INSTALL VASP-6.2.0#############

Steps: -

unzip vasp.6.2.0.zip

during compilation: -

#########################

for cshell

source /opt/ohpc/admin/lmod/8.1.18/init/csh

module load spack

source /home-ext/apps/spack/share/spack/setup-env.csh

spack load intel-oneapi-compilers@2022.0.1

spack load intel-oneapi-tbb

spack load intel-oneapi-mkl

spack load intel-oneapi-mpi@2021.5.0

spack load fftw@3.3.10%intel@2021.5.0 /tphl5ba

#########################

cp arch/makefile.include.linux_intel makefile.include

cat makefile.include

```
# Precompiler options
CPP_OPTIONS= -DHOST=\"LinuxIFC\"\
             -DMPI -DMPI_BLOCK=8000 -Duse_collective \
             -DscaLAPACK \
             -DCACHE_SIZE=4000 \
             -Davoidalloc \
             -Dvasp6 \
             -Duse_bse_te \
             -Dtbdyn \
             -Dfock_dblbuf

CPP          = fpp -f_com=no -free -w0  $*$(FUFFIX) $*$(SUFFIX) $(CPP_OPTIONS)
```

```
FC              = mpiifort
FCL             = mpiifort -mkl=sequential
FREE            = -free -names lowercase
FFLAGS          = -assume byterecl -w -xHOST
OFLAG           = -xCORE-AVX2
OFLAG_IN        = $(OFLAG)
DEBUG           = -O0

MKL_ROOT = /home-ext/apps/spack/opt/spack/linux-centos7-cascadelake/oneapi-2022.0.0/intel-oneapi-mkl-2022.0.1-iafmkapq755cglpc4c5lhzfpb27dc4in/mkl
MKL_PATH        = $(MKLROOT)/lib/intel64
BLAS            =
LAPACK          =
BLACS           = -lmkl_blacs_intelmpi_lp64
SCALAPACK       = $(MKL_PATH)/libmkl_scalapack_lp64.a $(BLACS)
OBJECTS         = fftmpiw.o fftmpi_map.o fft3dlib.o fftw3d.o
INCS            =-I$(MKLROOT)/include/fftw
LLIBS           = $(SCALAPACK) $(LAPACK) $(BLAS)
OBJECTS_O1 += fftw3d.o fftmpi.o fftmpiw.o
OBJECTS_O2 += fft3dlib.o

# For what used to be vasp.5.lib
CPP_LIB         = $(CPP)
FC_LIB          = $(FC)
CC_LIB          = icc
CFLAGS_LIB = -O
FFLAGS_LIB = -O1
```

```
FREE_LIB      = $(FREE)

OBJECTS_LIB= linpack_double.o getshmem.o

# For the parser library

CXX_PARS      = icpc

LLIBS           += -lstdc++


# Normally no need to change this

SRCDIR          = ../../src

BINDIR          = ../../bin


#=============================================

# GPU Stuff


CPP_GPU        = -DCUDA_GPU -DRPROMU_CPROJ_OVERLAP -DUSE_PINNED_MEMORY
-DCUFFT_MIN=28 -UscaLAPACK -Ufock_dblbuf

OBJECTS_GPU= fftmpiw.o fftmpi_map.o fft3dlib.o fftw3d_gpu.o fftmpiw_gpu.o

CC                = icc

CXX               = icpc

CFLAGS          = -fPIC -DADD_ -Wall -qopenmp -DMAGMA_WITH_MKL -DMAGMA_SETAFFINITY
-DGPUSHMEM=300 -DHAVE_CUBLAS

# Minimal requirement is CUDA >= 10.X. For "sm_80" you need CUDA >= 11.X.

CUDA_ROOT    ?= /usr/local/cuda/

NVCC              := $(CUDA_ROOT)/bin/nvcc -ccbin=icc -allow-unsupported-compiler

CUDA_LIB      := -L$(CUDA_ROOT)/lib64 -lnvToolsExt -lcudart -lcuda -lcufft -lcublas


GENCODE_ARCH        := -gencode=arch=compute_60,code=\"sm_60,compute_60\" \

                           -gencode=arch=compute_70,code=\"sm_70,compute_70\" \

                           -gencode=arch=compute_80,code=\"sm_80,compute_80\"
```

```
## For all legacy Intel MPI versions (before 2021)
#MPI_INC        = $(I_MPI_ROOT)/intel64/include/
# Or when you are using the Intel oneAPI compiler suite
MPI_INC        = $(I_MPI_ROOT)/include/

#########################################################################
make std
make gam
make ncl
Then;
ls -lrt bin | awk '{print $0}END{print NR " files found"}'
```

Note: -

1. Inside the Job submission script file please mention above module.