PR #23472 完整报告

sgl-project/sglang

[Intel GPU] Enable pipeline parallelism on XPU

合并时间：2026-04-24 10:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23472

执行摘要

- 一句话：XPU 流水线并行支持，修复死锁
- 推荐动作：建议精读此 PR，尤其关注设备无关化模式和奇偶排序的设计，对多硬件后端支持有借鉴意义。可快速合并。

功能与动机

此前 Pipeline Parallelism (PP) 仅支持 CUDA。在 Intel XPU 上启动 `--pp-size > 1` 的服务会立即崩溃（`RuntimeError: Tried to instantiate dummy base class Event`），因为 `SchedulerPPMixin` 硬编码了 `torch.cuda` 调用。即使修复了 CUDA 调用，`PP >= 2` 时所有 rank 在 `torch.distributed.isend` 中忙轮询等待匹配的 `recv`，导致 100% CPU 死锁。本 PR 使 PP 在 XPU 上正常工作，并泛化到任何非 CUDA 后端。

实现拆解

1. 设备无关化：在 `scheduler_pp_mixin.py` 中导入 `is_xpu`，将 `torch.cuda.Event`、`torch.cuda.current_stream()`、`torch.cuda.synchronize()` 替换为通过 `self.device_module`（由 `torch.get_device_module()` 返回）调用的对应方法。涉及 `event_loop_pp`、`event_loop_pp_disagg_prefill`、`event_loop_pp_disagg_decode`、`init_pp_loop_state`、`profile_and_init_predictor`、`_pp_commit_send_output_work_and_preprocess_output_tensors` 等多个函数。
2. 引入奇偶排序：重写 `_pp_send_recv_and_preprocess_output_tensors`，定义 `_do_send` 和 `_do_recv` 辅助闭包，根据 `is_xpu()` 和 rank 奇偶决定先发后收还是先收后发。对于 CUDA，保持原有全部先发后收的行为；对于 XPU，偶 rank 先发后收，奇 rank 先收后发，确保相邻 rank 总是有一个发送者和一个接收者同时就绪，避免死锁。
3. 验证：在 4× Intel XPU 上通过 `TP=1 PP=2/3/4` 和 `TP=2 PP=2` 配置，以及 `TestPPAccuracy.test_logprob` 测试。CUDA 行为未受影响。

关键文件：

- python/sglang/srt/managers/scheduler_pp_mixin.py（模块 调度器；类别 source；类型 core-logic；符号 _do_send, _do_recv, init_pp_loop_state, profile_and_init_predictor）：核心调度器模块，流水线并行逻辑所在；替换 CUDA 硬编码为设备无关调用，并引入奇偶排序修复 XPU 死锁。

关键符号：_do_send, _do_recv, _pp_send_recv_and_preprocess_output_tensors

关键源码片段

python/sglang/srt/managers/scheduler_pp_mixin.py

核心调度器模块，流水线并行逻辑所在；替换 CUDA 硬编码为设备无关调用，并引入奇偶排序修复 XPU 死锁。

```
# 类型提示从 torch.cuda.Event 改为 torch.Event（支持任意后端）
self.last_rank_comm_queue: deque[Tuple[torch.Event, PPProxyTensors]] = deque()

# 使用 self.device_module 替代 torch.cuda 进行流同步
if not self.pp_group.is_last_rank:
    if self.cur_batch:
        self.device_module.current_stream().wait_event(self.launch_event)

# 性能分析中设备同步
self.device_module.synchronize() # 替代 torch.cuda.synchronize()

# 在 _pp_send_recv_and_preprocess_output_tensors 中，根据设备决定顺序
send_first = (not is_xpu()) or ((self.pp_rank % 2) == 0)
if send_first:
    self._do_send(...)
    self._do_recv(...)
else:
    self._do_recv(...)
    self._do_send(...)
```

评论区精华

- 类型提示兼容性：gemini-code-assist[bot] 指出 torch.Event 在 PyTorch 2.4 之前不可用，且返回类型缺省 Optional。作者回应项目基线 PyTorch ≥2.9，无兼容问题，并修正了返回类型标注。
- 奇偶排序原理：ShangmingCai 质疑 isend 异步应先发，作者解释 XPU 的 isend 实际同步导致死锁，奇偶排序确保匹配；ShangmingCai 建议仅对 XPU 启用，作者通过 is_xpu() 门控实现，CUDA 行为不变。
  - 类型提示中 torch.Event 的兼容性 (design): 保留 torch.Event 并修正返回类型标注。
  - 基于 rank 奇偶性的 send/recv 顺序 (correctness): 仅在 XPU 上启用奇偶顺序，CUDA 保持原有行为。

风险与影响

- 风险：
  - CUDA 回归：设备模块替换理论上不影响 CUDA（get_device_module() 返回 torch.cuda），但需验证 torch.Event 在 CUDA 上的行为是否完全一致。项目基线 ≥2.9 降低了风险。
  - XPU 死锁：奇偶排序仅 XPU 启用，且经过单机多卡验证，但未覆盖分布式多节点场景。

- 其他后端：如果未来支持 AMD ROCm 或 Apple MPS，需要类似的门控，但目前未处理，设计上易于扩展。
- 影响：对 Intel XPU 用户是实质性改进，使流水线并行可用，大幅提升大模型推理吞吐。对 CUDA 用户无任何影响。代码库向设备无关迈进一步，降低后续支持新硬件的成本。仅修改一个核心文件，但涉及调度循环关键路径，需仔细 review。
- 风险标记：XPU 特有死锁风险，CUDA 回归需验证，PyTorch 版本依赖

关联脉络

- PR #23641 Revert "[Intel GPU] Enable pipeline parallelism on XPU": 回退了之前的 PP on XPU 实现，本 PR 在此基础上修复后重新启用。