Improving CASINO performance for models with large number of electrons

Lucian Anton[1], Dario Alfe[2], Randolph Q Hood[3], David Tanqueray[4]

[1]Nag Ltd, Oxford, UK
[2]University College of London, UK
[3]LLNL, USA
[4]CRAY UK, Reading, UK

CUG, Atlanta, May 2009

Outline

- Introduction
- Algorithms for distributed data
  - Shared memory
  - MPI two-sided
  - MPI one-sided & SHMEM
- Second level parallelism
- Final remarks & conclusions

QMC and CASINO

Quantum Monte Carlo techniques are used to compute electronic structure of solids, large molecules or nano-clusters:

- Very precise results
- Good scaling with system size
- Good parallel efficiency

CASINO is a QMC code developed by Theory of Condensed Matter group, Cambridge University.

Fortran 95 +MPI

http://www.tcm.phy.cam.ac.uk/~mdt26/casino2.html

Mathematical background I

Quantum many-body systems: N_e electrons, N_I ions.
Computationally challenging problem and of practical interest.

$$i\hbar\frac{\partial\Psi(R,t)}{\partial t}=-\frac{\hbar^2}{2\mathrm{m}}\sum_{i=1}^{N_e}\nabla_i^2\Psi(R,t)+V(R,R_I)\Psi(R,t)$$

$$E=\frac{\langle\Psi|H\Psi\rangle}{\langle\Psi|\Psi\rangle}$$

Mathematical Background II

$$D_{\uparrow}=\begin{vmatrix} \phi_1(r_1) & \phi_1(r_2) & \cdots & \phi_1(r_{N_{\uparrow}}) \\ \vdots & \cdots & \cdots & \vdots \\ \phi_{N_{\uparrow}}(r_1) & \phi_{N_{\uparrow}}(r_2) & \cdots & \phi_{N_{\uparrow}}(r_{N_{\uparrow}}) \end{vmatrix}$$

One particle solution using Slater determinants (DFT, HF)

QMC computations incorporate particle correlation into the solution.

$$\Psi = e^{J(\alpha, R, R_I)} D_{\uparrow} D_{\downarrow}$$

$$J(R) = \sum_{\substack{i<j, \\ \sigma_i, \sigma_j}} u_{\sigma_i, \sigma_j}(\alpha, |r_i - r_j|)$$

Mathematical Background III

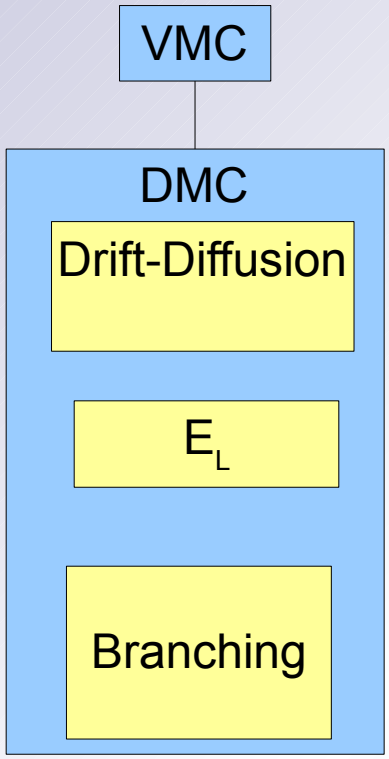
VMC: optimize Jastrow function parameters with

$$E(\alpha)=\int dR|\Psi(\alpha,R)|^2\frac{\boldsymbol{H}\Psi(\alpha,R)}{\Psi(\alpha,R)}$$

DMC: projects the trial wavefunction unto the groundstate.

$$\Psi(R,\tau)=\exp\left[-\frac{\boldsymbol{H}}{\hbar}\tau\right]\Psi(\alpha,R)$$

CASINO QMC computational steps



$$\frac{1}{(2\pi\tau)^{3N/2}}\exp\left(-\frac{\left(\boldsymbol{R}-\boldsymbol{R}'-\tau\boldsymbol{V}(\boldsymbol{R}')\right)^2}{2\tau}\right)$$

$$E_L(\boldsymbol{R})=\Psi^{-1}\boldsymbol{H}\Psi$$

$$\exp\left(-\frac{\tau}{2}\left[E_L(\boldsymbol{R})+E_L(\boldsymbol{R}')-2E_T\right]\right)$$

The origin of the memory problem

Practical computations start from electronic orbitals obtained with DFT and stored as B-Spline expansion.

