Multicore Applications in Physics and Biochemical Research

Hristo Iliev

Faculty of Physics

Sofia University "St. Kliment Ohridski"

3rd Balkan Conference in Informatics

The culmination of our HPC efforts so far:

PHYSON

Joint project between Sofia University and Technical University
Assembled by PERSY Ltd
Proudly hosted in the department of Theoretical Physics
Faculty of Physics, Sofia University "St. Kliment Ohridski"
http://physon.phys.uni-sofia.bg/ (in Bulgarian only)

Physon – the pink cluster

- 4 compute nodes:
  - dual Intel Xeon E5335 (4 cores @ 2 GHz)
  - 12 GB (16 GB @ node001) ECC DDR2-667 RAM
  - 250 GB SATA2 HDD
  - 2 x 1 Gbps Fast Ethernet
  - 1 x 20 Gbps 4x DDR InfiniBand
  - Scientific Linux 4.4 64-bit
  - Sun N1 Grid Engine executives
- NFS/NIS server:
  - Intel Core2 Duo E6600 (2 cores @ 2,4 GHz)
  - 2 GB ECC DDR2-667 RAM
  - 4 x 500 GB SATA2 HDD (total of 1,75 TB in ZFS RAIDZ1 array)
  - 2 x 1 Gbps Fast Ethernet
  - Sun Solaris
  - Sun N1 Grid Engine master

Sun N1 Grid Engine queues

- Interactive queue
  - For GUI and interactive programs
  - Up to 8 cores (1 node)
- Batch queues
  - For long running non-interactive tasks
  - Up to 24 cores (3 nodes)
  - Interactive node available for serial batch tasks at night
- Fair share scheduler (to be enabled soon)

Parallel environments

- ompi – for MPI tasks
  - OpenMPI – both GNU and Sun Studio compiled
  - Up to 24 cores per task
  - Uses SysV IPC on single node and InfiniBand transport between nodes
- threads – for threaded tasks
  - OpenMP – Sun Studio only
  - POSIX Threads
  - Intel TBB
  - Up to 8 cores per task (single node only)
  - Intel Cluster OpenMP considered

Ganglia monitoring system

And here they come:

APPLICATIONS

Multicore applications on Physon

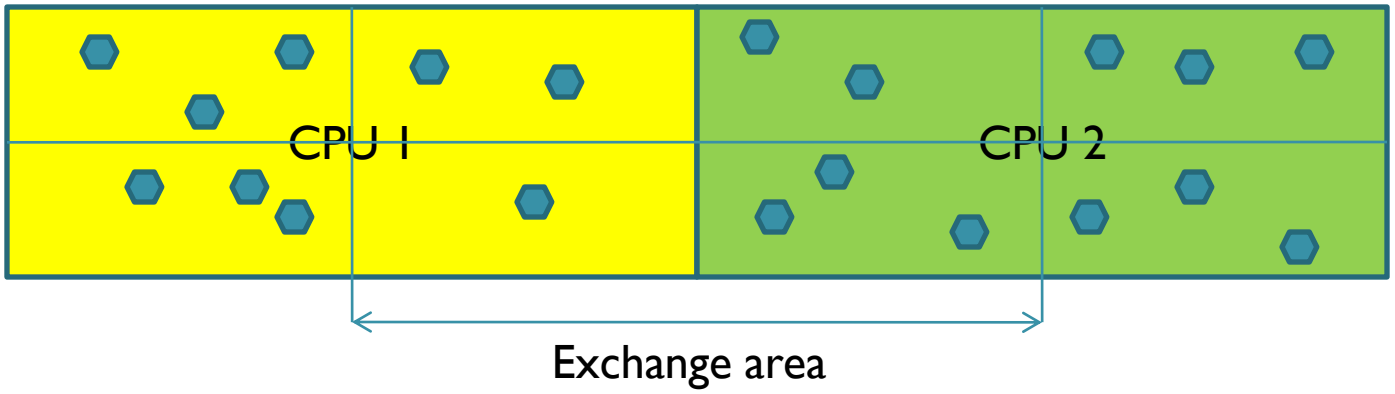
- Hybrid architecture – it is both shared and distributed memory machine
- Public domain codes:
  - GROMACS – molecular biology research
  - Quantum ESPRESSO – electronic structure and nanoscale simulations
- Custom codes
- Development tools and libraries:
  - Sun Studio 12 – C/C++/Fortran compilers and IDE, Sun Performance Library
  - GNU Compiler Collection – C, C++, FORTRAN77
  - OpenMPI

