PR #23645 完整报告

sgl-project/sglang

[Intel GPU] Enable pipeline parallelism on XPU

合并时间：2026-04-24 19:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23645

执行摘要

- 一句话：支持 XPU 流水线并行，设备无关化并修复死锁
- 推荐动作：值得精读，尤其是设备无关化和奇偶通信策略的设计。对于需要支持多后端（AMD、MUSA 等）的团队，此为可复用的模式。PR 的 review 讨论也展示了如何识别并修复因初始化顺序导致的问题。

功能与动机

Pipeline parallelism (PP) only ran on CUDA. On Intel XPU, launching any --pp-size > 1 server crashed at startup with RuntimeError: Tried to instantiate dummy base class Event because SchedulerPPMixin hard-codes torch.cuda.{Event, current_stream, synchronize}. Even after fixing the hard-coded CUDA calls, PP >= 2 livelocked during the first multi-rank communication: with XCCL on XPU, torch.distributed.isend busy-polls waiting for a matching recv rendezvous, so when every PP rank sent before receiving, all ranks spun at 100% CPU inside torch.distributed and none ever reached its recv.

实现拆解

1. 设备无关化

在 scheduler_pp_mixin.py 中，将对 torch.cuda.Event()、torch.cuda.current_stream()、torch.cuda.synchronize() 的调用替换为通过 get_device_module() 返回的设备模块的动态调用，并更新类型提示（torch.cuda.Event → torch.Event）。

2. 奇偶顺序 send/recv

在 _pp_send_recv_and_preprocess_output_tensors 中，基于 PP rank 的奇偶性调整 send 和 recv 的执行顺序，仅在 XPU 上生效。偶数 rank 先 send 后 recv，奇数 rank 先 recv 后 send，确保每一对相邻 rank 的 isend 都有匹配的等待 recv。

3. 修复 profile_and_init_predictor 的初始化顺序

使用模块级函数 get_device_module() 替代 self.device_module，因为 profile_and_init_predictor 在 Scheduler.init_overlap 之前调用，self.device_module 尚未赋值。

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）：PR 的唯一变更文件，核心调度器混入类，包含设备无关化、奇偶通信顺序及初始化修复。

关键符号：*event_loop_pp*, *event_loop_pp_disagg_prefill*, *event_loop_pp_disagg_decode*, _pp_send_recv_and_preprocess_output_tensors, _pp_commit_send_output_work_and_preprocess_output_tensors, init_pp_loop_state, profile_and_init_predictor, _do_send, _do_recv

关键源码片段

python/sglang/srt/managers/scheduler_pp_mixin.py

PR 的唯一变更文件，核心调度器混入类，包含设备无关化、奇偶通信顺序及初始化修复。

```
# 设备无关化：用 self.device_module.current_stream() 替代 torch.cuda.current_stream()
# 在 event_loop_pp 中
if not self.pp_group.is_last_rank:
    if self.cur_batch:
        self.device_module.current_stream().wait_event(self.launch_event)
        with torch.profiler.record_function("send_proxy_dict_to_next_stage"):
            self.send_proxy_work = self._pp_send_dict_to_next_stage(
                result.pp_hidden_states_proxy_tensors.tensors,
                async_send=True,
                msg_type="proxy",
            )

# 修复 profile_and_init_predictor 中设备同步的初始化顺序
# 使用 get_device_module() 替代 self.device_module（可能未赋值）
def profile_and_init_predictor(self: Scheduler):
    # ...
    device_module = get_device_module()
    device_module.synchronize() # 替代 torch.cuda.synchronize()
    start = time.perf_counter()
    # ... 模型前向推理 ...
    device_module.synchronize()
    latency = time.perf_counter() - start
    # ...
```

评论区精华

讨论 1：torch.Event 类型提示

- gemini-code-assist[bot] 建议将 torch.Event 替换为 Any，因为旧版 PyTorch 没有顶层 Event 类。
- siju-samuel 回应：项目要求 PyTorch ≥ 2.9，torch.Event 在 PyTorch 2.4 后已是公共类，保留更具体的类型提示更清晰。
- 结论：未采纳建议，类型提示保持不变。

讨论 2: profile_and_init_predictor 中 self.device_module 未初始化

- ShangmingCai 指出 CI 失败日志显示 'Scheduler' object has no attribute 'device_module'，原因是动态分块初始化（init_chunked_prefill）早于 self.device_module 的赋值（init_overlap）。
- siju-samuel 确认并使用 get_device_module() 替换所有相关调用，消除对实例属性的依赖。
- 结论：已修复，动态分块功能正常。
- 类型提示：torch.Event 与 Any 的选择 (design): 保留 torch.Event，因为兼容性基线满足要求，且提供更清晰的文档含义。
- profile_and_init_predictor 中 self.device_module 未初始化 (correctness): 用 get_device_module() 替换 self.device_module，消除对实例状态的依赖。

风险与影响

- 风险：
  1. 类型提示兼容性：torch.Event 在 PyTorch < 2.4 不可用，但项目基线为 2.9，风险可控。
  2. XPU 特定代码路径：奇偶顺序仅当 is_xpu() 为真时生效，CUDA 路径不受影响。测试已覆盖 CUDA 下的 PP 正确性。
  3. 动态分块依赖：修复前 profile_and_init_predictor 会静默失败并禁用动态分块；修复后行为正确，但仍需确保各初始化路径顺序。
  4. 单一文件修改：所有改动集中在 scheduler_pp_mixin.py，回归范围明确。
- 影响：
  - XPU 用户：现在可以使用 --pp-size > 1 进行多卡流水线并行，LLM 推理吞吐量显著提升（benchmark 显示 PP=4 时输入吞吐 13051 tok/s）。
  - CUDA 用户：无任何行为变化，因 get_device_module() 在 CUDA 上返回 torch.cuda，且奇偶顺序仅对 XPU 生效。
  - 其他后端：设备无关化设计为未来在 AMD、NPU 等后端启用 PP 扫清了障碍。
  - 维护影响：后续任何后端都只需在 get_device_module() 中注册，无需修改调度器逻辑。
  - 风险标记：核心调度器变更，XPU 特定代码路径，动态分块初始化顺序依赖

关联脉络

- PR #23472 [Intel GPU] Enable pipeline parallelism on XPU: 为前一个被回退的同一功能 PR，本 PR 为其第二版，修复了死锁和初始化问题。
- PR #23641 Revert "[Intel GPU] Enable pipeline parallelism on XPU": 回退预览版 PR，因 CI 中断；本 PR 在其基础上重新实现并修复问题。