



Comparing Parallel Performance for a Geometric Multigrid Solver Using Hybrid Parallelism, MPI Shared Memory and ~~Multiple~~ GPUs **Single**

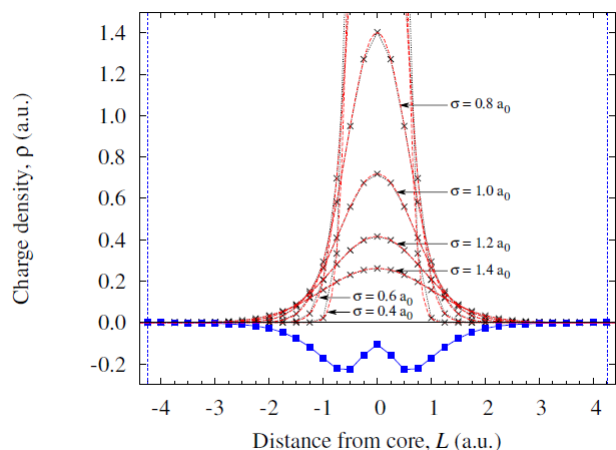
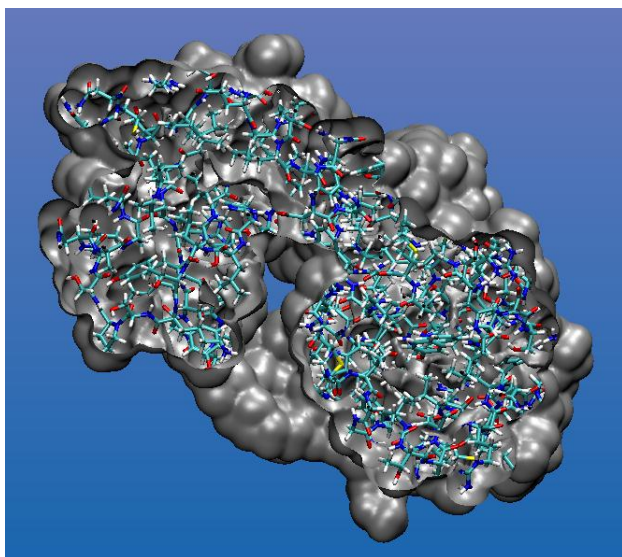
Lucian Anton, Mark Mawson, Vendel Szeremi
Scientific Computing Department,
STFC, Daresbury Lab, UK

Outline

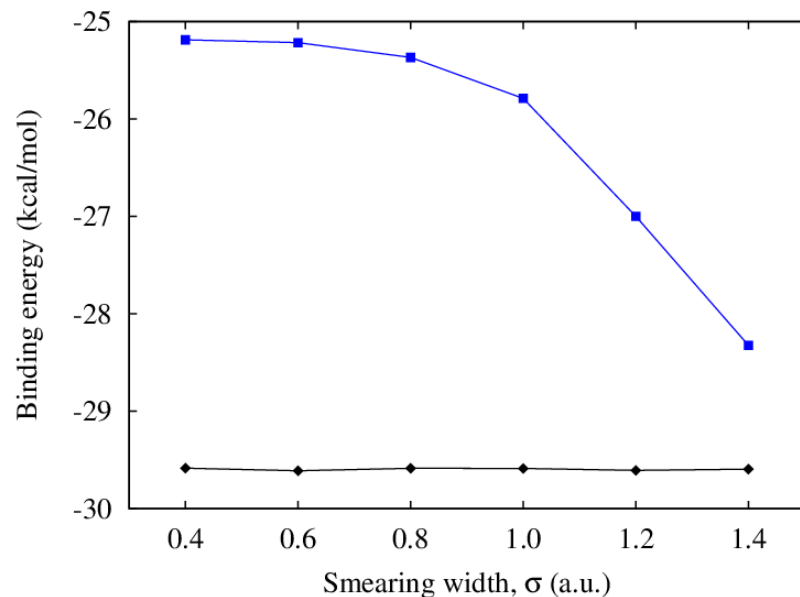
- Introduction
 - ◆ Solvation calculations in ONETEP
 - ◆ Description of the multigrid solver
- Algorithms for halo exchange
- Performance result on Cray XC30, Blue Gene Q and Fermi GPU
- Conclusions



T4 Lysozyme L99A/M102Q (2602 atoms) in implicit solvent



- First quantum chemistry study of an entire protein in implicit solvent
- Determination of optimum calculation parameters for Energy calculations (vacuum, solvent, binding)
 - Smeared core charges
 - Open boundary conditions
 - Higher finite difference order correction
 - NGWF radius
 - Dispersion



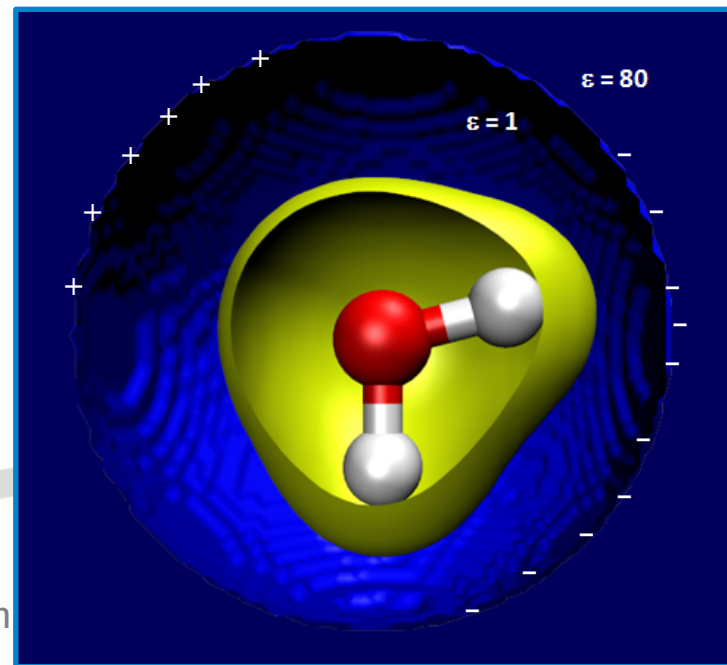
J. Dziedzic, S. J. Fox, T. Fox, C. S. Tautermann and C.-K. Skylaris, *Int. J. Quant. Chem.* **113** (2013) 771.

<http://www.hector.ac.uk/cse/distributedcse/reports/onetep/onetep.pdf>

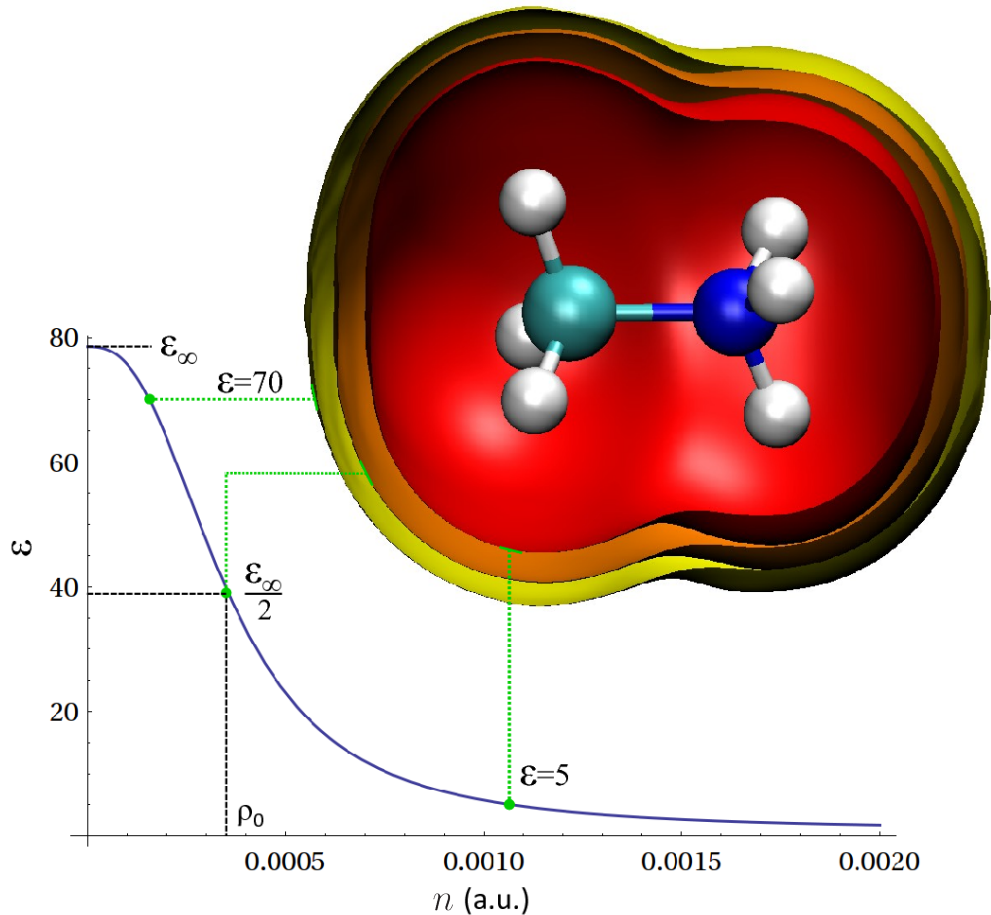
Implicit solvent approach

http://www.oerc.ox.ac.uk/sites/default/files/uploads/ProjectFiles/ASEArch/MultigridWorkshop/Dziedzic_onetep_solvation.pdf

- Treat only the solute explicitly and embed it within a suitably defined cavity, the inside of which is inaccessible to the solvent.
- Replace the solvent with an *unstructured dielectric continuum*, only retaining its **average** effect on the solute.