Molecular Dynamics

- Solves in discrete time steps equations of motion:

$$m_i \ddot{\mathbf{r}}_i = -\nabla_i U(\mathbf{r}_1, \ldots, \mathbf{r}_N)$$

- Force computation is the single time consuming operation – done in parallel using space decomposition techniques



Molecular Dynamics (2)

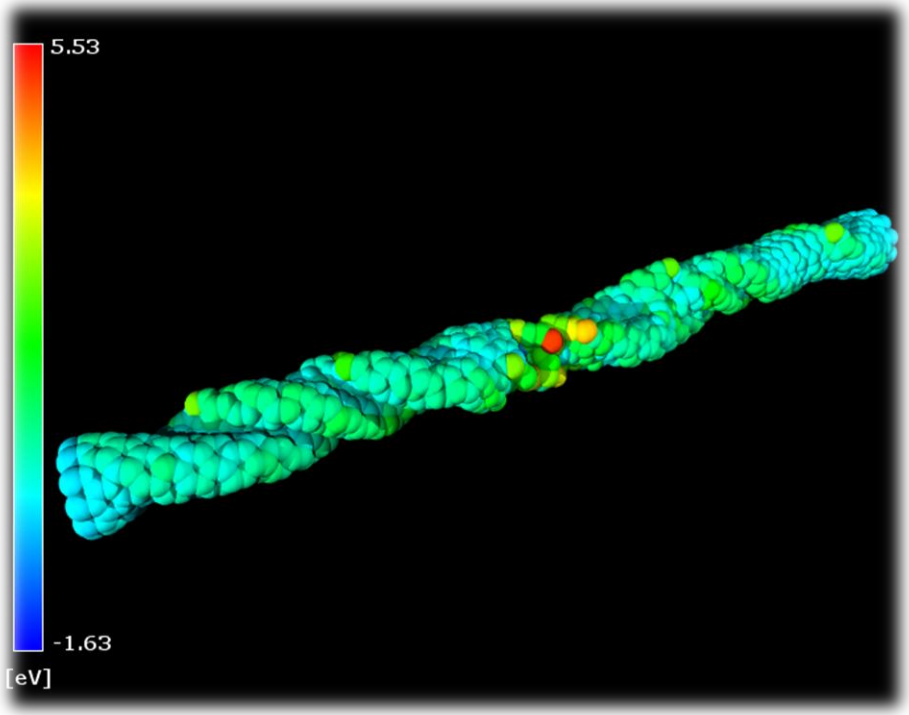
- Requires only local communication between neighboring CPUs for short ranged forces
- Asynchronous communication leads to good parallel scaling
- Long ranged forces require global (synchronous) communication – bad scaling ☹

Density Functional Theory

- A tool to compute/predict electronic structure and properties of molecules and crystals
- Minimizes $\langle \Psi | H | \Psi \rangle$ given a set of basis functions ϕ_i, usually planar waves or gaussians
- Essentially solves a large number of linear equations
- Requires vast amount of memory

Density Functional Theory (2)

- Can be combined with Molecular Dynamics – Car-Parrinello MD
- DFT parallelization:
  - Use parallel versions of basic linear algebra packages like BLAS and LAPACK
  - Distribute memory requirements between nodes
  - Global communication leads to bad parallel scaling

