

PR #2027 完整报告

THUDM/slime

Only upload per sample stats to wandb

合并时间: 2026-06-08 19:26

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2027>

执行摘要

- 一句话: 将 wandb 指标上传改为仅基于 per-sample 统计
- 推荐动作: 建议仔细阅读 `_compute_sglang_request_perf_metrics` 和 `_iter_sglang_generate_attrs` 的实现, 理解从 trace 提取指标的模式。若当前使用服务器端 wandb 指标, 需评估移除影响并调整监控策略。关注 `sglang.patch` 的版本兼容性。

功能与动机

之前的方案通过 `reinit_wandb_primary_with_open_metrics` 在 wandb 中开启 stats monitor 抓取 `sglang` 的 Prometheus 端点, 会上传大量服务器级指标 (如 GPU 内存、KV cache 等), 数据量大且非 per-sample 级别。新方案从 `sglang` 生成的 trace 中提取每个请求的性能指标, 计算统计后上传, 更精准且减轻 wandb 负载。

实现拆解

1. 在 `slime/ray/rollout.py` 中新增 per-sample 指标收集: 定义 `_SGLANG_REQUEST_PERF_FIELDS`、`_SGLANG_PREFILL_PERF_FIELDS`、`_SGLANG_DECODE_PERF_FIELDS` 常量元组, 从 trace attributes 映射到 wandb 指标名称; 新增 `_iter_sglang_generate_attrs` 生成器提取每个样本的 attrs; 新增 `_compute_sglang_request_perf_metrics` 函数遍历所有样本, 调用内部函数 `add_value` 收集有效数值, 最后通过 `compute_statistics` 计算统计量并返回汇总词典。`compute_perf_metrics_from_samples` 末尾会调用新函数合并结果。
2. 移除旧服务器端指标采集: 删除 `slime/utils/wandb_utils.py` 中的 `reinit_wandb_primary_with_open_metrics` 函数 (约 56 行), 该函数通过重新初始化 wandb 并配置 `x_stats_open_metrics_endpoints` 抓取 `sglang` 路由器的 Prometheus 指标; 删除 `slime/utils/logging_utils.py` 中的 `update_tracking_open_metrics` 函数; 更新 `train.py` 和 `train_async.py`, 移除对应调用。
3. 更新 `sglang patch`: 在 `docker/patch/latest/sglang.patch` 中增加对 PD 分解时序的支持, 包括在 KV 传递链中新增 `prefill_timing` 辅助缓冲区、定义 `PREFILL_TIMING_DEST_ATTRS` 字段列表、`decode` 端超时检测; 同时修复字段命名不一致问题 (如 `fwd_bootstrap_duration` → `fwd_prefill_bootstrap_duration`)。
4. 完善 trace 字段映射: 在 `slime/utils/trace_utils.py` 的 `SGLANG_TRACE_META_KEYS` 中添加新字段 (如 `queue_time`, `e2e_latency`, `decode_throughput`, `pd_prefill_bootstrap_duration` 等), 并在 `build_sglang_meta_trace_attrs` 中增加

sglang_request_id。

5. 更新 timeline viewer: 在 tools/trace_timeline_viewer.py 中修正 PD lane spec 字段名、在图例和颜色常量中补充 decode 侧 bootstrap/alloc_wait 字段, 支持新分解视图。
6. 新增文档: 添加 docs/en/advanced/observability.md 和 docs/zh/advanced/observability.md, 介绍 wandb 可观测性配置。
7. Dockerfile 微调: 更新镜像版本或依赖。

关键文件:

- slime/ray/rollout.py (模块 rollout 层; 类别 source; 类型 core-logic; 符号 `_compute_sglang_request_perf_metrics`, `_iter_sglang_generate_attrs`, `add_value`): 核心变更文件: 新增 per-sample 性能指标收集和聚合函数, 改变 wandb 指标上传粒度
- slime/utils/wandb_utils.py (模块 W&B 工具; 类别 source; 类型 core-logic; 符号 `reinit_wandb_primary_with_open_metrics`): 移除通过 wandb `x_stats_open_metrics_endpoints` 抓取服务器端指标的函数, 是整个迁移的关键删除
- docker/patch/latest/sglang.patch (模块 SGLang 补丁; 类别 test; 类型 test-coverage): 增加 PD 分解时序字段支持, 修复字段命名, 影响 slang 底层补丁
- slime/utils/trace_utils.py (模块 Trace 工具; 类别 source; 类型 core-logic; 符号 `build_sglang_meta_trace_attrs`): 更新 trace 字段白名单和构建函数, 确保新字段被记录
- tools/trace_timeline_viewer.py (模块 可视化工具; 类别 source; 类型 core-logic): 调整 PD 分解字段映射、图例和颜色, 支持新时序字段的可视化
- train.py (模块 入口脚本; 类别 source; 类型 dependency-wiring): 移除 `update_tracking_open_metrics` 调用, 体现架构调整
- train_async.py (模块 异步入口; 类别 source; 类型 dependency-wiring): 与 train.py 同步移除旧调用
- slime/utils/logging_utils.py (模块 日志工具; 类别 source; 类型 core-logic; 符号 `update_tracking_open_metrics`): 删除 `update_tracking_open_metrics` 函数
- docs/en/advanced/observability.md (模块 英文文档; 类别 docs; 类型 documentation): 新增加英文可观测性配置文档
- docs/zh/advanced/observability.md (模块 中文文档; 类别 docs; 类型 documentation): 新增中文可观测性配置文档
- docker/Dockerfile (模块 Docker 配置; 类别 infra; 类型 infrastructure): 微调镜像版本

