# PR #36680 完整报告

sgl-project/sglang

[Diffusion] Optimize Qwen-Image TP collectives and attention

合并时间：2026-09-01 10:28

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36680

## 执行摘要

- 一句话：Qwen-Image TP 集体通信与注意力性能优化
- 推荐动作：值得精读。该 PR 展示了三个可复用的设计决策：①把非 bit-exact 的算子融合包装成 request-scoped 质量门控（quality=lossless/high），默认保持数值兼容、按需拉满性能，避免了“性能 PR 改变输出语义”的经典冲突；②性能优化必须证明“改动分支确实执行”，PR 中的跨模型审计与负数控制组（FLUX.2 Klein、LTX-2）是高质量证据；③TP 通信、分段打包、FA3 调度三个优化各自保留 fallback 与单测。建议重点关注 qwen_image_added_qkv_site.py 的 QualityGatedFusion 复用方式和 USPAttention.forward 的分段前缀接口设计。

## 功能与动机

PR body 给出了三处相对 vLLM-Omni 的性能缺口：一是扩散模型直接实例化了旧版 custom all-reduce 实现，24 MiB 的 row-parallel 输出回退到 NCCL 而非 CUDA V2 调度器；二是每个注意力块在打包前物化了联合 text/image Q、K、V 张量，采样窗口内出现 1,212 次 CatArray launch 耗时 10.768 ms；三是 SGLang 对 batch-one 密集序列选择了 FA3 VarlenDynamic（159.403us/call），而vLLM选择了StaticPersistent（147.402us/call）。作者还通过跨模型 H200 ABBA 审计强调：并非所有 Qwen 检查点都能自动提速，且 fused added-QKV GEMM 因改变 BF16 归约结合序而不再 bit-exact，因此将其设计为 request-scoped 质量门控站点，默认 lossless 保持参考三段 GEMM。

## 实现拆解

1. TP 集体通信接入 SRT custom-all-reduce V2 调度器：在 python/sglang/multimodal_gen/runtime/distributed/group_coordinator.py 的 _init_srt_custom_allreduce 中，CUDA 平台改用 dispatch_custom_allreduce 选择实现；当命中 CustomAllReduceV2 时传入新增的 _DIFFUSION_CUSTOM_AR_MAX_SIZE = 32 MiB 工作区上限（默认 16 MiB 会让 24 MiB 张量回退 NCCL）。all_reduce 同时从私有接口 _all_reduce_impl 改为公开的 custom_all_reduce，并在返回 None（不支持）时回退 NCCL。ROCm/MUSA 仍走旧 CustomAllreduce。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/qwen_image.py（模块 模型运行时；类别 source；类型 data-contract；符号 _split_unquantized_merged_linear,

_get_added_qkv_projections）：核心实现：未量化 added-text QKV 融合为单一 packedGEMM，新增 _get_added_qkv_projections 与 _split_unquantized_merged_linear 双路径、质量门控挂接、分段前缀传入注意力，并调整 attn_mask 构造顺序。

