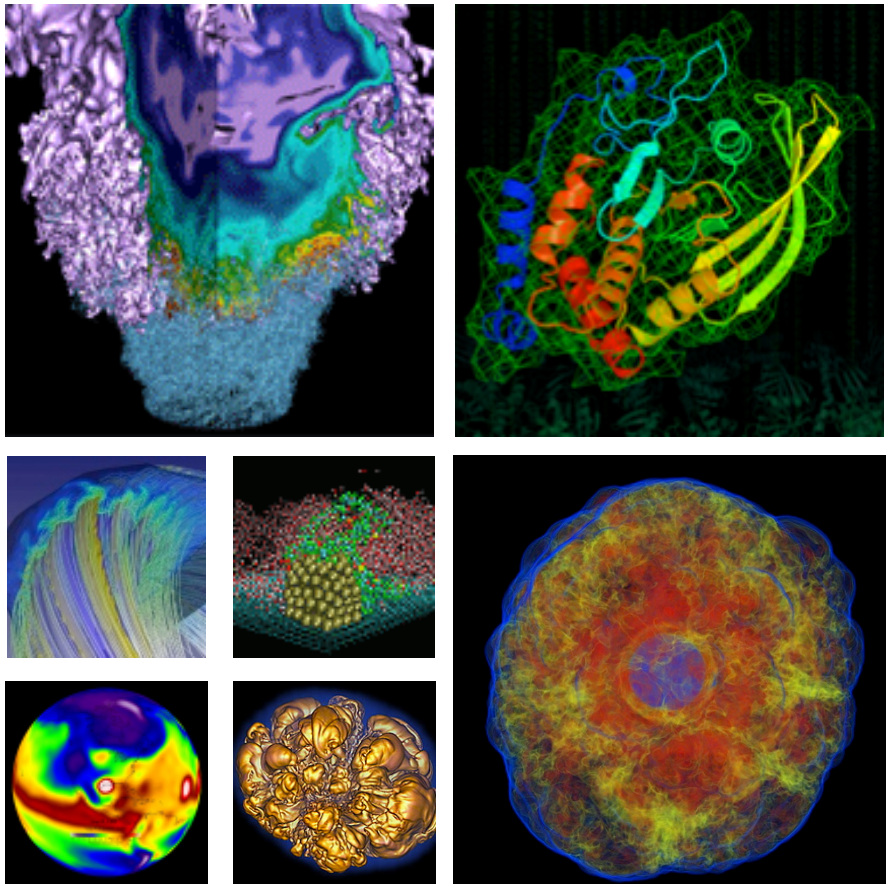
Using VASP at NERSC



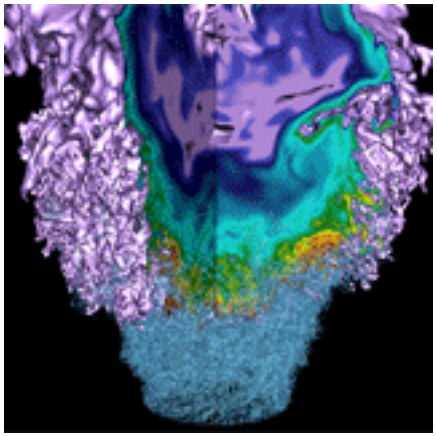
Zhengji Zhao
NERSC User Engagement Group

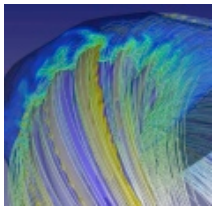
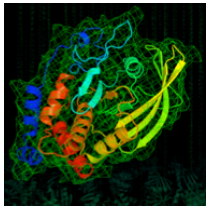
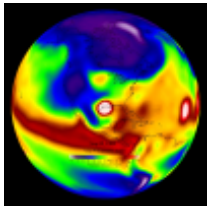
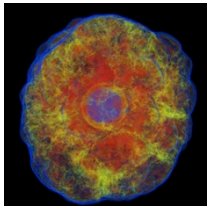
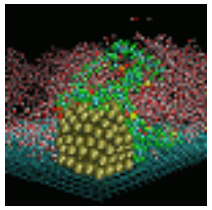
June 10, 2016