Density-dependent cavity of Fattebert and Gygi



$$\nabla \epsilon(\mathbf{r}) \nabla \phi(\mathbf{r}) = -4\pi \rho(\mathbf{r})$$

$$\epsilon(n(\mathbf{r})) = 1 + \frac{\epsilon_\infty - 1}{2} \left(1 + \frac{1 - (n(\mathbf{r})/\rho_0)^{2\beta}}{1 + (n(\mathbf{r})/\rho_0)^{2\beta}} \right)$$

Multigrid basics I

$$\nabla \cdot (\varepsilon(\mathbf{r}) \nabla \phi(\mathbf{r})) = -4\pi \rho_{\text{tot}}(\mathbf{r}),$$

➤ Multigrid methodology has 2 principles:

- ◆ Smoothing, fast at short wavelength
- ◆ Coarse grid

- A smooth error can be transferred to coarser grid without loss of information

$$\begin{array}{c}
 u_h^m \xrightarrow{\text{smooth}^{\nu_1}} \bar{u}_h^m \rightarrow \bar{d}_h^m = f_h - A_h \bar{u}_h^m \\
 \downarrow I_h^{2h} \\
 \bar{d}_{2h}^m \xrightarrow{\quad\quad\quad} A_{2h} \hat{v}_{2h}^m = \bar{d}_{2h}^m \\
 \uparrow I_{2h}^h \\
 \hat{v}_h^m \xrightarrow{\quad\quad\quad} \bar{u}_h^m + \hat{v}_h^m \xrightarrow{\text{smooth}^{\nu_2}} u_h^{m+1}
 \end{array}$$

operations $\sim O(N \log e)$

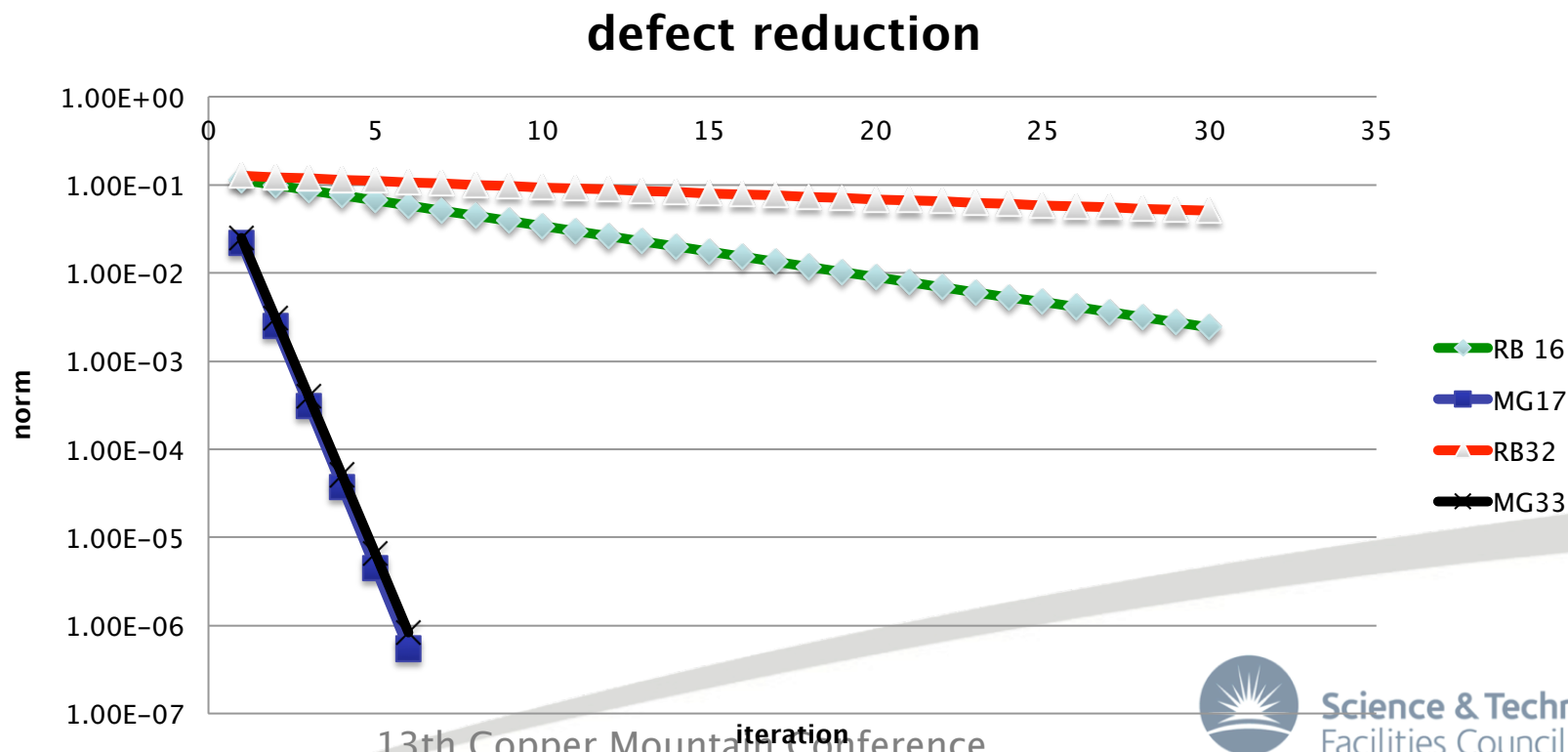
Full multigrid algorithm reaches discretisation accuracy



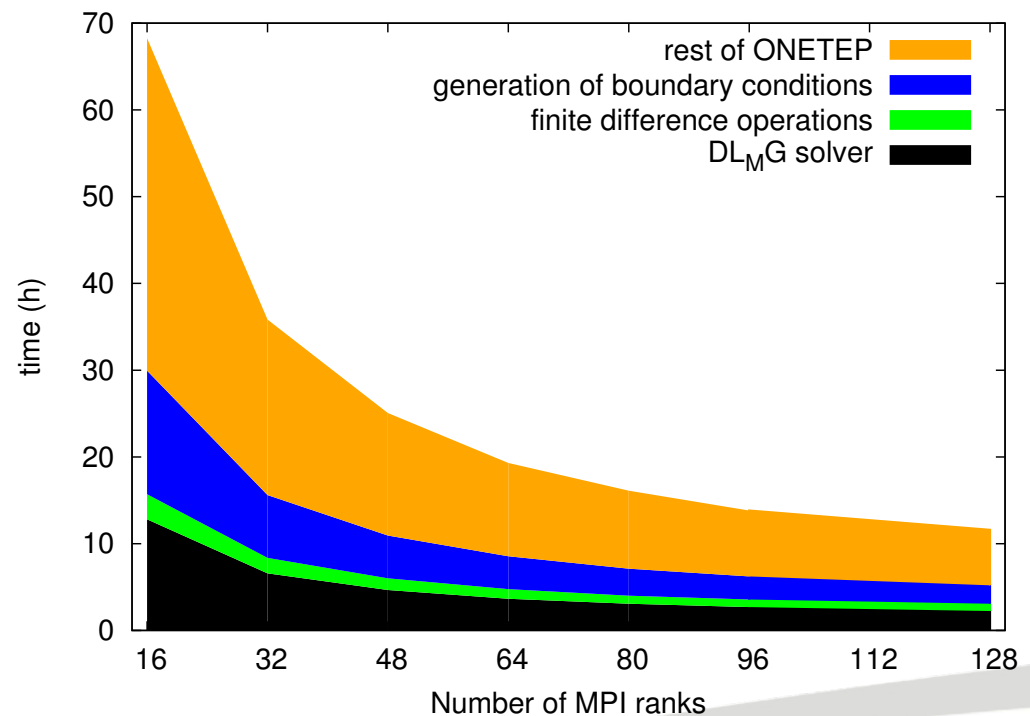
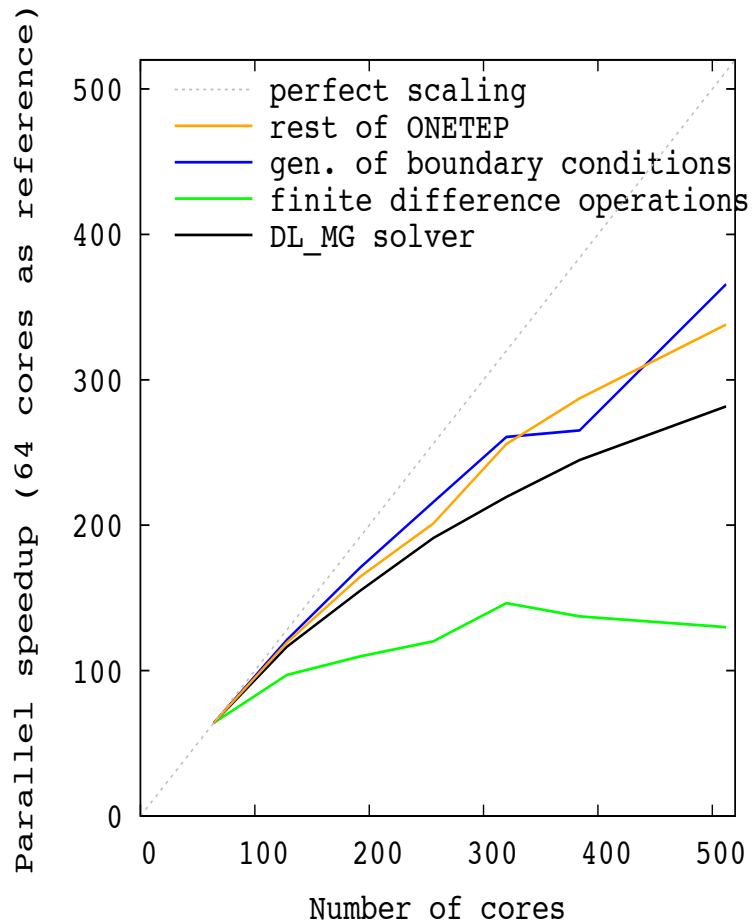
Science & Technology
Facilities Council

