

PR #34267 完整报告

sgl-project/sglang

config: pin the supplied-instance surface that a raw record would change

合并时间: 2026-08-15 15:40

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34267>

执行摘要

- 一句话: 新增配置暴露面钉测与 EPD 扳机, 护航 step-12 迁移
- 推荐动作: 值得精读, 尤其对正在做大型配置或数据契约迁移的团队: 这是「用可执行测量钉住迁移前沿」的高质量示范。重点看三点: `_MATRIX` 驱动的 `written set` 派生与矩阵完备性规则 (`passed-in` 字段排除、每个条目必须让 `resolution` 做决策); `may-write` 静态语义如何取代机制枚举; EPD tripwire 用真实解析而非固定 `double` 来保证翻转时必然报警。后续动作: `step-12 flip` 落地时逐项消化 `_EXPOSED` 清单, 并把 `chain` 拼写 (`model_runner.server_args.field`) 扩展为独立 `pin step`。

功能与动机

`step-12` 会让 `ServerArgs` 实例停留在用户原始输入, `resolution` 填充的字段不再出现在实例上。PR body 强调「Measured, not guessed: 297 distinct file/field pairs read one of the 125 fields resolution can write」; 第二轴则钉住发布后经 `get_context().override(...)` 写入、`supplied instance` 仍然读取的字段——这类读取从 `override` 落地起就用启动值作答。目的不是列出 bug, 而是把决定「resolved 值从哪来」的时机锁定在读取被写下的那一刻, 而不是迁移翻转期间。

实现拆解

1. 新增双轴暴露面钉测主体: `test/registered/unit/test_supplied_instance_exposure_ratchet.py` (+1165 行) 中 `TestSuppliedInstanceExposure` 同时注册 CPU (`base-a-test-cpu`, 估时 20 秒) 与 CUDA (`base-b`, `1-gpu-small`) 套件。类覆写 `_callTestMethod` 让测试 body 只执行一次——`ratchet` 不匹配是确定性的, CI 重试会在 `cleanup` 栈半展开的状态下重入。
2. 推导 `resolution` 写集合 (轴 1): `_MATRIX` 每条目代表一个解析家族 (EAGLE、DP、DWDP、HiCache、disaggregation、CP、LoRA、FP8 `page_size`、MIS 等), 真实构造 `ServerArgs` 后与 `dataclass` 默认做 `construct-and-diff`; 条目显式传入的字段被排除, 因此 DWDP、MIS 这类自写字段必须有专属条目。每个条目解析前恢复 `pristine` 进程状态 (`environ + EnvField._set_to_none`), `SGLANG_IS_IN_CI` 作为显式环境条目。`declare_late_resolution` 的写入者按关键字静态收集, 动态 `**detected` 站点用 `_LATE_RESOLUTION_DYNAMIC_SITES` 表防漂移。此后 `written set` 又统一为 `may-write` 语义: 把 `hook` 赋值 (`arg_groups/`)、记录自身方法赋值 (`mooncake layout` 重写、`deepseek-EP` 模式、`random_seed` 填充)、声明式 `override registry` (

MODEL_OVERRIDES 与 provider 注册)、alias 归一化元组全部静态收集后与矩阵并集。

3. 读取 census 与双向 pin: 扫描全仓 AST, 识别三种读取拼写——参数属性 server_args.field、字面量 getattr(server_args, 'field', default)、parked 形式 self.x = server_args 后的 self.x.field; owners (server_args.py、runtime_context.py、arg_groups/) 豁免; source= 不计入字段。found = reads $\cap$ written 与 _EXPOSED 双向比对: 新 pair 失败、消失 pair 失败。解析失败的模块直接让测试失败 (顺带发现 models/qwen2_classification.py 的 UTF-8 BOM, 扫描改用 utf-8-sig)。精确 pin 按主机拆分: _EXPOSED 处处断言, _EXPOSED_CUDA_ONLY 仅 CUDA 断言 (当前为空), AMD 有意不注册并写入 header 理由。
4. 轴 2: post-publish override 收集: 对 get_context().override(...) 做静态收集, 解析 **kwargs 展开 (dict 字面量、条件表达式、名称绑定 + 常量下标存储的 HiCache attach 形态), 不可解析则失败; update_server_args 按名豁免 (它的键集属于调用方)。该轴钉住 75 对 / 18 个字段。
5. EPD tripwire: test_kimi_k3_encoder_mode.py 新增 test_epd_rejection_reads_the_resolved_transfer_backend, 用迷你 Kimi-K3 配置真实解析一个 language-only TP2 启动, 断言 encoder_transfer_backend == 'zmq_to_tokenizer' 且 _reject_missing_dispatched_encoder_embedding 返回 503。测试前后恢复 environ 与 EnvField flags; mamba_radix_cache_strategy='no_buffer' 规避 CPU CI runner 无 GPU 栈的混合缓存分支。

