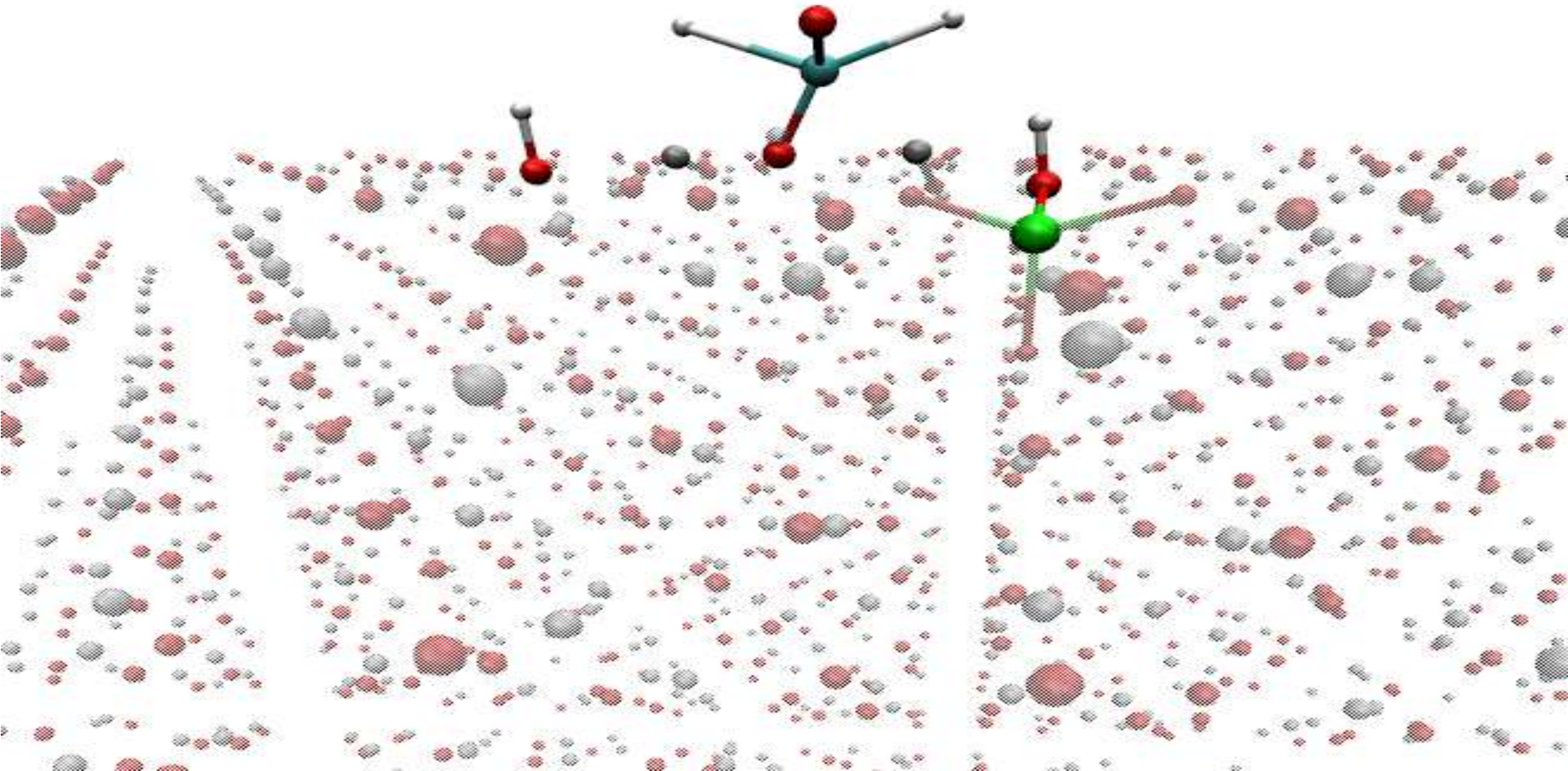
Efficient massively-parallel tools for the study of catalytic chemistry

