MONT-BLANC

http://www.montblanc-project.eu

Pharmacelera™

Enric Gibert, Enric Herrero, Albert Herrero

December 11th 2015

This project and the research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 288777.

Pharmacelera in 5 Minutes

- Hardware and software solutions for scientific computations

- The team is a combination of complementary profiles
  - Management = 10+ years of management at Intel, 15+ years in entrepreneurship
  - Technical = 13+ years of software & hardware development at Intel, 33 patents, 26 technical publications
  - Scientific = 20+ years in drug discovery, 300+ publications
  - Commercial = 20+ years in pharma industry

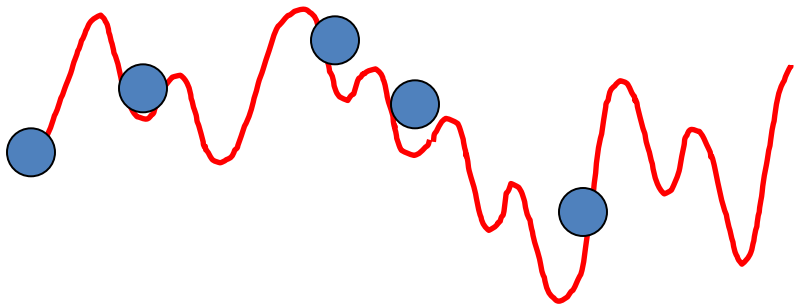
- 3 main activity areas
  - Integrated hardware / software solutions for drug discovery
  - Hardware acceleration services
    - **Established collaboration with Electronic and Atomic Protein Modelling (BSC)**
  - Preferred H2020 partner for applied research and innovation

- More information at www.pharmacelera.com (info@pharmacelera.com)

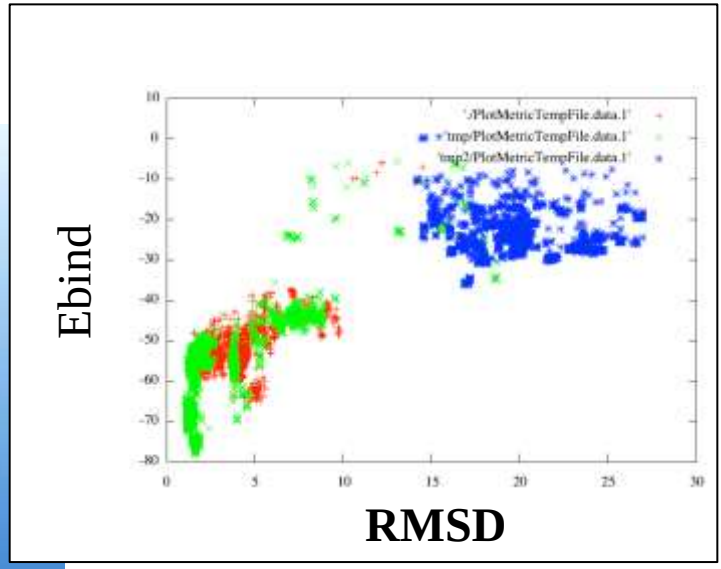
PELE (Protein Energy Landscape Exploration)

