

## Intel Parallel Studio XE 编译VASP

2018-01-15
cndaqiang
DFT
vasp
centos
Intel

之前尝试编译VASPUbuntu VASP安装和运行,但是在centos上进行重复时,各种报错,现在尝试了安装几次感觉自己的理解更多了,总结如下.这篇文章省略了很多命令,有看不懂的地方参考Ubuntu VASP安装和运行.

### 编译注意

- 硬盘空间足够  
编译时会产生很多临时文件,占据空间大, Intel编译器Intel Parallel Studio XE 2018 for linux安装后占11G.安装包3.5G,硬盘空间不足编译失败  
如果硬盘空间太小,可以尝试安装老版的intel编译器
- 内存足够  
使用fortran编译vasp时,内存1G编译过程中,进程被杀死.添加2G的虚拟内存,编译通过
- 编译器最好一致  
vasp需要数学库,mkl,fft,使用gfortran和fort编译产生的库文件不同,最后使用gfortran和fort编译vasp时容易冲突,所以只使用fort或gfortran其中的一种进行编译
- configure的一些参数
  - prefix 安装目录,是最后 make install 的安装地址
- 安装地址  
可以编译在 /home/username 即家目录下,这样只能自己使用  
也可以安装到根目录下的某目录(需要root权限).每个用户都可以使用
- PATH  
安装软件后,软件执行文件所在目录被添加到系统PATH路径后,才能在shell里直接输入命令如 `icc`,不添加则需要使用 `/opt/intel/bin/icc` 运行  
添加PATH的方法, 参考添加PATH
- 编译选项  
`configure -h` 可以查看生成makefile的编译选项,如CC(C编译器)FC(Fortran编译器)MPICC(并行C)MPIFC(并行FC)enable-mpi(执行并行)  
编译并行ftw时,指定intel编译器

```
./configure --prefix=/opt/fftw/ CC=icc F77=ifort MPICC=mpicc --enable-mpi
```

编译siesta

```
../Src/configure FC=ifort CC=icc MPIFC=mpifort --enable-mpi
```

### vasp编译说明

建议认真读一下vasp.4.6b的makefile文件,里面说的很详细makefile.linux\_ifc\_P4, 还有VASP.5.4.1里面的README

### vasp安装需要

- fortran等编译器  
intel: icc ifort  
gfortran等
- 数学库 BLAS BLACS LAPACK SCALAPACK  
intel:mkl含有  
分别从NETLIB编译安装  
SCALAPACK安装包可帮忙下载所有数学库
- fft  
vasp自带  
intel:mkl含有  
编译ftw
- 注意:  
若编译数学库,ftw,需和最后编译vasp使用同一fortran编译器  
编译数学库,需支持mpi并行,这样才能编译支持mpi的vasp

通过上面的分析,我们可以发现, intel的编译器Intel Parallel Studio XE包含了我们编译vasp所有的工具

### 编译准备

从Intel Parallel Studio XE注册账号, 获取安装序列号, 建议使用edu邮箱注册, 获取序列号时间短,我申请的开源贡献者账号好几天都没通过.当然也有使用license激活的,百度相关资源.  
此次我使用的版本Intel® Parallel Studio XE 2018 for Linux

vasp来源VASP,组里购买的正版,网上也可搜索的相关的资源,计算请使用正版  
此次我使用的文件vasp.5.4.1.24Jun15.tar.gz

### 编译环境

此次编译在vmware上运行的Centos7.3.10.0-514.el7.x86\_64.2G内存,17-7500U  
除了intel需要的库安装方法不一样外, vasp编译运行方式应该适用于所有Linux

### 编译过程

#### intel

#### 依赖

```
yum install glibc-devel.i686
yum install libstdc++.so.6
ldconfig
yum install gcc-c++
```

安装过程提示OS unsupported, 忽略  
有其他依赖缺少时,yum安装后再Re-check

另外在云服务器上尝试时

- KVM安装时提示 CPU unsupported, 暂未继续安装
- KVM安装提示 内核有问题, 暂未继续安装

#### 安装

```
./install.sh
```

同意协议,保持默认选项即可,默认安装到 `/opt/intel`,也可以自定义  
使用ubuntu16.04安装时,使用 `./install_GUI.sh` ,可以选择安装组件. 只安装64位,icc,ifort,mpi,mkl就可以运行. 安装后占空间约2G

