VASP on GPUs

When and how

Max Hutchinson

University of Chicago

November 18, 2015

Big thanks to

Carnegie Mellon group

- Michael Widom

ENS/IFPEN group

- Paul Fleurat-Lessard
- Thomas Guignon
- Ani Anciaux-Sedrakian
- Philippe Sautet

RWTH Aachen Group

- Stefan Maintz
- Bernhard Eck
- Richard Dronskowski

Big thanks to

University of Vienna group

- Georg Kresse
- Martijn Marsman
- Doris Vogtenhuber

NVIDIA

- Christoph Angerer
- Jeroen Bédorf
- Arash Ashari
- Mark Berger
- Sarah Tariq
- Dusan Stosic
- Paul Springer
- Jerry Chen
- Anthony Scudiero
- Darko Stosic
- Przemek Tredak
- Cliff Woolley

VASP on GPUs

When and how

Max Hutchinson
University of Chicago

November 18, 2015

What is VASP?

VASP is a complex package for performing ab-initio quantum-mechanical molecular dynamics (MD) simulations using pseudopotentials or the projector-augmented wave method and a plane wave basis set[1].

[1] VASP the GUIDE

Why VASP?

12-20% of CPU cycles @ HPC centers

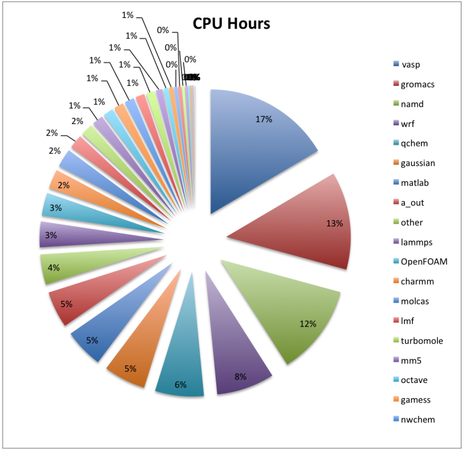
Academia

- Physics
- Physical chemistry
- Materials science
- Chemical engineering

Industry

- Materials
- Big semiconductor
- Oil and gas
- Chemicals

Usage @ Ohio SC's Oakley [2]



[2] 12/14 – 2/15, via pbsacct

A brief history

Multiple prototypes (2009-2012)

- Diagonalization for traditional DFT[3][4](IFPEN, ENS, Aachen)
- Exact-exchange for hybrid functionals[5](CMU, UChicago)

Cooperation and tuning (2012 - 2014)

- Merge prototypes with VASP 5.3.1
- Performance tune with NVIDIA engineers

[3] M. Hacene et al., DOI:10.1002/jcc.23096

[4] S. Maintz et al., DOI:10.1016/j.cpc.2011.03.010

[5] M. Hutchinson and M. Widom, DOI:10.1016/j.cpc.2012.02.017

A brief history

Multiple prototypes (2009-2012)

- Diagonalization for traditional DFT[3,4](IFPEN, ENS, Aachen)
- Exact-exchange for hybrid functionals[5](CMU, UChicago)

Cooperation and tuning (2012 - 2014)

- Merge prototypes with VASP 5.3.1
- Performance tune with NVIDIA engineers

[3] M. Hacene et al., `DOI:10.1002/jcc.23096`

[4] S. Maintz et al., `DOI:10.1016/j.cpc.2011.03.010`

[5] M. Hutchinson and M. Widom, `DOI:10.1016/j.cpc.2012.02.017`

A brief history

Multiple prototypes (2009-2012)

- Diagonalization for traditional DFT[3][4](IFPEN, ENS, Aachen)
- Exact-exchange for hybrid functionals[5](CMU, UChicago)

Cooperation and tuning (2012 - 2014)

- Merge prototypes with VASP 5.3.1
- Performance tune with NVIDIA engineers

[3] M. Hacene et al., DOI:10.1002/jcc.23096

[4] S. Maintz et al., DOI:10.1016/j.cpc.2011.03.010

[5] M. Hutchinson and M. Widom, DOI:10.1016/j.cpc.2012.02.017

A brief history

Acceptance and distribution (2015)

- GPU support accepted by Vienna
- Integrated development environments
- **Established correctness**
- **To be included in standard VASP releases**

Establishing correctness

We've taken a three-pronged approach to validation:

1. Internal testing against ~ 50 cases collected from collaborators
   - Focus on actively ported algorithms and models
2. Acceptance testing against ~ 100 cases by Vienna
   - Cover wider variety of VASP usage patterns
3. Beta testing by 37 early access groups
   - Cover a wider variety of hardware and environments

Establishing correctness

We've taken a three-pronged approach to validation:

1. Internal testing against ~ 50 cases collected from collaborators
   - Focus on actively ported algorithms and models
2. Acceptance testing against ~ 100 cases by Vienna
   - Cover wider variety of VASP usage patterns
3. Beta testing by 37 early access groups
   - Cover a wider variety of hardware and environments

