BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **17**, No 5
Special issue with selected papers from the workshop
"Two Years Avitohol: Advanced High Performance Computing Applications 2017"

Sofia • 2017 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2017-0056

On the Performance and Energy Consumption of Molecular Dynamics Applications for Heterogeneous CPU-GPU Platforms Based on Gromacs

Armen Poghosyan[1], *Hrachya Astsatryan*[2], *Wahi Narsisian*[2], *Yevgeni Mamasakhlisov*[3]

[1]*The International Scientific – Educational Center of the National Academy of Sciences of the Republic of Armenia, 24D, M. Baghramyan ave., 0019 Yerevan, Armenia*
[2]*Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia, 1, P. Sevak str., 0014 Yerevan, Armenia*
[3]*Yerevan State University, 1, A. Manoogian st., 0015 Yerevan, Armenia*
E-mails: sicnas@sci.am hrach@sci.am wahi@sci.am y.mamasakhlisov@ysu.am

Abstract: *High Performance Computing* (*HPC*) *accelerates life science discoveries by enabling scientists to analyze large data sets, to develop detailed models of entire biological systems and to simulate complex biological processes. As computational experiments, molecular dynamics simulations are widely used in life sciences to evaluate the equilibrium nature of classical many-body systems The modelling and molecular dynamics study of surfactant, polymer solutions and the stability of proteins and nucleic acids under different conditions, as well as deoxyribonucleic acid proteins are studied. The study aims to understand the scaling behavior of Gromacs* (***Gro****ningen* ***ma****chine for* ***c****hemical* ***s****imulations*) *on various platforms, and the maximum performance in the prospect of energy consumption that can be accomplished by tuning the hardware and software parameters. Different system sizes* (48K, 64K, *and* 272K) *from scientific investigations have been studied show that the GPU* (*Graphics Processing Unit*) *scales rather beneficial than other resources, i.e., with GPU support. We track* 2-3 *times speedup compared to the latest multi-core CPUs. However, the so-called "threading effect" leads to the better results.*

Keywords: *Molecular dynamics simulations, protein, DNA, sodium dodecyl sulphate, energy consumption, Gromacs, benchmarking, HPC.*

1. Introduction

The power consumption of large-scale High Performance Computing (HPC) systems is becoming a critical demand in the context of increasing the performance regardless of energy consumption [1]. Therefore, energy efficiency improvement becomes a challenge for HPC applications. HPC accelerates life science discoveries by enabling scientists to analyse large data sets, to develop detailed models of entire biological systems and to simulate complex biological processes.

As computational experiments, Molecular Dynamics (MD) simulations are widely used in life sciences to evaluate the equilibrium nature of classical many-body systems [2, 3]. The study of systems with a large number of atoms in long trajectory intervals (from nano to milliseconds) is required to explore a broad range of interesting phenomena, which is undoubtedly unfeasible without using appropriate HPC resources.

The modelling and MD study of two complex systems are examined: surfactant, polymer solutions [4] and the stability of proteins and nucleic acids under different conditions [5], as well as DeoxyriboNucleic Acid (DNA) proteins [6].

In case of surfactant and polymer solutions, the experimental timescale of protein denaturation is estimated to be in order of micro- to milliseconds. The surfactant-induced denaturation study has been solely available with the increase of computational resources, which make it feasible to conduct the research of more realistic systems for solid understanding of all mechanisms [7]. Before the long production run, it is important to check the performance regardless energy consumption to reveal the resource productivity. Such a long timescale simulation will enable to check the influence of Sodium Dodecyl Sulphate (SDS) on acyl coenzyme. A binding protein (ACBP) near Critical Micelle Concentration (CMC) is aimed to elucidate how the SDS operates at molecular level and reveal the mechanism of protein folding in terms of atomic interactions. The main point is to check whether denaturation of ACBP is induced by the association of alkyl sulphates to a protein or is provoked by protein-ions interactions. The mechanism of the protein denaturation induced by surfactant coming from real experiments assumes the binding of the surfactant counter-ions to sites of a protein molecule, i.e., the surfactant ions come close and bind to the oppositely charged groups of the protein, which causes the protein to unfold [8].

