PR #34264 完整报告

sgl-project/sglang

config: decisions keyed on the attention backend read the configured pair

合并时间：2026-08-15 15:38

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34264

执行摘要

- 一句话：注意力后端相关决策改读 prefill/decode 配置对，修复拆分加载下误读
- 推荐动作：值得精读。核心看点：(1) ' 构造器回答 effective name' 替代静态改名表的设计——对按宿主选型的别名后端是唯一正确解；(2) serving_attention_backend 严格镜像 hybrid dispatcher 的 per-mode 选择逻辑；(3) ' 可调用测试 + AST 静态守护 + 反向验证 ' 的组合防回归手段。建议关注 #34917（gpt-oss sinks 后续）与 Inkling split pair 的 forward_metadata 后端补齐工作。

功能与动机

PR body 明确指出：--attention-backend is one field of three: a launch that sets only --prefill-attention-backend or --decode-attention-backend leaves the base field at None。Seven decisions read that base field alone and therefore answered from a field the operator never set。这导致 split 配置下 chunked prefix cache 被误关、triton 内核被选到 tor 无法承载的后端（support_triton(None) 返回 True）、flashinfer 版本守卫永不触发、deterministic truncation 未设置等一系类行为偏差。

实现拆解

1. 统一取数入口：将所有决策点从 get_exec().kernel.attention_backend 改为 attention_backends()（返回带 base 字段回退的 (prefill, decode) 对）。逐站点判断所需半边：chunked prefix cache（misc_utils.py）、truncation align（scheduler.py 的 init_deterministic_inference_config）、req-to-token writer（allocation.py 的 write_cache_indices）取 prefill 半；mrope 的 interleaved rope 两半都要支持 triton；get_last_loc 保持双半读取；flashinfer 版本守卫改用 server_args.get_attention_backends()（engine.py）。
2. 新增 per-mode 选择函数：inkling_common/attn.py 新增 serving_attention_backend(forward_batch)，严格镜像 HybridAttnBackend._select_backend：decode/idle 走 decode 半，target-verify 按 speculative_attention_mode 选边，其余（含 draft-extend）走 prefill 半；并优先采纳 runner 打在 backend 上的 stamp。该函数同时替代 InklingAttention.forward 中凭意图选边的旧逻辑，修正 extend 阶段误用 decode 半导致 kwargs 组装崩溃的问题。
3. Draft 产物 stamp 机制（draft_utils.py）：DraftBackendFactory 所有工厂叶子由返回 backend 改为返回 (effective_name, backend)——因为 cutedsl_mla 的 draft-extend 实

际构造 trtllm-mla、"nsa" 是构造 dsa 的废弃别名、hybrid_linear_attn 按宿主选择 fa3/intel_amx/triton，静态改名表无法覆盖。_create_backend 把有效内核名写进 backend 的 prefill/decode_attention_backend_str，create_decode_backend 显式开启 stamps_children=True 让 EAGLE 每步子对象同样被打标。draft-extend 的 conv-sidecar 包装器复制内层 stamp。

4. gpt-oss sinks 决策删除 + trtllm 调用点 upcast：原按后端定 dtype 的决策无法服务混合对（FA4 断言 bf16、trtllm 消费 fp32），最终删除该决策，sinks 恒为 bf16；trtllm_mha_backend.py 在 forward_decode / forward_extend 调用点通过 _sinks_float32() 升精度并缓存（键为 id(sinks) + Tensor._version，纯 capture 下无条件转换）。
5. 测试与静态守护：新增 test/registered/unit/test_split_attention_backend_decisions.py（361 行），包含 TestSplitBackendsReachTheDecisions 的可调用测试（覆盖 extend/decode/idle/verify 及两种 spec mode）、AST ratchet（对 _PAIR_READERS 列出的 7 个文件 + engine.py 中任何 attention_backend 属性访问判红）和 TestDraftFactoryStamping（钉死容器子对象 stamp、draft override 优先、wrapper 复制、宿主相关别名、叶子不得返回别名的静态 guard）。验证：Qwen2-VL --cuda-graph-backend-prefill tc_piecewise 0 次 graph break 且字节一致；gpt-oss-20b 默认后端与 base 字节一致。