Agenda

- VASP access at NERSC
- Get started with available VASP builds at NERSC
- Common problems users run into
- Good practices
- Compiling VASP on NERSC systems

VASP Access at NERSC



NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

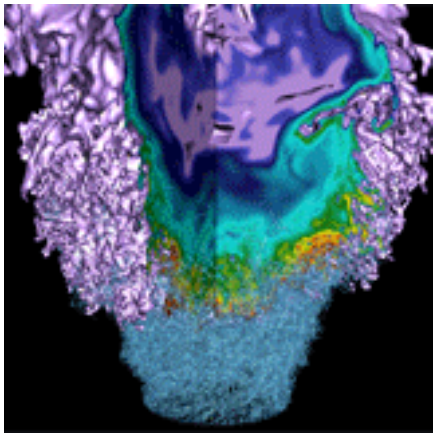
BERKELEY LAB

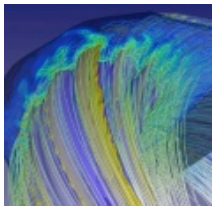
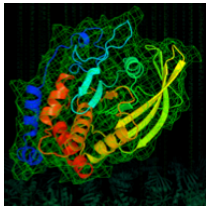
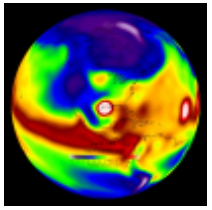
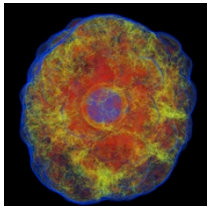
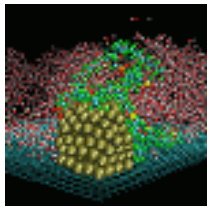
VASP is available to the users who have the license by themselves

- **Need to confirm your license with VASP developers**
  - Send your license info (license number, PI's name, etc) to *vasp.Materialphysik@univie.ac.at* and CC: *vasp_licensing@nersc.gov*
  - To avoid unnecessary delay, make sure your are a registered user under your PI's VASP license
  - This is a manual process. It may take a couple of days normally, sometime takes longer, e.g., when the VASP support staff is on vacation.
  - Once your license is confirmed by the VASP support at Vienna, NERSC gives you the access to the VASP binaries on NERSC machines. The access is controlled by a unix file group, vasp5, or vasp (for VASP 4). Type *groups* command to see if you have the access to the VASP binaries at NERSC.

https://www.nersc.gov/users/software/applications/materials-science/vasp/#toc-anchor-2

Get Started with Available VASP Builds



How many vasp builds are available at NERSC?

```
zz217@edison01:~> module avail vasp

---------------------- /usr/common/usg/Modules/modulefiles ----------------------
vasp/4.6.35_vtst      vasp/5.3.5          vasp/5.3.5_vtst-cce  vasp/5.4.1
vasp/5.3.2            vasp/5.3.5-cce(default)
vasp/5.3.2_vtst       vasp/5.3.5_vtst
```

```
zz217@cori01:~> module avail vasp
---------------------- /usr/common/software/modulefiles-----------------------------
vasp/5.3.5        vasp/5.3.5-cce     vasp/5.3.5_vtst    vasp/5.3.5_vtst-cce
vasp/5.4.1(default)
```

Modulefile naming convention: vasp/<version><_vtst><-compiler>

- Where the **version** is the official VASP release version; the **-compiler** is the compiler name that was used to build the code, when omitted, the Intel and Cray compilers were used for Edison and Hopper, respectively; and the **_vtst** denotes the builds with the third party contributed codes, VTST, Wannier90, etc.
- On Edison, vasp/5.3.5-cce is a build for the official VASP release, compiled with a Cray compiler; vasp/5.3.5_vtst-cce built with a Cray compiler, enabled VTST (U. Texas, Austin) and Wannier90.