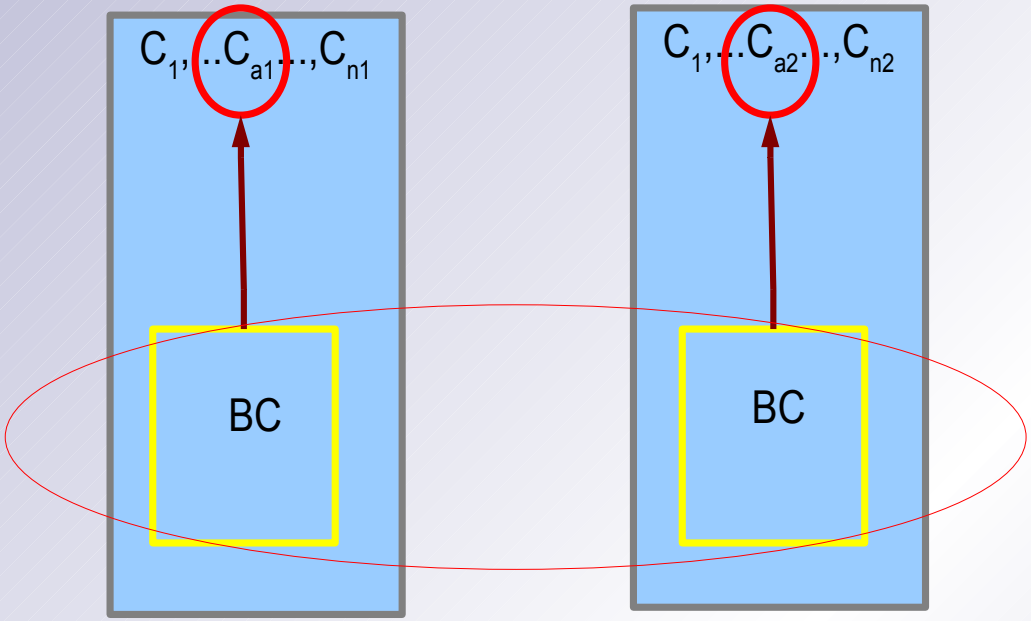
$$bc(N_o, 0:N_{gx}-1, 0:N_{gy}-1, 0:N_{gz}-1, N_s)$$

1024 electrons need 512 OPO

80 grid points in each direction direction

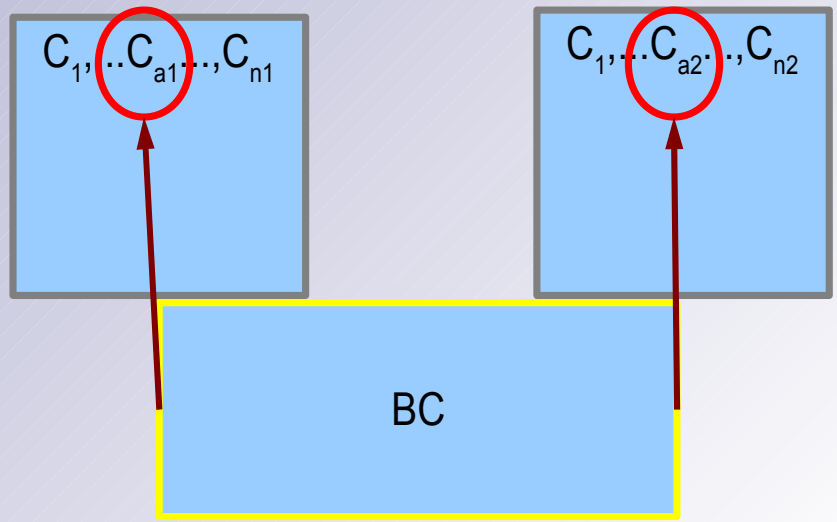
>2GB in double precision

Memory organisation



Can we share BC memory on a processor or a node?