#### 添加PATH

下面的路径与实际路径与intel编译器的版本有关,版本变更后适当修改  
执行

```
source /opt/intel/compilers_and_libraries_2018.0.128/linux/bin/compilervars.sh intel64
source /opt/intel/compilers_and_libraries_2018.0.128/linux/bin/iccvars.sh intel64
source /opt/intel/compilers_and_libraries_2018.0.128/linux/bin/ifortvars.sh intel64
source /opt/intel/compilers_and_libraries/linux/mkl/bin/mklvars.sh intel64
source /opt/intel/mpi/2018.0.128/bin64/mpivars.sh
```

或者讲上述命令添加到 `/etc/profile` 或 `~/./bashrc` ,具体含义添加PATH  
可用 `which icc ifort icpc mpifort` 检查是否添加成功

之后编译,若提示 `xxx:command not found`, 则再source一遍上述命令  
在编译时运行vasp时,若上述文件不在PATH内,也无法运行,需要先执行一遍

修改 `/etc/profile` 或 `~/./bashrc` 中就无需上述操作,登陆时source一下或者添加到文件永久修改都可以,看个人喜好

#### 编译并行ftw

下面的路径与实际路径与intel编译器的版本有关,版本变更后适当修改  
`make -h` 可不是很确定使用intel编译器编译并行版本的ftw命令为 `make libmhc`  
建议编译并行ftw使用下文编译ftw的方法

```
cd /opt/intel/compilers_and_libraries_2018.0.128/linux/mkl/interfaces/fftw3xf
make libmhc
```

编译后在当前文件夹内生成 `libfftw3xf_intel.a`

#### vasp

好像不需要vasp.5.lib也编译通过  
解压vasp.5.4.1.24Jun15.tar.gz后

```
cd vasp.5.4.1
cp arch/makefile.include.linux_intel makefile.include
```

修改makefile.include中内容

- 10行开始编译器配置

```
FC      = mpifort
FCL     = mpifort -mkl
```

- 19行开始,数学库配置如下

```
MKLROOT=/opt/intel/compilers_and_libraries_2018.0.128/linux/mkl
MKL_PATH    = $(MKLROOT)/lib/intel64
BLAS        = -L$(MKL_PATH) -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread
LAPACK      = -L$(MKL_PATH) -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread
BLACS       = -L$(MKL_PATH) -lmkl_scalapack_intel_lp64
SCALAPACK   = $(MKL_PATH)/libmkl_scalapack_lp64.o $(BLACS)
```

发现makefile.include中有 `LIB=LLIBS = $(SCALAPACK) $(LAPACK) $(BLAS)`,也可以

```
MKLROOT=/opt/intel/compilers_and_libraries_2018.0.128/linux/mkl
MKL_PATH    = $(MKLROOT)/lib/intel64
BLAS        = -L$(MKL_PATH) -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -
lmkl_blacs_intelmpi_lp64 -lmkl_scalapack_lp64
LAPACK      =
BLACS       =
SCALAPACK   =
```

- 26行做配置

```
OBJECTS    = fftmpi.o fftmpi_map.o fftw3d.o fft3dlib.o \  
             $(MKLROOT)/interfaces/fftw3xf/libfftw3xf_intel.a
INCS       = -I$(MKLROOT)/include/fftw
```

若自己编译ftw, 配置为(其中`opt/ftw`是我编译后安装的目录)

```
OBJECTS    = fftmpi.o fftmpi_map.o fftw3d.o fft3dlib.o \  
             /opt/fftw/lib/libfftw3_mpi.a
INCS       = -I/opt/fftw/include
```

最后我的makefile.included

编译

```
make
```

就在 `./build` 中生成了gamma版本的vasp,非线性版本的vasp,标准版本的vasp

```
gom ncl std
```

每个文件夹中都有一个vasp的可执行文件,添加PATH即可

### 运行vasp

若没有给数学库添加PATH,运行前需要source一下,具体内容, 前面都有  
vasp运行方式1)添加PATH直接输入vasp运行, 或类似这样 `~/vasp/vasp.5.4.1/build/std/vasp` 运行

是否添加PATH, 看组里习惯吧  
把输入文件放在一个文件夹中, 在该文件夹内运行vasp

可从Materials Project下载POSCAR, INCAR,KPOINTS,POTCAR从vasp网站下载,直接运算可能需要修改下载的INCAR,将结果与Materials Project结果比较

