PR #36972 完整报告

sgl-project/sglang

config: the resolution callbacks into the record go to zero

合并时间：2026-08-29 19:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36972

执行摘要

- 一句话：清零 arg_groups 对 ServerArgs 的 96 处回调用例
- 推荐动作：值得精读。本 PR 是配置系统架构收敛的关键一环，展示了如何系统性消除跨层反向依赖。重点关注：(1) test_no_hook_reads_a_field_off_the_record 包级守卫的设计思路；(2) from ... import 拷贝语义对测试接缝的影响分析；(3) memo 放在 record 上随 pickle 传递的取舍。同时建议跟踪 Codex 指出的 cutedsl_moe_max_num_tokens 遗漏调用是否已在后续修复中解决。

功能与动机

PR body 明确说明：在 #36792 之后解析管线已迁入 arg_groups/，但仍大量回读 record——普查发现 24 个 ServerArgs 成员上有 96 个调用点、503 行代码，其中 14 个成员除了解析期助手外没有任何其他调用者，纯属历史遗留。作者目标是让 arg_groups/ 对 record "no method call and no property read"，使依赖方向单一化。

实现拆解

本 PR 按 5 个 commit 分五步完成迁移：

1. Resolution-only 助手移入所属 hook 家族（commit 1） - generate_{decode,prefill,cpu}_*_batch_sizes 与 apply_cuda_graph_disaggregation_roles 移入 cuda_graph_hook.py； - reserve_for_graph_mb、reserve_for_deepep_a2a_mb、adjust_mem_fraction_for_vlm 移入 memory_hook.py； - 两个 dispatch-token 预算移入 moe_hook.py；is_mistral_native_format 移入 model_path_hook.py；is_attention_backend_not_set、get_default_attn_backend 移入 overrides.py。 - 关键的 get_default_attn_backend 原先是读 record 上的 tp_size（原始输入），而非解析决定的结果，迁移后改为通过 view 读取，顺带修复了该缺陷。
2. 声明接缝脱离 record（commit 2） - _resolved、_declare、_late_resolution 三个仅做转发的成员方法删除，61 个调用点改为直接调用 resolved_view、declare_resolution、declare_late_resolution。
3. Attention backend 对合并为一个函数（commit 3） - _resolved_attention_backends 与 get_attention_backends 合并为 attention_backends_of(resolved_view(record))； - use_mla_backend 与 should_report_expert_balancedness 变为自由函数。

4. 模型配置构建改为函数（commit 4） - get_model_config（56 个调用点）变为 overrides.model_config_of； - memo（_model_config 等）留在 record 上作为 resolution 中间产物，因为需要随 pickle 传到子进程，但回调本身离开。
5. 最后三个计算读加最后两个属性（commit 5） - post_capture_kv_sizing_planned、cutedsl_moe_max_num_tokens、max_prefill_buffer_tokens 改为纯函数，发布后的读取方改走 runtime_context 中的既有兄弟函数； - mamba_cache_chunk_size、max_speculative_num_draft_tokens 两个 property 同样迁移，因为属性读取不计入函数调用普查，这两个是补漏。

测试与接缝配套：新增 test_no_hook_reads_a_field_off_the_record 包级守卫测试，遍历 arg_groups/ 防止未来回读；use_mla_backend 等改为调用点导入，使 patch("...overrides.use_mla_backend") 成为单一测试接缝；hisparse_hook._is_hip 改为向 utils.common 询问平台探针。

未覆盖的一处：Codex review 指出 qwen35_flashinfer_fusion.py:40 仍调用 server_args.cutedsl_moe_max_num_tokens()，迁移后该实例会抛 AttributeError，需要在后续 commit 中把该调用方改为新自由函数。

关键文件：

- python/sglang/srt/server_args.py（模块 配置解析；类别 source；类型 dependency-wiring；符号 _declare, _apply_cuda_graph_disaggregation_roles, post_capture_kv_sizing_planned, reserve_for_graph_mb）：核心减重对象：删除 624 行、29 个成员方法，_declare、_apply_cuda_graph_disaggregation_roles、post_capture_kv_sizing_planned 等全部迁出，是本次重构的 " 减法 " 主战场。
- python/sglang/srt/arg_groups/overrides.py（模块 配置解析；类别 source；类型 dependency-wiring；符号 record_of, is_attention_backend_not_set, get_default_attn_backend, use_mla_backend）：迁移的主接收方：新增 346 行，收纳 record_of、model_config_of、use_mla_backend、attention_backends_of 等核心自由函数，并修复 get_default_attn_backend 误读 raw tp_size 的缺陷。
- python/sglang/srt/arg_groups/cuda_graph_hook.py（模块 CUDA 图；类别 source；类型 core-logic；符号 generate_prefill_cuda_graph_batch_sizes, generate_decode_cuda_graph_batch_sizes, generate_cpu_graph_batch_sizes, apply_cuda_graph_disaggregation_roles）：CUDA graph 相关助手集中迁入：generate_prefill/decode/cpu_cuda_graph_batch_sizes 与 apply_cuda_graph_disaggregation_roles 落位，并把 get_model_config() 调用替换为 model_config_of()、_resolved_attention_backends() 替换为 attention_backends_of()。