- python/sglang/multimodal_gen/runtime/layers/attention/layer.py（模块 注意力层；类别 source；类型 core-logic）：USPAttention 新增分段前缀参数与 fused_pack_segmented_qkv 快路径，并为 batch-one 密集序列切换 FA3 static persistent 调度器（cu_seqlens=None），直接影响所有带 2-D mask 的扩散模型注意力。
- python/sglang/multimodal_gen/runtime/distributed/group_coordinator.py（模块 分布式通信；类别 source；类型 dependency-wiring）：扩散 TP 集体通信接入 SRT custom-all-reduce V2 调度器并扩容 workspace 至 32 MiB，影响所有 CUDA TP>1 扩散模型。
- python/sglang/kernels/ops/diffusion/sites/qwen_image_added_qkv_site.py（模块 质量门控；类别 infra；类型 infrastructure；符号 mark_qwen_image_added_qkv_site, qwen_image_added_qkv_active, _site_reject_reason, mount_qwen_image_added_qkv）：新增 request-scoped 质量门控站点，将 fused added-QKV 的 mount/unmount 逻辑与 reject 原因封装，是 lossless/high 语义的关键基础设施。
- python/sglang/multimodal_gen/runtime/layers/linear.py（模块 线性层；类别 source；类型 core-logic；符号 apply_unquantized_linear）：提取 apply_unquantized_linear 公共函数，供 lossless 参考 GEMM 复用，避免语义漂移。
- python/sglang/multimodal_gen/configs/models/dits/qwenimage.py（模块 模型配置；类别 source；类型 data-contract）：新增权重加载映射，将 add_q/k/v_proj 合并到 to_added_qkv 的三段 shard；同时为 ModelOpt 量化检查点提供空的 quant_param_names_mapping 以避免误用运行时融合映射。
- test/registered/kernels/ops/diffusion/test_model_fast_paths.py（模块 快速路径测试；类别 test；类型 test-coverage；符号 _PackedAddedQKV, init, forward, test_qwen_added_qkv_lossless_uses_three_reference_gemms）：新增 lossless 三段参考 GEMM 与 high 单 GEMM 的等价性测试，验证质量门控语义与 packed 计数。
- test/registered/kernels/benchmark/diffusion/bench_varlen_segmented_pack.py（模块 打包基准；类别 test；类型 test-coverage；符号 _materialized_pack, _segmented_pack, benchmark）：新增分段打包 vs 物化拼接的 benchmark，量化 47.7 us -> 21.1 us 的收益并内置正确性断言。
- python/sglang/kernels/ops/diffusion/layout/varlen_pack_pad_triton.py（模块 打包内核；类别 infra；类型 infrastructure；符号 _fused_pack_segmented_qkv_kernel, fused_pack_segmented_qkv）：新增 fused_pack_segmented_qkv Triton 内核，支持文本前缀与图像主体分段直接打包。
- python/sglang/multimodal_gen/test/unit/test_diffusion_bcg_tp_graph_capture.py（模块 图捕获测试；类别 test；类型 test-coverage；符号 test_cuda_custom_allreduce_uses_v2_dispatch_and_diffusion_workspace, test_non_cuda_custom_allreduce_preserves_default_workspace, test_all_reduce_uses_public_custom_allreduce_api）：覆盖 custom-all-reduce V2 调度、32 MiB workspace、非 CUDA 回退与公开 API 调用的行为。

- test/registered/kernels/ops/diffusion/test_layout.py（模块 布局测试；类别 test；类型 test-coverage；符号 test_varlen_segmented_pack_matches_materialized_joint, test_fa_dense_scheduler_matches_single_sequence_varlen）：验证分段打包与物化拼接等价、FA3 dense 调度与 varlen 结果一致。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/denoising.py（模块 去噪管线；类别 source；类型 core-logic）：挂接质量门控的 mount/unmount，使 quality=high 生效、切换回 lossless 时在下一 batch 边界卸载。
- python/sglang/multimodal_gen/runtime/loader/transformer_load_utils.py（模块 权重加载；类别 source；类型 core-logic）：在 NVFP4 回退映射发现上保留本 PR 的 quant_param_names_mapping，并兼容 main 分支的 transformer override 量化检测。
- test/registered/kernels/ops/diffusion/test_sites.py（模块 门控测试；类别 test；类型 test-coverage；符号 test_qwen_image_added_qkv_site_is_request_scoped）：验证 added-QKV 门控是 request-scoped（mount/unmount 生命周期）。
- docs/cookbook/diffusion/Qwen-Image/Qwen-Image.mdx（模块 使用文档；类别 other；类型 core-logic）：记录双 H200 验证配方与 quality 语义，指导用户选择 lossless/high。
- docs/docs/sglang-diffusion/performance-optimization.mdx（模块 性能文档；类别 other；类型 core-logic）：同步 diffusion 性能优化文档，说明新增的 fused added-QKV 与分段打包。
- python/sglang/kernels/ops/diffusion/__init__.py（模块 内核导出；类别 infra；类型 infrastructure）：导出新增的 fused_pack_segmented_qkv 与 added-QKV 门控接口。
- python/sglang/kernels/ops/diffusion/README.md（模块 内核文档；类别 docs；类型 documentation）：内核文档同步新增内核说明。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-performance/SKILL.md（模块 技能文档；类别 docs；类型 documentation）：更新 diffusion 性能优化 skill 文档以匹配新的优化模式。

