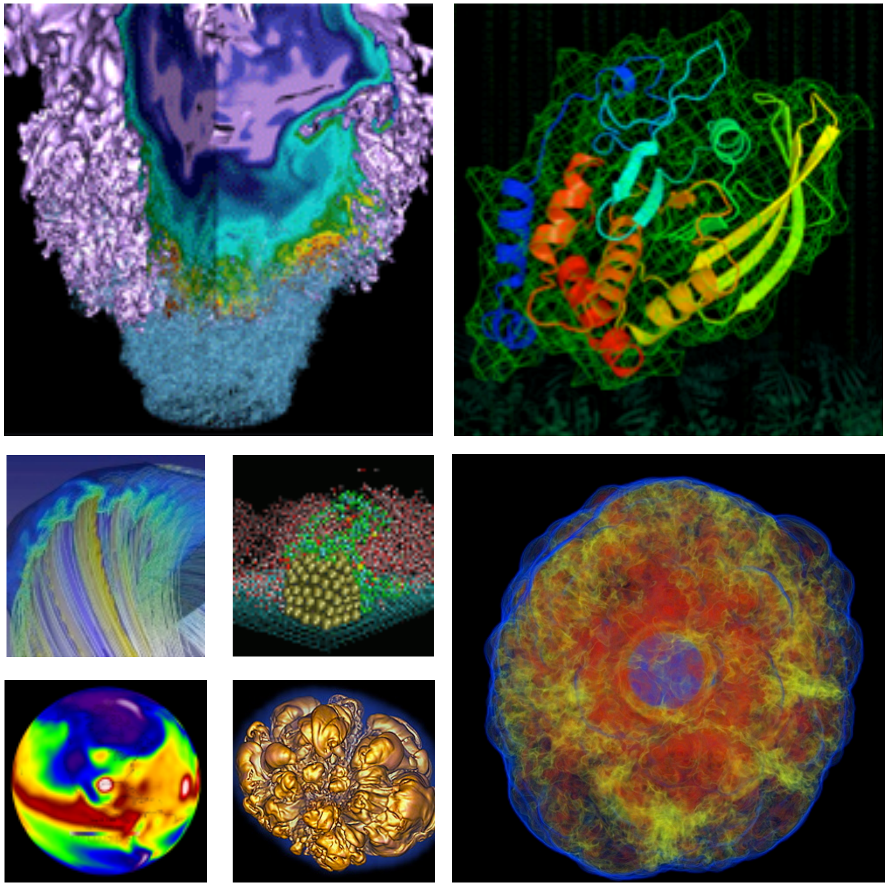
Using VASP at NERSC



NERSC 40 YEARS at the FOREFRONT 1974-2014

Zhengji Zhao
NERSC User Services Group

June 5, 2015

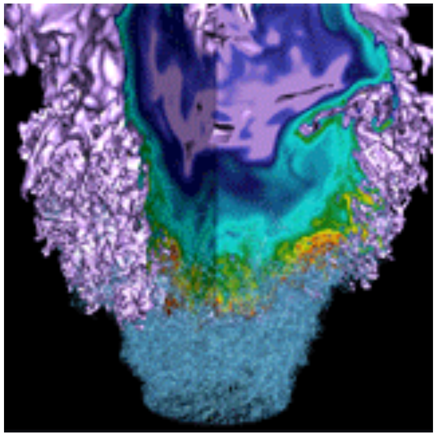
U.S. DEPARTMENT OF ENERGY | Office of Science

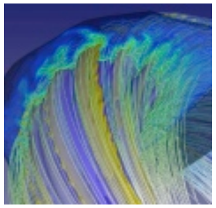
BERKELEY LAB
Lawrence Berkeley National Laboratory

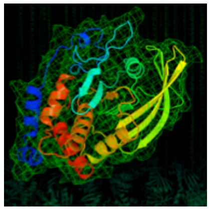
Agenda

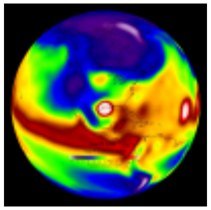
- VASP access at NERSC
- Get started with available VASP builds at NERSC
- Common problems users run into
- Good practices
- Compiling VASP on NERSC systems

