

FastGrid

The Accelerated AutoGrid Potential Maps Generation for
Molecular Docking

Marek Olšák, Jiří Filipovič, Martin Prokop

MEMICS'09

Content of the speech

- molecular docking
- GPU architecture
- electrostatic potential evaluation
- cutoff potentials evaluation
- performance
- conclusion

Molecular docking

Molecular docking is a method for prediction of position and orientation of one molecule to another when bound to each other to form a stable complex.

- the stable complex means the potential energy of molecules is minimal
- protein-ligand docking
 - the area of our interest
 - the specific category of molecular docking
 - important in computer-aided drug design
 - small molecule of ligand (putative drug) is docked into the binding site of a protein

Molecular docking

From computational point of view

- the minimum is searched by traversing the state space of spatial configurations of the molecules
- during the state space search, the energy of the protein-ligand complex is needed for each visited state
- the potential energy in the protein vicinity can be precomputed
 - the potential energy evaluation of large molecule is performed only once
 - the complex energy evaluation depends only on the size of the ligand
 - yields two computationally demanding parts of computation
 - precomputation of potential energy
 - searching the state space

GPU

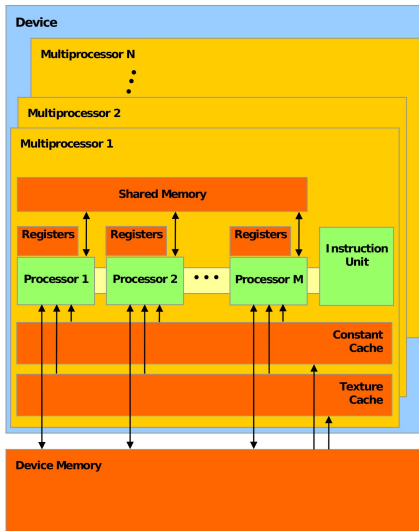
Graphics processing unit (GPU) is a many-core processor primarily targeted to accelerate graphics rendering

- tens of multiprocessors
- each multiprocessor consists of multiple cores performing the same instruction in time
- modern GPUs are programmable enough to perform general processing tasks

GPUs are powerful, but the programming model is difficult

- an order of magnitude higher arithmetic power and memory bandwidth
- SIMT instruction execution model
- in general no cache coherency, explicitly-hierarchical memory
- thousands of threads are necessary to fully exploit GPU power

GPU architecture



Precomputation of potential energy

We focus to the precomputational part

- the potential energy is evaluated for all points in a 3D lattice
- multiple maps for different energies and atoms groups are generated

Two types of energies

- long-range electrostatic energy, evaluated for all receptor's atoms in each lattice point
- short-range energies, evaluated only for near atoms (done by cutoff value)

Our implementation

Our work is based on the model used in AutoDock suite

- one of the most popular and the most cited docking software
- consists of AutoGrid (precomputation) and AutoDock (state space searching) programs
- we have accelerated the AutoGrid implementation

Hybrid algorithm

- the long-range energies evaluation is very regular and arithmetic-intensive, thus suitable for GPU acceleration
- the short-range energies evaluation has lower time complexity, thus can be computed on slower CPU in parallel

AutoGrid main loop

```
for all points of the grid do
  for all atoms of molecule do
    compute electrostatic potential
    if distance between point and atom < cutoff value then
      compute desolvation potential
      compute van der Waals potential
      compute hydrogen bonds potential
    end if
  end for
end for
```

GPU acceleration of electrostatic energy

Direct Coulomb summation using constant dielectric already published¹

- fits well to GPU architecture
- several optimizations introduced (outer loop unrolling, atoms in constant memory)

Extension of this algorithm

- AutoGrid uses distance-dependent dielectric
- using the coarser lattice yields poor performance

¹John E. Stone, James C. Phillips, Peter L. Freddolino, David J. Hardy, Leonardo G. Trabuco, Klaus Schulten. Accelerating molecular modeling applications with graphics processors. In *Journal of Computational Chemistry*, Volume 28 Issue 16, p 2618, 2007.

Algorithm modifications

Scaling for coarser lattices

- can be improved by reducing unrolling and improving independent parallelism performing more redundant computations

The distance dependent dielectric (DDD) can be precomputed into the table or computed in each atoms evaluation

- the recomputation in all atoms evaluations yields redundant work in particular threads
- the access into the table is not regular thus can yield performance issues on GPU
 - the global, constant or texture memory can be used

Performance of DDD implementations

Global memory

- not effective because of irregular access
- computationally expensive addressing

Recomputation

- most effective for smaller grids (does not exploit caches well)

Constant memory

- better than global
- the broadcast of constants is restricted
- addressing is computationally expensive
- cache shared with atoms data