关键符号：_split_unquantized_merged_linear, _get_added_qkv_projections, apply_unquantized_linear, fused_pack_segmented_qkv, _fused_pack_segmented_qkv_kernel, mark_qwen_image_added_qkv_site, qwen_image_added_qkv_active, mount_qwen_image_added_qkv, unmount_qwen_image_added_qkv, _site_reject_reason, _init_srt_custom_allreduce, USPAttention.forward

## 关键源码片段

### python/sglang/multimodal_gen/runtime/models/dits/qwen_image.py

核心实现：未量化 added-text QKV 融合为单一 packed GEMM，新增 _get_added_qkv_projections 与 _split_unquantized_merged_linear 双路径、质量门控挂接、分段前缀传入注意力，并调整 attn_mask 构造顺序。

```
# python/sglang/multimodal_gen/runtime/models/dits/qwen_image.py
# 关键变更：未量化 added-text QKV 权重保持 packed 常驻内存，
# 但根据请求 quality 决定走“三段参考 GEMM”还是“单 fused GEMM”。
```

```
def _split_unquantized_merged_linear(
    linear: MergedColumnParallelLinear, x: torch.Tensor
) -> tuple[torch.Tensor, torch.Tensor, torch.Tensor]:
    """把 packed 权重按 output_partition_sizes 切回三段，逐段做参考线性投影。

    这样做的原因是：packed 单 GEMM 会改变 BF16 归约结合序，
    导致结果与原始三个独立投影不是 bit-exact。
    """
    sizes = linear.output_partition_sizes
    if len(sizes) != 3:
        raise ValueError(f"Expected three packed projection shards, got {sizes}")
    weights = linear.weight.split(sizes, dim=0)
    biases = (
        linear.bias.split(sizes, dim=0)
        if linear.bias is not None
        else (None, None, None)
    )
    return tuple(
        apply_unquantized_linear(x, weight, bias)
        for weight, bias in zip(weights, biases)
    )


def _get_added_qkv_projections(
    self, encoder_hidden_states: torch.Tensor
) -> tuple[torch.Tensor, torch.Tensor, torch.Tensor]:
    # 未量化 packed 且当前请求非 high 时，回归参考三段 GEMM
    if self.use_fused_added_qkv:
        if (
            self._unquantized_added_qkv_is_packed
            and not qwen_image_added_qkv_active(self)
        ):
            return _split_unquantized_merged_linear(
                self.to_added_qkv, encoder_hidden_states
            )
        # quality=high：一次 fused GEMM 后按输出维切三份
        added_qkv, _ = self.to_added_qkv(encoder_hidden_states)
        return tuple(t.contiguous() for t in added_qkv.chunk(3, dim=-1))

    encoder_query, _ = self.add_q_proj(encoder_hidden_states)
    encoder_key, _ = self.add_k_proj(encoder_hidden_states)
    encoder_value, _ = self.add_v_proj(encoder_hidden_states)
    return encoder_query, encoder_key, encoder_value
```

### python/sglang/multimodal_gen/runtime/layers/attention/layer.py

USPAttention 新增分段前缀参数与 fused_pack_segmented_qkv 快路径，并为 batch-one 密集序列切换 FA3 static persistent 调度器（cu_seqlens=None），直接影响所有带 2-D mask 的扩散模型注意力。

```
# python/sglang/multimodal_gen/runtime/layers/attention/layer.py
# 变更核心：文本前缀不再先物化拼接，而是分段直接打包进有效行；
# batch==1 时用 FA3 的 static persistent 调度替代 varlen dynamic。

if indices.shape[0] > 0:
    all_valid = indices.shape[0] == bs * seq
    if segmented_prefix:
        # 文本（prefix）与图像（main）分段打包，避免三份 cat 后再 gather
        q_unpad, k_unpad, v_unpad = fused_pack_segmented_qkv(
            q_prefix, k_prefix, v_prefix, q, k, v, indices,
        )
    else:
        if all_valid:
            q_unpad, k_unpad, v_unpad = q, k, v
        else:
            q_unpad, k_unpad, v_unpad = fused_pack_qkv(q, k, v, indices)

    if bs == 1 or all_valid:
        # 单个 packed 序列即使 BCG bucket 有 padding 也是密集的；
        # 置空 cu_seqlens 让 FA3 走更快的 static persistent 调度
        dense_seq = indices.shape[0] if bs == 1 else seq
        out_dense = flash_attn_varlen_func(
            q=q_unpad.reshape(bs, dense_seq, *q_unpad.shape[-2:]),
            k=k_unpad.reshape(bs, dense_seq, *k_unpad.shape[-2:]),
            v=v_unpad.reshape(bs, dense_seq, *v_unpad.shape[-2:]),
            cu_seqlens_q=None,
            cu_seqlens_k=None,
            max_seqlen_q=dense_seq,
            max_seqlen_k=dense_seq,
            softmax_scale=self.softmax_scale,
            causal=False,
            ver=_fa_backend.fa_ver,
        )
        if all_valid:
            return out_dense
        # 需要把补齐的 mask 行写回 padded 布局
        return fused_scatter_to_padded(out_dense.flatten(0, 1), inv_indices, bs, seq)
```

