A Performance and Energy Evaluation of OpenCL-accelerated Molecular Docking

The 5th International Workshop on OpenCL - IWOCL 2017
Toronto, Canada
May 17th 2017

Leonardo Solis-Vasquez Andreas Koch

Technische Universität Darmstadt

Embedded Systems & Applications

Outline

1. Molecular docking

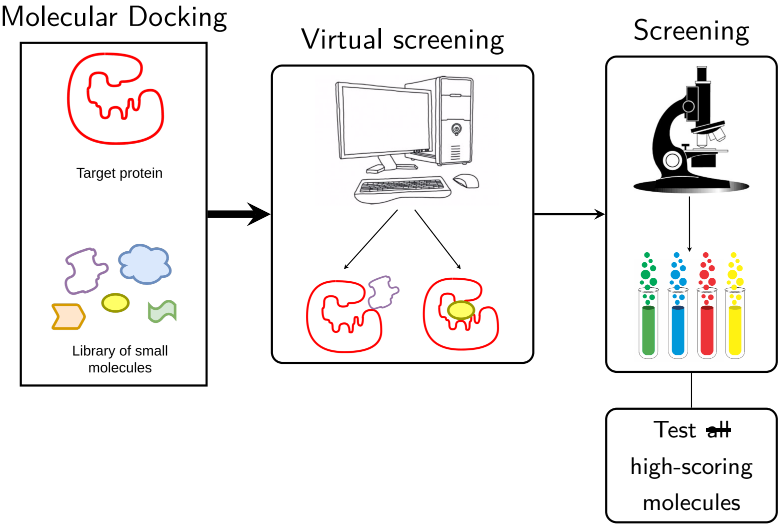
2. Parallel implementation

3. Evaluation

4. Concluding remarks

Molecular docking

"Predicting the best ways two molecules will interact"



Molecular Docking

Target protein

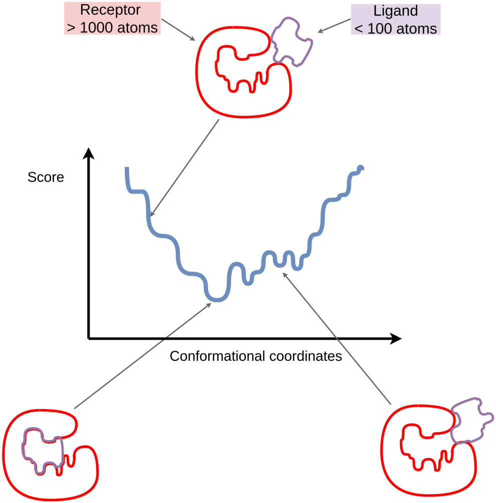
Library of small molecules

Virtual screening

Screening

Test ~~all~~ high-scoring molecules

Key aspects of docking



- Conformational coordinates
  - Position, orientation, torsion
- Scoring function
  - Predicting the energy of a particular pose
  - Lower score is better
  - Trade-off: speed vs. accuracy
- Search methods
  - Finding an optimal pose
  - Which search methods should be used?

Docking software: AutoDock 4.2[1]

- Based on a *Lamarckian Genetic Algorithm (LGA)*
- Binding positions are treated as entities of a population
- Optimized search: global + local
  - Global: entities are generated through genetic operations: *crossover*, *mutation*, *selection*
  - Local: only for selected entities (typ. 6% of population), new entities are generated using small deviations
- Score assignment to entities (binding energy)

[1] http://autodock.scripps.edu/

AutoDock scoring function

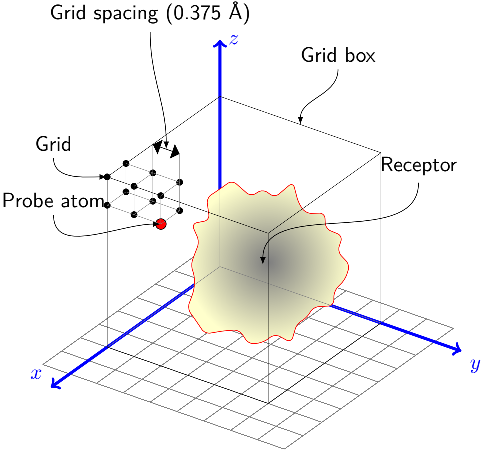
$$V = \overbrace{W_{vdw} \sum_{i,j} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}}\right)}^{\textit{Lennard-Jones}} + \overbrace{W_{hbond} \sum_{i,j} E(t) \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}}\right)}^{\textit{Hydrogen bonding}} +$$

$$\underbrace{W_{elec} \sum_{i,j} \left(\frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}}\right)}_{\textit{Coulomb's law}} + \underbrace{W_{sol} \sum_{i,j} (S_i V_j + S_j V_i) e^{\frac{-r_{ij}^2}{2\sigma^2}}}_{\textit{Desolvation}}$$

