Introducing k-Point Parallelism Into VASP

Asimina Maniopoulou
Numerical Algorithms Group Ltd,
HECToR CSE

HECToR

RESEARCH COUNCILS UK

Plan for the talk

- VASP
- Implementation of k-point parallelization
- Benchmarks -performance enhancement
- Scientific applications -paper

VASP

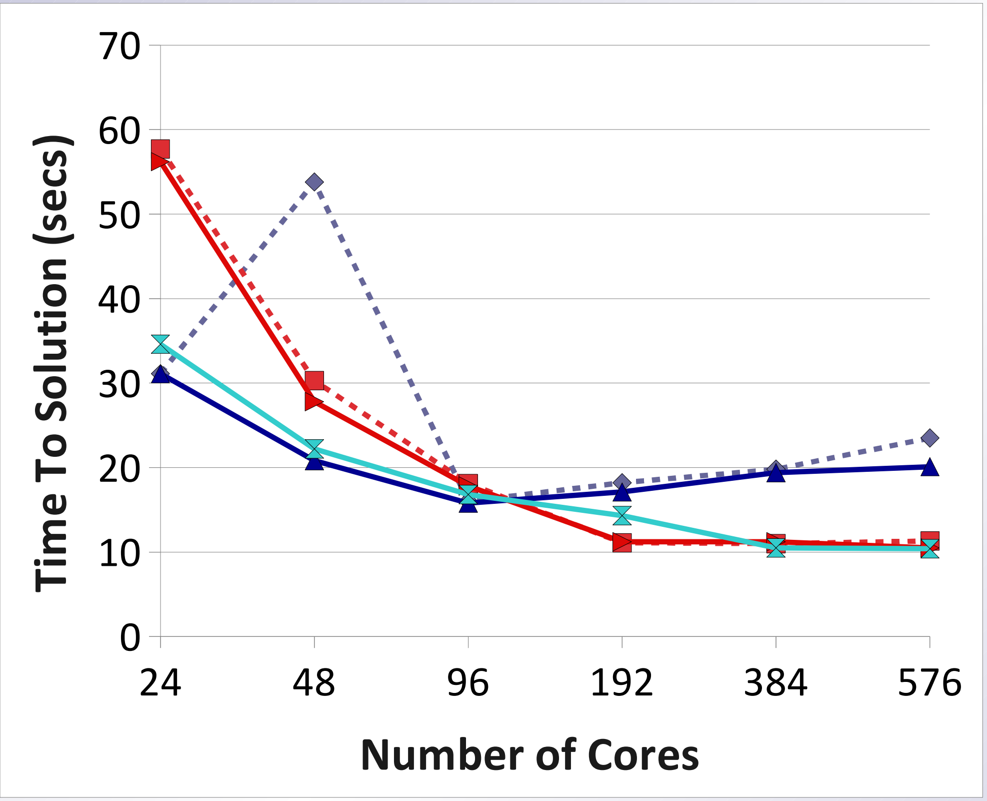
- Performs *ab initio* quantum mechanical molecular dynamics (MD) using pseudopotentials and a plane wave basis set
- Current release 5.2
- Implementation of k-point parallelism on VASP 5.2.2 -Funded by the dCSE mechanism within the HECToR Project
- One of the main new features in 5.2.2 is the ability perform calculations using hybrid functionals

Parallelism in VASP

- Parallelism in VASP is over plane waves
  - Which involves a FFT
- And bands
  - Which involves diagonalisation
- Controlled by NPAR
  - Set in INCAR
  - Diag is over NPAR cores
  - FFT is over P/NPAR cores

Parallelism in VASP

- Unfortunately neither FFTs or diags scale very well on parallel machines
  - More processors does not mean much faster



Diag example

- Thanks to Andy Sunderland of STFC Daresbury Laboratory for the numbers
- N = 7194
  - Bigger than VASP would use

