The Assembly of a Computing Platform for Studying Protein Inhibitors Against COVID-19 Replication

Leonardo Pacifici[1(✉)], Ribi Akbar[1], Andrea Lombardi[1], Giuseppe Vitillaro[2], and Maria Noelia Faginas Lago[1]

[1] Department of Chemistry, Biology and Biotechnology, via Elce di Sotto, 8, 06123 Perugia, Italy
leonardo.pacifici@collaboratori.unipg.it
[2] CNR-SCITEC, via Elce di Sotto, 8, 06123 Perugia, Italy

Abstract. A new highly efficient GPU-equipped computing platform for studying the molecular inhibition mechanisms of the Sars-Cov-2 virus by natural compounds and aptamers has been installed and configured. Studies will be carried out by means of molecular dynamics methods and programs. For this reason, we have assembled specific hardware components into a 4U rack, together with a NVIDIA RTX 3060 GPU for speeding up molecular dynamics calculations and visualizing their outcomes. In fact, not only computational resources, in terms of computing power and execution times, are needed by molecular dynamics programs adopted by us, but also a system allowing the rendering and visualization of large biomolecules and their trajectories, such as viruses and proteins, represents a key factor for our work. Details about platform implementation and preliminary tests carried out are discussed.

Keywords: GPU · Molecular dynamics · 3D rendering · Sars-Cov-2 · Covid-19

1 Introduction

Nowadays, the two main well tested strategies designed to eradicate the Covid-19 pandemic are based either on the development of a vaccine or on the identification of proteins inhibitors involved in the replication of the Sars-Cov-2 virus, responsible for the disease. Among proteins playing a leading role in the viral replication are Sars-Cov-2 3a (or viroporin 3a), an ion channel expressed on the membranes of infected human cells and Sars-Cov-2 MPro, produced in same cells and deeply involved in the mechanism of virus replication and its maturation. More in detail, the protein Sars-Cov-2 3a is expressed on the plasma membrane of the host cell, where it forms an ion channel whose activity promotes the transport of potassium ions (K^+) across the membrane [1] and increases the pathogenicity of the virus by promoting its replication and release from the cell.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13382, pp. 264–274, 2022.
https://doi.org/10.1007/978-3-031-10592-0_20

Thus, it has been supposed that inhibition of this channel could represent a potential antiviral therapy for Covid-19; among the aims of our work, in fact, there is the development of the theoretical and computational apparatus for working out molecular dynamics observables to be compared with experimental ones [2]. Moreover, it is important to point out the growing interest of scientific community on the topic of ion channels in human cells. From a biological point of view they are pores in the cell membrane that allow ions to pass through the impermeable lipid bilayer of the cells. Besides experimental works carried out in the past, previous molecular dynamics studies were not able to reproduce experimentally observed selectivities. Now, with the COVID-19 pandemic explosion, they are receiving increasing attention due to their role in the virus replication and, thus, in its circulation. Moreover, the work carried out in the last two years for identifying possible inhibitors of Sars-Cov-2 replication in human cells is also fostering computational biochemistry studies aimed at characterizing the mechanisms involved in the transfer of biomolecular species through ion channels. Therefore, this represents one of the pillars on which our work lays.

On the other hand, the viral protein Sars-Cov-2 M-Pro is one of the most studied antiviral targets, playing a key role in the processing of viral polyproteins, a family of proteins involved in the control of virus replication and maturation [3,4]. For this reason, scientific community is evaluating the effectiveness of many natural molecules for the inhibition of Sars-Cov-2 M-Pro activity in order to contrast viral replication.

The Molecular Dynamics group of the Department of Chemistry, Biology and Biotechnology of the University of Perugia within a multidisciplinary project will use classical molecular dynamics (MD) for studying Sars-Cov-2 inhibitors. MD is a theoretical chemistry method for analyzing the movement of atoms and molecules using pre-described potential functions called force fields. This method can be employed to investigate the structural rearrangement of chemical and biochemical systems, such as pure solvents, mixed solutions, gaseous separation and combustion processes [5,6]. However, a generic type of force field is often limited in its use when investigating particular systems, for instance those involving biomolecules. In this work we are facing just this problem, that is the formulation of a force field to be used in MD simulations. It represents part of a greater effort towards the development of an anti-Covid-19 therapy, by identifying molecular inhibitors for the two proteins (A) and (B), through the design of two molecular platforms similar for path but different for molecules used (natural compounds and RNA aptamers, respectively). In particular, efforts of our group are aimed at identifying both the potential sites on the Sars-Cov-2 3a channel protein and simulate the inhibitory interaction and the best potential high affinity sites of aptamers on the Sars-Cov-2 MPro protein, simulating the inhibitory interaction.