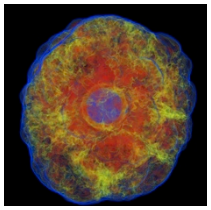
VASP Access at NERSC

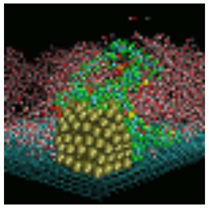










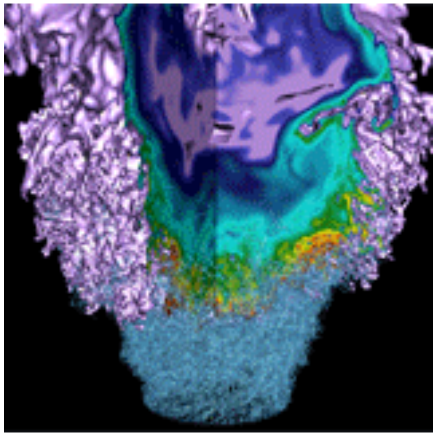


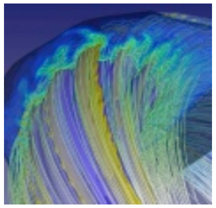
VASP is available to the users who have the license by themselves

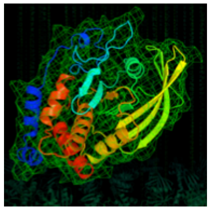
- **Need to confirm your license with VASP developers**
  - Send your license info (license number, PI's name, etc) to *vasp.Materialphysik@univie.ac.at* and CC: *vasp_licensing@nersc.gov*
  - To avoid unnecessary delay, make sure your are a registered user under your PI's VASP license
  - This is a 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.

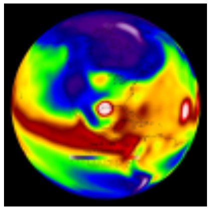
https://www.nersc.gov/users/software/applications/materials-science/vasp/#toc-anchor-2

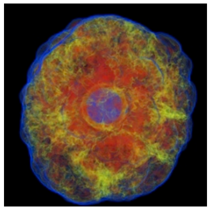
Get Started with Available VASP Builds

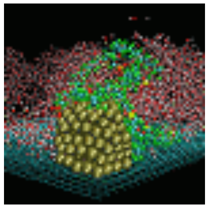












How many vasp builds are available at NERSC?

```
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/5.3.2             vasp/5.3.5-cce(default)
vasp/5.3.2_vtst        vasp/5.3.5_vtst
```

```
zz217@hopper06:~> module avail vasp

----------------------- /usr/common/usg/Modules/modulefiles ------------------------
vasp/4.6.35        vasp/5.3.2-pgi      vasp/5.3.3_vtst-pgi vasp/5.3.5_vtst
vasp/4.6.35.pkent   vasp/5.3.2_vtst-pgi vasp/5.3.5(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.
  - On Hopper Cray compiler builds are faster
- **Recommended: the default module**
- **To access, do: module load vasp**
- **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?

```
edison01> ls -ld /usr/common/usg/vasp/5.3.5-cce/bin
drwxr-x---+ 4 zz217 vasp5 512 Aug 22  2014 /usr/common/usg/vasp/5.3.5-cce/bin

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
-rwxrwxr-x+ 1 zz217 usg 69676474 Jul 31  2014 vasp_ncl
-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
```

vasp – general kpoint VASP build

gvasp – Gamma point only build

vasp_ncl – Non-collinear version

Other binaries are special builds per user request

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

```
edison01> ls -l /usr/common/usg/vasp/5.3.5_vtst-cce/bin
total 281800
-rwxrwxr-x 1 zz217 usg 80469818 May  8 14:26 gvasp
-rwxrwxr-x 1 zz217 usg 81125190 May  8 14:26 vasp
-rwxrwxr-x 1 zz217 usg 81657552 Mar 13 10:20 vasp_ncl
-rwxrwxr-x 1 zz217 usg 45308309 May  8 14:02 wannier90.x
```

Get started with VASP at NERSC

```
edison01> cat run.pbs
#PBS -N a154
#PBS -q regular
#PBS -l mppwidth=72
#PBS -l walltime=12:00:00
#PBS -j oe