How many VASP builds are available at NERSC? -cont

- **There are different compiler builds**
  - To provide alternatives because some problems may occur with one compiler build but not with another
  - On Edison, Intel compiler builds run faster than the Cray compiler builds; however, may run into LAPACK errors more often. So the default VASP module on Edison was built with a Cray compiler.
- **The default module is recommended**
- **To access, do: module load vasp**
- **For more info, do: module show vasp/<version string>**

What do the modulefiles do?

```
Edison01> module show vasp
-------------------------------------------------------------------
/usr/common/usg/Modules/modulefiles/vasp/5.3.5-cce:

module-whatis   VASP: Vienna Ab-initio Simulation Package

Access to the vasp suite is allowed only for research groups with existing
licenses for VASP. If you have a VASP license please email

  vasp.Materialphysik@univie.ac.at and CC: vasp_licensing@nersc.gov

with the information on which research group your license derives from.
The PI of the group as well as the institution and license number will help
speed the process.

setenv          PSEUDOPOTENTIAL_DIR /usr/common/usg/vasp/pseudopotentials/5.3.5
setenv          VDW_KERNAL_DIR /usr/common/usg/vasp/vdw_kernal
setenv          NO_STOP_MESSAGE 1
setenv          MPICH_NO_BUFFER_ALIAS_CHECK 1
prepend-path    PATH /usr/common/usg/vasp/vtstscripts/3.1
prepend-path    PATH /usr/common/usg/vasp/5.3.5-cce/bin
-------------------------------------------------------------------
```

Where do VASP binaries reside?

```
zz217@edison01> ls -ld /usr/common/usg/vasp/5.3.5-cce/bin
drwxr-x---+ 4 zz217 vasp5 512 Jan  6 21:24 /usr/common/usg/vasp/5.3.5-cce/bin

zz217@edison01> ls -l /usr/common/usg/vasp/5.3.5-cce/bin
total 395660
-rwxrwxr-x+ 1 zz217 usg   69225563 Jul 31  2014 gvasp
-rwxrwxr-x+ 1 zz217 usg   66153550 Aug 21  2014 makeparam
-rwxrwxr-x+ 1 zz217 usg   69601938 Aug 22  2014 vasp
-rwxr-xr-x+ 1 zz217 usg   69604098 Jul 31  2014 vasp.NMAX_DEG=128
-rwxrwxr-x+ 1 zz217 usg   60887560 Aug 21  2014 vasp.serial
lrwxrwxrwx  1 zz217 zz217       5 Jan  6 21:24 vasp_gam -> gvasp
-rwxrwxr-x+ 1 zz217 usg   69676474 Jul 31  2014 vasp_ncl
lrwxrwxrwx  1 zz217 zz217       4 Jan  6 21:24 vasp_std -> vasp
```

vasp – general kpoint VASP build

gvasp – Gamma point only build

vasp_ncl – Non-collinear version

Other binaries are special builds per user request

Note, the links to the vasp_std, vasp_gam to be consistent with vasp/5.4.1 version

VASP builds with VTST and Wannier90 enabled

```
edison01> module show vasp/5.3.5_vtst-cce
-------------------------------------------------------------------
/usr/common/usg/Modules/modulefiles/vasp/5.3.5_vtst-cce:

module-whatis   VASP: Vienna Ab-initio Simulation Package

Note: This build enabled the VTST 3.1 code (U. Texas, Austin) and Wannier90 1.2.0.

Access to the vasp suite is allowed only for research groups with existing
licenses for VASP. If you have a VASP license please email

  vasp.Materialphysik@univie.ac.at and CC: vasp_licensing@nersc.gov

with the information on which research group your license derives from.
The PI of the group as well as the institution and license number will help
speed the process.

setenv          PSEUDOPOTENTIAL_DIR /usr/common/usg/vasp/pseudopotentials/5.3.5
setenv          VDW_KERNAL_DIR /usr/common/usg/vasp/vdw_kernal
setenv          NO_STOP_MESSAGE 1
setenv          MPICH_NO_BUFFER_ALIAS_CHECK 1
prepend-path    PATH /usr/common/usg/vasp/vtstscripts/3.1
prepend-path    PATH /usr/common/usg/vasp/5.3.5_vtst-cce/bin
-------------------------------------------------------------------
```