关键符号: `_compute_sglang_request_perf_metrics`, `_iter_sglang_generate_attrs`, `add_value`, `compute_perf_metrics_from_samples`, `build_sglang_meta_trace_attrs`, `reinit_wandb_primary_with_open_metrics` (removed), `update_tracking_open_metrics` (removed)

关键源码片段

slime/ray/rollout.py

核心变更文件: 新增 per-sample 性能指标收集和聚合函数, 改变 wandb 指标上传粒度

```
def _compute_sglang_request_perf_metrics(all_samples: list[Sample]):
```

```

# 从所有样本中收集 slang generate attrs, 提取性能指标并计算统计值
attrs_by_request = list(_iter_slang_generate_attrs(all_samples))
if not attrs_by_request:
    return {}

values_by_metric: dict[str, list[float]] = {}
profiled_request_count = 0

def add_value(metric_key: str, source_key: str, attrs: dict) -> bool:
    # 尝试从 attrs 中提取数值, 如果有效则加入列表
    value = attrs.get(source_key)
    if not isinstance(value, (int, float)) or isinstance(value, bool) or not np.isfinite(value):
        return False
    values_by_metric.setdefault(metric_key, []).append(float(value))
    return True

for attrs in attrs_by_request:
    request_has_perf = False
    # 依次处理请求级、prefill 级、decode 级指标
    for metric_key, source_key in _SGLANG_REQUEST_PERF_FIELDS:
        request_has_perf |= add_value(metric_key, source_key, attrs)
    for metric_key, source_key in _SGLANG_PREFILL_PERF_FIELDS:
        request_has_perf |= add_value(metric_key, source_key, attrs)
    for metric_key, source_key in _SGLANG_DECODE_PERF_FIELDS:
        request_has_perf |= add_value(metric_key, source_key, attrs)
    if request_has_perf:
        profiled_request_count += 1

metrics: dict[str, float] = {
    "request/count": len(attrs_by_request),
    "request/profiled_count": profiled_request_count,
}
# 对每个指标汇总统计量 (平均值、标准差等), 并扁平化命名
for key, values in values_by_metric.items():
    if not values:
        continue
    metrics |= dict_add_prefix(compute_statistics(values), f"{key}/")
    metrics[f"{key}/count"] = len(values)

return metrics

```

评论区精华

未发现正式 review 评论。从提交历史 (8 次提交, 含 revert 和 fix) 可见开发过程中有反复调试, 特别是在 PD 时序字段的命名与 slang patch 整合上经历了调整。

- 暂无高价值评论线程

风险与影响

- 风险:

1. 依赖 sglang trace 的生成; 若 sglang 版本不兼容或未正确启用 trace, per-sample 指标会缺失。
2. 移除 `reinit_wandb_primary_with_open_metrics` 后, 依赖服务器端指标 (如 GPU 内存、KV cache 使用率) 的用户将无法获取这些数据。
3. `sglang.patch` 需与 sglang 版本精确匹配, 升级 sglang 时可能需同步更新 patch。
4. 字段命名变更 (如 `pd_bootstrap_duration` → `pd_prefill_bootstrap_duration`) 可能影响已有监控仪表板或告警规则。
 - 影响: 对用户: wandb 仪表板不再显示服务器级指标, 转而展示 per-sample 性能指标 (请求延迟、吞吐量、PD 分解时序), 粒度更细但范围缩小。对系统: 减少 wandb 上传数据量, 降低网络和 wandb 服务压力。对团队: 需更新监控仪表板和告警规则; 调试时可通过增强的 trace 字段定位性能瓶颈。
 - 风险标记: 移除服务器端指标, 依赖 sglang trace, 字段名称变更, 补丁版本兼容性

关联脉络

- 暂无明显关联 PR