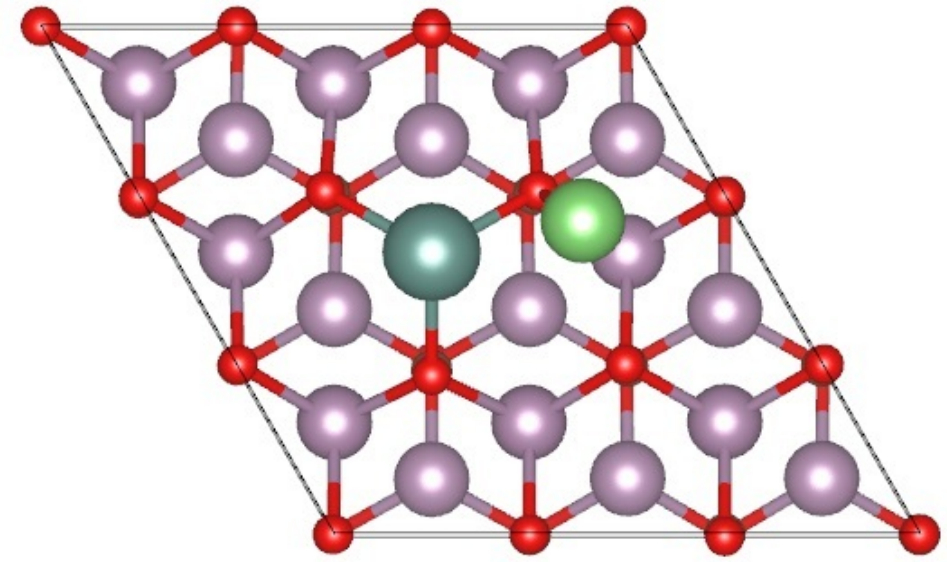


gpaw 安装合集（在线、离线、非root）（linux）

玻璃球球
材料科学与工程博士在读

gpaw 是一款效率比较高的第一性原理计算软件，在接近vasp的计算精度情况下，只消耗20%的时间。官网为：

GPAW
wiki.fysik.dtu.dk/gpaw/index.html

安装介绍：

Installation - GPAW
wiki.fysik.dtu.dk/gpaw/install.html#siteconfig

首先说结论:

如果犹豫不决，先把intel oneAPI装了，其自带了BLAS ,MPI, ScaLAPACK 等众多套件，省时省力，（参考第三方案）。

1. 下文前两方案为自己安装测试使用，不建议作为生产力。
2. 下文前两方案为自己安装测试使用，不建议作为生产力。
3. 下文前两方案为自己安装测试使用，不建议作为生产力。

一. root，在线，pip安装, 单节点（centos：最简洁版）

CentOS - GPAW
wiki.fysik.dtu.dk/gpaw/platforms/Linux/centos.html

这个可以安装在自己的电脑，用来作为测试使用。

1. 下载必要依赖包：

```
yum install epel-release #有些程序包在这里面
yum install libxc-devel openblas-devel openmpi-devel fftw-devel yum install blacs-open
```

2. 安装gpaw:

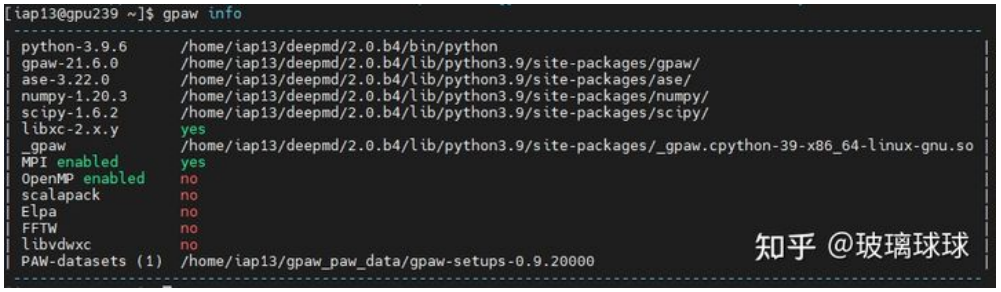
```
pip install gpaw
```

3. gpaw势文件:

```
gpaw install-data <dir>
```

其中势文件setups文件夹，将会自动存储在上文<dir>的自定义路径中。

（将自动从官网 wiki.fysik.dtu.dk/gpaw/... 下载，并自动解压。）



结果如上图所示，这种安装方式功能有限，单节点并行是可以的，不能跨节点运行。

二. root 离线，本地安装（同样测试使用）

1. **下载必要依赖包，（在线安装如下，离线rpm包安装不做展开）：**

```
yum install epel-release #有些程序包在这里面
yum install libxc-devel openblas-devel openmpi-devel fftw-devel
yum install blacs-openmpi-devel scalapack-openmpi-devel
```

从官网下载gpaw.tar.gz，解压并修改配置文件 siteconfig.py 。（若没有，可创建或者更改已存在的siteconfig_exapmle.py 为 siteconfig.py ）：

```
tar -xf gpaw***.tar.gz
cd gpaw
vim siteconfig.py
```

写入：

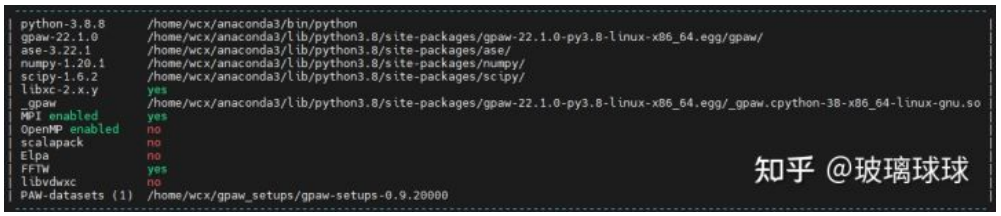
```
# blas
blas = True
if blas:
    libraries += ['openblas']


# FFTW3:
fftw = True
if fftw:
    libraries += ['fftw3']
```

```
# ScaLAPACK (version 2.0.1+ required):
scalapack = False
if scalapack:
    libraries += ['scalapack']

mpiblacs = False
if mpiblacs :
    libraries +=['mpiblacs']

# xc:
xc = True
if xc:
    libraries += ['xc']

# openmp:
openmp = False
if openmp:
    extra_compile_args += ['-fopenmp']
    extra_link_args += ['-fopenmp']
```

保存文件，其中 openmp 在并行安装时使用。（*若安装失败，删除build文件夹，并对症修改重试*）

2. 安装gpaw:

```
python setup.py install
gpaw info
```



结果如下图所示，其他辅助功能可按照类似方式配置。

ubuntu

https://wiki.fysik.dtu.dk/gpaw/platforms/Linux/ubuntu.html
wiki.fysik.dtu.dk/gpaw/platforms/Linux/ubuntu.html

Ubuntu 18.04+ - GPAW

Ubuntu 18.04+ - GPAW
wiki.fysik.dtu.dk/gpaw/platforms/Linux/ubuntu.html

注：ubuntu同centos，第一步依赖程序换成：

```
sudo apt install libopenblas-dev libxc-dev libscalapack-mpi-dev libfftw3-dev
```

前两种方案实在是太小儿科了。不装几天软件，不掉头发怎得对得起开源软件这几个字。准备好头发，开始吧！

三. 在线, root，多节点并行（intel oneAPI）（生产力）

1.oneAPI

intel oneAPI 为intel开发工具合计。包含众多功能组件，如**BT，HPC，AI**等。

oneMKL 为数学运算库，属于**Base Toolkit。**

oneMPI为并行计算库，属于**oneHPC** 高性能计算套件的一部分。

建议只安装**oneMKL+oneHPC，或者oneMKL+oneMPI**，不然会严重拖慢启动速度，影响文件传输效率等。

下载地址：

Get Intel® oneAPI Math Kernel Library
www.intel.com/content/www/us/en/devel...

Intel® oneAPI standalone component installation files
www.intel.com/content/www/us/en/devel...