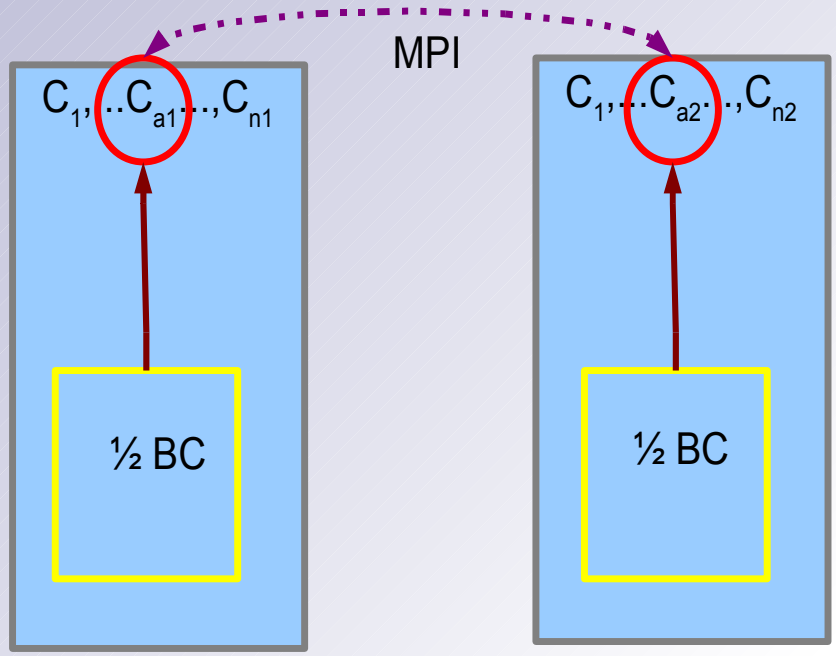
SHM



No MPI solution to share memory on a node, but one can use Unix inter process communication library:

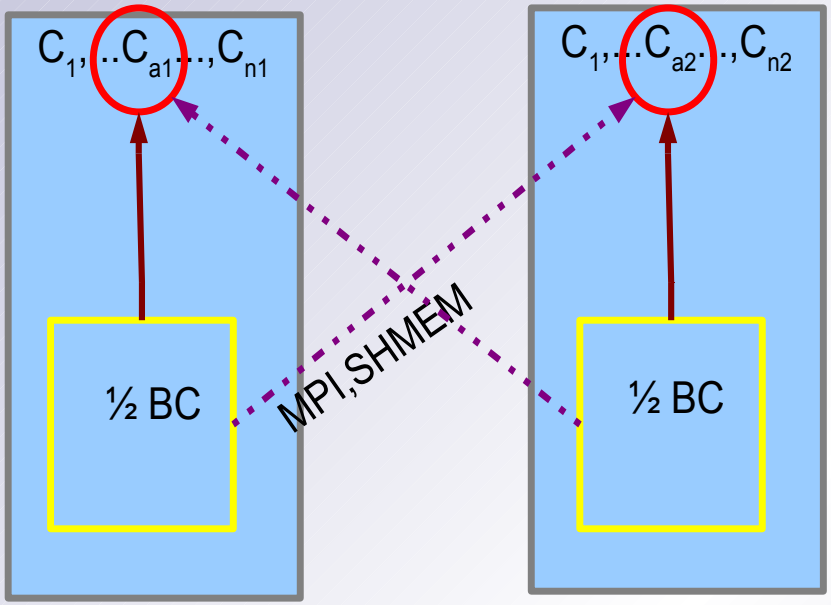
- Easy to implement.
- Needs C functions to allocate the shared memory.
- Cray pointers to pass the reference to the FORTRAN pointers.

