PR #7420 完整报告

PaddlePaddle/FastDeploy

[BugFix][XPU] Fix kv_cache management bug

合并时间：2026-04-16 15:45

原文链接：http://prhub.com.cn/PaddlePaddle/FastDeploy/pull/7420

执行摘要

- 一句话：修复 XPU model runner 在开启 attention store 时重复创建 KV cache 的问题。
- 推荐动作：该 PR 值得精读，特别是对于关注 XPU 平台优化和 KV cache 管理的工程师。关键设计决策在于将条件逻辑从单一检查扩展为多条件组合，这反映了对 cache 管理策略的细化，值得学习其与 GPU/Metax 实现对齐的思路。

功能与动机

根据 PR body 描述，当 XPU 开启 attention store 时，当前存在在 model runner 中重复创建 KV cache 的过程，需要解决重复分配 cache 问题。这可能导致内存浪费或潜在的性能问题。

实现拆解

1. 修改 KV cache 创建条件逻辑：在 fastdeploy/worker/xpu_model_runner.py 文件的 initialize_kv_cache 方法中，将原有的 create_cache_tensor 条件从 profile or self.scheduler_config.splitwise_role == "mixed" 扩展为更全面的判断。
2. 新增条件判断：新逻辑检查 fd_config.cache_config.num_cpu_blocks > 0 或 fd_config.cache_config.kvcache_storage_backend 或 self.fd_config.scheduler_config.splitwise_role != "mixed"，只有当这些条件都不满足时才创建 cache tensor。
3. 对齐标准实现：此修改旨在使 XPU model runner 的 KV cache 管理逻辑与 GPU/Metax 的标准实现保持一致，确保在开启 attention store 等场景下不会重复分配内存。

关键文件：

- fastdeploy/worker/xpu_model_runner.py（模块 工作器；类别 source；类型 core-logic；符号 initialize_kv_cache）：这是唯一修改的文件，包含 KV cache 初始化逻辑的核心变更，直接影响 XPU 平台的内存管理。

关键符号：initialize_kv_cache

关键源码片段

fastdeploy/worker/xpu_model_runner.py

这是唯一修改的文件，包含KV cache初始化逻辑的核心变更，直接影响XPU平台的内存管理。

```
def initialize_kv_cache(self, profile: bool = False) -> None:
```

```
# Check if gpu runner needs to create kv cache
# 1. During profiling, it creates its own kv cache.
# 2. GPU runner creates kv cache tensor unless p/d disaggregation is enabled.
# 修改前：create_cache_tensor = profile or self.scheduler_config.splitwise_role == "mixed"
# 修改后：扩展条件，检查多个配置项，避免在开启 attention store 时重复创建 cache
create_cache_tensor = profile or not (
    self.fd_config.cache_config.num_cpu_blocks > 0
    or self.fd_config.cache_config.kvcache_storage_backend
    or self.fd_config.scheduler_config.splitwise_role != "mixed"
)
if not create_cache_tensor:
    logger.info(f"Waiting for cache managers to create kv cache.. {cache_ready_signal.value}")
    while cache_ready_signal.value[local_rank] != 1:
        # 等待 cache 准备就绪
        pass
```

评论区精华

Review 中 PaddlePaddle-bot 的评论指出，原逻辑只检查 splitwise_role == "mixed"，忽略了 num_cpu_blocks 和 kvcache_storage_backend 两个条件，导致在开启 attention store 时会重复创建 KV cache。修改后的逻辑正确地修复了这个问题，代码简洁且符合项目规范。另一位 reviewer hong19860320 简单批准（LGTM）。没有出现争议或未解决的疑虑。

- KV cache 创建逻辑对齐 (correctness): 修改后的逻辑正确修复了问题，与 GPU/Metax 标准实现保持一致。

风险与影响

- 风险：技术风险较低：
- 回归风险：修改仅涉及条件逻辑，未改变核心算法，且与 GPU/Metax 实现对齐，回归风险可控。
- 性能风险：修复了重复内存分配问题，可能提升内存使用效率，但需验证新条件判断是否在所有 XPU 部署场景下正确。
- 兼容性风险：逻辑变更可能影响依赖特定 cache 创建行为的边缘场景，但 PR 描述未提及，需结合测试验证。
- 安全风险：无直接影响。
- 影响：影响范围：
- 用户影响：对使用 XPU 平台且开启 attention store 的用户，修复了潜在的 KV cache 重复创建问题，可能改善内存使用和稳定性。
- 系统影响：仅影响 XPU model runner 的 KV cache 初始化逻辑，不涉及其他模块或平台。
- 团队影响：为 XPU 平台维护提供了更一致的 cache 管理实现，便于后续开发和调试。 影响程度：中等，针对特定平台和配置的 bugfix，但涉及核心内存管理组件。
- 风险标记：配置逻辑变更 , 内存管理调整

关联脉络

- PR #7364 [BugFix][PD Disaggregation][KVCache] Fix low cache hit rate in PD split (disaggregation) scenario: 同样涉及 KVCache 管理问题的修复，但针对 PD 分离场景，而本 PR 针对 XPU 平台，两者互补展示了 cache 管理的复杂性。
- PR #7407 [BugFix][Scheduler]Fix FD_DISABLE_CHUNKED_PREFILL max_num_batched_tokens limit: 同为 bugfix 类型，涉及调度器和配置调整，与本 PR 在修复配置相关逻辑上有相似之处。