Establishing correctness

We've taken a three-pronged approach to validation:

1. Internal testing against ~ 50 cases collected from collaborators
   - Focus on actively ported algorithms and models
2. Acceptance testing against ~ 100 cases by Vienna
   - Cover wider variety of VASP usage patterns
3. Beta testing by 37 early access groups
   - Cover a wider variety of hardware and environments

Establishing correctness

We've taken a three-pronged approach to validation:

1. Internal testing against ~ 50 cases collected from collaborators
   - Focus on actively ported algorithms and models
2. Acceptance testing against ~ 100 cases by Vienna
   - Cover wider variety of VASP usage patterns
3. Beta testing by 37 early access groups
   - Cover a wider variety of hardware and environments

Beta testing

Three types of issues

- Use of unsupported features
- Merge with site-customized files (esp. main.F)
- Bugs in edge cases

Generally positive feedback

- “The short version is ‘it works”’
- “So far I found no problems, the code is fast and stable.”
- “Absolute time to solution is faster with GPUs.”

Release schedule

GPU support in official release

- Add CUDA paths and libraries to `makefile.include`
- `make gpu gpu_ncl`
- Executables are `bin/gpu` and `bin/gpu_ncl`

We expect the release by the end of the 2015.

Release schedule

GPU support in official release

- Add CUDA paths and libraries to `makefile.include`
- `make gpu gpu_ncl`
- Executables are `bin/gpu` and `bin/gpu_ncl`

We expect the release by the end of the 2015.

Feature support

Fully supported

- Davidson
- R-space projection
- RMM-DIIS
- Non-collinear
- Exact-exchange
- KPAR

Passively supported

- [sc]GW[0]
- Damped
- All (Algo)

Unsupported

- G-space projection
- NCORE > 1
- EFIELD_PEAD

Feature support

Fully supported

- Davidson
- R-space projection
- RMM-DIIS
- Non-collinear
- Exact-exchange
- KPAR

Passively supported

- [sc]GW[0]
- Damped
- All (Algo)

Unsupported

- G-space projection
- NCORE > 1
- EFIELD_PEAD

Feature support

Fully supported

- Davidson
- RMM-DIIS
- Exact-exchange
- R-space projection
- Non-collinear
- KPAR

Passively supported

- [sc]GW[0]
- Damped
- All (Algo)

Unsupported

- G-space projection
- NCORE > 1
- EFIELD_PEAD

Feature support

Fully supported

- Davidson
- R-space projection
- RMM-DIIS
- Non-collinear
- Exact-exchange
- KPAR

Passively supported

- [sc]GW[0]
- Damped
- All (Algo)

Unsupported

- G-space projection
- NCORE > 1
- EFIELD_PEAD

Traditional DFT

You should

- Run with MPS (multi-process service)
- Experiment with multiple CPU ranks per GPU

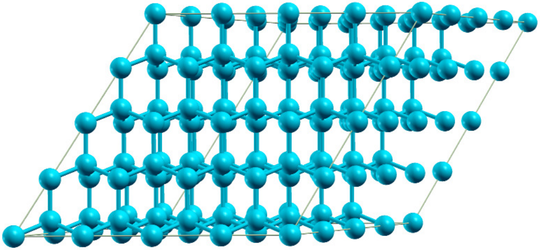
Works best

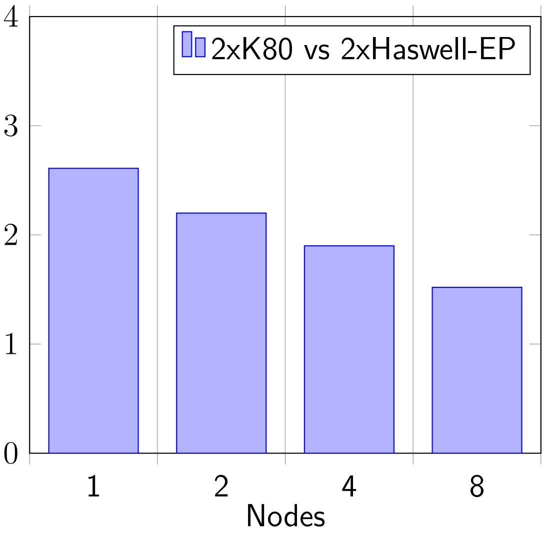
- Large numbers of bands
- Large numbers of plane-waves

You can expect 2-4x for large systems with CPU/GPU balance; better on GPU-heavy workstations.