Our k-point implementation

▶ The key hot spot of DFT calculation: triple loop for traversing spins, k-points and energy bands

- Existing parallelisation over bands
- k points are associated with the boundary conditions
- Multiple ones ultimately due to the use of periodic boundary conditions

- So we can exploit *k-point parallelism*
- Many other *ab initio* codes do this – e.g. CASTEP, CRYSTAL

Our -k point implementation

New parallelization: The number of k-points (outter loop) is distributed among a group of processes in a block cyclic distribution

- □ New communicators set-up
- □ New arrays set –up to allow efficient use of distributed memory
- □ Calculation of the k-points number before the communicators set-up
- □ Innermost loop (over bands) still performed in parallel

Our k point implementation

- But note not a universal panacea
  - Not all operations are parallel over k points
  - Not all operations involve k points
  - Double loops over k-points in HF calculations
  - Some extra communication
  - Large systems require few k points

Our k point implementation

- KPAR: a new parameter which should be specified in INCAR
- KPAR specifies the number of k point groups

  Thus if you have 10 k points

  - KPAR=2 results in two sets of processors, each with 5 k points
  - KPAR=5 has 5 groups with 2 k points

- Limitation : KPAR must divide **both** the number of processors **and** the number of k points exactly

Test Cases

Non Hartee-Fock exchange calculations:

Tested SCF cycles, Force calculations, Geometry Optimizations

Metals:

Case 1:H defect in 32 atoms of Pd. 10 k points

Case 2:As test case 1 but without using symmetry. Results in 108 k points

Insulators:

Case 3:Monoclinic Na_3AlH_6. 10 k points

Case 4:α-PbO_2 (Scrutinyite). 126 k points

Test Cases

Non Hartee-Fock exchange calculations:

Tested restart from WAVECAR file -

Different number of k-points

Case 5:Cu, 36 ions, 8 k points

Tested optical properties calculations

Case 6: CdTe, 8 ions, 20 k points

Tested phonon calculations

Case 7: CdTe, 8 ions, 216 k points

Test Cases

Hartee-Fock exchange calculations:

Tested SCF cycles, optical properties,

dos calculations, differerent values for NKRED, ENCUTFOCK

Case 8:Pb 24 k points (optical)

Case 9:CdTe, 8 ions, 4 kpoints

Case 10:CdTe, 8 ions, 20 kpoints points

Exploiting k point parallelism

Test 3 (10k)	Cores	Time (secs)	Speed Up
Original	72	386	1
KPAR=2	144	210	1.84
KPAR=5	360	119	3.24

Test 4 (126k)	Cores	Time secs	Speed Up
Original	72	1717	1
KPAR=2	144	912	1.88
KPAR=3	216	695	2.47

