PR #28304 完整报告

sgl-project/sglang

[perf] Use default torch compile mode for Wan2.2 T2V A14B

合并时间：2026-06-16 15:45

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28304

执行摘要

- 一句话：Wan2.2 T2V A14B 编译模式改为 default，提升 5% 性能
- 推荐动作：值得阅读，尤其是在扩散模型性能调优和配置抽象方面。此 PR 展示了如何将实验性优化（#28050 中的 default 模式）平滑推广到更多模型，并同时建立可扩展的配置机制。建议关注 _maybe_enable_torch_compile 的参数透传模式及 DiTConfig 的字段设计。

功能与动机

PR 延续了 #28050 的发现：default 编译模式相比 max-autotune-no-cudagraphs 在 diffusion 推理中具有更优的端到端性能。在 Wan2.2 T2V A14B 上，H100 4×GPU 验证 denoise 加快 5.70%、端到端加快 5.64%，peak memory 从 15.23 GB 降至 14.00 GB；B200 上也有相近收益。

实现拆解

1. 在 DiTConfig 基类添加 torch_compile_mode 字段：在 python/sglang/multimodal_gen/configs/models/dits/base.py 的 DiTConfig dataclass 中新增 torch_compile_mode 字段，默认值为 max-autotune-no-cudagraphs，并添加对应的 CLI 参数 --dit-config.torch-compile-mode，保留向后兼容。
2. 重构编译模式读取逻辑：修改 mova.py、denoising.py、paint.py 中 _maybe_enable_torch_compile 方法的模式解析逻辑：优先使用环境变量 SGLANG_TORCH_COMPILE_MODE，若未设置则从传入的 model_config（即 dit_config）中读取 torch_compile_mode，最后 fallback 至 max-autotune-no-cudagraphs。同时 _maybe_compile_dits 方法改为传递对应的模型配置对象。
3. 为 Wan2.2 T2V A14B 设置默认 default 模式：在 python/sglang/multimodal_gen/configs/pipeline_configs/wan.py 中将 Wan2_2_T2V_A14B_Config.dit_config.torch_compile_mode 设为 default。类似地，为 LTX 2.3 的 DiT 配置也做了相应设置。
4. 新增 LTX 2.3 pipeline 配置并修复注册：在 ltx_2.py 中新增 LTX23PipelineConfig 继承自 LTX2PipelineConfig，并在 registry.py 中将其注册到 Lightricks/LTX-2.3 模型路径下，同时更新导入。
5. 清理冗余环境变量 Fallback：从 pipeline_configs/base.py 中删除了不再需要的 SGLANG_TORCH_COMPILE_MODE 回退逻辑（-3 行）。

关键文件：

- python/sglang/multimodal_gen/configs/models/dits/base.py（模块 扩散模型；类别 source；类型 data-contract）：在 DiTConfig 基类中新增 torch_compile_mode 字段和对应 CLI 参数，是编译模式配置化的核心变更。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/mova.py（模块 扩散模型；类别 source；类型 data-contract；符号 _maybe_enable_torch_compile）：修改了核心函数 _maybe_enable_torch_compile 和 _maybe_compile_dits，实现从 model_config 读取编译模式。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/denoising.py（模块 扩散模型；类别 source；类型 core-logic）：在通用去噪阶段中同步更新编译模式读取逻辑，统一配置路径。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/hunyuan3d/paint.py（模块 扩散模型；类别 source；类型 data-contract）：在绘画管线初始化和编译部分同步更新编译模式获取逻辑。
- python/sglang/multimodal_gen/configs/pipeline_configs/wan.py（模块 扩散模型；类别 source；类型 core-logic）：为 Wan2.2 T2V A14B 显式设置 torch_compile_mode 为 'default'，是本次性能优化的直接体现。
- python/sglang/multimodal_gen/configs/models/dits/ltx_2.py（模块 扩散模型；类别 source；类型 data-contract）：为 LTX 2.3 的 DiT 配置设置默认编译模式（可能为 'default'）。
- python/sglang/multimodal_gen/configs/pipeline_configs/ltx_2.py（模块 扩散模型；类别 source；类型 core-logic；符号 LTX23PipelineConfig）：新增 LTX23PipelineConfig 配置类，支持 LTX 2.3 的特定设置。
- python/sglang/multimodal_gen/registry.py（模块 扩散模型；类别 source；类型 dependency-wiring）：修复 LTX 2.3 的 pipeline 注册，确保正确加载新配置类。