后续添加环境有多种方案，选择下面一个就好。

可选1. 需要添加到环境变量 .bashrc

这将所有的intel工具全部添加一遍，除了**启动速度很慢**，注意替换实际自己的路径，其他没毛病。

```
source ~/intel/oneapi/setvars.sh intel64 --force
```

如果intelpython安装了，又自己安装的anaconda3，运行完上述代码后，intelpython将会优先于anadonda3

想用anaconda3, 添加 .bashrc

```
source ~/intel/oneapi/setvars.sh intel64 --force
PATH=/home/xxx/anaconda3/bin:$PATH
```

另一种方案是直接更改setvars.sh，过滤掉intelpython，可能需要一定c语言的知识，相信大家没兴趣。

直接用 intelpython？ 你会打开新世界大门。

可选2. 这种方式重点**推荐**，需要添加到环境变量 .bashrc

因为我们编译只需要compiler，mpi，mkl 因此，只添加

```
source ~/intel/oneapi/mkl/2022.0.2/env/vars.sh intel64 --force
source ~/intel/oneapi/compiler/2022.0.2/env/vars.sh intel64 --force
source ~/intel/oneapi/mpi/2021.5.1/env/vars.sh intel64 --force
```

可选3. 可以尝试添加下述信息 .bashrc （这是老办法, **不建议用了**，适合以前的intel编译器版本，这里的路径需要自定义， 其他版本可能需要根据自身目录有所增减调整）。

```
export PATH=/opt/intel/oneapi/mpi/2021.4.0/bin:$PATH
MPI=/opt/intel/oneapi/mpi/2021.4.0
export C_INCLUDE_PATH=$MPI/include:$C_INCLUDE_PATH
export LIBRARY_PATH=$M
```

```
export LD_LIBRARY_PATH=$MPI/lib:$LD_LIBRARY_PATH

MPIRE=/opt/intel/oneapi/mpi/2021.4.0/lib/release
export LIBRARY_PATH=$MPIRE:$LIBRARY_PATH
export LD_LIBRARY_PATH=$MPIRE:$LD_LIBRARY_PATH

export PATH=/opt/intel/oneapi/mpi/2021.4.0/libfabric/bin:$PATH
MPI=/opt/intel/oneapi/mpi/2021.4.0/libfabric
export LIBRARY_PATH=$MPI/lib:$LIBRARY_PATH
export LD_LIBRARY_PATH=$MPI/lib:$LD_LIBRARY_PATH

export PATH=/opt/intel/oneapi/mkl/2021.4.0/bin/intel64:$PATH
MKL=/opt/intel/oneapi/mkl/2021.4.0
export C_INCLUDE_PATH=$MKL/include:$C_INCLUDE_PATH
export LIBRARY_PATH=$MKL/lib/intel64:$LIBRARY_PATH
export LD_LIBRARY_PATH=$MKL/lib/intel64:$LD_LIBRARY_PATH

.....
```

..... 代表缺啥补啥（微笑脸）。

2.libxc

下载：（这里建议不要搞太新的版本，5.2.0就挺好的，4.X可能有点兼容问题）

https://www.tddft.org/programs/libxc/download/
www.tddft.org/programs/libxc/download/

```
tar -xf libxc-5.2.0.tar.gz
cd libxc-5.2.0
./configure --enable-shared --disable-fortran --prefix=$HOME/libxc-5.2.0
make
make install
```

.bashrc 中写入（所有环境变量更改，需要source ~/.bashrc更新，后续不再强调）：

```
XC=~/libxc-5.2.0
export C_INCLUDE_PATH=$XC/include
export LIBRARY_PATH=$XC/lib
export LD_LIBRARY_PATH=$XC/lib
```

3.gpaw

安装好上述部分，不建议按照网站说明构建：即**不要**参考下面链接。

Old MKL Method
wiki.fysik.dtu.dk/gpaw/platforms/Linux/juropa.html

1. 配置环境。

解压 gpaw-xxxx.tar.gz, 修改 siteconfig.py 文件（没有就创建一个）中路径，以及检查 libraries 是否名称对应。

```
scalapack = True

library_dirs += ['/opt
```

```
libraries = ['mkl_intel_lp64' ,'mkl_sequential' ,'mkl_core',
             'mkl_lapack95_lp64',
             'mkl_scalapack_lp64', 'mkl_blacs_intelmpi_lp64',
             'pthread']


libraries += ['xc']
LIBXCDIR='/home/libxc-5.2.0/'
library_dirs += [LIBXCDIR + 'lib']
include_dirs += [LIBXCDIR + 'include']
```

