

PR #30492 完整报告

sgl-project/sglang

[refactor] Adopt `get_parallel()` everywhere and close out the parallel wrapper surface

合并时间: 2026-07-09 17:09

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/30492>

执行摘要

- 一句话: 全局统一并行拓扑访问模式, 启用采用检查
- 推荐动作: 值得精读。该 PR 展示了如何系统性地统一代码库中的 API 访问模式: 通过中心化 wrapper、逐步迁移、删除冗余、引入回归检查。其 `override()` 上下文管理器设计模式尤其值得借鉴。

功能与动机

根据 PR body: `ParallelContext` is the read-through surface for parallel topology, but adoption had gaps: the `decode-context-parallel` dimension was never wrapped, a few files regressed back to raw getters inside the already-swept directories, `dp_attention` still exported a re-export family shadowing the canonical accessors, and ~150 call sites outside the original sweep still read the raw getters directly.

实现拆解

1. 扩充 `ParallelContext` 维度: 在 `runtime_context.py` 的 `_PARALLEL_FIELDS` 和 `ParallelContext` 类中新增 `dcp_size`、`dcp_rank`、`dcp_group` 属性, 委托到 `parallel_state` 的对应 getter, 并注册到 `override()` 可覆盖字段。
2. 删除冗余重新导出: 在 `layers/dp_attention.py` 中移除 `compute_dp_attention_local_info` 整个函数、`_LOCAL_ATTN_DP_SIZE` / `_LOCAL_ATTN_DP_RANK` 全局变量, 以及 `get_attention_tp_group`、`get_attention_tp_rank`、`get_attention_tp_size`、`get_attention_cp_group`、`get_attention_cp_rank`、`get_attention_cp_size` 六个重新导出函数。这些函数的调用方直接改用 `get_parallel().attn_*`。
3. 全包调用点迁移: 对约 71 个文件中的并行拓扑读取调用进行替换。典型模式为将 `get_tensor_model_parallel_rank()` 改为 `get_parallel().tp_rank`, 将 `get_attention_tp_size()` 改为 `get_parallel().attn_tp_size`。涉及 `model_loader/loader.py`、`model_executor/` 下多个 runner 文件、`managers/`、`layers/` 子系统文件等。同时清理相应 `import` 语句, 统一从 `runtime_context` 导入 `get_parallel`。
4. 新增采用检查测试: 创建 `test/registered/unit/test_parallel_adoption_ratchet.py`, 用正则扫描 `sglang/srt` 包下排除豁免路径后的所有 `.py` 文件, 禁止出现直接调用原始 getter 的模式。豁免路径为 `distributed/` (定义原始 getter)、`layers/dp_attention.py` (委托基板)、`debug_utils/dumper.py` (第三方同名 getter)。

5. 适应性调整测试：部分测试因导入绑定被删除而改用 `get_parallel().override(...)` 模拟并行拓扑状态。

关键文件：

- `python/sglang/srt/runtime_context.py`（模块 运行上下文；类别 source；类型 core-logic；符号 `dcp_size`, `dcp_rank`, `dcp_group`）：并行拓扑 wrapper 的核心文件，新增 DCP 维度属性并更新 `override` 字段集。
- `python/sglang/srt/layers/dp_attention.py`（模块 DP 注意；类别 source；类型 core-logic；符号 `compute_dp_attention_local_info`, `get_attention_tp_group`, `get_attention_tp_rank`, `get_attention_tp_size`）：删除 6 个重新导出函数和 `compute_dp_attention_local_info`，简化初始化流程。
- `test/registered/unit/test_parallel_adoption_ratchet.py`（模块 合规检查；类别 test；类型 test-coverage；符号 `TestParallelAdoptionRatchet`, `test_no_legacy_parallel_getters_in_swept_dirs`）：新增采用检查测试，确保未来不允许回归原始 getter。
- `python/sglang/srt/model_loader/loader.py`（模块 模型加载；类别 source；类型 data-contract）：典型调用点迁移示例：从 raw getter 切换到 `get_parallel()`，同时清理 import。
- `python/sglang/srt/model_executor/model_runner_kv_cache_mixin.py`（模块 KV 缓存；类别 source；类型 data-contract）：大量 `get_attention_tp_size()` 替换为 `get_parallel().attn_tp_size`，展示跨模块迁移模式。
- `python/sglang/srt/model_executor/runner/decode_cuda_graph_runner.py`（模块 解码图运行；类别 source；类型 data-contract）：展示从 `distributed` 和 `dp_attention` 导入的 getter 一并迁移至 `get_parallel()`，并清理多余 import。

关键符号：`compute_dp_attention_local_info`, `get_attention_tp_group`, `get_attention_tp_rank`, `get_attention_tp_size`, `get_attention_cp_group`, `get_attention_cp_rank`, `get_attention_cp_size`, `dcp_size`, `dcp_rank`, `dcp_group`, `TestParallelAdoptionRatchet.test_no_legacy_parallel_getters_in_swept_dirs`