combining a Monte Carlo approach with protein structure prediction algorithms

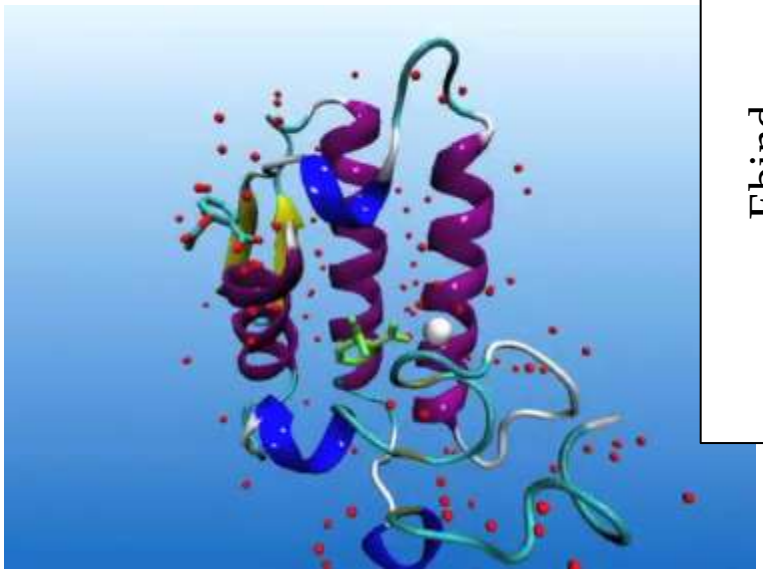
** Software developed by the Electronic and Atomic Protein Modelling group (Víctor Guallar, BSC)*



Provides a random walk along the Energy Landscape



Example of ligand (aspirin) search of the active site and binding. The crystallographic ligand is shown in green for comparison



Interaction Energy (Ebind) allows identifying the best bound minima (RMSD~1 Angstrom, analogous to the crystal)

Journal of Chemical Theory and Computation. 1 (6): NOV 2005

Tested application in Montblanc

- Molecular Solvent from PELE (molecular docking SW)
  - Primitive that accumulates 20%-30% of PELE's execution time
  - Molecular modeling for drug discovery
  - Programming language(s)
    - C, C++, OpenCL, …
    - ~5000 LOC
  - Parallel programming model(s)
    - None so far
  - Accelerator programming model(s)
    - OpenCL
  - Libraries / Dependencies
    - Boost, OpenCL, stdc++

Resources required

- Memory requirement per node: ~4MB
- Storage required
  - Small
- Minimum number of nodes / CPU
  - 1 CPU + GPU

Molecular Solvent – OpenCL Parallelization

2 different inputs:

- Large: 4357 atoms → 19M pairs
- Medium: 31313 atoms → 980M pairs

```
for each atom i {
  for each atom j {
    if (i != j) {
      compute interactions between atoms
    }
  }
  update alpha field of atom i
}
```

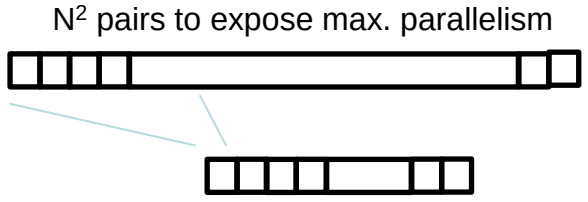
SCHEME 1: inner loop parallelization

```
for each atom i {
    for each atom j {
        if (i != j) {
            compute interactions between atoms
        }
    }
    update alpha field of i
}
```

SCHEME 2: outer loop parallelization

```
for each atom i {
    for each atom j {
        if (i != j) {
            compute interactions between atoms
        }
    }
    update alpha field of i
}
```

SCHEME 3: all pairs parallelization



Assuming no shared memory for code portability (OpenCL 1.X....)

- Create memory buffers and copy data

Evaluated Systems – Performance Only

AMD FirePro W5100 (high-end GPU 500€) = WORKSTATION

Memory
- 4GB GDDR5 memory, 128-bit memory interface, up to 96 GB/s memory BW

Compute Performance
- 768 stream processors (PEs) (12 compute units) → 64 PEs per CU
- 1.43 TFLOPS peak single precision FP performance
- 89.2 GFLOPS peak double precision FP performance

CPU
- Intel Xeon E3-1226 v3 (Haswell)

