

PR #28742 完整报告

sgl-project/sglang

[router] Raise chat body cap to 5 MiB for long contexts

合并时间: 2026-06-20 06:50

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28742>

执行摘要

此 PR 将 sgl-router 的 `/v1/chat/completions` 请求体大小上限从 1 MiB 提升到 5 MiB，以支持约百万 token 的长上下文请求。同时扩展了 `overlap_blocks` 直方图桶边界至 8000，并更新相应测试。变更集中、无 review 争议，属于对路由器功能的合理增强。

功能与动机

实验性路由器此前将请求体限制在 1 MiB，但对于约百万 token 的上下文，JSON 编码后超过 1 MiB，导致合法长上下文请求被拒绝（413）。该 PR 将限制提升至 5 MiB，覆盖 ~1M token 上下文，同时仍通过 `DefaultBodyLimit` layer 防止恶意客户端造成过大内存消耗。

实现拆解

1. 提升请求体上限：在 `experimental/sgl-router/src/server/routes/chat.rs` 中，将常量 `MAX_CHAT_BODY_BYTES` 从 `1 << 20` (1 MiB) 改为 `5 << 20` (5 MiB)。该限制由 axum 的 `DefaultBodyLimit` layer 在 handler 执行前检查，返回 413 状态码。
2. 扩展直方图桶边界：在 `experimental/sgl-router/src/server/metrics.rs` 中，`OVERLAP_BLOCKS_BUCKETS` 从最高 1000 扩展到 2000、4000、8000，避免长上下文下高 overlap block 数量全部落入 `+Inf` 桶。文档注释也相应更新，引用 `MAX_CHAT_BODY_BYTES` 而非绝对 token 数以避免未来漂移。
3. 更新单元测试：在 `experimental/sgl-router/tests/proxy/chat_routing.rs` 中，`oversized_request_body_returns_413` 测试改用 `MAX_CHAT_BODY_BYTES + 1` 替代硬编码的 2 MiB，使测试自动跟踪配置变化。`histogram_plus_inf_bucket_catches_overflow` 测试的观测值从 1001 改为 8001，验证新桶边界。

`experimental/sgl-router/src/server/routes/chat.rs`

核心变更：将请求体上限从 1 MiB 提升到 5 MiB，更新文档注释。

```
/// Per-route body-size cap on `/v1/chat/completions`. 5 MiB accommodates a
/// long context — a ~1 M-token context tokenized as JSON fits under this —
/// while preventing a hostile client from forcing the router to
/// heap-allocate hundreds of MiB before forwarding. The cap is wired in
/// `crate::server::app::build_router` as a route-level `DefaultBodyLimit`
/// layer; axum's `Bytes` extractor enforces it and returns 413
/// PAYLOAD_TOO_LARGE before this handler runs.
pub const MAX_CHAT_BODY_BYTES: usize = 5 << 20; // 之前是 1 << 20 (1 MiB)
```

experimental/sgl-router/src/server/metrics.rs

扩展 overlap_blocks 直方图桶边界至 8000，保持长上下文场景的可观测性。更新文档注释和单元测试。

```
/// Histogram bucket upper bounds for `sgl_router_overlap_blocks`. Blocks are
/// 32–64 tokens each, and the `MAX_CHAT_BODY_BYTES` cap bounds context length —
/// putting the practical ceiling for a maximum-length context in the low tens
/// of thousands of blocks. The ladder spans 0 → ~8k blocks at the resolution
/// worth charting; the `+Inf` bucket catches the longer-context tail beyond
/// 8000.
const OVERLAP_BLOCKS_BUCKETS: &[f64] = &[
    // 桶边界从 1000 扩展到 8000，新增 2000, 4000, 8000
    0.0, 1.0, 2.0, 4.0, 8.0, 16.0, 32.0, 64.0, 128.0, 256.0, 512.0, 1000.0, 2000.0, 4000.0,
    8000.0,
];
```

评论区精华

无人工 review 评论，仅 Gemini Code Assist 自动评论确认变更合理且测试通过。

风险与影响

- 风险：5 MiB 的请求体仍会被完整缓冲到内存中，但该上限足以防止 OOM。更大的请求体可能轻微增加序列化开销，但路由器仅做轻量 probe 解析，影响可忽略。
- 影响：用户可使用长上下文请求（如文档分析、长对话），消除了之前 1 MiB 限制导致的拒绝。团队需确保 worker 端也能处理更长上下文。

关联脉络

此次 PR 与之前的路由器优化 PR（如 PR#28717 对齐 TTFT 桶、PR#28744 入口 tokenize）形成系列改进，表明路由器正在持续优化以支持更长上下文和更高性能。