关键文件:

- test/registered/unit/test_supplied_instance_exposure_ratchet.py (模块 暴露面钉测; 类别 test; 类型 test-coverage; 符号 TestSuppliedInstanceExposure, _MATRIX, _PASSED, _EXPOSED): PR 主体: 双轴暴露面双向钉测, 1165 行纯新增。包含 _MATRIX 解析派生 written set、三种读取拼写的全仓 census、_EXPOSED/_EXPOSED_CUDA_ONLY 双向 pin、进程状态隔离与 CPU/CUDA 双注册。
- test/registered/unit/disaggregation/test_kimi_k3_encoder_mode.py (模块 编码器模式; 类别 test; 类型 test-coverage; 符号 test_epd_rejection_reads_the_resolved_transfer_backend, env_field_flags): 为 EPD 守卫增加 tripwire: 真实解析 Kimi-K3 TP2 纯语言启动, 保证 step-12 后 _reject_missing_dispatched_encoder_embedding 的静默失效会被测试失败捕获。

关键符号: TestSuppliedInstanceExposure._callTestMethod,
TestSuppliedInstanceExposure.setUp, _restore_process_state, env_field_flags,
test_epd_rejection_reads_the_resolved_transfer_backend

关键源码片段

test/registered/unit/test_supplied_instance_exposure_ratchet.py

PR 主体: 双轴暴露面双向钉测, 1165 行纯新增。包含 _MATRIX 解析派生 written set、三种读取拼写的全仓 census、_EXPOSED/_EXPOSED_CUDA_ONLY 双向 pin、进程状态隔离与 CPU/CUDA 双注册。

```
# test/registered/unit/test_supplied_instance_exposure_ratchet.py
```

```

# 双轴暴露面钉测: step-12 之后 ServerArgs 实例会保持用户原始输入,
# 凡 callee 从参数 server_args 读取「resolution 会填充」的字段,
# 都会从有效值变成 CLI 默认值。双向 pin: 新增 pair 就是新的 step-12
# 工作, 消失 pair 则须删除条目——清单是测量, 不是记忆。

# 每个矩阵条目只让 resolution 替它做决策; 显式传入的字段会被排除。
# 因此 DWDP / MIS 这类「自己写字段」的家族必须专门设条目,
# 否则它们的写入永远进不了 written set。
_MATRIX = (
    {}, # 纯默认
    {
        'speculative_algorithm': 'EAGLE',
        'speculative_num_steps': 3,
        'speculative_eagle_topk': 1,
        'speculative_num_draft_tokens': 4,
    },
    {'dp_size': 2, 'tp_size': 2, 'enable_dp_attention': True},
    # DWDP 自己写 dp_size / enable_dp_attention / ep_size; 上面的 DP 条目
    # 显式传入了这些字段, 所以必须让 DWDP 条目自行解析, 才能 pin 住
    # DP/EP 拓扑家族共 37 个读取对 (launcher、控制器、tokenizer、spec workers)。
    {'tp_size': 2, 'dwdp_size': 2},
    {'enable_hierarchical_cache': True, 'hichache_ratio': 2.0},
    {'disaggregation_mode': 'prefill'},
    {'tp_size': 2, 'attn_cp_size': 2},
    {'enable_lora': True, 'max_lora_rank': 16},
    {'kv_cache_dtype': 'fp8_e4m3', 'page_size': 64},
    # MIS 自己写 disable_radix_cache; attention_backend 显式传入,
    # 因为 handler 只断言 flashinfer 而不是切换它。
    {'enable_mis': True, 'attention_backend': 'flashinfer'},
)

# 只有真正的构造入参才豁免; tokenizer_path / served_model_name 总是由
# resolution 从 model_path 填充, 它们的读取者与普通 pair 一样是暴露面。
_PASSED = frozenset({'model_path', 'device', 'random_seed'})

# 双向精确清单 (节选): 新 pair 与消失 pair 都会让测试失败,
# 迫使「resolved 值从哪来」的决定发生在读取被写下的时刻。
_EXPOSED = {
    ('configs/embedding_model_spec.py', 'chunked_prefill_size'),
    ('configs/embedding_model_spec.py', 'cuda_graph_config'),
    ('configs/model_config.py', 'dtype'),
    ('configs/model_config.py', 'quantization'),
    ('disaggregation/encode_receiver.py', 'encoder_transfer_backend'),
    ('disaggregation/encode_server.py', 'tokenizer_path'),
    ('distributed/bootstrap.py', 'moe_a2a_backend'),
    ('entrypoints/engine.py', 'dp_size'),
    ('lora/marlin_lora_temp/policy.py', 'lora_paths'),
    ('managers/data_parallel_controller.py', 'soft_watchdog_timeout'),
    ('managers/tokenizer_manager.py', 'mm_feature_transport'), # parked 形式

```

}

评论区精华

这是一次典型的「自动化 review 找洞、作者逐轮封堵」迭代，最有价值的交锋集中在 census 覆盖面与测试真实性上：

Codex (P2) : `getattr(server_args, 'field', default)` 的读取被忽略, `embedding_model_spec.py` 读 `cuda_graph_config` 等未入册。ch-wan: Right — 收集器现在也数 `getattr` 字面量形式, 三个例子全部 pin 进 `_EXPOSED`, 计数 1121→1153 读、37→40 对; 反向验证探针读取会失败。

Codex (P2) : `MMEncoder` 把对象 `park` 成 `self.server_args`, `encode_server.py` 读 `self.server_args.encoder_transfer_backend` 未被统计。ch-wan: census 现在跟踪同 class 内的 `self.x = server_args` 赋值并统计 `self.x.field`, 新增 9 读 / 5 对; `parking` 到其他对象、容器或计算名仍不可见, `docstring` 已写明。

ch-wan (self-review) : 机制枚举每轮 review 都漏 (`DFLASH hook`、`MODEL_OVERRIDES` 声明式 `registry`、`alias` 归一化循环), `syntax` 静默跳过还会伪装成表面收缩。作者裁决: `written set` 采纳 `may-write` 语义整体收集, 解析失败 `fail loudly`, `BOM` 文件当场暴露; `_EXPOSED` 拆成 `_EXPOSED` + `_EXPOSED_CUDA_ONLY`, 杜绝「共享精确列表无法表达 `CUDA-only` 写入」的死结。

ch-wan 最终 re-review (clean) : Prior bugs and suggestions addressed on this head... Ratchet + EPD tripwire look solid. No residual findings.

- `getattr` 字面量读取未计入 census (correctness): 作者扩展 census 覆盖 `getattr` 字面量形式, 三个例子全部 pin 进 `_EXPOSED`, 计数 1121→1153 读、37→40 对, 并用探针读取反向验证 `ratchet` 会精确报出 pair。
- `parked` 形式 `self.x = server_args` 与继承边界 (correctness): 同 class 内的 `parked` 读取已入 census (新增 9 读 / 5 对, 含 `managers/tokenizer_manager.py` 的 `mm_feature_transport`); 继承场景作为已知边界处理, `docstring` 明确 `parking` 到其他对象、容器或计算名仍不可见。
- 矩阵完备性: `DWDP` / `NGRAM` / `MIS` 必须让 `resolution` 自己做决策 (testing): 作者补 `{tp_size: 2, dwdp_size: 2}` (`DP/EP` 拓扑家族 37 对, 含 `TokenizerControlMixin.init_communicators`) 与 `{enable_mis, attention_backend: 'flashinfer'}` (`mem_cache/kv_cache_builder.py` 等), 并确立规则: 条目必须让 `resolution` 做出它要测的那个决策。
- `post-publish override` 的 `**kwargs` 展开 (correctness): 收集器现在静态解析 `dict` 字面量展开 (字面量、条件之间的字面量、函数内绑定的名称加常量下标存储), 无法解析则 `fail loudly`; `update_server_args` 按名豁免。由此 pin 住 6 个 `HiCache` 写 / 读对。
- 写集合采用 `may-write` 语义而非机制枚举 (design): 采纳 maintainer 裁决: `written set` 静态收集所有可写机制 (`hook` 赋值、记录自身方法赋值、声明式 `override registry`、`drift` 守卫的 `alias` 元组、`late-resolution` 关键字), 与矩阵并集后 pin 到 297 对 / 125 字段; 静态收集到但永不触发的写点是上游该删的死分支, 而不是缩小 census 的理由。

- EPD tripwire 必须来自真实解析，且原始值应为 auto (testing): tripwire 改用真实 Kimi-K3 TP2 迷你配置解析出 record，断言守卫以 503 拒绝且 `encoder_transfer_backend == 'zmq_to_tokenizer'`；step-12 后同一启动传给守卫 'auto'，拒绝静默停止，测试失败即信号。
- 进程状态隔离与 CI 重试 (testing): 矩阵每个条目解析前恢复 pristine 状态 (environ + MRO 遍历的 `_set_to_none` 标志)；tripwire 同样快照恢复；类覆写 `_callTestMethod` 只跑一次 body。
- 单一精确 pin 无法表达 capability-gated 写入 (design): 拆为 `_EXPOSED` (处处断言) 与 `_EXPOSED_CUDA_ONLY` (仅 CUDA 断言，当前为空)；AMD 有意不注册，理由写入 header (`is_hip()` 门控写会移动精确集合，任何 pinning host 都无法验证)。
- census 剩余边界: chain 拼写、asdict、config_value、方法调用 (question): chain 拼写已被 docstring 点名并作为独立 step；asdict / config_value / 方法调用属于已知边界，未在本 PR 内 pin。step-12 翻转时这些读取会静默看到默认值，是本 ratchet 的 false-negative 空洞。

风险与影响

- 风险：
 1. census 已知盲区成为 step-12 切换时的未钉暴露面：chain 拼写 `model_runner.server_args.field` (约 150 读)、继承类 `parked` 读取、`dataclasses.asdict(server_args)` 批量序列化、`config_value('load_format')` 动态读、`get_attention_backends()` 方法内字段读取都不在 census 内，raw 实例下会静默看到默认值且本测试不会失败。
 2. written set 依赖矩阵与静态收集的完备性：任何未入 `_MATRIX` 的新解析家族、或声明式 registry 中非字面键的写入，仍可能漏出；hook may-write 收集缓解了机制枚举问题，但 `update_server_args` 按名豁免的面积取决于调用方键集。
 3. 主机差异：CPU 与 CUDA 的 written set 不同，`_EXPOSED_CUDA_ONLY` 当前为空；AMD 上 `is_hip()` 门控写入完全不在 pin 范围内 (有意为之，但 ROCm 解析面没有等价精确 pin)。
 4. 测试基建风险：真实解析 + 全仓 AST 扫描，CPU 估时 20 秒；CUDA runner 不可用则 capability 面覆盖缺失；若新增 sticky 的 EnvField 状态而恢复逻辑未同步，会污染同 worker 后续文件。
 5. 开发摩擦：双向 pin 意味着任何新增的 `server_args` 读取 `resolution` 字段必须先入册，否则 CI 红；这是有意设计，但在 step-12 之前会给功能开发增加显式入册成本。- 影响：对用户与运行时零影响 (纯测试)。对系统而言，它为 step-12 「raw-input config」转换提供了精确到 (file, field) 的工作清单与回归防线，297 + 75 对覆盖 disaggregation、eplb、lora、managers、distributed、layers、entrypoints 等数十个模块。对团队而言，所有涉及 `ServerArgs` 字段读取的 PR 都会经过此 ratchet 的评审，配置一致性测试家族 (与 `test_resolution_is_reproducible.py` 等) 进一步成形；CI 新增 CPU (base-a-test-cpu) 与 CUDA (base-b, 1-gpu-small) 双注册，扩展了覆盖面积。- 风险标记：纯测试无生产代码变更，census 已知盲区 (chain 拼写等)，依赖真实解析矩阵与双 runner，双向 pin 增加开发摩擦

关联脉络

- PR #34269 config: state the bag contract as what resolution produced, and the skill rule that goes with it: 同属于 step-12 配置治理系列 (gc-sr) : 该 PR 定义 bag 契约与 skill 规则, 本 PR 的 plan doc global_context/12-raw-input-config.md 是同一迁移计划的组成部分, 一个在文档 / 测试契约层铺路, 一个在暴露面测量层铺路。
- PR #34913 [CI] Move the static ratchets back to CPU unit tests: 仓库正在系统化引入 ratchet 型测试: 该 PR 调整静态 ratchet 在 CI 的归属, 与本 PR 的暴露面 ratchet 注册进 CPU/CUDA 套件同属测试基建与配置一致性战线。