cd $PBS_O_WORKDIR

module load vasp

aprun -n 72 vasp

edison01> qsub run.pbs
3024984.edique02
```

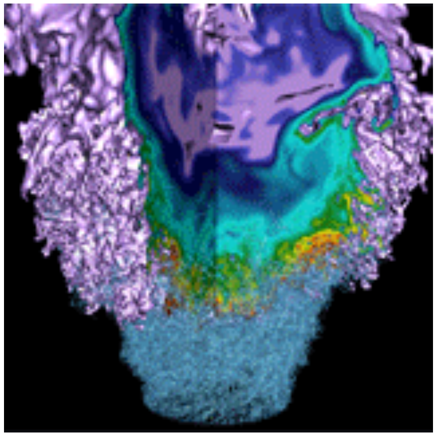
Commonly used commands:

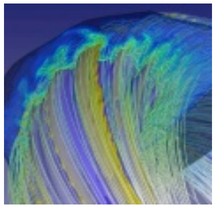
```
qstat –u <username>
qstat –f <jobid>
qs –u <username>
showq
checkjob –v <your jobid>
qdel
qhold <jobid>
qrls <jobid>
qalter –l walltime=24:00:00 <jobid>
```

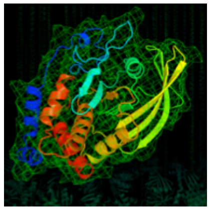
See man pages of these commands for more details

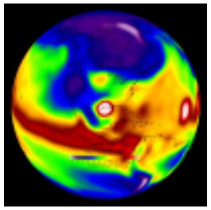
https://www.nersc.gov/users/software/applications/materials-science/vasp/

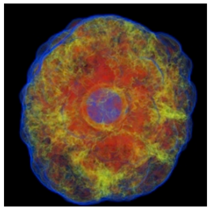
Common Problems Users Run into

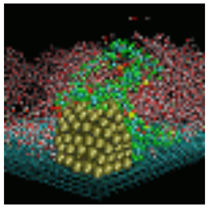












VASP not found error

- **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
  - Csh users may see this when invoking bash as non-login, non interactive shell, invoking login bash shell in your job script may help: #!/bin/bash –l

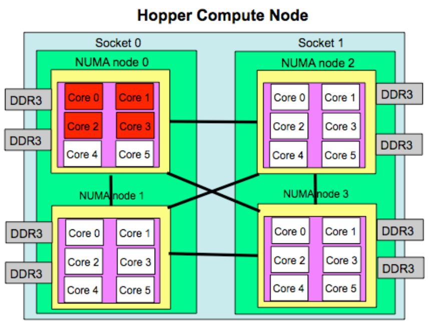
Running VASP jobs with too many cores

- **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.

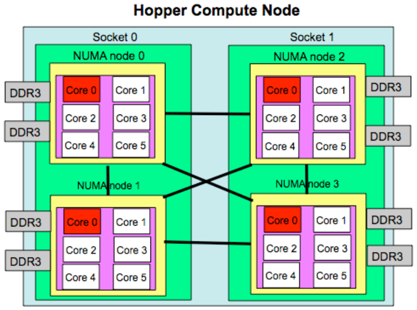
Running with too small number of cores - Out of Memory error

- **Error message: OOM killer terminated this process**
  - Edison 2.6 GB/core; Hopper 1.3 GB/core; 24 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
  - When running on unpacked nodes, spread MPI tasks evenly on the NUMA nodes on the node for optimal performance using the –N –S options of aprun

Non-optimal: aprun –n128 –N4



Optimal: aprun –n128 –N4 –S1



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 1 core/atom for each kpoint group; the total number of cores = KPAR x 1 core/atom x (# of atoms)
  - If there are multiple images, total number of cores = IMAGES x 1 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 this env to be set

Hung jobs – difficult to debug

- **If your VASP jobs hang or run slower, set the env UGNI_CDM_MDD_DEDICATED=2 to disable the shared memory domain descriptors (MDDs), this may or may not help.**

