Comparing Parallel Algorithms for Van der Waals Energy with Cell-List Technique for Protein Structure Prediction *

Daniel R. F. Bonetti[1†], Gesiel Rios Lopes[1†], Alexandre C. B. Delbem[1†], Paulo S. L. Souza[1†], Kalinka C. Branco[1†], Gonzalo Travieso[1∓]

[1] University of São Paulo (USP – ICMC[†] – IFSC [∓])
São Carlos-SP – Brazil

daniel.bonetti@gmail.com, gesielrios@usp.br

{acbd, pssouza, kalinka}@icmc.usp.br, gonzalo@ifsc.usp.br

Abstract. *This paper compares the runtime of three distinct parallel algorithms for the evaluation of an* ab initio *and full-atom approach based on GA and cell-list technique, in order to minimize the van der Waals energy. The three parallel algorithms are developed in C and use one of these programming models: MPI, OpenMP or hybrid (MPI+OpenMP). Our preliminary results show that van der Waals Energy are executed faster and with better speedups when using hybrid and more flexible parallel algorithms to predict the structure of larger proteins. We also show that for small proteins the communication of MPI imposes a high overhead for the parallel execution and, thus the OpenMP presents a better relation cost x benefit in such cases.*

1. Introduction

The protein structure prediction (PSP) from its amino acid sequence is a complicated and expensive task, since that, according to Levinthal's paradox [Levinthal 1968], there are a vast number of conformations possible to reach the correct native state, which take an long time to evaluate, whereas in real proteins take only a few seconds or less to reach their native state. In addition, Anfinsen's thermodynamic hypothesis [Anfinsen 1972] states that, at least for small globular proteins, the native structure is a unique, stable, and kinetically accessible minimum of the free energy. Levinthal's paradox and Anfinsen's hypothesis allow us to formulate *ab initio* PSP as an optimization problem. In this context, Genetic Algorithms (GAs) have produced relevant results [Lima 2006, Dorn et al. 2011].

In the recent literature, there are different parallel solutions available aiming to minimize the execution time of such algorithms [Benítez and Lopes 2010, Bonetti et al. 2010, Bonetti et al. 2013]. Parallel algorithms in this context usually have distinct and limited performance, mainly because they are specific for one programming model and/or computer architecture. In [Bonetti et al. 2010] and [Bonetti et al. 2013], instead by just making parallel the van der Waals energy from its $O(n^2)$ algorithm, the authors first improved the efficiency of the energy using the cell-list algorithm, enabling the complexity reduction to $O(n)$.

This paper compares the performance (runtime) of three specific parallel algorithms for the evaluation of an *ab initio* and full-atom approach based on GA and cell-list technique, to minimize the van der Waals energy. The three parallel algorithms are

*This research project is sponsored by FAPESP (2013/07375-0).

developed in C and use one of these programming models: MPI, OpenMP or hybrid (MPI+OpenMP). We show, in our experiments, the importance to develop adaptive algorithms to explorer the benefits of different molecules, geometry, architectures and programming paradigms. Indeed, our preliminary results show that van der Waals Energy evaluation is faster when using hybrid parallel algorithms for larger proteins, even when they use as basis algorithms specific for determined programming paradigm (as message passing). A hybrid version is capable to minimize negative aspects of such specific programming paradigms. For small proteins, for example, the MPI communication imposes a high overhead with communication and, thus, in these cases, OpenMP presents better results. On the other hand, OpenMP does not maintain its performance for larger structures. Hybrid algorithms can act in this context, mitigating such problems.

The remaining of this paper is structured as follows. Section 2 introduces concepts related to the van der Waals calculation. Section 3 shows the method of cell-list. The configuration of the experiments and their results are presented in Section 4. Finally, Section 5 concludes this paper.

2. Van der Waals Energy

Van der Waals energy frequently describes the energy of a molecule. The Lennard-Jones potential (also known as Lennard-Jones 12-6) allows to calculate the van der Waals energy of a molecule [Jones 1924]. The van der Waals energy varies according to the distance of the pair of atoms and the type of atoms (hydrogen, carbon, nitrogen, oxygen, etc.), as shown in Equation 1, where r_{ij} is the relative distance.

$$r_{ij} = \frac{d_{i,j}}{R_i + R_j}. \tag{1}$$

The Lennard-Jones potential used in Protein Structure Prediction (PSP) is shown in Equation 2:

$$f_{LJ}(r_{ij}) = \begin{cases} Ar_{ij}^{-12} - Br_{ij}^{-6} \text{ if } r_{ij} > 0.8, \\ C \text{ if } r_{ij} \leq 0.8, \end{cases} \tag{2}$$

where A and B are constants experimentally determined based on characteristics of the environment, and C is given by $Ar_{ij}^{-12} - Br_{ij}^{-6}$ with $r_{ij} = 0.8$.

