MURCIA: fast parallel solvent accessible surface area calculation on GPUs and application to drug discovery and molecular visualization

E.J. Cepas Quiñonero[1], H. Pérez-Sánchez[1], W. Wenzel[2], J.M. Cecilia[1], J.M. García[1]

[1]Computer Engineering and Technology Department, University of Murcia, Spain;
[2]Institute of Nanotechnology, Karlsruhe Institute of Technology, Germany

SUMMARY

Bioinformatics methods can considerably aid clinical research, providing very useful insights and predictions working at the molecular level in fields like Virtual Screening, Molecular Dynamics and Protein Folding. Nevertheless, such methods present computational bottlenecks whenever they deal with accurate biophysical models. Such a case is the calculation of the solvent accessible surface area (SASA). We show a novel method called MURCIA that exploits last generation massively parallel Graphics Processing Unit (GPU) hardware to considerably speedup SASA calculations, being at the moment one of the fastest methods in the range 10 – 17000 atoms and which also provides information for the visualization of molecular surfaces in standard molecular graphics program like VMD, Pymol and Chimera.

INTRODUCTION

It is very important in clinical research to determine the safety and effectiveness of current drugs and to accelerate findings in basic research (discovery of new leads and active compounds) into meaningful health outcomes. Both objectives imply to be able to process the vast amount of protein structure data available in biological databases like PDB and also derived from genomic data using techniques as homology modelling (Sánchez 1998). Screenings in lab and compound optimization are expensive and slow methods, but bioinformatics can vastly help clinical research for the mentioned purposes by providing prediction of the toxicity of drugs and activity in non-tested targets, by evolving discovered active compounds into drugs for the clinical trials. All this can be done thanks to the availability of bioinformatics tools and Virtual Screening (VS) methods that allow to test all required hypothesis before clinical trials. Nevertheless, VS methods fail to make good toxicity and activity predictions since they are constrained by the access to computational resources; even the nowadays fastest VS methods cannot process large biological databases in a reasonable time-frame. This imposes, thus a serious limitation in many areas of translational research. We have previously studied how exploitation of last generation massively parallel hardware architectures like GPUs can tremendously overcome this problem accelerating the required calculations and allowing the introduction of improvements in the biophysical models not affordable in the past (Guerrero 2011, Pérez-Sánchez 2011). Between the most relevant computationally intensive kernels present in current VS methods, we may highlight the calculation of the molecular surface in terms of the solvent accessible surface area (SASA). We can model efficiently solvation in an implicit way by the calculation of SASA and posterior consideration of the hydrophobic and hydrophilic character of individual atoms (Eisenberg 1986), being this method widely applied nowadays in protein structure prediction and protein-ligand binding. There have been several efforts to develop a fast method for the SASA calculation.

To the best of our knowledge, the fastest method nowadays is POWERSASA (Klenin 2011). Its running time depends linearly on the number of atoms of the molecule. We propose a new method called MURCIA (Molecular Unburied Rapid Calculation of Individual Areas) that uses the GPU as underlying hardware and which runs around 15 times faster than POWERSASA for the usual proteins that we found in most VS methods, with less than 25000 atoms. Another advantage of MURCIA is that it can rapidly provide molecular surface information useful for fast visualization in several molecular graphics programs.

METHODS

SASA calculation using atomic grid

All atoms of the molecule are specified by their centers and SASA radii, which depend on their Van der Waals radius, and therefore on their atomic type, plus the water molecule radius. MURCIA calculates individual SASA values through the next three Kernels:

- GenGrid: we build a grid of points around each atom

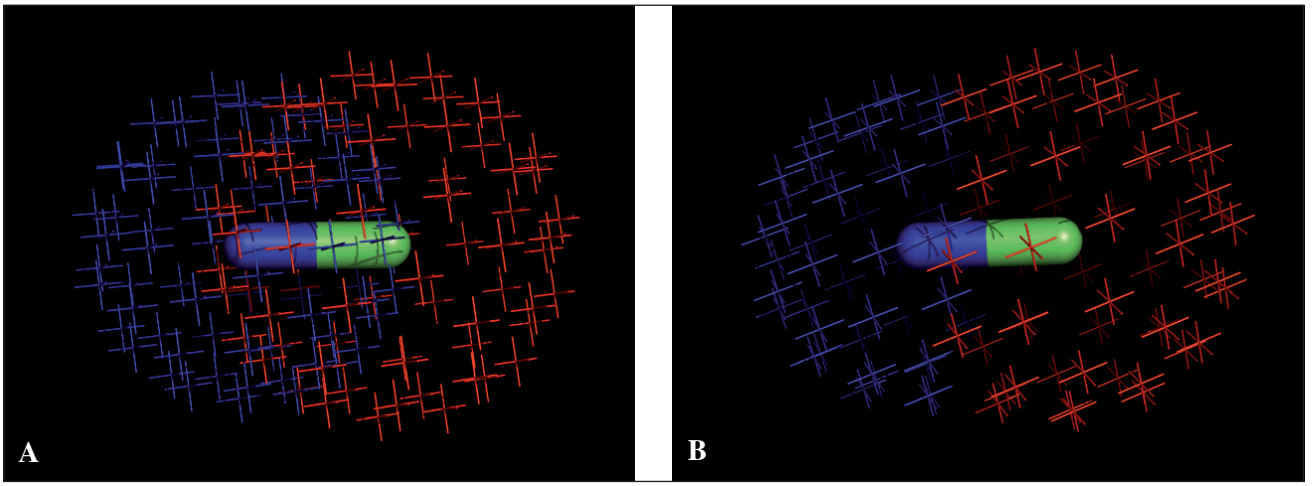

Figure 1 - *Atomic grids for a molecule with two atoms (A) both grids overlap, situation previous to the SASA calculation (B) only non-buried grid points are shown.*

following the procedure developed by Lebedev et al. (Lebedev 1999) for the numerical integration over a sphere. This grid guarantees a high precision in integrations, using a very low number of grid points over an unit sphere. In our case we use 72 points. An example of the grid is shown in *Figure 1.A*.

- Neighbours: we calculate the list of its closest neighbours for each atom. The distance threshold is equal two times the highest value of the highest SASA radii. Atoms are sorted in the lists starting from the closest ones.
- Out_points: as depicted in *Figure 2* for each atom *i,,* we perform the following calculation for each grid point *k*; we calculate squared distance to the first neighbour atom *j* of the list. If this distance is smaller than the SASA radius of atom *j*, then we flag this grid point as buried. Otherwise we continue the same procedure calculating distances versus the other atoms of the list. If the grid point *k* is not eventually flagged as buried, then it is stored as contributor to SASA for atom *i*. Once this procedure is finished for all grid points of atom *i*, we will have *n* non-buried grid points, and individual SASA for this atom will be calculated according to a *(n/72)* fraction of the sphere surface of radius corresponding to the SASA radius of this atom. At the same time, all coordinates of non-buried grid points are stored for posterior molecular visualization. The same procedure is applied to all atoms *i* of the molecule. An example of the resulting non-buried grid points is shown in *Figure 1.B*.

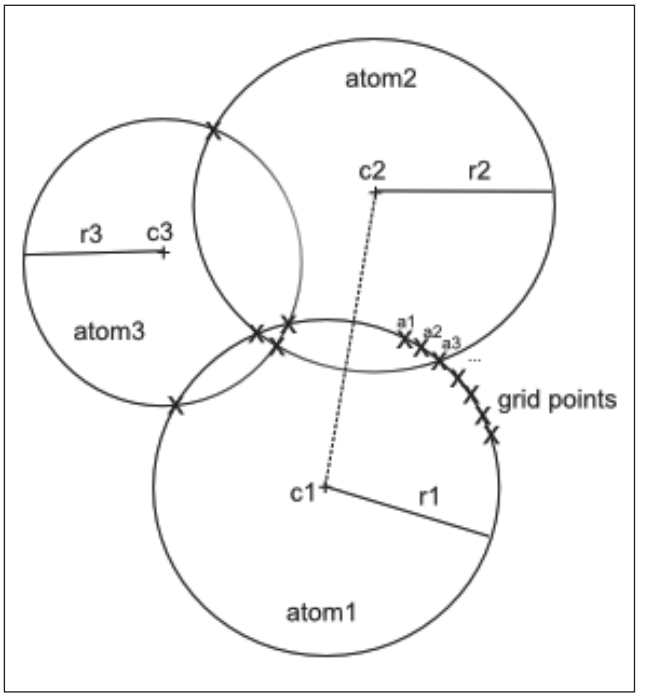

Figure 2 - *Depiction in 2D of the SASA calculation in MURCIA.*

GPU implementation

We used the version 4.0 of the CUDA programming model (NVIDIA 2011) in our parallel implementation with a NVIDIA Tesla C2050 GPU.

