

PR #35017 完整报告

sgl-project/sglang

[Scheduler] Add configurable decode interval after prefill

合并时间: 2026-08-20 03:01

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/35017>

执行摘要

- 一句话: 新增 prefill 后 decode 间隔参数, 防 DP 下 decode 饿死
- 推荐动作: 值得精读。核心亮点是“复用已有全局同步 flag、零额外通信实现跨 rank 调度协同”的设计, 对理解 SGLang DP attention 下的调度公平性问题很有帮助。建议关注两点:
 - 一是 `_should_defer_prefill()` 的计数在空转轮次下也会消耗, 实际间隔效果需要实测校准;
 - 二是该开关与 `prefill_delayer` 等其他调度策略的叠加行为尚未验证, 未来可能演进为统一的调度节奏控制机制。

功能与动机

PR body 明确指出: *With DP attention, each rank schedules attention work locally while all ranks still participate in the same MoE collective. A rank with repeated chunked-prefill continuation work can therefore keep winning SGLang's prefill-first scheduling decision and starve decode work on peer ranks. In the AgentX workload this left requests waiting while the actual decode batch remained small.* 作者还说明 DP-attention 的 MLP-sync all-gather 已经暴露了是否有任意 rank 选择 EXTEND batch, 因此可以直接复用这个全局同步信号来协调各 rank 的 prefill/decode 节奏, 而不需要额外通信。

实现拆解

1. 新增服务端参数: 在 `python/sglang/srt/server_args.py` 中定义 `prefill_decode_interval` (int, 默认 0, NS('schedule')) , 并在 `_run_resolution_pipeline` 中调用新增的 `_validate_prefill_decode_interval()`, 对负值抛出 `ValueError('--prefill-decode-interval must be non-negative.')`。校验放在 `dummy-model` 边界之前, 所有启动路径都会生效。
2. 调度器状态初始化: 在 `python/sglang/srt/managers/scheduler.py` 的 `init_chunked_prefill()` 中读取参数并初始化 `_prefill_decode_interval_remaining = 0`, 作为每 rank 的本地间隔计数。
3. 核心调度逻辑: 新增两个方法 `_arm_prefill_decode_interval(batch)` 与 `_should_defer_prefill()`。前者在每个批次选出后调用, DP attention 下用全局同步的 `batch.is_extend_in_batch` 判断是否武装间隔, 非 DP 下用 `batch.forward_mode.is_extend()`; 后者在 `get_next_batch_to_run()` 的 prefill 分支前检查, 每次调用消耗 1 个计数并返回是否应跳过本轮 prefill, 间隔生效期间 `new_batch = None`, 让当前 batch 继续 decode。

4. 测试配套：新增 test/registered/unit/managers/test_scheduler_prefill_decode_interval.py，覆盖禁用时不武装、非 DP 使用本地 forward mode、DP 使用全局同步标志、decode 批次不重新武装四种场景；test/registered/unit/server_args/test_server_args.py 增加参数解析与负值校验测试；test/registered/unit/managers/test_scheduler_chunked_req_gate.py 为 mock scheduler 补充新增字段以避免属性缺失。
5. 无文档与部署配套改动：body 明确说明不需要 AIPerf 源码变更，benchmark 原型直接使用现有 per-session 路由行为；文档更新未勾选。

关键文件：

- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 _should_defer_prefill, _arm_prefill_decode_interval, init_chunked_prefill, get_next_batch_to_run）：核心实现文件：在调度器核心路径 get_next_batch_to_run 中新增 prefill 延迟分支，并新增 _arm_prefill_decode_interval / _should_defer_prefill 两个方法实现间隔状态机。
- python/sglang/srt/server_args.py（模块 服务参数；类别 source；类型 configuration；符号 prefill_decode_interval, _validate_prefill_decode_interval）：新增服务参数 prefill_decode_interval 及其非负校验逻辑，是功能的入口配置。
- test/registered/unit/managers/test_scheduler_prefill_decode_interval.py（模块 调度测试；类别 test；类型 test-coverage；符号 _make_scheduler, _make_batch, TestPrefillDecodeInterval, test_disabled_interval_does_not_arm）：新增的调度器单元测试文件，覆盖四种关键场景，直接决定了间隔状态机的正确性验证。
- test/registered/unit/server_args/test_server_args.py（模块 参数测试；类别 test；类型 test-coverage；符号 test_prefill_decode_interval）：为新增参数补充解析与校验测试，保证 --prefill-decode-interval 的合法值和非法负值都能被正确拦截。
- test/registered/unit/managers/test_scheduler_chunked_req_gate.py（模块 测试夹具；类别 test；类型 test-fixture）：适配测试夹具：为 mock scheduler 对象补充 prefill_decode_interval 与 _prefill_decode_interval_remaining 字段，避免因调度器新增属性导致既有测试报错。