No shared memory between CPU and GPU

ARM MALI T-604 = MONTBLANC

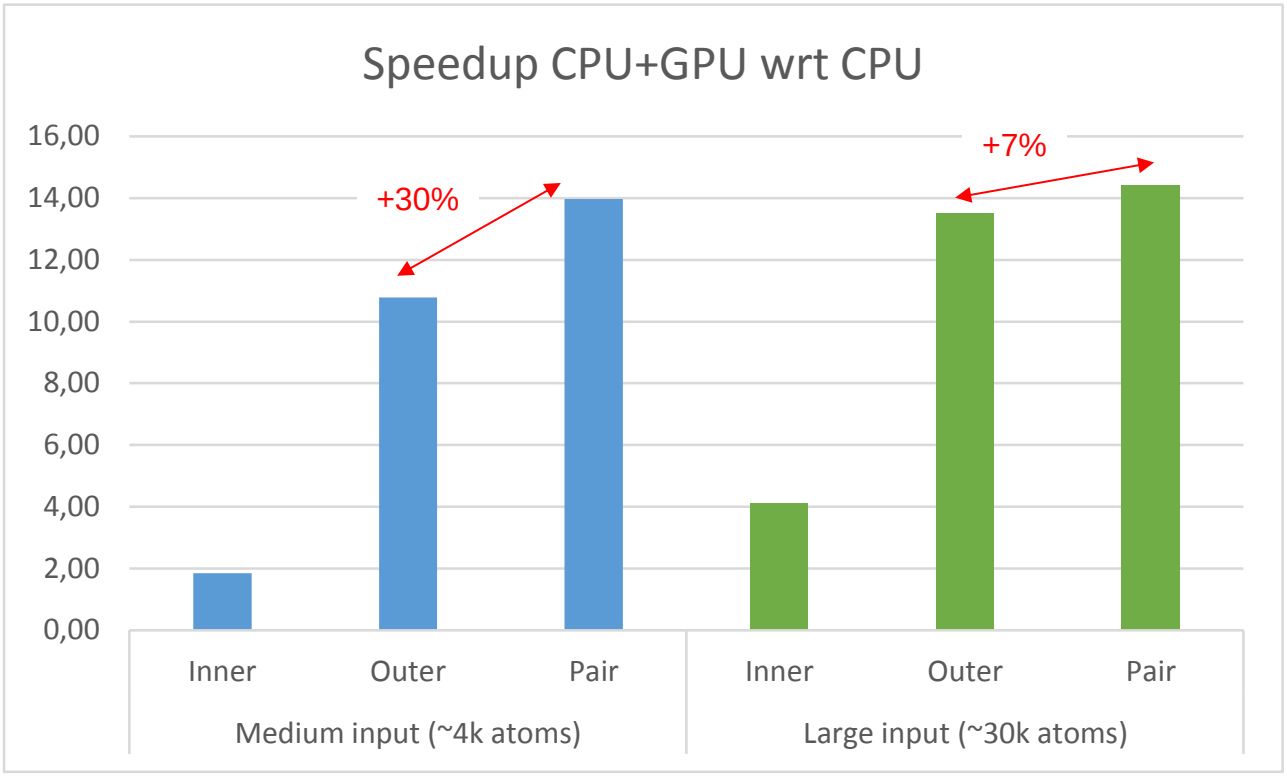
Memory
- 256KB L2

Compute Performance
- 8 ALUs (PEs) (4 compute units) → 2 PEs per CU
- 25.5 GFLOPS peak double precision FP performance (??)

CPU
- ARM v7

Shared memory between CPU and GPU → we have not made use of SVM!