Exploiting k point parallelism

Smallest Speed-Up noticed for KPAR=2

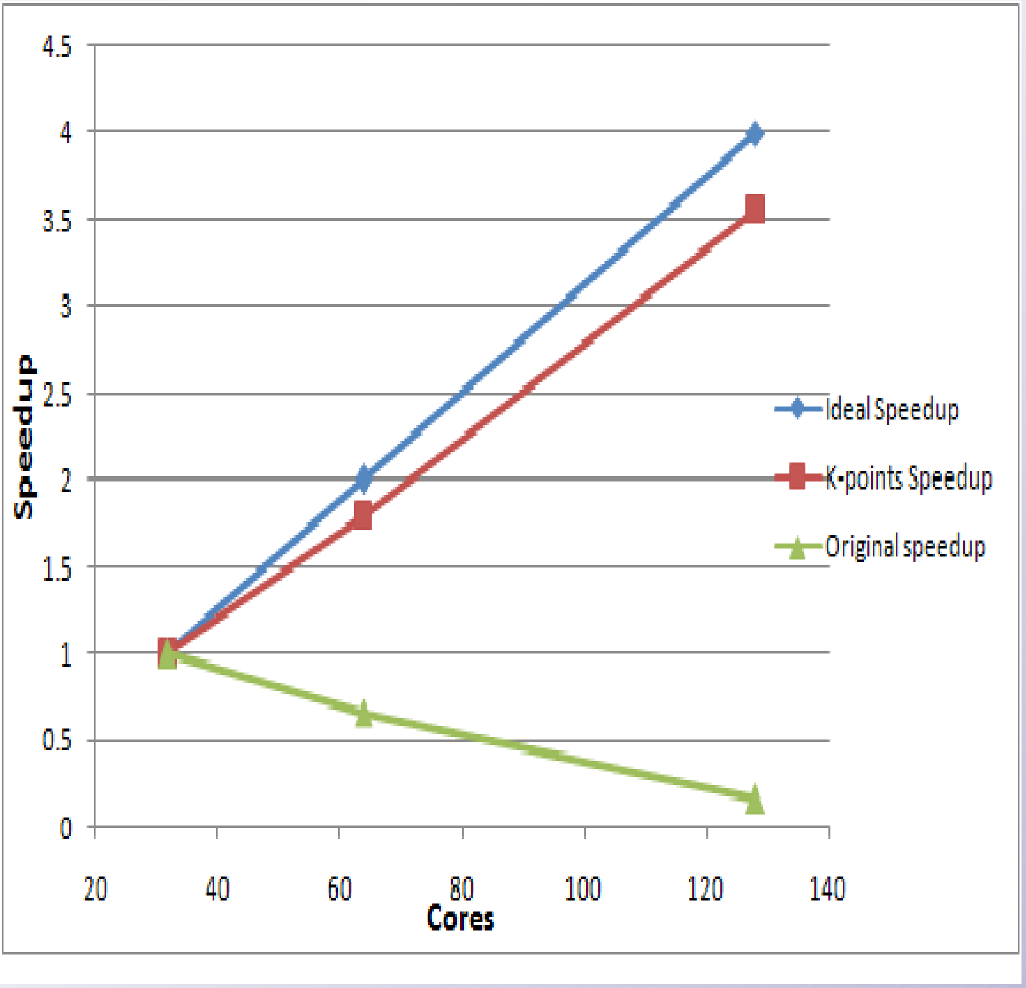
Test 10 (20k)HF	Cores	Time secs	Speed Up
Original	32	192	1
KPAR=2	64	136	1.41

