VASP – Application Execution Guidelines for vSMP Foundation Aggregated Virtual Machine

Overview

VASP is a multi-process application that uses MPI for inter-process communication. While it is possible to run VASP on the aggregation platform with any MPI implementation, using MPICH2 tuned for vSMP Foundation may yield a performance improvement of 5-15%.

Building VASP with vSMPICH2 and ifort

1. Add path to mpif90.

```
export PATH=<path to MPICH2>/bin:$PATH
```

2. Edit Makefile with the following:

```
FC=mpif90 -f90=ifort
BLAS=-L<MKL path> -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5 -lpthread
LAPACK=-L<MKL path> -lmkl_lapack
FFT3D=fftmpi.o fftmpi_map.o fft3dlib.o
```

3. CPP flags for VASP complex

```
CPP        = $(CPP_) -DMPI  -DHOST=\"LinuxIFC\" -DIFC \
                -Dkind8 -DNGZhalf -DCACHE_SIZE=16000 -DPGF90 -Davoidalloc \
           -DMPI_BLOCK=16000 -Duse_collective

For Gamma, add "-DwNGZhalf".
```

Running VASP with MPICH2 tuned for vSMP

Run VASP as you usually do and add the following environment variables:

```
export VSMP_PLACEMENT=PACKED
export VSMP_MEM_PIN=YES
mpirun -np <#> vasp
```

NPAR value in INCAR

The Performance of VASP varies depending upon selection of NPAR values and the specific job.

- Try power of 2 up to the total number of processors.
- NPAR is always a factor of the number of CPUs used
- The optimal value of NAPR is dependent on the input since VASP parallelism comes from 2 types of operations: Matrix calculations and FFTs. NPAR should be higher as the ratio between required matrix calculations vs. FFTs is higher.