

PR #26850 完整报告

sgl-project/sglang

Add parallel-rank dump filenames and pipeline-global layer remapping to dumper

合并时间: 2026-06-08 14:49

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/26850>

执行摘要

- 一句话: 为调试转储器增加并行 rank 文件名和层重映射
- 推荐动作: 值得精读, 特别是插件式参数名重映射的设计模式, 可借鉴用于其他需要跨 rank 归一化的场景。建议作者回复 review 中提出的异常捕获建议或纳入后续 PR。

功能与动机

在并行环境下调试模型时, 多个 rank 的转储文件会写入共享目录, 仅靠 `step/rank/dump_index` 构建的文件名容易冲突; 同时 pipeline 并行下 `model.named_parameters()` 返回的层索引是局部值 (例如每个阶段都有 `layers.0`), 导致同名参数碰撞。因此需要为文件名添加rank标记, 并支持将参数名中的局部层索引映射为全局值。

实现拆解

1. 配置扩展: 在 `DumperConfig` 中添加 `include_parallel_rank_in_filename: bool = False` 字段, 并通过 `from_kv_pairs` 支持环境变量或 kv 方式配置。
2. 文件名标记收集: 新增 `_collect_parallel_rank_tags()` 函数, 遍历框架插件列表 (`_plugins`), 从每个插件的 `collect_parallel_info()` 中提取预定义的 rank 键集合 (`_PARALLEL_RANK_KEYS`), 遇到冲突时第一个插件优先。
3. 文件名嵌入: 在 `_dump_single` 方法内部, 当配置启用时, 调用上述函数获取标记字典并合并到 `full_kwargs` 中, 随后由 `_format_tags` 拼接到文件名中。
4. 参数名重映射插件钩子: 在基类 `_FrameworkPlugin` 中新增 `transform_model_param_name(self, model, param_name) -> Optional[str]` 方法, 默认返回 `None` (保持原样)。
5. `dump_model` 中应用钩子: 在 `Dumper.dump_model()` 的 `for param_name, param` 循环内, 逐一遍历每个插件并调用 `transform_model_param_name`, 将返回的非空值作为最终参数名。
6. Megatron 插件实现: `_MegatronPlugin.transform_model_param_name` 从 Megatron 的 `model_property_function` 获取 `get_transformer_layer_offset`, 计算全局偏移并正则替换参数名中的 `layers.N`。
7. 单元测试: 在 `test/registered/debug_utils/test_dumper.py` 中新增 `TestParallelRankInFilename` 类, 包含配置默认值、kv 解析、多插件合并、冲突优先级、空插件跳过、禁用 / 启用检查共 7 个测试; 以及 `TestTransformModelParamName` 类对重

命名功能进行测试。

关键文件：

- python/sglang/srt/debug_utils/dumper.py (模块 调试器；类别 source；类型 core-logic；符号 DumperConfig.include_parallel_rank_in_filename, _collect_parallel_rank_tags, _FrameworkPlugin.transform_model_param_name, _MegatronPlugin.transform_model_param_name)：核心变更文件：添加配置项、文件名标记函数、插件钩子定义及 Megatron 实现、dump_model 和 _dump_single 中的使用逻辑。
- test/registered/debug_utils/test_dumper.py (模块 测试；类别 test；类型 test-coverage；符号 TestParallelRankInFilename, TestTransformModelParamName)：新增测试类 TestParallelRankInFilename 和 TestTransformModelParamName，覆盖新功能的配置、收集、文件名生成和管道重命名行为。

关键符号：_collect_parallel_rank_tags, transform_model_param_name, _MegatronPlugin.transform_model_param_name, Dumper.dump_model, Dumper._dump_single

关键源码片段

python/sglang/srt/debug_utils/dumper.py

核心变更文件：添加配置项、文件名标记函数、插件钩子定义及 Megatron 实现、dump_model 和 _dump_single 中的使用逻辑。

```
# python/sglang/srt/debug_utils/dumper.py
```

```
# (1) DumperConfig 新增字段
```

```
@dataclass(frozen=True)
```

```
class DumperConfig(_BaseConfig):
```

```
    # ... 其他字段 ...
```

```
    # When True, append parallel-rank tags (pp_rank/tp_rank/...) to dump filenames so
```

```
    # tensors from different ranks do not collide when dumped into a shared directory.
```

```
    include_parallel_rank_in_filename: bool = False
```

```
# (2) _collect_parallel_rank_tags 收集 rank 标记
```

```
_PARALLEL_RANK_KEYS = ("pp_rank", "tp_rank", "cp_rank", "ep_rank", "etp_rank")
```

```
def _collect_parallel_rank_tags() -> dict[str, int]:
```

```
    """从各个插件收集并行 rank 信息，合并后返回。"""
```

```
    result: dict[str, int] = {}
```

```
    for plugin in _plugins:
```

```
        info = plugin.collect_parallel_info()
```

```
        if not info:
```

```
            continue
```

```
        for key in _PARALLEL_RANK_KEYS:
```

```
            if key in info and key not in result:
```

```
                result[key] = info[key]
```

```
    return result
```

