

PR #50841 完整报告

vllm-project/vllm

[CPU] Enable tcmalloc for s390x

合并时间: 2026-08-05 10:55

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50841>

执行摘要

- 一句话: 为 s390x 平台启用 tcmalloc, 优化内存分配
- 推荐动作: 该 PR 值得快速浏览, 作为 "平台白名单扩展" 的典型示例: 展示了如何用最小改动 (追加枚举值) 为架构补齐既有能力。不建议精读实现细节, 但值得关注两个决策: 一是 `should_bundle_tcmalloc` 与运行时 `LD_PRELOAD` 的两段式设计 (构建期打包 + 运行期发现), 二是 Dockerfile 多阶段构建安装 `gperftools` 的做法。若团队有 s390x 用户, 建议后续补充一条构建期断言, 确保 tcmalloc 确实被找到并打包。

功能与动机

PR body 明确说明目的是 "Enable tcmalloc for s390x"。此前 tcmalloc 的 `LD_PRELOAD` 与打包逻辑只覆盖 ARM 和 X86, s390x 作为受支持的 CPU 架构被遗漏, 导致该平台无法享受 tcmalloc 降低内存分配开销的收益 (对应 `cpu.py` 中 "`LD_PRELOAD libtcmalloc, bundled under vllm/libs to reduce memory allocation overhead`" 的注释意图)。

实现拆解

本 PR 从构建、运行时、容器镜像和文档四个层面将 s390x 纳入 tcmalloc 支持范围, 改动小而直接:

1. 运行时启用 tcmalloc 预加载 (`vllm/platforms/cpu.py`): 在 `check_and_update_config` 中, 将 `CpuArchEnum.S390X` 加入 `cpu_architecture in (CpuArchEnum.ARM, CpuArchEnum.X86, ...)` 的判断, 使 s390x 上启动 vLLM 时也会从 `vllm/libs` 目录查找 `libtcmalloc*.so*` 并写入 `LD_PRELOAD`。该逻辑是条件式的, 只有找到对应 `.so` 才会生效, 不会破坏现有 x86/ARM 路径。
2. 构建期绑定 tcmalloc (`setup.py`): 在 `should_bundle_tcmalloc()` 的 `platform.machine()` 白名单中加入 `s390x`, 使 s390x 构建 wheel 时也执行 tcmalloc 的发现与打包逻辑 (通过 `find_tcmalloc` 从 `ldconfig` 输出中查找系统 `libtcmalloc`)。
3. 容器镜像支持 (`docker/Dockerfile.s390x`): 新增多阶段构建目标 `gperftools-build`, 下载 `gperftools 2.16` 源码并以 `--enable-minimal` 配置编译; 在最终镜像中通过绑定挂载安装 `libLTLIBRARIES` 并执行 `ldconfig`, 最后设置 `ENV LD_PRELOAD=/usr/local/lib/libtcmalloc_minimal.so.4`。
4. 安装文档 (`docs/getting_started/installation/cpu.s390x.inc.md`): 新增 warning 小节, 说明从源码构建时需要自行编译 `gperftools` 并设置 `LD_PRELOAD`, 同时说明 Docker 镜

像已内置该配置。

测试配套：本次改动未新增任何测试文件；验证仅依赖 PR body 中在 IBM s390x 环境（podman 容器）运行 [facebook/opt-125m](#) 推理的日志，日志显示服务正常启动并完成 warmup 与 KV cache 初始化。