VASP builds with VTST and Wannier90 enabled

```
zz217@edison01> ls -l  /usr/common/usg/vasp/5.3.5_vtst-cce/bin
total 340624
-rwxrwxr-x 1 zz217 usg   100514105 Feb 19 16:21 gvasp
-rwxrwxr-x 1 zz217 usg   101455347 Feb 19 16:21 vasp
lrwxrwxrwx 1 zz217 zz217        5 Feb 17 16:33 vasp_gam -> gvasp
-rwxrwxr-x 1 zz217 usg   101515371 Feb 19 16:21 vasp_ncl
lrwxrwxrwx 1 zz217 zz217        4 Feb 17 16:33 vasp_std -> vasp
-rwxrwxr-x 1 zz217 usg    45308309 May  8  2015 wannier90.x
```

Get started with VASP at NERSC

Running vasp via a batch job

```
edison01> cat run.slurm
#!/bin/bash -l
#SBATCH -J test_vasp
#SBATCH -p debug
#SBATCH -N 2
#SBATCH -t 00:30:00
#SBATCH –o test_vasp.o%j

module load vasp
srun -n 48 vasp_std

edison01> sbatch run.slurm
Submitted batch job 518354
```

Running interactively

```
edison01> salloc –N 2 –p debug –t 30:00
…
module load vasp
srun -n 48 vasp_std
```

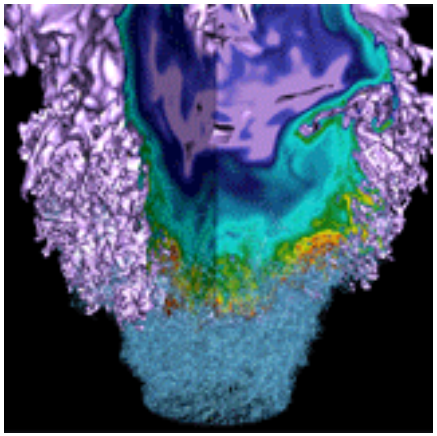
Commonly used commands:

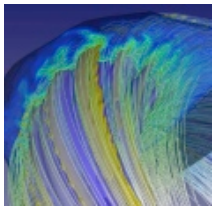
```
squeue –u <username>
scancel <jobid>
scontrol show  <jobid>
sqs –u <username>
sinfo –s
scontrol hold <jobid>
scontrol release <jobid>
Scontrol update job <jobid> TimeLimit=24:00:00
Scontrol update job <jobid> QOS=premium
For more info, read man pages
```

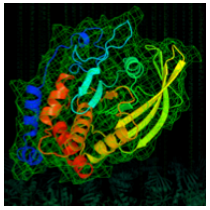
```
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH –A <your repo>
```

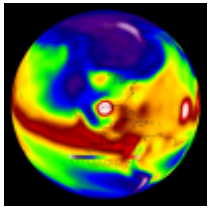
https://www.nersc.gov/users/software/applications/materials-science/vasp/

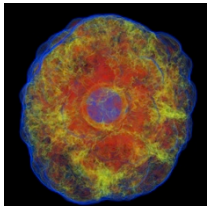
Common Problems Users Run into

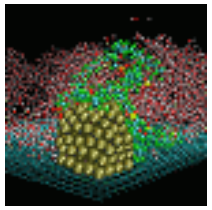












VASP not found error

- **No access to VASP – check if you have access to the VASP binaries at NERSC. If not, confirm your license.**

  usgsw@edison01:~> groups

  usgsw oprofile usg # user usgsw does not have access to VASP

- **Failed to load a vasp module –**
  - You will also see an error in your job standard error file: module: command not found

Running VASP jobs with too many cores

- **VASP in general considered to scale up to 1 core/atom. If running outside the parallel scaling region, various errors may occur**
- **internal ERROR RSPHER:running out of buffer**
  - This error was seen when using too many cores for a small system, and was also seen when a small NPAR + LPLANE is used, so that number of cores per orbital is similar or even larger than NGZ (VASP manual: *LPLANE=.TRUE. should only be used if NGZ is at least 3*(number of cores)/NPAR*).
  - There could be multiple fixes including modifying the source code to allocate larger work arrays, but you can eliminate this error by reducing the total number of cores or using a larger NPAR (closer to the default value), or setting LPALNE=.FALSE.