Tom Keal

23 Sep 2009

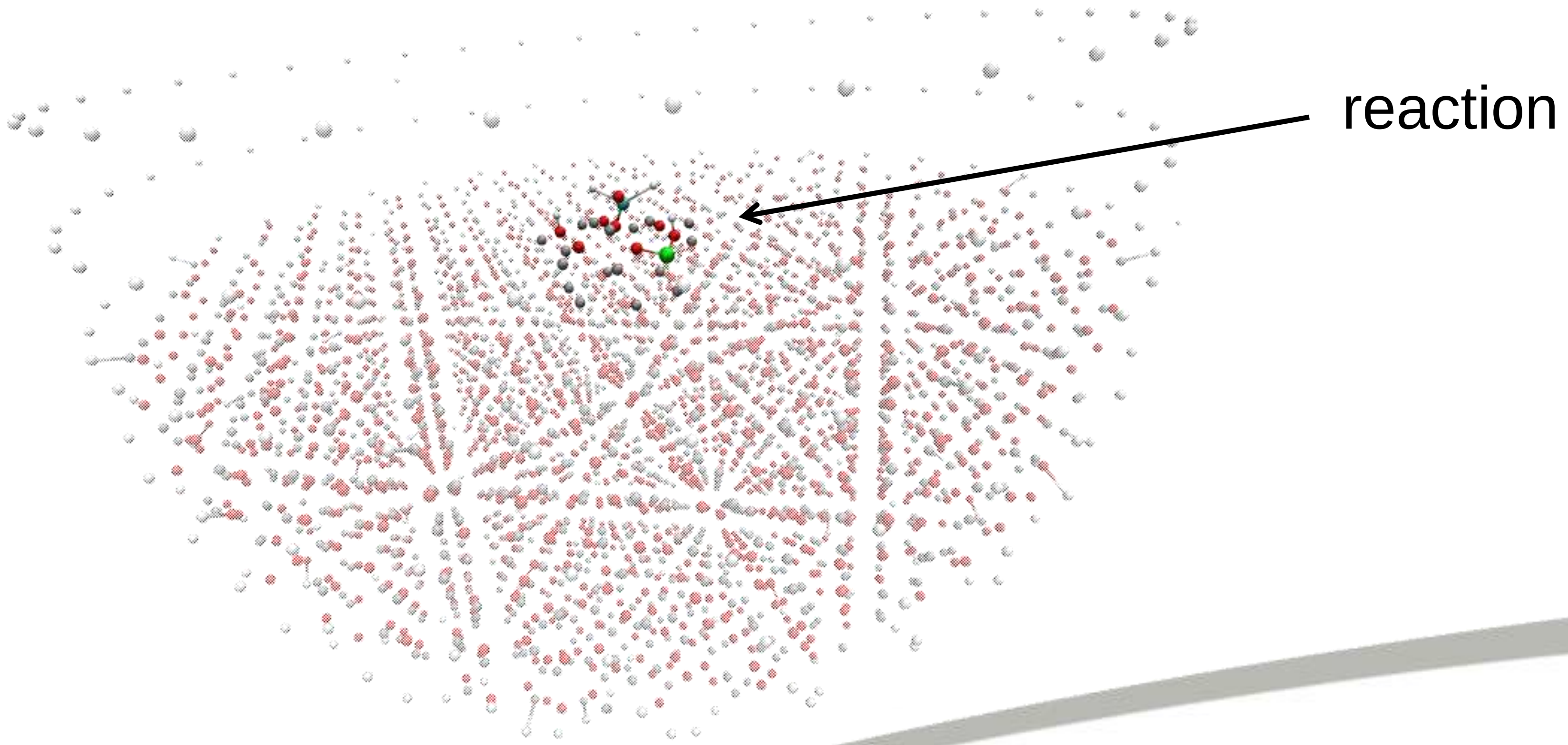
Catalytic chemistry example

- Methanol synthesis on Al-doped zinc oxide (UCL)