MPI-2S



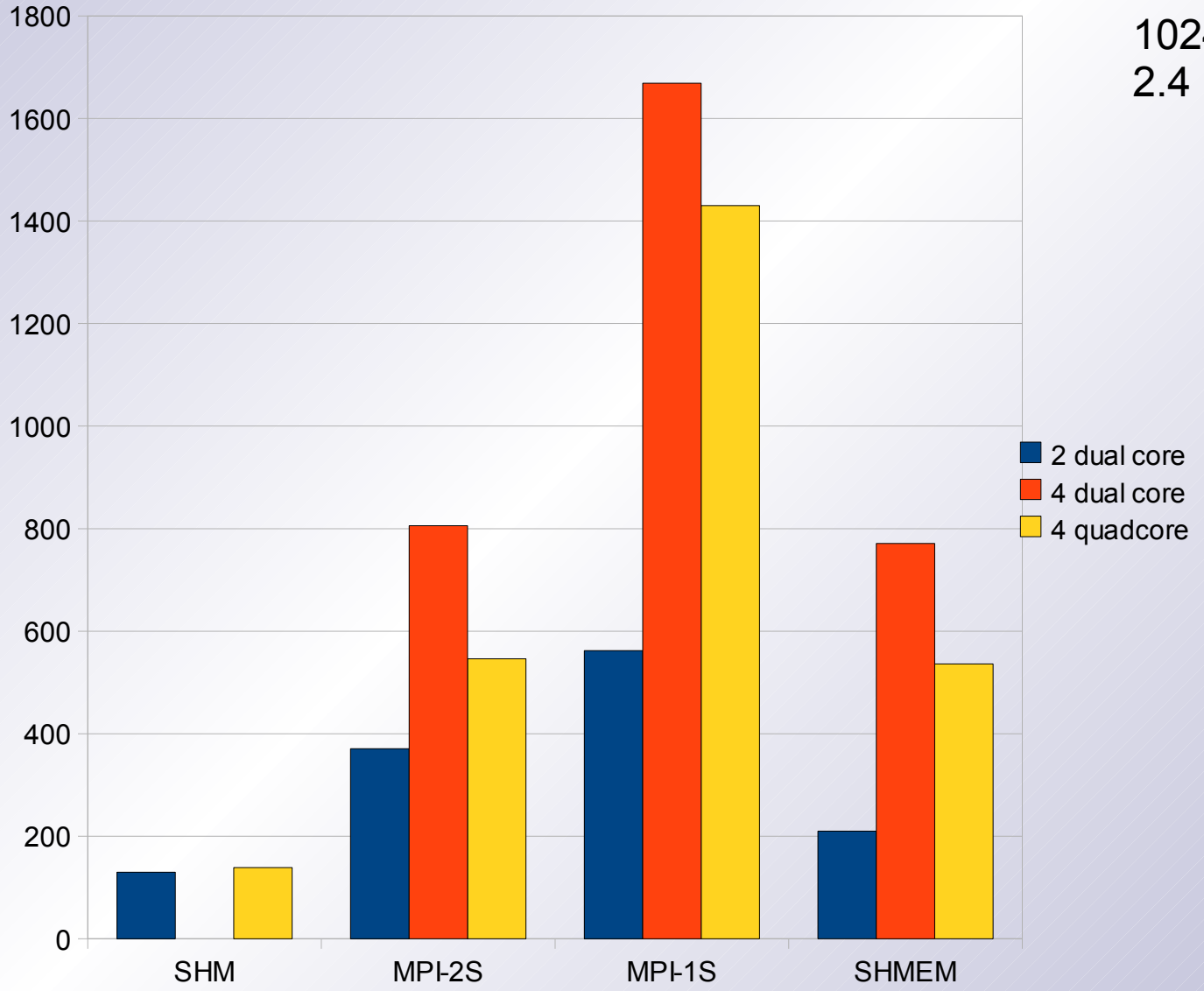
- No need of shared memory
- Fully compliant with CASINO coding standard
- Call for orbital computation must be synchronous

MPI-1S



Can we avoid the synchronisation of MPI-2S with MPI one-sided or CRAY SHMEM library?