A molecule's van der Waals energy can be obtained by summing of the interaction of all pairs of atoms. It results in $\frac{n^2-n}{2}$ interactions, where n is the number of atoms of the molecule, showed in Equation 3:

$$E_{vdw} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{LJ}\left(r_{ij}\right). \tag{3}$$

3. Method of Cell-Lists

Cell-list is a general technique to improve the efficiency of algorithms responsible to calculate pairs of particles separated by a cutoff [Allen and Tildesley 1987]. It creates cells with least the cutoff length enabling the interaction of only atoms inside the cell

and neighboring cells. In this study, the cell-list technique is adapted to the van der Waals evaluation, in which the particles are represented by the atoms in the molecule configuration. We will use Cell-list Algorithm (CA) to describe the sequential cell-list algorithm developed, as well as Cell-list Parallel Algorithm (CPA) to describe the parallel version. CPA is divided into CPA with OpenMP, CPA with MPI and hybrid CPA.

4. Results and Discussion

We used the cluster belonging to LCR[1] to evaluate the runtime of the proposed approaches. The cluster has 14 nodes and is divided into two groups according to its characteristics. The first group has 10 nodes with AMD Dual-Core 64 bits 2.8 GHz processors and 4 GB of RAM. The second group has 4 nodes with Intel Core i7 64 bits 2.67 GHz processors and 12 GB of RAM. The operation system is GNU/Linux Ubuntu with kernel 2.6.26-2. All nodes have two network adapters: one for the file system and another for messages in MPI, both connected to two independent 3Com Gigabit Ethernet switches.

The inputs of the algorithms are based on proteins that differ in structure and size. Eight different sizes of proteins were chosen from PDB. Protein 1A11 was selected to be the lower bound, with only 390 atoms. Protein 1HTO represents the upper bound of experiments, with 147,900 atoms. Other proteins were selected to cover the range of proteins to evaluate the scalability of the proposed techniques. The accuracy reported from our algorithms remained unchanged for all executions, and for space reason, it will not showed in this paper.

The methods were statistically compared using the p-value of the Welch Two Sample t-test with 95% confidence interval. The comparisons made were: CA with Quadratic Algorithm (QA) in Equation 3; CPA OpenMP with 8 and 16 processors with CA; CPA MPI, processors ranging from 2 to 18 with CA; CPA hybrid, processors ranging from 8 to 32 with CA; CPA hybrid, processors ranging from 8 to 32 with CPA MPI; and finally, CPA hybrid, processors ranging from 8 to 16 with CPA OpenMP. All tests were performed for all 8 proteins, rendering in 176 tests. The highest p-value obtained was 0.002 and occurred between techniques CPA hybrid with CPA OpenMP, both with 8 processors.

4.1. Speedup of Cell-list Algorithm

Figure 1(a) shows the speedup achieved using the proposed CA in comparison to QA. Points represent the experimental data, and the line represents the predicted linear model. Indeed, the CA reduced the complexity from $O(n^2)$ to $O(n)$. The speedup line predicted linear increases according to the size of the protein. Even for small proteins, the speedup is significant. For protein 1AI0 with 4,728 atoms, the speedup is 5. Larger proteins did produce speedups more impressive, as 1HTO, which resulted in a speedup of 127. The improvement relies on the size of the cell grid (which also depends on the number of atoms). The larger the number of atoms, the larger will be the cell grid.