关键文件：

- test/registered/unit/test_split_attention_backend_decisions.py（模块 回归测试；类别 test；类型 test-coverage；符号 TestSplitBackendsReachTheDecisions, TestDraftFactoryStamping, _publish, test_no_listed_decision_reads_the_base_field_alone）：新增 361 行测试，是整套防回归机制的核心：可调用测试钉死 serving_attention_backend 在 4 种 forward mode × 2 种 spec mode 下的选边；AST ratchet 对 8 个文件的任何 attention_backend 属性访问判红；TestDraftFactoryStamping 反向验证容器 / 子对象 /wrapper 的 stamp 传播。
- python/sglang/srt/speculative/draft_utils.py（模块 投机解码；类别 source；类型 core-logic；符号 DraftBackendFactory._create_backend, DraftBackendFactory.create_decode_backend, DraftBackendFactory.create_draft_extend_backend, DraftBackendFactory._create_dsa_decode_backend）：Draft 产物 stamp 机制的主体：所有工厂叶子改为返回 (effective_name, backend)，_create_backend 负责把 stamp 写到容器与子对象，draft-extend 包装器复制内层 stamp。这是本 PR 设计最重的改动，修复 EAGLE eager 循环中未打标子对象回退错 target 对的问题。
- python/sglang/srt/models/inkling_common/attn.py（模块 注意力后端；类别 source；类型 data-contract；符号 serving_attention_backend）：新增 serving_attention_backend()，按 forward_mode 严格镜像 HybridAttnBackend._select_backend 选边；InklingAttention.forward 由读 base 字段改为调用该函数，修复 split 配置下 extend 阶段误用 decode 半导致 kwargs 组装与 fused prologue 门控错乱的问题。

- python/sglang/srt/mem_cache/allocation.py（模块 内存缓存；类别 source；类型 dependency-wiring；符号 write_cache_indices, get_last_loc）：write_cache_indices 改用 prefill 半决定是否走 triton 路径（唯一调用方是 alloc_for_extend），避免 decode 半 gating 让混合启动的每次 extend 都掉入逐请求 .item() 同步的慢路径；get_last_loc 保持双半读取（verify token 可能由任一半服务）。
- python/sglang/srt/model_executor/model_runner_components/misc_utils.py（模块 前缀缓存；类别 source；类型 data-contract；符号 maybe_disable_chunked_prefix_cache）：chunked prefix cache 是 prefill 特性，maybe_disable_chunked_prefix_cache 改读 prefill 半；此前 split-only 启动下 base 为 None 会误关该特性。
- python/sglang/srt/layers/rotary_embedding/mrope.py（模块 旋转编码；类别 source；类型 dependency-wiring；符号 get_cos_sin_with_position）：interleaved rope 内核在 prefill 与 decode 两阶段都会运行，改要求两半都支持 triton 才走 triton 路径；support_triton(None) 返回 True 的陷阱此前会让 --prefill-attention-backend torch_native 误走 triton。
- python/sglang/srt/batch_overlap/two_batch_overlap.py（模块 批次调度；类别 source；类型 dependency-wiring；符号 derive_fields_related_to_seq_len_for_two_chunk）：derive_fields_related_to_seq_len_for_two_chunk 计算的是 extend 位置，改由 prefill 半驱动。
- python/sglang/srt/managers/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 init_deterministic_inference_config）：init_deterministic_inference_config 映射的是 prefill 旋钮（SPLIT_TILE / PREFILL_TRUNCATION_ALIGN），改读 prefill 半；此前 split-only 启动下 truncation align 从未被设置。
- python/sglang/srt/entrypoints/engine.py（模块 启动入口；类别 source；类型 core-logic；符号 _set_envs_and_config）：flashinfer 版本守卫改判断 flashinfer in server_args.get_attention_backends()，使 split 启动也能触发版本下限检查；同时被纳入 AST ratchet。
- python/sglang/srt/layers/attention/trtllm_mha_backend.py（模块 注意力后端；类别 source；类型 core-logic；符号 TRTLLMHAAttnBackend._sinks_float32, TRTLLMHAAttnBackend.forward_decode, TRTLLMHAAttnBackend.forward_extend）：gpt-oss sinks 决策删除后，trtllm 后端在调用点上负责 bf16 -> fp32 升精度，并以 id + Tensor._version 缓存、纯 capture 标志区分 graph 重放，避免 eager 每层一次转换 kernel。