Multigrid basics II

- DL-MG solver
 - ◆ Red-black Gauss-Seidel smoother and solver
 - ◆ Half weigh restriction
 - ◆ Bilinear interpolation

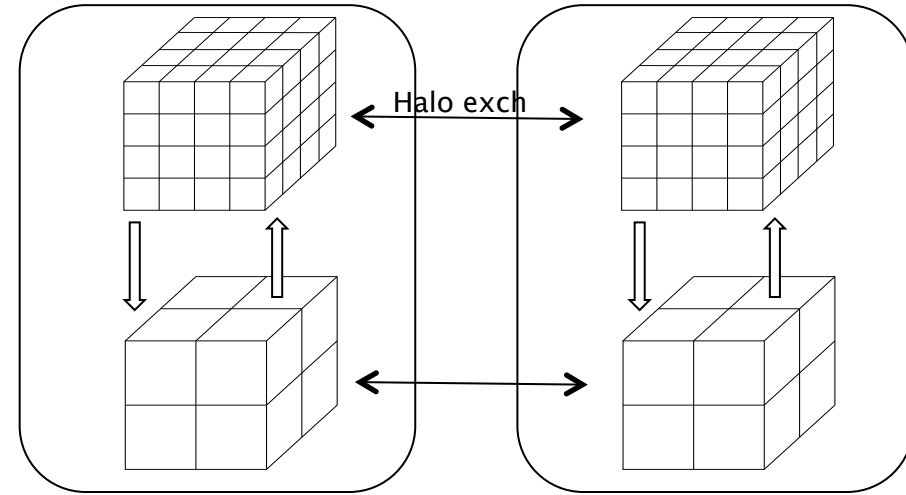
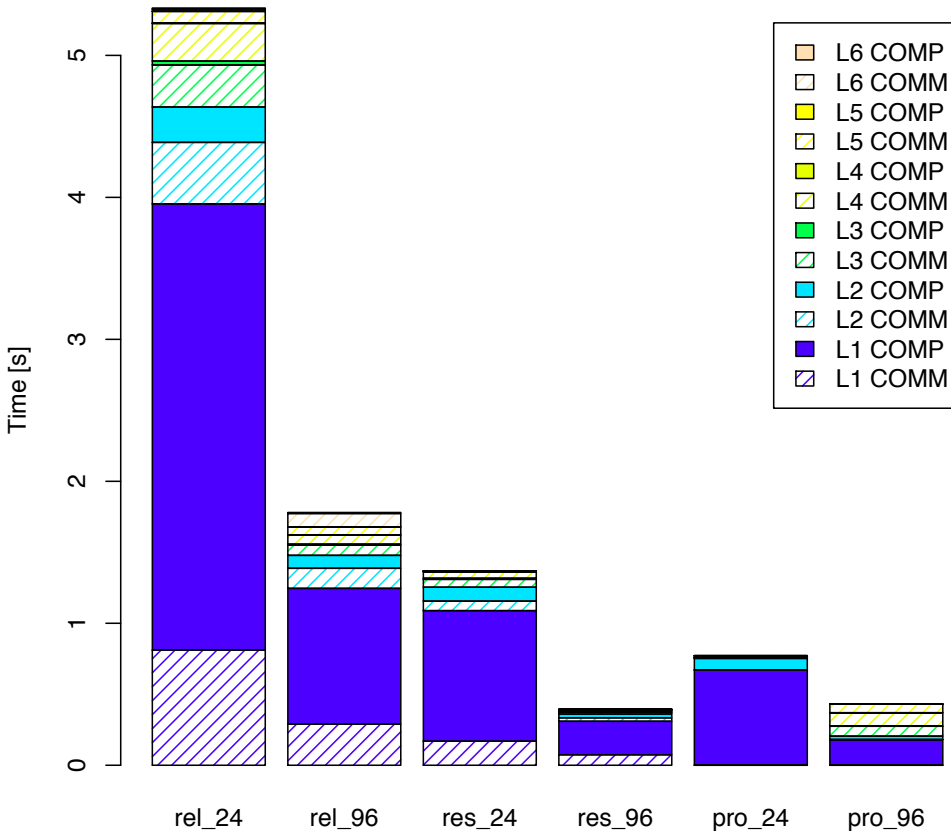


