

CP2K-4.1的编译



作者 神的第57个名字 (/u/847aa3e5e059) [+ 关注](#)

2017.05.23 11:47 字数 2148 阅读 41 评论 0 喜欢 0

(/u/847aa3e5e059)

第一性原理计算的相关软件的编译对于新手来说真的很麻烦，但这一问题不应该成为这个方向刚入门的学习者或是研究者的门槛。

作为一个交叉学科的学生，去年第一次接触到这种第一性原理计算的软件，当时有一个大师兄带我入门，VASP 和 LAMMPS 的编译也是师兄做的，没我啥事儿，也没觉得这事儿有多麻烦。

这次编译CP2K是因为它的4.1版本实现了一个跟课题比较相关的功能，需要跑一些例子，但是师兄自己也没编译过这款软件，最初我们试图按照官网的预编译版本来试着安装，没有成功。后来找到一篇关于用Gfortran编译cp2k4.1 (<http://www.cndba.cn/shuye100/article/1723>)的博客，在自己本地机上的安装基本遵从了这篇博客的指导，成功了。先说一下PC的配置吧，i5的64位处理器，4G内存，系统是Ubuntu14.。

但是试着跑了CP2K官网 (<https://www.cp2k.org/>)的几个例子，在水的几何结构优化那儿就跑不动了，报的是内存错误，应该是PC带不动了，那接下来必须得试着在服务器上安装了。我们这边服务器的类型是

这边顺带插一句，就是关于在Linux系统下安装软件的问题。除了不同发行版自带的包管理工具外，我们可以尝试着从源码安装。因为在服务器上我们是没有root权限的，这种情况下就必须放弃包管理工具。但是这对没在Linux下安装过软件的初学者来说也是一项很艰巨的任务。这里有一篇参考博客关于Linux下软件的安装 (<http://blog.csdn.net/u010509774/article/details/50593231>)，初学者可以一看。

在服务器上，首先还是试着用那篇博客给出的方式来安装。CP2K之所以难装，一方面就在于其依赖很多跟数学和物理相关的包，譬如 BLAS，LAPACK，FFTW 等，可能有一个版本不合适，就会报出一堆的错误。而且，CP2K的官网说的很明白，对 gcc 的版本有要求，可怜我们服务器上默认的版本太低，还得自己去装，希望暑假过后新换的机器上软件版本能跟得上。



故事到这里并不是一个童话里的欢喜结局，这也是我写这篇的目的所在。在我们的服务

器上按照步骤来安装老是报错，具体什么错误也没有留照保存，我们看的不是很懂，但估计是 gcc 编译器的问题。

接下来偶然间在一个类似于计算化学公社的地方看到一篇帖子，大意是说目前为了方便安装CP2K，其4.1版本已自带一键安装的脚本，脚本的位置比较隐蔽，反正我是找了挺久才在官网找到其位置，在目录 `cp2k-4.1/tools/toolchain` 下有一个脚本

`install_cp2k_toolchain.sh`，如图：

```
$ ls
PERS  README_FOR_USERS  build  install  install_cp2k_toolchain.sh  scripts
$
```

自带脚本

下面介绍下这个脚本的功能，大致就是调用上图中的 `scripts` 目录下的各种包的安装脚本，

```
arch_base.tpl      install_elpa.sh    install_libxsmm.sh  install_parmetis.sh  install_valgrind.sh
checksums.sha256   install_fftw.sh    install_make.sh      install_pexsi.sh      package_versions.sh
common_vars.sh     install_gcc.sh     install_mmetis.sh    install_ptscotch.sh   parse_if.py
get_openblas_arch.sh  install_lcov.sh    install_mkl.sh        install_quip.sh        signal_trap.sh
install_acml.sh     install_libint.sh   install_mpich.sh      install_reflapack.sh  tool_kit.sh
install_binutils.sh  install_libsmm.sh   install_openblas.sh   install_scalapack.sh
install_cmake.sh    install_libxc.sh    install_openmpi.sh    install_superlu.sh
```

`scripts`目录下的安装脚本

如果包安装都已完成，脚本再引导你做接下来几步就能完成全部安装过程，最终会在 `cp2k-4.1/exe` 下生成一个 `local` 目录，内容是串行和并行版的 `cp2k.*` 可执行程序。

我在PC上测试了下这个脚本是否是可用的，大概一两个小时后，发现确实编译成功了并生成了各个版本的可执行文件。但是在服务器上这个方法又出现了些小问题，譬如很多次出现了域名没有办法解析。

