PR #31165 完整报告

sgl-project/sglang

Drop ModelRunner's duplicated parallel-degree fields and read them via self.ps

合并时间：2026-07-14 16:02

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31165

执行摘要

- 一句话：删除 ModelRunner 中重复的并行度字段，统一通过 ps 对象访问
- 推荐动作：建议阅读以理解并行状态访问的新规范，特别是新贡献者应注意所有并行度信息应从 ps 对象获取，而非 ModelRunner 属性。另外关注 frozen_kv_mtp_worker_v2.py 中 pp_size 未置 1 的潜在问题，如有条件建议后续修复。

功能与动机

ModelRunner kept self.tp_rank/tp_size/pp_/dp_/moe_/attn_cp_ alongside the ParallelState in self.ps. Remove the duplicates: build self.ps directly from the constructor args and read every parallel rank/size through self.ps. (no property shims).

实现拆解

1. ModelRunner.__init__精简：移除 12 个冗余字段（tp_rank、tp_size、pp_rank、pp_size、dp_rank、attn_dp_size、moe_ep_rank、moe_ep_size、attn_cp_rank、attn_cp_size、moe_dp_rank、moe_dp_size），这些值原封不动来自 self.ps，后续代码全部改为 self.ps.xxx。
2. 构造函数内引用更新：在 init_weight_updater、init_weight_exporter、init_remote_instance_weight_transporter、initialize 等方法中，将原 self.tp_rank 等替换为 self.ps.tp_rank 等。
3. 外部消费者适配：bootstrap.py 的 init_torch_distributed 函数内去除解包局部变量的代码，直接使用 ps.xxx；weight_checker.py 的 _parallelism_info 方法从 self._model_runner.ps 读取；base_runner.py、flashinfer_autotune.py 等文件中不再使用 mr.tp_size 而是 mr.ps.tp_size。
4. FrozenKVMTP Worker 改造：两个 Worker 类的构造函数从接收多个 int 参数改为接收完整的 ps 对象，并利用 dataclasses.replace(ps, pp_rank=0) 设置 spec worker 的 pp_rank 为 0。
5. 其他文件：attention backend、eplb manager、expert_backup_client、EAGLE CUDA graph runner 等约 20 个文件中的引用被更新。

关键文件：

- python/sglang/srt/model_executor/model_runner.py（模块 模型执行器；类别 source；类型 data-contract；符号 init, init_weight_updater, init_weight_exporter, initialize）：

核心变更文件：移除 12 个冗余并行度字段，统一通过 self.ps 访问，是本次重构的主入口。

- python/sglang/srt/distributed/bootstrap.py（模块 分布式初始化；类别 source；类型 core-logic；符号 init_torch_distributed）：分布式初始化函数 init_torch_distributed 中去除了局部变量解包，直接使用 ps 属性，是外部适配的关键点。
- python/sglang/srt/speculative/frozen_kv_mtp_worker_v2.py（模块 推测解码；类别 source；类型 dependency-wiring；符号 FrozenKVMTPDraftWorker.init, FrozenKVMTPVerifyWorker.init）：spec worker 构造函数从接收多个 int 改为接收完整的 ps 对象，是依赖契约变更的重点文件，且 Review 对其有专门建议。
- python/sglang/srt/utils/weight_checker.py（模块 权重校验；类别 source；类型 core-logic；符号 _parallelism_info）：_parallelism_info 方法从 mr.xxx 改为 mr.ps.xxx 获取并行度信息，是消费者适配的示例。
- python/sglang/srt/model_executor/runner/base_runner.py（模块 运行器；类别 source；类型 data-contract；符号 warmup, _dummy_run）：引用了 mr.pp_size 和 mr.attn_cp_size 等属性，改为 mr.ps.xxx，是基础运行器的适配。

关键符号：ModelRunner.init, init_weight_updater, init_torch_distributed, WeightChecker._parallelism_info, FrozenKVMTPDraftWorker.init

关键源码片段

python/sglang/srt/distributed/bootstrap.py

分布式初始化函数 init_torch_distributed 中去除了局部变量解包，直接使用 ps 属性，是外部适配的关键点。

```
def init_torch_distributed(
    *,
    server_args: ServerArgs,
    model_config: ModelConfig,
    device: str,
    ps: ParallelState, # 不再解包为局部变量，直接使用 ps 属性
    dist_port: int,
    is_draft_worker: bool,
    local_omp_cpuid: Optional[List[int]],
):
    tic = time.perf_counter()
    logger.info("Init torch distributed begin.")
    try:
        torch.get_device_module(device).set_device(ps.gpu_id)
    except Exception:
        logger.warning(
            f"Context: {device=} {ps.gpu_id=} ... {ps.tp_rank=} {ps.tp_size=}"
        )
        raise
    # ... 后续所有引用从 ps 获取，不再有局部变量
```

评论区精华

gemini-code-assist[bot] 在 frozen_kv_mtp_worker_v2.py 中建议：除了设置 pp_rank=0，还应设置 pp_size=1，否则当 PP 启用时 self.ps.pp_size > 1 为 True，会触发不兼容的 PP 断言和初始化逻辑。作者未采纳此建议，PR 已合并。

- Speculative worker pp_size should be set to 1 (correctness): 作者未采纳该建议，PR 已合并；后续可能需观察是否引发行错误。

风险与影响

- 风险：主要风险在于 spec worker（FrozenKVMTPDraftWorker、FrozenKVMTPVerifyWorker）的 pp_size 未被显式置 1。若 PP 启用时 spec worker 仍会被构建，可能引发 assert self.support_pp 失败。但当前代码中 model_runner.__init__内 if self.ps.pp_size > 1 处会根据 ps.pp_size 判断，而 spec worker 的 ps 对象中 pp_size 仍为原始值（>1），因此该断言可能触发。不过作者认为 spec worker 不会真正执行到该代码路径。此外，移除大量字段可能影响使用反射或直接属性引用的第三方模块。
- 影响：影响范围：约 20 个文件，涉及模型运行器、分布式初始化、推测解码、权重校验、注意力后端等模块。行为无变化，所有并行度读取改由统一来源（ps），数据一致性更强。对用户透明，无 API 变更。对团队维护有利：避免双重维护，减少不一致风险。
- 风险标记：spec worker pp_size 未对齐，并行度字段移除可能破坏反射依赖

关联脉络

- PR #31169 Split initialize() into orchestration helpers: 同一系列重构，拆分 ModelRunner 的逻辑，涉及同一文件的进一步模块化。
- PR #31168 Extract cuda-graph setup into a module: 同一系列重构，从 ModelRunner 中提取 CUDA graph 设置模块，与本 PR 共同推进 ModelRunner 瘦身。
- PR #31167 Extract attention-backend setup into a module: 同一系列重构，从 ModelRunner 中提取注意力后端设置，与本 PR 的字段移除是并行工作。
- PR #31166 Narrow component dependencies to injected fields instead of ModelRunner: 同一系列重构，解耦组件对 ModelRunner 的依赖，与本 PR 的字段移除配合减少耦合。