Running with too small number of cores - Out of Memory error

- **Error message: OOM killer terminated this process**
  - Edison 2.6 GB/core; 24 cores per node
  - Cori $GB GB/core; 32 cores per node
- **To avoid**
  - Consider to use ~ 1 core/atom
  - Gamma point calculations, use gvasp instead of vasp
  - Use more cores and/or run on unpacked nodes, e.g., use 12 core/node

How many cores to use?

- **Using larger number of cores may not reduce the time to solution cost effectively**
  - VASP may spend most of the time in the communication even if it does not run into error
- **1 Core/atom is a good reference**
  - Conservatively, you may go with 0.5 core/atom or slightly less
  - Use NPAR ~ sqrt (total number of cores used) where applicable
  - If there are many kpoints, use 0.5 core/atom for each kpoint group; the total number of cores = KPAR x 0.5 core/atom x (# of atoms)
  - If there are multiple images, total number of cores = IMAGES x 0.5 core/atom x (# of atoms)
- **This will also effectively reduce the Out of Memory (OOM) error**

Error when total number of cores is not divisible by NPAR

- **M_divide: can not subdivide**
- **NPAR default is the same as the total number of cores**
  - Max NPAR=256
  - NPAR ~ sqrt (total number of cores) performs better than NPAR=1 or the default NPAR(= total number of cores) when applicable

Error due to aliasing buffers in MPI collective calls in VASP code

- **Fatal error in PMPI_Allgatherv: Invalid buffer pointer, error stack: PMPI_Allgatherv(1235): MPI_Allgatherv(sbuf=0x9ed6000, scount=64, MPI_DOUBLE_COMPLEX, rbuf=0x9ed6000, rcounts=0xb27f7c0, displs=0xad81140, MPI_DOUBLE_COMPLEX, comm=0xc4000003) failed PMPI_Allgatherv(1183): Buffers must not be aliased. Consider using MPI_IN_PLACE or setting MPICH_NO_BUFFER_ALIAS_CHECK**
  - This error was seen with HSE jobs

    export MPICH_NO_BUFFER_ALIAS_CHECK=1 #for bash/sh
    setenv MPICH_NO_BUFFER_ALIAS_CHECK 1 # for csh/tcsh
  - In NERSC VASP modules, we set MPICH_NO_BUFFER_ALIAS_CHECK=1 to bypass the buffer aliasing error checking in MPI collective calls.
  - If you run your own VASP builds may need to set this env

Hung jobs – difficult to debug

- **File system issues – Lustre, global homes**
  - File systems may slow or hang due to hardware/software issues, excessive use from some users, users over $HOME quota, etc. Jobs may hang or run slower due to under-performing file systems
- **Srun –u to disable the IO buffering of Slurm**

LAPACK's ZHEGV failure

- **Error EDDDAV: Call to ZHEGV failed. Returncode = 25 2 48**
  - ZHEGV solves (diagonalizes) the complex generalized Hermitian eigenproblem A*x=(lambda)*B*x, where B must be positive definite (i.e., its eigenvalues are positive).
  - For some systems VASP may generate a matrix B that is not positive definite (the reason for this is not well understood) and ZHEGV returns an error code. --comments provided by Osni Marques at LBNL.
- **This error was more frequently seen with Intel builds on Edison**
- **Switching to Cray builds may help**
- **Edison default vasp module, vasp/5.3.5-cce, was built with a Cray compiler**

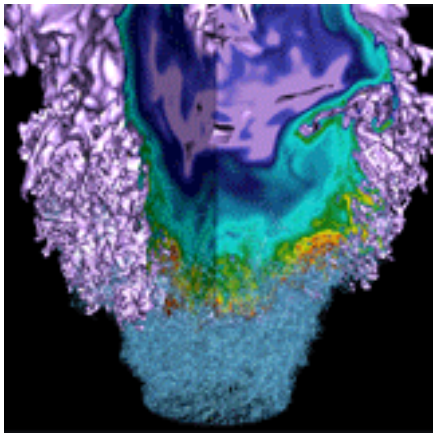
Known issues with the Wannier90 enabled builds

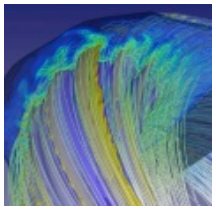
- **Cray compiler builds do not work well with Wannierf90 both on Edison and Cori**
  - lib-4095 : UNRECOVERABLE library error
    A WRITE operation is invalid if the file is positioned after the end-of-file.