Both tasks will be carried out by means of MD simulations, making use of the well known suites of programs DLPOLY [7] and NAMD [8]. However, in order to perform the calculations mentioned above use has to be made of a suitable computing platform. As a matter of fact, dynamics simulations of large biomolecules, like viruses or proteins, are difficult to carry out due to the large

number of atoms involved in the process. They request very long computing times and, in some cases, large amounts of RAM and storage memory. Another key factor in these works is concerned with the visualization of the outcomes of related simulations. When one has to visualize trajectories of hundreds of vibro-rotating molecules a standard computing platform results to be inadequate due to video rendering problems. In such cases, a GPU (Graphic Processing Unit) represents the best choice, both from a computational and cost point of view (not cheap but not too expensive either), to overcome issues related to long computing times and 3D video rendering processing. In fact, thanks to their highly parallel structure GPUs allow to speed up calculations [9,10] and make it possible the visualization of large biomolecular systems. Therefore, in order to carry out dynamics simulations outlined above a new computing platform equipped with a high performance GPU has been assembled. The paper is articulated as follows: in Sect. 2 the features of the computing platforms are discussed. In Sect. 3 preliminary results on the capabilities of the platform are illustrated. In Sect. 4 concluding remarks are given.

2 The Assembly of the Computing Platform

In order to build a suitable computing platform for our purposes we have assembled it in house. Its hostname is *axis* and it is equipped with:

- ASUS PRIME TRX40-PRO S motherboard
- AMD Ryzen Threadripper 3960X@3.8 GHz processor with 24 cores and 48 threads
- 128 GB DDR4 SDRAM
- 6 Crucial MX500 1TB SSD
- NVIDIA GeForce RTX 3060 with 3584 CUDA cores, 12 GB GDDR6 RAM and compute capability 8.6 (NVIDIA driver version 510.47.03).

We have installed on *axis* the Ubuntu 20.04 LTS server operating system (OS). In particular, we created a RAID 6 software implementation at OS level (using block-level striping with two parity blocks distributed across the six SSDs) and we installed a ZFS filesystem with a RAID-Z2 configuration at application level. This configuration guarantees high protection against data corruption, support for high storage capacities and, in general, great performances. On the pool we have also activated both compression and dedup. Finally, to the end of best compiling and running the computational packages cited above we also installed the Intel oneAPI toolkit (version 2022.1.2) and the CUDA one (version 11.6) [11].

As mentioned above, *axis* is intended for running extensive molecular dynamics calculations and for visualizing related outcomes. Taking into account the systems to be studied, for carrying out preliminary test calculations we selected NAMD, a parallel molecular dynamics code specifically designed for simulating large biomolecular systems. We adopted the 2.14 multicore release of NAMD. We built two complete NAMD binaries from source codes, a first one (standard) that makes use of C/C++ and Fortran compilers, Charm++/Converse,

TCL and FFTW libraries and a second one exploiting, in addition, the CUDA GPU acceleration. In more detail, use has been made of the g++ and gfortran compilers, while the FFTW libraries linked are those of the Intel oneAPI toolkit.

3 Preliminary Calculations on *Axis*

In order to perform tests on *axis* we ran preliminary calculations with NAMD using both binaries. The system selected is the one involving the Sars-Cov-2 3a protein, made of 11840 water molecules and 12415 residues. The total number of atoms amounts to 58352. In Fig. 1 is schematized the biomolecular system made by a lipid bilayer membrane, the Sars-Cov-2 3a protein forming the ion channel, sodium and chlorine ions. Water molecules are not shown in the figure. In order to build the initial configuration of the system and the inter- and intramolecular interactions we took advantage of:

- Orientations of Proteins in Membranes (OPM) database [12], providing spatial arrangements and orientation of the membrane protein with respect to the hydrocarbon core of the lipid bilayer;
- Protein Data Bank (PDB) [13], providing information about the shape of proteins;
- CHARMM-GUI [14], to interactively build the system and prepare its inputs, as well as for setting up the force field parametrization to obtain interaction potentials.