关键符号：serving_attention_backend, DraftBackendFactory._create_backend, DraftBackendFactory.create_decode_backend, DraftBackendFactory.create_draft_extend_backend, maybe_disable_chunked_prefix_cache, write_cache_indices, get_last_loc, init_deterministic_inference_config, MRotaryEmbedding.get_cos_sin_with_position, TRTLLMHAAttnBackend._sinks_float32, derive_fields_related_to_seq_len_for_two_chunk

关键源码片段

test/registered/unit/test_split_attention_backend_decisions.py

新增 361 行测试，是整套防回归机制的核心：可调用测试钉死 serving_attention_backend 在 4 种 forward mode × 2 种 spec mode 下的选边；AST ratchet 对 8 个文件的任何 attention_backend 属性访问判红；TestDraftFactoryStamping 反向验证容器 / 子对象 /wrapper 的 stamp 传播。

```
# 静态守护：任何名为 attention_backend 的属性读取都视为 base 字段误读。
# 不管 base 表达式怎么拼（bag 链、record、或局部别名
# k = get_exec().kernel; k.attention_backend），都会被捕获；
# pair 访问器是函数调用，不会命中 Attribute 检查。
def test_no_listed_decision_reads_the_base_field_alone(self):
    offenders = []
    for rel, why in _PAIR_READERS.items():
        tree = ast.parse((_PACKAGE_ROOT / rel).read_text())
        for node in ast.walk(tree):
            if isinstance(node, ast.Attribute) and node.attr == "attention_backend":
                offenders.append(f"{rel}:{node.lineno}: base-only read ({why})")
    self.assertEqual(
        [],
        offenders,
        "these decisions must read attention_backends() (the pair with the "
        "base-field fallback), not the base field:\n" + "\n".join(offenders),
    )
```

python/sglang/srt/speculative/draft_utils.py

Draft 产物 stamp 机制的主体：所有工厂叶子改为返回 (effective_name, backend)，_create_backend 负责把 stamp 写到容器与子对象，draft-extend 包装器复制内层 stamp。这是本 PR 设计最重的改动，修复 EAGLE eager 循环中未打标子对象回退错 target 对的问题。

```
def _create_backend(
    self, backend_name: str, backend_map: dict, error_template: str,
    stamps_children: bool = False,
):
    # 先取带 base 回退的配置对，再按调用方需求取 decode 或 prefill 半
    prefill_backend, decode_backend = attention_backends()
    configured = (
        decode_backend
        if backend_name == "decode_attention_backend"
        else prefill_backend
    )
    backend_type = self.draft_attn_backend or configured
    if backend_type not in backend_map:
        raise ValueError(error_template.format(backend_type=backend_type))

    # 每个工厂叶子返回 (effective_name, backend)：别名构造器
    #  （cutedsl_mla -> trtllm_mla）、废弃 key（nsa -> dsa）和按宿主
    # 选型的 hybrid_linear_attn，都必须由实际运行的构造器回答自己的
    # 名字 —— 静态改名表无法表达这些映射。
    stamp, backend = backend_map[backend_type]()
    if backend is not None:
```

```
        backend.prefill_attention_backend_str = stamp
        backend.decode_attention_backend_str = stamp
        # EAGLE 多步容器的每步子对象会被直接放进 ForwardContext,
        # 所以子对象必须带同样的 stamp；只有 decode 容器打开这个
        # 开关（stamps_children=True 是显式契约，不靠 getattr 探测）。
        if stamps_children:
            for child in backend.attn_backends:
                child.prefill_attention_backend_str = stamp
                child.decode_attention_backend_str = stamp
    return backend

# create_draft_extend_backend 末尾：conv-sidecar 包装器是真正进入
# ForwardContext 的对象，必须复制内层 backend 的 stamp，否则会回退
# 到 target 的配置对。
wrapped = attn_backend_wrapper_for_draft_extend(self.draft_model_runner, backend)
if wrapped is not backend and wrapped is not None and backend is not None:
    wrapped.prefill_attention_backend_str = backend.prefill_attention_backend_str
    wrapped.decode_attention_backend_str = backend.decode_attention_backend_str
return wrapped
```

