PR #26567 完整报告

sgl-project/sglang

Speed up DeepGEMM JIT warmup with per-PP-rank parallel compile

合并时间：2026-06-02 10:51

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26567

执行摘要

- 一句话：PP 并行 DeepGEMM JIT 预热，启动时间减少约 60%
- 推荐动作：建议精读以下设计决策：
  1. batch size 的硬件感知推导方法（从 SM 数量和 block_m 推算 n_splits 区间），可推广到其他类似场景。
  2. _dummy_run 的 forward_mode_override 设计，解耦了 forward mode 与 is_generation，提高可测试性。
  3. 对 PD 分解模式的优雅处理（根据 disaggregation_mode 跳过不必要的 DECODE/EXTEND）。
  4. 将并行预热逻辑封装在 compile_utils.py，保持 kernel_warmup 的简洁性。

功能与动机

Today the startup warmup /generate request flows through PP stages serially: stage k can only start its DeepGEMM JIT compile after stage k-1 finishes. For large MoE models like DeepSeek-V4 this dominates startup time.

实现拆解

1. 环境变量开关：在 environ.py 中新增 SGLANG_PP_PARALLEL_DEEPGEMM_WARMUP（默认关闭），作为功能的守护条件。
2. 调用入口：在 model_runner.py 的 kernel_warmup() 方法中，在 FlashInfer 自动调谐之后，检查环境变量、ENABLE_JIT_DEEPGEMM、pp_size > 1、非 speculative 等条件，满足时调用 pp_parallel_deep_gemm_warmup(self)。
3. 核心逻辑：在 compile_utils.py 中新增 pp_parallel_deep_gemm_warmup 函数。该函数根据设备的 SM 数量推导 5 个代表 batch size（使得覆盖 n_splits 从 n_sms 到 1 的各个区间），并根据 disaggregation_mode 决定是否执行 DECODE 和 EXTEND 的 dummy forward。对于每个 bs，通过 _dummy_run(forward_mode_override=ForwardMode.DECODE/EXTEND) 触发对应形状的 JIT 编译。
4. _dummy_run 扩展：_dummy_run 新增 forward_mode_override 参数，替代原来仅靠 is_generation 判断 EXTEND/DECODE 的逻辑，使得生成模型也可以通过 override 执行 EXTEND 模式的 dummy。同时修复了 pp_proxy_tensors 切片条件（由 DSA-specific 改为通用判断）、传递 hc_hidden_size 等，保证 DSv4 等模型的兼容性。

5. 配套调整：compile_utils.py 顶部新增 import time、ForwardMode、ceil_align 等导入，避免引入循环依赖。

关键文件：

- python/sglang/srt/model_executor/model_runner.py（模块 模型运行器；类别 source；类型 data-contract；符号 kernel_warmup, _dummy_run）：内核预热入口和 dummy forward 的核心修改，新增 forward_mode_override 参数，修改 pp_proxy_tensors 切片逻辑
- python/sglang/srt/layers/deep_gemm_wrapper/compile_utils.py（模块 DeepGEMM 编译；类别 source；类型 core-logic；符号 pp_parallel_deep_gemm_warmup）：新增 pp_parallel_deep_gemm_warmup 函数，实现并行预热核心逻辑
- python/sglang/srt/environ.py（模块 环境配置；类别 source；类型 configuration；符号 SGLANG_PP_PARALLEL_DEEPGEMM_WARMUP）：新增 SGLANG_PP_PARALLEL_DEEPGEMM_WARMUP 环境变量开关

关键符号：kernel_warmup, _dummy_run, pp_parallel_deep_gemm_warmup

关键源码片段

python/sglang/srt/model_executor/model_runner.py

内核预热入口和 dummy forward 的核心修改，新增 forward_mode_override 参数，修改 pp_proxy_tensors 切片逻辑

```
def kernel_warmup(self):
    """Warmup and tune kernels before cuda graph capture."""
    if self.device != "cuda":
        return

    if self._should_run_flashinfer_autotune():
        self._flashinfer_autotune()

    # PP-parallel DeepGEMM warmup: 仅在启用环境变量、pp_size > 1、
    # JIT DeepGEMM 已开启且非 speculative 模式时执行
    if (
        envs.SGLANG_PP_PARALLEL_DEEPGEMM_WARMUP.get()
        and deep_gemm_wrapper.ENABLE_JIT_DEEPGEMM
        and self.pp_size > 1
        and not self.spec_algorithm.is_speculative()
    ):
        from sglang.srt.layers.deep_gemm_wrapper.compile_utils import (
            pp_parallel_deep_gemm_warmup,
        )

        pp_parallel_deep_gemm_warmup(self)
```