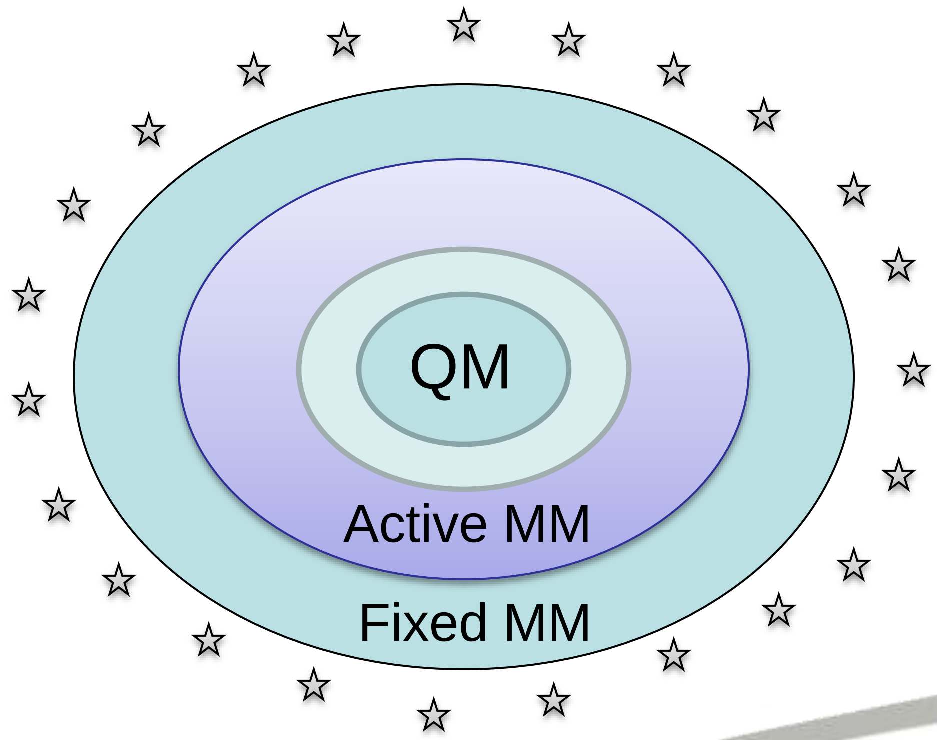
Cluster calculations

- Cut out a representative part of the surface



QM/MM cluster calculations

- Quantum mechanical description required for reaction
- Molecular mechanical description for environment



Point charges mimic the bulk environment

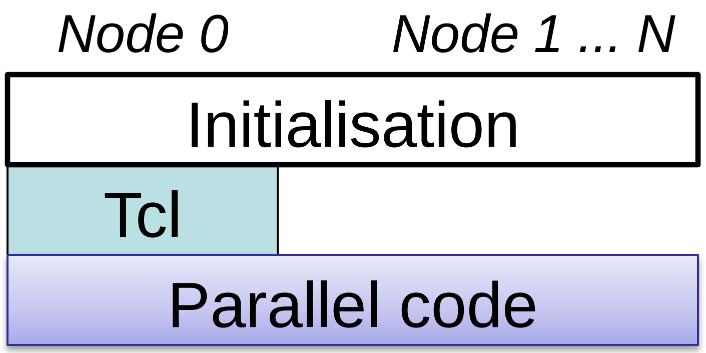
ChemShell

- Computational chemistry environment
  - www.chemshell.org
  - Tcl front end, C/Fortran behind the scenes
- Particularly useful for QM/MM calculations
  - Interfaces with external QM and MM programs to obtain E/g
  - ChemShell forms combined QM/MM energy/gradient
- Utilities for cutting clusters
- For our cluster calculations…
  - GAMESS-UK for the QM region
  - GULP for the MM region