ONETEP Solvation calculation performance

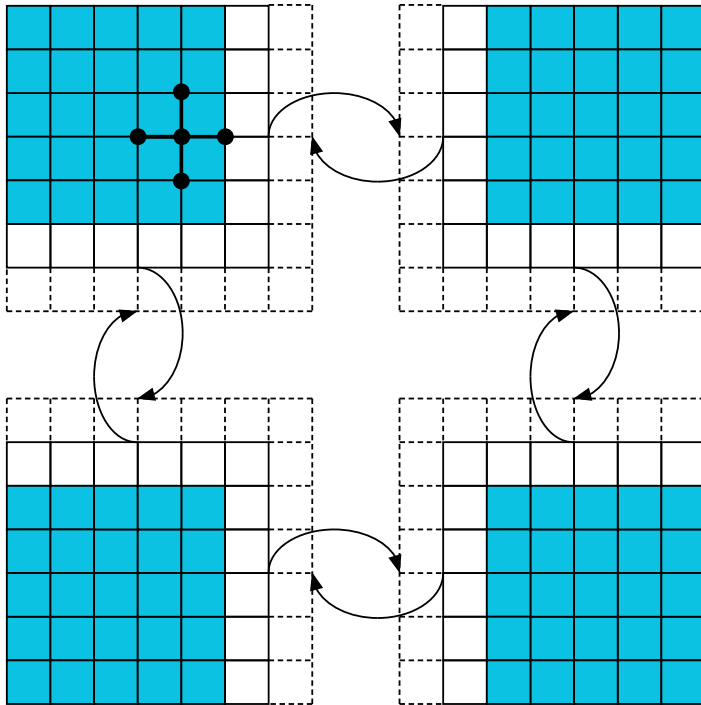


Baseline profile

Breakout of solver times, Archer/Intel, Baseline 1 Thread

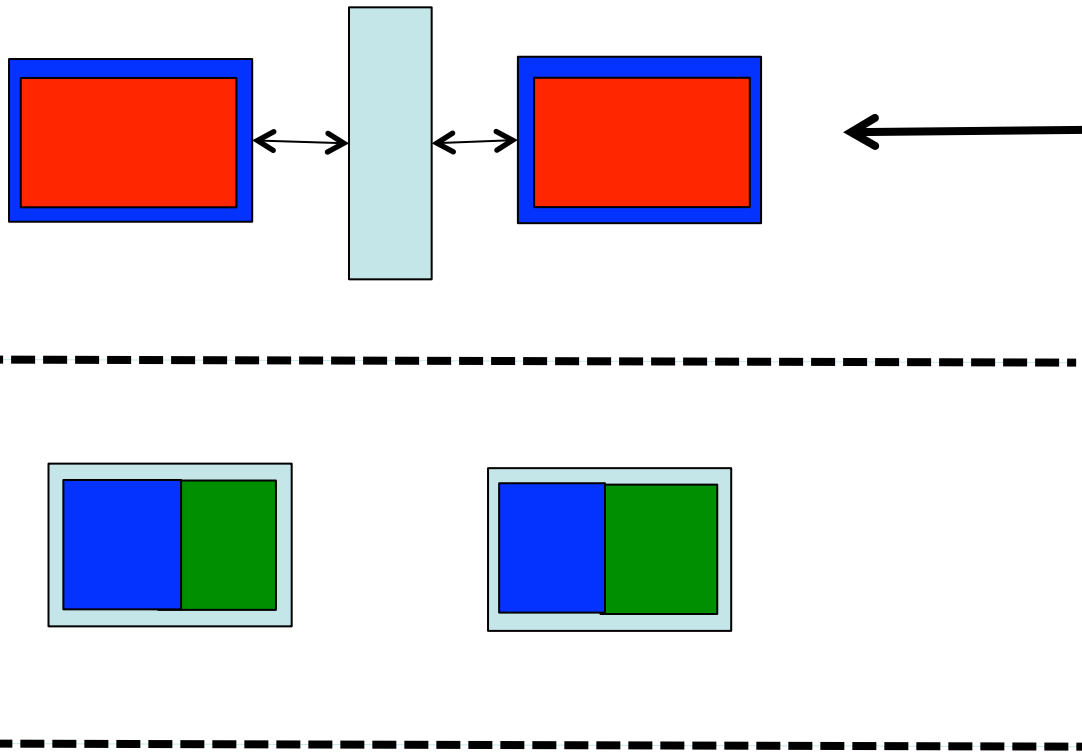


Concurrency in parallel stencil computation (red-black)



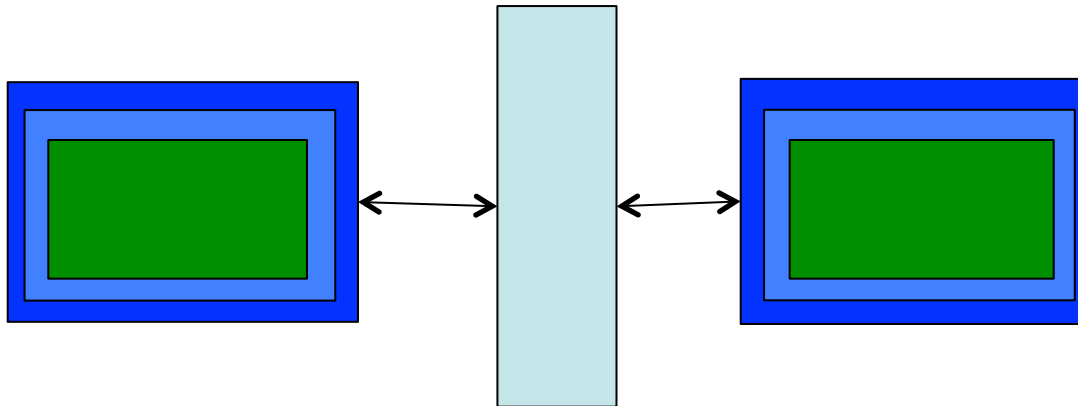
- Halo exchange
- Surface updates independent of inner updates

Algorithms I : Baseline



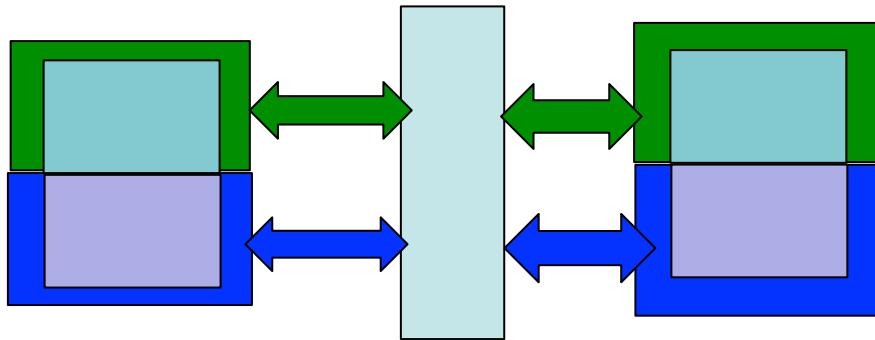
- Two stages
 - ◆ Halo-exchange
 - ◆ Grid update
- Easy to implement
- Unused cores in communication stage

Algorithms II: Computation Communication Overlap



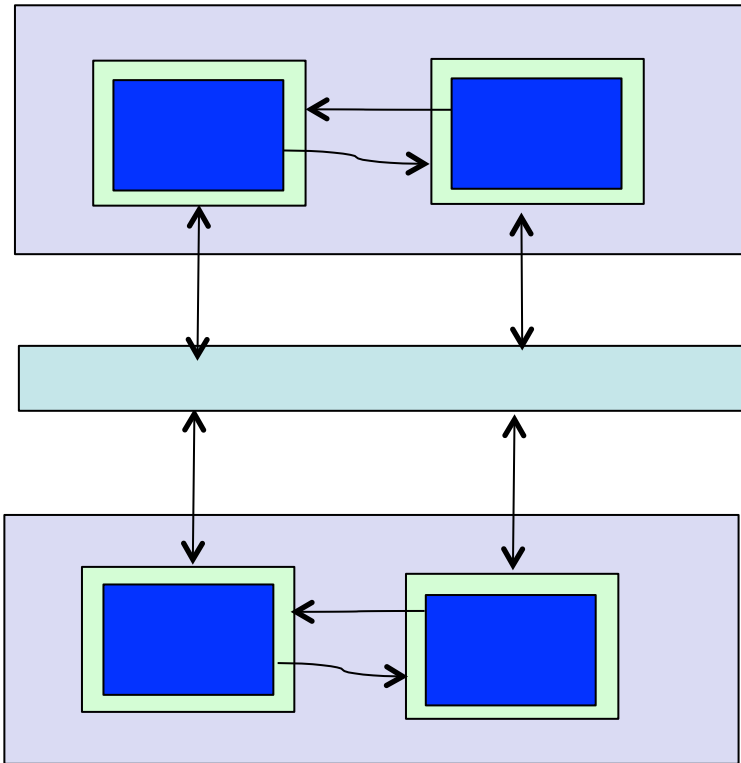
- Master threads exchange halos and updates surface
- Other threads update the inside of the local grid
- Less synchronization
- Load balance needs attention
- Can one thread use the whole bandwidth?
- G. Hager, G. Schubert, T. Schoenemeyer, and G. Wellein: *Prospects for Truly Asynchronous Communication with Pure MPI and Hybrid MPI/OpenMP on Current Supercomputing Platforms*. Proc. Cray Users Group Conference 2011 ([CUG 2011](#)), May 23-26, 2011, Fairbanks, AK.

Algorithms II: Threaded MPI with helper threads



- Easy load balancing as every thread takes care of its halo sector
- Needs threaded MPI implementation
- Needs helper thread to progress the communication while the interior points are updated

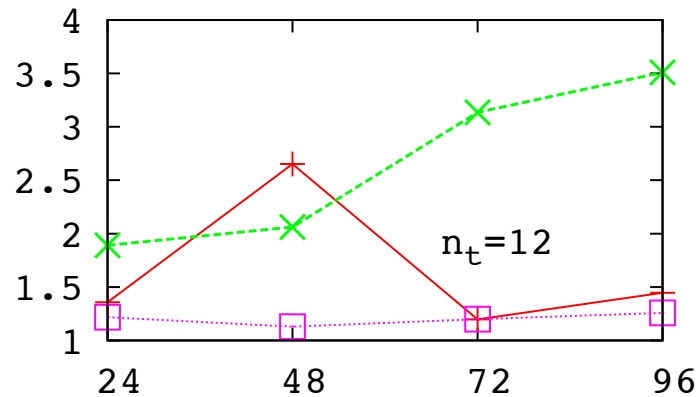
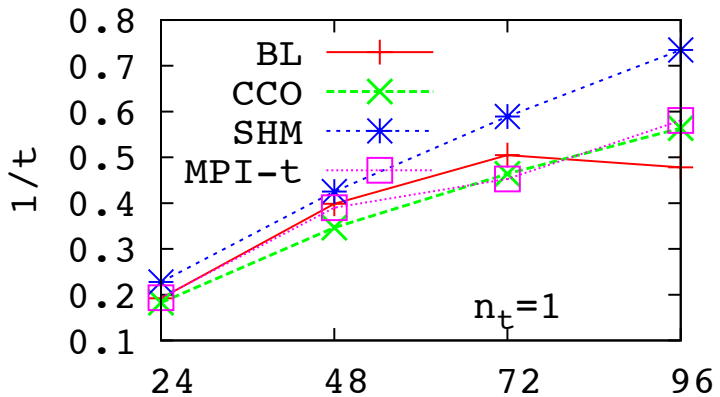
Algorithms IV: Shared Memory with MPI-3



