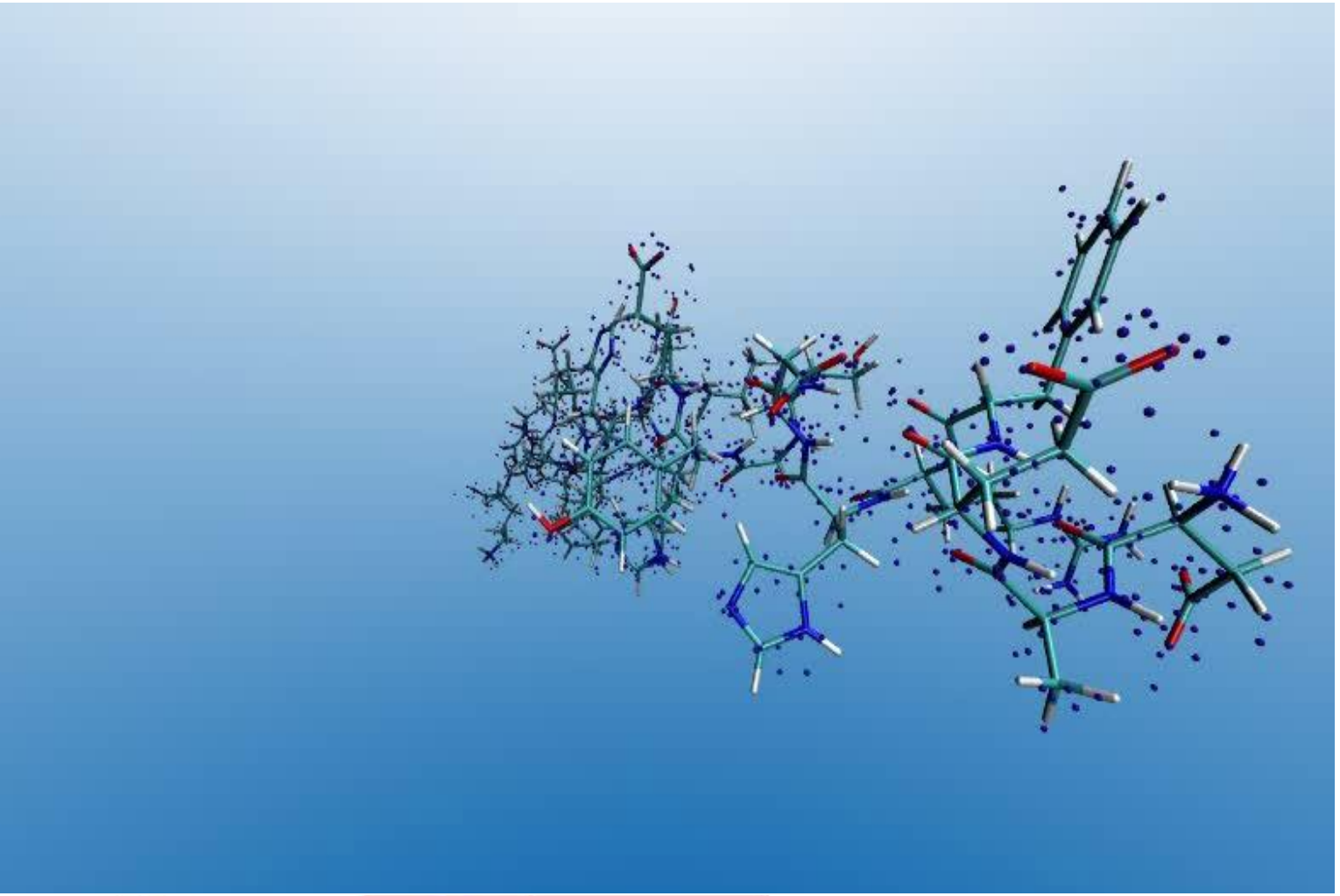
Quantum chemistry towards exascale with QMC=Chem