Science & Technology Facilities Council

ChemShell in parallel

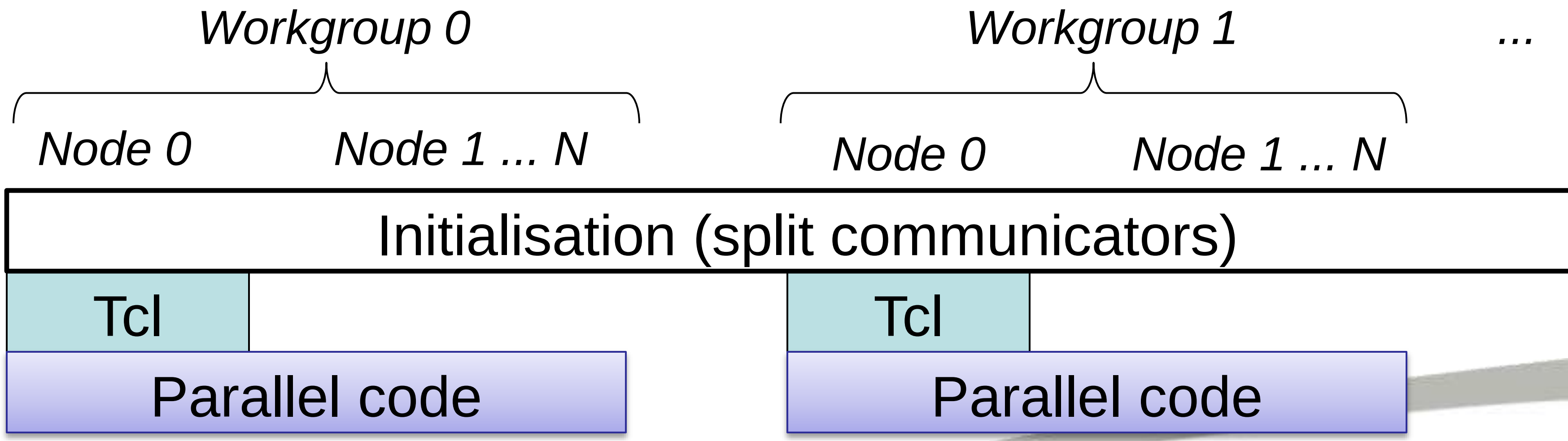
- ChemShell can run in parallel using MPI



- Takes advantage of parallel external programs
  - E.g. parallel energy/gradient evaluation in GAMESS-UK
- However, this approach does not scale well to 000's of processors

Task-farming parallelism

- Aim is to parallelise ChemShell algorithms as well
  - E.g. parallel Hessian evaluation, geometry optimisation, etc.
- Task-farming approach: divide up processors into workgroups working independently on tasks



Task-farming parallelism

- Workgroups are essentially independent
  - Separate stdout/stderr
  - Separate working directories to prevent file conflicts
  - All lower-level parallelism (e.g. GAMESS-UK calculations) occurs within a single workgroup. Therefore the workgroup communicator must be passed to GAMESS.
- All workgroups are controlled via a single Tcl input script
  - Tcl commands to allow workgroup-specific tasks
  - Workgroups can be explicitly synchronised
  - Local ChemShell objects can be made available globally

Science & Technology Facilities Council

Finite difference Hessians

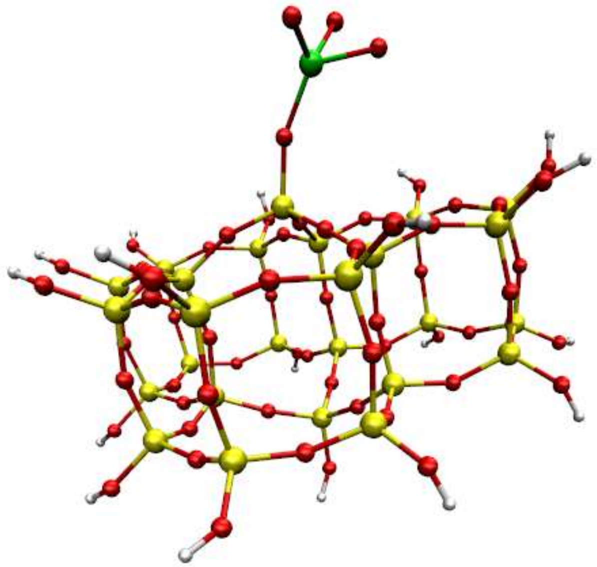
- Second derivative matrix of the energy, calculated numerically using first derivatives (gradients)
- ChemShell’s ‘force’ command split up into three stages to allow task-farmed execution:

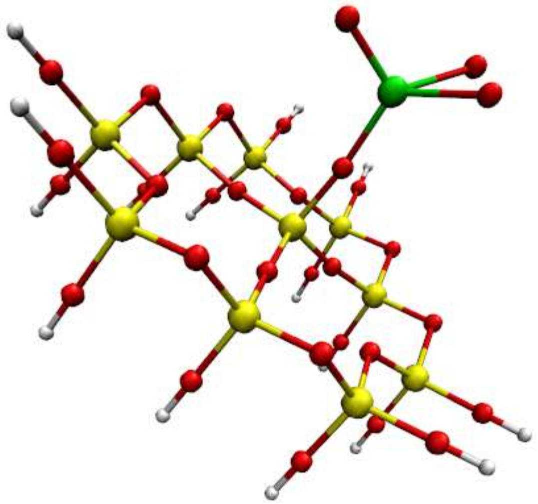
1. Precalculate the required gradients
    - Divided up by atom number (static load-balancing)
2. Make gradient objects available globally
3. Build the Hessian from the precalculated gradients