Example: Si super-cell

- 512 Si atoms
- 1282 bands
- 864000 PWs
- Algo = Normal





Hybrid functionals (exact-exchange)

You should

- Use 1 or 2 CPUs rank per GPU
- Set NSIM = NBAND / (2*NCPU)

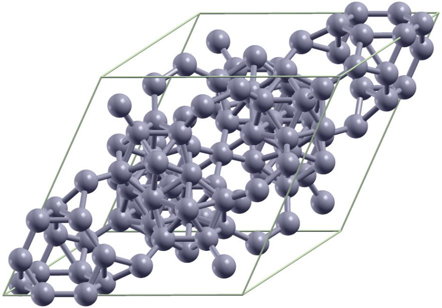
Works best

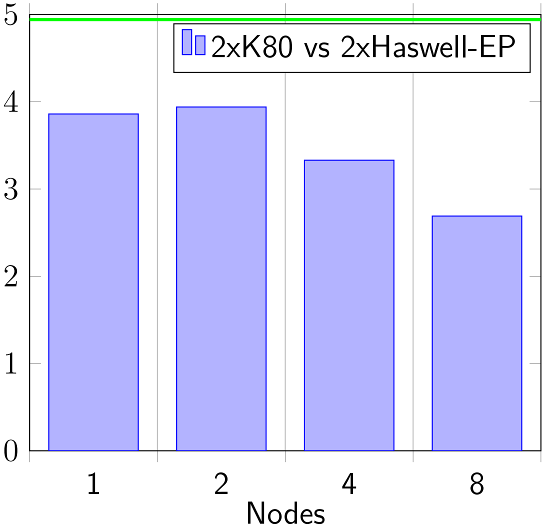
- Large numbers of plane-waves
- Small number of ionic types

You can expect 1.5-6x, highly dependent on system size; better on GPU-heavy workstations.

Example: β-rhombohedral boron

- 105 Boron atoms
- 216 bands
- 110592 PWs
- Algo = Normal





Road-map: Features

1. Gamma-point for very large unit cells
2. G-space projection for small to medium unit cells
3. Van der Waals density functional (vdF-DF)
4. Random phase approximation (RPA)
5. Active support for [sc]GW[0]
6. NCORE > 1 for highly parallel runs

Road-map: Performance

- Better performance for moderate sizes
  - Add blocking to all core kernels
  - Add batching to all library calls
- Better performance for large sizes
  - Update Magma support
  - Merge with threaded code base to reduce ranks per GPU
- Better performance for hybrid functionals
  - Parallelize outer loops
  - Pad projection sizes

Road-map: Performance

- Better performance for moderate sizes
  - Add blocking to all core kernels
  - Add batching to all library calls
- Better performance for large sizes
  - Update Magma support
  - Merge with threaded code base to reduce ranks per GPU
- Better performance for hybrid functionals
  - Parallelize outer loops
  - Pad projection sizes

Road-map: Performance

- Better performance for moderate sizes
  - Add blocking to all core kernels
  - Add batching to all library calls
- Better performance for large sizes
  - Update Magma support
  - Merge with threaded code base to reduce ranks per GPU
- Better performance for hybrid functionals
  - Parallelize outer loops
  - Pad projection sizes

Road-map: Performance

- Better performance for moderate sizes
  - Add blocking to all core kernels
  - Add batching to all library calls
- Better performance for large sizes
  - Update Magma support
  - Merge with threaded code base to reduce ranks per GPU
- Better performance for hybrid functionals
  - Parallelize outer loops
  - Pad projection sizes

Summary

GPU VASP will give you the right answer

- Extensive testing in Beta and for Vienna's acceptance

GPU VASP will give 2-4x performance on moderate to large systems

- The bigger the better

We are continuing to add feature support and improve performance

- Gamma-point is next on the list

When you get GPU support in your next VASP release, try it.

Summary

GPU VASP will give you the right answer

- Extensive testing in Beta and for Vienna's acceptance

GPU VASP will give 2-4x performance on moderate to large systems

- The bigger the better

We are continuing to add feature support and improve performance

- Gamma-point is next on the list

When you get GPU support in your next VASP release, try it.

Summary

GPU VASP will give you the right answer

- Extensive testing in Beta and for Vienna's acceptance

GPU VASP will give 2-4x performance on moderate to large systems

- The bigger the better

We are continuing to add feature support and improve performance

- Gamma-point is next on the list

When you get GPU support in your next VASP release, try it.

Summary

GPU VASP will give you the right answer

- Extensive testing in Beta and for Vienna’s acceptance

GPU VASP will give 2-4x performance on moderate to large systems

- The bigger the better

We are continuing to add feature support and improve performance

- Gamma-point is next on the list

When you get GPU support in your next VASP release, try it.

Summary

GPU VASP will give you the right answer

- Extensive testing in Beta and for Vienna's acceptance

GPU VASP will give 2-4x performance on moderate to large systems

- The bigger the better

We are continuing to add feature support and improve performance

- Gamma-point is next on the list

When you get GPU support in your next VASP release, try it.

More performance