关键符号：_should_defer_prefill, _arm_prefill_decode_interval, _validate_prefill_decode_interval, get_next_batch_to_run, init_chunked_prefill

关键源码片段

python/sglang/srt/managers/scheduler.py

核心实现文件：在调度器核心路径 `get_next_batch_to_run` 中新增 prefill 延迟分支，并新增 `_arm_prefill_decode_interval` / `_should_defer_prefill` 两个方法实现间隔状态机。

```
def _should_defer_prefill(self) -> bool:
    # 间隔计数为 0 时不做任何拦截，调度行为与原来完全一致。
    if self._prefill_decode_interval_remaining == 0:
        return False

    # 每个 scheduler 轮次调用一次，消耗 1 个计数，直到降为 0。
    self._prefill_decode_interval_remaining -= 1
```

```

return True

def _arm_prefill_decode_interval(self, batch: Optional[ScheduleBatch]) -> None:
    # 默认配置为 0 或批次为空时不武装，保持向后兼容。
    if self.prefill_decode_interval == 0 or batch is None:
        return

    # DP attention 下 `is_extend_in_batch` 由 MLP-sync all-gather 全局同步，
    # 即使本 rank 只有 decode 工作、其他 rank 正在 prefill，也会触发间隔；
    # 非 DP 场景则直接使用本地 forward mode 判断。
    is_extend = (
        batch.is_extend_in_batch
        if self.require_mlp_sync
        else batch.forward_mode.is_extend()
    )
    if is_extend:
        self._prefill_decode_interval_remaining = self.prefill_decode_interval

# get_next_batch_to_run 中的关键分支：间隔生效期间跳过 prefill 调度。
if self.dlrm_config is not None:
    new_batch = self.get_new_batch_dlrm(running_batch)
elif self._should_defer_prefill():
    # 不调度新 prefill，让当前 batch 继续 decode。
    new_batch = None
else:
    prefill_plan = self.get_new_batch_prefill(running_batch)
    new_batch = prefill_plan.batch_to_run

# 选完批次并完成 MLP-sync 准备后，再判断是否需要重新武装间隔；
# decode 批次不会触发重新武装，因此计数可以在 decode 期间自然耗尽。
ret = self.dp_attn_adapter.maybe_prepare_mlp_sync_batch(ret, need_sync=need_mlp_sync)
self._arm_prefill_decode_interval(ret)

```

python/sglang/srt/server_args.py

新增服务参数 `prefill_decode_interval` 及其非负校验逻辑，是功能的入口配置。

```

# 新增调度参数：prefill 后强制运行多少个 decode 轮次。
# DP attention 模式下该间隔会跨所有 DP rank 同步，0 表示禁用。
prefill_decode_interval: A[
    int,
    "The number of decode rounds to run after a prefill batch before scheduling the next prefill.
    In data-parallel attention mode, the interval is synchronized across all DP ranks. Set to 0 to
    disable.",
    NS("schedule"),
] = 0

```

参数校验：只接受非负整数，非法值在启动早期直接报错。

```
def _validate_prefill_decode_interval(self):
    if self.prefill_decode_interval < 0:
        raise ValueError("--prefill-decode-interval must be non-negative.")
```