4.2. Parallel cell-list with OpenMP

Although the simplest parallel implementation of the van der Waals uses OpenMP, it can produce good results. Figure 1(b) shows the speedup achieved for CPA using OpenMP

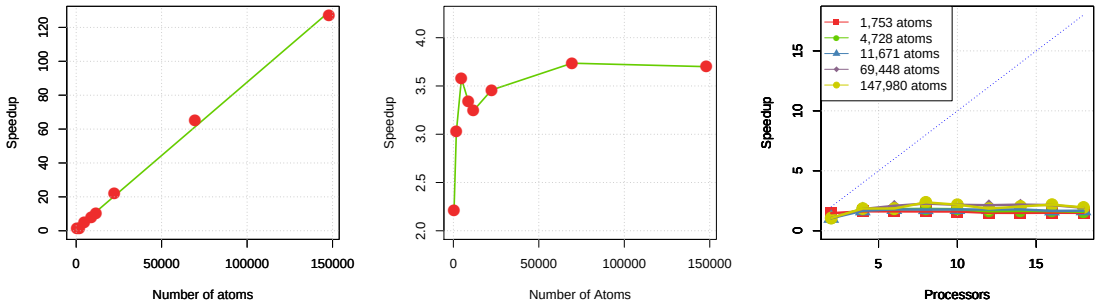
[1]Reconfigurable Computer Laboratory, group of research of Embedded and Evolutionary and Systems at Institute of Mathematics and Computer Sciences at São Paulo University, Brazil.

about CA. The experiment was performed in a node containing an i7 processor with 4 physical cores. The speedup is close to 4, indicating that the tasks were properly distributed among processors and the computational time of the van der Waals calculation was proportionally reduced by the number of cores.

The high number of atoms inside a single cell could be a small disadvantage of cell-list since it will have to compute more interactions, always performed in QA. Points four and five of Figure 1(b) show a depression in the speedup since these proteins are more globular than the others used. However, this speedup is still good.

4.3. Parallel cell-list with MPI

Figure 1(c) shows the speedup achieved when the van der Waals energy was computed using the CPA with MPI. The experiments were performed in 9 nodes of AMD processors. For small proteins, such as 1BFI (1,753 atoms), the speedup was not significant, due to inter-process communication. The cell-list procedure is so fast that the communication time strongly influences the total computational time, and the parallelization for small proteins is not viable. On the other hand, for proteins above 4,728 atoms, the speedup is significant for a small number of processors.



(a) (b) (c)

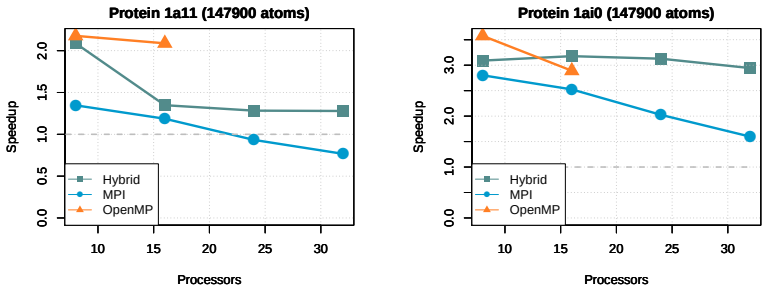
Figure 1. (a) Speedup achieved with CA in relation to QA. (b) Speedup achieved with CPA with OpenMP in relation to CA. (c) Speedup achieved for CPA with MPI in relation to CA.

4.4. Parallel cell-list with OpenMP and MPI (hybrid)

An expected good way to take advantage of both paradigms, OpenMP and MPI, is to use a hybrid paradigm, in which we can explore features of i7 processors with OpenMP. Besides, it can be used on several nodes with MPI. Figure 1(c) shows that for above 5 processors the efficiency is frozen. We ran the hybrid in 4 nodes of the cluster that contain the i7 processor, performing only four communication calls. After receiving the atoms by MPI, each node computes the specific region of the cell grid, splitting the tasks through OpenMP, i.e., each node could compute several tasks (8 in the i7 processor) using only one communication.

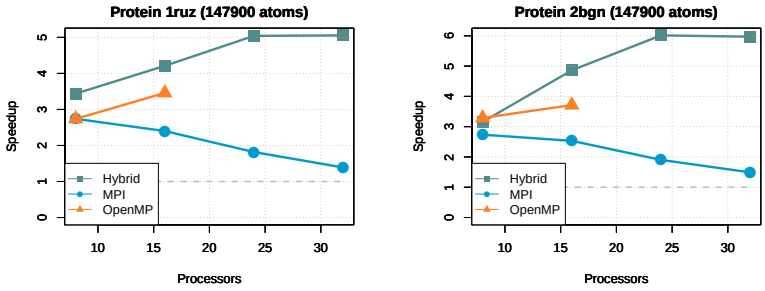
Figures 2(a), 2(b), 3(a) and 3(b) show the speedup achieved by the proposed methods (CPAs). For small proteins such as 1A11, the OpenMP paradigm is more adequate (Figure 2(a)). That happens since the number of cores of one node is higher than the number of tasks. For protein 1AI0 (Figure 2(b)), the number of processors is significant when considering the hybrid and the OpenMP alone, since, for above 15 processors, the

hybrid approach is fastest. In Figure 3(a), the hybrid is the fastest in all cases. The use of more processors than tasks will again produce the same speedup. Figure 3(b) shows the increase in the speedup for the hybrid approach. In all four cases, the use of MPI isolated is not a good approach. However, when combined with OpenMP, it produces better results.