A. Scemama,
M. Caffarel
Laboratoire de Chimie et Physique Quantiques
CNRS - IRSAMC
Université de Toulouse, France

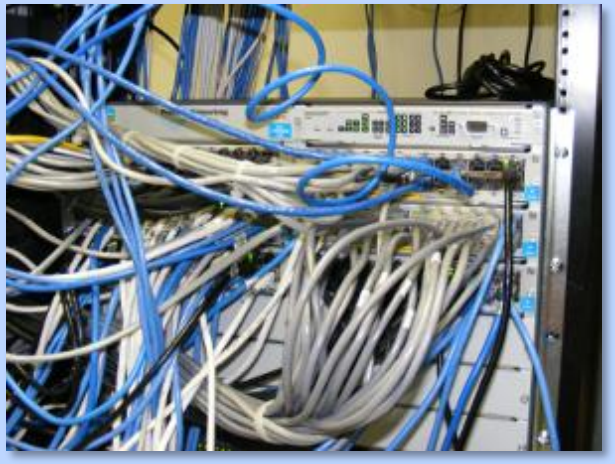
E. Oseret,
W. Jalby
Exascale Computing Research Laboratory
GENCI-CEA-Intel
Université de Versailles St Quentin, France

www.cnrs.fr

Ab initio Quantum chemistry

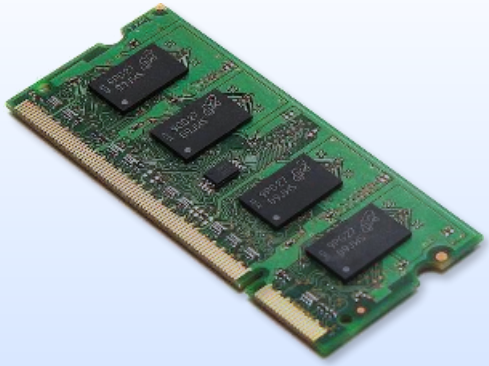
- Many chemical problems need highly accurate models, using an *ab initio* quantum mechanical description (Configuration Interaction or Coupled Cluster methods).

- Such methods need a large amount of memory and disk space.



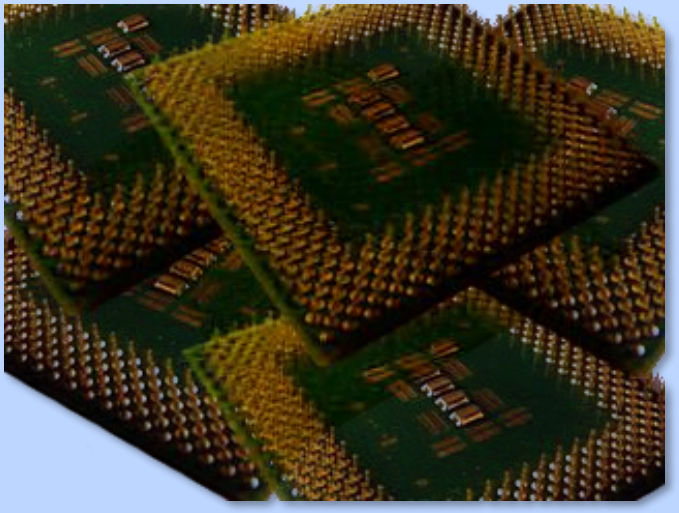
- Due to their iterative nature, synchronizations limit the parallel efficiency.
- These methods are not yet suited to massively parallel machines

Quantum Monte Carlo



Quantum Monte Carlo (QMC) are methods that :