test/registered/unit/managers/test_scheduler_prefill_decode_interval.py

新增的调度器单元测试文件，覆盖四种关键场景，直接决定了间隔状态机的正确性验证。

```
def _make_scheduler(*, interval: int, require_mlp_sync: bool) -> Scheduler:
    # 用 __new__ 绕过完整初始化，只为被测方法准备必要字段。
    scheduler = Scheduler.__new__(Scheduler)
    scheduler.prefill_decode_interval = interval
    scheduler._prefill_decode_interval_remaining = 0
    scheduler.require_mlp_sync = require_mlp_sync
    return scheduler
```

```
def _make_batch(*, local_extend: bool, global_extend: bool):
    # 同时模拟本地 forward mode 与 DP 全局同步的 EXTEND 标志。
    return SimpleNamespace(
        forward_mode=SimpleNamespace(is_extend=lambda: local_extend),
        is_extend_in_batch=global_extend,
    )
```

```
class TestPrefillDecodeInterval(unittest.TestCase):
    def test_dp_interval_uses_globally_synchronized_extend_flag(self):
        scheduler = _make_scheduler(interval=2, require_mlp_sync=True)

        # 本 rank 本地是 decode，但其他 DP rank 正在 prefill，
        # 全局 flag 为 True 时也必须武装间隔，保证所有 rank 节奏一致。
        scheduler._arm_prefill_decode_interval(
            _make_batch(local_extend=False, global_extend=True)
        )

        self.assertEqual(scheduler._prefill_decode_interval_remaining, 2)
        self.assertTrue(scheduler._should_defer_prefill())
```

评论区精华

该 PR 没有代码级 review 评论，maintainer YAMY1234 直接 APPROVED。评论区主要是 CI 运维与基准验证记录：作者多次发起 `/rerun-failed-ci`，最终以 `bypass-fastfail` 标签合入；另外作者补充了 GB300 上固定 ISL 131072 长上下文 benchmark 的结果，但评论内容被截断，未展示完整数值。核心设计决策（DP 下复用全局同步 `is_extend_in_batch` 标志）在 PR body 中已有说明，未引发争议。

- CI 重跑与基准验证 (other): 无未解决的技术疑虑；功能开关默认关闭，CI 失败由 `bypass-fastfail` 放行。

风险与影响

- 风险:

1. 核心调度路径变更: `get_next_batch_to_run()` 是调度器每轮的核心决策路径, 新增的 `_should_defer_prefill()` 分支对所有非 DLLM 路径生效。默认值为 0 时行为完全不变, 但一旦开启且配置过大, `prefill` 吞吐会明显下降, 排队延迟上升。
2. 计数语义隐患: `_should_defer_prefill()` 在每个 scheduler 轮次无条件递减, 即使当前等待队列为空、根本没有可调度的 `prefill`, 也会消耗间隔计数; 这意味着“N 轮 decode 间隔”实际可能包含空转轮次, 与直觉略有偏差, 需要在实际负载中验证间隔的真实效果。
3. 对全局 flag 的耦合: DP 模式下依赖 `batch.is_extend_in_batch` 由 MLP-sync all-gather 同步的语义, 如果未来 DP attention 后端或 MLP-sync 机制变化, 全局标志的可靠性需要重新确认; 非 DP 路径与 DP 路径共用同一套计数逻辑, 但判定来源不同, 未来维护时需注意。
4. 测试覆盖局限: 所有测试均为单元级, 且作者说明开发 VM 的 base Python 环境缺少 `torch/numpy`, 无法本地收集 `pytest`; 缺少真实 DP 多 rank 场景下的端到端调度节奏验证, CI 也有失败记录, 靠 `bypass-fastfail` 放行。- 影响: 对用户: 提供新的调优旋钮 `--prefill-decode-interval`, 在 DP attention + 长上下文负载 (如 AgentX) 下可通过加大间隔换取 decode 交互性, 默认 0 时无任何行为变化。对系统: 仅在 scheduler 决策路径新增一个条件分支和一个计数状态, 不改变模型计算、采样或 KV cache 逻辑; 开启后可能以牺牲 `prefill` 吞吐为代价换取 decode 公平性。对团队: 新增 1 个服务参数、2 个调度器方法和 4 个单元测试, 维护成本低; 该参数与已有的 `prefill_delayer`、`enable_priority_scheduling` 等调度机制并列存在, 未来可能需要统一调度策略。整体影响范围为低到中, 默认关闭有效控制风险。- 风险标记: 核心调度路径变更, 默认关闭需显式开启, 复用 DP 全局同步 flag 有耦合, 间隔计数按轮次消耗含空转, 缺少多 rank 集成测试

关联脉络

- PR #35071 [PD] Overlap prefill DP-rank bootstrap queries: 同属 DP/ 多 rank 场景下的调度关键路径优化: 该 PR 在 decode 关键路径预取 DP-rank 查询以减少等待, 本 PR 则通过全局同步 EXTEND 标志统一 `prefill/decode` 节奏; 两者都服务于提升 DP attention 下的 decode 交互性, 但一个在 `disaggregation` 模块、一个在 `scheduler` 模块。