Once the input files are built, four steps are needed for running a full MD simulation with NAMD:

1. Minimization, to find the global energy minimum
2. Annealing, to heat the system (in the current case, from 60K to 300K)
3. Equilibration, to equilibrate the system after heating
4. Molecular Dynamics, to solve classical equations of motion.

For our simulations we adopted a NPT ensemble, a cut-off radius of 14 Å, an external electric field of 1.5 V and a timestep equal to 2 fs for a total of 10000 steps, corresponding to a simulation time of 0.02 ns. In more detail, we use a 2 fs time step because this is the most common one. Increasing the timestep might cause the system to explode because the total energy becomes very large. With respect to the whole simulation time, since this protein is a very complex system, the event normally happens in the period of nanoseconds or microseconds. Therefore for this system, we are trying to run simulations for more than 10^8 steps (200 ns), although for testing the application on our new platform we reduced the number of steps to 10000, as mentioned above. The use of an electric field is a way to force the ion to pass through the ion channel. Here we also refer

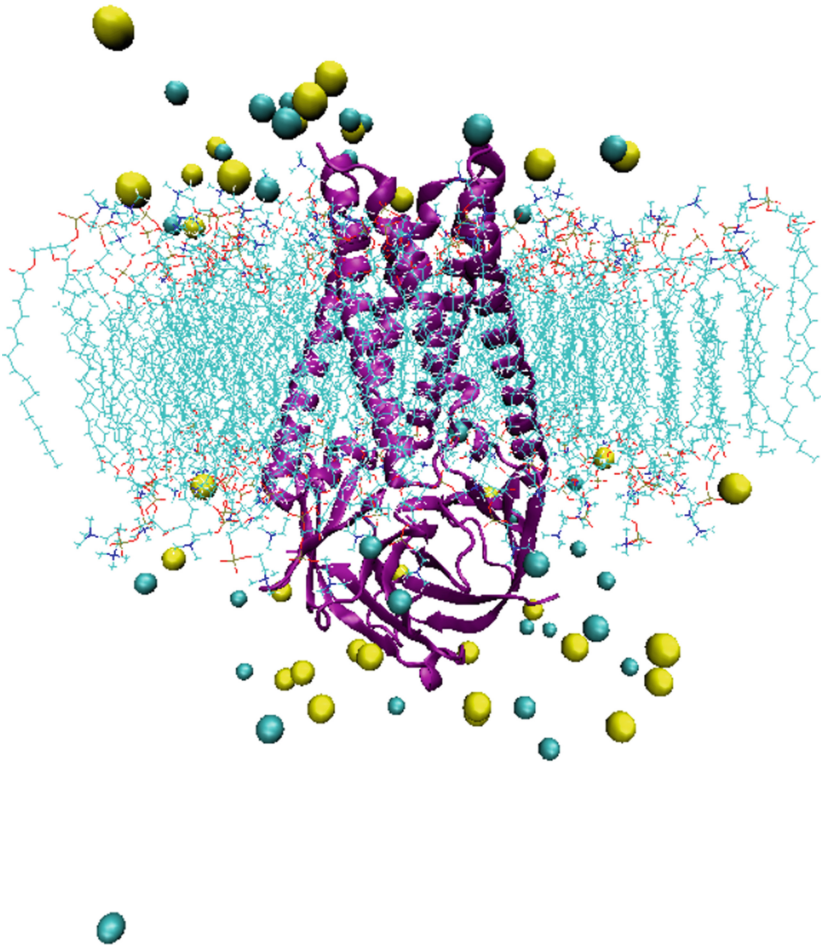

Fig. 1. Screenshot of the lipid membrane (light blue), the protein (violet) and Na^+ and Cl^- ions (yellow and cyan, respectively) (Color figure online)

to the voltage-gated ion channel that involves the electric field for the passage of ions from extracellular to the cytosol. Regarding the composition of the systems, the ion concentration, etc. we refer to the experimental data. We measured the wall clock times for steps 1 and 4, using both the standard implementation of NAMD and the one obtained using the CUDA acceleration, running the binaries on an increasing number of cores.