Timing results



1024 e
2.4 GB of BC

Second level parallelism I

Why is needed?

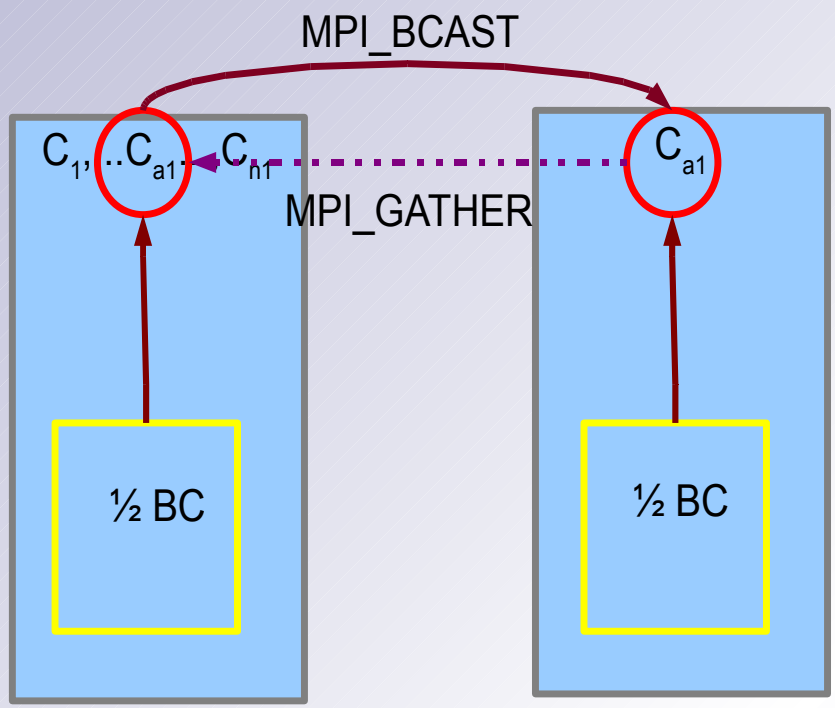
For one configuration step we have:

- The sums involved in computing the energy terms scales as N^2
- Slater matrix elements: N^2
- Slater determinant: N^3 (LU decomposition) or N^2(cofactor matrix)

$$T[O(10^4)] \approx 10^{2.x} T[O(10^2)]$$

QMC algorithms for electronic structure at the petascale
K P Esler *et al*, J Phys: Conf Series, **125**(2008) 0122057

