

PR #34097 完整报告

sgl-project/sglang

docs(skill): record where config is read now that the seed is off limits

合并时间: 2026-08-10 05:46

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34097>

执行摘要

- 一句话: 技能文档落定 config 种子禁读规则与分形状读取入口
- 推荐动作: 值得精读, 尤其关注三点: 1) 分形状读取的决策映射 (bag / get_parallel() / 具名访问器 / configured_*_size() / runner stamp), 这是理解 SGLang 配置架构的钥匙; 2) 读棘轮测试 (test_global_config_read_ratchet.py) 如何用 AST 收集器把架构约束钉死, 以及词法作用域遮蔽的处理; 3) “文档评审反哺代码质量”的流程——纯文档 PR 的评审催生了 sarvam_moe.py 与收集器的真实修复。若只关心运行时行为, 可跳过。

功能与动机

PR body 说明: 原 tier 表把 `get_server_args()` 描述为 “debugging, dumps, per-runner fork copies”, 但 reader-convergence 之后它同时是业务代码“永远不得读取字段”的种子——“it is now also the thing business code must never read a field off”。文档需要记录每种配置形态现在该去哪里读: resolved leaf 到其 bag、live topology 到 `get_parallel()`、由实例推导的值到 `runtime_context` 具名访问器、被遮蔽的配置尺寸到 `configured_*_size()`、runner 解析值到 runner。同时修正两处过期引用: 已被删除的 writer ratchet 与 `declare_load_time_override`。另因 #34086 因 force-push 事故被 GitHub 误关 (“a bad force-push on my side briefly pointed several member branches at the same commit”), 本 PR 取代它继续承载同一份内容与原评审线程。

实现拆解

1. 重写 tier 总览表: `get_server_args()` 一行从“调试、dump、per-runner fork 副本”改为“启动记录 + 业务代码禁读”, 并标注读棘轮 (read ratchet) 把 `get_server_args().field`、别名绑定、`getattr(..., "field")` 等可识别形式钉死在 0; 运行时计算的名字与整对象交接为构造性豁免。这是全文的纲领, 后续所有读取指引都以此为前提。
2. 新增 “Reading config: the seed is off limits” 决策映射: 按数据形状给出五类入口——已解析叶节点走命名空间 bag (`get_exec()`、`get_schedule()` 等)、实时拓扑走 `get_parallel()`、由发布叶节点推导的值走 `runtime_context` 具名访问器 (如 `attention_backends()` 从 `get_exec().kernel` 派生, 保留 post-publish override 可见性)、被 live 属性遮蔽的配置尺寸走 `configured_{tp,pp,moe_dp,attn_cp}_size()` (DCP 无 `configured` 访问器, 未装组时 `attn_dcp_size/dcp_enabled` 安全返回 1/False 但只报有效拓扑)、runner 解析值走 runner 属性或构造参数。

3. 重写 “Reads that legitimately stay on a ServerArgs instance”: 删除 “per-runner fork 字段” 段落 (draft-worker 配置深拷贝已不存在, 所有 worker 共享同一发布实例); 保留 handed-instance 边界 (tokenizer 家族、entrypoints/、tokenizer 进程 MM 处理器、MMEncoder、GrammarManager); 新增测试双重规则 “a double publishes its config, it does not inject one on a stand-in”; 补上 ModelRunner._load_format_scope 有界 override 作为唯一受认可的对共享 bag 的临时写入。
4. 同步修正 `get_parallel()` 小节与 Guardrails: shadowed 尺寸的 config-intent 读取从 `server_args.X` 改指 `configured_*_size()`, 并说明 DCP 未装组时的安全降级语义; Guardrails 新增第 5 条全局 config 读棘轮 (基线 0、豁免按 owner module、_CONFIGURED_SIZE_CALL_SITES 登记表), 重写第 4 条使“决策的替代物”明确为 bag 叶节点 / 具名访问器 / runner stamp, 而非 `get_server_args().field`。
5. 配套与验证: 本 PR 无代码与测试改动 (单提交仅改 SKILL.md)。评审回复中声称的棘轮收集器扩展 (module-level/annotated/instance-attribute/ 条件绑定 / 参数名遮蔽、module-qualified 识别) 与 `sarvam_moe.py` 修复不在本 diff 内, 应位于 reader-convergence 配套栈的其他 PR, 基于可见材料无法最终确认其合并顺序。

