PR #34173 完整报告

sgl-project/sglang

[diffusion] Make torch.compile opt-in for speed mode

合并时间：2026-08-10 10:22

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34173

执行摘要

- 一句话：speed 模式默认关闭 torch.compile，改为模型 opt-in
- 推荐动作：值得精读，尤其适合 diffusion 服务维护者。该 PR 是一个小体量但影响全局默认策略的变更：以 E2E 数据推翻「speed 即 compile」的直觉假设，并保留三层优先级（显式参数 > 模型部署配置 > 全局默认）。建议关注 adjust_based_on_performance_mode 中「模型 opt-in 且用户未显式传参」的生效条件，以及后续模型逐渐补齐 opt-in 的演进方向。

功能与动机

PR body 明确指出：performance_mode=speed 此前会对所有未显式退出的 diffusion pipeline 启用 torch.compile，但 H100 E2E 覆盖显示这不是可靠默认——12 个可编译 preset 回退、1 个持平、仅 4 个提升。典型数据如 qwen 首延迟 8.534s -> 14.764s（+73.0%）、zimage 0.513s -> 2.578s（+402.5%）、flux2-klein 0.259s -> 1.845s（+612.4%）。作者还强调，少数获胜 preset 与其他未验证 checkpoint 共享 pipeline 配置，因此不应基于孤立结果扩大默认开启范围。

实现拆解

1. 数据契约调整：在 python/sglang/multimodal_gen/configs/pipeline_configs/model_deployment_config.py 中，ModelDeploymentConfig 的字段 speed_mode_enable_torch_compile_by_default 默认值从 True 改为 False，并同步更新注释，说明 torch.compile 对已高度优化的 diffusion 内核可能反而更慢，因此改为模型 opt-in。
2. 执行逻辑保持不变：python/sglang/multimodal_gen/runtime/server_args/auto_tune.py 的 adjust_based_on_performance_mode 中 speed 分支的判断逻辑本身未改动——仍然只有「模型部署配置 opt-in 且用户未显式传参」时才自动开启 compile，因此默认值翻转即可生效；显式 --enable-torch-compile true/false 的优先级不受影响。
3. 测试配套：python/sglang/multimodal_gen/test/unit/test_server_args.py 中，原 test_speed_mode_enables_torch_compile_by_default 改写为 test_speed_mode_keeps_torch_compile_off_by_default，断言默认关闭；原 test_speed_mode_preserves_explicit_torch_compile_off 扩展为 test_speed_mode_preserves_explicit_torch_compile_setting，用子测试覆盖 False/True 两种显式传参；新增 test_speed_mode_honors_model_torch_compile_opt_in，通过

mock get_model_deployment_config 返回 ModelDeploymentConfig(speed_mode_enable_torch_compile_by_default=True) 验证模型 opt-in 钩子仍然生效。

4. 文档配套：docs/docs/sglang-diffusion/api/cli.mdx 与 docs/docs/sglang-diffusion/deployment_cookbook.mdx 更新了 performance-mode 的说明，明确 speed 模式默认保持 eager，模型可通过部署配置 opt-in，用户可用 --enable-torch-compile true 显式开启。

关键文件：

- python/sglang/multimodal_gen/configs/pipeline_configs/model_deployment_config.py（模块 部署配置；类别 source；类型 data-contract；符号 ModelDeploymentConfig, speed_mode_enable_torch_compile_by_default）：数据契约核心：speed_mode_enable_torch_compile_by_default 默认值从 True 翻转为 False，是本次策略变更的根因所在；该字段被 auto_tune.py 在 speed 模式下发判断。
- python/sglang/multimodal_gen/runtime/server_args/auto_tune.py（模块 自动调优；类别 source；类型 core-logic；符号 adjust_based_on_performance_mode）：speed 模式下发判断的执行入口 adjust_based_on_performance_mode；逻辑未变但注释明确「仅验证过收益的模型 opt-in」，是策略落地位置。
- python/sglang/multimodal_gen/test/unit/test_server_args.py（模块 参数测试；类别 test；类型 test-coverage；符号 test_speed_mode_keeps_torch_compile_off_by_default, test_speed_mode_preserves_explicit_torch_compile_setting, test_speed_mode_honors_model_torch_compile_opt_in, test_speed_mode_uses_minimax_h3_compile_policy）：完整的测试配套，覆盖默认关闭、显式传参保留、模型 opt-in 三条路径，是行为契约的守护。
- docs/docs/sglang-diffusion/api/cli.mdx（模块 接口文档；类别 docs；类型 documentation）：更新 CLI 文档中 --performance-mode 的语义说明，避免用户误解 speed 模式默认 compile。
- docs/docs/sglang-diffusion/deployment_cookbook.mdx（模块 部署文档；类别 docs；类型 documentation）：部署 cookbook 中 speed 模式表格与 auto 模式说明同步更新，明确 eager 默认与 opt-in 路径。