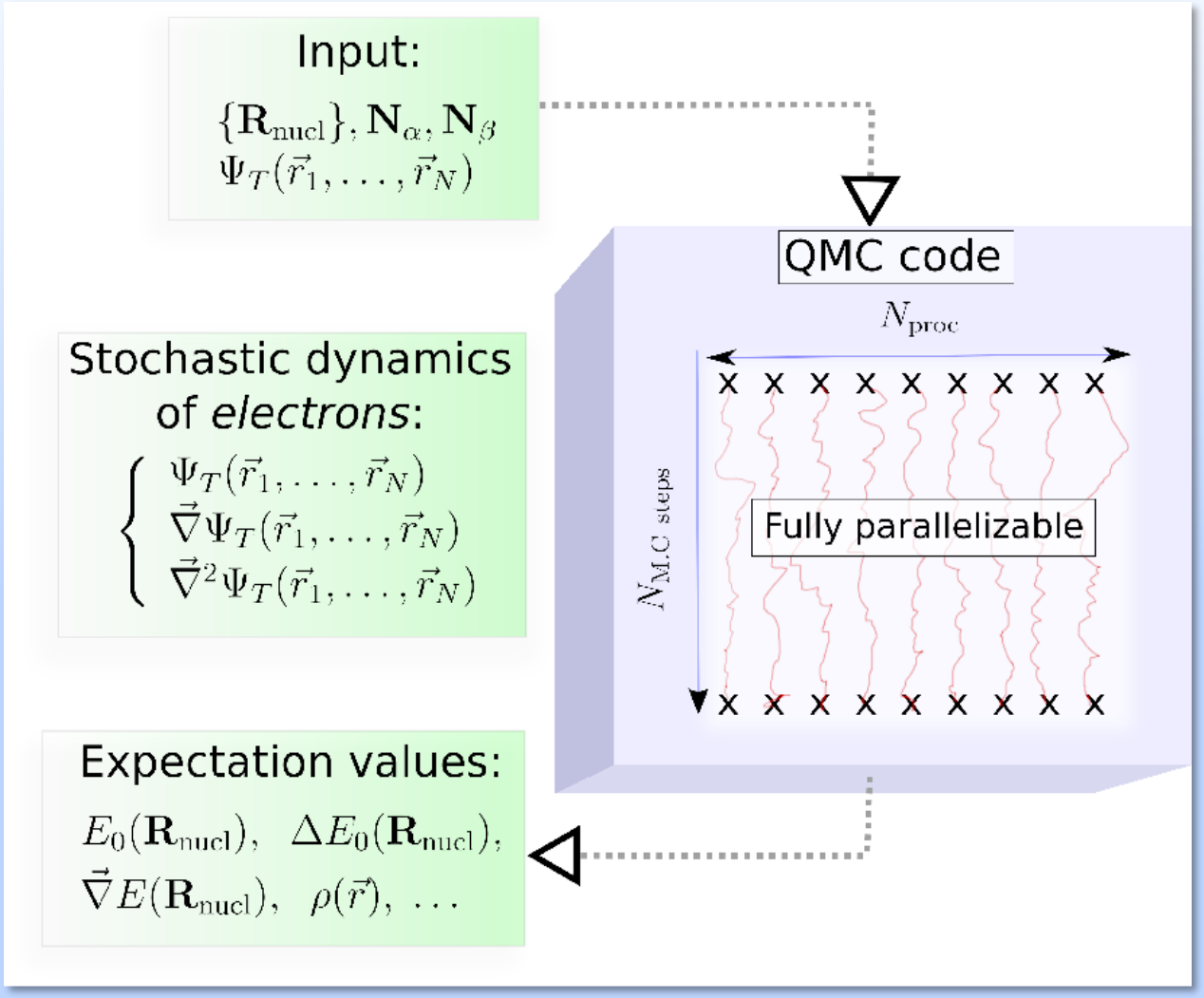
- require a small amount of memory (~100MB per core)
- make very few network communications
- have a much better scaling than standard methods with the size of the chemical system