Table 1. Wallclock time measurements for minimization runs using the standard implementation and the CUDA based one

Number of cores	$t_{standard}$(s)	t_{CUDA}(s)
1	432.72	21.93
2	221.46	14.27
4	116.43	11.81
6	81.21	10.92
8	62.78	10.72
10	53.49	10.50
12	46.10	10.32
14	40.81	10.64
16	36.77	10.55

Table 2. Wallclock time measurements for molecular dynamics runs using the standard implementation and the CUDA based one

Number of cores	$t_{standard}$(s)	t_{CUDA}(s)
1	2426.39	111.87
2	1224.20	61.30
4	627.76	39.91
6	434.42	35.41
8	331.80	33.81
10	274.52	32.42
12	233.98	32.16
14	205.81	32.12
16	180.64	30.68

In Table 1 and 2 the measured computing times for the minimization and MD simulations are shown, respectively. As can be seen from the Tables, in both cases the CUDA binary is drastically faster in terms of computing time. In particular, the acceleration is more pronounced when less cores are used. In fact, on a single core the time saving factor is about 20 with respect to the standard implementation, while on 16 cores it reduces to about 3.5 and 6 for the minimization simulation and for the MD one, respectively. This is likely due to the fact that the use of a larger number of cores in the calculation on one hand implies a greater number of CUDA *threads* running on the GPU (in fact, the percentage use of the GPU for MD simulations is 20% using one core and 60% using two cores), on the other hand more CPU-GPU data transfers are needed, resulting in an overall time overhead that reduces the time gain when compared to the multicore execution of the standard NAMD implementation. This is also confirmed by speedups estimates in Fig. 2, 3, 4 and 5, plotted as a function

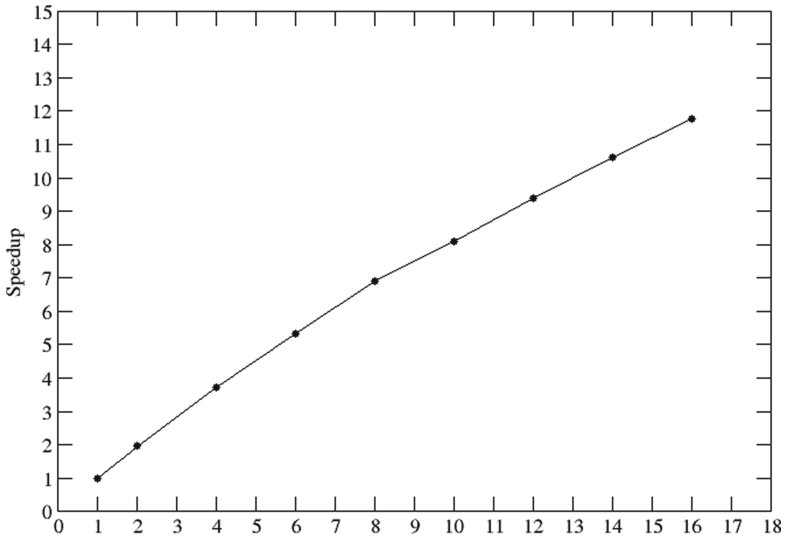

Fig. 2. Calculated speedup for the minimization runs using the standard implementation

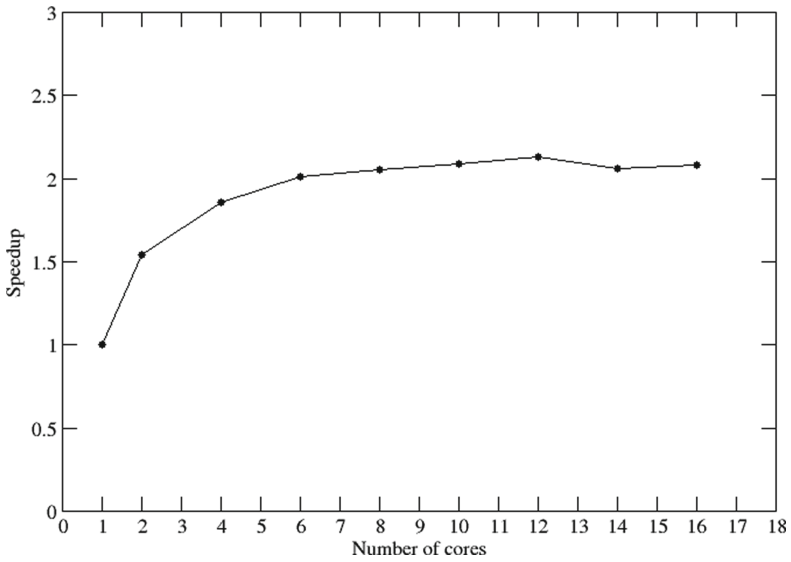

Fig. 3. Calculated speedup for the minimization runs using the CUDA based implementation