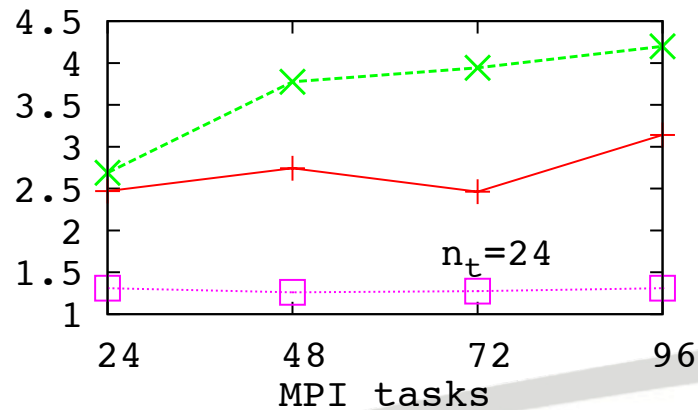
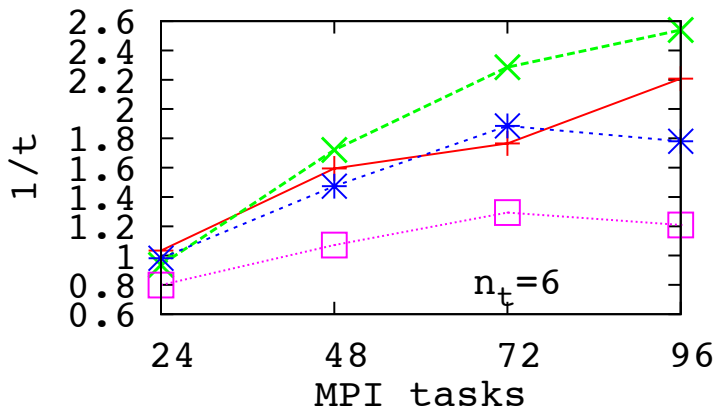
- Arrays shared inside the node (MPI_allocate_shared)
 - ◆ Avoids MPI stack
 - ◆ Better serial optimisation
- Requires extra synchronization inside the node (MPI_Win_fence)

T. Hoefler, J. Dinan, D. Buntinas, P. Balaji, B. Barrett, R. Brightwell, W. Gropp, V. Kale and R. Thakur:
MPI + MPI: a new hybrid approach to parallel programming with MPI plus shared memory Journal of Computing.
Springer, May 2013, doi: 10.1007/s00607-013-0324-2

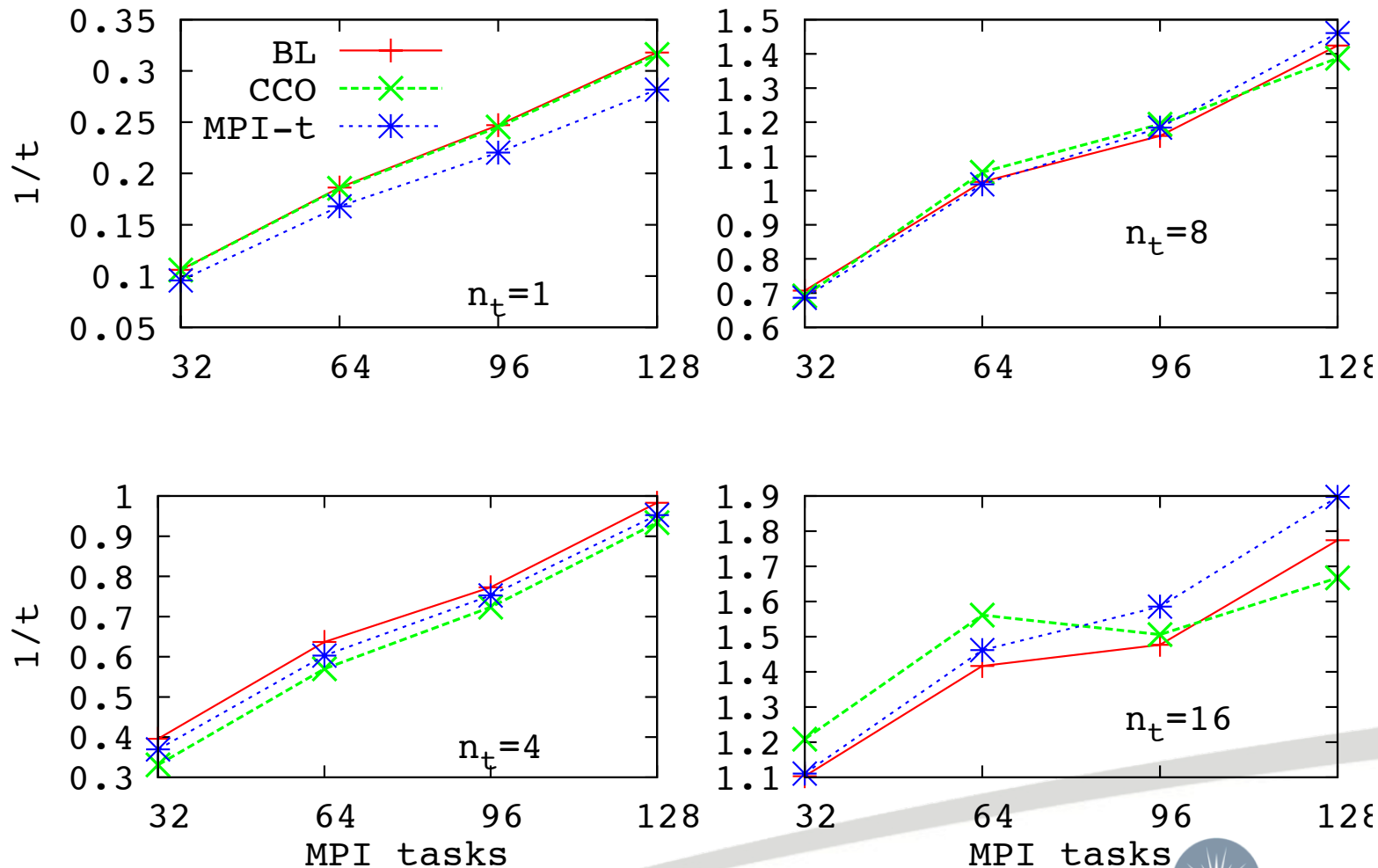
Smoother timings, 1D MPI ARCHER (Cray XC30)