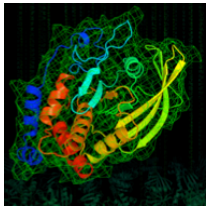
Be aware that jobs may fail due to various system issues

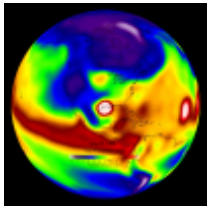
- **System hardware and software failures occur**
  - File systems under-performing
  - File systems not available, GPFS expel
  - Batch system error
  - Upgrades on OS, system, library and application software
  - Node failure
  - Network failure
  - Power glitch
- **Users misuse the system may affect other users**
  - Pounding the file systems, and make them unresponsive affecting all user jobs running out of the same file systems
  - Running I/O intensive parallel jobs out of $HOME
- **Report problems to consult@nersc.gov**
  - Help us to resolve the problem faster

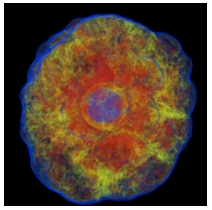
Good practices

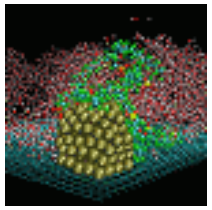












Which system to use?

Goal: get more computing done with a given allocation within a given time period

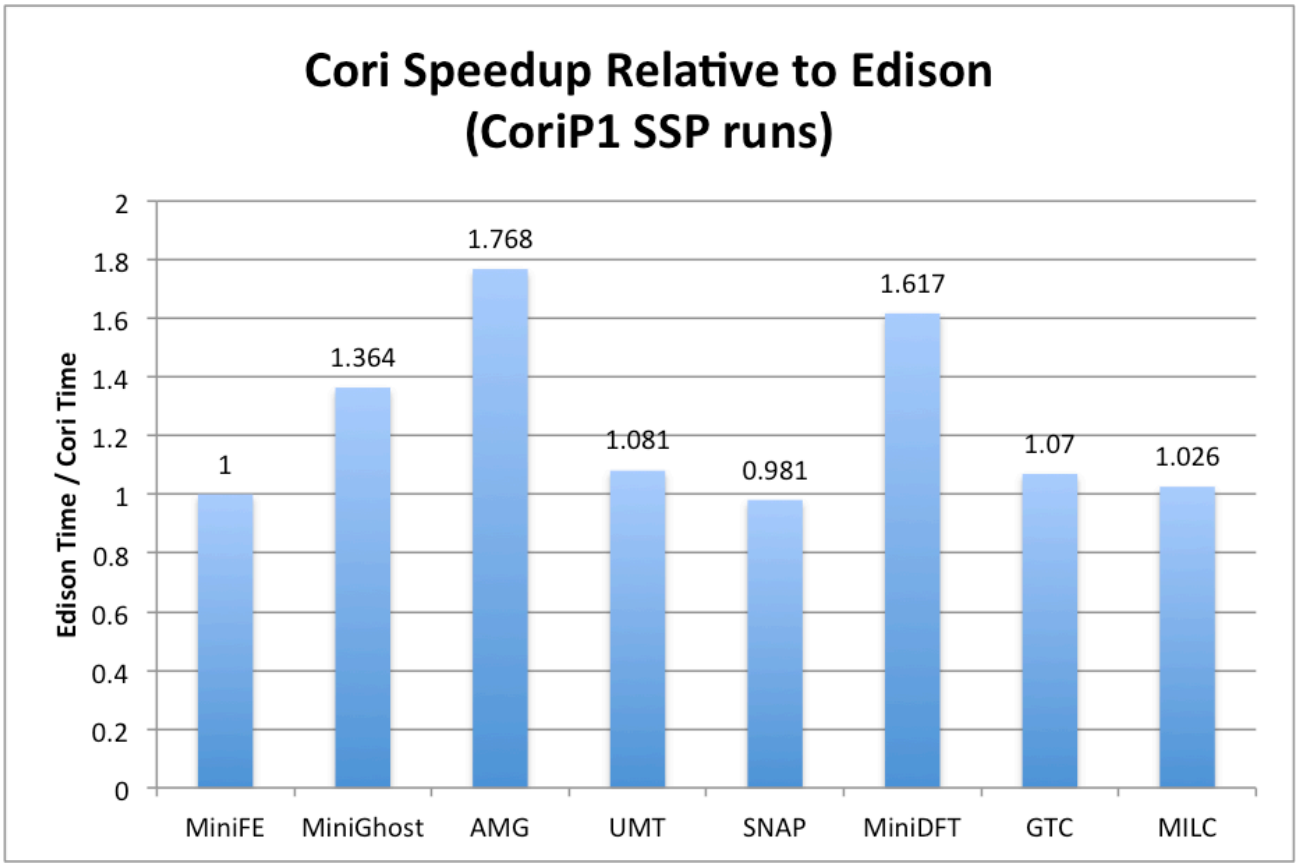
- **Charging:**
  - Edison has a machine charge factor 2, however, since most codes run twice faster on Edison, the MPP charging for the same core jobs on both systems should be similar. **VASP on Edison can easily outperform Hopper by 2-3 times.**

