PR #24330 完整报告

sgl-project/sglang

fix(router): configure HTTP client connection settings

合并时间：2026-05-08 02:42

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24330

执行摘要

本 PR 为 SGLang Model Gateway 的 HTTP 客户端连接设置（连接超时、空闲连接池大小、TCP keepalive）添加了 CLI 和环境变量配置能力，延续 #24329 的模式。所有新选项均保留现有默认值，完全向后兼容。合并后运维人员无需修改源码即可针对不同部署环境调整连接行为。

功能与动机

根据 PR 描述，这是 #24329 的后续，目标是“Make SGLang Model Gateway's remaining upstream HTTP client connection settings configurable”。之前只有 pool_idle_timeout 可配置，而 connect_timeout、pool_max_idle_per_host、tcp_keepalive 仍为硬编码。本 PR 将它们暴露为可配置项，使网关更适应多种网络环境（如高延迟、高并发）。

实现拆解

1. 数据模型扩展：在 config/types.rs 中定义三个默认常量（连接超时 10s、空闲池每主机 500、TCP keepalive 30s），在 RouterConfig 中加入对应字段，并通过 #[serde(default = "...")] 保证反序列化时兼容。
2. 构建器方法：在 config/builder.rs 中添加三个 setter 方法，沿用已有链式调用风格。
3. CLI 参数与环境变量：在 main.rs 中以 --connect-timeout-secs 等形式添加参数，支持 SMG_CONNECT_TIMEOUT_SECS 等环境变量，并在构建 RouterConfig 时注入。
4. 验证增强：在 config/validation.rs 中增加校验，确保 connect_timeout_secs 和 tcp_keepalive_secs 不为零。
5. 生效点替换：在 app_context.rs 中，将 Client::builder() 的硬编码参数（connect_timeout(Duration::from_secs(10)) 等）替换为配置值。
6. 文档同步：更新 mdx 和 md 两套文档，在 HTTP Client 配置表中添加新选项说明。
7. 测试覆盖：在 types.rs 的模块测试中添加反序列化默认值测试，确保未提供配置时回退正确。

sgl-model-gateway/src/config/types.rs

核心变更文件：定义新常量、结构体字段、默认值函数和反序列化测试，构成数据契约。

```
// sgl-model-gateway/src/config/types.rs（关键片段）

use serde::{Deserialize, Serialize};
```

```rust
// 默认常量：空闲超时 50s（已有），连接超时 10s，每主机最大空闲 500，TCP keepalive 30s
pub const DEFAULT_POOL_IDLE_TIMEOUT_SECS: u64 = 50;
pub const DEFAULT_CONNECT_TIMEOUT_SECS: u64 = 10;
pub const DEFAULT_POOL_MAX_IDLE_PER_HOST: usize = 500;
pub const DEFAULT_TCP_KEEPALIVE_SECS: u64 = 30;

/// 主路由配置
#[derive(Debug, Clone, Serialize, Deserialize)]
pub struct RouterConfig {
    // ... 其他字段 ...
    #[serde(default = "default_pool_idle_timeout_secs")]
    pub pool_idle_timeout_secs: u64,

    // 新增三个 HTTP 客户端连接设置，均提供默认值函数
    #[serde(default = "default_connect_timeout_secs")]
    pub connect_timeout_secs: u64,
    #[serde(default = "default_pool_max_idle_per_host")]
    pub pool_max_idle_per_host: usize,
    #[serde(default = "default_tcp_keepalive_secs")]
    pub tcp_keepalive_secs: u64,
    // ...
}

// 默认值函数
fn default_connect_timeout_secs() -> u64 { DEFAULT_CONNECT_TIMEOUT_SECS }
fn default_pool_max_idle_per_host() -> usize { DEFAULT_POOL_MAX_IDLE_PER_HOST }
fn default_tcp_keepalive_secs() -> u64 { DEFAULT_TCP_KEEPALIVE_SECS }

// 测试：字段未提供时应回退到默认值
#[cfg(test)]
mod tests {
    use super::*;

    #[test]
    fn test_router_config_http_client_deserialization_defaults() {
        let config = RouterConfig::default();
        let mut json = serde_json::to_value(&config).unwrap();
        let obj = json.as_object_mut().unwrap();
        // 移除所有 HTTP 客户端相关字段，模拟未提供
        obj.remove("pool_idle_timeout_secs");
        obj.remove("connect_timeout_secs");
        obj.remove("pool_max_idle_per_host");
        obj.remove("tcp_keepalive_secs");
        let deserialized: RouterConfig = serde_json::from_value(json).unwrap();
        // 断言回退到默认值
        assert_eq!(deserialized.connect_timeout_secs, DEFAULT_CONNECT_TIMEOUT_SECS);
        assert_eq!(deserialized.pool_max_idle_per_host, DEFAULT_POOL_MAX_IDLE_PER_HOST);
        assert_eq!(deserialized.tcp_keepalive_secs, DEFAULT_TCP_KEEPALIVE_SECS);
    }
```

```
}
```

