PR #35298 完整报告

sgl-project/sglang

[Fix] DCP: advertise the logical KV-event block size

合并时间：2026-08-19 11:55

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35298

执行摘要

- 一句话：修复 DCP 下 KV 事件块大小广告错误，路由命中率从 0 恢复
- 推荐动作：值得精读。三个亮点：一是 " 契约与实现漂移导致静默故障 " 的根因分析范式——命中率恰好为 0 且本地指标健康是强信号；二是 " 必要不充分校验 " 的取舍——讨论中确认 tp_size % dcp_size 既不必要（dist init 已兜底）也不充分（真正分组维度是 attn_tp_size），最终放弃；三是跨仓库协同修复同一契约问题的协作方式，可作为公共接口设计的参考。

功能与动机

PR body 明确指出问题的严重性与隐蔽性："KV events are chunked by the tree's page size, so they go out at the logical width. /server_info's kv_events descriptor advertised the physicalpage_size. A KV-aware router honoring the documented contract... hashes requests at a width no emitted block can ever match. Stores apply cleanly, every lookup misses, and the router's KV hit rate is silently exactly 0while all local metrics look healthy." 也就是说，DCP 下路由缓存命中率静默归零，但本地指标一切正常，属于典型的契约与实际实现漂移导致的隐蔽故障。

实现拆解

1. 新增逻辑宽度属性：在 python/sglang/srt/server_args.py 的 ServerArgs 上增加只读属性 kv_event_block_size，实现为 self.page_size * self.dcp_size。该公式必须与 mem_cache/kv_cache_configurator.py 中分配器构建页宽、mem_cache/kv_cache_builder.py 中 radix tree 继承页宽保持一致，避免三处定义漂移。
2. 修正描述器广告值：describe_kv_events_publisher() 返回的 block_size 从 page_size 改为 self.kv_event_block_size，并同步更新 docstring 中的 wire contract 说明；同时明确 dp_size 故意不按 DCP 缩放，因为 DCP 是在 rank 内分片而非新增 publisher。该函数还保留了原先的若干防御性返回 None 分支（空配置、非法 page_size、协议非 tcp、端口非法等），防止占位 block_size 再次造成静默未命中。
3. 放弃多余的参数校验：早期版本在 _handle_dcp_validation 中加入 tp_size % dcp_size 的 guard，review 中发现真正需要校验的是随后才解析的 attn_tp_size，且 parallel_state.py 在 dist init 时已抛出 RuntimeError("tensor_model_parallel_size (N) must be divisible by decode_context_parallel_size (M)")，因此最终移除该 guard，避免重复且不充分的检查。

4. 测试配套：在 test/registered/unit/server_args/test_server_args.py 新增 TestDcpKvEventContract，覆盖 DCP（page_size=64, dcp_size=4 期望 block_size=256）与非 DCP（期望保持 64）两条路径，并覆盖 page_size=1 + dcp_size 的边界形态（分配器仍按 dcp_size 分页）；测试在 main 上红、修复后绿。

关键文件：

- python/sglang/srt/server_args.py（模块 服务配置；类别 source；类型 core-logic；符号 kv_event_block_size）：核心修复文件：新增 kv_event_block_size 属性并修正 describe_kv_events_publisher 的广告值，是本次契约纠偏的入口。
- test/registered/unit/server_args/test_server_args.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestDcpKvEventContract, test_kv_events_descriptor_reports_logical_block_size, test_kv_event_block_size_widens_a_single_token_page）：新增 TestDcpKvEventContract 单测，锁死 DCP 与非 DCP 两条路径的 block_size 契约，防止回归。

关键符号：kv_event_block_size, describe_kv_events_publisher, test_kv_events_descriptor_reports_logical_block_size, test_kv_event_block_size_widens_a_single_token_page

关键源码片段

python/sglang/srt/server_args.py

核心修复文件：新增 kv_event_block_size 属性并修正 describe_kv_events_publisher 的广告值，是本次契约纠偏的入口。

```
    # server_args.py —— 新增逻辑宽度属性：DCP 下事件实际发出宽度
    @property
    def kv_event_block_size(self) -> int:
        """KV 事件的实际发出宽度：DCP 下 radix tree 以
        ``page_size * dcp_size`` 分页（见 mem_cache/kv_cache_builder.py），
        事件也按该逻辑宽度分块，因此必须与分配器公式保持一致。
        """
        return self.page_size * self.dcp_size


    def describe_kv_events_publisher(self) -> Optional[dict]:
        """构造 /server_info 中 kv_events 描述，供 KV-aware 路由器订阅。
        契约关键约束：subscribers MUST hash prompts at block_size。
        """
        # 惰性导入，避免加载 server_args 时顶层引入 disaggregation / zmq
        from sglang.srt.disaggregation.kv_events import KVEventsConfig

        raw = self.kv_events_config
        page_size = self.page_size
        # page_size 非法时返回 None：占位 block_size 会让路由器按错误粒度
        # 哈希，造成静默 KV 未命中，宁可明确告知无 publisher
        if not raw or page_size is None or page_size <= 0:
```

```
            return None
        try:
            cfg = KVEventsConfig.from_cli(raw)
        except Exception:
            # 配置损坏时服务器启动已失败，此处保持 /server_info 可访问
            return None
        if cfg.publisher == "null" or not cfg.endpoint:
            return None
        if not cfg.endpoint.startswith("tcp://"):
            return None

        # 端点 host/port 解析过程略去，逻辑与
        # ZmqEventPublisher.offset_endpoint_port 对齐

        return {
            "publisher": cfg.publisher,
            "endpoint_host": host,
            "endpoint_port_base": port,
            "topic": cfg.topic,
            # 关键修正：原实现广告物理 page_size，DCP 下路由器哈希宽度
            # 与任何实际发出的 block 都不匹配，命中率静默为 0
            "block_size": self.kv_event_block_size,
            # dp_size 故意不乘 dcp_size：DCP 在 rank 内分片，不新增 publisher
            "dp_size": self.dp_size,
        }
```

