

PR #24506 完整报告

sgl-project/sglang

Add prefixed _log helper in dumper

合并时间: 2026-05-06 16:54

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/24506>

执行摘要

- 一句话: 引入集中式 _log 函数, 统一 dumper 日志格式
- 推荐动作: 值得合并, 代码逻辑清晰且测试覆盖到位。建议后续跟进 review 中的多行前缀改进和删除 .gitignore 中无关变更。

功能与动机

dumper 模块原有日志通过 print 直接输出, 缺乏统一的格式标识, 导致多 rank 场景下日志难以区分来源和时间。通过引入集中式 _log 函数, 可以确保所有 dumper 日志带有 rank 和时间戳前缀, 提升调试可读性。

实现拆解

变更分为以下三步:

1. 新增 _log 函数 (python/sglang/srt/debug_utils/dumper.py: 新增 _log(msg) 函数, 格式为 [Dumper, rank={rank}, t={time:.3f}] {msg}, 并设置 flush=True 确保实时输出。
2. 替换所有 print 调用: 将 step()、apply_source_patches()、_dump_single()、_ensure_exp_name()、_torch_save()、_collective_with_timeout()、_cleanup_old_dumps() 中的 print 替换为对 _log 的调用, 并调整消息内容, 移除原手动添加的 [Dumper] 前缀和时间戳。
3. 更新测试 (test/registered/debug_utils/test_dumper.py): 新增 TestLog 类及其 test_log_format 方法, 验证 _log 输出的格式包含 [Dumper, rank= 和 , t=; 修改 test_silent_skip 中的断言以适应新的日志前缀格式。

关键文件:

- python/sglang/srt/debug_utils/dumper.py (模块 调试工具; 类别 source; 类型 core-logic; 符号 _log): 核心变更文件, 新增 _log 函数并替换所有 print 调用, 统一日志格式。
- test/registered/debug_utils/test_dumper.py (模块 测试; 类别 test; 类型 test-coverage; 符号 TestLog, test_log_format): 新增 TestLog 测试类验证 _log 格式, 并修复其他测试因格式变化导致的断言失败。

关键符号: _log

关键源码片段

python/sglang/srt/debug_utils/dumper.py

核心变更文件，新增 `_log` 函数并替换所有 `print` 调用，统一日志格式。

```
# 新增 _log 函数：带 rank 和时间戳的统一日志输出
# 使用 _get_rank() 动态获取当前进程的 rank，确保多 rank 日志可区分
def _log(msg: str) -> None:
    print(f"[Dumper, rank={_get_rank()}, t={time.time():.3f}] {msg}", flush=True)

# 替换示例：在 step() 中替换原来的 print(...)
def step(self):
    ...
    self._state.step += 1
    _log(f"step={self._state.step}")

# 在 _dump_single 中替换 print，移除多余前缀
# 原来：print(f"[{tag}] [{rank}, {time.time()}] {path} ...")
# 现在：_log(f"[{tag}] {path} ...")
# rank 和 t 由 _log 统一管理

def _torch_save(value, path: str):
    ...
    except Exception as e:
        # 原来：print(f"[Dumper] Observe error={e} ...")
        # 现在：_log(f"Observe error={e} ...")
        _log(f"Observe error={e} when saving data, skip the tensor")
```

test/registered/debug_utils/test_dumper.py

新增 `TestLog` 测试类验证 `_log` 格式，并修复其他测试因格式变化导致的断言失败。

```
# 新增 TestLog 类，验证 _log 输出包含 rank 和时间戳前缀
class TestLog:
    def test_log_format(self):
        with _capture_stdout() as captured:
            _log("hello")
        out = captured.getvalue()
        assert "hello" in out, out
        assert "[Dumper, rank=" in out, out # 确认包含 rank 标识
        assert ", t=" in out, out # 确认包含时间戳

# 已存在的测试 test_silent_skip 中，断言从 "[Dumper] Observe error="
# 改为 "[Dumper, rank=" in out and "Observe error=" in out
# 以匹配新的日志前缀格式
def test_silent_skip(self, tmp_path, capsys):
    ...
    captured = capsys.readouterr()
    assert "[Dumper, rank=" in captured.out
    assert "Observe error=" in captured.out
```

```
assert "skip the tensor" in captured.out
```

评论区精华

review 中 gemini-code-assist[bot] 提出了两点改进建议: `-.gitignore` 变更不合适: 添加 `.claude/scheduled_tasks.lock` 与 PR 初衷无关, 建议在个人全局 gitignore 中处理工具相关文件。

- `_log` 函数应支持多行消息: 当前实现只给第一行加前缀, 如果 msg 包含多行 (如 traceback), 后续行会缺少标识, 建议拆行逐行前缀。
 - `.gitignore` 中新增 `.claude/scheduled_tasks.lock` 不合适 (other): 未明确回复, 但该变更未被移除, 建议后续移除。
 - `_log` 函数不支持多行消息前缀 (design): 未采纳, 当前实现仍为单行前缀。

风险与影响

- 风险: 风险较低。变更仅限于字符串格式和日志输出方式, 未改动模块核心逻辑。但 `_log` 函数当前不支持多行消息, 若未来传递多行字符串, 日志可能混淆。此外, 若外部工具或脚本依赖原有 `print` 格式解析, 可能因格式变化而失效。
- 影响: 影响范围限于 `dumper` 模块内部和关联测试。所有 `dumper` 日志输出的一致性和可读性得到提升, 但用户若直接解析原始日志格式 (如 `[Dumper]`) 则需调整。测试覆盖率得到增强, 增加了 `_log` 格式的专项验证。
- 风险标记: 日志格式变更可能影响外部解析, `_log` 不支持多行消息, `.gitignore` 引入无关变更

关联脉络

- PR #24507 Support cross-system tensor grafting in dumper: 同一作者 fzyzcjy 的系列 PR 之一, 均涉及 `dumper` 模块增强, 本 PR 为新功能提供日志基础设施。
- PR #24508 Support t2b direction and overlap protection in dumper grafter: 继续扩展 `dumper grafter` 功能, 本 PR 的 `_log` 函数将被后续功能使用。
- PR #24509 Support user-supplied recv-side transform in dumper grafter: 同样属于 `dumper grafter` 系列, 共享 `_log` 函数。
- PR #24510 Support multi-rank exchange via `all_gather_object` in dumper grafter: 进一步扩展 `dumper` 分布式支持, 本 PR 的日志统一有助于调试多 rank 场景。
- PR #24511 Support per-call extras and dataclass transform input in dumper grafter: `dumper grafter` 功能增强系列, 本 PR 提供基础日志能力。
- PR #24512 Enhance diff and tensor-info logging in dumper grafter: 增强 `dumper` 日志, 与本 PR 的 `_log` 函数直接相关。
- PR #24513 Add e2e test with log snapshot in dumper grafter: 添加端到端测试, 可能依赖本 PR 的 `_log` 函数进行日志断言。