The DNA proteins are essential for biosensors, due to the possible applications in the fields of social interest such as defence and security, medical research and clinical diagnostics, etc., [9]. The molecular recognition element can be a biological molecule, such as single-stranded DNA (ssDNA) proteins, or a complex biological system. The general concept for DNA sensor design is that the target DNA must be captured at the recognition layer as a result of hybridization, which is based on the ability of ssDNA to recognize its counterpart strand with a complementary nucleotide sequence. Experimentally, the hybridization of free DNA in solution takes only seconds depending on the sequence length and concentration [10, 11]. The bottleneck of hybridization kinetics is expected to be the stage of ssDNA strands diffusion. At the same time, the timescale of hybridization is mostly slow on the surface in comparison with those in bulk [12]. The present benchmark study has also been performed for the subsequent investigation of the stability and kinetics of Acyl-Coenzyme ACBP and dsDNA in the presence of such low – molecular compounds as surfactants and counter ions. In particular, it is planned to simulate in atomic details the mechanical unzipping followed by hybridization of a dodecamer of helical B-DNA with the sequence d(CGCAAATTTCGC)2 with Gromacs and NAMD simulation packages. The simulations with GPU/CPU hybrid show that unzipping takes about 0.7-1 ns depending on the value of the extending force, while the stage of dsDNA formation is expected to take about 10-102 ms. Hybridization kinetics depends on the multiple factors, including nucleotide sequence, length of the chain, countering and other co-solutes concentrations, temperature, pH, etc. Thus, the proposed research requires multiple massively parallel computations under a wide range of environmental conditions.

In case of successful investigation of the ssDNA hybridization in bulk, we also plan to extend our research to the nucleic acids hybridization on the surface. It is also planned to perform a long-run MD simulation of surfactant/protein complexes near CMC, which requires a couple of microsecond runs.

The aim is to study the performance and the energy consumption of the above-discussed two studies using heterogeneous CPU-GPU resources. The remaining content of the paper is organized as follows. The computational platforms and testing data can be found in Section 2 and Results are given in Section 3. Finally, the conclusion and directives for future research are drawn in Section 4.

2. Simulation systems

The following three different scientific systems (48K, 64K and 272K) have been utilized and studied to benchmark the applications using the Gromacs package:

The System I – Acyl-Coenzyme ACBP in water bulk in the presence of Sodium Dodecyl Sulphate (SDS) and dodecyl-**b**eta-D-Maltoside (bDM). The system consists of ~48,000 atoms in 8×8×8 nm^3 cell.

The System II – B-DNA

(5'-D(*CP*GP*CP*GP*AP*AP*TP*TP*CP*GP*CP*G)-3')

in water bulk in the presence of NaCl. The system consists of ~62,800 atoms in 7.56×5.362×16.0 nm^3 cell.

The System III – Acyl-Coenzyme ACBP in water bulk in the presence of SDS near CMC. The system consists of ~272,000 atoms in a 14×14×14 nm^3 cell.

The systems mentioned above, which are rather complex and require PME (Particle Mesh Ewald) calculations, are chosen to study the performances to calibrate the platform/duration ratio for further long production runs.

The following parameters are used for the experiments:

PME electrostatics – 0.5 fs time step.

Van der Waals forces – 1.2 and 1.4 nm with corresponding pressure and temperature control.

Denaturation of the dsDNA the extension rate – 10 nm/ns with the harmonic force constant 103 $kJ.mol^{-1}.nm^{-2}$.

Steps – 5000/10000/375000/3750000 with and without writing output trajectories and coordinate files.

Two mentioned systems have been benchmarked to estimate optimal performance for further MD runs and elucidate how SDS and bDM operate at the molecular level, as the interaction of the protein with surfactants is somewhat important.

The third system has been used to estimate optimal performance for further MD runs and elucidate how the external force, applied to the center of mass of the one strand of dsDNA operate at the molecular level and to estimate the characteristic time of hybridization after the force denaturation.

3. Computational resources

For this investigation, the ARIS supercomputer, which is one of the most powerful computer architectures available in the South East Europe, is used. ARIS, which is operated by Greek Research and Technology Network, is a general-purpose supercomputer with a total sustained performance of ~180 TFlop/s based on the Linpack Benchmark. ARIS was ranked 468th fastest supercomputer in the world on Top500 (June 2015 iteration). The supercomputer consists of 532 computational nodes divided by four levels:

Regular nodes without any accelerators (Thin nodes) – 426 nodes.

Accelerated nodes equipped with NVIDIA Tesla K40m cards (GPU nodes) – 44 nodes.

New generation Intel Phi accelerated nodes (Phi nodes) – 18 nodes.

Fat compute nodes with significant number of cores/memory in compare with thin nodes (Fat nodes) – 44 nodes.