## python/sglang/kernels/ops/diffusion/sites/qwen_image_added_qkv_site.py

新增 request-scoped 质量门控站点，将 fused added-QKV 的 mount/unmount 逻辑与 reject 原因封装，是 lossless/high 语义的关键基础设施。

```
# python/sglang/kernels/ops/diffusion/sites/qwen_image_added_qkv_site.py
# 将三个 BF16 文本投影 pack 成一个 GEMM 会改变归约结合序，
# 因此不是 bit-exact。默认 lossless 用三段独立切片 GEMM；
# quality=high 才 mount 单 GEMM 路径。

_FUSION = QualityGatedFusion(
    name="Qwen-Image fused added-QKV",
```

```
    marker_attr="_sgl_qwen_image_added_qkv_site",
    enabled_attr="_sgl_qwen_image_added_qkv_enabled",
)


def mark_qwen_image_added_qkv_site(module: nn.Module) -> None:
    # 标记未量化 Qwen-Image 注意力站点，初始为 unmounted
    _FUSION.mark(module)


def qwen_image_added_qkv_active(module: nn.Module) -> bool:
    # 返回当前请求是否挂载了 packed added-QKV GEMM
    return _FUSION.is_enabled(module)


def _site_reject_reason(site: nn.Module) -> str | None:
    # 只在"未量化 + 三段 packed"的站点上允许挂载
    linear = getattr(site, "to_added_qkv", None)
    if linear is None:
        return "missing to_added_qkv"
    if getattr(linear, "quant_config", None) is not None:
        return "quantized packed projection"
    if len(getattr(linear, "output_partition_sizes", ())) != 3:
        return "packed projection does not contain three shards"
    return None


def mount_qwen_image_added_qkv(root: nn.Module) -> bool:
    return _FUSION.mount(root, reject_reason=_site_reject_reason, logger=logger)


def unmount_qwen_image_added_qkv(root: nn.Module) -> None:
    _FUSION.unmount(root)
```

## 评论区精华

该 PR 的 review 评论为空，评论区的 4 条均为作者 BBuf 的状态说明：

1. ci-data 一致性 GT pin 修订：1ed919b73d 将 H100 一致性数据 pin 到 ci-data-diffusion 合并提交 883cf11，并澄清 88b17d4 捕获的是未合并的 auto-residency 实验（#35335 / #36703）输出，Sana 两阶段管线在 post-warmup 组件放置变化下并非数值不变，与 Qwen-Image 运行时改动无关，后续实验设置了 SanaWMPipelineConfig.supports_auto_residency = False 并回滚 pin。
2. 质量门控跟进：e5b6cdcbfd 将 fused added-QKV GEMM 挂到 request-scoped 的 quality=high，默认 lossless 用三段参考 GEMM，GB300 上 lossless/high/unmount 路径测试通过（23 个质量站点测试、14 个导入 /Qwen 选择测试）。
3. 两次 CI 重跑请求。

- ci-data 一致性 GT pin 修订与 Sana 误 pin 排查 (testing): 已解决：保留 Qwen fused-QKV GT，回滚 Sana 帧到 pre-88b17d4，后续实验设置 supports_auto_residency=False 并各自回滚 pin。
- fused added-QKV 的 quality 门控后续 (design): 已解决：PR body 中的性能 /SSIM 数字被明确标注为 high 路径结果。
- CI 失败重跑 (other): 已处理：CI 重跑完成（PR 最终 merged）。