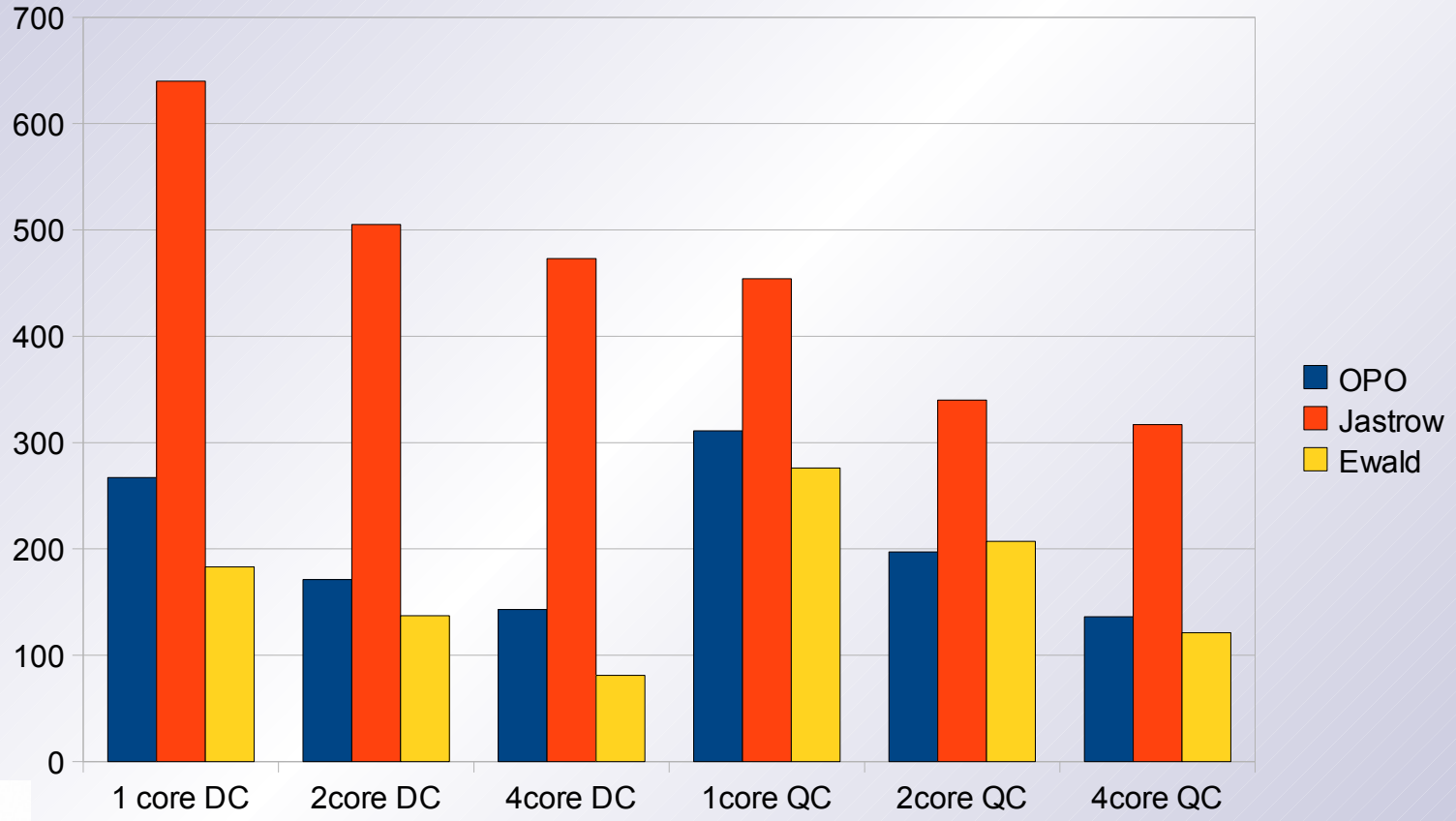
Second level parallelism II



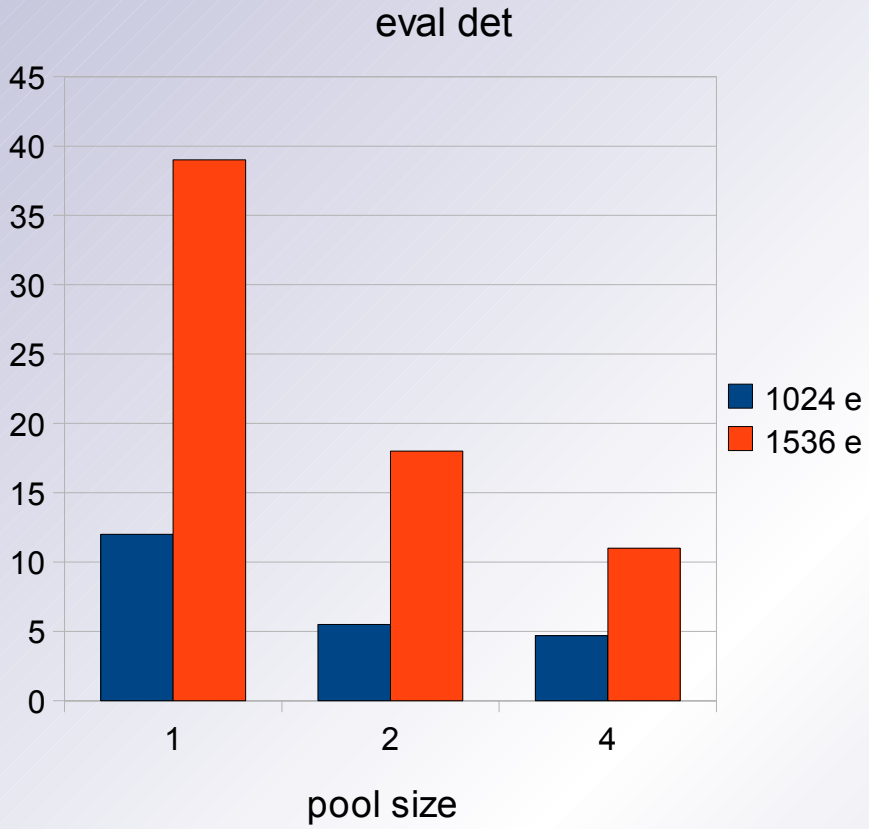
- The pool computes for the same configuration: OPO, Jastrow factor, energy components, Slater determinats.
- The computation is controlled by the pool head.