The computational nodes of ARIS are interconnected through Infiniband high-performance, multi-purpose network architecture, where the signal rate is 56 Gbit/s (~7 GB/s) (FDR 14x Infiniband technology), and the latency is ~0.7 ms. Note that the version of 5.1.4 of Gromacs is available on ARIS supercomputer with standard compilation and optimization options.

4. Results and discussions

4.1. 48K system

To understand the performance behaviour of Gromacs for a single node and efficiency of OpenMP parallelization, the performance of 48K system depending on some threads (Fig. 1) is studied. The number of tasks is selected with the -ntasks option of SLURM (Slurm Workload Manager) job scheduler, while the controlling threads have been done using the mdrun command line (flag) -ntomp and the environment variable OMP_NUM_THREADS. Fig. 1 shows that OpenMP parallelization is efficient when the number of threads does not exceed the total number of cores, which means the total number of cores with a thread per node should be applied to achieve the maximum performance.

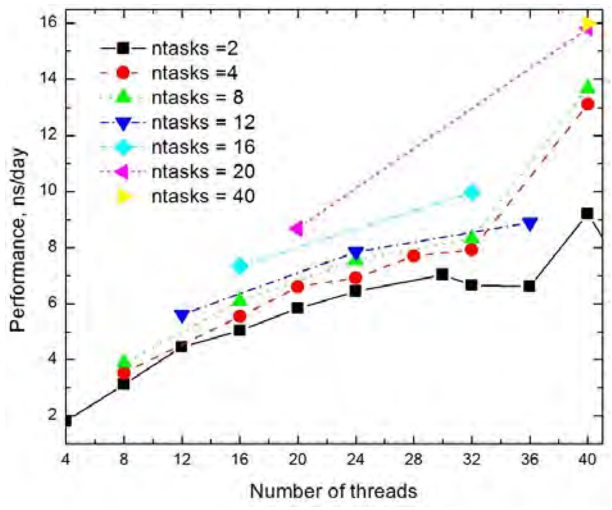

Fig. 1. Single node benchmarking by changing export OMP_NUM_THREADS and mdrun -ntomp tag for Series I

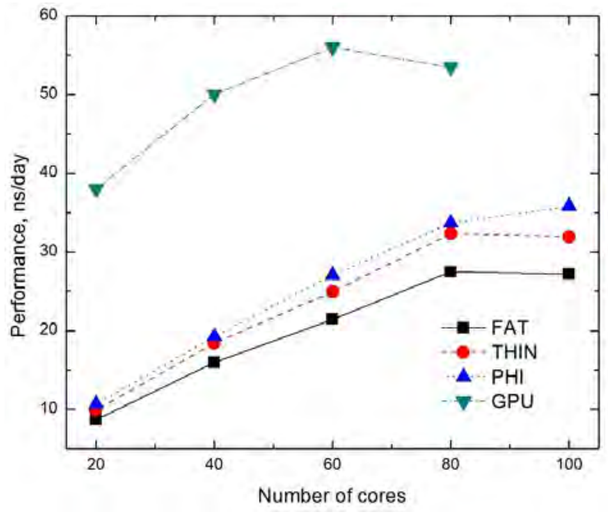

Fig. 2. Benchmarking by changing the number of nodes/tasks for Series I

The second step was to understand the scaling behaviour of System I on all available platforms, we provide the Gromacs performance as a function of nodes/cores (Fig. 2). The mentioned benchmark was done in a simple way, i.e., by adding only #SBATCH – -nodes=*N* (*N*=1, 2, 3) and #SBATCH – -ntasks-per-node=40 line to SLURM script. Note that the performance for System I was carried out manipulating the automatic PME load balancing and 1 OpenMP thread per MPI process parameters. The curves plotted in Fig. 2 show that increasing the number of cores/nodes leads to the increase of performance, i.e., we track that MPI parallelization scales well up to 80 cores. Note that the further increase of cores/nodes makes impossible to run, as the Gromacs found no domain composition with the given box and cell size (the test system is too small to scale beyond 100 cores).

To understand “threading” effect, several experiments have been carried out by changing modes/tasks with a relation with *N* threads, when *N*=1, 2, … , 9. The corresponding curve for the 48K system implemented in FAT nodes is presented in Fig. 3. It is obvious that trying TPN (Tasks Per Node)×Threads=40 lets us reach a maximum performance of around ~62 ns per 1 d, i.e., when MPI process×OpenMP threads are equal the number of physical cores in a node (40 on FAT nodes) we track significant speed up. Note that the further increase of threads (accelerating more than 40 computing units per nodes) leads to a sharp decrease of performance (10 nodes, TPN=8, Threads=5 the value is 61.5 ns per 1 d, while Threads=6 is ~4.5 ns per 1 d).