but:
require a large amount of CPU time

The evolution of massively parallel machines is very favorable to QMC methods

QMC methods are good candidates for exascale

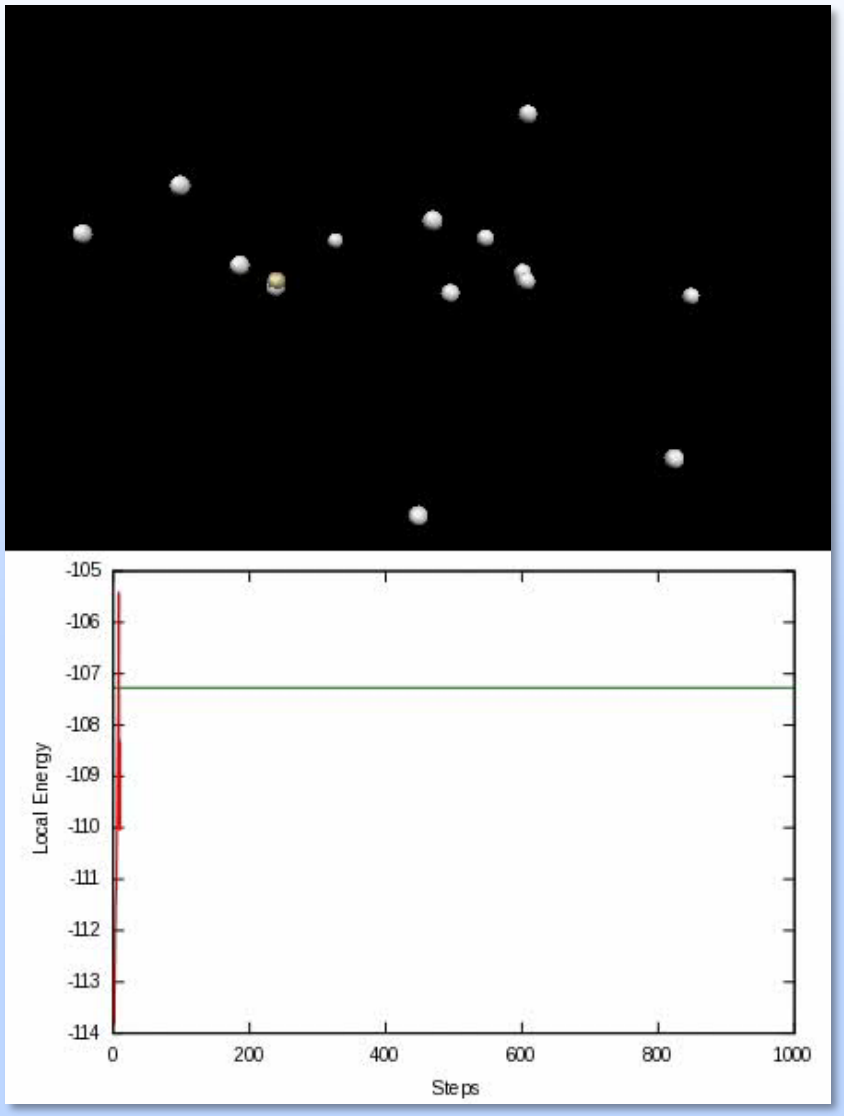


During the run, each process is completely autonomous (single core processes).
The scaling is *ideal* !

Communications are mandatory *only* at the initialization and the finalization stages.