关键文件:

- `.claude/skills/sclang-runtime-context/SKILL.md` (模块 技能文档; 类别 docs; 类型 documentation; 符号 `get_server_args`, `get_parallel`, `configured_tp_size`, `configured_pp_size`): 唯一变更文件: 重写 config 读取权威规范, 涵盖 tier 表、“Reading config: the seed is off limits” 决策映射、handed-instance 边界、测试双重规则与 Guardrails 读棘轮条目, 并修正 `get_parallel()` 小节等过期表述。

关键符号: `get_server_args`, `get_parallel`, `configured_tp_size`, `configured_pp_size`, `configured_moe_dp_size`, `configured_attn_cp_size`, `attention_backends`, `max_speculative_num_draft_tokens`, `preserve_config`

评论区精华

评审主体是作者自审 (ch-wan, 借 #34087 作为 stack review 载体) 与 Codex bot 的 P2 建议。核心交锋有三条线: 其一, 自审发现未修改的 `get_parallel()` 段落仍指引 `server_args.X`, 与新版 `configured_*_size()` 指引直接矛盾, 属于 blocker, 作者在 PR 内改写段落并交叉链接; 其二, Codex 连续指出多个“instance-only”访问器 (`attention_backends()`、`cuteds1_moe_max_num_tokens()`、`max_speculative_num_draft_tokens()`、`configured_*_size()`) 的实际输入都是已发布叶节点, 读种子会错过 post-publish override, 作者逐一确认并改为从 bags 派生; 其三, 读棘轮的形态覆盖边界成为拉锯战——module-level/annotated/ 属性挂载 / 条件绑定别名、整对象交接、调用点登记粒度, 作者的选择是“扩展收集器 + 修真实代码” (`sarvam_moe.py` 被实地修复) 而不是收窄声明。最终 fresh re-review 报 bugs: 0、suggestions: 3、nits: 3。

- 相邻 `get_parallel()` 段落仍指引 `server_args.X`, 与新规则自相矛盾 (correctness): 本 PR 内改写该段落指向 `configured_tp_size()/configured_pp_size()/configured_moe_dp_size()/configured_attn_cp_size()` (DCP 走 `get_parallel().attn_dcp_size/.dcp_enabled`), 并交叉链接 “Reading config” 节, 两处不再互相矛盾。

- `attention_backends()` 被归为 instance-only, 但输入全是 `exec.kernel` 叶节点 (correctness): ch-wan 确认属实: 改为从 `get_exec().kernel` 派生 (`kernel.prefill_attention_backend` or `base`, `kernel.decode_attention_backend` or `base`) , 文档移入 bag 派生组。
- `max_speculative_num_draft_tokens / cutedsl_moe_max_num_tokens` 读种子会拿到过期边界 (correctness): 两者均改为从 bags 派生 (`max_spec` 含 `adaptive` 分支) ; instance-only 组收缩为 `mamba_cache_chunk_size`、`uses_mla_backend`、`process_model_config`。
- `configured_size()` 实为 bag-backed, 文档原先描述相反 (correctness): `configuredsize()` 改为读 parallel bag 自身叶节点 (绕过 live property 遮蔽), 文档两处表述同步更正: 只有 bag 支撑的读取 (含 `configured*_size()`) 看到 `override`, 实例派生访问器看不到。
- 读棘轮对别名形态的覆盖盲区: module-level、属性挂载、条件绑定、改名导入 (correctness): 收集器按词法作用域扩展 (module-level/annotated/instance-attribute/ 嵌套语句, 参数名遮蔽按作用域处理), 并反向验证; `sarvam_moe.py` 实修 (改读 `attention_backends()` 与 `get_memory().disable_radix_cache`), 基线保持 0。alias 链接拷贝与改名导入未见修复回复, 可能仍有盲区。
- 整对象交接 `f(get_server_args())` 是否违反零读取基线 (question): 文档明示两类构造性豁免并点名为豁免: 运行时计算的名字 (`getattr(sa, name)`) 与整对象交接 (`f(get_server_args())` 视为调用者选择实例, `callee` 读属 `supplied-instance` 契约), `census` 工具负责审计这些形态。
- DCP 的 config-intent 读取指引: 无 `configured` 访问器 (correctness): 文档改为: 当前不存在 config-intent DCP 调用点 (访问器随上游 #33925 删除); `live` 成员未装组时安全返回 `1/False` 但报有效拓扑; 未来需要 `requested size` 须自建 `configured_dcp_size()` 并登记调用点。
- “每个调用点都登记”的保证过强 (testing): 收集器补上 `module-qualified` 调用识别; 文档明确 (`file`, `accessor`) 粒度与“抓新文件 / 新访问器”的真实保证, 并说明不做逐行追踪是刻意取舍 (避免 `map` 因换行抖动) 。
- `MMEncoder` 的 per-worker 配置豁免理由不成立 (design): 重写为 `construction-path` 约定而非语义豁免: `self.server_args` 读取今天与 bags 一致只因是同一对象, `post-publish override` 不会到达它; per-runner 段落同步删除 `draft deepcopy` 表述, 并补上 `_load_format_scope` 有界 `override`。
- “没有任何测试 stub 访问器”与现存测试冲突 (testing): 声明收窄为“本次 sweep 转化的两个测试”; 访问器级 stub 例外与全局禁令在文档中互相限定。

