

PR #26245 完整报告

sgl-project/sglang

Support DP-aware PD router dispatch

合并时间: 2026-06-25 06:02

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/26245>

执行摘要

此 PR 修复了 `PDRouter` 在 DP-aware 模式下使用虚拟 `url@rank` 作为真实 HTTP 目标的问题，通过统一使用物理 URL、在请求中注入 DP rank 等改造，显著提升 DP-aware 部署的稳定性和性能。核心变更集中在 `pd_router.rs` 和 `worker.rs`，包含新增请求准备函数和 URL 解析工具，并同步重构了 `worker_builder.rs`。

功能与动机

在 `--dp-aware` 模式下，`PDRouter` 选中的 worker 后直接向 `worker.url()` 转发原始请求，该 URL 包含虚拟后缀 `@rank`（如 `http://10.66.4.202:28714@0`）。这导致两个问题：

- `prefill` 请求未收到 `data_parallel_rank`，无法感知自身 DP rank；
- `decode` 请求无法获知 `prefill` 的 DP rank，只能回退到猜测逻辑（`bootstrap_room % dp_size` 或 `/query_dp_ranks`）；
- 部分路径（如 `/health_generate`、`metadata proxy`）直接将 `url@rank` 作为真实 HTTP 目标，导致请求转发到非法地址。

正如 PR body 所述："This made PD routing effectively worker-aware but not DP-rank-aware, and left DP-aware virtual URLs exposed to HTTP dispatch paths."

实现拆解

1. 新增 `PreparedWorkerRequest` 结构体（`pd_router.rs`）：封装 `endpoint_url` 和 `body`，统一请求准备输出。
2. 引入 `worker_endpoint_url` 方法：通过 `worker.base_url()` 构建 HTTP 端点，替代直接使用 `worker.url()`。对所有 worker 统一使用物理 URL。
3. 实现 `prepare_worker_request` 方法：对 DP-aware worker 调用 `worker.prepare_request()` 在请求体注入 `data_parallel_rank`；非 DP 模式保持原请求。
4. 增加 `inject_prefill_dp_rank_for_decode` 方法：在 `decode` 请求中插入 `disagg_prefill_dp_rank` 字段，携带 `prefill` 的 DP rank，使 `decode` 能精确获取预填充的 KV 缓存来源。
5. 整合 `prepare_pd_worker_requests` 入口：组合以上步骤，替换原有离散请求构建逻辑，统一处理 `prefill` 和 `decode` 请求。
6. 提取 `parse_bootstrap_host_from_url` 函数（`worker.rs`）：剥离 URL 中的 `@rank` 后缀，解析真实 bootstrap host；`DPAwareWorker` 新增 `bootstrap_host` 字段，避免虚拟 URL

泄露。

7. 重构 `worker_builder.rs`: 使用新函数替代内联 URL 解析, 删除 24 行重复代码。

`sgl-model-gateway/src/routers/http/pd_router.rs`

核心路由文件, 实现 DP-aware 请求准备和 rank 注入, 改动最大 (+203/-20)。

```
// 在 decode 请求中注入 prefill 的 DP rank, 确保 decode 知道
// 应从哪个 DP rank 的 KV 缓存开始引导, 避免回退到
// /query_dp_ranks 端点
fn inject_prefill_dp_rank_for_decode<'a>(
    decode_request: Cow<'a, Value>,
    prefill_worker: &dyn Worker,
) -> Result<Cow<'a, Value>, String> {
    // 若 prefill worker 没有 DP rank (非 DP-aware), 则无需注入
    let Some(prefill_dp_rank) = prefill_worker.dp_rank() else {
        return Ok(decode_request);
    };

    let mut decode_request = decode_request.into_owned();
    let Some(obj) = decode_request.as_object_mut() else {
        return Err(
            "Failed to insert disagg_prefill_dp_rank because request body is not an object"
                .to_string(),
        );
    };

    // 插入 disagg_prefill_dp_rank 字段, 值为 prefill 的 DP rank
    obj.insert(
        Self::DISAGG_PREFILL_DP_RANK_KEY.to_string(),
        Value::from(prefill_dp_rank as u64),
    );
    Ok(Cow::Owned(decode_request))
}
```

`sgl-model-gateway/src/core/worker.rs`

新增 `parse_bootstrap_host_from_url` 函数和 `DPAwareWorker` 的 `bootstrap_host` 字段, 确保正确解析物理 host。

```
/// 从 worker URL 中解析真实的 bootstrap host, 支持 dp rank 后缀 (如
/// http://10.66.5.115:20664@3) 和直接 URL。若 URL 格式无效则默认返回
/// "localhost"。
pub(crate) fn parse_bootstrap_host_from_url(url: &str) -> String {
    // 尝试剥离 @rank 后缀, 仅当后缀可解析为整数时认为有效
    let metadata_url = match url.rsplit_once('@') {
        Some((base_url, rank)) if rank.parse::<usize>().is_ok() => base_url,
        _ => url,
    };

    // 使用 url crate 解析 host, 失败时添加 http:// 前缀重试,
```

```

// 最终 fallback 为 localhost
match url::Url::parse(metadata_url) {
    Ok(parsed) => parsed.host_str().unwrap_or("localhost").to_string(),
    Err(_) if !metadata_url.contains(":/") => {
        match url::Url::parse(&format!("http://{}", metadata_url)) {
            Ok(parsed) => parsed.host_str().unwrap_or("localhost").to_string(),
            Err(_) => {
                tracing::warn!(
                    "Failed to parse URL '{}', defaulting to localhost",
                    metadata_url
                );
                "localhost".to_string()
            }
        }
    }
}
Err(_) => {
    tracing::warn!(
        "Failed to parse URL '{}', defaulting to localhost",
        metadata_url
    );
    "localhost".to_string()
}
}
}

```

评论区精华

该 PR 由 [xiaoxiaohehe001](#) 直接批准，无额外 review 讨论。PR body 中提供了 GB300 DEP4/DEP8 的 A/B 验证数据，引用原文：

- DP-aware 模式：cache hit 94.4%，TTFT p50/p95 1.11s / 5.38s，吞吐 1110.9 tok/s，2 个错误 / 3908 测量。
- 非 DP-aware 基线：cache hit 74.5%，TTFT p50/p95 4.22s / 10.00s，吞吐 937.3 tok/s，9 个错误 / 3519 测量。

这些数据直观展示了 DP-rank 感知路由带来的显著增强。

风险与影响

- 风险：核心路由路径调整，若非 DP-aware 模式存在未覆盖分支可能导致回归；URL 解析依赖 @rank 后缀格式，未来变更需保持兼容；缺少端到端集成测试，新代码在复杂 topology 下可能暴露问题。但当前代码在非 DP 分支保持原逻辑，且单元测试覆盖了多种 URL 格式。
- 影响：DP-aware 用户将获得显著更低的延迟和更稳定的请求处理；系统消除了虚拟 URL 暴露的安全风险；团队获得更清晰的代码结构和可维护的 URL 解析工具。

关联脉络

此 PR 关联 Issue #26066，并吸收了 #26237 (Fix DP-aware PD router worker URLs) 的修复范围。在此基础上，本 PR 额外实现了 prefill DP rank 向 decode 的传播（

disagg_prefill_dp_rank)，使得 decode 不需要从 bootstrap_room % dp_size 猜测或调用 /query_dp_ranks 端点，进一步提升了路由精确度。结合近期路由相关的优化（如 #29075 结果 D2H 拷贝优化），可见 SGLang 团队正在系统性地提升 PD 分离部署的可靠性和效率。