In order to obtain speedup measurements versus the sequential counterpart version, an Intel Xeon E5450 cluster was used.

This model allows writing parallel programs for GPUs using extensions of the C language. We describe here how the previous kernels are implemented on the GPU:

- GenGrid: It generates atomic grids from the molecular input file. It divides the number of calculation for atoms into CUDA blocks, and assigns a number of threads per block proportional to the number of grid points per atom (72), so each thread computes only one grid point per atom.
- Neighbours: It creates one CUDA block per atom and a variable number of threads per block. Each thread computes for each atom *i* the distances to the other atoms *j*. All threads from a block cooperate together to calculate all its neighbours using CUDA

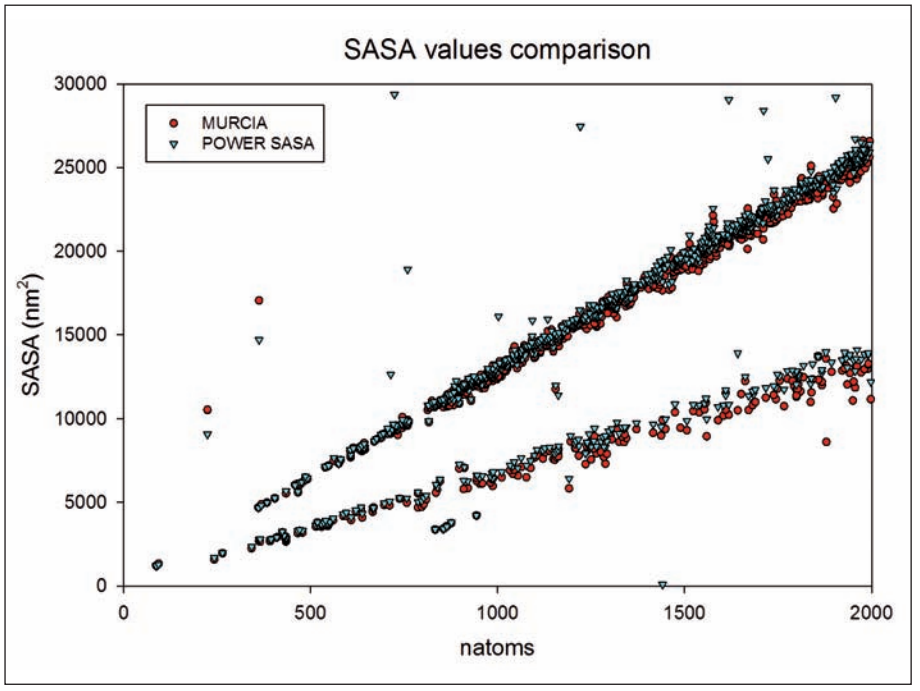

Figure 3 - *Comparison of the SASA values calculated by POWERSASA and MURCIA. A diverse set of the PDB database was used for the calculations.*

shared memory for storing variables commons to all threads of a block.

- Out_Points: It establishes the values of number of blocks and threads per block in the same way as the GenGrid kernel does. Each thread computes only distances between only one grid point and all of its neighbours.

RESULTS AND DISCUSSION

In order to check the accuracy of our method, we check MURCIA calculations with previous POWERSASA results (Klenin 2011). *Figure 3* shows an overall good concordance between both methods. POWERSASA uses a very accurate method for the calculation of SASA.

There are some cases where MURCIA deviates from the POWERSASA ones. We think this is due to the insufficient number of points (72) used for the atomic grids.

Figure 4 shows a performance comparison between MURCIA and POWERSASA. In the interval 10 to 17000 atoms, MURCIA runs faster than POWERSASA, achieving maximum speedups of 15X. For bigger molecules (20000-100000 atoms) POWERSASA runs faster than MURCIA. We have also checked that