(3) _dump_single 中使用 (当启用时)

```
def _dump_single(self, ...):
    full_kwargs = {
        "name": name,
        "dump_index": self._state.dump_index,
        **tags,
    }
    if self._config.include_parallel_rank_in_filename:
        full_kwargs.update(_collect_parallel_rank_tags()) # 添加 rank 标记
    full_filename = _format_tags(full_kwargs) + ".pt"
```

(4) dump_model 中应用参数名重命名钩子

```
def dump_model(self, model, name_prefix="param", ...):
    for param_name, param in model.named_parameters():
        for plugin in _plugins:
            # 每个插件都有机会改写参数名; 返回 None 则表示不修改
            param_name = plugin.transform_model_param_name(model, param_name) or param_name
        self._dump_inner(
            name=f"{name_prefix}__{param_name}",
            value=param,
            ...
        )
```

(5) 基类 _FrameworkPlugin 定义默认钩子

```
class _FrameworkPlugin:
    def transform_model_param_name(
        self, model: "torch.nn.Module", param_name: str
    ) -> Optional[str]:
        """子类重写此方法以转换参数名, 返回 None 表示不变。"""
        return None
```

(6) _MegatronPlugin 覆盖: 将局部层索引转换为全局索引

```
class _MegatronPlugin(_FrameworkPlugin):
    def transform_model_param_name(self, model, param_name):
        offset = self._get_transformer_layer_offset(model)
        if offset is None or offset == 0:
            return None # 无需修改
        # 使用正则替换 "layers.N" -> "layers.(N+offset)"
        import re
        return re.sub(r"layers\.\d+", lambda m: f"layers.{int(m.group(1)) + offset}", param_name)

    def _get_transformer_layer_offset(self, model):
        try:
            from megatron.core.transformer.transformer_layer import (
                get_transformer_layer_offset,
            )
            config = self._get_model_config(model)
            return get_transformer_layer_offset(config)
```

```
except (ImportError, AttributeError, AssertionError):  
    return None
```

评论区精华

Review 主要围绕异常健壮性: gemini-code-assist[bot] 建议在 `_collect_parallel_rank_tags()` 中调用 `plugin.collect_parallel_info()` 时包裹 `try-except Exception`, 因为分布式 API 可能在未初始化时抛出 `RuntimeError`; 同样在 `_MegatronPlugin._get_transformer_layer_offset` 和 `_get_model_config` 中建议扩大异常捕获范围。然而从最终提交看, 作者未采纳这些修改 (PR 已合并但相关代码仅捕获了 `(ImportError, AttributeError, AssertionError)`)。

- 异常捕获以提升健壮性 (correctness): 未采纳, 最终代码仅捕获 `(ImportError, AttributeError, AssertionError)`, 未使用宽泛异常。

风险与影响

- 风险: 异常处理不完整: `_collect_parallel_rank_tags()` 调用插件 `collect_parallel_info()` 未包裹大范围异常, 若某些插件在分布式环境未初始化时抛出非 `AttributeError` 之外的异常 (如 `RuntimeError`), 可能导致转储过程崩溃。配置默认安全: `include_parallel_rank_in_filename` 默认关闭, 文件名标记功能不会影响已有用户。参数名重命名: `dump_model` 中修改参数名, 可能影响依赖参数名格式的下游工具; 但仅作用于转储文件名, 不影响模型本身。无性能风险: 两种功能仅在转储时附加少量开销。
- 影响: 直接用户: 使用 SGLang 并行推理并需要调试的开发者, 可以通过启用配置避免文件冲突并获得全局一致的参数名。影响范围: 仅限于调试转储器模块, 不涉及推理路径, 默认不启用, 零影响。团队协作: 新增的插件钩子 `transform_model_param_name` 为其他框架 (如非 Megatron) 提供了扩展点, 便于未来支持更多并行策略。
- 风险标记: 异常处理不完善, 新配置项默认关闭, 参数名重命名影响下游工具

关联脉络

- PR #26874 Speed up dump comparator percentile computation using numpy: 同属 `debug_utils` 模块, 均为调试工具链改进。