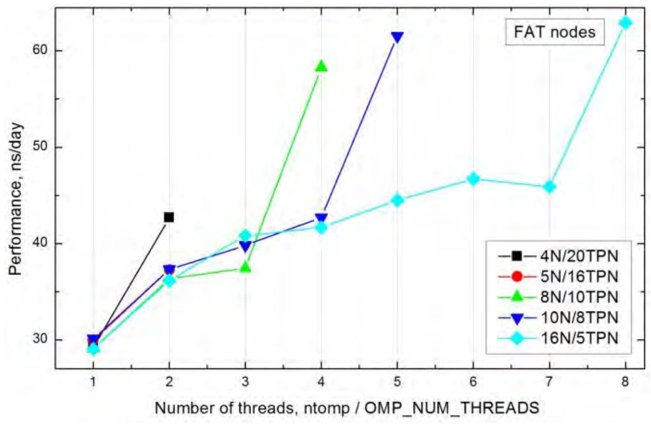

Fig. 3. Benchmarking depending on a number of threads by ntomp/OMP_NUM_THREADS. FAT nodes Benchmarking by changing a number of nodes/tasks for Series I

To achieve maximum performance on given hardware requires complex examination of mdrun parameters. We have tested the system by using the flags “-pin on” and “-dlb yes” with Gromacs. Note that changing mdp parameters (no pressure coupling, no write trajectories, and confout) slightly increased the performance. For instance, dynamic load balanced is on, we received ~65.6 ns per 1 d (default is 61.5 ns per 1 d with “threads=5, nodes=10 and TPN=8” set), and consequently pin=on leads to the increase of performance (~66.7 ns per 1 d), while -noconfout tag leads to the value of ~66.02 ns per 1 d. Note that when all cores are used by mdrun module, “-pin” flag needs to be on in order to give an increase of performance, i.e., “-pin on” stops the system kernel from moving mdrun processes between cores (locking the cores and do not allow to jump around), which is less wasteful. The dynamic load balancing (“-dlb” tag), which is automatically running when load imbalance is 5% or more, is of particular importance for inhomogeneous systems. The Gromacs mdrun -noconfout option and correspondingly nst(*x*, *v*, *f*)out = 0 prevent writing final coordinate and trajectory files, which in some manner gains computational time. For optimal performance, we also suggest to add mdrun –resethway and –maxh = 0.05 options as well. Note that the resetting all time and cycle counters (–resethway option) corrects the benchmarks’ results and also get the value hovers around ~74 ns per 1 d.

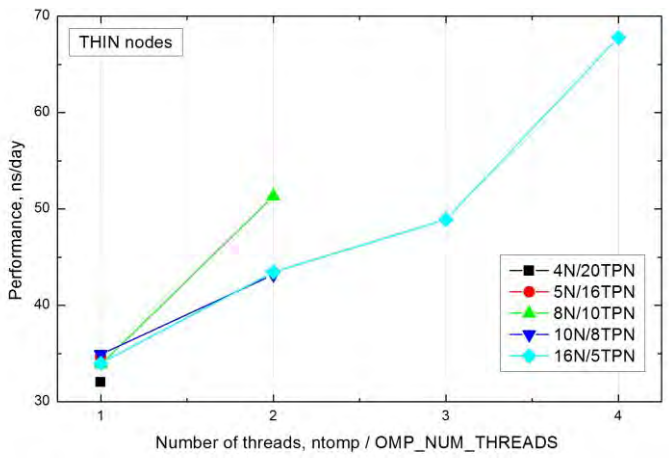

Fig. 4. Benchmarking depending on a number of threads by ntomp/OMP_NUM_THREADS. THIN nodes. Benchmarking by changing a number of nodes/tasks for Series I

The "threading" behaviour is almost same in Thin nodes, where the difference is only the number of physical cores in a node (Thin – 20 cores per node). The plotting is shown in Fig. 4, where one can see the increase of performance with the increase of threads up to a certain value. Further increase of threads as in case of FAT nodes, we get worse values of performance. The maximum value of above-mentioned parameters ("-pin on", "-dlb yes", -noconfout, –resethway and –maxh =0.05 for "threads=4, nodes=16 and TPN=5" set) was fixed at 82.709 ns per 1 d.

