conda Package for ABACUS-GPU

Table of Contents

1. 用户使用指南

1. 面向需求

- 为拓广conda Package对不同硬件的兼容性，同时保持conda一键部署的特点。
- 为习惯conda的用户提供快速体验ABACUS GPU版本的渠道。
- 目前有两个版本的conda Package：
- 一、本机兼容版本：适合机器配有NVIDIA GPU，并装有cuda11.8版本的用户进行ABACUS-GPU的直接部署。
- 二、从头构建版本：适合机器配有NVIDIA GPU，对于[没有安装CUDA\没有正确安装CUDA\安装了CUDA]的用户。conda将采用cudatoolkit-dev软件包，帮助自动化进行CUDA配置。配置后的cuda版本为11.7（cudatoolkit-dev的最新版本）

2. 使用conda安装GPU版本的ABACUS

```
1  ---------------------------------------------------------------------------
2  wget https://repo.anaconda.com/archive/Anaconda3-2023.03-Linux-x86_64.sh
3  # done
4  ---------------------------------------------------------------------------
5  chmod u+x Anaconda3-2023.03-Linux-x86_64.sh
6  sudo bash Anaconda3-2023.03-Linux-x86_64.sh
7  # interactive installation
8  # installation complete
9  exit
10 #exit current shell & re-login
11 ---------------------------------------------------------------------------
12 # 1 - get package
13   ---to do
14 # 2 - create environment with this package
15       conda create -n abacus_gpu_env abacus -c deepmodeling -c conda-forge
16 # done
```

```
----------------------------------------------------------------------------
conda activate abacus_gpu_env
which abacus
# begin using abacus #
conda deactivate
----------------------------------------------------------------------------
```

3. NVIDIA驱动与CUDA的配置

- NVIDIA驱动需要安装在你的机器上，版本不低于：450.80.02
- Check by `nvidia-detector` or `lspci` `lshw`
- CUDA 12.1 Release Notes

4. 已知限制

- 无法使用DeePKS.
- 本机兼容版本要求CUDA版本为11.8；从头构建版本使用的CUDA版本固定为11.7.
- CMake编译模式为Release模式。

2. Package构建过程

2.1 NVDIA驱动及CUDA环境安装

- conda-forge中的cudatoolkit是CUDA的子集，而从CUDA官网下载脚本运行安装or从conda-forge安装cudatoolkit-dev是安装完整CUDA（不包括驱动）

> cudatoolkit-dev:
>
> The toolkit includes GPU-accelerated libraries, debugging and optimization tools, a C/C++ compiler and a runtime library to deploy your application. This package consists of a post-install script that downloads and installs the full cuda toolkit(compiler, libraries, **with the exception of cuda drivers**).
>
> Post-install详解
>
> cudatoolkit:
>
> This CUDA Toolkit includes GPU-accelerated libraries, and the CUDA runtime for the Conda ecosystem. For the full CUDA Toolkit with a compiler and development tools visit https://developer.nvidia.com/cuda-downloads

2.2 Git Clone ABACUS 源码

进入conda目录对元数据进行更改。两个文件分别为meta.yaml和conda_build_config.yaml。

再回到abacus-develop目录进行conda build。

```
git clone https://github.com/deepmodeling/abacus-develop
```

2.3 机器配置与元数据更改

a. CXX compiler identification：

conda_build_config.yaml中指定cxx_compiler_version:

```
cxx_compiler_version:
 - 10.4.0 #earlier than 11.0
 c_compiler_version:
 - 10.4.0
```

做到指定gcc和g++的版本，原因是CUDA版本与gcc版本存在依赖关系。gcc/g++/c++ version 与 CUDA version之间的关系

b. Conda_build_config.yaml中对Cuda Compiler设置

```
  cuda_compiler:
  - nvcc
```

c. meta.yaml中requirements:build:

```
  - {{ compiler('cuda') }}
```

d. ABACUS的GPU版本编译：meta.yaml中build:script:

```
cmake -B conda_build ${CMAKE_ARGS} -DCMAKE_BUILD_TYPE=Release -DUSE_CUDA=1  -DENABLE_NATIVE_OPTIMIZATION=1
```