### 常见问题

这里放一些, 安装过程中的问题和解决方案

#### intel64

在intel的路径中 `ia32` 代表32位, `intel64` 代表64位

#### 数学库MKL

#### 数学库的调用

数学库可以分为静态链接和动态链接,两种方式都可以  
如 `/opt/intel/compilers_and_libraries_2018.0.128/linux/mkl/lib/intel64`  
里面有很多库文件 `libmkl_blacs_intelmpi_lp64.a libmkl_blacs_intelmpi_ilp64.so`  
其中拓展名为 `.a` 是静态库, `.so` 的为动态库

- 静态链接  
连接方式

```
LIB=-L/opt/intel/compilers_and_libraries_2018.0.128/linux/mkl/lib/intel64/libmkl_blacs_intelmpi_
_ilp64.a
```

- 动态链接  
连接方式

```
LIB=-L/opt/intel/compilers_and_libraries_2018.0.128/linux/mkl/lib/intel64 -
lmkl_blacs_intelmpi_ilp64
```

#### mpif90

编译vasp时若FC设置为mpif90,报错

```
gfortran: command not found
```

这是因为, intel MPI命令中mpif90调用gfortran进行编译,gfortran没安装报错  
若装上gfortran又会因为:数学库等是使用intel的fort编译的,和gfortran又有冲突报错

最好的解决方案,是FC=mpifort  
此内容参考mpif90 from cluster toolkit pointing to gfortran和科大李会民老师-MPI编译环境的使用

#### 编译时报错哪个数学库文件

检查数学库文件名是否正确,数学库是否选对,如  
`BLACS = -lmkl_blacs_intelmpi_lp64` 因为我们使用的是intel的mpi所以blacs使用 `libintelmpi`, 若使用openmpi,则设置为 `libmkl_blacs_openmpi_lp64`

#### 使用ftw

若不使用intel的ftw,下载ftw  
解压进入相关文件夹后,生成使用intel编译支持并行的makefile文件

```
./configure --prefix=/opt/fftw/ CC=icc F77=ifort MPICC=mpicc --enable-mpi
make
make install
```

则 `makefile.include` 中设置为

```
OBJECTS    = fftmpi.o fftmpi_map.o fftw3d.o fft3dlib.o \  
             /opt/fftw/lib/libfftw3_mpi.a
INCS       = -I/opt/fftw/include
```

此处参考ftw 编译安装说明

#### ftw不支持mpi报错

编译ftw时,若使用 `make libintel64`, 则编译的ftw不支持mpi,编译时会对 `libfftw3xf_intel.a` 报错

#### 报错

#### segmentation fault occurred

```
fortrtl: severe (174): SIGSEGV, segmentation fault occurred
Image      PC          Routine      Line      Source
vasp       00000000013CD4E0  Unknown    Unknown   Unknown
libpthread-2.17.so 000007F6E8288FE0  Unknown    Unknown   Unknown
```

每次运行前在shell中执行

```
ulimit -s unlimited
ulimit -m unlimited
ulimit -c unlimited
ulimit -d unlimited
```

再运行vasp

也可以添加 `ulimit -s unlimited` 到 `/etc/profile` 或 `~/bashrc`, 每次登陆自动执行

在Linux下写程序的时候, 如果程序比较大, 经常会遇到“段错误”(segmentation fault) 这样的问题,ulimit为shell内建指令, 可用来控制shell执行程序的资源

```
-a      显示当前资源限制的设定.
-c      <core文件上限>  设定core文件的最大值, 单位为区块.
-d      <数据节区大小>  程序数据节区的最大值, 单位为KB.
-f      <文件大小>      shell所能建立的最大文件, 单位为区块.
-H      设定资源的硬性限制, 也就是管理员所设下的限制.
-m      <内存大小>      指定可使用内存的上限, 单位为KB.
-n      <文件数>          指定同一时间最多可开启的文件数.
-p      <缓冲区分大小>  指定管道缓冲区分的大小, 单位为512字节.
-s      <堆叠大小>       指定堆叠的上限, 单位为KB.
-S      设定资源的弹性限制.
-t      <CPU时间>        指定CPU使用时间的上限, 单位为秒.
-u      <程序数目>       用户最多可开启的程序数目.
-v      <虚拟内存大小>   指定可使用的虚拟内存上限, 单位为KB
```

