PR #21320 完整报告

sgl-project/sglang

feat: add --strict-ports option for predictable port assignment

合并时间：2026-03-27 16:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21320

执行摘要

此 PR 引入了 --strict-ports 选项，使服务器在 HTTP、scheduler 和 master 端口被占用时立即失败，而非自动选择其他端口，解决了端口分配不确定性问题，是一个向后兼容的功能增强，提升了配置可预测性。

功能与动机

当前，当请求端口被占用时，settle_port 方法会自动选择可用端口，这导致用户无法预知服务器最终使用哪个端口，客户端配置不可靠，且多个服务器实例间易产生端口冲突。引用 PR body 中描述："users who need to know exactly which ports will be used"，此 PR 旨在通过添加 --strict-ports 选项，确保端口分配的确定性，满足需要精确端口配置的使用场景。

实现拆解

主要改动集中在 python/sglang/multimodal_gen/runtime/server_args.py 文件中：

- 添加字段：在 ServerArgs 类中添加 strict_ports: bool = False 字段，作为默认关闭的严格端口模式开关。
- 修改逻辑：在 _adjust_network_ports 方法中，根据 strict_ports 值分支处理：
  - 启用时：使用 is_port_available 检查端口可用性，若不可用则抛出 RuntimeError，包含明确的错误提示。python if not is_port_available(self.port): raise RuntimeError(f"Port {self.port} is unavailable and --strict-ports is enabled...")
- 禁用时：保持原有自动端口选择逻辑（调用 settle_port）。
- CLI 集成：在 add_cli_args 方法中添加 --strict-ports 参数，使用 StoreBoolean action，便于命令行使用。

另一个文件 python/sglang/multimodal_gen/registry.py 的改动最初是为了优化错误处理，但根据 review 讨论，为避免隐藏配置问题，最终被 revert 回抛出 RuntimeError，与 strict-ports 功能无关。

评论区精华

review 中，gemini-code-assist[bot] 提出了两个关键讨论点：

1. 正确性担忧：关于 registry.py 的错误处理变更，评论指出"Changing RuntimeError to logger.debug and returning None can hide potential configuration issues"，建议回退

到异常以快速失败。这被采纳，在后续 commit 中 revert。

2. 设计改进：关于 strict-ports 错误消息，评论建议“The error messages for unavailable scheduler_port and master_port are not as helpful”，要求统一所有端口错误消息，包含解决建议。这导致错误消息被一致化，提升了用户体验。

风险与影响

风险分析：

- 启用 --strict-ports 后，端口被占用时服务器启动失败率增加，但这正是功能设计目标，无额外性能或安全风险。
- 错误处理逻辑需确保错误消息清晰准确，避免用户混淆；审查显示已通过改进消息解决。
- 兼容性：默认行为不变，现有用户不受影响，新增选项可选使用。

影响评估：

- 用户：获得更可控的端口配置能力，便于部署和调试，尤其适用于多实例环境。
- 系统：新增选项轻量级，不干扰核心功能，测试计划覆盖正常和失败场景，确保可靠性。
- 团队：需了解新选项以有效利用，代码变更集中在单个文件，维护成本低。

关联脉络

- 此 PR 在 body 中提及“Related to #21284”，但未在提供的历史 PR 列表中，具体关联未知。
- 与近期其他 PR（如 #21435 安全绑定端口）无直接功能关联，但同属基础设施改进范畴，体现了仓库对网络配置的持续优化趋势。
- 独立性强，作为独立功能添加，不影响其他模块。