Exploiting k point Parallelism

- phonon, 20 k-points

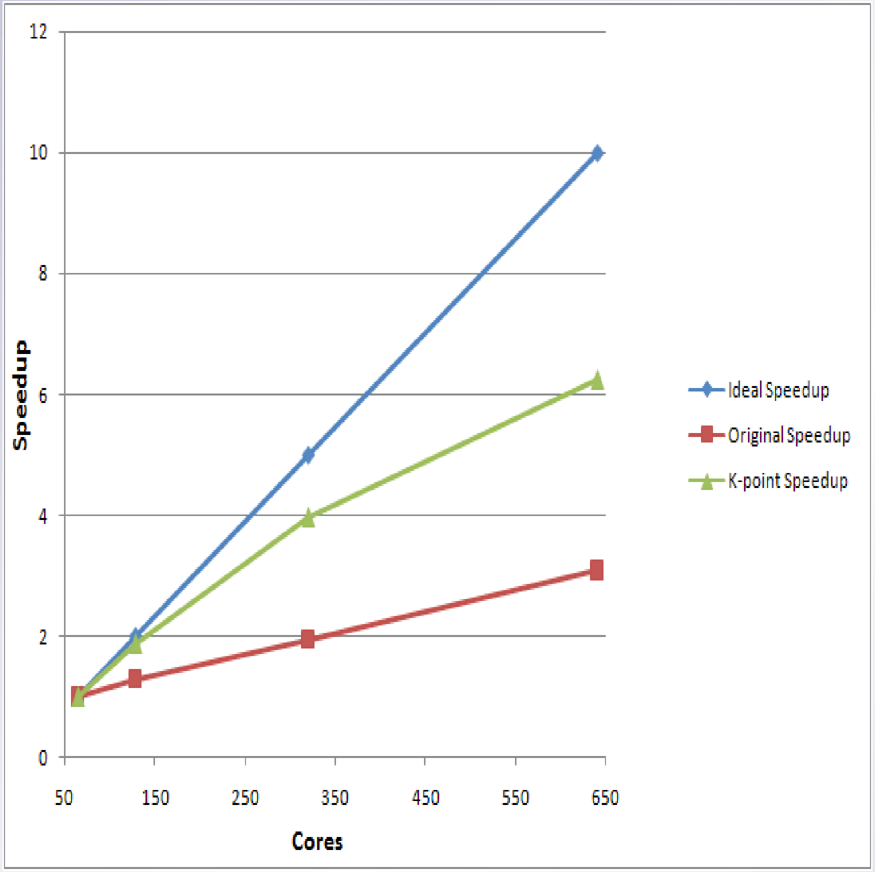
only FFT in parallel

	cores	time	speed up
v 5.2.2	64	603.29	1
KPAR = 2	64	221.94	2.718
v 5.2.2	128	2447.3	1
KPAR =4	128	112.41	22.04
v 5.2.2	160	4651.7	1
KPAR =5	160	64.96	71.61