```
setenv UGNI_CDM_MDD_DEDICATED 2  #for csh/tcsh users
export UGNI_CDM_MDD_DEDICATED=2 #for bash shell users
```

- **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

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 Hopper**
  - lib-4095 : UNRECOVERABLE library error
    A WRITE operation is invalid if the file is positioned after the end-of-file.
- **Use Intel or PGI compiler builds for Wannier90 enabled VASP runs on Edison and Hopper, respectively**

Fatal error in MPI_Allreduce: Message truncated

- Fatal error in MPI_Allreduce: Message truncated, error stack:

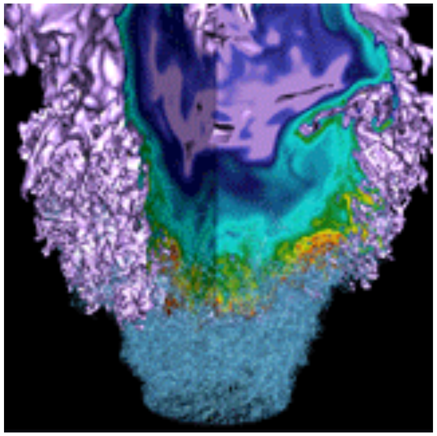
  ```
  MPI_Allreduce(1005)............: MPI_Allreduce(sbuf=0x10c789c0,
  rbuf=0x3e89380, count=29160, MPI_DOUBLE_PRECISION, MPI_SUM,
  comm=0x84000006) failed
  MPIR_Allreduce_impl(857).......:
  MPIR_Allreduce_intra(267)......:
  MPIR_Reduce_impl(1200).........:
  MPIR_Reduce_intra(984).........:
  MPIR_Reduce_redscat_gather(561):
  do_cts(559)...................: Message truncated; 170128 bytes received but
  buffer size is 116640
  MPIR_Allreduce_intra(297)......:
  MPIR_Bcast_impl(1455)..........:
  MPIR_CRAY_Bcast(440)...........:
  MPIR_CRAY_Bcast_Tree(152)......:
  ```

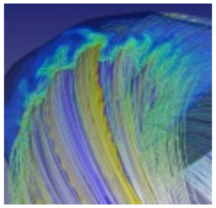
- This error was seen with the VTST enabled VASP when KPAR was used
- Still under investigation, no good solution yet

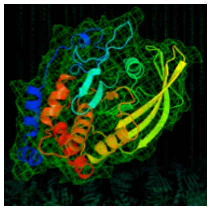
Be aware that jobs may fail due to various system issues

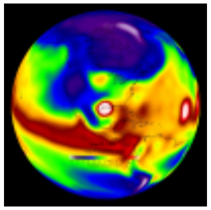
- **System hardware and software failures occur**
  - File systems under-performing
  - File systems not available, GPFS expel
  - Batch system error
  - Upgrades on OS, system, library and application software
  - Node failure
  - Network failure
  - Power glitch
- **Users misuse the system may affect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

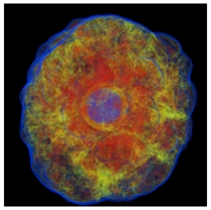
Good practices

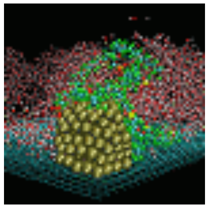












Which machine to use?

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 machine to use? - cont

- **Queue policy:**
  - Edison favors larger jobs, the reg_small jobs have a lower priority on Edison. Specifically, the reg_xbig, reg_big, and reg_med jobs all have a 3-day queue priority boost with a significant charging discount. Hopper does not favor larger jobs (anymore) both in queue priority and charging discount, so your reg_small jobs should have a better queue turnaround (but jobs will run twice longer by average).
  - You can submit shorter dependency jobs to create backfill opportunity for your reg_small jobs on Edison (and on Hopper). Or bundling up many reg_small jobs, if they do similar computations, to get a higher queue priority and a charging discount.
  - For dependency jobs, see

    https://www.nersc.gov/users/computational-systems/edison/running-jobs/example-batch-scripts/#toc-anchor-10
  - For bundling jobs, see

    https://www.nersc.gov/users/computational-systems/edison/running-jobs/example-batch-scripts/#toc-anchor-5
- **You can use either system to run your jobs**

Choose 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
  - 5TB combined scratch quota on Hopper (/scratch1 + /scratch2)
  - Note: files more than 12 weeks old (by access time) will be purged
  - You can back up your files to your project directory, /project/projectdires/<your repo>, which has 4TB quota, after 12 weeks, or to HPSS after analysis.

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

- **The debug queue has the highest priority on Edison and Hopper (4 day boost)**

  INCAR:

  ```
  LWAVE= .FALSE.
  LCHARG= .FALSE.
  NELMDL= -1
  NELM=2
  NSW=1
  ```

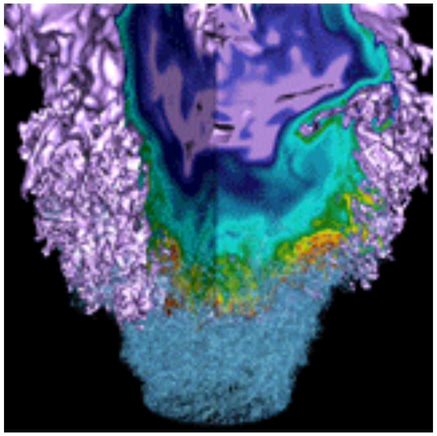
  http://cms.mpi.univie.ac.at/vasp/vasp/Parallelisation_NPAR_NCORE_LPLANE_KPAR_tag.html

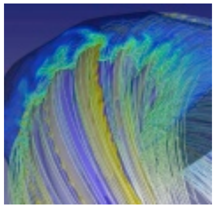
- **Also do debug runs to see if your jobs could run to completion before submitting your long jobs**
  - Make sure no memory failure during the run
  - You can also run the top command to monitor your job's memory usage over time on the compute nodes. See the backup slides about how to do this on Edison and Hopper.

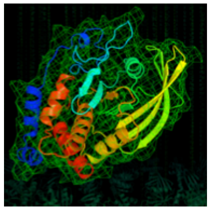
Ask questions and send problem reports to NERSC consultants

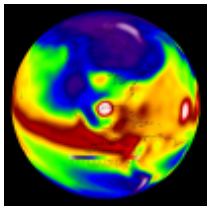
- **Email: consult@nersc.gov**
- **Phone: 1-800-666-3772**
- **We may not solve all your problems, but we may be able to help you to find workarounds**

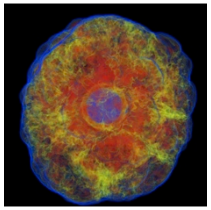
Compiling VASP on NERSC systems

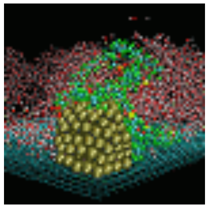








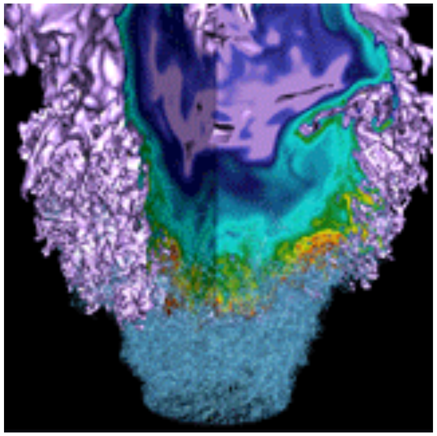


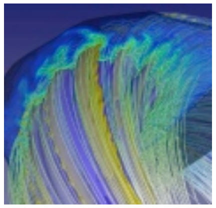


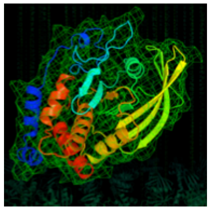
Where to find the makefiles

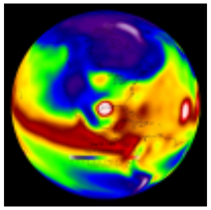
- **We provide makefiles for the users who want to compile the codes by themselves – open to everyone, do not need to confirm your license to access the makefiles**
- **The makefiles are available at /usr/common/usg/vasp/<version-string>/makefiles, try the build script, build.sh**
- **VASP was built with two compilers in general, Intel and Cray compilers on Edison; or Cray and PGI compilers on Hopper**
  - Intel Compiler +MKL + fftw3 wrapper routines from MKL, you are encouraged to try more aggressive Intel optimization flags, http://theor.jinr.ru/guide/soft/intel/Quick-Reference-Card-Intel-Compilers-v15.pdf
  - Cray Compiler +Libsci + fftw 3
  - PGI compiler +libsci + fftw 3

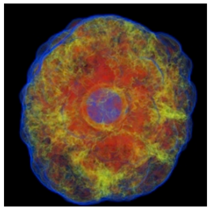
Thank you

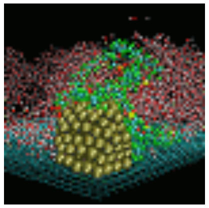












NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

Backup slides

How to run the top command on Edison/ Hopper compute nodes?

- **Cray compute nodes used to be black boxes to users, and you couldn't run the top command on the compute nodes where your job is running on.**
- **Cray has provided a tool called pcmd, which enables you to run almost any commands on the set of nodes that are allocated to your job.**
- **The pcmd is available on mom nodes only, to use**
  - ssh edimom01 #Edison has 24 mom nodes 01, 02, …, 24
  - module load nodehealth
  - apstat|grep <your username> # to find your job's apid
  - pcmd -a <apid> "top -b -n 1 |head -40"
  - pcmd –n 5560 "cat /proc/meminfo" # -n 5560 means to run the cat command on the node, nid05560, allocated to your job.
  - man pcmd   #for more info