关键符号：adjust_based_on_performance_mode, get_model_deployment_config, test_speed_mode_keeps_torch_compile_off_by_default, test_speed_mode_preserves_explicit_torch_compile_setting, test_speed_mode_honors_model_torch_compile_opt_in

关键源码片段

python/sglang/multimodal_gen/configs/pipeline_configs/model_deployment_config.py

数据契约核心：speed_mode_enable_torch_compile_by_default 默认值从 True 翻转为 False，是本次策略变更的根因所在；该字段被 auto_tune.py 在 speed 模式下发判断。

```
"""模型部署配置：决定模型在部署时的默认策略。"""

from dataclasses import dataclass
```

```
from typing import Literal

OffloadComponentName = Literal["dit", "text_encoder", "image_encoder", "vae"]


@dataclass(frozen=True)
class ModelDeploymentConfig:
    auto_dit_layerwise_offload: bool = False
    auto_dit_layerwise_offload_high_memory_disable_gb: float | None = None
    keep_resident_min_available_gb: float | None = None
    # 仅 VAE 常驻：体积小，对内存影响可忽略；大编码器仍走 offload 策略
    keep_resident_components: tuple[OffloadComponentName, ...] = ("vae",)
    fsdp_auto_min_available_memory_gb: float | None = None
    fsdp_auto_requires_cfg: bool = True
    fsdp_auto_requires_default_parallelism: bool = True
    auto_enable_cfg_parallel: bool = True
    # degree 1 表示禁用 CFG 并行，把 GPU 留给序列并行使用
    auto_cfg_parallel_degree_by_num_gpus: tuple[tuple[int, int], ...] = ()
    # torch.compile 改为模型 opt-in：diffusion 内核已被高度优化时，
    # compile 可能比 eager 更慢，因此默认关闭，仅验证过收益的模型开启
    speed_mode_enable_torch_compile_by_default: bool = False
    supports_cfg_parallel: bool = True

    def get_auto_cfg_parallel_degree(self, num_gpus: int) -> int:
        for candidate_num_gpus, cfg_degree in self.auto_cfg_parallel_degree_by_num_gpus:
            if candidate_num_gpus == num_gpus:
                return cfg_degree
        return 2
```

python/sglang/multimodal_gen/runtime/server_args/auto_tune.py

speed 模式下发判断的执行入口 adjust_based_on_performance_mode；逻辑未变但注释明确「仅验证过收益的模型 opt-in」，是策略落地位置。

```
def adjust_based_on_performance_mode(self) -> None:
    """根据 performance_mode 调整服务参数（speed / memory / auto）。"""
    args = self.server_args
    args.performance_mode = self._normalize_performance_mode()

    if current_platform.is_cpu():
        return

    if args.performance_mode == "speed":
        logger.info("Applying performance_mode=speed")
        # 生效条件：模型部署配置 opt-in，且用户未显式设置该参数
        if (
            self._deployment_config().speed_mode_enable_torch_compile_by_default
            and not args.enable_torch_compile
            and not args.is_arg_explicitly_set("enable_torch_compile")
        ):
```

```
            # 只有验证过 compile 收益的模型才在这里被默认开启
            args.enable_torch_compile = True
            logger.info(
                "performance_mode=speed enables torch.compile "
                "(pass --enable-torch-compile false to opt out)"
            )
        elif not args.enable_torch_compile and not args.is_arg_explicitly_set(
            "enable_torch_compile"
        ):
            # 默认保持 eager，用户可显式 --enable-torch-compile true 加入
            logger.info(
                "performance_mode=speed keeps torch.compile disabled for "
                "this model (pass --enable-torch-compile true to opt in)"
            )
        # ... 后续 FSDP / 常驻内存策略与本次变更无关，保持不变
```

