServerArgs.from_kwargs(**base_dict)
```

评论区精华

本 PR 没有收到实质性 review 评论（review_comments_count 为 0）。3 条 issue 评论均为作者 mickqian 自己触发的 CI 命令 /tag-and-rerun-ci 与 /tag-and-rerun-ci extra，对应 PR body 中的“Remote CI lint was triggered with /tag-and-rerun-ci; no local tests were run under the SGLang-Diffusion guidance”。作者自审自合（author 与 merged_by 均为 mickqian），关键决策（直接报错拒绝而非继续降级兼容）未经过外部 reviewer 质询，风险主要由常规 + extra 两套远程 CI 覆盖。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 破坏性参数移除：所有仍传 --warmup、--server-warmup、--decoder-tp 的 CLI 用户会立即因 argparse 报 unrecognized arguments；通过配置文件或

ServerArgs.from_dict/from_kwargs 传入旧键会触发 _reject_retired_args 抛 ValueError。仓库内调用点已全部更新，但仓库外的脚本、部署配方未纳入管控，需要发布说明高亮该 breaking change。

2. disagg 角色语义保持：launch_server.py 中 encoder 角色从 warmup=True + server_warmup=False 改写为 warmup_mode="request"，与原语义等价；但若外部代码隐式依赖过“unset 时 warmup 默认行为”，可能受影响。_adjust_warmup 中 BCG 先设 server、disagg 再降级为 request 的顺序保证了非 monolithic 角色不会执行 HTTP 启动预热。
3. 测试覆盖依赖单测与 CI：本地未跑测试，依赖远程 CI（Run #31109524523 常规 + #31109523671 extra）验证，test_server_args.py 对退役参数拒绝路径有直接覆盖，但未覆盖所有历史组合的迁移场景。 - 影响：用户影响：使用 SGLang Diffusion 的所有 CLI 用户与配置中心必须从 --warmup/--server-warmup/--decoder-tp 迁移到 --warmup-mode/--decoder-sp，属于用户可见的破坏性变更。系统影响：消除了布尔字段与枚举字段双轨并存的配置模型，_adjust_warmup 逻辑显著简化，运行时查询点统一为单一枚举判断，降低后续误用风险。团队影响：作为近期 diffusion hygiene 系列（33843-33845 等）的一部分，继续收敛配置面与入口面，为后续扩散功能迭代提供更干净的参数基础。 - 风险标记：破坏性参数移除，CLI 行为变更，配置契约变更，无外部 review

关联脉络

- PR #33843 [diffusion] consolidate pipeline core hygiene: 同批次 diffusion hygiene 重构，同样修改了 server_args.py，统一 pipeline core 的重复逻辑与参数表面。
- PR #33844 [diffusion] simplify disaggregation transport hygiene: 同批次分解传输层清理；本 PR 同步调整 launch_server.py 中 disagg 角色的预热配置，两者共同收敛分解部署的参数契约。
- PR #33845 [diffusion] centralize entrypoint API hygiene: 同一入口 API 卫生清理系列，涉及 http_server.py 与 diffusion_generator.py；本 PR 对这两个文件仅做了 warmup 判断点的单一替换。