关键源码片段

`python/sglang/srt/runtime_context.py`

并行拓扑 wrapper 的核心文件，新增 DCP 维度属性并更新 `override` 字段集。

`python/sglang/srt/runtime_context.py`（片段）

```
_PARALLEL_FIELDS = frozenset(
{
    # ... 已有字段 ...
    "dcp_size", # 新增 DCP world size
    "dcp_rank", # 新增 DCP rank
    # ...
    "dcp_group", # 新增 DCP group
})
```

)

```
class ParallelContext:
    """并行拓扑命名空间；唯一实例状态是 _overrides。"""
    # ...

    @property
    def dcp_size(self) -> int:
        """返回 decode-context-parallel world size。"""
        return self._v("dcp_size", _ps().get_dcp_world_size)

    @property
    def dcp_rank(self) -> int:
        """返回 decode-context-parallel rank。"""
        return self._v("dcp_rank", _ps().get_dcp_rank)

    @property
    def dcp_group(self) -> Any:
        """返回 decode-context-parallel group。"""
        return self._v("dcp_group", _ps().get_dcp_group)
```

python/sglang/srt/layers/dp_attention.py

删除 6 个重新导出函数和 compute_dp_attention_local_info，简化初始化流程。

```
# python/sglang/srt/layers/dp_attention.py (变更后)

# 删除的符号：
# _LOCAL_ATTN_DP_SIZE, _LOCAL_ATTN_DP_RANK (全局变量)
# compute_dp_attention_local_info (整个函数)
# get_attention_tp_group / rank / size (6 个 re-export 函数)

def initialize_dp_attention(
    server_args: ServerArgs,
    model_config: ModelConfig,
):
    global _ATTN_DP_RANK, _ATTN_DP_SIZE
    dp = get_flags().dp
    dp.max_len_with_idle = (
        getattr(model_config.hf_config, "hybrid_override_pattern", None) is not None
    )
    enable_dp_attention = server_args.enable_dp_attention
    dp_size = server_args.dp_size
    attn_cp_size = server_args.attn_cp_size
    dp.enabled = enable_dp_attention

    tp_rank = get_tensor_model_parallel_rank()
    tp_size = get_tensor_model_parallel_world_size()

    _, _, _ATTN_DP_RANK, _ = compute_dp_attention_world_info(
```

```
enable_dp_attention, tp_rank, tp_size, dp_size, attn_cp_size
)
# 已移除 compute_dp_attention_local_info 调用和 _LOCAL_ATTN_DP_* 赋值
_ATTN_DP_SIZE = dp_size if enable_dp_attention else 1
# ...
```

评论区精华

无公开 review 评论，PR 由作者自行合并。PR 正文和提交信息已清晰说明动机和修改范围。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 回归风险（高）：约 71 个文件中约 80 个调用点被替换，任何遗漏或错误的映射（如 `attn_tp_size` 与 `get_attention_tp_size` 语义不一致）可能导致运行时错误。PR 中引入了采用检查测试作为回归屏障，并通过 DeepSeek-V3.2 tp2 冒烟测试验证热路径。
 - 删除风险（中）：`compute_dp_attention_local_info` 和局部 DP 变量被彻底删除，需确认没有任何外部模块（如 benchmark 脚本、实验性特性）依赖这些符号。PR body 称从不在读写路径外使用。
 - 测试覆盖风险（低）：单元测试套件和注意力 unittest 树（179 passed, 538 subtests）保持与 base 一致，但更大规模集成测试未被包含。
 - 第三方兼容性（低）：`debug_utils/dumper.py` 被豁免，因为其 Megatron 插件使用同名第三方 getter。
- 影响：
 - 开发体验：显著降低认知负载——整个 `sglang/srt` 包统一通过 `get_parallel()` 获取并行拓扑，配合 `override()` 上下文管理器使测试更加简单可靠。
 - 代码库健康度：消除了并行访问的两种风格（原始 getter vs wrapper），提升一致性；共减少约 415 行代码，删除大量重复 re-export。
 - 影响范围：跨 71 个文件，覆盖模型加载、运行时、缓存、注意力层、量化、调试等多个子系统，但每个文件改动量很小（多为替换调用点 + 调整 import）。
 - 向后兼容：无外部 API 破坏——删除的是内部 re-export 函数和从未公开的 `compute_dp_attention_local_info`。
 - 风险标记：大规模调用点替换，依赖关系复杂，测试覆盖要求高，可能遗漏调用点

关联脉络

- PR #30491 Unknown: 本 PR stacked on #30491，是同一系列重构的先行 PR。
- PR #30493 [refactor] Retire the legacy config accessor and the remaining process singletons: 同一作者同时期提交的另一个基础设施重构 PR，目标类似——统一访问模式并清理遗留代码。