Science & Technology Facilities Council

Hessian benchmarks

- GAMESS-UK, DFT (B3LYP), lanl2 basis (pure QM)
- Two VO_3/silicate clusters:

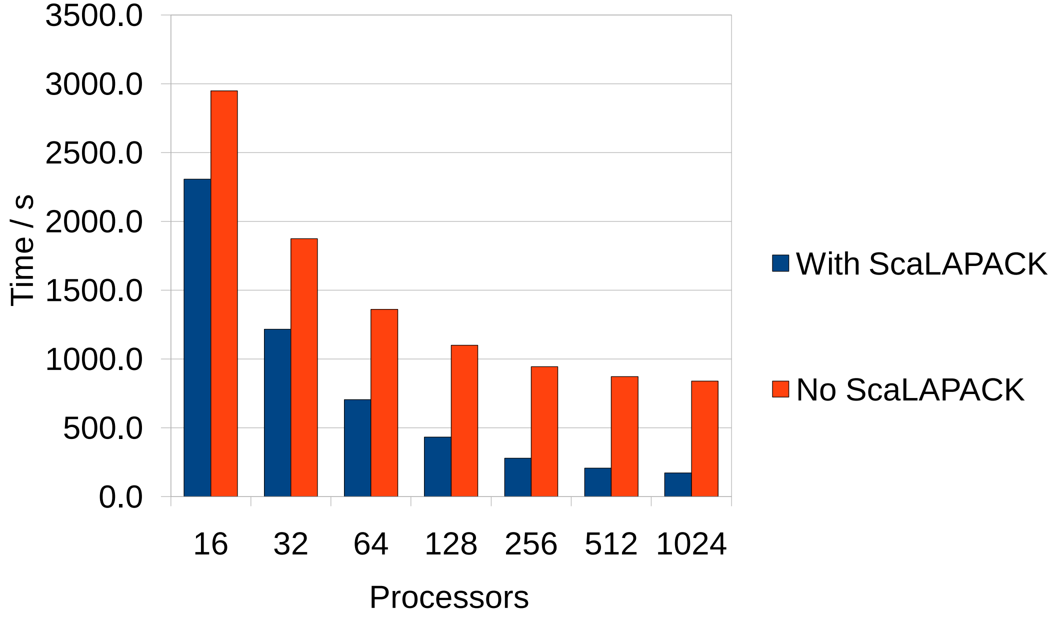

Cluster 1: 111 atoms


Cluster 2: 57 atoms

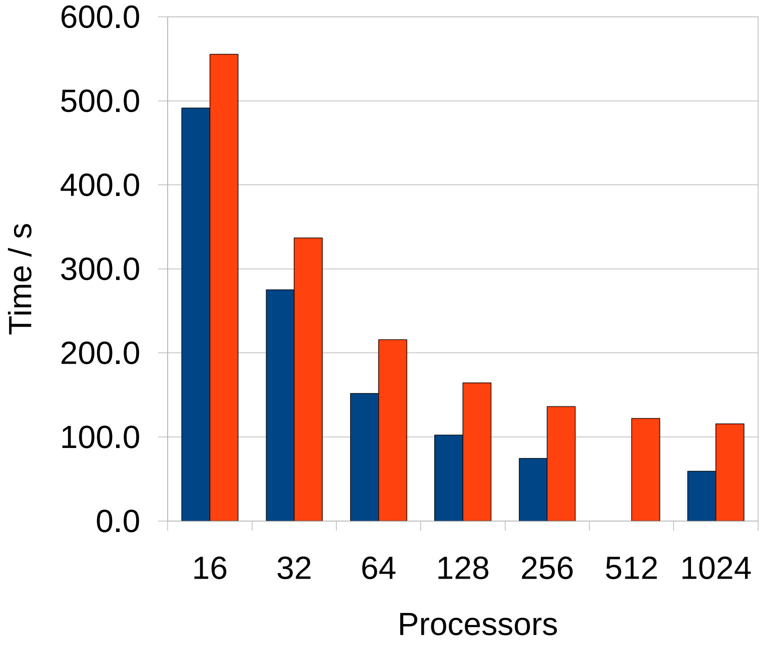
Single point calculations

- Problem: ScaLAPACK routines in GAMESS-UK do not support split communicators
  - compare with and without ScaLAPACK

