PR #24287 完整报告

sgl-project/sglang

[Diffusion] Optimize Hunyuan3D shape denoising

合并时间：2026-05-06 10:10

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24287

执行摘要

- 一句话：融合 norm/QK norm 路径，提升 24.9% 去噪延迟
- 推荐动作：值得精读，特别是 _FluxRMSNorm.weight 属性的暴露模式（兼容 checkpoint 与 fused 接口）和 apply_qk_norm 的使用方式。建议后续在类似模型中推广此 fusion 模式。

功能与动机

Hunyuan3D shape denoising 中存在多个分离的 norm 和 modulation 操作，kernel launch 开销较高。通过融合这些操作为单一 kernel call，可显著降低 GPU 调度代价，同时保持模型数值精度。PR 提供了详细的基准测试结果，验证优化有效性。

实现拆解

1. 在 hunyuan3d.py 中从 runtime.layers.layernorm 导入新的 fused 组件：LayerNormScaleShift、ScaleResidualLayerNormScaleShift 和 apply_qk_norm。
2. 新增工具函数 _fused_add_gate，基于 torch.addcmul 实现残差与门控的融合计算，用于 double/single stream block 的 residual gate。
3. 改造 _FluxRMSNorm：添加 variance_epsilon 和 hidden_size 属性，并增加 weight 属性（返回 self.scale）以兼容 fused QK norm 接口。
4. 改造 _FluxQKNorm.forward：调用 apply_qk_norm 替代手动 norm，传入 contiguous 张量并允许 inplace 操作，减少数据拷贝。
5. 在 _FluxDoubleStreamBlock 和 _FluxSingleStreamBlock 中将 nn.LayerNorm 替换为 LayerNormScaleShift 和 ScaleResidualLayerNormScaleShift，实现 modulation 与 norm 的 kernel 融合。
6. 在 hunyuan3d_shape.py 的 Hunyuan3DShapeSaveStage.forward 中，将 OutputBatch 构造参数由 timings 改为 metrics，确保 shape-only 路径也能记录性能数据。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/hunyuan3d.py（模块 模型 DiT；类别 source；类型 core-logic；符号 _fused_add_gate, weight, _FluxRMSNorm, _FluxQKNorm）：核心模型实现，包含 norm fusion、QK norm 融合、residual gate 优化等主要变更，对性能提升贡献最大

- python/sglang/multimodal_gen/runtime/pipelines_core/stages/hunyuan3d_shape.py（模块 流水线阶段；类别 source；类型 core-logic）：shape-only 保存路径调整为传递 metrics 而非 timings，保留性能数据

关键符号：_fused_add_gate, _FluxRMSNorm.weight, _FluxQKNorm.forward, _FluxDoubleStreamBlock.init, _FluxSingleStreamBlock.init, Hunyuan3DShapeSaveStage.forward

关键源码片段

python/sglang/multimodal_gen/runtime/models/dits/hunyuan3d.py

核心模型实现，包含 norm fusion、QK norm 融合、residual gate 优化等主要变更，对性能提升贡献最大

```
def _fused_add_gate(
    residual: torch.Tensor, x: torch.Tensor, gate: torch.Tensor
) -> torch.Tensor:
    # 使用 addcmul 融合残差和门控，替代 residual + x * gate 的分离操作
    return torch.addcmul(residual, x, gate)


class _FluxRMSNorm(nn.Module):
    def __init__(self, dim: int):
        super().__init__()
        self.scale = nn.Parameter(torch.ones(dim))
        self.variance_epsilon = 1e-6 # 显式存储 epsilon，便于统一管理
        self.hidden_size = dim # 记录隐藏维度，兼容 fused 接口

    @property
    def weight(self) -> nn.Parameter:
        # 暴露 weight 属性，使得外部 fused QK-norm 可以通过标准接口访问 scale
        return self.scale

    def forward(self, x: torch.Tensor):
        x_dtype = x.dtype
        x = x.float()
        rrms = torch.rsqrt(
            torch.mean(x**2, dim=-1, keepdim=True) + self.variance_epsilon
        )
        return (x * rrms).to(dtype=x_dtype) * self.scale


class _FluxQKNorm(nn.Module):
    def __init__(self, dim: int):
        super().__init__()
        self.dim = dim
        self.query_norm = _FluxRMSNorm(dim)
        self.key_norm = _FluxRMSNorm(dim)
```

```
def forward(
    self, q: torch.Tensor, k: torch.Tensor, v: torch.Tensor
) -> Tuple[torch.Tensor, torch.Tensor]:
    # 使用 fused apply_qk_norm 替代逐个 norm 调用，
    # contiguous 确保内存布局，allow_inplace 允许就地更新减少拷贝
    q, k = apply_qk_norm(
        q=q.contiguous(),
        k=k.contiguous(),
        q_norm=self.query_norm,
        k_norm=self.key_norm,
        head_dim=self.dim,
        allow_inplace=True,
    )
    return q.to(v), k.to(v)
```

评论区精华

审核由 mickqian 批准，gemini-code-assist 汇总了重构内容。无公开争议或未解决疑虑。CI 全部通过。

- 暂无高价值评论线程

风险与影响

- 风险：主要风险来自新 fused 组件的正确性：LayerNormScaleShift 等可能对数值精度产生影响；但 PR 提供了输出对比（mp4 渲染）和性能数据，确认结果一致。此外，apply_qk_norm 的 allow_inplace=True 参数可能改变输入张量，但已通过 contiguous 确保安全。影响范围仅限 Hunyuan3D shape 模型，其他 diffusion 模型未受影响。
- 影响：用户侧：shape denoising 延迟降低 24.9%，端到端降低 6.7%，生成质量不变。系统侧：减少 kernel launch 次数，降低 GPU 占用。团队侧：新增的 fused 组件可复用于其他 diffusion 模型的优化。
- 风险标记：核心路径变更，缺少测试覆盖，inplace 操作风险

关联脉络

- 暂无明显关联 PR