python/sglang/srt/layers/attention/trtllm_mha_backend.py

gpt-oss sinks 决策删除后，trtllm 后端在调用点上负责 bf16 -> fp32 升精度，并以 id + Tensor._version 缓存、纯 capture 标志区分 graph 重放，避免 eager 每层一次转换 kernel。

```
def _sinks_float32(self, sinks: torch.Tensor) -> torch.Tensor:
    """bf16 sinks 权重的精确 float32 升精度。

    按源张量缓存、权重原地更新（version 自增）时重新派生，
    eager 步不付每层转换 kernel 的代价。真正的 CUDA graph
    capture 下无条件发射转换：captured kernel 每次 replay 重新
    读取权重，这正是让 graph 内原地更新可见的机制 —— 缓存会
    过期。注意 breakable-cuda-graph 范围不算 capture（该模式下
    attention 在 graph 段之间仍走 eager），所以这里检查纯 capture
    标志，而不是 get_is_capture_mode()。
    """
    if _capture_mode.is_capture_mode:
        return sinks.to(torch.float32)
    key = id(sinks) # 依赖 autograd 的 _version 私有契约：
    # 它在 copy_ 式权重更新时自增；若未来被移除会在首个 forward
    # 响亮失败，而不是静默提供过期缓存。
    cached = self._sinks_fp32_cache.get(key)
    if cached is not None and cached[0] == sinks._version:
        return cached[1]
    converted = sinks.to(torch.float32)
    self._sinks_fp32_cache[key] = (sinks._version, converted)
    return converted
```

评论区精华

Review 共 30 条评论，主要由 ch-wan 逐条回复 Codex 及自查意见，核心交锋如下：

- Inkling 按模式选后端（P1）：Codex 指出无条件取 decode 半会让 extend 走错 kwargs 与 fused prologue。作者回复：fixed，per-forward-mode 通过 serving_attention_backend() 选择，完全镜像 HybridAttnBackend._select_backend，并新增全模式测试。
- gpt-oss sinks dtype（P1，三轮交锋）：先改为 " 任一 half 含 trtllm 即 fp32"，Codex 指出 FA4 断言 bf16 会炸；再改为 " 只有纯 trtllm 才 fp32"，Codex 又指出 standalone draft 场景下 draft backend 在模型构造后才应用，构造期无法判断。最终结论：'the decision is removed instead of patched'——sinks 恒 bf16，trtllm 调用点 upcast，独立行为问题拆到 #34917。
- 多步 EAGLE 子容器未 stamp（bug）：作者承认 'EAGLE's multi-step container hands its per-step children to the ForwardContext directly, so an unstamped child fell back to the target pair mid-loop'，以 stamps_children=True 显式契约修复，并补 TestDraftFactoryStamping 钉死。
- 别名 stamp 必须是实际内核：'the hybrid constructor picks fa3/intel_amx/triton by host, so no static rename can ever say what it builds'——废弃静态改名表，构造器自己回答 effective name。
- _sinks_float32 缓存与 _version 私有 API：Codex 多次质疑 Tensor._version 非公开契约，作者选择保留并文档化：'autograd's in-place version counter is the only per-tensor signal that ticks on copy_-style weight updates ... a future removal fails loudly at the first forward rather than serving a stale cache'。
- AST ratchet 覆盖面：nit 指出只匹配 *.kernel.attention_backend 会漏掉局部别名，且 engine.py 未入表。作者将守卫改为捕获任何 attention_backend 属性读取并加入 engine.py。
  - Inkling forward 需按 forward_mode 选后端（P1）(correctness): 已解决：新增 serving_attention_backend() 并覆盖全模式测试；split pair 可运行性问题单独留待后端补齐。
  - gpt-oss sinks dtype 无法服务混合后端对（P1，三轮交锋）(design): 已解决：删除 dtype-by-backend 决策，trtllm 调用点 upcast + 缓存；行为分类问题拆到 #34917。
  - 多步 EAGLE 子容器未 stamp（bug）(correctness): 已解决：_create_backend 增加 stamps_children=True 显式契约，子对象与容器同 stamp；TestDraftFactoryStamping 反向验证。
  - 别名 / 宿主相关构造器的 stamp 必须是实际内核名（P1）(design): 已解决：构造器回答有效内核名；静态 guard 禁止任何叶子返回别名（反向验证）。
  - sinks upcast 缓存与 Tensor._version 私有 API (design): 已解决：缓存按 id + _version，纯 capture 无条件转换，私有 API 契约写入注释。
  - AST ratchet 防本地别名与 engine.py 覆盖 (testing): 已解决：任意拼写的 base 字段属性读取都会被捕获，pair 访问器因是函数调用而豁免。