Cluster 1



Cluster 2



Hessian benchmark results

- Single gradient timings suggest that 32 workgroups is optimal

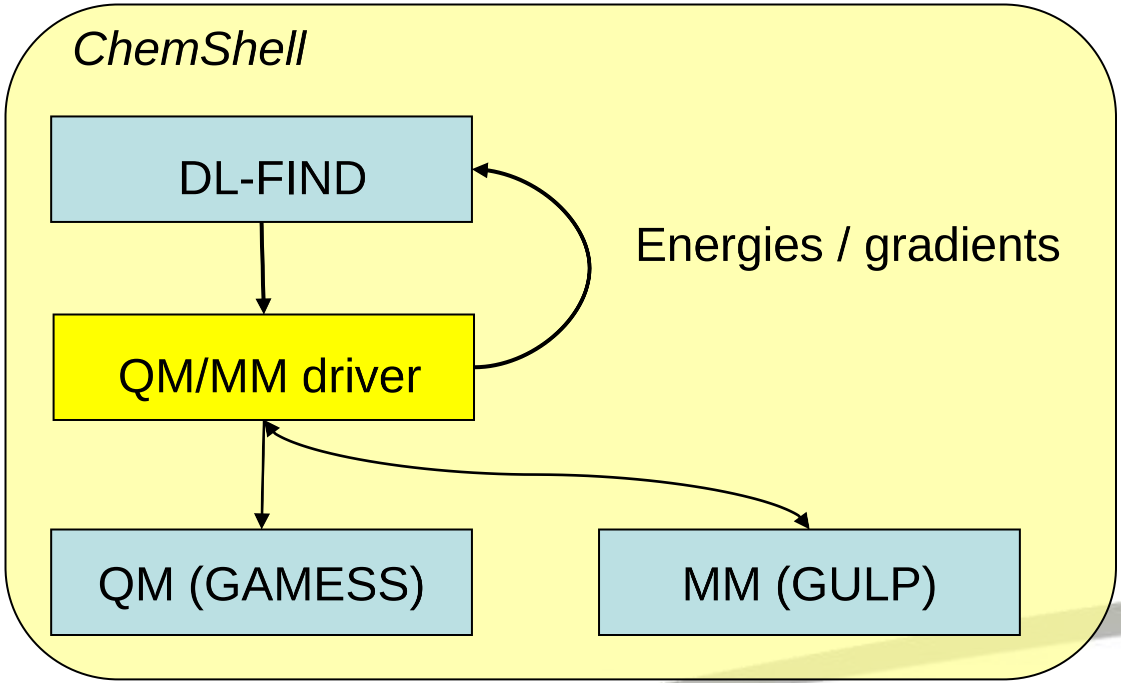
CLUSTER 1	*Finite difference type*	
Time / s	*Single point*	*Two point*
Original	28441.2	57842.1
32 workgroups	12335.4	23690.8
Task farming speed up factor		
vs Original	2.3	2.4

CLUSTER 2	*Finite difference type*	
Time / s	*Single point*	*Two point*
Original	5769.8	11430.5
Single workgroup	9150.9	18219.6
32 workgroups	1256.6	2454.6
Task farming speed up factor		
vs Single workgroup	7.3	7.4
vs Original	4.6	4.7

- Original = single workgroup with ScaLAPACK
- Single workgroup = no ScaLAPACK (like with like)