参考vasp.5.3错误fortrtl: severe (174): SIGSEGV, segmentation fault occurred

#### RLIMIT\_MEMLOCK too small

并行运算时

```
mpirun -genv I_MPI_DEVICE rdssm -machinefile host.fiel -n 4
/home/cndaqiang/soft/vasp.5.4.1/build/std/vasp
[0] DAPL startup: RLIMIT_MEMLOCK too small
```

使用

```
ulimit -l unlimited
```

这条命令涉及root的权限, 所以,添加到 `/etc/profile`, 也只能以root用户计算

推荐解决方案

参考Best Known Methods for Setting Locked Memory Size

修改 `/etc/security/limits.conf`,填入

```
username hard memlock unlimited
username soft memlock unlimited
```

参考组里服务器的配置,允许所有用户,设置为

```
* soft memlock unlimited
* hard memlock unlimited
* soft memlock unlimited
* soft stack unlimited
* soft nproc unlimited
* hard memlock unlimited
* hard nproc unlimited
* hard nproc unlimited
```

reboot重启生效

#### 这个错误

```
Fatal error in MPI_Alltoallv: Other MPI error, error stack:
MPI_Alltoallv(665).....: MPI_Alltoallv(cbuf=0x7f760c875340, scnts=0x7f760e1e7a00,
rdisp1s=0x7f760e1e7a00, MPI_INTEGER, rbuf=0x7f760c8eb380, rcnts=0x7f760e1e79a0,
rdisp1s=0x7f760e1e79e0, MPI_INTEGER, comm=0x84000007) failed
MPIR_Alltoallv_inpl(416).....: fail failed
MPIR_Alltoallv(373).....: fail failed
MPIR_Alltoallv_intra(226).....: fail failed
MPIR_Mwaitall_inpl(223).....: fail failed
MPIR_IDL_CH31_Progress(623).....: fail failed
pkt_RTS_handler(317).....: fail failed
do_cts(662).....: fail failed
MPIR_nem_lmt_dcp_start_recv(302): fail failed
dcp_recv(165).....: Internal MPI error! Cannot read from remote process
Two workarounds have been identified for this issue:
1) Enable ptrace for non-root users with:
echo 0 | sudo tee /proc/sys/kernel/yama/ptrace_scope
2) Or, use:
I_MPI_SHM_LMT=shm
```

解决

```
sudo su
I_MPI_SHM_LMT=shm
echo 0 | tee /proc/sys/kernel/yama/ptrace_scope
```

[未解决]dapl fabric is not available and fallback fabric is not enabled

并行运算时

```
mpirun -genv I_MPI_DEVICE rdssm -machinefile host.fiel -n 4
/home/cndaqiang/soft/vasp.5.4.1/build/std/vasp
[0] MPI startup(Q): dapl fabric is not available and fallback fabric is not enabled
```

DEBUG后

```
cannot open dynamic library libdat2.so.2
```

处理方案是修改 `/etc/dot.conf` 填入类似

```
afa-v2-mix5_0-1u u2.0 nonthreadsafe default libdaplcomm.so.2 dapl.2.0 "mix5_0 1" ""
```

的东西,参考 Using Connect-IB with Intel MPI  
dapl fabric is not available and fallback fabric is not enabled with IMPI 4.0.0

不想看了,以后再解决,先去掉 `-genv I_MPI_DEVICE rdssm` 参数运行

#### Similar Posts

- Intel Parallel Studio XE 编译siesta
- vasp计算流程-非自治计算
- vasp计算流程-静态自治
- vasp计算流程-结构优化
- win10 Bash 图形化运行程序
- VASP输出文件总结

上一篇 xxxxx: command not found

下一篇 flowplayer快捷键

#### Comments

2条评论
cndaqiang.github.io

❤️ 推荐
🔗 分享
评分最高
登录

加入讨论...
通过以下方式登录
或注册一个 DISQUS 帐号
姓名

chempeng · 5个月前
学习了, 感谢博主! 期待交流讨论 https://chempeng.github.io/
4 回复 · 分享 ·

cndaqiang · 5个月前
我开始学vasp,之前看过你的博客, 很有帮助
1 回复 · 分享 ·

🔒 私密
🔒 私密
在您的网站上使用 Disqus 添加 Disqus 隐私政策 隐私政策 隐私政策
DISQUS