of the number of cores. From Fig. 2 and 4, where the speedups obtained using the standard compiled binary of NAMD are shown, it can be seen an almost linear speedup for the minimization runs, which improves slightly for the MD

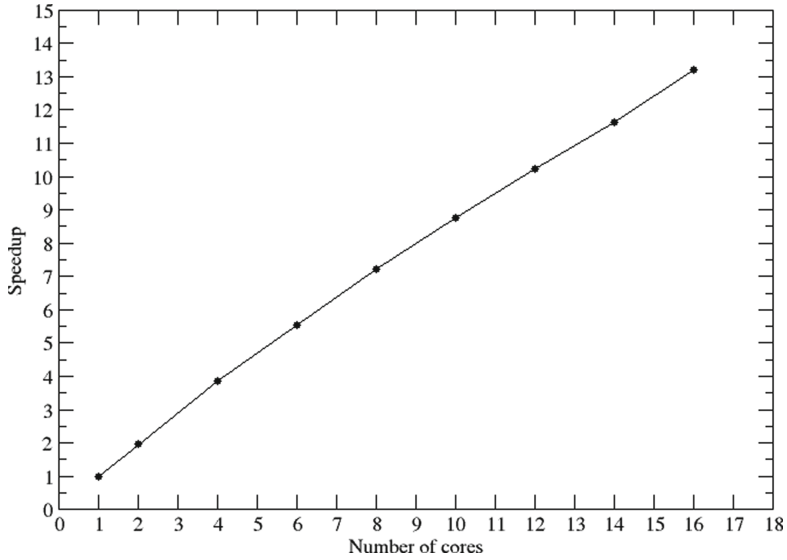

Fig. 4. Calculated speedup for the MD runs using the standard implementation

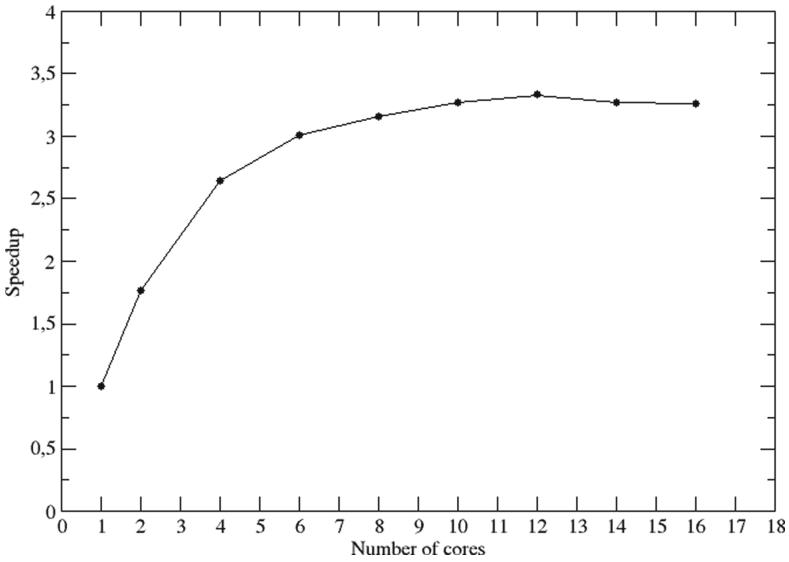

Fig. 5. Calculated speedup for the MD runs using the CUDA based implementation

ones. This means that executing NAMD on more than one core guarantees good performances and that the program scales well with the number of cores.