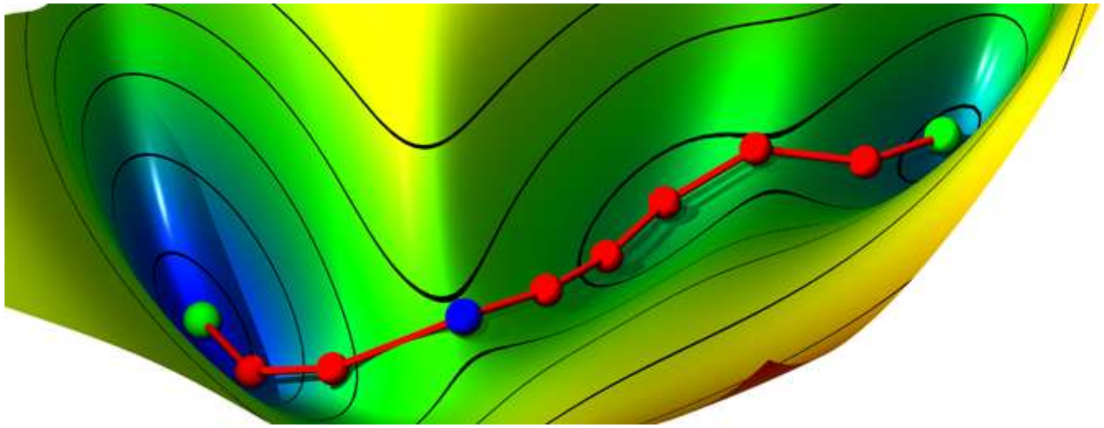
DL-FIND

- An open-source geometry optimisation library
- Interface to ChemShell for QM/MM optimisations



Nudged elastic band method

- Optimising reaction paths: finds the minimum energy path



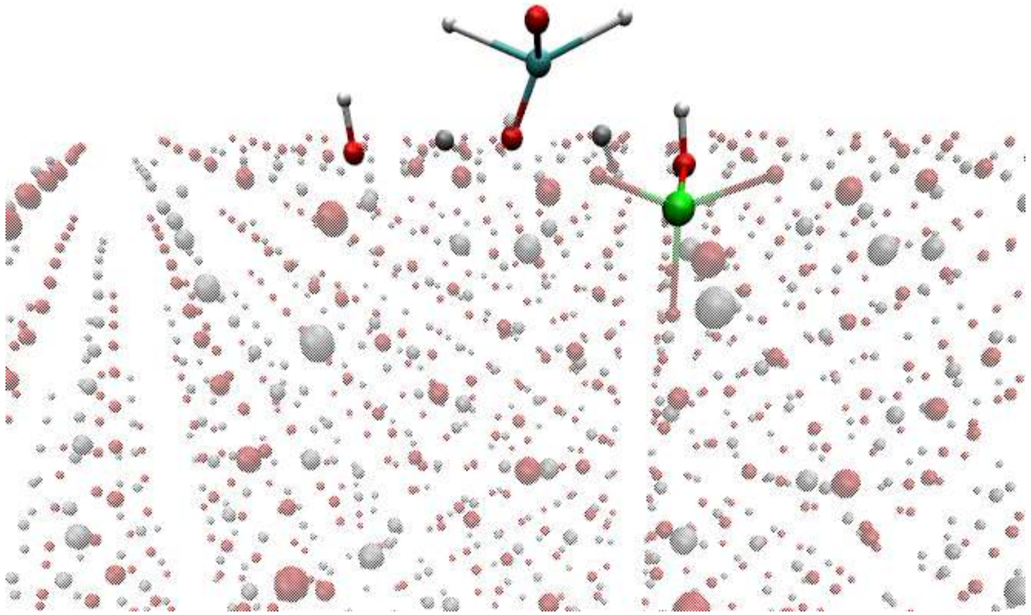
- Multiple images, connected by spring forces.
- Climbing image to find transition state
- Image e/g evaluations are independent, so can be parallelised

Parallel NEB

- ChemShell/DL-FIND parallel interface required
  - Pass the relevant MPI communicators, workgroup information
- Each workgroup runs DL-FIND
  - Tasks allocated according to workgroup ID
  - Energies/gradients shared between workgroups at the end of each cycle (allreduce)
- Some complications compared to serial case
  - Status, restarts
  - First cycle in serial for benefit of QM program (guess vectors)
  - Frozen images speed up serial calculations vs. parallel