The behaviour of Intel Xeon Phi nodes is a bit different in comparison with other nodes. The current version of Gromacs does not support offloading of Intel Xeon Phi, i.e., it works as simple compute nodes without 61-cores support. The "threading effect" gives us the similar results as in case of Thin nodes (Fig. 5).

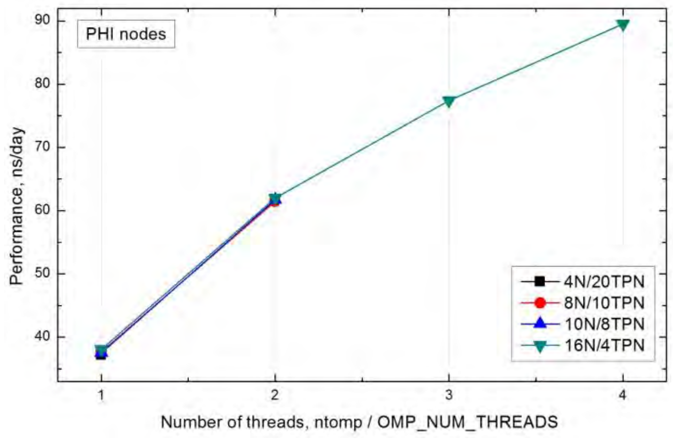

Fig. 5. Benchmarking depending on a number of threads by ntomp/OMP_NUM_THREADS. PHI nodes. Benchmarking by changing a number of nodes/tasks for Series I

4.2. 64K system

Similar simulations have been carried out for 64K system, where the performance depending on a number of threads is presented in Fig. 6. Here, we track the same tendency, as in case of the 48K system, i.e., the optimal [tasks multiply to threads per core] case is the same on a single node. We argue that a small difference in a number of particles of the systems does not lead to cardinal changes when using single node.

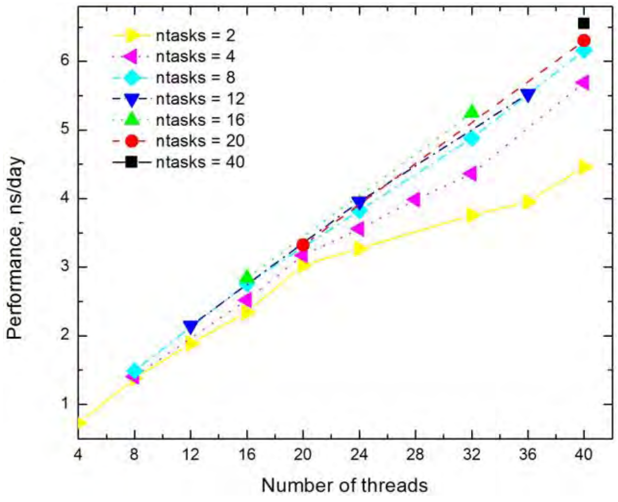

Fig. 6. Single node benchmarking by changing export OMP_NUM_THREADS and mdrun -ntomp tag for Series II

The scaling behaviour of System II on all available platforms has been revealed using some optimizations. The corresponding plots were monitored in Fig. 7. We argue the data are similar in case I, i.e., the increase of a number of cores/nodes lead to the increase of performance. As in case of the 48K systems, here, we also show that the GPU is ~5-6 times faster than the CPUs.

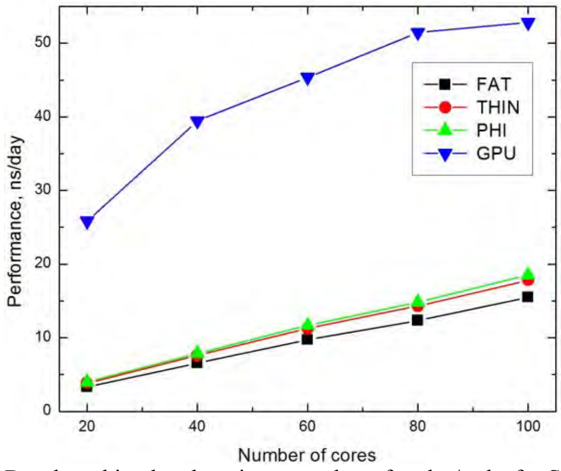

Fig. 7. Benchmarking by changing a number of nodes/tasks for Series II

Exploring FAT node “threading” on the 64K system, we see that a maximum performance reaches when TPN×Threads=40 (~50.05 ns per 1 d). It is worth mentioning that the further increase of threads leads up to 10 times worse results (for instance, 10 nodes, TPN=8, Threads=5 (~41.4 ns per 1 d), while for 6 is ~4,4 ns per 1 d). The FAT nodes threading data for the 64K system is shown in Fig. 8.