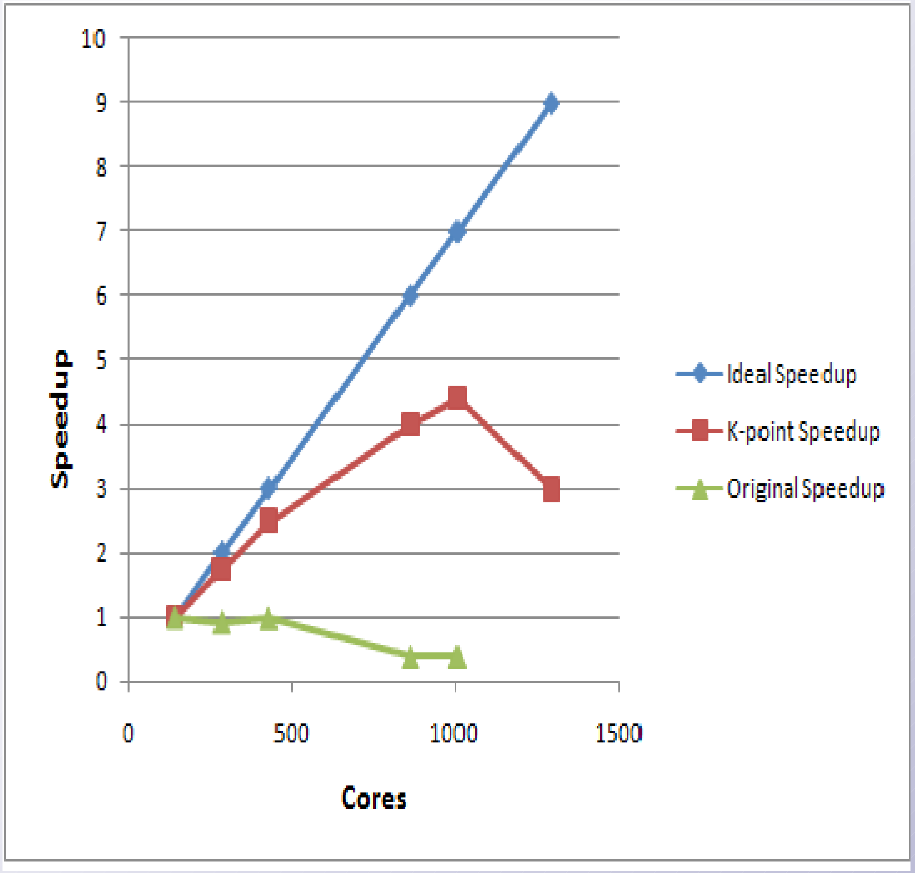


Exploiting k point Parallelism

- 10 k-points

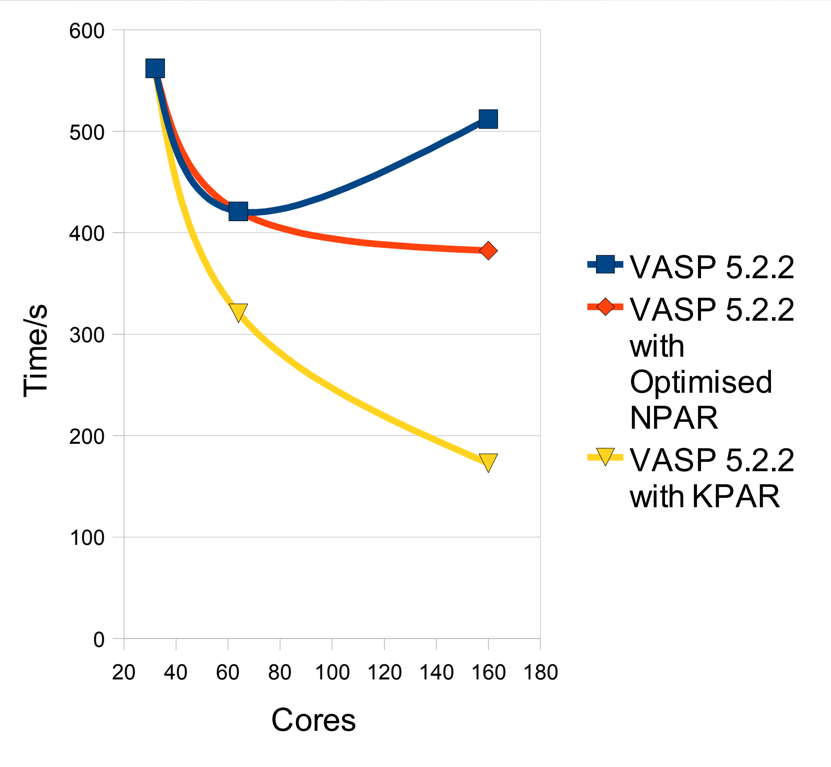


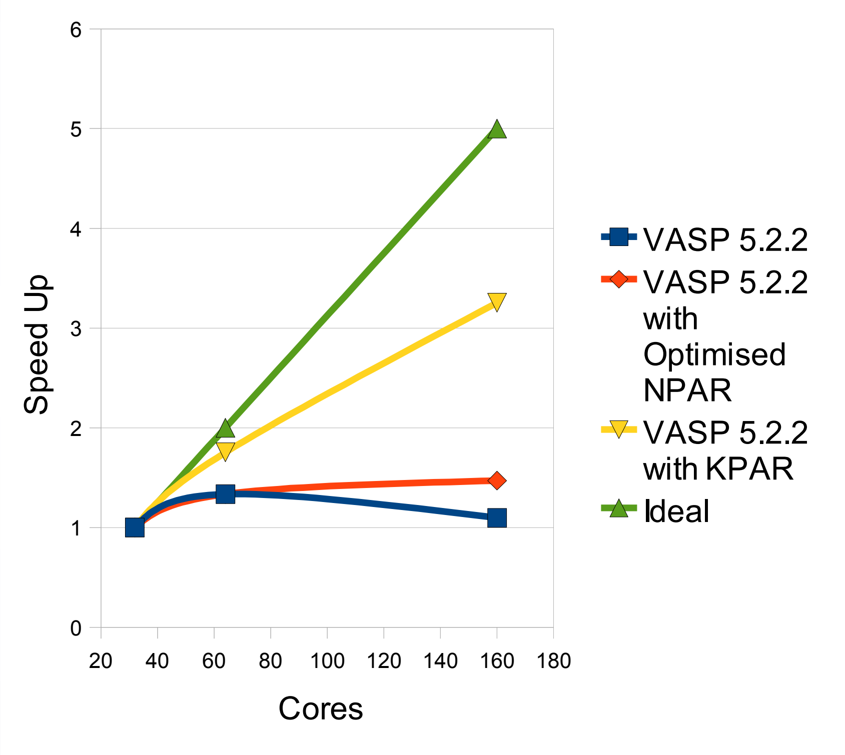
- 126 k-points



Exploiting k point Parallelism

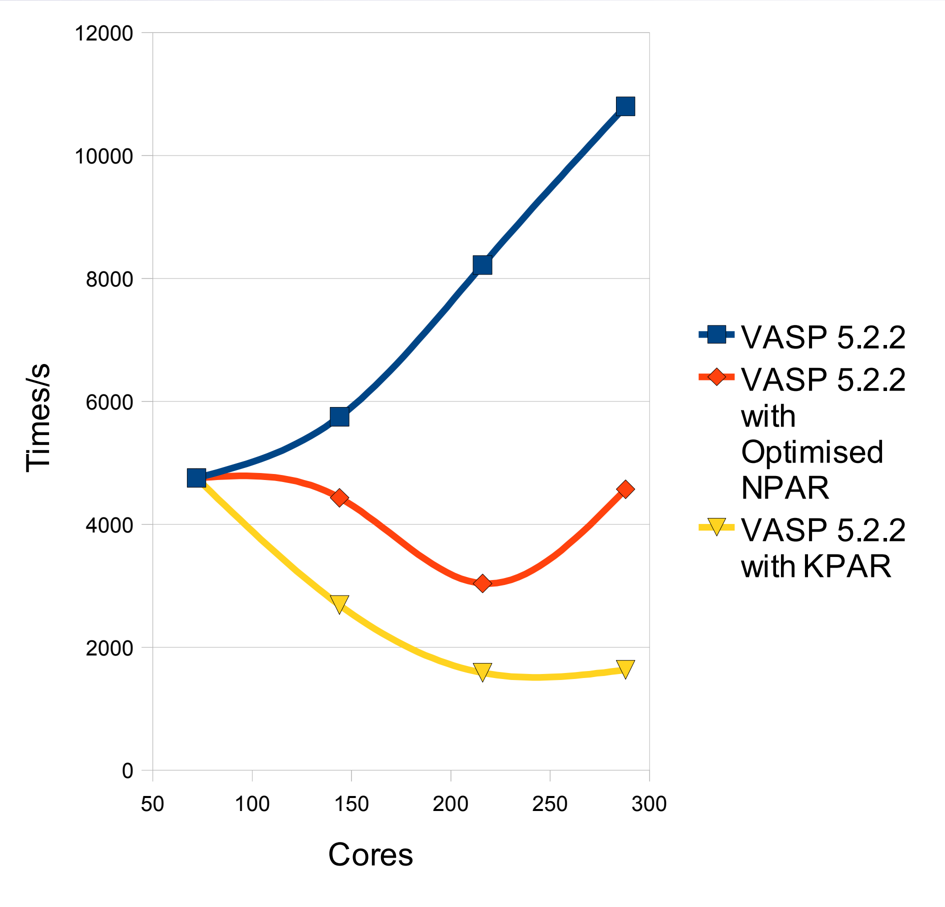
Test 1 – 10 k points

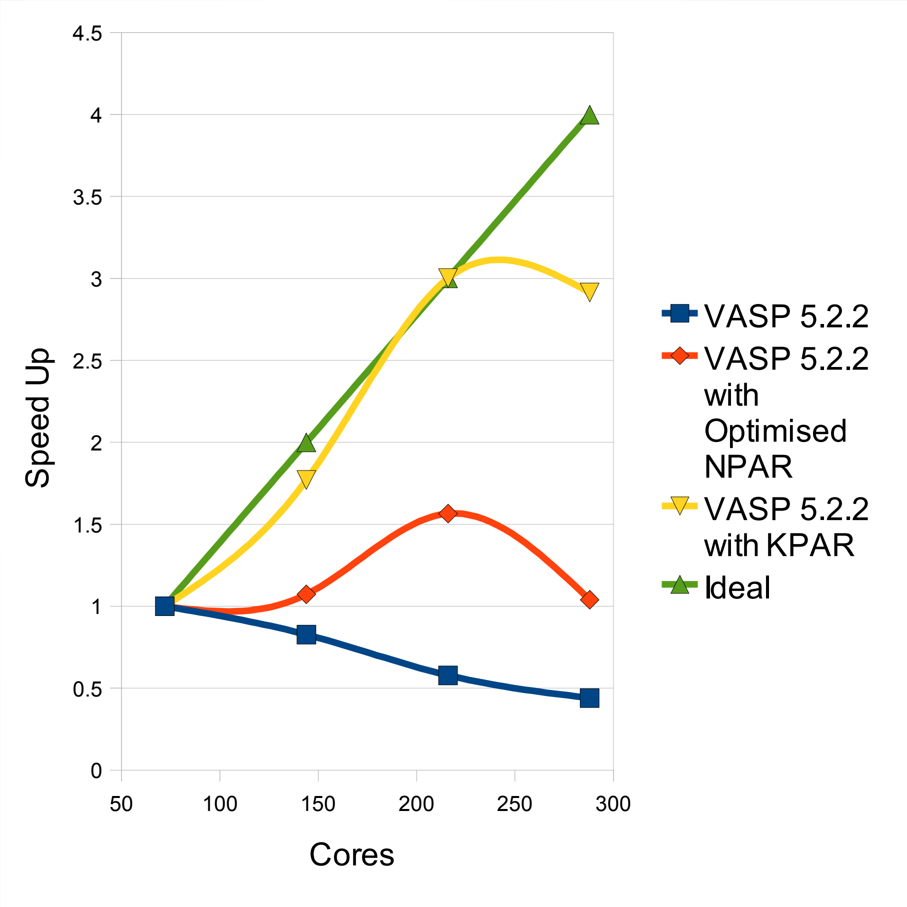




Exploiting k point Parallelism

Test 2 – 108 k points





Scientific applications I

Solution energies of tetravalent dopants in metallic VO_2 -Dr Ricardo Grau-Crespo

- Regulation of the transition temperature of VO_2 from the semiconductor to metallic phase via dopants