关键符号：model_config_of, record_of, attention_backends_of, use_mla_backend, post_capture_kv_sizing_planned, cutedsl_moe_max_num_tokens, generate_decode_cuda_graph_batch_sizes, generate_prefill_cuda_graph_batch_sizes, generate_cpu_graph_batch_sizes, apply_cuda_graph_disaggregation_roles, is_mistral_native_format, required_mori_dispatch_tokens_per_rank, required_pplx_dispatch_tokens_per_rank

关键源码片段

python/sglang/srt/server_args.py

核心减重对象：删除 624 行、29 个成员方法，_declare、_apply_cuda_graph_disaggregation_roles、post_capture_kv_sizing_planned 等全部迁出，是本次重构的 " 减法 " 主战场。

```
# python/sglang/srt/server_args.py —— 迁移后的 record 边界

# 本 PR 的核心减法：以下成员方法全部删除，调用方改走 arg_groups 自由函数。
# 删除后 record 只保留字段、原始输入快照、resolution stash 与 memo。

# 删除前（base 版本）的形态示例：
# def _declare(self, source: str, **fields: Any) -> None:
# """This record's handlers declaring their resolution writes."""
# from sglang.srt.arg_groups.overrides import declare_resolution
# declare_resolution(self, source, **fields)
#
# 删除后：61 个调用点直接调用 arg_groups.overrides.declare_resolution(record, ...)

# 删除前：
# def post_capture_kv_sizing_planned(self) -> bool:
# """Whether KV sizing was decided post-capture."""
# ...
# 删除后：同逻辑移入 overrides.py 的自由函数 post_capture_kv_sizing_planned(record)，
# 且发布后读取方一律走 runtime_context 中的既有兄弟函数。

# 关键保留决策：model_config_of 的 memo（_model_config / _model_config_built_from）
# 仍留在 record 上 —— 它是 resolution 中间产物，必须随 pickle 传到子进程，
# 且解析期没有其他 per-record 存储位置。离开的是回调，不是缓存。

def replace_resolved(self, source: str, **changes: Any) -> ServerArgs:
    """A copy of this record that stays resolved, and says what it changed.

    `dataclasses.replace` 构建的新实例不带 raw snapshot、stash 与 finished 标志，
    下一次 publish 会重新解析并丢掉所有决议；Ray 路径替换 dist_init_addr 时走到这里。
    变更追加进 stash 而非直接写字段：投影读 raw snapshot + declarations，
    直接写字段会让 copy 自己的决议发布成 parent 的 raw 值。
    """
    # 显式枚举实例上除字段外的一切：raw snapshot、stash、以及解析期 memo ——
    # 包括 model-configuration memo，copy 直接继承而非重建。
    field_names = {field.name for field in dataclasses.fields(self)}
    for name, value in vars(self).items():
        ...
    object.__setattr__(replacement, "_resolution_finished", True)
    return replacement
```

python/sglang/srt/arg_groups/cuda_graph_hook.py

CUDA graph 相关助手集中迁入：generate_prefill/decode/cpu_cuda_graph_batch_sizes 与 apply_cuda_graph_disaggregation_roles 落位，并把 get_model_config() 调用替换为 model_config_of()、_resolved_attention_backends() 替换为 attention_backends_of()。

```
# python/sglang/srt/arg_groups/cuda_graph_hook.py —— 迁移后的核心分支

def disable_tc_piecewise_cudagraph_if_incompatible(server_args: Any):
    """TcPiecewise（torch.compile + piecewise）与下列配置不兼容。

    大部分规则来自 torch.compile / dynamo 的限制；每条规则读 resolved 或 resolving
    view 上已经决议的值，不再触碰 record 方法。
    """
    # 函数级导入：避免模块顶层循环依赖，同时保持调用点接缝可 patch
    from sglang.srt.arg_groups.overrides import model_config_of

    cfg = resolving_view(server_args)
    rules = [
        # model-arch 黑名单：原 server_args.get_model_config().is_piecewise_...
        # 迁移后经自由函数读取同一个 memo 化的模型配置
        ("model-arch blacklist",
         lambda: model_config_of(server_args).is_piecewise_cuda_graph_disabled_model),
        ("DP attention", lambda: resolved_view(server_args).enable_dp_attention),
        ("full torch.compile mode", lambda: cfg.enable_torch_compile),
        ("pipeline parallelism (pp_size > 1)", lambda: cfg.pp_size > 1),
        # LoRA 在 tc_piecewise 下被 dynamo 阻断（per-batch LoRABatchInfo
        # rebinds 破坏 break guards）；breakable/full 支持 LoRA
        ("LoRA", lambda: bool(cfg.lora_paths) or cfg.enable_lora),
        ("multimodal model",
         lambda: model_config_of(server_args).is_multimodal
         and not model_config_of(
             server_args
         ).is_multimodal_piecewise_cuda_graph_supported),
        ("CPU offload / hierarchical cache",
         lambda: cfg.cpu_offload_gb > 0 or cfg.enable_hierarchical_cache),
    ]
    ...
```