Performance Results on Workstation

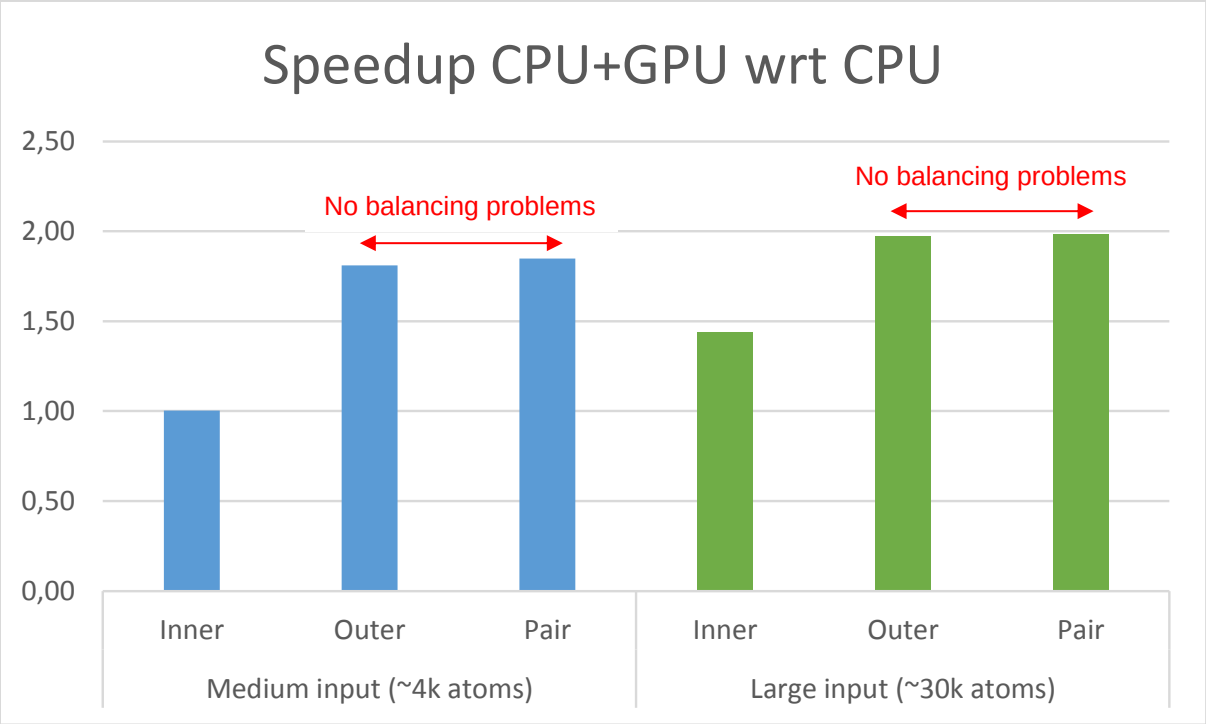


Baseline (CPU) execution times:

- Medium input = 1.13 seconds
- Large input = 52 seconds

Speedup severely impacted when work item group size is not max (256)

Performance Results on Montblanc



Baseline (CPU) execution times:

- Medium input = 4.30 seconds
- Large input = 226 seconds

Workstation vs Montblanc (Performance)

- Execution times in CPU
  - Workstation CPU between 3.8x and 4.3x better than Montblanc's
- Compiler role for CPU
  - Montblanc: 2.6x difference between –O3 and –O0
  - Workstation: 1.6x difference between –O3 and –O0
- Montblanc
  - Mobile GPU offers good performance (not outstanding)
  - Hidden support for shared memory in OpenCL 1.1 → not tested!
  - Lower amount of CEs → less prone to load unbalance

Feedback on platform usability

- Porting code: 1 hour effort!
  - Update paths: /apps/opencl_sdk/1.1.0/include, …
  - Change routine that checks SVM availability (OpenCL 2.0)
  - Add ‘–lm’ to Makefile
  - Change ‘AMD’ device by ‘ARM’ Device
  - Remove ‘-x spir’ from clBuildProgram
- Any advantage / disadvantage of using our platforms?
  - ☑ IT support
  - ☑ Availability
  - ☒ Apparently some nodes going down suddenly
- No additional tools have been used

Ongoing Steps

- Pharmacelera
  - Use CodeXL in workstation to perform detailed GPU characterization
  - Parallelize the execution of non-bonding energy computations using OpenCL
- Electronic and Atomic Protein Modelling (BSC)
  - Full parallelization of PELE
    - MPI for inter-core
    - OpenMP for intra-core
  - Full characterization of MPI/OpenMP/OpenCL implementation