其他辅助功能，如 fftw 可按照类似方式配置。

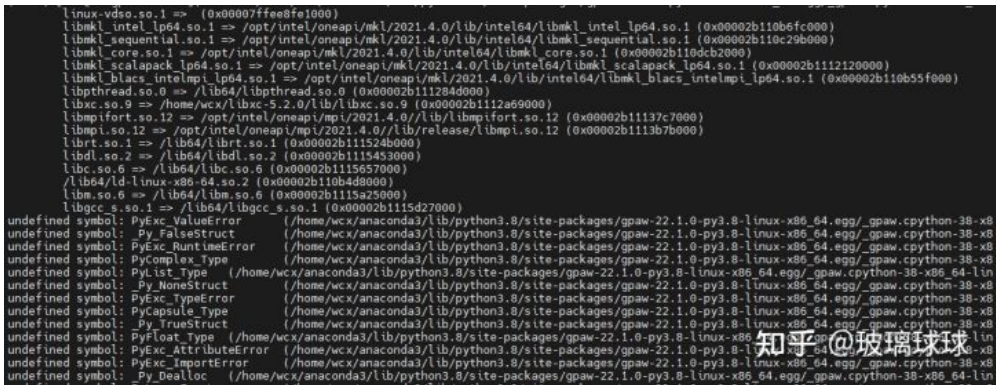
2. 安装测试。(自信点，此步可以跳过)

因为安装过程可能遇到很多问题，可以尝试先build, 发现没有问题后再install 。

```
python setup.py build
```

如果没有报错，继续输入下文，否则参考文章最后的**报错合集。**

```
ldd -r build/lib.linux-x86_64-3.8/_gpaw.cpython-38-x86_64-linux-gnu.so
```



输出如图所示，

- 检查所有'mkl_intel_lp64' ,'mkl_sequential' ,'mkl_core', 'mkl_scalapack_lp64', 'mkl_blacs_intelmpi_lp64', 'mpi', 'mpifort' 的链接是否链接到oneapi。
- 检查'xc' 是否链接到上文装的地址（libxc.so.9 => /home/wcx/libxc-5.2.0/lib/libxc.so.9 ）（很重要，参考报错合集）。
- 检查所有的undefined symbol， 确定都是以 Pyxxxxx, _Pyxxxx开头的。
- 上述没有问题，继续，否则排查问题。

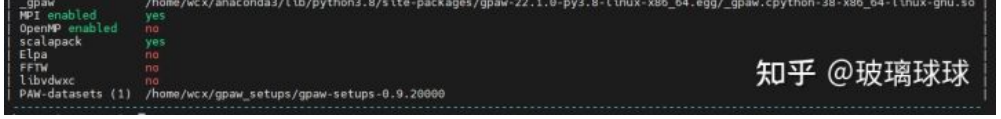
3.安装

```
python setup.py install
gpaw info
```

下载数据gpaw势文件，上传，解压过程不做介绍，完成！

Atomic PAW Setups
wiki.fysik.dtu.dk/gpaw/setups/setups.html#installation...

若安装遇见问题，回到步骤"**2.安装测试"**，或者参考报错合集。



四. 完全离线, 非root，编译安装，多节点并行 （ intel oneAPI ）

对于不能联网服务器，比如某些使用intel cpu的超算，安装最为麻烦，其他种类的cpu请参考他们的文档。

离线安装步骤与**方案四**一致。因为离线的原因，安装位置需要指定自己用户的一个文件夹，所有安装使用本地即可，如intel oneAPI。

五. 下面为部分包的安装，仅做参考，不确定对每个人有效

若使用oneAPI, fftw是**冗余**的，可以不用装。

fftw：(先安装fftw, 再gpaw。）

FFTW Download Page
www.fftw.org/download.html

编译安装，其中配置参数注意：

(并行版本)，需要

```
./configure MPICC=mpicc --prefix=$HOME/fftw --disable-fortran --enable-mpi --enable-sh
```