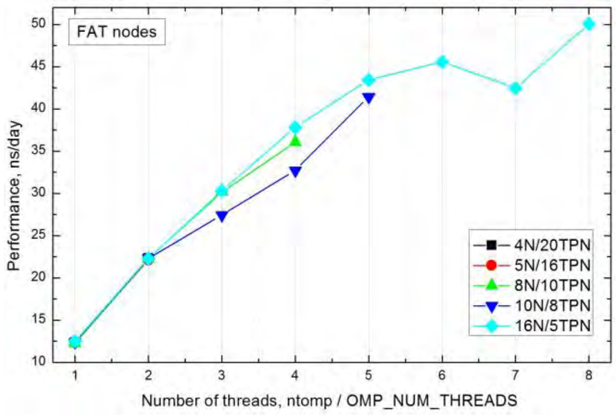

Fig. 8. Benchmarking depending on a number of threads by ntomp/OMP_NUM_THREADS. FAT nodes. Benchmarking by changing a number of nodes/tasks for Series II

Thin node “threading” experiments show a small difference between data, as the number of physical cores in a Thin node is 20. The corresponding curves are shown in Fig. 9. Although the performance is a bit lower than in case of the 48K system, here, we see that increasing of threads leads to the better performance.

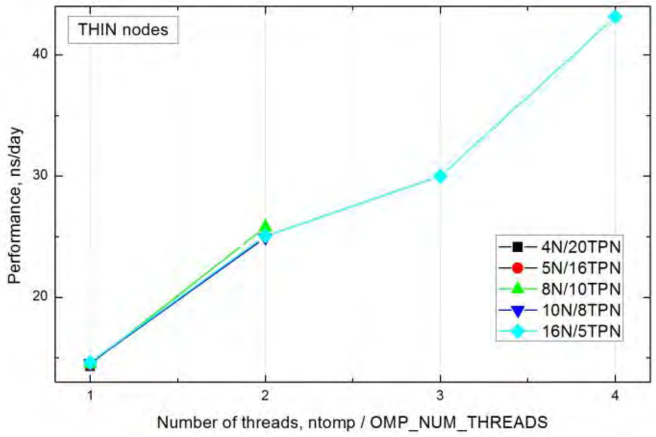

Fig. 9. Benchmarking depending on a number of threads by ntomp/OMP_NUM_THREADS. THIN nodes. Benchmarking by changing a number of nodes/tasks for Series II

The behaviour of Intel Xeon Phi nodes in case of the 64K system is worse compared to the 48K system and we track the sharp decrease of performances with the increase of a number of threads. As discussed previously, this is maybe due to the incompatibility of Gromacs package with 61-cores support PHI machines. The corresponding curves are provided in Fig. 10.

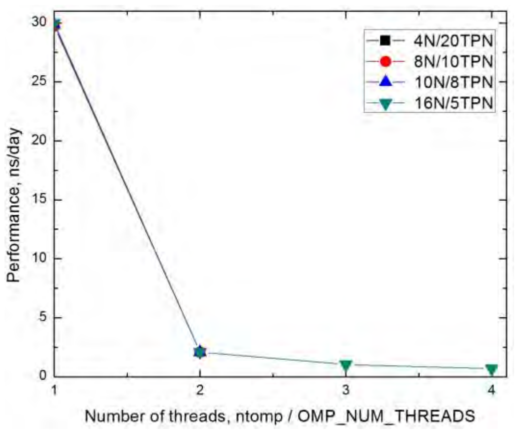

Fig. 10. Benchmarking depending on a number of threads by ntomp/OMP_NUM_THREADS. PHI nodes. Benchmarking by changing a number of nodes/tasks for Series II

We argue that in case of PHI nodes, the so-called "threading" effect does not lead to significant drift of performance values.

4.3. 272K system

To study the performance behaviour of large systems, we have replicated the existing system up to 272K atoms. However, here we present only Gromacs performance as a function of nodes/cores without any threading effects. In Fig. 11, the corresponding curves are shown.