python/sglang/srt/layers/deep_gemm_wrapper/compile_utils.py

新增 pp_parallel_deep_gemm_warmup 函数，实现并行预热核心逻辑

```
def pp_parallel_deep_gemm_warmup(model_runner) -> None:
    """为每个 PP rank 执行本地 dummy DECODE 和 EXTEND 前向传播，
    使得各 rank 的 DeepGEMM JIT 编译可以并行进行，而不是通过
    启动时的 /generate 请求串行触发。

    通过环境变量 SGLANG_PP_PARALLEL_DEEPGEMM_WARMUP 启用。
    """
    # n_splits ≈ n_sms / ceil(bs / block_m)，其中 block_m = 64；
    # 选取 5 个代表性的 bs 覆盖实际流量中的各个 n_splits 区间
    #  （最小 decode 形状、中低、两个中等、n_splits = 1 的 ~5K 预填充）。
    # 对 bs 执行 ceil_align 确保满足 attn_cp_size 对齐要求
    #  （DSA 预填充 CP 需要 seq_len % cp_size == 0）。
    n_sms = torch.cuda.get_device_properties(model_runner.device).multi_processor_count
    block_m = 64
    cp = max(model_runner.attn_cp_size, 1)
    batch_sizes = sorted(
        {
            ceil_align(bs, cp)
            for bs in (
                1, # n_splits = n_sms
                2 * block_m, # 中低区间
                max(n_sms // 8, 2) * block_m, # 中等区间
                max(n_sms // 4, 4) * block_m, # 中等区间
                n_sms * block_m, # n_splits = 1
            )
        }
    )

    # PD 分解模式下，prefill-only 节点不应执行 decode dummy，
    # decode-only 节点不应执行 extend dummy，避免 OOM 或索引器异常。
    disagg_mode = model_runner.server_args.disaggregation_mode
    run_decode = model_runner.is_generation and disagg_mode != "prefill"
    run_extend = disagg_mode != "decode"

    logger.info(
        "PP-parallel DeepGEMM warmup start "
        "(pp_rank=%d, tp_rank=%d, batch_sizes=%s, disagg=%s).",
        model_runner.pp_rank,
        model_runner.tp_rank,
        batch_sizes,
        disagg_mode,
    )

    t0 = time.perf_counter()
    with torch.inference_mode():
        for bs in batch_sizes:
            if run_decode:
                model_runner._dummy_run(
                    batch_size=bs, forward_mode_override=ForwardMode.DECODE
```

```
            )
        if run_extend:
            model_runner._dummy_run(
                batch_size=bs, forward_mode_override=ForwardMode.EXTEND
            )

    logger.info(
        "PP-parallel DeepGEMM warmup done in %.2fs (pp_rank=%d).",
        time.perf_counter() - t0,
        model_runner.pp_rank,
    )
```

评论区精华

1. Fridge003: 要求将 _pp_parallel_deep_gemm_warmup 函数移至 compile_utils.py，并添加环境变量保护。作者已采纳，将函数移入 compile_utils.py 并添加 SGLANG_PP_PARALLEL_DEEPGEMM_WARMUP（默认关闭）的 env gate。
2. Fridge003质疑 batch size 的选择。whybeyoung解释：DeepGEMM 的 grouped GEMM 依据 n_splits ≈ n_sms / ceil(bs/block_m) 选择 JIT 内核，通过 5 个 bs 覆盖所有档位。该解释被接受。
3. Fridge003指出 _dummy_run 中原有 not self.is_generation 判断在引入 forward_mode_override 后应改为 capture_forward_mode == EXTEND。BBuf确认这一修改不仅正确，还修复了在"生成模型 + override=EXTEND"场景下的潜在 bug。
4. Fridge003建议移除 pp_proxy_tensors 切片中对 is_dsa_enable_prefill_cp() 的依赖，使其通用化。作者在第 8 个提交中移除条件判断，改为通用切片逻辑（attn_cp_size > 1 时执行切片）。
5. gemini-code-assist[bot]建议在异常日志中使用 exc_info=True 以记录完整 traceback。该建议在后续提交中因去掉了 try/except 未被采用，但仍有参考价值。

- 函数位置与环境变量保护 (design): 采用：函数移至 compile_utils.py，新增 SGLANG_PP_PARALLEL_DEEPGEMM_WARMUP 环境变量（默认关闭）。
- Batch size 选取依据 (design): 接受解释，batch size 保持基于 SM 数量的推导。
- is_generation 改为 forward_mode_override 的正确性 (correctness): 修改被接受。
- pp_proxy_tensors 切片条件应通用化 (correctness): 在第 8 个提交中移除 DSA-specific 检查，改为仅在 attn_cp_size > 1 时通用切片。
- 异常日志记录完整 traceback (other): 后续提交去掉了 try/except，异常直接上抛，未采用该建议但仍保留参考价值。

风险与影响

- 风险:
  1. 回归风险：_dummy_run 的逻辑修改（forward_mode_override、extend buffer 设置、pp_proxy_tensors 切片）可能影响现有 FlashInfer autotune 路径。但 BBuf 和 Fridge003 已审阅确认兼容，且在 CI 中验证通过。

2. 兼容性：新增环境变量默认关闭，不影响现有行为。但启动时若设置该变量，仅在 PP>1 且 DeepGEMM JIT 开启时生效，不涉及其他模型。
3. 无测试覆盖：PR 没有添加直接测试文件，依赖现有 CI 和手动验证。对于 *edge case*（如 PD 分解、CP 配置）可能存在遗漏。
4. 资源占用：并行编译可能同时启动多个 DeepGEMM JIT 进程，但受限于 SGLANG_JIT_DEEPGEMM_COMPILE_WORKERS 变量，风险可控。

- 影响：
  1. 用户影响：DeepSeek-V4 等大型 MoE 模型启动时间可缩短约 60%（从 9-12 分钟降至 3.5-4 分钟），显著提升部署体验。其他模型不受影响。
  2. 系统影响：无运行时性能变化，仅影响启动阶段。并行编译可能短暂增加 CPU 和 GPU 内存使用，但已在 batch size 和 worker 数量上加以限制。
  3. 团队影响：新增的内部函数和配置项需要维护；_dummy_run 的接口扩展（增加可选参数）向后续 warmup 扩展开放。 - 风险标记：核心路径变更（_dummy_run），缺少测试覆盖，仅对 DeepGEMM 生效，默认关闭降低风险

关联脉络

- 暂无明显关联 PR