PR #35932 完整报告

sgl-project/sglang

Make draft attention backends extensible

合并时间：2026-08-22 14:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35932

执行摘要

- 一句话：草稿注意力后端白名单迁移到共享注册表并开放扩展
- 推荐动作：适合快速阅读，作为理解 SGLang server_args 注册表扩展模式的小样例。若你正在维护外部注意力后端集成，建议关注 add_draft_attention_backend_choices 的调用时机（必须在 ServerArgs 构造前）。若你关心配置系统重构，本 PR 与近期大量 config 相关 PR 展示的注册表模式一致。

功能与动机

PR body 指出：Out-of-tree integrations can register general attention backends, but draft workers keep a separate private allowlist. As a result, a draft-compatible external backend cannot opt into the common draft worker path. 因此需要把这个私有白名单开放为可扩展的共享注册表，让外部后端能在不修改核心逻辑的前提下复用草稿 worker 的公共路径。

实现拆解

1. 在 python/sglang/srt/server_args.py 中新增模块级常量 DRAFT_ATTENTION_BACKEND_CHOICES，内容与原本的 _SUPPORTED_DRAFT_BACKENDS 完全一致（flashinfer/fa3/fa4/triton/ascend/trtllm_mha），并补充注释说明 trtllm_mha 仅适用于 decode-only 密集 MQA 草稿。
2. 新增注册辅助函数 add_draft_attention_backend_choices(choices)，对 DRAFT_ATTENTION_BACKEND_CHOICES 执行 extend，与已有的 add_attention_backend_choices / add_deterministic_attention_backend_choices 等扩展入口保持同一模式。
3. 在 python/sglang/srt/speculative/draft_worker_common.py 中删除私有元组 _SUPPORTED_DRAFT_BACKENDS，改为从 server_args 导入 DRAFT_ATTENTION_BACKEND_CHOICES；_resolve_draft_attention_backend_fallback 中的白名单判断和告警日志均改用它。
4. 测试配套：未新增单元测试文件，作者通过 pre-commit、py_compile 和 registry-to-resolver smoke test 验证注册与解析链路；该改动要求注册发生在 ServerArgs 构造 / 解析之前。

关键文件：

- python/sglang/srt/server_args.py（模块 参数注册；类别 source；类型 core-logic；符号 add_draft_attention_backend_choices, DRAFT_ATTENTION_BACKEND_CHOICES）：核心变更文件：新增 DRAFT_ATTENTION_BACKEND_CHOICES 白名单和 add_draft_attention_backend_choices 扩展入口，使草稿注意力后端可以像普通注意力后端一样被外部注册。
- python/sglang/srt/speculative/draft_worker_common.py（模块 投机解码；类别 source；类型 dependency-wiring；符号 _resolve_draft_attention_backend_fallback）：将草稿 worker 的私有白名单改为引用共享注册表，fallback 逻辑随之使用可扩展列表，是本 PR 的消费侧变更。

关键符号：add_draft_attention_backend_choices, _resolve_draft_attention_backend_fallback

关键源码片段

python/sglang/srt/server_args.py

核心变更文件：新增 DRAFT_ATTENTION_BACKEND_CHOICES 白名单和 add_draft_attention_backend_choices 扩展入口，使草稿注意力后端可以像普通注意力后端一样被外部注册。

```
# 草稿解码（draft decoding）可用的注意力后端白名单。
# 该列表与 ATTENTION_BACKEND_CHOICES 分离，因为草稿 worker 的
# 后端支持范围更小（例如 trtllm_mha 只支持 decode-only 的密集 MQA 草稿）。
# 外部集成可通过 add_draft_attention_backend_choices() 在 ServerArgs
# 构造 / 解析前扩展该列表，使自研后端能进入公共草稿 worker 路径。
DRAFT_ATTENTION_BACKEND_CHOICES = [
    'flashinfer',
    'fa3',
    'fa4',
    'triton',
    'ascend',
    'trtllm_mha',
]


# 为草稿 worker 注册额外的注意力后端名称。
def add_draft_attention_backend_choices(choices):
    DRAFT_ATTENTION_BACKEND_CHOICES.extend(choices)
```

python/sglang/srt/speculative/draft_worker_common.py

将草稿 worker 的私有白名单改为引用共享注册表，fallback 逻辑随之使用可扩展列表，是本 PR 的消费侧变更。

```
def _resolve_draft_attention_backend_fallback(
    *, server_args: ServerArgs, algo_label: str
) -> str:
    # 优先使用显式指定的草稿后端；未指定时从目标模型的后端解析结果中继承。
    draft_backend = server_args.speculative_draft_attention_backend
    if draft_backend is None:
```

```
        draft_backend, _ = server_args.get_attention_backends()
    if draft_backend is None:
        # 平台默认：AMD 用 triton，否则用 flashinfer
        return 'triton' if torch.version.hip else 'flashinfer'

    # 不在共享注册表中或未注册的后端，回退到平台默认值并告警。
    # 通过 DRAFT_ATTENTION_BACKEND_CHOICES 判断，而不是私有常量，
    # 这样外部注册的后端也能被识别，而不是被误判为不支持。
    if draft_backend not in DRAFT_ATTENTION_BACKEND_CHOICES:
        fallback = 'triton' if torch.version.hip else 'flashinfer'
        logger.warning(
            '%s draft worker only supports attention_backend in %s for now, '
            'but got %r. Falling back to %s.',
            algo_label,
            DRAFT_ATTENTION_BACKEND_CHOICES,
            draft_backend,
            fallback,
        )
        return fallback
    return draft_backend
```

评论区精华

该 PR 没有实质 review 讨论线程；唯一评论是作者触发的 /tag-and-rerun-ci。作者在 body 中说明 Accuracy/Speed Tests 不适用，因为内置后端的计算路径未改变。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险集中在缺少正式回归测试；DRAFT_ATTENTION_BACKEND_CHOICES 是模块级可变列表，若外部代码在 ServerArgs 完成解析后再调用 add_draft_attention_backend_choices，新增项不会生效且无报错。此外，共享列表可被外部意外修改，不过这与 SGLang 现有注册函数风格一致。fallback 逻辑与改动前等价，因此内置后端行为无回归风险。
- 影响：对内置用户无影响：后端解析顺序、默认值、回退逻辑均不变。对外部集成者有实际收益：自研的兼容草稿注意力后端现可通过注册函数进入公共草稿 worker 路径，无需 fork 或 patch SGLang。对团队而言，消除了 draft_worker_common.py 中重复定义的白名单，降低未来新增内置后端时漏改的风险。
- 风险标记：缺少单元测试，注册时机敏感，解析后注册无效，低风险小改动

关联脉络

- PR #29143 Add intel_xpu to DETERMINISTIC_ATTENTION_BACKEND_CHOICES: 同样修改 server_args.py 中的注意力后端选择列表，展示了共享注册表模式的持续演进。
- PR #35907 config: constructing a config no longer resolves it: 重构 ServerArgs 构造与解析时机，与本 PR 的注册时机（构造前）直接相关。

- PR #35905 config: record resolution writes in a declaration stash: 同为 server_args 注册表与解析机制的重构系列，体现配置系统统一注册模式的趋势。