PR #26937 完整报告

sgl-project/sglang

Add per-rank staggered weight loading for improved TP I/O concurrency

合并时间：2026-06-03 11:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26937

执行摘要

- 一句话：TP 权重加载排序与交错 I/O 优化
- 推荐动作：值得仔细阅读 loader.py 中交错逻辑的实现，并确认默认行为变更已广而告之。建议在 test/registered 中添加一个加载相关测试，覆盖 k=-1，0，1，2 等场景，确保回归捕获。

功能与动机

PR body 指出两个问题：

1) 内部迭代器里的 sorted() 调用覆盖了 SGLANG_SORT_WEIGHT_FILES 的设置，且仅有 safetensors 和 multi-thread 两条路径有排序，其他路径不一致； 2) TP>1 时所有 rank 以完全相同顺序读取文件，无法利用并发 I/O 优势。通过统一排序控制和使用交错排序来提升多 rank 的 I/O 并发度。

实现拆解

1. 环境变量升级（environ.py）：将 SGLANG_SORT_WEIGHT_FILES 从 EnvBool(False) 改为 EnvInt(0)，新增三种语义：-1（不排序）、0（仅排序，默认）、k>0（排序并按因子 k 交错）。
2. 集中排序控制（loader.py）：在 _prepare_weights() 中移除旧的 if envs.SGLANG_SORT_WEIGHT_FILES.get(): hf_weights_files.sort()，替换为根据 k 值和 TP 信息的集中排序 + 交错逻辑。当 k>0 时，将文件列表按 (tp_size * k) 分组，每个 rank 在其组内循环偏移 (tp_rank * k)，达成交错。
3. 移除迭代器内硬编码排序（weight_utils.py）：从 safetensors_weights_iterator 和 buffered_multi_thread_safetensors_weights_iterator 中删除 sorted_files = sorted(hf_weights_files)，改用传入的 hf_weights_files（此时已由 loader 排序 / 交错）。仅保留 _prefetch_all_checkpoints 内部对 sorted(hf_weights_files) 的调用，确保预取顺序一致性。
4. 对齐文本编码器加载（text_encoder_loader.py）：将 text_encoder_loader 中 _prepare_weights 的条件从 if envs.SGLANG_SORT_WEIGHT_FILES.get() 改为 if envs.SGLANG_SORT_WEIGHT_FILES.get() >= 0，在支持排序的同时避免不必要的交错（因为文本编码器无 TP 拆分）。

关键文件：

- python/sglang/srt/model_loader/loader.py（模块 模型加载；类别 source；类型 data-contract；符号 _prepare_weights）：核心修改：将排序和交错逻辑集中到 _prepare_weights，取代旧的单行 sort 调用；新增基于 TP 大小的分组交错算法。
- python/sglang/srt/model_loader/weight_utils.py（模块 模型加载；类别 source；类型 data-contract；符号 safetensors_weights_iterator, buffered_multi_thread_safetensors_weights_iterator）：删除迭代器内硬编码的 sorted()，改为直接使用传入的文件列表；预取时仍保留 sorted() 以保证一致。
- python/sglang/srt/environ.py（模块 配置；类别 source；类型 core-logic；符号 SGLANG_SORT_WEIGHT_FILES）：定义新的整型环境变量，注释说明各取值语义（-1/0/k>0），是功能配置的入口。
- python/sglang/multimodal_gen/runtime/loader/component_loaders/text_encoder_loader.py（模块 文本编码器；类别 source；类型 core-logic；符号 _prepare_weights）：对齐加载逻辑，仅在排序时使用 >=0 检查，并显式说明不应用交错（无 TP）。

关键符号：_prepare_weights, safetensors_weights_iterator, buffered_multi_thread_safetensors_weights_iterator

关键源码片段

python/sglang/srt/model_loader/loader.py

核心修改：将排序和交错逻辑集中到 _prepare_weights，取代旧的单行 sort 调用；新增基于 TP 大小的分组交错算法。

```
def _prepare_weights(self, source, revision, fall_back_to_pt):
    # ... 过滤、去重等逻辑 ...
    # Sort and optionally stagger weight files (see SGLANG_SORT_WEIGHT_FILES).
    # k=-1: no sort; k=0: sort only; k>0: sort + stagger by (tp_rank * k).
    k = envs.SGLANG_SORT_WEIGHT_FILES.get()
    if k >= 0:
        hf_weights_files.sort()
        if k > 0:
            tp_size = get_tensor_model_parallel_world_size()
            if tp_size > 1:
                tp_rank = get_tensor_model_parallel_rank()
                group_size = tp_size * k
                staggered: List[str] = []
                for i in range(0, len(hf_weights_files), group_size):
                    group = hf_weights_files[i : i + group_size]
                    n = len(group)
                    # 每个 rank 在组内循环偏移 (tp_rank * k) 个位置
                    staggered.extend(group[(j + tp_rank * k) % n] for j in range(n))
                hf_weights_files = staggered
    return hf_folder, hf_weights_files, use_safetensors
```