Texture memory

- good addressing
- good cache locality
- most effective for larger grids

Short-range energies

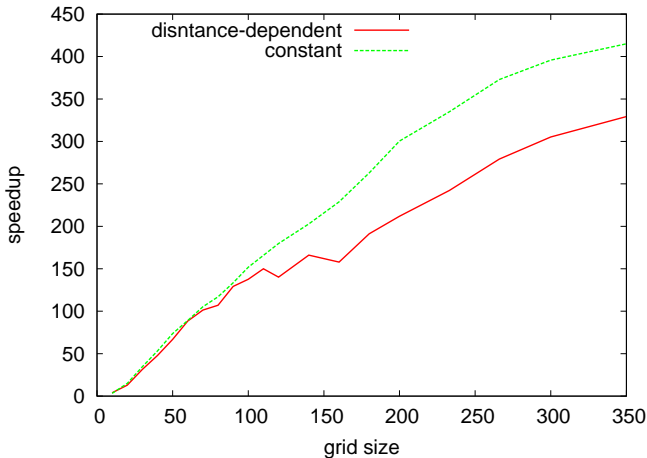
The near atoms need to be found quickly

- we cannot use the distance from electrostatic energy computation
- the recomputation yields $\mathcal{O}(n)$ instead of $\mathcal{O}(1)$ in each atom evaluation
- we can build the coarse grid
 - each cell contains intersecting atoms
- in energies evaluation
 - the cells in the area where energies are computed are selected
 - the atoms in these cells can be closer than cutoff

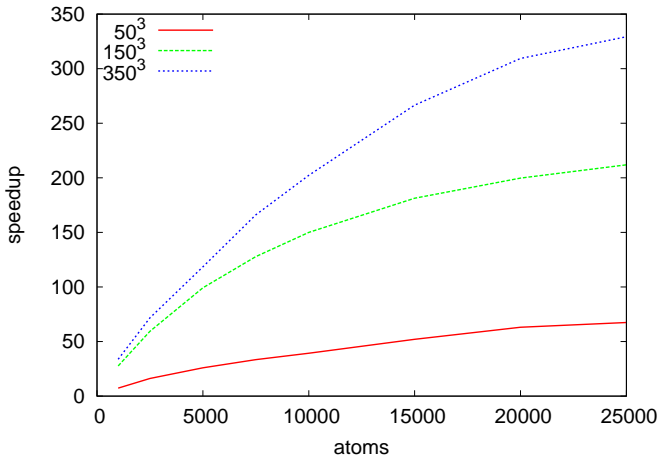
The nearest hydrogen needs to be found quickly

- the k-NN is used

Speedup over AutoGrid

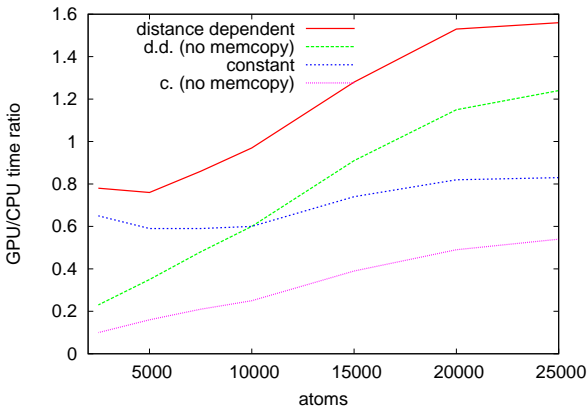


Speedup over AutoGrid



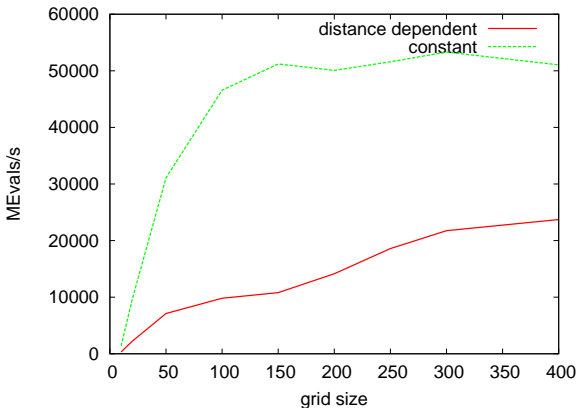
Performance analysis

Despite better time complexity, the CPU algorithm can still be a bottleneck.



Performance analysis

The constant dielectric algorithm scales better because no texture cache utilization issues.



Conclusion

- the acceleration of AutoGrid yielding two orders of magnitude speedup has been presented
- the architecture is hybrid
 - novel extension of GPU electrostatic potential evaluation algorithm using distance-dependent dielectric
 - CPU implementation independent on output of GPU algorithm
- the evaluation of the performance allows us to deeper understand presented algorithms scaling