- Atom indexes: i and j
- Molecule size (# atoms): receptor >1000, ligand <100

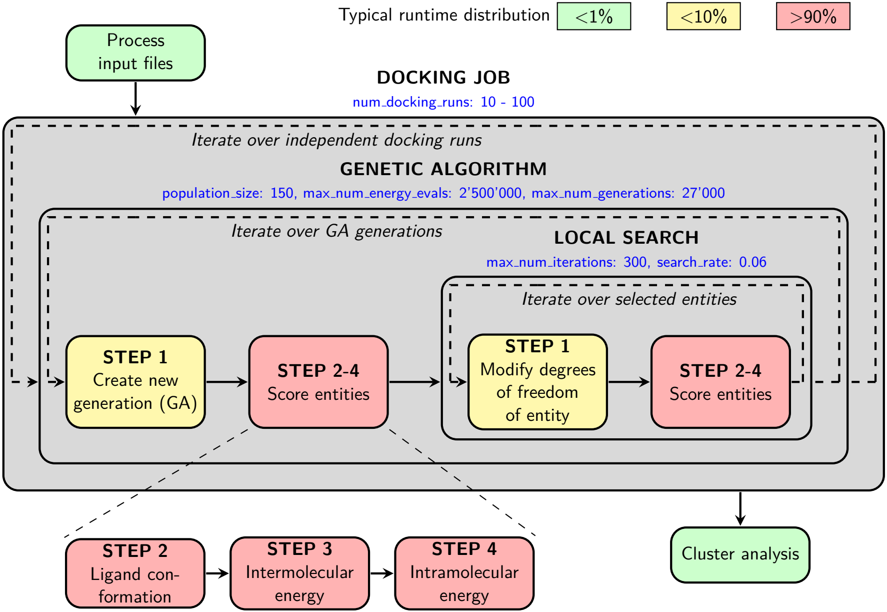
- Physics-based approach from molecular mechanics
- Energy of molecular binding (Kcal mol^{-1})
- Calibrated with 188 complexes

AutoDock grid maps



- Calculates the intermolecular energy
- Precomputes interactions for each type of ligand atom
- Faster (∼100x) than pairwise methods
- Drawback: receptor is rigid, limits search space

AutoDock implementation



Typical runtime distribution

<1%	<10%	>90%

Process input files

DOCKING JOB

num_docking_runs: 10 - 100

Iterate over independent docking runs

GENETIC ALGORITHM

population_size: 150, max_num_energy_evals: 2'500'000, max_num_generations: 27'000

Iterate over GA generations

STEP 1 Create new generation (GA)

STEP 2-4 Score entities

LOCAL SEARCH

max_num_iterations: 300, search_rate: 0.06

Iterate over selected entities

STEP 1 Modify degrees of freedom of entity

STEP 2-4 Score entities

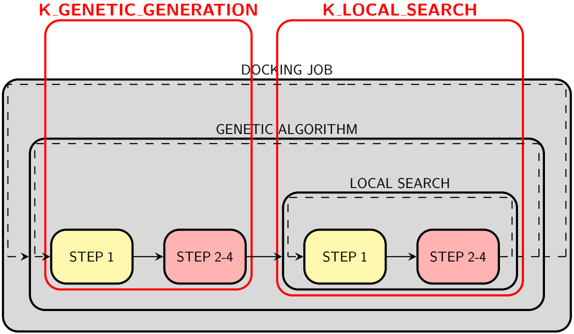
STEP 2 Ligand conformation

STEP 3 Intermolecular energy

STEP 4 Intramolecular energy

Cluster analysis

Parallelism opportunities



- Based on a CUDA reference[2]
- Exploit more parallelism by merging two outer loops
- Multiple entities are distributed into work-groups
- Grid calculation and intramolecular energy (STEP 2-4) are processed by work-items

[2] Pechan et al. *"Molecular Docking on FPGA and GPU platforms"*

Contribution of this work I

- Porting from CUDA to OpenCL
- Further optimizations
  - Enable the size configuration of processing elements
    - Tested with 16, 32, 64, 128 work-items
  - Usage of OpenCL native functions
    - Lower-accuracy arithmetic does not decrease the docking quality

Built-in single precision math function	Minimum accuracy in ULP (Unit in the Last Place)		
	Full	Half	Native
sin	$\leqslant 4$	$\leqslant 8192$[3]	Implementation-defined
cos	$\leqslant 4$	$\leqslant 8192$	Implementation-defined
divide	$\leqslant 2.5$	$\leqslant 8192$	Implementation-defined
sqrt	$\leqslant 4$	$\leqslant 8192$	Implementation-defined
powr	$\leqslant 16$	$\leqslant 8192$	Implementation-defined
exp	$\leqslant 3$	$\leqslant 8192$	Implementation-defined