NERSC-6 Application Benchmarks

Application	CAM	GAMESS	GTC	IMPACT-T	MAESTRO	MILC	PARATEC
Concurrency	240	1024	2048	1024	2048	8192	1024
Streams/Core	2	2	2	2	1	1	1
Edison Time (s)	273.08	1,125.80	863.88	579.78	935.45	446.36	173.51
Hopper Time(s)	348	1389	1338	618	1901	921	353
Speedup[1)]	**2.5**	**2.5**	**3.1**	**2.1**	**2.0**	**2.1**	**2.0**

[1)] Speedup=[Time(Hopper)/Time(Edison)]*Streams/Core

Which system to use - cont



- Cori has a charge factor of 2.5 while Edison is 2. As long as an application runs more than 25% faster on Cori, running on Cori would be more MPP charging efficient.
- VASP would be about 30% faster on Cori (same core), so both systems should be similar in MPP charge.

Helen He at NERSC provided the figure

Which system to use? - cont

- **Queue policy:**
  - Edison favors larger jobs, the small regular jobs (1-682 nodes) have a lower priority on Edison. Cori has the same priority for all jobs.
  - In general shorter jobs have more backfill opportunity. Make sure to avoid unnecessary long walltime requests.
  - Bundling up many small regular jobs, if they do similar computations, to get a higher queue priority and a charging discount.
- **You can use either system to run your jobs**

https://www.nersc.gov/users/computational-systems/edison/running-jobs/queues-and-policies/
https://www.nersc.gov/users/computational-systems/cori/running-jobs/queues-and-policies/

Choose the right file systems to run jobs on

- **Your global homes, $HOME, is not the right file system to run your VASP jobs**
  - You may exceed your home quota, 40GB, and cause system issues.
  - Slowdown yourself and also other users
- **The Lustre file systems ($SCRATCH) are recommended file systems to run your jobs both for a larger storage space and a better I/O performance.**
  - 10TB scratch quota on Edison; 100TB on /scratch3 (for I/O intensive workload only)
  - 20TB scratch quota on Cori (/cscratch1)
  - Note: files are not accessed for more than 8 weeks on Edison and 12 weeks on Cori will be purged
  - You can back up your important files to your project directory, /project/projectdires/<your repo>, which has 4TB quota, or to HPSS after analysis.

Short scaling tests are recommended to choose optimal core counts and NPAR

- **The debug queue has a high priority on Edison and Cori**

  INCAR:

  LWAVE= .FALSE.

  LCHARG= .FALSE.