风险与影响

- 风险: 纯文档改动, 无运行时风险。主要风险: 1) 文档 - 实现漂移——文中承诺的 `configured*_size()` 读 parallel bag 语义、`_CONFIGURED_SIZE_CALL_SITES` 登记表、棘轮覆盖形态均以配套栈当前实现为准, 后续若新增 DCP 的 config-intent 调用点或扩展别名形态, 文档会过期; 2) 棘轮声明边界——alias-to-alias 链式拷贝与 `import ... as` 改名导入两类盲区在可见回复中未见确认修复, “baselines are 0”的表述可能仍过强; 3) 误导 agent——该 skill 是 AI agent 修改 `runtime_context` 相关代码时的首选参考, 作者自审即

承认旧段落会导致 agent 做出错误重构，任何残留矛盾都会被放大为错误代码；4) 配套依赖——文档若先于配套代码合并，会短暂出现“文档声称收集器已扩展而代码未包含”的窗口。

- 影响：对用户与运行时无任何影响。对开发者与 AI agent workflow 影响显著：该 skill 是 .claude/skills 中面向 runtime_context 改动的治理文档，后续所有涉及配置读取的新代码、重构与测试编写都会以其为指南；新增的“测试双重必须 publish 配置”规则改变组件测试的写法；团队层面为 reader-convergence 架构提供了可执行约束，降低误读种子配置的概率。影响面广但程度温和。
- 风险标记：文档与实现易漂移，依赖 reader-convergence 配套栈，棘轮覆盖声明可能仍过强，指导性文档影响 agent 重构

关联脉络

- PR #34086（同内容 PR，因 force-push 事故被误关）：PR body 明确说明本 PR “Supersedes #34086”：作者侧 bad force-push 使 GitHub 误读为无提交并关闭 / 自动合并了栈内多个 PR，原评审线程继续适用于本内容。
- PR #34087（reader-convergence 栈评审载体）：ch-wan 的自审评论多处署名“Stack review from the reader-convergence review vehicle (#34087)”，是本 PR 及配套栈的评审通道。
- PR #33925（DCP 读取改走 get_parallel() 的上游 PR）：评审回复中说明该上游 PR 删除了 configured_dcp 访问器的消费者、DCP 读取改走 get_parallel()，是本文档 DCP 表述（无 config-intent DCP 调用点）的直接依据。
- PR #34133 config: derive the runner's DCP topology from its ParallelState: 历史 PR 中与本文档 DCP 讨论同一条功能线：DCP 拓扑由 ParallelState 派生、统一 attn_dcp_* 命名，与 get_parallel().attn_dcp_size/.dcp_enabled 的文档描述一致。