

PR #23995 完整报告

sgl-project/sglang

[connector] Add Azure Blob Storage connector (az:// and *.blob.core.windows.net URLs)

合并时间: 2026-05-12 05:20

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/23995>

执行摘要

- 一句话: 新增 Azure Blob Storage 连接器, 支持 az:// 和 blob.core.windows.net 协议
- 推荐动作: 建议 merge, 本 PR 实现干净、遵循现有模式、可维护。后续可补充针对 blobfile mock 的单元测试, 以及更新文档以反映支持 Azure URL。值得关注的设计决策是: lazy import 模式已作为处理可选依赖的标准在 S3 connector 中应用, 本 PR 进一步验证了该模式的扩展性。

功能与动机

PR 描述指出: 'sglang.srt.connector.create_remote_connector currently only supports s3://, redis://, and remote-instance URLs. Azure Blob Storage URLs (az:// and https://.blob.core.windows.net/) raise ValueError: Invalid connector type', 这阻碍了 Azure 用户直接加载模型权重。本 PR 提供第一类支持, 允许 Azure 用户直接使用 --model-path az://...。

实现拆解

1. 新增 Azure 连接器文件: 创建 `python/sglang/srt/connector/azure.py`, 定义 `AzureBlobConnector` 类, 继承 `BaseFileConnector`。类初始化时 lazy-import `blobfile`, 实现 `glob`、`pull_files`、`weight_iterator` 等方法。`weight_iterator` 通过 `pull_files` 将 `*.safetensors` 下载到本地, 再用 `runai_safetensors_weights_iterator` 读取。
2. 修改调度逻辑: 在 `python/sglang/srt/connector/__init__.py` 中新增 `_is_azure_blob_url` 辅助函数, 检测 `az://` 和 `https://*.blob.core.windows.net/*` 格式。在 `create_remote_connector` 中插入 `elif` 分支, 在识别到 Azure URL 时 lazy-import `AzureBlobConnector` 并返回实例。
3. 保持可选依赖: `blobfile` 仅在 `AzureBlobConnector` 中被 import, `_is_azure_blob_url` 中也不直接 import, 因此不安装 `blobfile` 不会影响现有功能。用户若使用 Azure URL 时会收到明确的 `ImportError` 指引。
4. 文件过滤工具: 同时提供了 `_filter_allow` 和 `_filter_ignore` 两个辅助函数, 基于 `fnmatch` 实现模式过滤, 被 `list_files` 调用。

关键文件:

- `python/sglang/srt/connector/azure.py` (模块 连接器; 类别 source; 类型 core-logic; 符号 `AzureBlobConnector`, `init`, `glob`, `pull_files`): 核心新增文件, 定义了

AzureBlobConnector 类，实现所有 Azure 连接逻辑。

- python/sclang/srt/connector/__init__.py (模块 连接器; 类别 source; 类型 core-logic; 符号 _is_azure_blob_url) : 修改调度入口, 新增 _is_azure_blob_url 函数, 在 create_remote_connector 中添加 Azure 路由分支。

关键符号: AzureBlobConnector.init, AzureBlobConnector.glob,
AzureBlobConnector.pull_files, AzureBlobConnector.weight_iterator,
AzureBlobConnector.close, list_files, _filter_allow, _filter_ignore, _normalize_url,
_is_azure_blob_url

关键源码片段

python/sclang/srt/connector/__init__.py

修改调度入口, 新增 _is_azure_blob_url 函数, 在 create_remote_connector 中添加 Azure 路由分支。

```
# python/sclang/srt/connector/__init__.py 中的新增部分

# 检测 Azure Blob Storage URL: az:// 或 blob.core.windows.net 的 HTTPS URL
def _is_azure_blob_url(url: str, connector_type: str) -> bool:
    """判断是否为 Azure Blob Storage 支持的 URL 格式。"""
    if connector_type == "az":
        return True
    return connector_type == "https" and ".blob.core.windows.net" in url

# 在 create_remote_connector 中增加分支
def create_remote_connector(url, device=None, **kwargs) -> BaseConnector:
    connector_type = parse_connector_type(url)
    if connector_type == "redis":
        return RedisConnector(url)
    elif connector_type == "s3":
        return S3Connector(url)
    elif connector_type == "instance":
        return RemoteInstanceConnector(url, device)
    elif _is_azure_blob_url(url, connector_type):
        # 延迟导入, 保持 blobfile 为可选依赖
        from sclang.srt.connector.azure import AzureBlobConnector
        return AzureBlobConnector(url)
    else:
        raise ValueError(f"Invalid connector type: {url}")
```

评论区精华

没有 review 评论或讨论。PR body 中作者邀请 ByronHsu 和 zhyncs 审核, 但未见实际评论。一个 bot 评论提及每日配额耗尽, 不构成技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更仅影响权重加载的 I/O 路径，模型前向逻辑完全不变。主要风险点：
 - 1) blobfile 库在特定 Azure 环境下可能存在认证兼容性问题，但已在文档中说明使用标准凭证链；
 - 2) 没有测试覆盖新增的 connector，可能遗漏边界情况（如网络错误、空目录）；
 - 3) weight_iterator 先将所有 safetensors 下载到本地，对大型模型可能增加临时磁盘占用。
- 影响：正面影响：Azure 用户现在可以直接使用 Azure Blob Storage URL 加载模型，无需预下载或外挂补丁。无负面影响：blobfile 是可选依赖，不安装不影响现有功能；调度逻辑的改动仅增加一个 elif 分支，不影响原有流程。对非 Azure 用户完全透明。
- 风险标记：缺少测试覆盖，引入新可选依赖，临时磁盘占用

关联脉络

- 暂无明显关联 PR