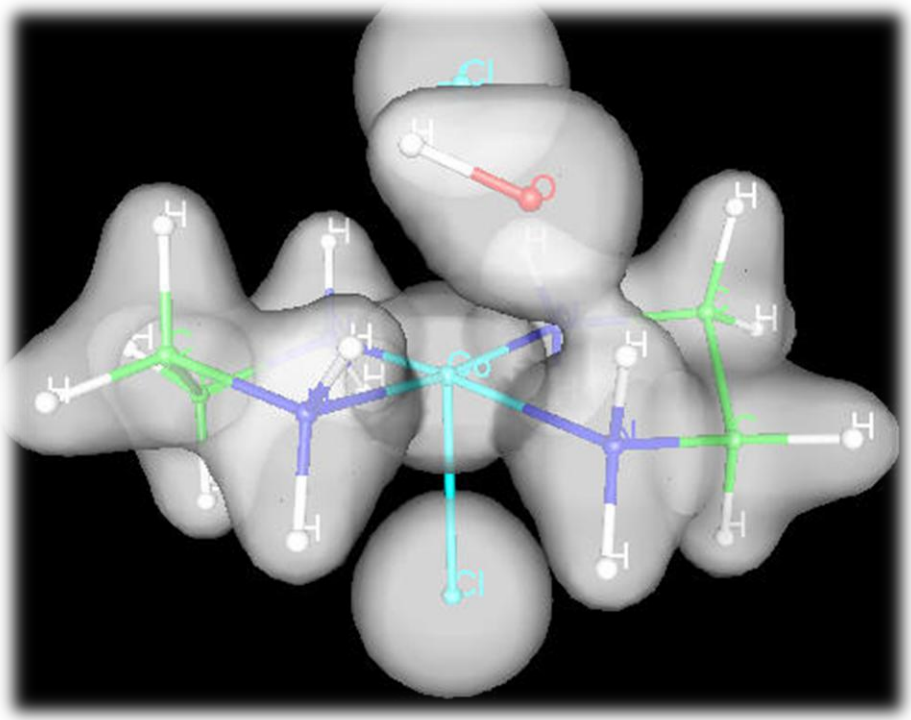


Buckling of carbon nanotube under rotational load – MD simulation using Brenner-Tersoff potential and custom code (BOZA).

Application: nanotechnology

BOZA

- Molecular Dynamics tool for simulations of carbon nanostructures
- Written in FORTRAN 77
- Parallelized using OpenMP
- Runs on shared memory computers
- Compiles as pure serial application on uniprocessors or with compilers that do not understand OpenMP



Electron density for cis-$[Co(en)_2Cl_2]$

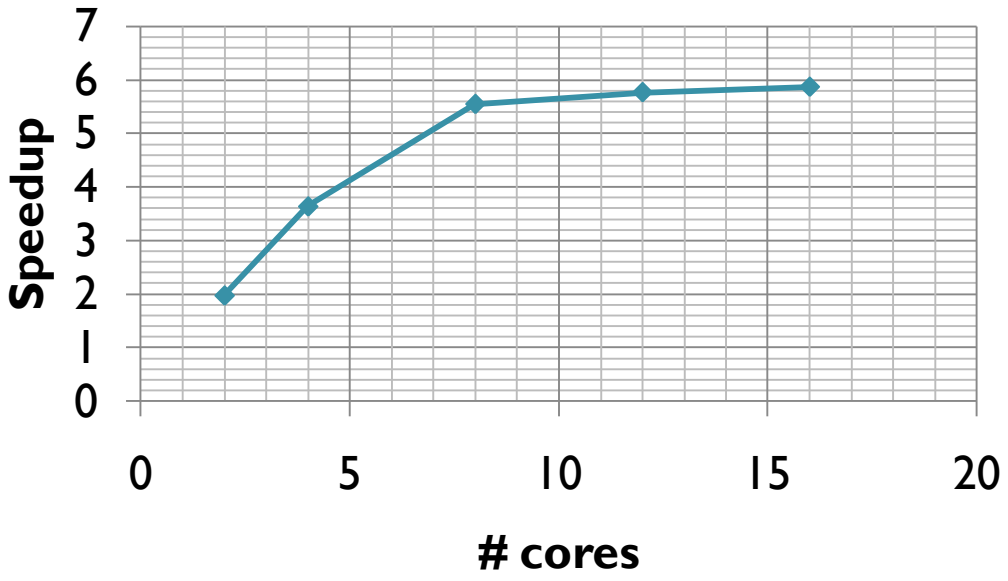
Application: nanotechnology

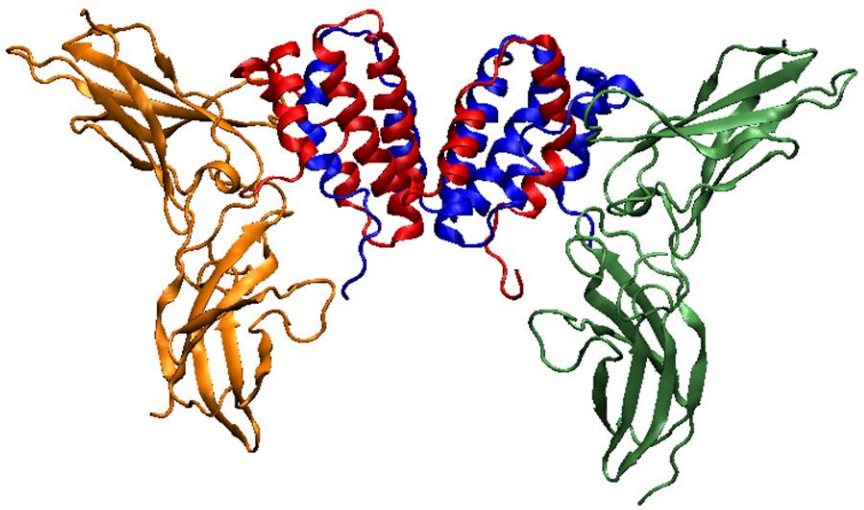
Quantum ESPRESSO

- Swiss army knife for quantum simulations
  - Electronic structure
  - Geometry optimization
  - Car-Parrinello molecular dynamics
- Written in Fortran 90/95
- Parallelized using MPI library calls
- Runs on both distributed memory (clusters) and shared memory machines