风险与影响

- 风险：
  - 用户可见行为变化（预期但需部署注意）：*engine.py* 的 flashinfer 版本守卫现在对 split 启动生效，旧 flashinfer + *--decode-attention-backend* flashinfer 会启动即失败（fail fast），而非运行期神秘报错；*scheduler.py* 的 truncation align 对 prefill-only 配置开始设置，可能改变 kernel 选择。
  - 缓存依赖私有 API：trtllm_mha_backend.py 的 _sinks_float32 以 id(sinks) + Tensor._version 缓存；若权重对象被整体 rebind（而非 *copy_* 原地更新）绕过 _version，缓存可能失效；若 PyTorch 移除 _version 则直接抛错而非静默错误。作者已文档化该契约并接受此风险。
  - *draft stamp* 契约的隐性假设：DraftBackendFactory._create_backend 现在要求所有叶子返回 (stamp, backend) 二元组，且 *decode* 容器必须暴露 attn_backends 属性（由 stamps_children=True 显式保证）。任何遗漏都会在解包处崩溃，测试已覆盖主要叶子，但第三方自定义后端不在保护范围内。
  - Inkling split pair 实际不可用：作者在 review 中自述 HybridAttnBackend.forward_metadata 缺失导致 split pair 从未真正跑通，本 PR 的 *per-mode* 选择在该场景下无端到端验证（fa4 亦为 Blackwell-only，Hopper 无法验证）。
  - 静态 *ratchet* 维护成本：AST 守卫对 7+1 个文件任何 *attention_backend* 属性访问判红，未来若出现合法的新式读取（如经访问器属性）需要维护豁免。
  - *mrope* 双半判断收紧：*get_cos_sin_with_position* 由单一半改为两半都必须支持 triton 才走 triton 路径，混合配置下可能回退到慢路径 Python 实现——这是有意保守，但需观察性能。
- 影响：
  - 用户 / 部署：使用 split attention backend 的启动（prefill/*decode* 异后端）获得与 base 字段启动一致的正确语义，*chunked prefix cache*、*deterministic* truncation、flashinfer 版本检查不再被静默跳过。gpt-oss + trtllm 默认后端字节级一致。
  - 系统：7 个横跨 *scheduler*、*mem_cache*、*rotary*、*engine*、*speculative* 的决策点统一读 pair；*serving_attention_backend* 成为 Inkling 模型与 hybrid 调度器之间的契约；*draft* 产物 *stamp* 成为新约定。
  - 团队：本 PR 是 *config* 栈（#34270）中唯一带用户可见行为变化的成员，后续 #34819 等消费迁移依赖本 PR 建立的 *pair* 读取约定；新增的 AST *ratchet* 将成为该栈持续集成的保护网。
  - 风险标记：split 启动下用户可见行为变化，依赖 Tensor._version 私有 API, Inkling split pair 无端到端验证，static ratchet 需持续维护，draft stamp 契约隐性假设

关联脉络

- PR #34265 *config: a named entry point for the resolution pipeline, and the last dynamic config read*: 同一 *config* 栈（#34270）前序成员：提供解析管线命名入口并清理最后一个动态配置读取，为本 PR 的 *pair* 读取建立基础。

- PR #34266 config: the alias form of the runner-side instance read: 同一栈前序成员：runner 侧别名读取迁移到配置 bag，attention_backends() 的 stamp 语义与 runner 侧配置对由此而来。
- PR #34267 config: pin the supplied-instance surface that a raw record would change: 同一栈前序成员：以 ratchet 钉测暴露面，本 PR 的 AST 静态守护与其一脉相承。
- PR #34819 config: the post-publish consumers of the supplied-instance surface read the bags: 同一栈后续成员：17 个消费点从 server_args 迁移到 bags，依赖本 PR 建立的 pair 读取约定。
- PR #34913 [CI] Move the static ratchets back to CPU unit tests: 后续 CI 调整：本 PR 新增的静态 ratchet（含 test_split_attention_backend_decisions.py 的守卫）被移回 CPU 单测以降低提交延迟，二者直接关联。