# 风险与影响

- 风险：
  1. 正确性风险（BF16 归约序变化）：_split_unquantized_merged_linear 与 fused GEMM 的 chunk(3, dim=-1) 归约序不同，非 bit-exact。虽然默认 lossless 保持参考语义，但 quality=high 下输出与旧实现存在数值差异，需要 SSIM/ 一致性测试持续守护（已有 test_qwen_added_qkv_lossless_uses_three_reference_gemms）。
  2. TP 集体通信风险：group_coordinator.all_reduce 从私有 _all_reduce_impl 改为公开 custom_all_reduce，要求 V2 实现返回 None 时回退 NCCL；若某个实现返回 None 的时机与图捕获语义不兼容，可能影响 BCG 捕获路径。32 MiB 工作区对多模型共存时的显存占用有影响。
  3. 分段打包的 SP 限制：q_prefix 分段路径显式 raise NotImplementedError（不支持 SP/Ulysses），qwen_image.py 中仅非 sp_text_sharded 走该路径；若未来启用 SP，需完整实现分段 all-to-all，否则静默回退到 join_seqs 的语义也需要验证。
  4. FA3 static persistent 调度：bs == 1 时通过 cu_seqlens=None 选择静态调度器，对 indices.shape[0] == 0（全 False mask）仍有保护，但 BCG 捕获下 shape 变化可能触发重新捕获。
  5. ci-data GT pin 依赖：H100 一致性数据依赖外部 sgl-project/ci-data-diffusion 仓库提交，Sana 误 pin 事件表明 pin 管理存在跨 PR 相互污染的风险。
- 影响：影响范围集中在 sglang/multimodal_gen 与 sglang/kernels 的扩散模型运行路径：
  1. 任何 native CUDA 扩散模型 TP>1 运行时都会经过新的 custom-all-reduce 调度器和 32 MiB workspace（FLUX、Qwen-Image 系列、Z-Image、Ideogram、Cosmos3、MiniMax-H3 等）；
  2. 带 2-D mask 与 varlen 元数据的 FA 注意力受益于分段打包与 static persistent 调度（ERNIE-Image、Krea-2、Ideogram 4、LingBot、Z-Image、Qwen-Image 家族）；
  3. 只有 quality=high 的未量化 Qwen-Image 架构检查点使用 fused added-QKV GEMM。Qwen 家族整体并未全面加速：仅 T2I BCG 双卡路径明显（Qwen-Image-2512 +8.58%），Edit/Layered/FireRed 单卡路径为 ±0.5% 内中性。对团队而言，引入了 request-scoped 质量门控的第二个使用站点（此前已有规范模式 QualityGatedFusion），并形成"跨模型 ABBA 审计 + 负数对照"的验证方法论，值得后续性能 PR 借鉴。 - 风险标记：核心路径变更，非 bit-exact 优化需质量门控守护，TP 通信工作区扩容影响显存，分段前缀暂不支持 SP 路径，ci-data 外部 pin 依赖

# 关联脉络

- PR #37129 [Diffusion] Fuse Qwen-Image residual norm and NVFP4 quantization: 同属 Qwen-Image diffusion 性能优化线，修改同一文件 qwen_image.py 与 modelopt_quant.py，本 PR 合并后需保持 NVFP4 路径不受 fused added-QKV 影响。
- PR #37123 [Diffusion] Fuse Qwen-Image FP8 QKV projection and Blackwell epilogue: 同为 Qwen-Image QKV 投影融合方向（FP8 QKV），本 PR 的 unquantized added-QKV 融合与之互补，共享 qwen_image.py 与 modelopt_quant.py 改动上下文。
- PR #37141 [Diffusion] Fuse FLUX.2 token concatenation and NVFP4 quantization: 同属 diffusion 打包 / 拼接类优化，且共用 test_model_fast_paths.py 与 test_layout.py 等快速路径测试文件。
- PR #35177 feat(unified-memory): three sub-pools for mamba + hybrid-SWA models: 与本 PR 无直接功能关联，但同属 multimodal_gen/mem_cache 大规模重构线，且都引入新的契约与大规模测试配套，可作为仓库演进脉络参考。
- PR #37293 [Cookbook] Add DeepSeek-V4-Flash-Vision-Exp to the DeepSeek-V4 page: 同为文档 /cookbook 与模型配方更新，说明仓库在持续扩充 diffusion 与多模态配方文档。