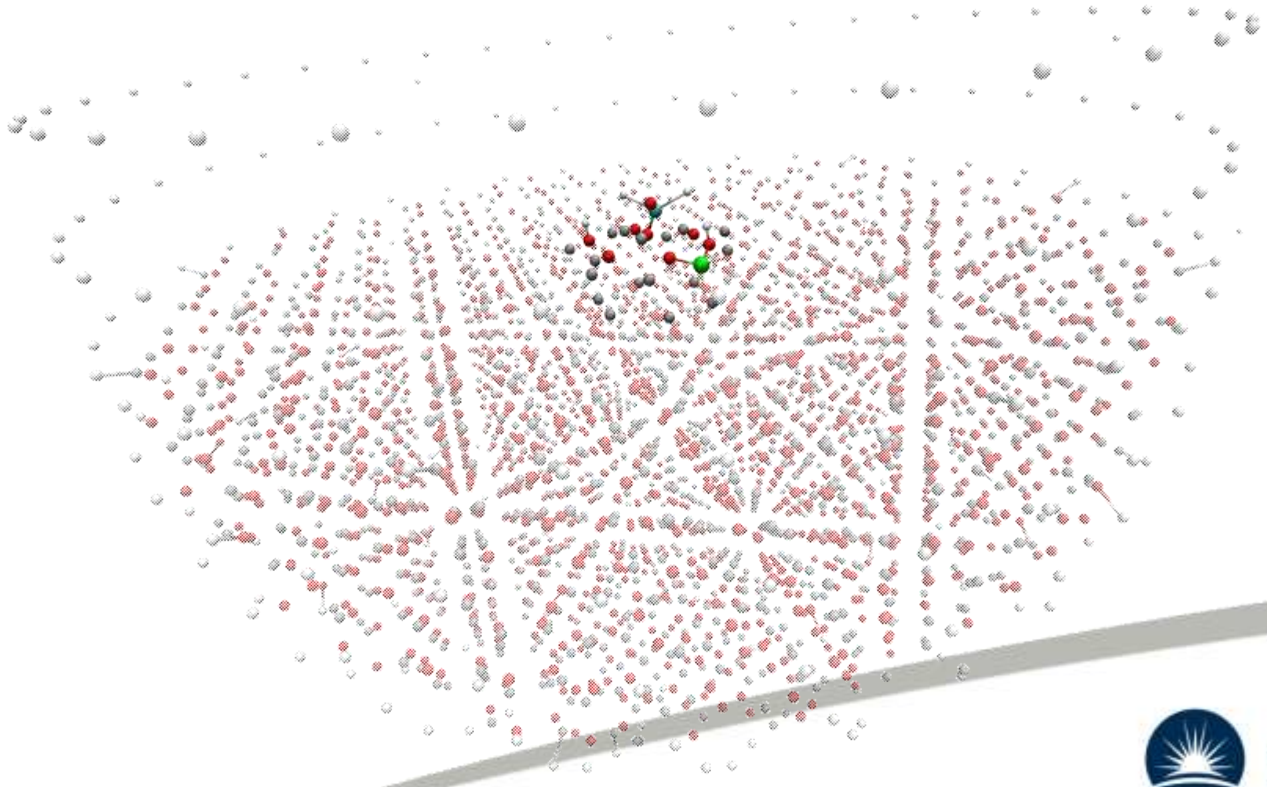
Parallel NEB test

- NEB to find barrier of hydrogen exchange for CO_2 on Al-doped ZnO surface:



Parallel NEB test

- QM/MM cluster calculation
  - MM: GULP must use MPI communicator from ChemShell



Science & Technology Facilities Council

Parallel NEB test details

- 3207 atoms
  - QM: 32 atoms, DFT(B97-1), double zeta basis (ECP for Zn)
  - MM includes shell model for polarisation
- 10 NEB images (including end points and climbing images)
- Speed up > 2 expected over standard 1024-processor run
  - Preliminary results (Time for 50 cycles of NEB):

1 workgroup of 1024 procs	10 hours
4 workgroups of 256 procs	3 hours

Science & Technology Facilities Council

Summary and outlook

- Task-farm parallel framework implemented in ChemShell
  - GAMESS-UK and GULP made ‘task farm aware’
- Finite difference Hessian code parallelised
- Parallel interface between ChemShell & DL-FIND
- Parallel NEB implemented

- Next milestone: interface ChemShell with DL-FIND’s parallel optimisation algorithms
  - Genetic algorithm, stochastic search

Science & Technology
Facilities Council

Acknowledgements

- Paul Sherwood
- Huub van Dam
- Gargi Dutta
- NAG for funding

Science & Technology Facilities Council