python/sglang/multimodal_gen/test/unit/test_server_args.py

完整的测试配套，覆盖默认关闭、显式传参保留、模型 opt-in 三条路径，是行为契约的守护。

```
    def test_speed_mode_keeps_torch_compile_off_by_default(self):
        args = self._from_dict_with_pipeline_config(
            QwenImagePipelineConfig(),
            kwargs={
                "model_path": "Qwen/Qwen-Image",
                "performance_mode": "speed",
            },
        )
        # 默认策略翻转：speed 模式不再自动开启 torch.compile
        self.assertFalse(args.enable_torch_compile)


    def test_speed_mode_preserves_explicit_torch_compile_setting(self):
        for enabled in (False, True):
            with self.subTest(enabled=enabled):
                args = self._from_dict_with_pipeline_config(
                    QwenImagePipelineConfig(),
                    kwargs={
                        "model_path": "Qwen/Qwen-Image",
                        "performance_mode": "speed",
                        "enable_torch_compile": enabled,
                    },
                )
                # 用户显式传参优先级最高，不受默认值翻转影响
                self.assertEqual(args.enable_torch_compile, enabled)


    def test_speed_mode_honors_model_torch_compile_opt_in(self):
        with patch.object(
            QwenImagePipelineConfig,
```

```
            "get_model_deployment_config",
            return_value=ModelDeploymentConfig(
                speed_mode_enable_torch_compile_by_default=True
            ),
        ):
            args = self._from_dict_with_pipeline_config(
                QwenImagePipelineConfig(),
                kwargs={
                    "model_path": "Qwen/Qwen-Image",
                    "performance_mode": "speed",
                },
            )
        # 模型通过部署配置 opt-in 后，speed 模式仍会开启 compile
        self.assertTrue(args.enable_torch_compile)
```

评论区精华

该 PR 没有产生实质性的 review 交锋：评审人 BBuf 直接 APPROVED 且未留下文字评论；两个 issue 评论均为作者触发的 /tag-and-rerun-ci CI 重跑指令。决策依据集中在 PR body 中的 H100 E2E 数据——作者明确指出 12 个可编译 preset 回退、仅 4 个受益，并解释获胜 preset 与未验证 checkpoint 共享 pipeline 配置，因此不扩大默认开启范围。这体现了「默认保守、按部署配置精确 opt-in」的决策原则。

- review 无实质讨论，仅 CI 重跑 (other): 策略变更无争议，单 approve 合入

风险与影响

- 风险：
  - 默认行为变更：所有使用 performance_mode=speed 的 diffusion 部署在升级后将默认走 eager。对原本受益于 compile 的模型（如 wan-ti2v、ideogram4-fp8），若用户不显式传参，会丢失 compile 带来的收益。
  - 模型 opt-in 覆盖不足：目前仅在测试中 mock 了模型钩子场景；真实 pipeline config 中哪些模型已设置 speed_mode_enable_torch_compile_by_default=True 需要逐一核对，否则原本受益的 preset 也需用户手动开启。
  - 回归风险低：PR 未改动任何模型 forward 或 kernel 语义，精度路径不变。
  - 测试深度有限：新测试只覆盖 ServerArgs 解析层，未覆盖真实 pipeline 是否按配置走 eager/compile 路径的端到端验证。
- 影响：
  - 用户侧：speed 模式默认免去编译冷启动开销，对 zimage、flux2-klein、qwen 等此前严重回退的 preset 可显著降低首延迟；需要 compile 的用户可显式传入 --enable-torch-compile true。
  - 系统侧：默认资源配置与文档描述同步更新，performance-mode=speed 的语义从「默认 compile」变为「默认 eager、按模型 opt-in」。
  - 团队侧：后续为 diffusion 模型添加 compile 支持时必须先在目标部署配置上完成收益验证，再修改 ModelDeploymentConfig，形成规范化的性能治理流程。

- 风险标记：默认行为变更，影响全部 diffusion speed 模式用户，模型 opt-in 覆盖待验证，eager 与 compile 收益因模型而异

关联脉络

- PR #34172 [diffusion] LTX-2 quality=high fused RMSNorm+modulate + FFN GELU epilogue (H200 ltx23-one-stage denoise 45.85->43.24 s, ~matches torch.compile): 同属 diffusion 性能路径：通过融合 kernel 获得收益而不依赖 torch.compile，与本次默认关闭 compile 的策略互为印证。
- PR #34174 [diffusion] BCG: auto-capture the default warmup resolution instead of hard-requiring --warmup-resolutions (H200 SANA denoise 0.73->0.457 s with a single flag): 同属 diffusion serving 参数与性能策略调整，体现 diffusion 性能治理的持续迭代方向。