关键文件：

- `vllm/platforms/cpu.py`（模块 平台层；类别 source；类型 core-logic；符号 `check_and_update_config`）：核心运行时改动：在 CPU 平台初始化时，将 s390x 加入 `LD_PRELOAD tcmalloc` 的架构白名单，是本次功能生效的入口。
- `setup.py`（模块 构建脚本；类别 source；类型 core-logic；符号 `should_bundle_tcmalloc`）：构建期决定是否绑定 `tcmalloc` 到 `wheel` 中，s390x 加入白名单后，源码构建会在 s390x 上查找并打包系统 `libtcmalloc`。
- `docker/Dockerfile.s390x`（模块 容器镜像；类别 infra；类型 infrastructure）：容器镜像侧配套：多阶段构建 `gperftools` 并设置 `LD_PRELOAD`，让 s390x Docker 用户开箱即用 `tcmalloc`。
- `docs/getting_started/installation/cpu.s390x.inc.md`（模块 安装文档；类别 docs；类型 documentation）：为源码安装用户补充 `tcmalloc` 手动编译与 `LD_PRELOAD` 配置指引，保证非 Docker 场景也能受益。

关键符号：`check_and_update_config`, `should_bundle_tcmalloc`

关键源码片段

`vllm/platforms/cpu.py`

核心运行时改动：在 CPU 平台初始化时，将 s390x 加入 `LD_PRELOAD tcmalloc` 的架构白名单，是本次功能生效的入口。

```
# vllm/platforms/cpu.py 中 check_and_update_config 的关键分支
# 该逻辑在 CPU 平台初始化时执行：先处理 libgomp 的 LD_PRELOAD，
# 再处理 tcmalloc。此处把 S390X 加入白名单，使 s390x 与 ARM/X86
# 一样享受内存分配优化。
if (
    platform.system() == "Linux"
    and cpu_architecture
    in (CpuArchEnum.ARM, CpuArchEnum.X86, CpuArchEnum.S390X) # 新增 S390X
    and "libtcmalloc" not in ld_preload_str
):
    vllm_pkg = os.path.dirname(os.path.dirname(__file__))
    tcmalloc_so = None
    # 在 vllm 包内查找打包进来的 tcmalloc 动态库，
    # 找不到时静默跳过，保证缺失场景下不阻断启动。
    for pattern in ("libtcmalloc_minimal*.so*", "libtcmalloc.so*"):
        tcmalloc_so_candidates = glob.glob(
            os.path.join(vllm_pkg, "libs", pattern)
        )
        if tcmalloc_so_candidates:
```

```

tcmalloc_so = tcmalloc_so_candidates[0]
break

if tcmalloc_so is not None:
    # 把 tcmalloc 放在 LD_PRELOAD 最前面, 确保其分配器优先生效
    if ld_preload_str:
        ld_preload_str = f"{tcmalloc_so}:{ld_preload_str}"
    else:
        ld_preload_str = tcmalloc_so
    os.environ["LD_PRELOAD"] = ld_preload_str

```

setup.py

构建期决定是否绑定 tcmalloc 到 wheel 中, s390x 加入白名单后, 源码构建会在 s390x 上查找并打包系统 libtcmalloc。

```

# setup.py 中决定是否将 tcmalloc 捆绑进 wheel 的判断函数
# 只有 CPU 目标 + Linux + 白名单架构才会执行 tcmalloc 发现与打包。
def should_bundle_tcmalloc() -> bool:
    import platform

    return (
        VLLM_TARGET_DEVICE == "cpu"
        and sys.platform.startswith("linux")
        # 新增 s390x: 此前仅 aarch64/x86_64 会捆绑 tcmalloc
        and platform.machine() in ("aarch64", "x86_64", "s390x")
    )