Grid size :
449x545x609

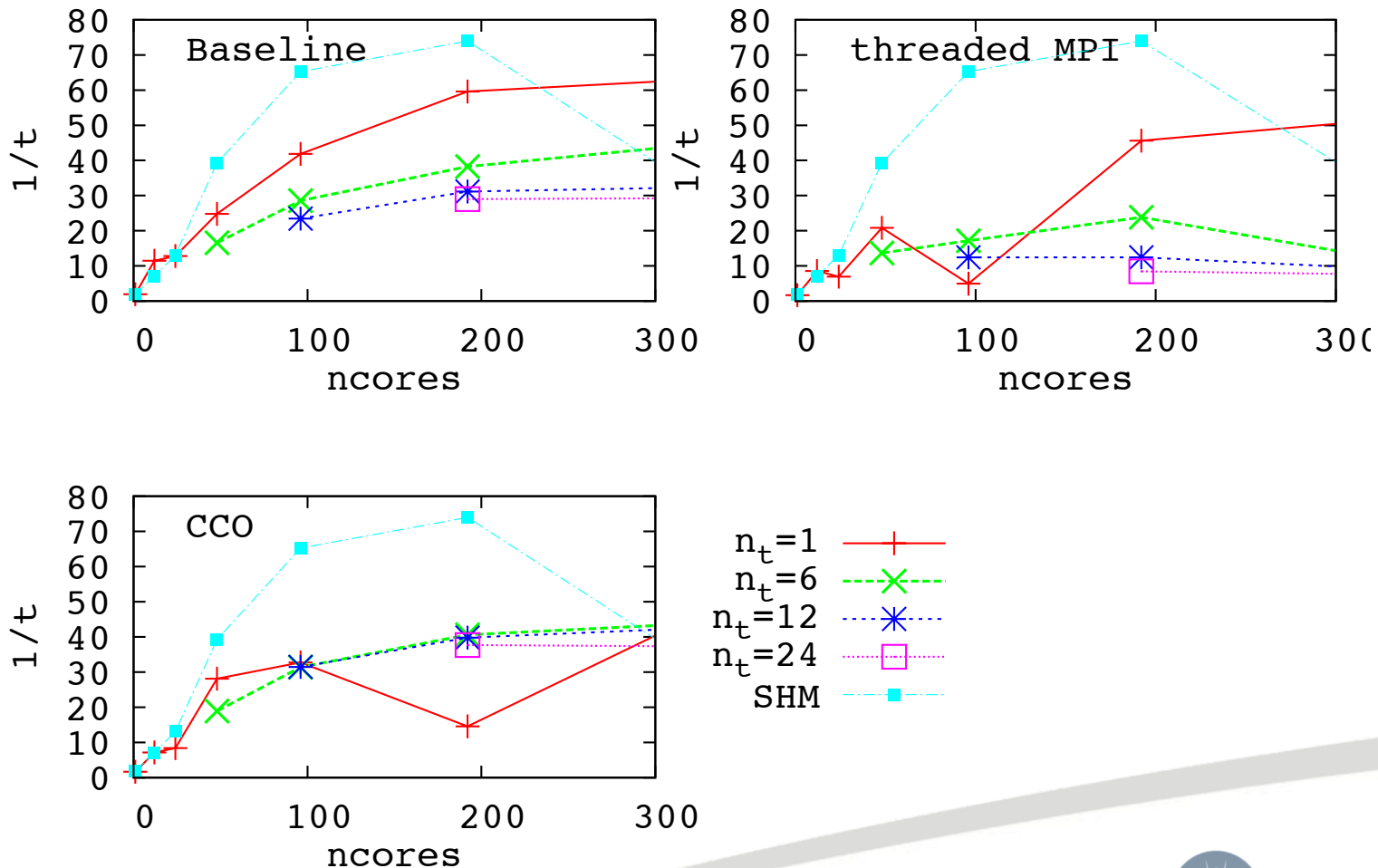


Smoother timings, 1D MPI Blue Gene Q



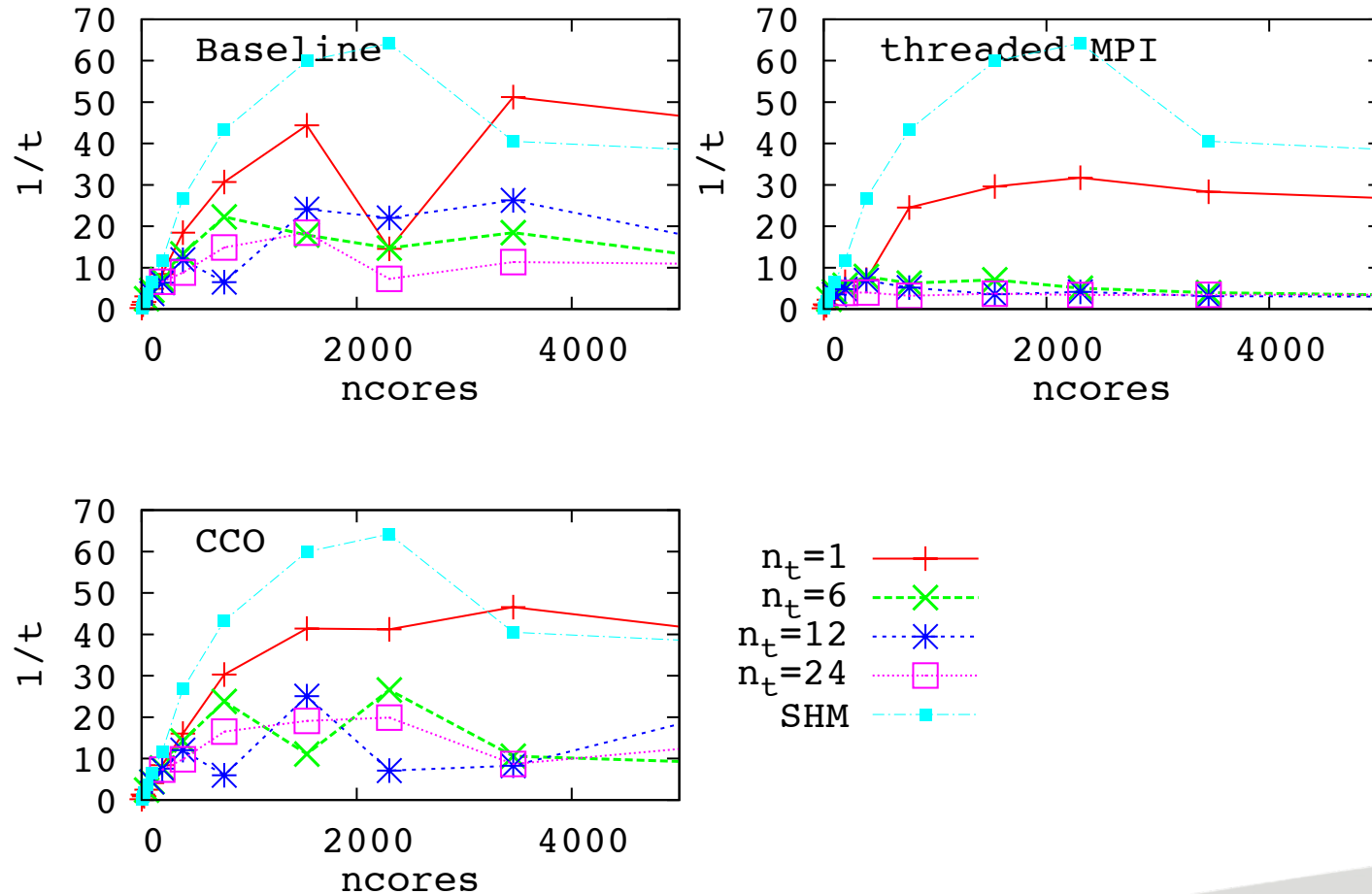
3D smoother

ARCHER 129³



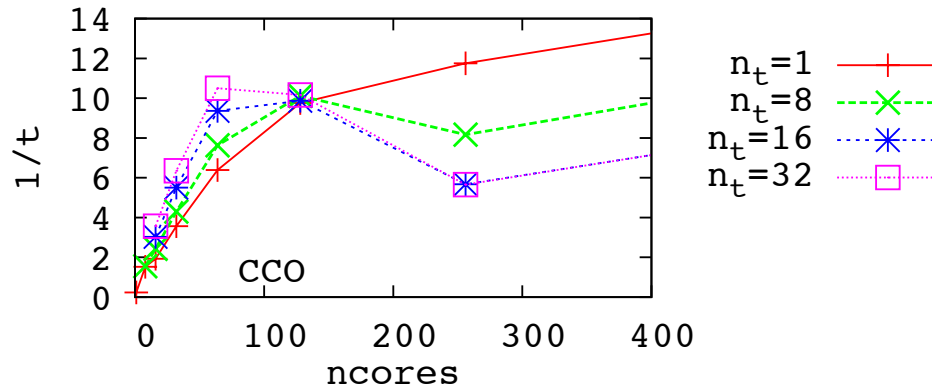
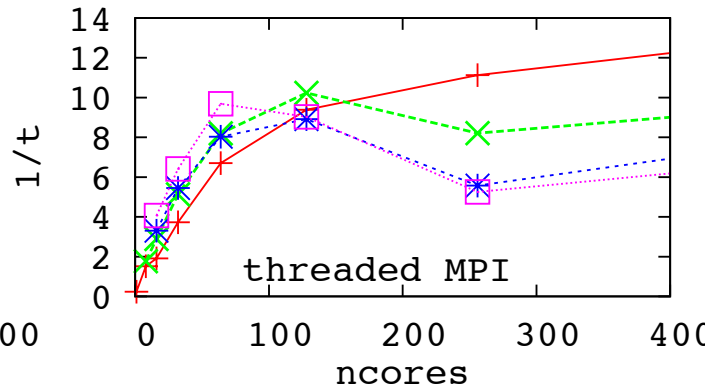
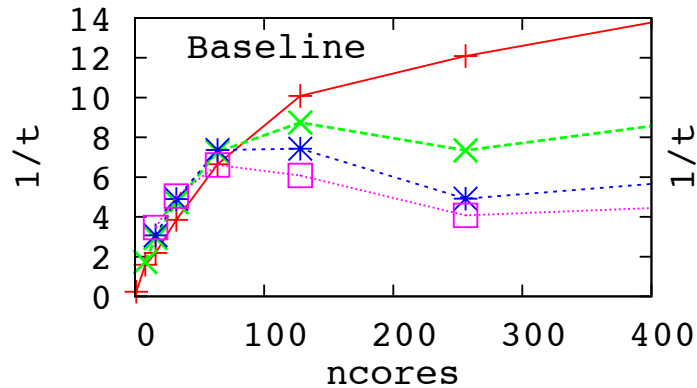
3D smoother