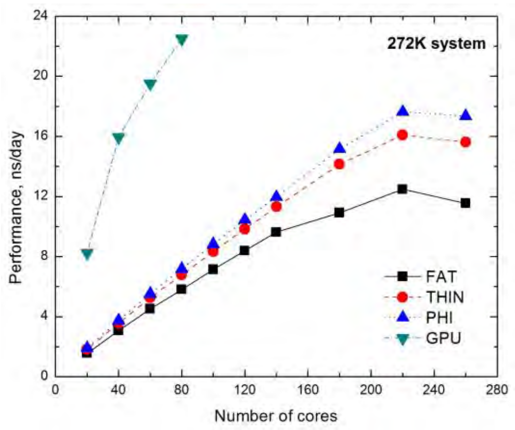

Fig. 11. Benchmarking by changing a number of nodes/tasks for Series III – 272K atoms

As in case of previous systems, here, we provide the data with automatic PME load balancing and 1 OpenMP thread per MPI process. The plots in Fig. 11 proved that increasing the number of cores/nodes leads to the increase of performance until 220 cores, that is to say, MPI parallelization scales well up to 220 cores.

The GPU support surely shows a sharp increase of performance as expected, i.e., dealing with GPU leads to 3-4 times better performance that bare CPU.

5. The energy minimization for the systems

One of the most critical challenges in HPC is the reduction of total power consumption of an application by taking into account the performance, and the most optimized usage of resources in terms of nodes, cores etc. Gromacs, which is widely used by different research groups in the field of molecular dynamic simulation, must think about how to setup and run the system to decrease the energy consumption on different platforms. Gromacs has a highly optimized design to use efficiently of any underlying hardware. It can be run on a single need and benefit from the multithreading feature using OpenMP, and it can be run with Message Passing Interface (MPI) library to use intranode parallelism efficiently. Also for newer versions, the calculation of short-range nonbonded forces can be done on GPUs.

During this study three different Gromacs systems have been examined (48K, 64K, and 128K). Various combinations of nodes/cores to evaluate the best performance/energy consumption relationship do all the conducted experiments. The results of the experiments are provided by the following tables (all absent values are due to the missing of domain decomposition for these combinations).

Table 1. Simulation times and energy consumption per watt for the 48K system

Platform	1 node simulation time (min)/energy watt (h)	2 nodes simulation time (min)/energy watt (h)	3 nodes simulation time (min)/energy watt (h)	4 nodes simulation time (min)/energy watt (h)
FAT	30/54.5	42/152.6	-/-	28/203.46
GPU	7/16.3	10/46.6	15/105	19/177.3
Thin	28/22.86	31/50.63	33/80.85	35/114.3

Table 1 shows that the best combination of performance with minimum energy consumption for 48K systems is by using one GPU node.

Table 2. Simulation time and energy consumption per watt for the 64K system

Platform	1 node simulation time (min) /energy watt (h)	2 nodes simulation time (min) /energy watt (h)	3 nodes simulation time (min) /energy watt (h)	4 nodes simulation time (min) /Energy watt (h)	5 nodes simulation time (min) /energy watt (h)
FAT	48/87.2	-/-	90/490.5	-/-	150/1362.5
GPU	55/128.3	71/331.3	92/644	109/1017	135/1575
Thin	733/598.6	739/1207	744/1822	762/2489	763/3115

Table 2 shows that the best combination of performance with minimum energy consumption for 64K systems is by using a single FAT node.

Table 3. Simulation time and energy consumption per watt for the 272K system

Platform	1 node simulation time /energy watt	2 nodes simulation time /energy watt	3 nodes simulation time /energy watt	4 nodes simulation time /energy watt
FAT	15:37/1702 h	15:41/3418 h	17:10/5613 h	-/-
GPU	2:52/401 h	-/-	3:37/1379 h	4:10/2333 h
Thin	13:11/645 h	13:30/1323 h	13:41/2011 h	14:08/2770 h

Table 3 demonstrates that the best combination of performance with minimum energy consumption for 272K systems is by using a single GPU node.

6. Conclusion

A series of benchmarks have been carried out to reveal the performance behaviour of Gromacs package on different platforms through the evaluation of the threads/cores/nodes conceptions.

Three systems have been taken into account 48K, 64K and huge 272K atoms correspondingly. Accordingly, the following platforms were taken for benchmarking: (i) Regular nodes without any accelerators (Thin nodes) – 426 nodes; (ii) Accelerated nodes equipped by NVIDIA Tesla K40m cards (GPU nodes) – 44 nodes; (iii) New generation Intel Phi accelerated nodes (Phi nodes) – 18 nodes, and (iv) Fat compute nodes with larger number of cores/memory in compare with thin nodes (Fat nodes) – 44 nodes.