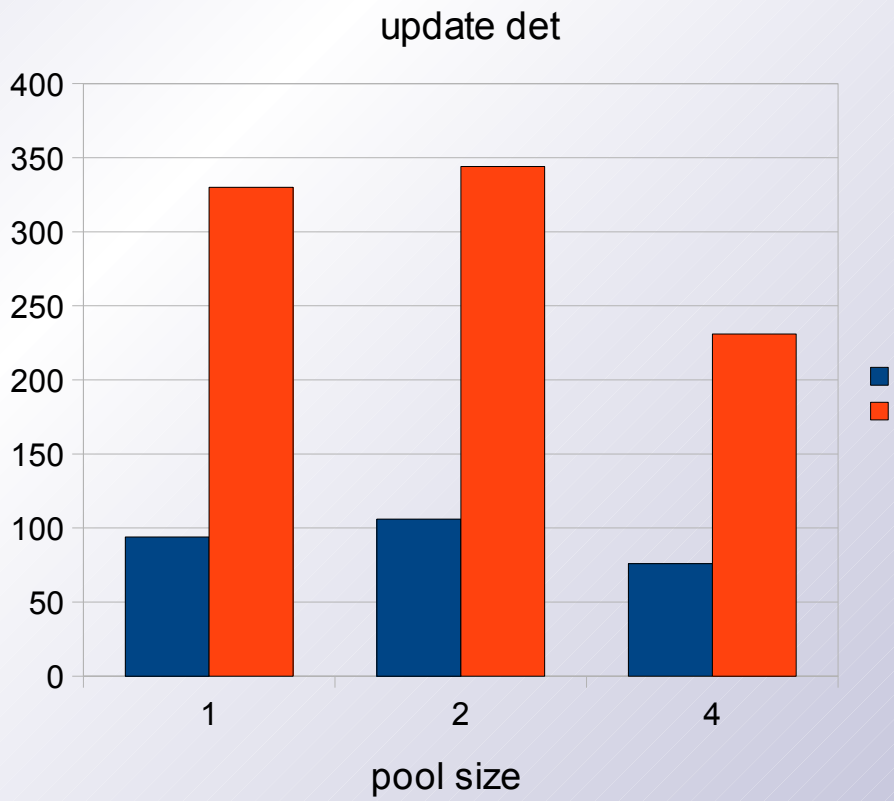
SLP timings I

1536 electrons, BC ≈4.8 GB



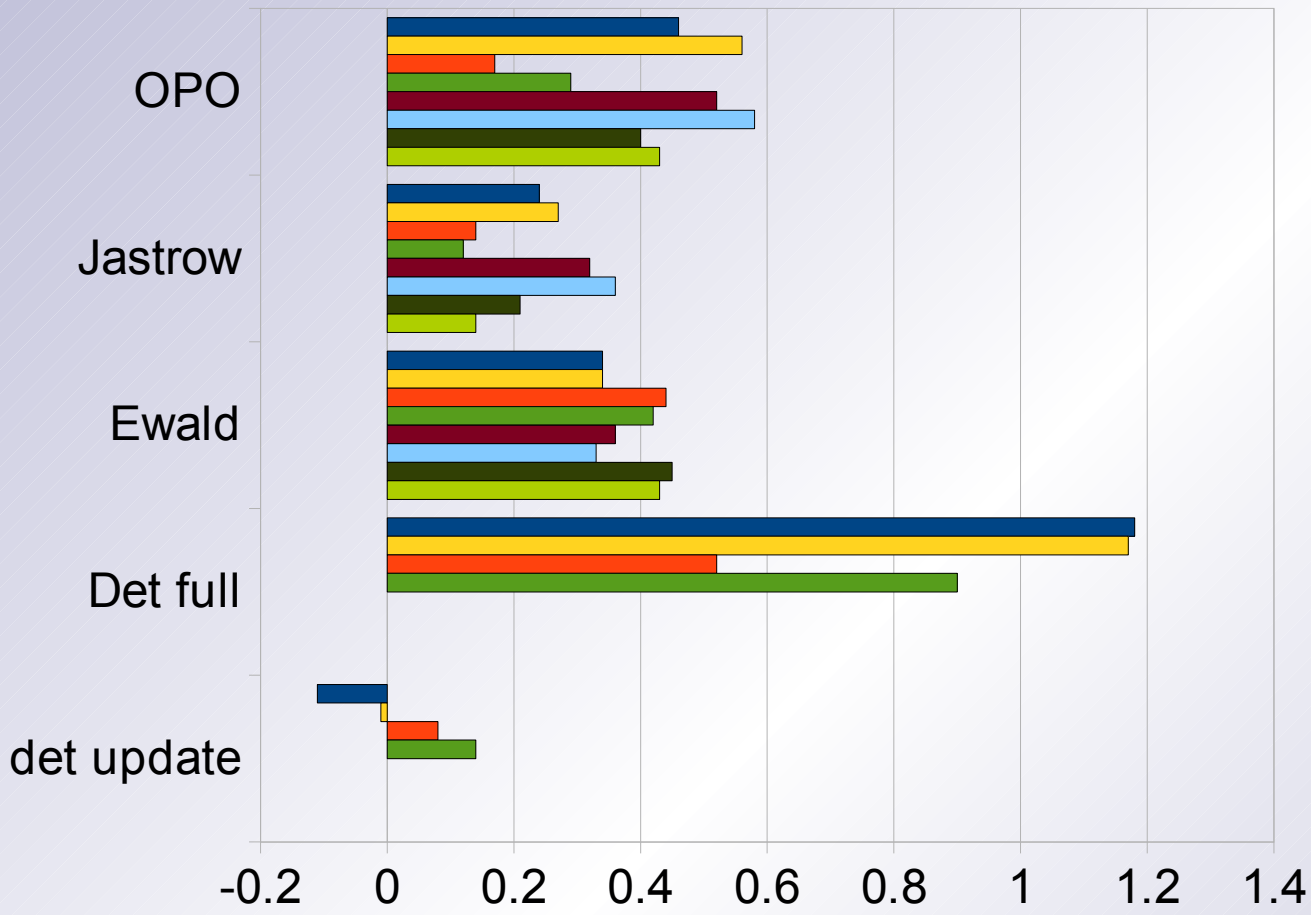
SLP timing II





Dual core processors

SLP efficiency



S1=1024 e
S2=1536 e

$$\eta = \frac{t_1 / t_n - 1}{n - 1}$$

Conclusions

- Shared memory with Unix interprocess communication
- Alternative MPI two-sided implementation
- IO performance improved as well, better cache utilisation
- SLP reached efficiency close to 50%
- OpenMP discussion