sgl-model-gateway/src/main.rs

添加 CLI 参数和环境变量，透传新配置到 builder。

```
// sgl-model-gateway/src/main.rs (CLI 参数定义 )
// ==================== HTTP Client ====================
/// 连接池空闲超时（秒）
#[arg(long, env = "SMG_POOL_IDLE_TIMEOUT_SECS", default_value_t = DEFAULT_POOL_IDLE_
TIMEOUT_SECS, help_heading = "HTTP Client")]
pool_idle_timeout_secs: u64,

/// 新建上游 HTTP 连接超时（秒）
#[arg(long, env = "SMG_CONNECT_TIMEOUT_SECS", default_value_t = DEFAULT_CONNECT_
TIMEOUT_SECS, help_heading = "HTTP Client")]
connect_timeout_secs: u64,

/// 每主机最大空闲连接数
#[arg(long, env = "SMG_POOL_MAX_IDLE_PER_HOST", default_value_t = DEFAULT_POOL_MAX_
IDLE_PER_HOST, help_heading = "HTTP Client")]
pool_max_idle_per_host: usize,

/// TCP keepalive 空闲时间（秒）
#[arg(long, env = "SMG_TCP_KEEPALIVE_SECS", default_value_t = DEFAULT_TCP_KEEPALIVE_
SECS, help_heading = "HTTP Client")]
tcp_keepalive_secs: u64,
```

评论区精华

> Gemini Code Assist: While you're making other HTTP client settings configurable, it would be a good idea to also make pool_idle_timeout configurable. It's currently hardcoded to 50 seconds. This would improve consistency and give operators more control over connection pool behavior.
>
> revanthreddy-hai（作者）: pool_idle_timeout is handled separately in #24329. This PR is intentionally scoped to the remaining reqwest HTTP client settings: connect_timeout, pool_max_idle_per_host, and tcp_keepalive.

该讨论凸显了 PR 的范围意识：作者明确区分 #24329 与本 PR 的边界，避免一次性改动过大。评审人未再反对，PR 合并。

风险与影响

- 风险：极低。所有新增字段都有与旧硬编码相同的默认值，现有配置即使完全移除新字段也会回退到合理默认值。验证层阻止了 0 值误设。
- 影响：仅限 Model Gateway 组件。运维人员现在可以通过 CLI 或环境变量精细控制上游连接行为，尤其有利于优化高并发代理场景。默认行为不变，现有用户无感知迁移成本。

关联脉络

本 PR 是 #24329 的直系后续，共同完成了 Model Gateway HTTP 客户端连接池核心参数的配置化。结合近期 #24555（LoRA 多节点一致性）和 #24329 等 PR，可见团队正系统性地提升网关的可运维性。未来可能继续推进连接重试、健康检查等参数的配置化。