2.4 构建ABACUS包与创建环境

```
conda build . -c conda-forge
```

```
conda create -n my_abacus_env abacus -c local -c conda-forge
```

2.5 相关文档

Anaconda compiler tools – conda-build 3.24.0+0.g30af5caa.dirty documentation

2.6 Recipe

1. 本机兼容版本

```
# Install ABACUS by conda:
#   conda create -n abacus_env abacus -c deepmodeling -c conda-forge
#   conda activate abacus_env
# Docs: https://abacus.deepmodeling.com/en/latest/quick_start/easy_install.html#

# This conda package can also be built locally with
#   conda build . -c conda-forge
# And install with
#   conda create -n my_abacus_env abacus -c local -c conda-forge

# Feedstock: https://github.com/deepmd-kit-recipes/abacus-feedstock
# Package:   https://anaconda.org/deepmodeling/abacus

{% set version = os.popen('git describe --tags --abbrev=0').read().strip('\n').l

package:
  name: abacus
  version: {{ version }}

source:
  path: ..
  # git_url: https://github.com/deepmodeling/abacus-develop.git
  # git_rev: v3.0.0

build:
  skip: true  # [not linux]
  script: |
    export CMAKE_PREFIX_PATH=`python -c "import torch;print(torch.__path__[0])"`
    cmake -B conda_build ${CMAKE_ARGS} -DCMAKE_BUILD_TYPE=Release -DENABLE_NATIV
    cmake --build conda_build -j`nproc`
    cmake --install conda_build
  # ${CMAKE_ARGS} applys restrictions for CMake to search libs under conda build
  # See https://conda-forge.org/docs/maintainer/knowledge_base.html#using-cmake
  string: {{ GIT_BUILD_STR }}
  number: {{ GIT_DESCRIBE_NUMBER }}

requirements:
  build:
    - {{ compiler('cxx') }}
    - {{ compiler('cuda')}}
    - make
```

```
    - cmake

  host:
    - {{ mpi }}
    - openblas=*=openmp*
    - elpa=*=mpi*
    - fftw=*=mpi*
    - cereal
      # pytorch
      # libxc
    - cudatoolkit=11.8

  run:
    - libopenblas=*=openmp*
    - scalapack
    - elpa=*=mpi*
    - fftw=*=mpi*
    - cudatoolkit=11.8

test:
  commands:
  # Dry run ABACUS to verify dynamic libs are present.
    - abacus
    - mpirun -n 2 abacus

  # Run end-to-end tests. This may take long time; disabled by default.
  # Unit tests are not built here.
  # Hence, some tests(ienvelope) requiring additional validation components are
  # Please uncomment the codes below if necessary.

  #   - cd tests/integrate && bash Autotest.sh
  # source_files:
  #   - tests/integrate
  #   - tests/PP_ORB

about:
  home: http://abacus.ustc.edu.cn/
  doc_url: https://abacus.deepmodeling.com/
  dev_url: https://github.com/deepmodeling/abacus-develop
  license: LGPL-3.0
  license_family: LGPL
  license_file: LICENSE
  summary: An electronic structure package based on plane wave and numerical ato
  description: >
    ABACUS (Atomic-orbital Based Ab-initio Computation at UStc) is an open-sourc
    The package utilizes both plane wave and numerical atomic basis sets with th
    to describe the interactions between nuclear ions and valence electrons. ABA
```

```
    Apart from single-point calculations, the package allows geometry optimizati

extra:
  recipe-maintainers:
    - caic99
```

```
channel_sources:
  - conda-forge
  - defaults
channel_targets:
  - deepmodeling
mpi:
  - mpich
  # - openmpi

cuda_compiler:
  - nvcc

c_compiler_version:
  - 10.4.0 # [linux]

cxx_compiler_version:
  - 10.4.0 # [linux]
```