[3] Minimum 11 bits of accuracy, $\leqslant 8192$ ULP

Contribution of this work II

- ... Further optimizations
  - Optimization of grid calculation
    - ★ Elimination of redundant terms, better grouping of sub-expressions
    - ★ Number of multiplications was reduced: 24 down to 5

Original:

$$GetGrid(gd,sz_x,sz_y,sz_z,atomtype,z,y,x) {=} {*}(gd+sz_x*(y+sz_y*(z+sz_z*atomtype))+x) \quad \color{red}{(3\,mult)}$$

$$\left.\begin{array}{l} cube_{000} = GetGrid(gd, sz_x, sz_y, sz_z, atomtype, z_{low}, y_{low}, x_{low}) \\ \dots \\ cube_{111} = GetGrid(gd, sz_x, sz_y, sz_z, atomtype, z_{high}, y_{high}, x_{high}) \end{array}\right\} \color{red}{(8\,equations,\ 24\,mult\,in\,total)}$$

Optimized:

$$g1{=}sz_x, \quad g2{=}sz_x*sz_y, \quad g3{=}sz_x*sz_y*sz_z \quad \color{red}{(Moved\,out\,of\,the\,parallel\,region)}$$

$$ylg1{=}y_{low}*g1, \quad yhg1{=}y_{high}*g1, \quad zlg2{=}z_{low}*g2, \quad yhg2{=}z_{high}*g2, \quad m{=}atomtype*g3 \quad \color{red}{(5\,mult)}$$

$$\left.\begin{array}{l} c_{000} = x_{low} + ylg1 + zlg2 \\ \dots \\ c_{111} = x_{high} + yhg1 + zhg2 \end{array}\right\} \rightarrow \left.\begin{array}{l} cube_{000} = *(gd + c_{000} + m) \\ \dots \\ cube_{111} = *(gd + c_{111} + m) \end{array}\right\} \color{red}{(8\,equations,\ 5\,mult\,in\,total)}$$

Contribution of this work III

- ... Further optimizations
  - Minimization of host-device communication using memory mapping
    - ★ Docking progress is monitored by host on each generation cycle
    - ★ More docking runs, larger the device-to host copy latency

```
docking_job {
 while (progress(evals_of_runs, num_generations) < 100%) {
  K_GENETIC_GENERATION();
  K_LOCAL_SEARCH();
  evals_of_runs = clEnqueueMapBuffer(size_evals_of_runs);
  num_generations++;
 }
}
```

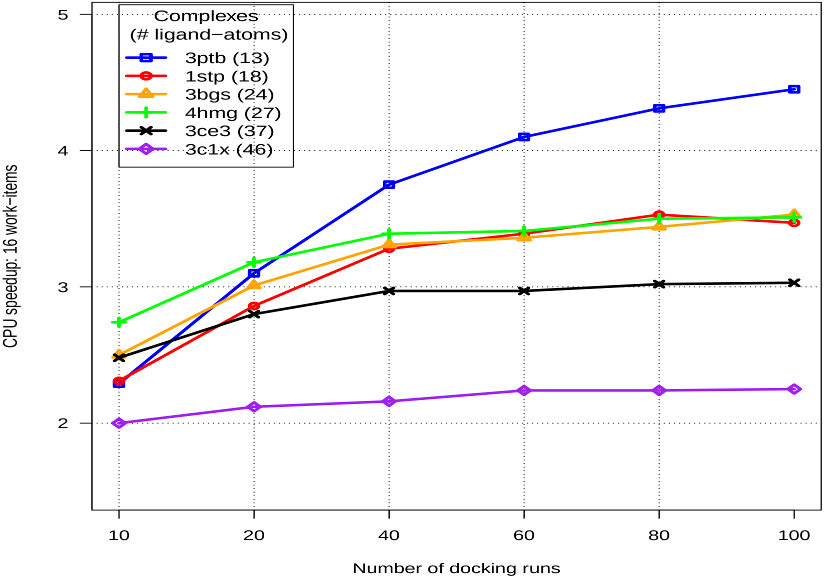
- Evaluation of energy consumption on CPU/GPU platforms
  - For both sequential baseline and accelerated versions

