PR #25548 完整报告

sgl-project/sglang

Quiet test_bs_1_speed CI log

合并时间：2026-05-18 10:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25548

执行摘要

- 一句话：清理 speculative decoding CI 测试日志
- 推荐动作：该 PR 值得合并，属于低风险的质量改进。建议后续修复 print_output 对 stream 模式的支持，以完全实现静默功能。

功能与动机

PR 标题和描述明确提到“Reduce log noise in the spec-decoding bs_1_speed CI path”，核心动机是降低 CI 日志噪音，避免冗余日志干扰调试和监控。

实现拆解

1. 限制 flush cache 日志打印（scheduler.py）：在 flush_cache 方法中，将 logger.info 调用包裹在 if self.is_stats_logging_rank: 条件下，确保只有 DP 组内的统计 rank 打印该日志，避免多 rank 重复输出。
2. 增强 send_one_prompt 函数（send_one.py）：为 send_one_prompt 函数新增 label（可选字符串，用于在结果表格上方打印标签）和 print_output（布尔值，控制是否打印模型输出文本，默认 True）参数。当 print_output=False 时，即使非 stream 模式也不会打印输出文本。
3. 简化 spec_decoding_kit 调用（spec_decoding_kit.py）：将 test_bs_1_speed 中原本独立的 print(f'attempt {attempt}: ...') 调用替换为带 label 参数的 send_one_prompt 调用，并将 print_output 设为 False，从而移除冗余打印，同时保留 attempt 信息在表格标签中。

关键文件：

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic）：核心调度器中的缓存刷新日志限制为统计 rank，减少多 rank 日志噪音。
- python/sglang/test/send_one.py（模块 测试工具；类别 test；类型 test-coverage；符号 send_one_prompt）：send_one_prompt 函数新增 label 和 print_output 参数，为调用者提供输出控制能力。
- python/sglang/test/kits/spec_decoding_kit.py（模块 测试工具；类别 test；类型 test-coverage）：测试混入类中简化 attempt 打印，改为由 send_one_prompt 内部处理。

关键符号：send_one_prompt

关键源码片段

python/sglang/srt/managers/scheduler.py

核心调度器中的缓存刷新日志限制为统计 rank，减少多 rank 日志噪音。

```
# python/sglang/srt/managers/scheduler.py

def flush_cache(self, empty_cache: bool = True):
    """Flush memory pools and optionally empty device allocator cache."""
    if self.is_fully_idle():
        self.cur_batch = None
        self.last_batch = None
        self.tree_cache.reset()
        self.req_to_token_pool.clear()
        self.token_to_kv_pool_allocator.clear()
        self.grammar_manager.clear()
        self.reset_metrics()

        if self.draft_worker:
            self.draft_worker.clear_cache_pool()

        if empty_cache:
            current_platform.empty_cache()
        # 仅由每个 DP 组的统计 rank 打印日志，避免多 rank 重复输出
        if self.is_stats_logging_rank:
            logger.info("Cache flushed successfully!")
        success = True
    else:
        logging.warning(
            f"Cache not flushed because there are pending requests. "
            f"#queue-req: {len(self.waiting_queue)}, "
            f"#running-req: {len(self.running_batch.reqs)}"
        )
        success = False
    return success
```

python/sglang/test/send_one.py

send_one_prompt 函数新增 label 和 print_output 参数，为调用者提供输出控制能力。

```
# python/sglang/test/send_one.py

def send_one_prompt(
    args: BenchArgs,
    label: Optional[str] = None,
    print_output: bool = True,
):
    base_url = f"http://{args.host}:{args.port}"
    # ... 构造输入和发送请求 ...
    latency = ret["meta_info"]["e2e_latency"]
```

```
    speed = ret["meta_info"]["completion_tokens"] / latency
    tokens = ret["meta_info"]["completion_tokens"]

    # 仅在 print_output 为 True 且非 stream 时打印输出文本
    if not args.stream and print_output:
        print(ret["text"])

    print()
    if label is not None:
        print(label) # 在结果表格上方打印标签，例如 "attempt 2"
    headers = ["Latency (s)", "Tokens", "Acc Length", "Speed (token/s)"]
    rows = [[f"{latency:.3f}", f"{tokens}", f"{acc_length:.3f}", f"{speed:.2f}"]]
    msg = tabulate.tabulate(rows, headers=headers, tablefmt="pretty")
    print(msg)

    return acc_length, speed
```

python/sglang/test/kits/spec_decoding_kit.py

测试混入类中简化 attempt 打印，改为由 send_one_prompt 内部处理。

```
# python/sglang/test/kits/spec_decoding_kit.py

def test_bs_1_speed(self):
    args = BenchArgs(port=int(self.base_url.split(":")[-1]), max_new_tokens=2048)
    acc_length, speed = 0.0, 0.0
    for attempt in range(1, self.bs_1_speed_attempts + 1):
        requests.get(self.base_url + "/flush_cache")
        # 传入 label 和 print_output=False，由 send_one_prompt 打印表格时附带标签
        acc_length, speed = send_one_prompt(
            args, label=f"attempt {attempt}", print_output=False
        )
        if acc_length > self.accept_length_thres and speed > self.bs_1_speed_thres:
            break
    requests.get(self.base_url + "/flush_cache")
    # ... 断言和 CI 摘要写入 ...
```

评论区精华

gemini-code-assist[bot] 指出 print_output 参数在 stream 模式下被忽略，stream 循环仍会输出块到 stdout。这表示 print_output 功能不完整，但鉴于 spec_decoding_kit.py 调用时未设置 --stream，该问题在当前上下文中不影响使用。

- print_output 在 stream 模式下被忽略 (correctness): 当前 PR 未修复该问题，但 spec_decoding_kit.py 未使用 stream 模式，因此不影响本次变更。

风险与影响

- 风险：无显著风险。变更仅影响日志打印和测试辅助函数的行为，不涉及核心推理逻辑。但需注意 print_output 对 stream 模式支持不完整，未来若在 stream 模式下调用

send_one_prompt 且期望静默，可能导致意外输出。

- 影响：
  1. 对 CI 日志：显著减少 test_bs_1_speed 测试的日志量，尤其在多 DP 组配置下，每个测试仅打印一次缓存刷新日志。
  2. 对 send_one_prompt 用户：新增参数为其他测试或脚本提供了更灵活的输出控制。
  3. 对下游测试：无需更改，因新参数有默认值，保持向后兼容。 - 风险标记：print_output 对 stream 模式无效

关联脉络

- 暂无明显关联 PR