

PR #36621 完整报告

sgl-project/sglang

config: a parallel size has one spelling; a patched scope declares its own

合并时间: 2026-08-28 03:56

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36621>

执行摘要

- 一句话: 统一并行尺寸读取, 删除 `get_parallel().config` 双拼写
- 推荐动作: 值得精读。本 PR 的设计论证 (为什么一个拼写足够、哪些地方必须问 communicator 本身) 是配置 API 收敛的范例; `patch_tensor_parallel_group` 用声明式 `override` 替代隐式全局替换的做法也值得借鉴。阅读时重点关注 PR body 中的构造审计表、`moe_dp_size` 七个站点逐一分析, 以及 Codex 指出的 `kimi_k25` IPC 计数风险。

功能与动机

PR body 明确指出: After #36250, `get_parallel().config` served only five names — the ones that were both a live `ParallelContext` property and a `parallel` config leaf: `tp_size`, `pp_size`, `attn_cp_size`, `dcp_size`, `moe_dp_size`. Bare meant "ask the process group", `.config` meant "what the launch asked for", and every reader had to know which one it wanted. This deletes the distinction. 也就是说, 双拼写让每个调用点都必须做一次语义判断, 且容易在启动前 / 进程组未建时踩到 'Distributed environment is not initialized'。由于进程组就是按配置宽度构建的, 两种拼写本应永远相等 (除两个特例), 因此删除区分是安全的。

实现拆解

1. 收敛核心 API: 在 `python/sglang/srt/runtime_context.py` 中, `ParallelContext` 删除 `config` 属性和五个 live 尺寸属性, 新增 `__getattr__` 统一回答所有 `parallel` bag 叶子 (含 `nccl_port` 等纯配置叶子); 未发布时抛 `ValueError("parallel" not published)`, 未知名字仍是 `AttributeError`。
2. 声明式覆盖特殊场景: 在 `python/sglang/srt/distributed/parallel_state.py` 中, `patch_tensor_parallel_group` 现在用 `get_parallel().override(tp_size=..., tp_rank=..., tp_group=...)` 在作用域内声明被替换的成员, 作用域外读到的是启动配置; `get_moe_cp_size()` 改为返回 `_get_moe_dp_group().world_size`, 因为 `_MOE_DP = _ATTN_CP` 别名分支下 live 宽度与配置不同。
3. 全量调用点迁移: 将 99 处 `.config.<size>` 读取折叠为 bare 名, 涉及 `scheduler.py`、`layers/cp/base.py`、`entrypoints/engine.py`、`ray/engine.py`、`ray/data_parallel_controller.py` 等; 7 个 `moe_dp_size` 站点逐一审查, 其中 `model_loader/loader.py`、`ipc_loader.py`、`dumper.py` 改用 `get_moe_cp_size()` 保持 live 语义, `qwen3_moe.py`、`mellum.py` 的断言在 `moe_dp_size == 1` 时仍成立。

4. 配套签名与测试调整: `compute_world_size(config)` 改为 `compute_world_size(*, enable_dp_attention, dp_size, tp_size, pp_size)`, 适配 Ray driver 在无进程组时调用; 两个 scheduler 测试从 `mock get_parallel().config` 迁移到 `override_server_args` 发布上下文, 并改用 `compute_world_size(**shape)`。
5. 测试与文档同步: 删除 419 行的 `test_launch_path_reads_configured_sizes.py` (针对旧双拼写的 AST 普查), 精简 `test_global_config_read_ratchet.py` 中的 `_CONFIGURED_SIZE_CALL_SITES` 注册表; 重写 `sglang-runtime-context` skill 中关于已删除 API 的文档。

关键文件:

- `python/sglang/srt/runtime_context.py` (模块 运行时上下文; 类别 source; 类型 core-logic; 符号 `config`, `tp_size`, `pp_size`, `moe_dp_size`): 核心变更文件: `ParallelContext` 从 'live 属性 + config 跳转' 收敛为单一命名空间, `__getattr__` 统一回答 `parallel` 叶子, 删除 `config` 属性与五个 live 尺寸属性。
- `python/sglang/srt/distributed/parallel_state.py` (模块 并行状态; 类别 source; 类型 core-logic; 符号 `patch_tensor_parallel_group`, `get_moe_cp_size`): `patch_tensor_parallel_group` 改为声明式 `override tp_size/tp_rank/tp_group`; `get_moe_cp_size` 改为直接询问 MoE 数据并行组。
- `python/sglang/srt/managers/scheduler.py` (模块 调度器; 类别 source; 类型 core-logic; 符号 `get_internal_state`, `dispatch_event_loop`, `configure_scheduler_process`): 主要消费者: `ParallelState` 构造、`compute_world_size` 调用、`dispatch_event_loop` 分支等大量 `.config`。读取收敛为 bare 名。
- `python/sglang/srt/server_args.py` (模块 服务参数; 类别 source; 类型 core-logic; 符号 `compute_world_size`): `compute_world_size` 签名从 `config` 对象改为四个关键字参数, 适配无进程组时的启动路径。
- `python/sglang/srt/models/kimi_k25.py` (模块 模型实现; 类别 source; 类型 data-contract; 符号 `materialize_item_features`): review 指出的潜在风险点: IPC consumer count 在 `patch_tensor_parallel_group` 作用域内会读到覆盖后的子组宽度。
- `test/registered/unit/test_launch_path_reads_configured_sizes.py` (模块 启动路径; 类别 test; 类型 deletion; 符号 `_live_shadowed`, `_multiprocessing_names`, `_spawns_from_a_size`, `_parallel_bag_names`): 该 AST 普查测试针对旧双拼写 API (live 属性 vs config 叶子), 删除后由新 `__getattr__` 契约测试替代。
- `test/registered/unit/test_global_config_read_ratchet.py` (模块 配置守卫; 类别 test; 类型 test-coverage; 符号 `_live_shadowed_sizes`, `_parallel_config_reads`, `TestParallelConfigReadSpellings`): 删除 `_CONFIGURED_SIZE_CALL_SITES` 注册表与对应测试, 因为 `.config`。读取已全部折叠为 bare 名。
- `test/registered/unit/managers/test_scheduler_internal_state_world_size.py` (模块 调度器测试; 类别 test; 类型 test-coverage; 符号 `_make_parallel_config`, `_shape`, `_get_internal_state`): 从 `mock get_parallel().config` 迁移到 `override_server_args` 发布上下文, 验证 `compute_world_size` 新签名。

关键符号: `ParallelContext.getattr`, `get_moe_cp_size`, `patch_tensor_parallel_group`, `compute_world_size`, `Scheduler.get_internal_state`, `Scheduler.dispatch_event_loop`,

materialize_item_features

关键源码片段

python/sglang/srt/runtime_context.py

核心变更文件：ParallelContext 从 'live 属性 + config 跳转 ' 收敛为单一命名空间，__getattr__ 统一回答 parallel 叶子，删除 config 属性与五个 live 尺寸属性。

```
class ParallelContext:
```

```
    """并行拓扑命名空间：一个名字只有一种拼写。
```

排名 (rank) 与进程组句柄通过 @property 直读 canonical getter，回答 "当前活跃的进程组"；其余名字 (tp_size 及同类尺寸叶子，以及 nccl_port 这类仅配置存在的叶子) 一律从已发布的 parallel bag 回答，任何进程在发布后任何时刻都可读。尺寸读配置，是因为进程组就是按配置宽度构建的。

两个例外必须问进程组本身：initialize_model_parallel 在 attn_cp_size > moe_dp_size 时把 _MOE_DP 别名到 _ATTN_CP，所以想读 MoE communicator 宽度要调用 get_moe_cp_size()；patch_tensor_parallel_group 在更小的 TP 组下运行，它通过 override 在作用域内声明 tp_size / tp_rank / tp_group。

```
    __slots__ = ("_overrides", "_config")
```

```
    def __init__(self):
```

```
        self._overrides = {}
```

```
        self._config = None # parallel 配置 bag，发布时挂接
```

```
    def __getattr__(self, name):
```

```
        # 下划线开头是内部成员，直接交给默认的 AttributeError 路径
```

```
        if name.startswith("_"):
```

```
            raise AttributeError(f"{type(self).__name__!r} has no {name!r}")
```

```
        # 尚未发布时给出命名空间的错误信息，便于定位
```

```
        if self._config is None:
```

```
            raise ValueError("'parallel' not published")
```

```
        try:
```

```
            # 叶子从已发布的配置 bag 读取，与 resolution 决策一致
```

```
            return getattr(self._config, name)
```

```
        except AttributeError:
```

```
            # 未知名字仍然是 AttributeError，保持 Python 惯例
```

```
            raise AttributeError(f"{type(self).__name__!r} has no {name!r}")
```