2. 从头构建版本

```
# Install ABACUS by conda:
#   conda create -n abacus_env abacus -c deepmodeling -c conda-forge
#   conda activate abacus_env
# Docs: https://abacus.deepmodeling.com/en/latest/quick_start/easy_install.html#

# This conda package can also be built locally with
#   conda build . -c conda-forge
# And install with
#   conda create -n my_abacus_env abacus -c local -c conda-forge

# Feedstock: https://github.com/deepmd-kit-recipes/abacus-feedstock
# Package:   https://anaconda.org/deepmodeling/abacus

{% set version = os.popen('git describe --tags --abbrev=0').read().strip('\n').l

package:
  name: abacus
```

```
  version: {{ version }}

source:
  path: ..
  # git_url: https://github.com/deepmodeling/abacus-develop.git
  # git_rev: v3.0.0

build:
  skip: true  # [not linux]
  script: |
    export CMAKE_PREFIX_PATH=`python -c "import torch;print(torch.__path__[0])"`
    cmake -B conda_build ${CMAKE_ARGS} -DCMAKE_BUILD_TYPE=Release -DUSE_CUDA=1 -
    # Native Optimization: Do not enable it if generated code will run on differ
    cmake --build conda_build -j`nproc`
    cmake --install conda_build
  # ${CMAKE_ARGS} applys restrictions for CMake to search libs under conda build
  # See https://conda-forge.org/docs/maintainer/knowledge_base.html#using-cmake
  string: {{ GIT_BUILD_STR }}
  number: {{ GIT_DESCRIBE_NUMBER }}

requirements:
  build:
    - {{ compiler('cuda') }}
    - {{ compiler('cxx') }}
    - make
    - cmake
    - cudatoolkit-dev=11.7
    - cudatoolkit=11.7

  host:
    - {{ mpi }}
    - openblas=*=openmp*
    - elpa=*=mpi*
    - fftw=*=mpi*
    - cereal
    - cudatoolkit-dev=11.7
    - cudatoolkit=11.7
      #- pytorch
      # - libxc

  run:
    - libopenblas=*=openmp*
    - scalapack
    - elpa=*=mpi*
    - fftw=*=mpi*
    - cudatoolkit-dev=11.7
    - cudatoolkit=11.7
```

```yaml

test:
  commands:
  # Dry run ABACUS to verify dynamic libs are present.
    - abacus
    - mpirun -n 2 abacus

  # Run end-to-end tests. This may take long time; disabled by default.
  # Unit tests are not built here.
  # Hence, some tests(ienvelope) requiring additional validation components are
  # Please uncomment the codes below if necessary.

  #   - cd tests/integrate && bash Autotest.sh
  # source_files:
  #   - tests/integrate
  #   - tests/PP_ORB

about:
  home: http://abacus.ustc.edu.cn/
  doc_url: https://abacus.deepmodeling.com/
  dev_url: https://github.com/deepmodeling/abacus-develop
  license: LGPL-3.0
  license_family: LGPL
  license_file: LICENSE
  summary: An electronic structure package based on plane wave and numerical ato
  description: >
    ABACUS (Atomic-orbital Based Ab-initio Computation at UStc) is an open-sourc
    The package utilizes both plane wave and numerical atomic basis sets with th
    to describe the interactions between nuclear ions and valence electrons. ABA
    Apart from single-point calculations, the package allows geometry optimizati

extra:
  recipe-maintainers:
    - caic99
```

```yaml
channel_sources:
  - conda-forge
  - defaults
channel_targets:
  - deepmodeling
mpi:
  - mpich
  # - openmpi
  #
cuda_compiler:
```

```
  - nvcc
cuda_compiler_version:
  - 11.7
c_compiler_version:
  - 10.4
cxx_compiler_version:
  - 10.4
```