(非并行版本)

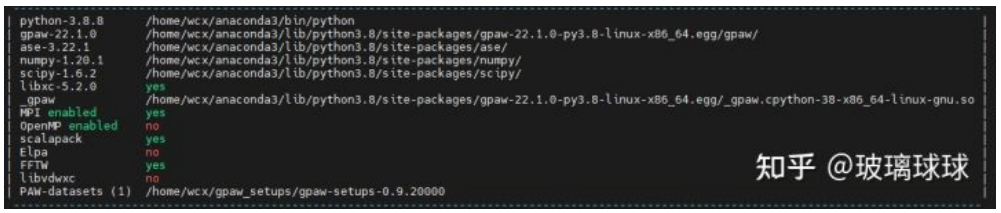
```
./configure  --prefix=$HOME/fftw --disable-fortran
```

（需要使用这个fftw安装vasp的，--disable-fortran 可以去掉, 若使用oneAPI装vasp, fftw是也是冗余的，可以不用装。 ）

```
make
make install
```

添加文件*.bashrc:*

```
export PATH=~/fftw/bin:$PATH
export LD_LIBRARY_PATH=/$HOME/fftw/lib/:$LD_LIBRARY_PATH
export C_INCLUDE_PATH=$HOME/fftw/include/:$C_INCLUDE_PATH
export LIBRARY_PATH=$HOME/fftw/lib/:$LIBRARY_PATH
```



报错合集：

错误1

```
'undefined symbol: xc_func_set_ext_params, undefined symbol: xc_version_string'
```

原因：安装旧版本xc, 如 yum install libxc-devel, 又编译了libxc5.2.0 导致连接错误。

处理：删除旧版本，如位置 /usr/lib64/libxc.***

错误2

```
'undefined symbol: xc_lda_exc'
```

原因：libxc路径设置错误,或者没source。

处理：

siteconfig.py , 注意设置自己的路径。

```
libraries += ['xc']
LIBXCDIR='/home/libxc-5.2.0/'
library_dirs += [LIBXCDIR + 'lib']
include_dirs += [LIBXCDIR + 'include']
```

.bashrc

```
export PATH=~/libxc-5.2.0/bin:$PATH
XC=~/libxc-5.2.0
export C_INCLUDE_PATH=$XC/include:$C_INCLUDE_PATH
export LIBRARY_PATH=$XC/lib:$LIBRARY_PATH
export LD_LIBRARY_PATH=$XC/lib:$LD_LIBRARY_PATH
```

错误3

```
mpicc not found
```

原因：Intel oneapi 没有添加环境变量。

处理：环境变量添加

```
source /opt/intel/oneapi/setvars.sh intel64 --force
```

错误4

```
Fatal error in PMPI_Init: Other MPI error, error stack:
MPIR_Init_thread(143).........:
MPID_Init(1310)..............
....
```

原因：系统 GNU gcc 的mpicc 优先于Intel oneapi 的 mpicc

处理：环境变量添加

```
source /opt/intel/oneapi/setvars.sh intel64 --force
ulimit -s unlimited
```

错误5

```
..../app/anaconda3/bin/mpicc: line 301: x86_64-conda_cos6-linux-gnu-cc: command not fc
```

原因：系统 GNU gcc 的mpicc 优先于Intel oneapi 的 mpicc

处理：环境变量添加

```
source /opt/intel/oneapi/setvars.sh intel64 --force
```

看到这儿了，猛男要赞。

编辑于 2022-04-15 13:24

傅里叶变换（Fourier Transform） 第一性原理计算

3 条评论 切换为时间排序

写下你的评论...

Restart 19 小时前

Onempi找了好久是真的没找到，之前服务器有Inter mpi/mkl,但是好像编译不了这个。🤣

赞

玻璃球球 (作者) 回复 Restart 17 小时前

就在intel官网上，多翻翻就看到了🫣

赞

DFT中

看文献也看到这个方