test/registered/unit/server_args/test_server_args.py

新增 TestDcpKvEventContract 单测，锁死 DCP 与非 DCP 两条路径的 block_size 契约，防止回归。

```
class TestDcpKvEventContract(CustomTestCase):
    """DCP 将 radix tree 页宽扩为 page_size * dcp_size，广告的
    KV 事件块大小必须反映该逻辑宽度。"""

    KV_EVENTS = '{"publisher":"zmq","topic":"kv","endpoint":"tcp://*:5557"}'

    def test_kv_events_descriptor_reports_logical_block_size(self):
        # 广告物理 page_size 会让路由器按错误宽度哈希，命中率静默归零
        args = ServerArgs(
            model_path="dummy",
            tp_size=4,
            dcp_size=4,
            page_size=64,
            kv_events_config=self.KV_EVENTS,
        )
        # DCP 下逻辑宽度为 64 * 4 = 256
        self.assertEqual(args.describe_kv_events_publisher()["block_size"], 256)
        # 非 DCP 场景宽度保持不变，行为不受影响
        args = ServerArgs(
```

```
            model_path="dummy", page_size=64, kv_events_config=self.KV_EVENTS
        )
        self.assertEqual(args.describe_kv_events_publisher()["block_size"], 64)

    def test_kv_event_block_size_widens_a_single_token_page(self):
        # page_size=1 + DCP 也是真实部署形态：分配器仍按 dcp_size 分页
        args = ServerArgs(model_path="dummy", tp_size=8, dcp_size=8, page_size=1)
        self.assertEqual(args.kv_event_block_size, 8)
```

评论区精华

Review 中的核心交锋集中在三处：

1. 参数校验的必要性与充分性：alexnails 质疑 tp_size % dcp_size 检查 "really this is that tp_size split by the number of attention heads, does this matter here?"——因为真正参与分组的其实是 attn_tp_size（在 handler 运行后才解析），且 parallel_state.py 已有 dist init 校验。作者确认后移除 guard，并在 PR body 中感谢 alexnails 的质疑。
2. API 形态：alexnails 询问 kv_event_block_size 是否应使用 @property，作者采纳并改为 property。
3. 文档风格：kpham-sgl 建议精简新属性的 docstring，作者按要求完成。

- tp_size % dcp_size 参数校验是否必要且充分 (design): 作者确认 parallel_state.py 在 dist init 时会抛出 RuntimeError（tp_size 必须整除 dcp_size），该 guard 既非必要也非充分，最终移除。
- kv_event_block_size 的 API 形态 (style): 作者改为 @property，作为派生自 page_size 与 dcp_size 的只读计算属性。
- 新属性 docstring 精简 (documentation): 作者按要求精简，保留核心信息（宽度公式与出处）。

风险与影响

- 风险：
  1. 非 DCP 场景：dcp_size=1 时 kv_event_block_size == page_size，行为完全不变，回归风险极低。
  2. 公式耦合风险：kv_event_block_size = page_size * dcp_size 硬编码在 server_args.py，若未来分配器或 radix tree 的页宽公式变化（如引入显存对齐、分页因子调整），需要同步修改，否则会再次漂移；建议长期将该公式抽取为单一来源。
  3. 下游订阅方同步：sgl-router 的 BlockSizeOracle 与 Dynamo 集成需要拉取新的逻辑宽度；旧版本订阅方若已按物理宽度缓存，本次变更会改变其行为——但按 PR body 证据，修复前命中率本就是 0，实际无存量损失。
  4. 测试范围：单测只覆盖描述器返回值，未自动验证真实 ZMQ wire 上 BlockStored.block_size 与描述器一致；不过 PR body 提供了 4x H200 的端到端实测数据作为补充证据。
- 影响：

1. 对用户 / 系统：DCP 部署下 KV-aware 路由（sgl-router、Dynamo）的命中率从静默 0 恢复至理论上限（实测 6 个 1686-token 提示词从 0/6 恢复到 4.2857/6），直接提升缓存复用与吞吐；同时揭穿了 " 本地指标正常 " 的假象。
2. 对团队：为 KV 事件契约补充了文档化说明（dp_size 为何不缩放、block_size 的语义），降低后续订阅方误用概率；Dynamo 侧同步修复（ai-dynamo/dynamo#12765）表明该契约已成为跨仓库公共接口。
3. 影响范围：仅作用于 /server_info 描述器和新增属性，不触碰分配器、radix tree 与 ZMQ 发布逻辑，回归面小。 - 风险标记：DCP 路由命中率回归，页宽公式跨文件耦合，下游订阅方需同步，依赖既有 dist init 校验

关联脉络

- PR #12765 fix(sglang): use DCP-aware KV event block size: ai-dynamo/dynamo 仓库对同一问题的消费端修复：Dynamo 集成直接读取 server_args.page_size 导致事件转换器拒绝 DCP 块。本 PR 修复引擎侧广告契约，使任何遵守契约的订阅方无需自带 DCP 感知。