我们开始根据错误指示的位置一步步去检查 `scripts` 下的安装脚本，这些安装脚本的工作机制是一样的，首先检查 `cp2k-4.1/tools/toolchain/build` 下是否有所需要的安装包，下图里展示的是 `install_fftw.sh` 的内容。



```
case "$with_fftw" in
__INSTALL__)
    echo "===== Installing FFTW ====="
    pkg_install_dir="${INSTALLDIR}/fftw-${fftw_ver}"
    install_lock_file="$pkg_install_dir/install_successful"
    if [ -f "${install_lock_file}" ] ; then
        echo "fftw-${fftw_ver} is already installed, skipping it."
    else
        if [ -f fftw-${fftw_ver}.tar.gz ] ; then
            echo "fftw-${fftw_ver}.tar.gz is found"
        else
            download_pkg ${DOWNLOADER_FLAGS} \
                https://www.cp2k.org/static/downloads/fftw-${fftw_ver}.tar.gz
        fi
        echo "Installing from scratch into ${pkg_install_dir}"
        [ -d fftw-${fftw_ver} ] && rm -rf fftw-${fftw_ver}
        tar -xzf fftw-${fftw_ver}.tar.gz
        cd fftw-${fftw_ver}
```

fftw的安装脚本

如果在服务器上已经安装了fftw，我们还可以通过在运行脚本时指定命令行参数来指定安装的fftw的lib目录的位置，像这样 `./install_cp2k_toolchain.sh --with-fftw=[fftw_lib_dir]` (这里假定在 `toolchain` 目录下)。如果脚本能在该目录下找到需要的库文件，就会直接输出该包已经安装好的信息，并且开始下一个包的安装。如果在 `build` 目录下没有需要的安装包，就会联网下载，很多问题就是出在这一步。

于是我们接下来就将之前在PC上通过这个脚本安装的cp2k-4.1下面的 `build` 目录里面的所有压缩包(这个脚本写的就是这样，这些安装包没有删除)发送到服务器的cp2k-4.1下面的 `build` 目录，那么脚本就不会启动联网下载，而是直接开始安装每个包了。这里需要提一下的是 `MKL` 库，这个库可以选择安装，因为它实现的功能基本就由 `BLAS` 和 `LAPACK` 共同实现了，可以通过命令行选项 `--with-mkl=no` 来取消对 `MKL` 的检查。

安装各种包的过程也不是一帆风顺的，一些数学和物理的包没有问题，像是 `BLAS`，`LAPACK`，`LIBINT`，`LIBXC` 等等，最难装的还是 `GCC`，我们的服务器上的GCC版本不够，个人分配的空间也不大，就没有试着在那边装了。后来是换到了天河二号上安装，在那边没有重新安装GCC，基本上是一次成功了。

这里还有个插曲，在安装成功各个包去到 `makefile` 下 `make` 的时候，死活编译不过去，并提示是 cp2k 源码的错误。当时的心情真是跟坐过山车一样刺激，师兄跟我绷不住了，最终师兄决定去请教一个专门搞软件开发的合作者。到了第二天，师兄喊我去继续安装，说是那个合作者一眼瞅了输出信息就大致知道了是 `BLAS` 还是 `LAPACK` 安装时出的问题，可能是链接的库给错了，也可能是安装的时候 `make` 出来的静态库不对。反正最后是重新编译了一下 `BLAS` 还是 `LAPACK` 结果一路绿灯安装成功。这里是可执行文件：



```
cp2k.pdbg
cp2k.popt
cp2k.psmf
cp2k.sdbg
cp2k_shell.pdbg
cp2k_shell.popt
cp2k_shell.psmf
cp2k_shell.sdbg
cp2k_shell.sopt
cp2k_shell.ssmf
cp2k.sopt
cp2k.ssmf
```

cp2k的可执行文件

这几天在天河二号上跑了一下水的几何结构优化和尿素溶于水的QMMM算法的例子，都跑出来了，没有问题。对于编译过很多类似软件的老司机来说，这些问题可能都不是问题，但是对于刚入门的初学者来说，走这些弯路很耗费精力，考验耐心。

BASIS_SET	H2O-1.restart.bak-1	H2O-1.restart.bak-3	h2o.out	POTENTIAL
H2O-1.restart	H2O-1.restart.bak-2	H2O.inp	H2O-pos-1.xyz	


水的几何结构优化的例子

我写的这一篇并不是手把手地指导入门者去怎么样怎么样安装，但这是所有初学者最渴望的事情，有一篇万能的一步一步的指导手册，躺着键入一些代码就行。但是考虑到各个机器的型号和性能，以及系统和所安装软件的版本也不尽相同，可能还不能有一些万能的安装模板。

安装这个软件大概消耗了我小两周的时间，如果让我总结一些一般性的指导原则，大概就是：

- 安装那些依赖的数学库和物理库的时候要格外小心，要生成正确的静态库，并在后续安装时要给出正确的静态库的位置。
- 如果提示有源码错误，那估计是安装的依赖库有问题。
- GCC 和其他一些库的版本很重要，一定要根据官网的指示来。



 随笔 (/nb/10123109)

举报文章 © 著作权归作者所有



神的第57个名字 (/u/847aa3e5e059)

写了 5921 字，被 0 人关注，获得了 0 个喜欢
(/u/847aa3e5e059)

+ 关注

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

赞赏支持

喜欢 (/sign_in) | 0







更多分享

(http://cwb.assets.jianshu.io
/notes
/images
/12677023
/weibo
/image_d14b4a42dc88.jpg)



登录 (/sign_in)后发表评论

评论

智慧如你，不想发表一点想法 (/sign_in)咩~