(a) (b)

Figure 2. (a) Speedup achieved with the three proposed algorithms for protein 1A11 (390 atoms). (b) Speedup achieved with the three proposed algorithms for protein 1AI0 (4,728 atoms).



(a) (b)

Figure 3. (a) Speedup achieved with the three proposed algorithms for protein 1RUZ (22,380 atoms). (b) Speedup achieved with the three proposed algorithms for protein 2BGN (69,448 atoms).

5. Conclusions

The van der Waals energy used to evaluate the quality of proteins in GA can be efficiently computed using the cell-list technique, which evaluates proteins using an accurate algorithm with linear complexity.

Moreover, flexible parallel techniques applied to cell-list can be used to reduce the run time of the GA. By flexible, we mean the use of distinct parallel programming paradigms in a same algorithm, which, together, can explore diverse benefits available in parallel platforms.

This paper compares the results of three cell-list parallel programs: OpenMP version developed in [Bonetti et al. 2013], MPI developed in [Bonetti et al. 2010] and the new hybrid parallel implementation of cell-list using MPI and OpenMP concomitantly. All parallel versions reduced the running time of the van der Waals energy calculation,

when compared to the sequential version. The hybrid version, however, shows significant speedups independently from the size of the protein.

Our results show that the trade-off between communication and computation times of the MPI algorithm was not good and its speedup is limited. On the order hand, the openMP implementation is more suitable for small proteins, since its communication time is very short when compared to networks in a cluster. Therefore, the hybrid program using MPI and OpenMP has enough flexibility to speedup both small and large sizes of proteins. Results in our experiments show that the hybrid approach presents speedups, regarding the running time and maintain the same accuracy of the results. However, for smaller proteins, as expected, OpenMP using only one node with four cores can achieve speedups near to the speedups of the hybrid solution, offering, in such cases, a better relation cost x benefit.

This paper presented preliminary results belonging to an ongoing research project, focused on discovery new flexible parallel algorithms for PSP. As future work, it will be applied cell-list technique to reduce the complexity of other energy functions such as electrostatic, solvation and hydrogen bond energies, in order to produce more accurate and efficient GA for PSP. Furthermore, the results presented in this paper will be compared with tools like CHARMM, GROMACS, NAMD and AMBER.

References

Allen, M. P. and Tildesley, D. J. (1987). *Computer Simulation of Liquids*. Oxf. Un. Press.

Anfinsen, C. B. (1972). Studies on the principles that govern the folding of protein chains. *Nobel Lecture*, pages 103–119.

Benítez, C. M. V. and Lopes, H. S. (2010). Protein structure prediction with the 3d-hp side-chain model using a master–slave parallel genetic algorithm. *Journal of the Brazilian Computer Society*, 16(1):69–78.

Bonetti, D. R., Delbem, A. C., Travieso, G., and Souza, P. S. L. (2013). Enhanced van der waals calculations in genetic algorithms for protein structure prediction. *Concurrency and Computation: Practice and Experience*, 25(15):2170–2186.

Bonetti, D. R. F., Delbem, A. C. B., Travieso, G., and de Souza, P. S. L. (2010). Optimizing van der waals calculi using cell-lists and mpi. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1 –7.

Dorn, M., Buriol, L., and Lamb, L. (2011). A hybrid genetic algorithm for the 3-d protein structure prediction problem using a path-relinking strategy. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2709 –2716.

Jones, J. E. (1924). On the determination of molecular fields. ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A*, 106(738):463–477.

Levinthal, C. (1968). Are there pathways for protein folding? *Journal de chimie physique*, 65:44–45.

Lima, T. W. (2006). Algortimos evolutivos para predição de estruturas de proteínas. Master's thesis, ICMC-USP, São Carlos, SP.