- 18 (4x4x4) mesh and 75 (8x8x8) k-point mesh for pure VO_2 (to decide on the dopants' mesh)
- 21 (4x4x4) k-points mesh for three dopants

- Indicatively, 13 times speed-up (v 15 the ideal speed-up) with the 8x8x8n mesh.

Scientific applications I I

Dielectric function of Epitaxially Strained Indium Oxide -Dr Aron Walsh

- Use in optoelectronic devices
- Optical properties convergence in respect with k-points:
  - slow
  - essential - valence to conduction band separations vary greatly across k-space

Scientific applications I I

- Results with max number of k-points within 12 hour queue on HECToR
- Results with k-parallelized code on HECToR

High frequency dielectric constant	*k*-point grid (Gamma centred)	Irreducible *k*-point grids
8.576 (), 8.729 ()	1×1×1	1
5.927 (), 5.996 ()	2×2×2	8
5.782 (), 5.755 ()	4×4×4	36
5.781 (), 5.743 ()	6×6×6	112

Outstanding issues

- What is the future of this?
  - It might get it in the official release
- Known limitations
  - KPAR values as noted above
  - vasp.xml output not available
  - Noncollinear magnetism cases

Acknowledgements

- Dr Ian Bush

Thank you ^{3}

- Ricardo Grau-Crespo, Aron Walsh, Erlend Davidson, Scot Woodley

Thank you