ARCHER 257³



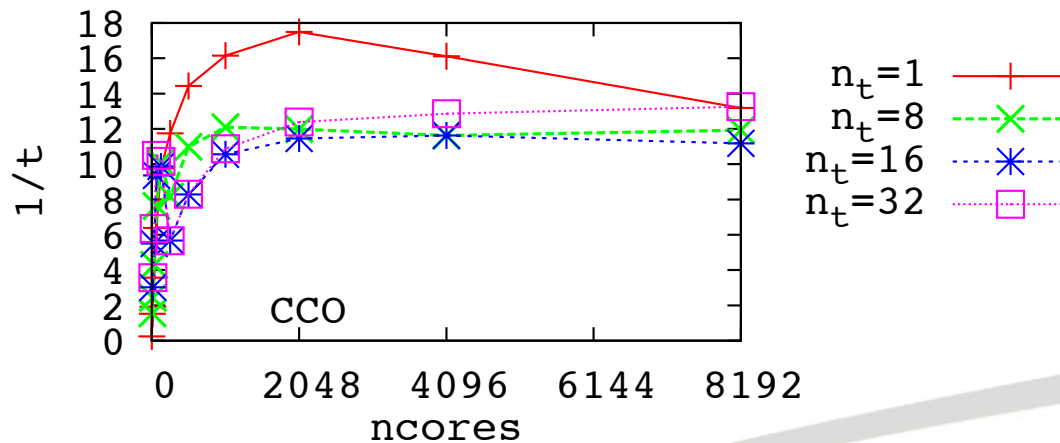
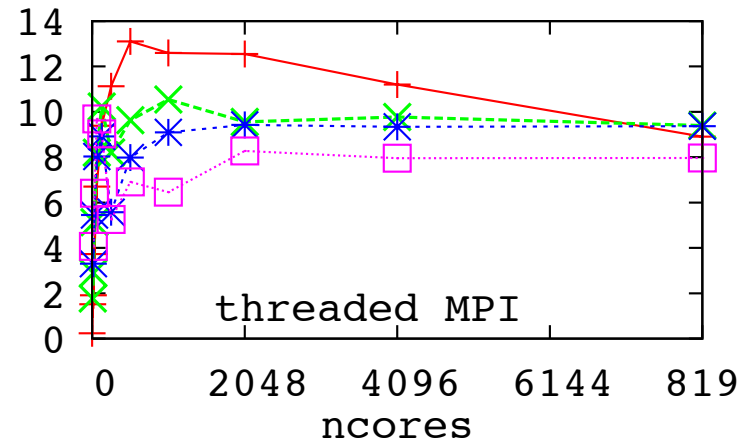
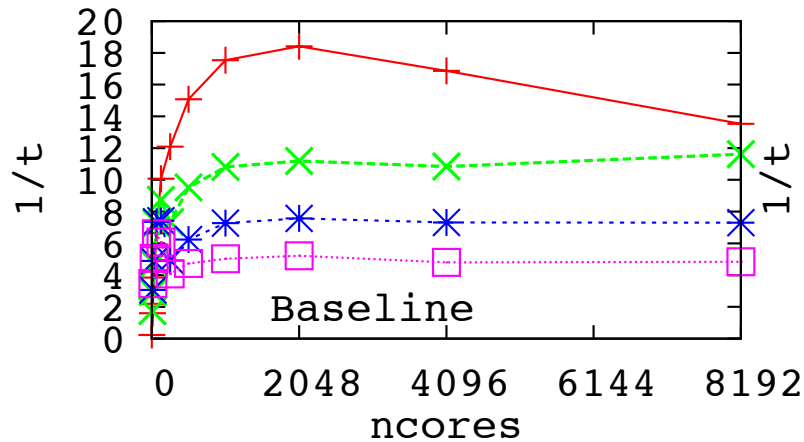
3D smoother

BGQ 129³

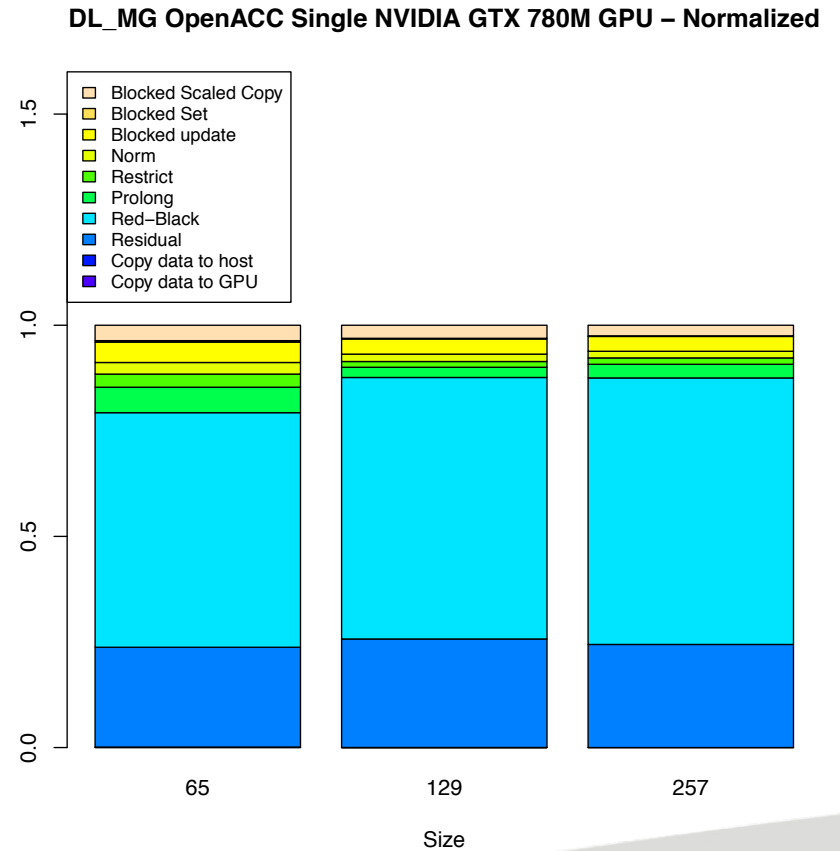
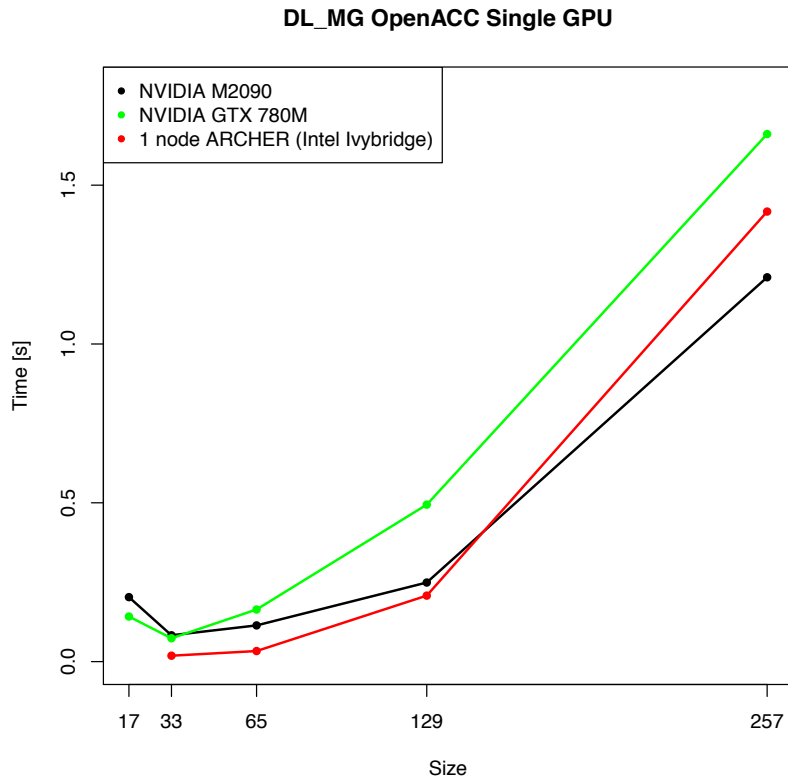