Test description

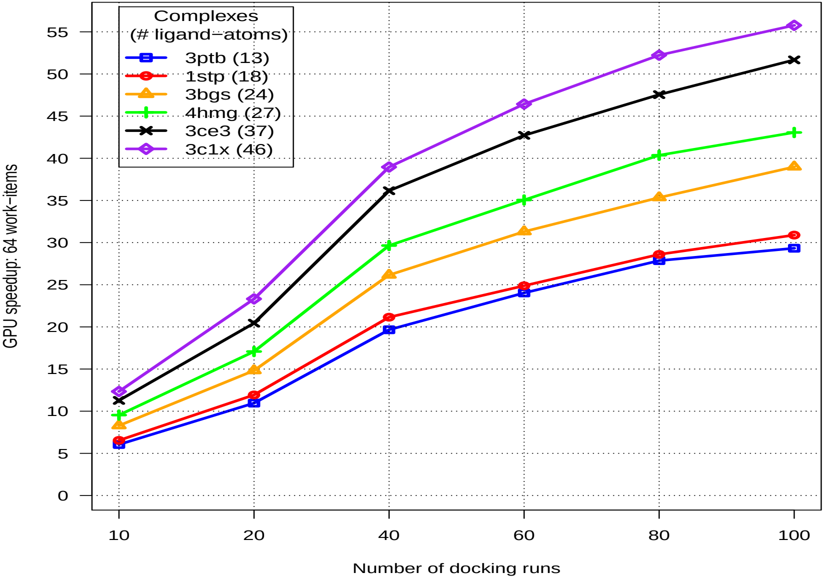
- Typical AutoDock LGA configuration
  - Number of runs, population size, etc
- Redocking experiment
  - Recovering the structure of a known complex and its interaction
  - Comparison between reference solution and our accelerated implementation
- Criteria for functional correctness
  - Metrics: binding energy, spatial deviation, size of best cluster
    - ⋆ Δ(binding energy) $\leqslant$ 1 Kcal mol^{-1}
    - ⋆ Spatial deviation $\leqslant$ 2 Å
    - ⋆ Minimum best cluster size $\geqslant$ 25% (# runs)
- Total of twenty ligand-receptor PDB[4]complexes
- Target system
  - CPU: i5-6600K (4 cores) @3.5GHz
    - ⋆ A CPU core is used as sequential baseline
  - GPU: AMD R9-290X (2816 multiprocessors) @1GHz

[4] Protein Data Bank: http://www.rcsb.org/pdb

Speedup (execution time): CPU



Speedup (execution time): GPU



Results summary: speedup

- Complete program execution is measured
  - Input and output require less than 1% of total execution time

- Results for 100 docking runs
  - Number of work-items[5]: CPU: 16, GPU: 64

PDB	Execution time (s)			Speedup	
complex	Baseline	Par. CPU	Par. GPU	CPU	GPU
3ptb	586.27	131.77	19.99	4.45	29.33
1stp	836.47	241.06	27.08	3.17	30.89
3bgs	1102.88	312.20	28.29	3.53	38.98
4hmg	1416.22	403.12	32.89	3.51	43.06
3ce3	1867.69	617.00	36.15	3.03	51.67
3c1x	2841.84	1265.72	50.96	2.25	55.77

- Geometric mean of speedup on 20 ligand-receptor complexes
  - CPU: ∼3.3x, GPU: ∼40.4x

[5] Best-speedup configuration determined experimentally

Results summary: computing-platform energy

- Power measured using performance counters
  - Sampling interval[6] of 50 ms
  - Power samples are integrated over time to obtain energy

- Results for 100 docking runs
  - Number of work-items: CPU: 16, GPU: 64

PDB	Energy consumption (KJ)			Efficiency gain	
complex	Baseline	Par. CPU	Par. GPU	CPU	GPU
3ptb	11.80	5.95	2.39	1.98	4.92
1stp	16.69	11.72	3.74	1.42	4.47
3bgs	21.56	15.13	4.16	1.43	5.18
4hmg	28.07	19.43	4.81	1.44	5.84
3ce3	36.27	30.39	5.84	1.19	6.21
3c1x	54.85	61.15	8.72	0.89	6.29

- Geometric mean of energy efficiency gain on 20 ligand-receptor complexes
  - CPU: ∼1.4x, GPU: ∼5.4x

[6] Shortest practical interval supported by measurement tools

Concluding remarks

- Portable docking implementation
- Achieved functional correctness
  - Binding energy
  - Spatial deviation
  - Size of best cluster
- Achieved performance gains
  - Max. speedup: 4x (CPU) and 56x (GPU)
  - Max. energy efficiency: 2x (CPU) and 6x (GPU)

A Performance and Energy Evaluation of OpenCL-accelerated Molecular Docking

Leonardo Solis-Vasquez

solis@esa.tu-darmstadt.de

https://www.esa.cs.tu-darmstadt.de/