The experiments argue that compare to CPUs GPUs scales rather good performance. The simulations with GPU support leads to 2-3 times speedup compared to the latest multi-core CPUs. It is also stated that so-called "threading effect" leads better results on all platforms. It should be also noted that in case of 64K system the Phi benchmarking show worse results, which is most probably due to a disagreement between software package and Phi processors, as it works as simple compute nodes without 61-cores support.

From the energy consumption perspective, we can state that for 48K and 227K systems using one GPU node is giving better performance/energy results, and for 64K systems the best solution is to use a single FAT node. All in all the usage of multiple nodes are not providing a good performance due to intensive inter process communication. Therefore, the usage of more nodes increases the energy consumption.

***Acknowledgments*:** The research leading to these results has been co-funded by the European Commission under the H2020 Research Infrastructures contract No 675121 (Project VI-SEEM) and contract No 653194 (subcontract entitled "Persistent Identifier Services for the life science community in Armenia"), and the Armenian State Committee of Science under the Project No 15T-1B292 (Towards the Green e-Infrastructures).

References

1. D'Hollander, E. H., J. J. Dongarra, I. Foster, L. Grandinetti, G. R. Joubert. Transition of HpcTowards Exascale Computing (Advances in Parallel COmputing), IOS Press, 2013, p. 232.
2. Alder, B., T. Wainwright. Studies in Molecular Dynamics. I. General Method. – J. Chem. Phys., Vol. **31**, 1959, No 2, 459.
3. Tuckerman, M., G. Martyna. Understanding Modern Molecular Dynamics Methods: Techniques and Applications. – J. Phys. Chem., Vol. **104**, 2000, No 2, pp. 159-178.
4. Poghosyan, A. H., L. H. Arsenyan, H. V. Astsatryan. Dynamic Features of Complex Systems: A Molecular Simulation Study. – Modeling and Optimization in Science and Technologies. Vol. **2**, 2014, pp. 117-121.

5. P o g h o s y a n, A. H., H. V. A s t s a t r y a n, A. A. S h a h i n y a n. Parallel Peculiarities and Performance of GROMACS Package on HPC Platforms. – International Journal of Scientific and Engineering Research, Vol. **4**, 2013, No 12, pp. 1755-1761.
6. K a r a p e t i a n, A. T., Z. A. G r i g o r y a n, Y. S. M a m a s a k h l i s o v, M. V. M i n a s y a n t s, P. O. V a r d e v a n y a n. Theoretical Treatment of Helix-Coil Transition of Complexes DNA with Two Different Ligands Having Different Binding Parameters. – Journal of Biomolecular Structure and Dynamics, Vol. **34**, 2016, No 1, pp. 201-205.
7. O t z e n, D. E. Protein-Surfactant Interactions: A Tale of Many States. – Biochim. Biophys. Acta, Vol. **1814**, 2011, No 5, pp. 562-591.
8. J o n e s, M. N. A Theoretical Approach to the Binding of Amphipathic Molecules to Globular Proteins. – Biochem. J., Vol. **151**, 1975, No 1, pp. 109-114.
9. I v n i t s k i, D., I. A b d e l H a m i d, P. A t a n a s o v, E. W i l k i n s. Biosensors for Detection of Pathogenic Bacteria. – Biosensors and Bioelectronics, Vol. **14**, 1999, No 7, pp. 599-624.
10. G a o, Y., L. K. W o l f, R. M. G e o r g i a d i s. Secondary Structure Effects on DNA Hybridization Kinetics: A Solution Versus Surface Comparison. – Nucleic Acids Res., Vol. **34**, 2006, No 11, pp. 3370-3377.
11. M o r r i s o n, L. E., L. M. S t o l s. Sensitive Fluorescence-Basedthermodynamic and Kinetic Measurements of DNA Hybridization in Solution. – Biochemistry, Vol. **32**, 1993, No 12, pp. 3095-3104.
12. A z a m, M. S., J. M. G i b b s - D a v i s. Monitoring DNA Hybridization and Thermal Dissociation at the Silica/Water Interface Using Resonantly Enhanced Second Harmonic Generation Spectroscopy. – Analyt. Chem., Vol. **85**, 2013, No 17, pp. 8031-8038.