3D smoother

BGQ 129³ ER



GPU



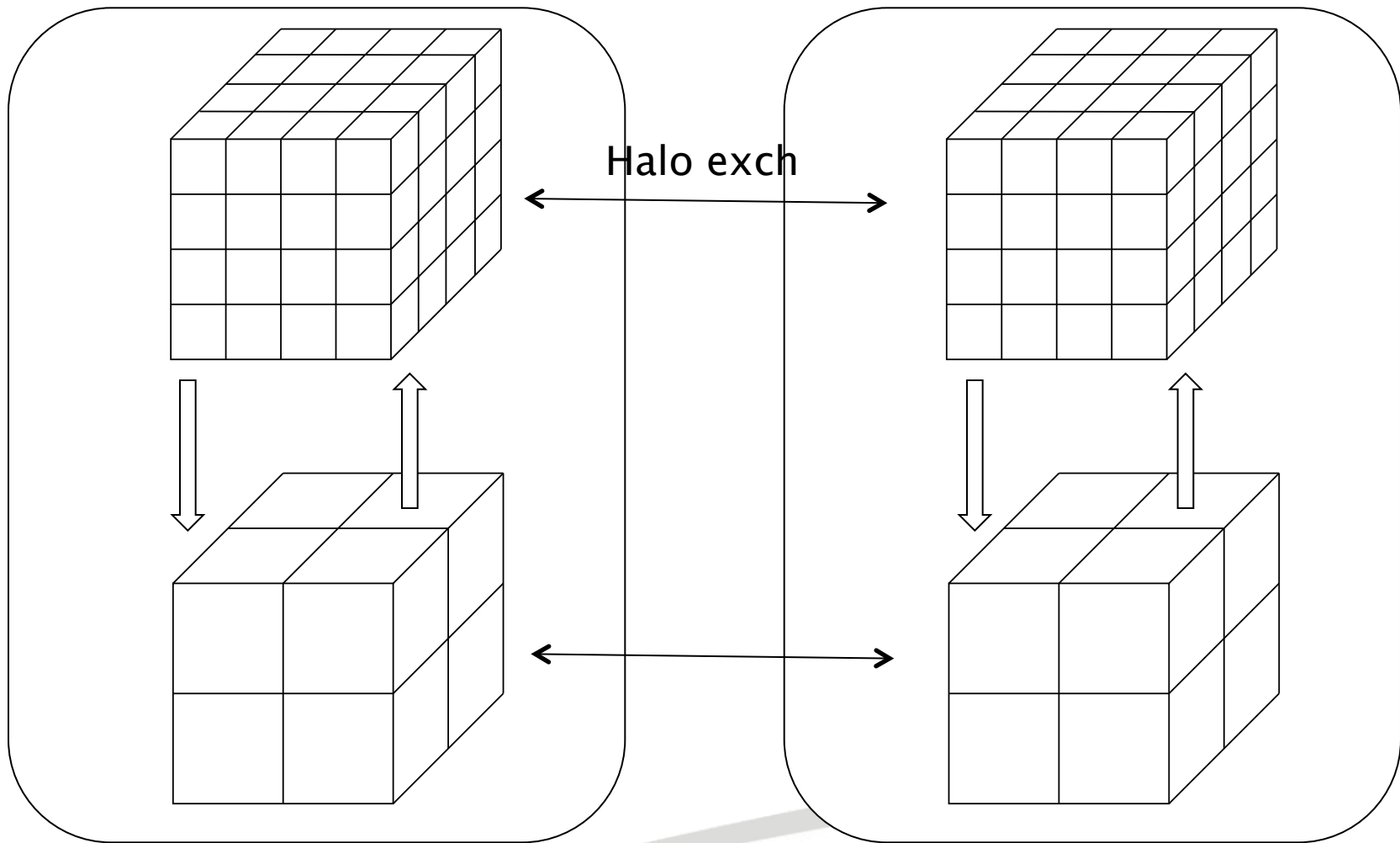
Conclusions

- Shared memory offers a significant boost to parallel performances (scaling range and speed)
- Computation communication overlap is helpful for 1D MPI topology
 - ◆ It needs a tuning mechanisms for “production” strength implementation
- Threaded MPI help performance on BGQ at small core counts.
- Future steps:
 - ◆ These communication patterns can be tested fast in the initialisation stage of the MG solver.
 - ◆ The best one can chosen on the flight.

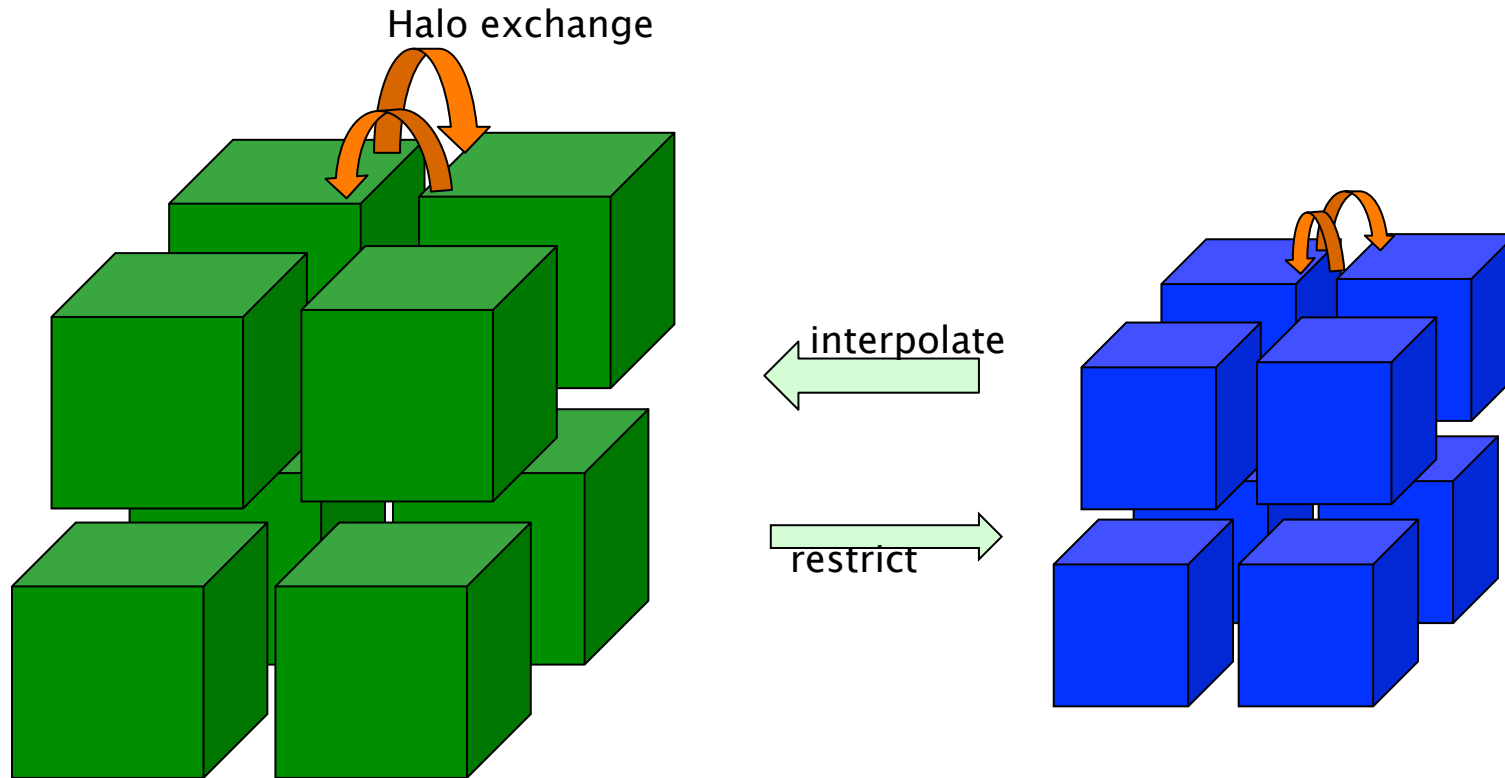
Acknowledgments

- Chris Skylaris and Jacek Dziedzic (U Southampton & U Gdansk)
- Funding from HECToR DSCE program and CCP-ASEArch
- UK HPC facilities
 - ◆ ARCHER (UK national supercomputing facility)
 - ◆ Hartree Centre (IBM-STFC collaboration)
 - ◆ Emerald: e-Infrastructure South GPU supercomputer





Parallel Multigrid



ONETEP

<http://www2.tcm.phy.cam.ac.uk/onetep/>

http://www.oerc.ox.ac.uk/sites/default/files/uploads/ProjectFiles/ASEArch/MultigridWorkshop/Skylaris_ONETEP_Jan2014.pdf

Linear-scaling DFT

- Physical principle

Nearsightedness of electronic matter

W. Kohn, *Phys. Rev. Lett.* **76**, 3168 (1996)

In molecules with non-zero band gap, the density matrix decays exponentially

$$\rho(\mathbf{r}, \mathbf{r}') \sim e^{-\gamma|\mathbf{r}-\mathbf{r}'|} \rightarrow 0 \quad \text{as} \quad |\mathbf{r} - \mathbf{r}'| \rightarrow \infty$$

- Linear-scaling approaches

Truncate exponential “tail”

$$\rho(\mathbf{r}, \mathbf{r}') = 0 \quad \text{when} \quad |\mathbf{r} - \mathbf{r}'| > r_{\text{cut}}$$

- Practical implementation

- Localised orbitals
- Sparse matrices
- No diagonalisation
- Sparse matrix algorithms – $O(N)$ memory and CPU cost