关键符号：_maybe_enable_torch_compile, _maybe_compile_dits, DiTConfig.add_cli_args

关键源码片段

python/sglang/multimodal_gen/configs/models/dits/base.py

在 DiTConfig 基类中新增 torch_compile_mode 字段和对应 CLI 参数，是编译模式配置化的核心变更。

```
@dataclass
class DiTConfig(ModelConfig):
    # ... 其他字段 ...
    # 新增字段：编译模式，默认值保持与之前环境变量一致
    torch_compile_mode: str = 'max-autotune-no-cudagraphs'

    @staticmethod
    def add_cli_args(parser: Any, prefix: str = 'dit-config') -> Any:
        # ... 原有参数 ...
        # 新增 CLI 参数，允许用户通过命令行覆盖编译模式
        parser.add_argument(
            f'--{prefix}.torch-compile-mode',
```

```
            type=str,
            dest=f'{prefix.replace("-", "_")}.torch_compile_mode',
            default=DiTConfig.torch_compile_mode,
            help='torch.compile mode for the DiT model',
        )
        return parser
```

python/sglang/multimodal_gen/runtime/pipelines_core/stages/model_specific_stages/mova.py

修改了核心函数 _maybe_enable_torch_compile 和 _maybe_compile_dits，实现从 model_config 读取编译模式。

```
def _maybe_enable_torch_compile(
    self,
    module: nn.Module,
    server_args: ServerArgs,
    model_config: object | None = None, # 新增参数，允许从模型配置读取编译模式
):
    # ... 平台检查和跳过逻辑 ...
    compile_kwargs: dict[str, object] = {'fullgraph': False, 'dynamic': None}
    else:
        # 优先使用环境变量，否则从 model_config 读取，最后回退到默认值
        mode = os.environ.get('SGLANG_TORCH_COMPILE_MODE') or getattr(
            model_config,
            'torch_compile_mode',
            'max-autotune-no-cudagraphs',
        )
        compile_kwargs['mode'] = mode
        logger.info('Compiling %s with mode: %s', module.__class__.__name__, mode)
    module.compile(**compile_kwargs)

def _maybe_compile_dits(self, server_args: ServerArgs):
    if self._torch_compiled or not server_args.enable_torch_compile:
        return
    module_configs = [
        (self.video_dit, server_args.pipeline_config.dit_config),
        (self.video_dit_2, server_args.pipeline_config.dit_config),
        (self.audio_dit, server_args.pipeline_config.audio_dit_config),
    ]
    for module, model_config in module_configs:
        if module is not None:
            self._maybe_enable_torch_compile(module, server_args, model_config)
    self._torch_compiled = True
```

评论区精华

本 PR 的 review 仅有一条 Approval 来自 mickqian，无额外讨论或请求变更。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 编译模式变更风险：*default* 模式在 H100/B200 上验证通过，但其他 GPU（如 A100）或非标准场景可能性能不同；但可通过环境变量或 CLI 降级。
  - 配置抽象完整性：*model_config* 参数可能为 None，代码中有 *getattr* 和默认 fallback，行为安全，但若 *model_config* 对象不包含 *torch_compile_mode* 属性也会回退，需确保所有传入了正确 *config*。
  - LTX 2.3 注册变更：新配置类继承自旧类，行为兼容，但可能影响显式引用 LTX2PipelineConfig 的代码，需注意。
  - 缺少测试覆盖：无新增测试用例，依赖现有 CI 验证。
- 影响：
  - 用户影响：Wan2.2 T2V A14B 和 LTX 2.3 用户将获得开箱即用的性能提升（约 5%）和显存降低；可通过 *--dit-config.torch-compile-mode* 或环境变量自定义编译模式。
  - 系统影响：无破坏性变更，所有其他 diffusion 模型行为不变。
  - 团队影响：统一了编译模式的配置入口，降低了维护成本。
  - 风险标记：编译模式变更，配置抽象完整性，LTX 2.3 注册变更，缺少测试覆盖

关联脉络

- PR #28050 Perf: Use *default* torch compile mode for diffusion models: 此 PR 延续了 #28050 中关于 *default* 编译模式性能更优的发现，并将其应用到 Wan2.2 T2V A14B。