  NELMDL= -1

  NELM=2

  NSW=1

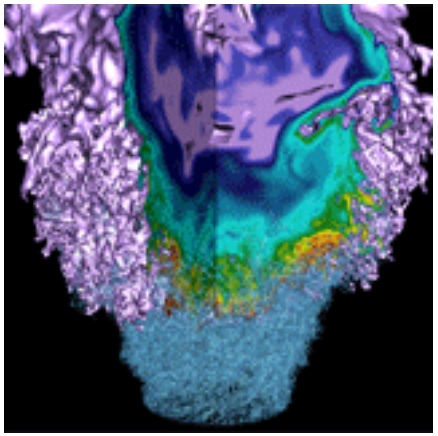
  http://cms.mpi.univie.ac.at/vasp/vasp/Parallelisation_NPAR_NCORE_LPLANE_KPAR_tag.html

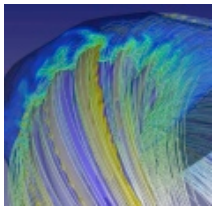
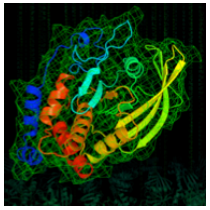
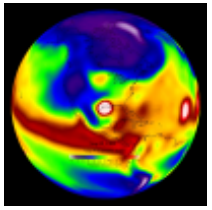
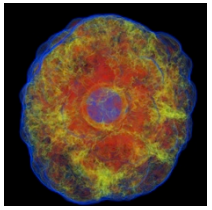
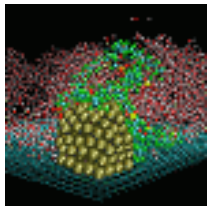
- **Also do debug runs to see if your jobs could run to completion before submitting your long jobs**
  - Make sure no memory and other failures during the run

Ask questions and send problem reports to NERSC consultants

- **Email: consult@nersc.gov**
- **Phone: 1-800-666-3772 menu option 3.**
- **We may not solve all your problems, but we may be able to help you to find workarounds**

Compiling VASP on NERSC systems



NERSC 40 YEARS at the FOREFRONT 1974-2014

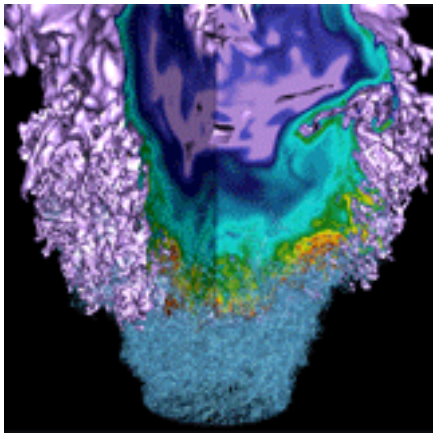
U.S. DEPARTMENT OF ENERGY | Office of Science

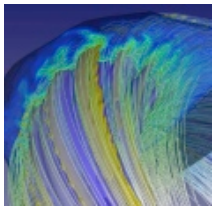
BERKELEY LAB

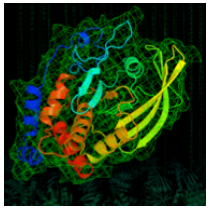
Where to find the makefiles

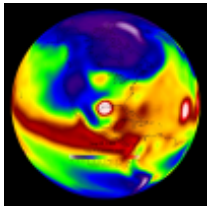
- **We provide makefiles for the users who want to compile the codes by themselves – open to everyone, do not need to confirm your license to access the makefiles**
- **The makefiles are available at the vasp installation directories**
  - Instdir=/usr/common/usg on Edison; Instdir=/usr/common/software on Cori
  - VASP 5.3.5: $instdir/vasp/<version-string>/makefiles
  - VASP 5.4.1: the makefile.include is available at $instdir/vasp/5.4.1.
  - Try the build script, build.sh for VASP 5.3.5, or run make after get the make.include file
  - To compiler other older versions (<5.3.5) you may just need to replace the SOURCE variable in the makefile with the list of source files that match your VASP version (get the source list from the makefiles that were distributed with your vasp code).
- **VASP was built with two compilers in general, Intel and Cray compilers on Edison and Cori.**
  - Intel Compiler +MKL + fftw3 wrapper routines from MKL
  - Cray Compiler +Libsci + fftw 3

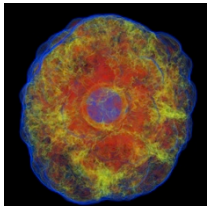
Thank you

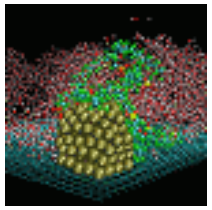












NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB