

PR #2711 完整报告

radixark/miles

test(e2e): verify multi-turn Anthropic sessions

合并时间: 2026-08-30 12:01

原文链接: <http://prhub.com.cn/radixark/miles/pull/2711>

执行摘要

- 一句话: 新增 Anthropic 会话验证与 SGLang 契约闸门
- 推荐动作: 值得精读。三个设计点最有价值: 一是硬 / 软失配的分层判定配合 sidecar 日志, 解决 "rollout 重试吞掉验证失败" 的经典问题; 二是用配对 checkout 把上游外部 PR 不跑的 CPU 套件搬进自家 CI, 形成行为保持闸门; 三是端口独立分配对并发部署的改进。阅读时注意临时闸门的清理时机与 Qwen3.6 阈值放宽的后续收敛。

功能与动机

PR body 明确动机: 六请求的 tool/result/text 历史、推理回放与受支持的中间 system 插入必须保持 TITO 前缀; Nemotron 与 GLM-4.7 的不完整响应必须在不推进无效历史的前提下重试。同时, SGLang PR #35127 重构了与运行时无关的 Anthropic 转换接缝, 但其外部 PR 策略不运行所属的 CPU 套件, 因此 Miles 需要一个临时配对 checkout 闸门来保证行为保持。

实现拆解

1. 新增 Anthropic 六轮验证代理

`miles/utils/test_utils/anthropic_session_verify_agent.py` (新增 386 行):

- `run_agent` 以 `httpx` 客户端驱动三轮工具 / 结果 / 文本循环, 工具轮 `tool_choice` 强制 `get_weather`、文本轮置为 `none`; 每轮经 `_post_complete` 校验响应体, 对不完整响应在同一父节点重试 (上限 8 次)。
- 通过 `/health` 的 `anthropic_intermediate_system_supported` 读取路由能力, 与 `fixed_template_append_roles` 的 `system` 能力取交集, 再与模型契约 `anthropic_intermediate_system_expectation` (`required/forbidden`) 比对; MiniMax 家族由 `_assert_supported_tito_model` 直接排除。
- `_assert_canonical_records` 校验 6 条记录、`input_ids` 前缀连续、`assistant` 序列化回放 (`loose_tool_call_message_matches`) 以及会话树中 `active` 路径 (允许被 `superseded` 的重试叶保留)。

2. 扩展通用验证代理

`miles/utils/test_utils/session_verify_agent.py`:

- 新增 `_chat_complete`: `finish_reason == "length"` 时重试, 重试 `seed` 递增 `_RETRY_SEED_STRIDE` (100 万), 保证采样结果不同; 初始与每步 `chat` 均改走该入口。

- 新增 `_journal_verifier_assertions` 装饰器与 `_append_session_verify_record`: 断言失败先写入 `MILES_SESSION_VERIFY_METRICS_PATH` sidecar 再抛出, 避免 rollout 循环把异常当作可重试样本失败吞掉。
- `_verify_tito_samples` 将硬失配 (`special_token_count` 等) 从 "立即 raise" 改为 "记录 + 由 run 级闸门聚合失败", 软失配 `assistant_text` 仅记录; 无 sidecar 环境仍立即 raise。

3. 运行器与 E2E 矩阵改造

`session_verify_runner.py` + `tests/e2e/sclang/test_session_server_multi_role/_common.py` :

- `run_session_verify` 增加 `wire_format` 参数 (`openai/anthropic`) ; `_session_verify_env` 对 `anthropic` 清空 CI history 记录目录避免串号; `assert_session_verify_metrics` 增加 `require_append_tool` 开关。
- `ModelConfig` 新增 `verify_anthropic`、`anthropic_intermediate_system_expectation`、`anthropic_assistant_text_threshold`; `run_both_versions` 从 2 趟变为 3 趟 (`v1 openai`、`v2 openai`、`v2 anthropic`) , `anthropic` 趟使用 `loose_tool_call` 消息匹配器与独立阈值。

4. SGLang 转换契约 CPU 桥

`tests/fast/router/test_sclang_anthropic_conversion.py` (新增) : 从 `SGLANG_SOURCE_ROOT` 环境变量定位配对 checkout (钉在 `sclang PR #35127` 的 `e95eae48`) , 以 `subprocess` 运行 `sclang` 自带的 `anthropic test_utils.py` 与 `test_serving.py` , 并注册进 `stage-a-cpu`; 缺失文件或子进程非零退出即失败。

5. 端口分配修复

`miles/ray/rollout/router_manager.py` + `miles/utils/arguments.py` :
`_resolve_session_server_ports` 在未显式指定 `--session-server-port` 时逐个 worker 用 `find_available_port` 独立分配, 不再要求连续段; 显式 `start` 仍锚定连续区间, 参数 `help` 文案同步更新。

6. 测试与文档配套

新增 `test_anthropic_session_verify_agent.py` (`FakeAsyncClient` 驱动 6 轮并校验 payload/ 树结构) 、 `test_sclang_anthropic_conversion.py`; 扩展 `test_session_verify_agent.py`、`test_session_verify_runner.py`、`test_router_manager.py`;
`docs/user-guide/cli-reference.md` 更新参数说明。

关键文件:

- `miles/utils/test_utils/anthropic_session_verify_agent.py` (模块 会话验证; 类别 `source`; 类型 `core-logic`; 符号 `_assert_supported_tito_model`, `_verify_intermediate_system`, `_assert_intermediate_system_expectation`, `_build_payload`) : 新增 Anthropic 六轮会话验证代理, 是本 PR 的核心源码: 驱动三轮 `tool_use/tool_result/text` 循环、中间 `system` 契约校验与 `canonical` 会话树断言。
- `miles/utils/test_utils/session_verify_agent.py` (模块 验证代理; 类别 `source`; 类型 `core-logic`; 符号 `_append_session_verify_record`, `_journal_verifier_assertions`, `_chat_complete`, `_verify_tito_samples`) : 通用验证代理新增截断重试 (`seed` 递增) 与断

言 /sidecar 日志机制，保证 rollout 重试不会吞掉验证失败证据。

- miles/utils/test_utils/session_verify_runner.py (模块 验证运行器；类别 source；类型 core-logic；符号 SessionWireFormat, run_session_verify, _session_verify_env, assert_session_verify_metrics)：运行器引入 wire_format 双通道、CI history 隔离与 run 级聚合闸门，是 Anthropic 通道接入 E2E 的编排层。
- miles/ray/rollout/router_manager.py (模块 路由调度；类别 source；类型 core-logic；符号 _resolve_session_server_ports)：生产代码变更：session server 端口自动分配从连续段改为逐端口独立分配，消除一个已占用端口拖垮整批启动的问题。
- tests/fast/router/test_sglang_anthropic_conversion.py (模块 转换桥接；类别 test；类型 test-coverage；符号 test_sglang_anthropic_conversion_contract)：SGLang Anthropic 转换契约的 CPU 配对 checkout 闸门，直接回应上游 sglang PR #35127 外部 PR 策略缺口。
- tests/e2e/sglang/test_session_server_multi_role/_common.py (模块 测试矩阵；类别 test；类型 test-coverage；符号 SessionEndpoint, ModelConfig, run_one, run_both_versions)：E2E 运行矩阵从 v1/v2 OpenAI 两趟扩展为 v1/v2 OpenAI + v2 Anthropic 三趟，并承载模型级 Anthropic 契约配置。
- tests/fast/utils/test_utils/test_anthropic_session_verify_agent.py (模块 会话验证；类别 test；类型 test-coverage；符号 _Response, _response_fixtures, _snapshot, test_run_agent_runs_six_turns_and_checks_canonical_records)：585 行新测试，用 FakeAsyncClient 完整模拟六轮请求 / 响应与会话树，锁定验证代理的 canonical 断言与重试语义。

关键符号：run_agent, _post_complete, _assert_canonical_records, _assert_anthropic_tool_response, _assert_anthropic_text_response, _verify_intermediate_system, _assert_intermediate_system_expectation, _chat_complete, _journal_verifier_assertions, _append_session_verify_record, _verify_tito_samples, run_session_verify, _session_verify_env, assert_session_verify_metrics, run_one, run_both_versions, _resolve_session_server_ports