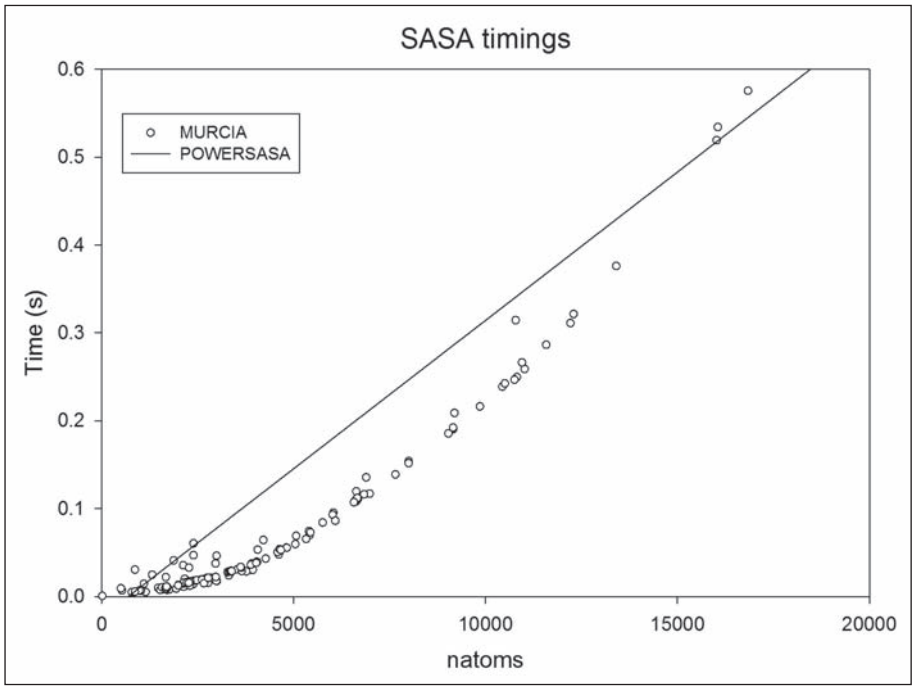

Figure 4 - *Comparison of timings for SASA calculation using MURCIA and POWERSASA (since its dependence with the number of atoms is linear and for the sake of clarity, a solid line is used to represent its timings results). A diverse set of the PDB database was used for the calculations.*

MURCIA runs around 30X times faster than MSMS (Sanner 1996). In conclusion, we have developed a fast and efficient method for the SASA calculation, implemented on GPU hardware, and which can also be used for fast visualization of molecular surfaces using information calculated for the non-buried atomic surfaces.

The method is not yet optimal and there are several improvements we are working on. First, we are checking how using more dense grids influences on the precision of the SASA calculation.

Second, the main bottleneck of the program resides in the calculation of neighbours; we are testing at the moment a better strategy, which calculates much faster the neighbour's list.

Third, we are testing in some molecular graphics programs (VMD, Chimera, Pymol) how this method speedups visualization of molecular surfaces. The program is available upon request.

ACKNOWLEDGEMENTS

This research was supported by the Fundación Séneca (Agencia Regional de Ciencia y Tecnología, Región de Murcia) under grants 00001/CS/2007 and 15290/PI/2010, by the Spanish MEC and European Commission FEDER under grants CSD2006-00046 and TIN2009-14475-C04 and a postdoctoral contract from the University of Murcia (30th December 2010 resolution). We also thank Centro de Supercomputación de la Fundación Parque Científico de Murcia for providing supercomputing time.

REFERENCES

[1] Eisenberg D, McLachlan AD. Solvation energy in protein folding and binding. Nature 1986; 319: 199-203.

[2] Guerrero GD, Pérez-Sánchez HE, Wenzel W, Cecilia and García JM. Effective parallelization of non-bonded interactions kernel for virtual screening on GPUs, Proc. 5th Int. Conf. on Practical Applications of Computational Biology & Bioinformatics (PACBB 2011).

[3] Klenin KV, Tristram F, Strunk T, Wenzel W. Derivatives of molecular surface area and volume: simple and exact analytical formulas. J Comput Chem 2011; 32: 2647-2653.

[4] Lebedev VI, Laikov DN. A quadrature formula for the sphere of the 131st algebraic order of accuracy. Doklady Mathematics 1999; 59: 477-481.

[5] NVIDIA. NVIDIA CUDA Programming Guide 4. 2011.

[6] Pérez-Sánchez H, and Wenzel W. Optimization methods for virtual screening on novel computational architectures. Curr Comput Aided Drug Des 2011; 7, 44-52.

[7] Sanner MF, Olson AJ, Spehner JC. Reduced surface: an efficient way to compute molecular surfaces. Biopolymers 1996; 38: 305-320.

[8] Sánchez R, Sali A. Large-scale protein structure modeling of the Saccharomyces cerevisiae genome. PNAS 1998; 95; 13597-13602.