PR #24329 完整报告

sgl-project/sglang

fix(router): make HTTP pool idle timeout configurable

合并时间：2026-05-07 13:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24329

执行摘要

此 PR 为 SGLang Model Gateway 添加了上游 HTTP 连接池空闲超时的可配置支持，新增 --pool-idle-timeout-secs 参数与环境变量 SMG_POOL_IDLE_TIMEOUT_SECS，默认值为 50 秒。变更向后兼容，配套更新了构建器、应用上下文及多份文档。

功能与动机

根据 PR 描述，原本上游 HTTP 连接池空闲超时硬编码为 50 秒，无法在部署环境中与各类网关（如 ingress-nginx、Envoy、ALB 等）的 keep-alive / idle-timeout 设置对齐。此变更允许运维人员通过 CLI 参数或环境变量自由调整，简化了与基础设施的超时匹配。

实现拆解

1. 配置类型扩展：在 sgl-model-gateway/src/config/types.rs 中定义常量和默认值函数，RouterConfig 新增 pool_idle_timeout_secs 字段，并使用 #[serde(default)] 保持反序列化向后兼容。
2. 构建器方法：在 src/config/builder.rs 增加 pool_idle_timeout_secs 设置方法，支持编程式配置。
3. CLI 参数注册：在 src/main.rs 的 CliArgs 结构体新增 --pool-idle-timeout-secs，默认值引用常量，并传递给 RouterConfigBuilder。
4. 运行时消费：在 src/app_context.rs 的 with_client 方法中，将 reqwest Client::builder().pool_idle_timeout() 参数由固定 50 秒改为读取配置字段。
5. 文档同步：更新 docs_new 和 docs 两个文档目录下的功能说明，以及项目 README.md，添加参数表格。

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

核心配置类型变更，新增字段、默认值函数和单元测试，是整个配置可配置化的基础。

```
// 定义包级常量作为默认值唯一来源
pub const DEFAULT_POOL_IDLE_TIMEOUT_SECS: u64 = 50;

// RouterConfig 结构体新增字段
#[derive(Debug, Clone, Serialize, Deserialize)]
pub struct RouterConfig {
    // ... 其他字段 ...
    /// 连接池空闲超时秒数，默认 50 秒
```

```
    #[serde(default = "default_pool_idle_timeout_secs")]
    pub pool_idle_timeout_secs: u64,
    // ...
}

// 默认值工厂函数
fn default_pool_idle_timeout_secs() -> u64 {
    DEFAULT_POOL_IDLE_TIMEOUT_SECS
}

// 测试反序列化默认值
#[test]
fn test_router_config_pool_idle_timeout_deserialization_default() {
    let config = RouterConfig::default();
    let mut json = serde_json::to_value(&config).unwrap();
    json.as_object_mut().unwrap().remove("pool_idle_timeout_secs");
    let deserialized: RouterConfig = serde_json::from_value(json).unwrap();
    assert_eq!(deserialized.pool_idle_timeout_secs, default_pool_idle_timeout_secs());
}
```

sgl-model-gateway/src/app_context.rs

运行时消费配置，将参数应用到实际的 HTTP 客户端构建。

```
fn with_client(mut self, config: &RouterConfig, timeout_secs: u64) -> Result<Self, String> {
    let has_tls_config = /* ... */;

    let mut client_builder = Client::builder()
        // 使用配置化的空闲超时，而非硬编码 50
        .pool_idle_timeout(Some(Duration::from_secs(config.pool_idle_timeout_secs)))
        .pool_max_idle_per_host(500)
        .timeout(Duration::from_secs(timeout_secs))
        .connect_timeout(Duration::from_secs(10))
        .tcp_nodelay(true)
        .tcp_keepalive(Some(Duration::from_secs(30)));

    // ... TLS 配置 ...
}
```

评论区精华

- gemini-code-assist[bot]指出 types.rs 和 main.rs 中都出现了魔法数字 50，建议抽取为公共常量。作者采纳并实现，PR 最终获得批准。该讨论体现了代码规范的良好实践。

风险与影响

风险：整体风险低。新字段通过 #[serde(default)] 确保向后兼容，默认值与旧行为一致。潜在风险在于运维配置不当导致连接过早关闭或资源占用过高，可通过文档提示规避。无安全影响。

影响：仅影响 SGLang Model Gateway 组件。部署人员可获得新的调整手段，与上游基础设施超时对齐，减少不必要的连接重建。代码层面减少了魔法数字，提升可维护性。

关联脉络

本 PR 为 model-gateway 模块的独立改进，与近期其他 PR（如 observability 增强、KV cache 优化等）无直接关联。但展示了 SGLang 项目逐步将硬编码参数转化为可配置项的趋势。后续可能继续为其他连接参数（如连接超时、最大空闲连接数）提供类似配置化支持。