```

docker/Dockerfile.s390x

容器镜像侧配套: 多阶段构建 gperftools 并设置 LD_PRELOAD, 让 s390x Docker 用户开箱即用 tcmalloc。

```

# docker/Dockerfile.s390x 中新增的 gperftools 构建阶段
# 独立 stage 避免污染最终镜像的构建工具链
FROM base AS gperftools-build
WORKDIR /tmp
ARG GPERFTOOLS_VERSION=2.16
# 从官方 release 拉取源码, --enable-minimal 只构建 minimal 版 (够用且体积小)
RUN curl -LO https://github.com/gperftools/gperftools/releases/download/gperftools-
    ${GPERFTOOLS_VERSION}/gperftools-${GPERFTOOLS_VERSION}.tar.gz && \
    tar -xzf gperftools-${GPERFTOOLS_VERSION}.tar.gz && \
    cd gperftools-${GPERFTOOLS_VERSION} && \
    ./configure --enable-minimal && \
    make -j$(nproc)

# 最终镜像: 安装 libLTLIBRARIES (即 libtcmalloc_minimal.so.4) 并刷新动态链接缓存
RUN --mount=type=bind,from=gperftools-build,src=/tmp/gperftools-2.16,target=/gperftools \
    make -C /gperftools install-libLTLIBRARIES && \
    /sbin/ldconfig

# 预设 LD_PRELOAD, 使容器内所有进程默认使用 tcmalloc 分配器

```

ENV LD_PRELOAD=/usr/local/lib/libtcmalloc_minimal.so.4

评论区精华

本次 PR 几乎没有实质性技术讨论：

Reviewer [claude\[bot\]](#): "This pull request is from a fork — automated review is disabled."

Reviewer [bigPYJ1151](#): 直接 APPROVED（无评论）。

维护者先后触发了 3 次 Buildkite CI（#81960、#82142、#82260），分别在两次 merge main 之后重跑，但评论中未出现针对实现细节的质疑或设计权衡讨论。可以推断该改动属于平台白名单扩展，reviewer 判断风险较低，因此快速合入。

- 无实质技术讨论，机器人跳过审查后由维护者直接批准 (other): 无技术争议，改动按原样合入。

风险与影响

• 风险：

1. 缺少测试覆盖：没有为 s390x 的 tcmalloc 路径新增任何单元测试或 CI 验证。
should_bundle_tcmalloc 和 cpu.py 中的白名单分支在 s390x 上是否按预期工作，依赖真实环境验证；若 s390x 轮子构建机未安装 libtcmalloc，find_tcmalloc 会静默返回 None，导致打包时静默跳过，行为与预期不一致但无报错。
2. Dockerfile 的 LD_PRELOAD 覆盖：ENV LD_PRELOAD=/usr/local/lib/libtcmalloc_minimal.so.4 是硬编码设置，会覆盖用户在运行容器时通过 -e LD_PRELOAD=... 传入的自定义值（ENV 指令对同名字段取优先级最低的默认值），可能影响用户自定义预加载行为。
3. 大端平台兼容性：s390x 是大端架构，日志中已出现 PyTorch 关于 checkpoint 字节序的警告。tcmalloc 在内存分配层无字节序问题，但与其他库（如 protobuf，文档中已有 workaround）的交互仍需在真实推理负载中观察。
4. 回归面控制：由于所有判断均为条件式且只追加枚举值，x86/ARM 路径不受影响，回归风险低。

• 影响：

- 用户影响：s390x（IBM Linux on Z）用户使用 Docker 镜像或自建 wheel 时将自动获得 tcmalloc 预加载，KV cache 和权重分配的内存开销有望降低；源码安装用户需按新增文档手动配置。
- 系统影响：修改集中在平台启动路径与构建脚本，对默认（x86_64/aarch64）部署零影响；对 vLLM 的 s390x 支持链是一次能力补齐。
- 团队影响：改动小、无测试配套，后续合入后需依赖社区 s390x 用户反馈来验证实际收益；维护者需关注是否需要在 CI 中补充 s390x 的 smoke test。
- 风险标记：缺少测试覆盖，平台专用启用，环境变量修改

关联脉络

- PR #50915 [Bugfix][CPU] Fix macOS build: std::sqrt is not constexpr under libc++: 同属 CPU 平台构建 / 编译基础设施修复，关联较弱，但反映出 vLLM 对不同 CPU 架构（s390x、macOS）构建兼容性的持续关注。