This is not observed in Fig. 3 and 5, where speedups obtained running the CUDA implementation of NAMD are shown. In fact, as it can be seen from the

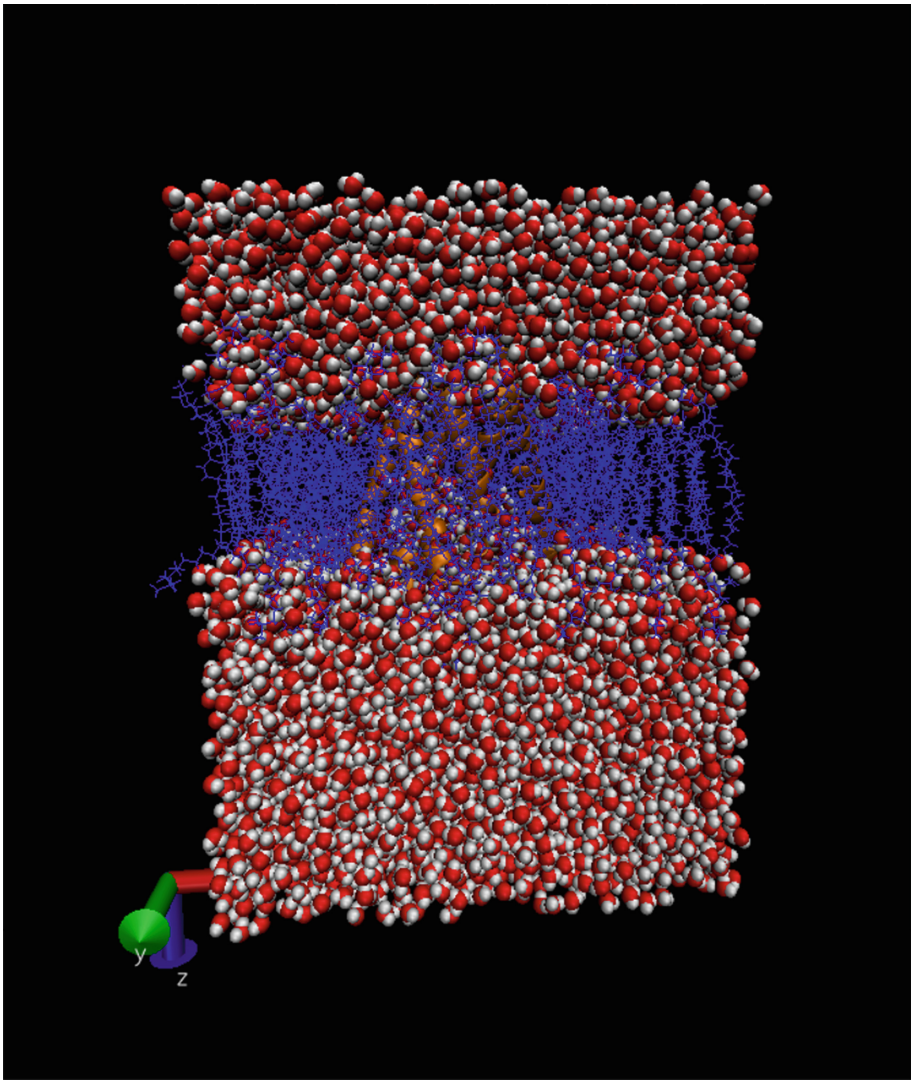

Fig. 6. Screenshot of the biomolecular system from VMD (Color figure online)

Figures, it is not linear. Speedup values increase slightly up to 8 cores for the minimization runs and to 10 for the MD ones, to become flat afterwards. The better performances shown in Fig. 5 are likely due to a heavier use of the GPU in carrying out molecular dynamics calculations whereas algorithms employed in the NAMD minimization step do not use effectively the GPU. In any case, the flattening of the curves in both simulations is probably due to CPU-GPU data transfer overhead, which masquerade the computational gain arising from a parallel multicore execution on the GPU *threads*.

As already mentioned before, the assembly of the current computing platform has been made necessary not only for the computational requirements of the programs chosen for studying the biomolecular systems described above, but also to make it possible the rendering and visualization of related outputs. In fact, a standard platform not equipped with a suitable GPU does not allow the use of any program for visualizing molecules and related trajectories, when dealing with a great number atoms, as it is the case. For our purposes use has been made of the VMD tool [15], a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting. As an example, in Fig. 6 is depicted a typical screenshot obtained by manipulating the output of NAMD with VMD. The image is so clear that is possible to recognize the spike protein (orange), water molecules and the lipid membrane (blue), while the ions are not represented. The visualization of thousands of trajectories is also a relatively easy task when performed with the GPU acceleration.