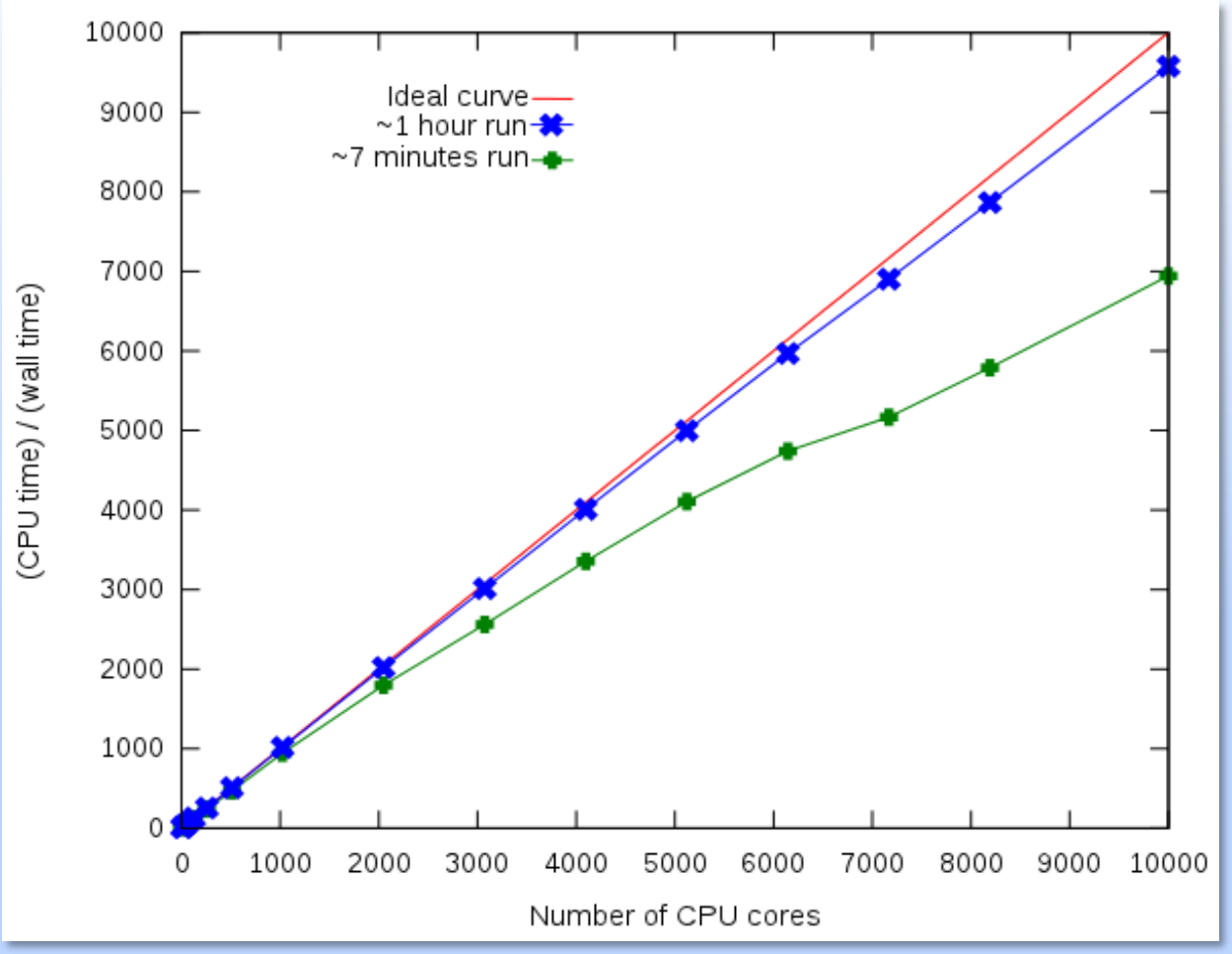
The initialization and finalization times don't depend on the length of the run.

QMC=Chem key points



- All the processes are completely independent
- Additional compute nodes can be added/removed dynamically to a running simulation
- Fault tolerance : any compute node can fail without killing the whole run
- I/O and network communications are fully asynchronous
- An almost ideal scaling with the number of cores is obtained
- Very good single core performance

Parallel efficiency of QMC=Chem



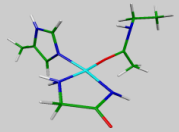
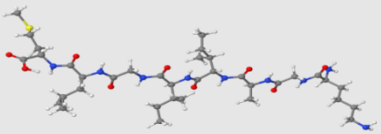
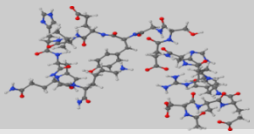
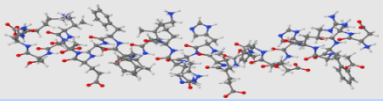
The performance of the application is determined by the efficiency of the *single-core* executable

Benchmarks performed on Curie (TGCC/CEA/GENCI, France) in April 2011.
The blue curve is estimated from the data collected on the green curve

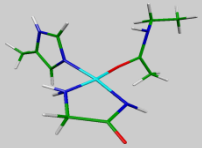
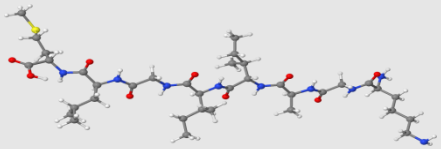
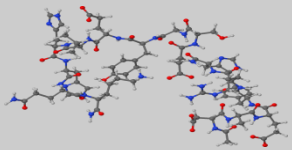
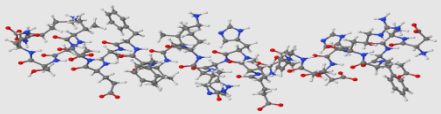
Single-core performance