关键源码片段

miles/utils/test_utils/session_verify_agent.py

通用验证代理新增截断重试 (seed 递增) 与断言 /sidecar 日志机制，保证 rollout 重试不会吞掉验证失败证据。

```
"""侧车日志与断言护栏 (session_verify_agent.py 新增核心单元) 。”””
```

```
def _append_session_verify_record(entry: dict) -> bool:
    # 将验证失败的证据写入 sidecar JSONL，避免 rollout 重试把失败吞掉
    metrics_path = os.environ.get("MILES_SESSION_VERIFY_METRICS_PATH")
    if not metrics_path:
        return False
    payload = (json.dumps(entry) + "\n").encode()
```

```

fd = os.open(
    metrics_path,
    os.O_WRONLY | os.O_APPEND | os.O_CREAT | getattr(os, "O_CLOEXEC", 0),
    0o600,
)
try:
    written = os.write(fd, payload)
    if written != len(payload):
        raise OSError(f"short metrics sidecar write: expected {len(payload)} bytes, wrote {written}")
finally:
    os.close(fd)
return True

```

```

def _journal_verifier_assertions(stage: str):
    # 装饰器：断言失败先落盘，再重新抛出，保证异常路径也可追溯
    def decorate(func):
        @wraps(func)
        async def wrapped(*args, **kwargs):
            try:
                return await func(*args, **kwargs)
            except AssertionError as exc:
                _append_session_verify_record(
                    {
                        "verification_error": {
                            "stage": stage,
                            "type": type(exc).__name__,
                            "message": str(exc)[:1000],
                        }
                    }
                )
            raise

        return wrapped

    return decorate

```

```

async def _chat_complete(client, base_url, messages, request_kwargs, *, label):
    # finish_reason == "length" 的截断响应在同一父节点重试；
    # 每次重试递增 seed，保证采样结果不同
    for retry in range(_MAX_INCOMPLETE_TURN_RETRIES + 1):
        attempt_kwargs = request_kwargs
        if retry and request_kwargs.get("seed") is not None:
            attempt_kwargs = {
                **request_kwargs,
                "seed": request_kwargs["seed"] + retry * _RETRY_SEED_STRIDE,
            }

```

```

response = await _chat(client, base_url, messages, attempt_kwargs, label=label)
if response["choices"][0].get("finish_reason") != "length":
    return response
raise AssertionError(f"{label} exhausted {_MAX_INCOMPLETE_TURN_RETRIES} retries after
finish_reason='length'")

```

miles/ray/rollout/router_manager.py

生产代码变更：session server 端口自动分配从连续段改为逐端口独立分配，消除一个已占用端口拖垮整批启动的问题。

```

def _resolve_session_server_ports(start: int | None, workers: int) -> list[int]:
    """显式 start 时返回连续端口段；未指定时每个 worker 端口独立自动分配。"""
    if workers < 1:
        raise ValueError("--session-server-workers must be at least 1.")
    # TODO(#1837): Refactor IP/port allocation; keep this naive for now.
    if start is None:
        # 旧逻辑要求整段连续端口，任意一个端口被占用就整体失败；
        # 新逻辑逐个 find_available_port，互不牵连
        search_start = random.randint(5000, 6000)
        ports = []
        while len(ports) < workers:
            port = find_available_port(search_start)
            if port not in ports:
                ports.append(port)
            search_start = port + 1
        return ports
    return list(range(start, start + workers))

```

评论区精华

本 PR 没有实质性的行内 review 评论（Shi-Dong 以空 body APPROVED，claude[bot] 仅发模板提示），讨论主要来自作者评论与 PR body 的 Review Focus：

- 作者明确设置合并门禁："Wait for sglang bump and rebase, then we can merge this PR."——需要先用 sglang bump 与 rebase 消除版本错位再合并。
- 合并前作者给出最终验证证据："stage-a-cpu (2) / run-cpu 选择并通过了 test_sglang_anthropic_conversion_contract (18.82s)。The bridge executes SGLang #35127's owned test_utils.py and test_serving.py; the direct paired-checkout run on miles-h200-dev-2 reported 84 passed, 55 subtests passed."
- PR body 明确分割了硬 / 软失配边界：硬失配 (special_token_count / special_token_type / non_assistant_text) 必须在 run 级为 0；assistant_text 为软阈值。Qwen3.6 的 Anthropic 转换会归一化分隔符，因此仅 Anthropic 通道阈值放宽到 1.0，OpenAI 阈值与所有硬闸门保持不变。
- 端口隔离设计：自动分配逐 worker 独立选端口，显式 --session-server-port 仍锚定连续区间。
- 合并前需等待 sglang bump 与 rebase (other): sglang bump 与 rebase 落地后合并（最终 head 为 b8f5667c），合并前验证通过。