评论区精华

唯一实质 review 来自 Codex bot 的 P1 建议：

> Update the remaining CuTeDSL budget caller — When Qwen3.5 FlashInfer MNNVL CuTeDSL fusion prepares its workspace, resolve_max_m() still calls server_args.cutedsl_moe_max_num_tokens() in qwen35_flashinfer_fusion.py:40. This commit removes that ServerArgs method and only introduces this free function, so a real ServerArgs instance raises AttributeError during prepare_qwen35_flashinfer_fusion() before CUDA-graph capture.

即 commit 5 删除方法后遗漏了一处运行时调用方，该处会在 CUDA graph 捕获前抛 AttributeError。PR 已合并，该问题需要在后续修复中跟进。

- cutedsl_moe_max_num_tokens 遗漏调用方导致 AttributeError (correctness): 建议更新该调用方使用新自由函数；PR 已合并，需后续修复确认。

风险与影响

- 风险：本 PR 是 50 文件、约 1000 行净改动的大规模重构，风险集中在：

1. 遗漏调用方：Codex 已指出 qwen35_flashinfer_fusion.py:40 的 cutedsl_moe_max_num_tokens() 调用未被迁移，真实 ServerArgs 会抛 AttributeError。这是 commit 5 引入的运行时回归源头，且发生在 Qwen3.5 MNNVL 特定路径上，常规 CI 未必覆盖。
2. 测试接缝语义变化：from ... import 是拷贝语义，原先 patch ServerArgs.use_mla_backend 类属性即可全局生效；现在改为按调用点导入后，所有测试必须改为 patch 模块绑定，遗漏会导致测试静默失效或误报。
3. memo 生命周期：model_config_of 的 memo 仍挂在 record 上，依赖 pickle 到子进程的语义；若未来出现不在 record 上创建 memo 的路径，可能重复解析或读到过期配置。
4. 导入抖动：server_args.py 删除了 glob、math、is_hip 等大量导入，若有其他模块仍经 server_args 间接使用这些符号会立刻断裂。 - 影响：
   - 对用户：无直接行为变化，但 Qwen3.5 MNNVL CuTeDSL 路径存在潜在 AttributeError 回归。
   - 对系统：ServerArgs 减少 624 行、29 个方法，依赖方向收敛为 arg_groups → overrides 单向；后续配置解析的 patch 接缝从类属性改为模块函数，测试维护方式发生范式转变。
   - 对团队：这是 5-PR 堆叠系列的第 2 环，后续 #36973-#36975 都依赖本 PR 的接口边界；合并后需要全量 CI 验证，特别是 Qwen3.5、MNNVL、CUDA graph 相关用例。
   - 风险标记：遗漏调用方回归，测试接缝语义变化，memo 生命周期依赖，跨 50 文件大规模重构，Qwen3.5 MNNVL 特定路径风险

关联脉络

- PR #36896 config: the resolution pipeline's dispatcher leaves the record: 本堆叠系列第 1 环，让 resolution pipeline 的 dispatcher 脱离 record，本 PR 基于其上构建。
- PR #36973 config: six more runtime readers ask the bags: 系列第 3 环，运行时 6 个模块改从配置 bag 读参数，承接本 PR 建立的自由函数边界。
- PR #36974 config: the dead record parameters go: 系列第 4 环，移除本 PR 普查暴露出的 10 个死参数与 18 个调用点。
- PR #36975 config: the lazy imports that buy nothing become eager: 系列第 5 环，把无收益的惰性导入改回 eager，与本 PR 的导入收敛一脉相承。
- PR #36925 config: CI vehicle for the arg_groups server_args series: CI 载体 PR，专门运行整个系列对 main 的验证而非合入。
- PR #36792 config: the resolution pipeline lives in arg_groups: 本 PR 动机中提到的基础：解析管线迁入 arg_groups 后仍回读 record，催生了本次清零。