The scaling of one Monte Carlo step is limited by

- A matrix inversion, via the Intel MKL library (O(N^3))
- Matrix-matrix multiplications using an efficient sparse-dense implementation (O(N^2))

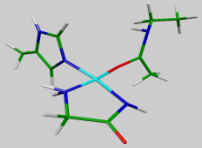
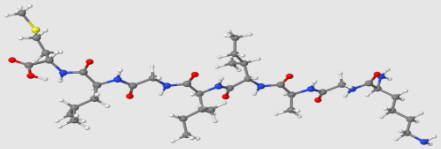
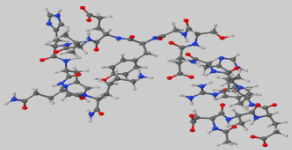
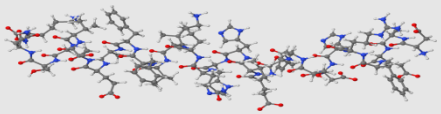
Molecular System	Number of electrons	% Peak performance of Multiplication	% Peak performance of Inversion	time(Inversion)/ time(Multiplication)
	158	50%	24%	0.5
	434	64%	53%	0.6
	1056	58%	68%	2.0
	1731	53%	68%	2.6

Single-core performance

Molecular System	Number of electrons	RAM/core (MB)	CPU time / step on Core2[1] (s)	CPU time / step on Sandy Bridge[2] (s)
	158	9.8	0.0073	0.0033
	434	65	0.0504	0.0186
	1056	133	0.3421	0.0980
	1731	313	1.2480	0.4226

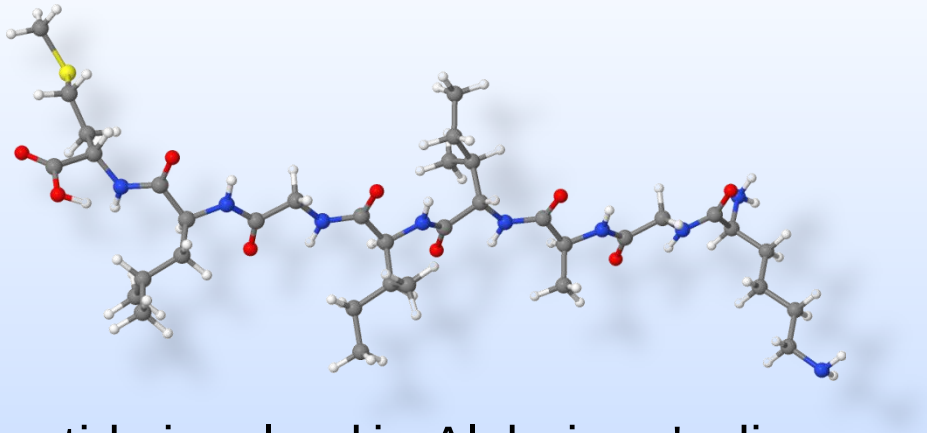
1. Intel Xeon 5140, Core 2 2.33GHz, Dual core, 4MB shared L2 cache
2. Intel Xeon E31240, Sandy Bridge 3.30GHz, Quad core, 256KB L2 cache/core, 8MB shared L3 cache

Single-core performance

Molecular System	Number of electrons	RAM/core (MB)	% CPU Peak performance on Core2[2]	% CPU Peak performance on Sandy Bridge[2]
	158	9.8	25%	23%
	434	65	34%	38%
	1056	133	37%	49%
	1731	313	47%	55%

1. Intel Xeon 5140, Core 2 2.33GHz, Dual core, 4MB shared L2 cache
2. Intel Xeon E31240, Sandy Bridge 3.30GHz, Quad core, 256KB L2 cache/core, 8MB shared L3 cache

Latest performance results



We have modelled a peptide involved in Alzheimer's disease on a BullX supercomputer:

- Intel Sandy Bridge sockets, 2.7GHz, 8 cores, 20MB cache
- 190 dual socket nodes (3 040 cores)
- 64GB RAM/node

An average of 27.8 TFlops/s was obtained on a 7 minutes run.

For a 1 hour run we would obtain 31.2 TFlops/s.

We would be able to reach 1PFlops/s with 97 500 cores for 1 hour.