QE: performance

# of cores	Speedup	Parallel efficiency (speedup per core)
2	1,98	99,2%
4	3,64	91,1%
8	5,55	69,3%
12	5,77	48,1%
16	5,87	36,7%





Interferon-gamma alpha-receptor complex studied with MD simulation using GROMACS

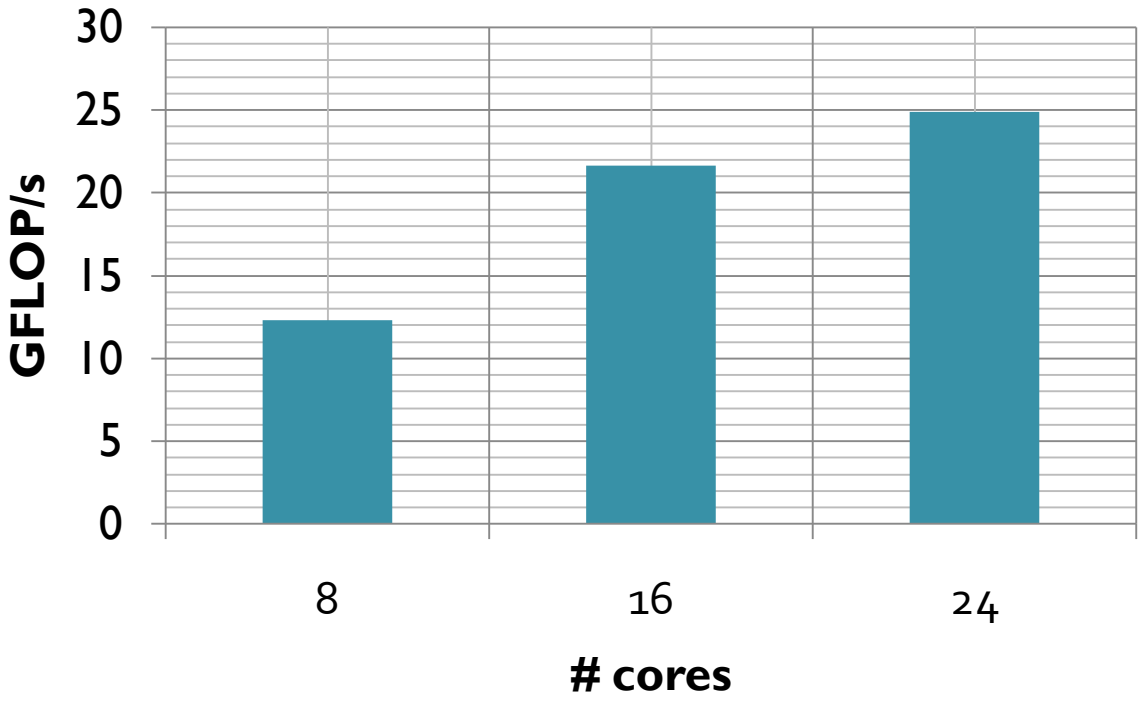
Application: drug design

GROMACS

- Molecular Dynamics tool for simulations in molecular biology and biochemistry
- Written in C
- Parallelized using MPI library calls
- Many built-in force fields
- Runs on both distributed memory (clusters) and shared memory machines

GROMACS: performance

# of cores	Computation speed	Comp. speed per core
8	12,30 GFLOP/s	1,54 GFLOP/s
16	21,65 GFLOP/s	1,35 GFLOP/s
24	24,91 GFLOP/s	1,04 GFLOP/s



Propaganda:

CHALLENGES

Challenges

- Parallel programming still not popular at Sofia University
- Most people still using moderate PCs to do research
- Masters course "Parallel programming and applications":
  - Already taught for two years (MPI only)
  - MPI and OpenMP basics
  - Performance measurement and optimization
  - Using batch processing engines and HPC
- Demands for computational power are growing!

And thus it ends.

THANK YOU FOR YOUR ATTENTION!

hristo@phys.uni-sofia.bg