python/sglang/srt/model_loader/weight_utils.py

删除迭代器内硬编码的 sorted()，改为直接使用传入的文件列表；预取时仍保留 sorted() 以保证一致。

```
def safetensors_weights_iterator(
    hf_weights_files: List[str],
    disable_mmap: bool = False,
    prefetch: bool = False,
    prefetch_num_threads: int = 4,
    drop_cache_after_load: bool = False,
) -> Generator[Tuple[str, torch.Tensor], None, None]:
    """Iterate over the weights in the model safetensor files."""
    enable_tqdm = (
        not torch.distributed.is_initialized() or torch.distributed.get_rank() == 0
    )

    # 预取时仍然使用 sorted() 以确保页面缓存被按序访问
    if prefetch and not disable_mmap:
        _prefetch_all_checkpoints(
            sorted(hf_weights_files), num_threads=prefetch_num_threads
        )

    for st_file in tqdm(
        hf_weights_files, # 直接使用已排序 / 交错的列表
        desc="Loading safetensors checkpoint shards",
        disable=not enable_tqdm,
        bar_format=BAR_FORMAT,
        position=tqdm._get_free_pos(),
    ):
        # ... 加载逻辑 ...
```

python/sglang/srt/environ.py

定义新的整型环境变量，注释说明各取值语义（-1/0/k>0），是功能配置的入口。

```
class Envs:
    # fmt: off
    # Model & File Download
    SGLANG_USE_MODELSCOPE = EnvBool(False)
    # Controls weight-file ordering for load-time I/O optimization.
    # -1 : no sorting, no staggering; preserves original file order.
    # 0 : sort files only; maximizes ordering but may reduce cross-rank I/O concurrency.
    # k>0: sort files and stagger per-rank order with factor k.
    # Files are processed in groups of (tp_size * k), and rank r starts each
    # group at offset (r * k), improving multi-rank I/O concurrency while
    # keeping access relatively ordered.
    SGLANG_SORT_WEIGHT_FILES = EnvInt(0)
```

评论区精华

无实质性 review 讨论。ShangmingCai 审核并批准，仅提交了两次 lint 修复提交。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 回归风险：默认行为从 EnvBool(False)（不排序）变为 EnvInt(0)（排序），可能影响依赖原始文件顺序的旧部署。PR 中说明这是有意的，但需要在发布说明中强调。
  2. 兼容性：环境变量类型从布尔型变为整型，若存在旧配置直接写 SGLANG_SORT_WEIGHT_FILES=true 会导致解析失败。但 EnvInt 的 parse 会处理常见表达式（如 True 被转换为 1），风险可控。
  3. 性能回归：在缓存充足场景下，仅排序模式的加载时间比不排序差（155s vs 56s），但交错模式有助于缓解此问题。 - 影响：用户角度：TP 模型部署的启动加载时间可改善，特别是共享文件系统场景下；用户可通过设置 SGLANG_SORT_WEIGHT_FILES=1 开启交错获得加速。系统角度：减少了权重加载阶段的文件顺序耦合，使多 rank 的 I/O 分布更均匀，有利于扩展到大 TP 规模。团队角度：统一的排序控制点降低了后续增加新加载路径时的认知负担，并在 text_encoder_loader 中展示了可复用的模式。 - 风险标记：默认行为变更，缺少测试覆盖，缓存充足场景性能回退

关联脉络

- PR #27084 [diffusion] Optimize Cosmos3 i2v latent prep: 同为扩散模型性能优化方向，体现持续对加载和预处理阶段的 efficiency 关注。
- PR #27086 [diffusion] Clamp WanVAE decode output in place: 同为扩散模型性能优化，减少冗余分配。
- PR #26970 [perf] Replicate embed_tokens to drop the post-embed all-reduce: 同为优化 TP 场景下的性能，但侧重点不同（减少通信 vs 提升 I/O 并发）。