python/sglang/srt/distributed/parallel_state.py

patch_tensor_parallel_group 改为声明式 override tp_size/tp_rank/tp_group；get_moe_cp_size 改为直接询问 MoE 数据并行组。

```
def patch_tensor_parallel_group(tp_group):
```

```
    # 在另一个 TP 组下运行一个作用域：draft 工作进程以目标模型的
```

```

# attention-TP 宽度运行 draft 模型。
# 现在通过 override 声明三个会变的成员，作用域内读到的是修补后的
# 宽度，作用域外仍读启动配置；draft_tp_context 因此回到一行。
_TP = tp_group
try:
    with get_parallel().override(
        tp_size=tp_group.world_size,
        tp_rank=tp_group.rank_in_group,
        tp_group=tp_group,
    ):
        yield
finally:
    yield # 生成器收尾，确保作用域正常退出

```

```

def get_moe_cp_size():
    # 返回 MoE 数据并行组的世界大小。
    # 这是读者真正会共同操作的 communicator 的宽度，
    # 与 _MOE_DP = _ATTN_CP 别名分支下的配置值不同。
    return _get_moe_dp_group().world_size

```

评论区精华

唯一的 review 讨论来自 Codex 自动审查 (P2)：在 `python/sglang/srt/models/kimi_k25.py:736`, `ipc_consumer_count = max(get_parallel().tp_size, 1)` 由 `get_parallel().config.tp_size` 改写而来，但该代码运行在 `patch_tensor_parallel_group` 作用域内，会读到被覆盖的 `draft/attention` 子组宽度，而不是 `tokenizer pool` 的完整配置消费者数。Codex 指出在 `encoder-DP` 模式下只有被选中的 `rank` 重建图像，其余确认槽位未设置，最终可能阻止有界 `IPC leases` 被回收。评论没有后续回复，PR 已合并，该问题可能留待后续修复。

- `patch_tensor_parallel_group` 覆盖 `tp_size` 后 `IPC` 确认计数失真 (`correctness`): 评论无后续回复，PR 已合并，问题可能留待后续修复。

风险与影响

- 风险:
 1. `IPC` 租约泄漏风险: `models/kimi_k25.py` 的 `ipc_consumer_count` 在 `patch_tensor_parallel_group` 作用域内会读子组宽度，`encoder-DP` 下可能导致确认槽位缺失、`IPC` 租约无法回收。
 2. 配置语义变化: 七个 `moe_dp_size` 读取点中，`loader.py`、`ipc_loader.py`、`dumper.py` 已改用 `live` 的 `get_moe_cp_size()`，但 `qwen3_moe.py`、`mellum.py` 的断言在 `moe_dp_size != 1` 且 `attn_cp_size > moe_dp_size` 时会比较真实值（虽然 `resolution` 已断言 `moe_dp_size == 1` 兜底）。
 3. 测试保护减少: 删除了 `test_launch_path_reads_configured_sizes.py` 和 `_CONFIGURED_SIZE_CALL_SITES` 注册表，对启动路径读取配置尺寸的 `AST` 级回归保

护消失，若未来重新引入 live 读取将无自动告警。

4. 广泛迁移的遗漏风险：99 处 .config 读取折叠为 bare 名，任何一处若本意是 live 值（非 `_MOE_DP = _ATTN_CP` 分支）会静默改变语义。 - 影响：影响面覆盖 48 个生产文件和 14 个测试文件，涉及调度器、并行状态、模型加载、Ray 启动、DP 控制器、多模态 gen 的并行状态等核心模块。所有读取并行尺寸的调用点（约 330 处）语义统一，消除了 'bare vs config' 的认知负担；对用户无直接可见变化，对开发者是 API 简化，但要警惕 `patch_tensor_parallel_group` 作用域内的语义覆盖。团队后续维护时需要记住：尺寸默认读配置，只有 `get_moe_cp_size()` 等显式 live 访问才问进程组。 - 风险标记：核心路径变更，跨模块 API 重构，潜在 IPC 租约泄漏，测试保护删除，配置语义变化

关联脉络

- PR #36622 config: the record is not an object that gets passed around: 同一五 PR 系列 (gc-p5) 的下一步，基于本 PR 之上，继续清理配置对象传递。
- PR #36676 Refactor server_args constants and layout: 同一 server_args/ 配置读取重构方向，移动常量与布局。
- PR #36681 Move server args config parser under utils: 配置解析器迁移，与配置访问统一化属于同一演进主线。