- SGLang 配对 checkout 的 CPU 行为保持闸门 (design): 实现并验证: direct paired-checkout run 报告 84 passed, 55 subtests passed; stage-a-cpu 合约测试 18.82s 通过。
- Anthropic 验证器 hard vs soft mismatch 边界 (design): 硬失配改为 sidecar 记录 + 聚合闸门; OpenAI 阈值与硬闸门保持不变, 仅 Anthropic 通道放宽到 1.0。
- session server 端口分配行为变更 (design): 修改 _resolve_session_server_ports 为逐端口独立 find_available_port, 并新增 test_none_auto_allocates_each_port_independently。

风险与影响

- 风险:
 - SGLang 配对依赖为临时闸门: test_sglang_anthropic_conversion.py 钉在 sglang PR #35127 的 e95eae48, 若 SGLANG_SOURCE_ROOT 未设置或 sglang bump 后 pin 过期, 契约测试会失败或失去意义; 需要后续 PR 清理。
 - 硬失配改为聚合判定: _verify_tito_samples 在有 sidecar 时不再立即 raise, 失败判定后移到 assert_session_verify_metrics。若 metrics 文件在分布式 (多节点) rollout worker 间不可见 (当前验证基于 colocate 单机), 硬失配可能被静默漏掉; 无 sidecar 环境仍立即 raise, 有测试覆盖, 风险可控但需注意部署形态。
 - Qwen3.6 阈值放宽到 1.0: anthropic_assistant_text_threshold=1.0 相当于该通道下格式化差异不设软闸门, 可能掩盖真实回归, 依赖硬闸门兜底。
 - 端口分配竞态: _resolve_session_server_ports 逐个 find_available_port, 32 个 worker 时存在 TOCTOU 竞态与扫描开销; 显式 start 连续段语义保留, 但并发作业显式端口冲突仍是既有问题。
 - 多轮重试放大 GPU 时间: _post_complete 每轮最多重试 8 次、OpenAI 通道 _chat_complete 重试 2 次, 极端情况下 E2E 耗时会明显放大。
- 影响:
 - 用户 / 开发者: session 验证工具链新增 Anthropic 通道与模型级契约配置 (verify_anthropic、anthropic_intermediate_system_expectation、anthropic_assistant_text_threshold), 新增模型家族必须显式声明 Anthropic 契约, 否则 E2E 直接失败。
 - CI/ 测试矩阵: H200 GPU E2E 从每模型 2 趟变为 3 趟 (v1 openai、v2 openai、v2 anthropic), GPU 墙钟成本约增加 50%; stage-a CPU 新增一个约 19 秒的合约测试。
 - 路由 / 部署: session server 端口分配语义变化, 未显式指定端口的部署从 "连续段" 变为 "相互独立的端口", 依赖固定连续端口段的脚本需检查。
 - 团队协作: SGLang 上游转换逻辑的回归保护从上游侧移到了 Miles 侧, 短期内 Miles 需要维护配对 checkout 的 pin。
 - 风险标记: SGLang 临时配对依赖, 硬失配改为聚合判定, qwen3.6 宽松阈值, 端口分配竞态, 多节点 sidecar 可见性

关联脉络

- PR #2774 feat(session): configure session server workers explicitly: 本 PR 直接修改了 #2774 引入的 `_resolve_session_server_ports` 与 `--session-server-workers` 参数语义, 将 " 连续端口段 " 改为 " 独立自动分配 ", 属于同一功能线的延续。
- PR #2759 feat(tito): support Qwen3.5 and Qwen3.6 templates: 同一 E2E 目录 `tests/e2e/sglang/test_session_server_multi_role/` 下的 `test_qwen36.py` 在本 PR 中新增 Anthropic 中间 system 契约声明; TITO 模板的 `allowed_append_roles` 决定 Anthropic 中间 system 是否可插入。
- PR #2714 Bump sglang to v0.5.18: 作者明确要求 " 等 sglang bump 与 rebase 后再合并 "; 本 PR 的 SGLang 配对 checkout 闸门 (sglang PR #35127) 也依赖 sglang 版本对齐。
- PR #2604 fix(rollout): raise server readiness timeout to 120s: 同样修改 `miles/ray/rollout/router_manager.py` 的 session server 生命周期配置, 与本 PR 的端口分配改动同属 session server 部署可靠性演进。