4 Conclusions

In this work we have described the in-house assembly of a new computing platform called *axis* for the study of large biomolecular systems involved in the replication of the Sars-Cov-2 virus. *Axis* benefits from a ZFS filesystem with a RAID-Z2 configuration at application level and from a NVIDIA GPU of the RTX series with 12 GB of RAM memory. In order to test its capabilities and carry out performance tests we have compiled NAMD sources and have run preliminary molecular dynamics calculations on a system made by a human cell membrane with the ion channel represented by the Sars-Cov-2 3a protein. Related performance measurements demonstrate that calculations carried out with the CUDA acceleration are dramatically faster than those performed with the non-CUDA executable. On the other hand, the standard binary scales better with the number of cores when compared to the CUDA based one, showing an almost linear speedup up to 16 cores, whereas the CUDA one is far from linearity (the curve becomes flat starting from eight-ten cores), most likely due to the heaviness of data transfers from CPU to GPU (and vice-versa), when the number of processes involved in the calculation increases. Thanks to the rapid progress in GPU technology we will be able to run large batches of molecular dynamics calculations with NAMD within reasonable times and to visualize its outcomes. Finally, based on the results we are going to obtain by running extended MD calculations on this new computing platform we might extend them to other biological systems.

Acknowledgements. The authors thank MUR and Perugia University for financial support through the AMIS project ("Dipartimenti di Eccellenza-2018–2022"). AL and NFL also thank Fondazione Cassa di Risparmio di Perugia n.# 220.0513 to C.E.

References

1. Castaño-Rodriguez, C., et al.: Role of severe acute respiratory syndrome coronavirus viroporins E, 3a, and 8a in replication and pathogenesis. MBio **9**(3), e02325-17 (2018)
2. Kern, D., et al.: Cryo-em structure of sars-cov-2 orf3a in lipid nanodiscs. Nat. Struct. Mol. Biol. **28**(7), 573–582 (2021)
3. Zhang, X., Tan, Y., Ling, Y., et al.: Viral and host factors related to the clinical outcome of covid-19. Nature **583**, 437–440 (2020)
4. Jin, Z., Du, X., Xu, Y., et al.: Structure of mpro from sars-cov-2 and discovery of its inhibitors. Nature **582**, 289–293 (2020)
5. Faginas-Lago, N., Albertí, M., Costantini, A., Laganà, A., Lombardi, A., Pacifici, L.: An innovative synergistic grid approach to the computational study of protein aggregation mechanisms. J. Mol. Model. **20**(7), 1–9 (2014). https://doi.org/10.1007/s00894-014-2226-4
6. Lombardi, A., Faginas-Lago, N., Laganà, A.: Grid calculation tools for massive applications of collision dynamics simulations: carbon dioxide energy transfer. In: Murgante, B. (ed.) ICCSA 2014. LNCS, vol. 8579, pp. 627–639. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09144-0_43
7. Smith, W., Forester, T.R.: `DL_POLY_2.0`: a general-purpose parallel molecular dynamics simulation package. J. Mol. Graph. **14**(3), 136–141 (1996)
8. Phillips, J.C., et al.: Scalable molecular dynamics with namd. J. Comput. Chem. **26**(16), 1781–1802 (2005)
9. Pacifici, L., Nalli, D., Laganà, A.: Quantum reactive scattering on innovative computing platforms. Comput. Phys. Commun. **184**(5), 1372–1380 (2013)
10. Pacifici, L., Nalli, D., Skouteris, D., Laganà, A.: Time dependent quantum reactive scattering on GPU. In: Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2011. LNCS, vol. 6784, pp. 428–441. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21931-3_33
11. Cook, S.: CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, 1st edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2012)
12. Lomize, A.L., Pogozheva, I.D.: Orientations of proteins in membranes (OPM) database, 27 April 2022
13. Berman, H.M., et al.: The protein data bank. Nucleic Acids Res. **28**(1), 235–242 (2000)
14. Wu, E.L., et al.: Charmm-GUI membrane builder toward realistic biological membrane simulations. J. Comput. Chem. **35**(27), 1997–2004 (2014)
15. Humphrey, W., Dalke, A., Schulten, K.: VMD: visual molecular dynamics. J. Mol. Graph. **14**(1), 33–8 (1996)