



EGi-Engage

Implementation and evaluation of AMBER and/or GROMACS

Date	30 March 2016
Activity	WP6n
Lead Partner	CIRMMP
Document Status	FINAL
Document Link	https://documents.egi.eu/document/2774

Abstract

Here we benchmarked the AMBER and GROMACS software packages for molecular dynamics (MD) simulations to compare their performance on single and multi-core CPUs, which are widely available within the EGI infrastructure, with respect to their performance on GPGPUs, which are a scarcer resource. Our data showed that GPGPUs yield an improvement in the speed of MD calculations as high as a 150 factor with respect to a single core, or 4.7 with respect to a 64-core system. This opens up the possibility not only to perform longer simulations to investigate slow motions that are relevant to biological function but also to address larger molecular systems such as the macromolecular machines of the cell. We extended our benchmark to two other software tools, PowerFit and DisVis, which also showed great benefit from the availability of GPGPUs with an acceleration with respect to a single core CPU of a 20-30-fold. With respect to 32-core CPUs, the acceleration afforded by GPGPUs ranges from a 2-fold (for DisVis and PowerFit) up to an 8-fold for MD.



This material by Parties of the EGI-Engage Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

The EGI-Engage project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 654142 <http://go.egi.eu/eng>

COPYRIGHT NOTICE



This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

DELIVERY SLIP

	<i>Name</i>	<i>Partner/Activity</i>	<i>Date</i>
From:	Antonio Rosato	CIRMMP/NA6	11/03/2016
Moderated by:	Małgorzata Krakowian	EGI.eu/NA1	
Reviewed by	Carlos Fernandez	CESGA/PMB	22/03/2016
	Francisco Sanz	BIFI	22/03/2016
	Mariusz Sterzel	CYFRONET/PMB	18/03/2016
Approved by:	AMB and PMB		4/04/2016

DOCUMENT LOG

<i>Issue</i>	<i>Date</i>	<i>Comment</i>	<i>Author/Partner</i>
v.1	22/02/2016	First draft	Antonio Rosato/CIRMMP
v.2	24/02/2016	Integrated GROMACS section	Ales Krenek / CESNET
v.3	29/02/2016	Review	Alexandre Bonvin / UU; Marco Verlatto /INFN
v.4	10/03/2016	Final draft	Antonio Rosato/CIRMMP
FINAL	30/03/2016	FINAL version	Antonio Rosato/CIRMMP

TERMINOLOGY

A complete project glossary is provided at the following page: <http://www.egi.eu/about/glossary/>

Contents

1	Introduction	5
1.1	Infrastructures used and their relationship with JRA2	6
2	Implementation and benchmarking of AMBER	7
2.1	rMD benchmark with the AMPS-NMR portal	7
2.2	Unrestrained MD benchmark for a large macromolecular adduct	9
3	Implementation and benchmarking of GROMACS	13
3.1	Software and experiment setup	13
3.2	Unrestrained MD benchmark	13
4	Other software tools for Structural Biology	16
4.1	PowerFit	16
4.2	DisVis	18
5	Conclusions	22
6	Future work	23
7	References	24

Executive summary

Molecular dynamics (MD) is extensively used in Structural Biology research applications to simulate the motions (i.e. local/global changes in 3D structure) of biological macromolecules (proteins, nucleic acids) over time and as a function of different environmental conditions. Simulations can also include experimental data in order to refine or improve the agreement with the data of 3D structural models.

Here we benchmarked the AMBER and GROMACS software packages for MD simulations to compare their performance on single and multi-core CPUs, which are widely available within the EGI infrastructure, with respect to their performance on GPGPUs, which are a scarcer resource. Our data showed an improvement in the speed of calculations as high as a 150 factor with respect to a single core or 4.7 with respect to a server blade harbouring 64 cores when moving to the GPGPU environment. This opens up the possibility not only to perform longer simulations to investigate slow motions that are relevant to biological function but also to address larger molecular systems such as the macromolecular machines of the cell. We extended our benchmark to two other software tools DISVIS and PowerFit, which also showed great benefit from the availability of GPGPUs with an acceleration with respect to a single core CPU of a 20-30-fold. With respect to 32-core systems, the acceleration afforded by GPGPUs ranges from a 2-fold (for DisVis and PowerFit) up to an 8-fold for MD.

The present results demonstrate the usefulness of GPGPU resources for various applications in Structural Biology. This is particularly true for simulation of molecular dynamics with the AMBER package, but also applies to GROMACS, another MD program, and for new, more recently developed tools covering other aspects of Structural Biology not directly related to dynamics.

1 Introduction

In order to understand how biological macromolecules (protein, nucleic acids) function one must know their 3D structure, i.e. the relative spatial position of all atoms in the system, and dynamics, i.e. how such positions change over time at a given temperature. Structural Biology is the scientific discipline that addresses such issues at the experimental and computational level. The characterization of the 3D structure of biological macromolecules can be achieved through different experimental techniques. The choice of the most appropriate technique often depends on the size of the system of interest, namely by the number of atoms whose position should be defined. This is described by the molecular weight of the system, expressed in Daltons (Da, which is the unified atomic mass unit i.e. the standard unit for indicating mass on an atomic or molecular scale). NMR spectroscopy is routinely used for systems up to 15-20 kDa, X-ray diffraction can be used for systems of virtually any size, and finally electron microscopy is normally applied to systems of at least 50 kDa. Regarding molecular motions, NMR spectroscopy is the only experimental approach that can afford direct, atomic-level information. Unfortunately, not all atoms in the system can be characterized by NMR spectroscopy, so it is virtually impossible with current techniques to achieve the same level of experimental characterization for all regions of a biological macromolecule.

Molecular dynamics (MD) simulations are powerful tools for the exploration of the conformational energy landscape accessible to biological macromolecules. All-atom MD simulations provide a complete description of the molecular motions affecting all parts of the system of interest, at the same level of spatial resolution. Such simulations thus circumvent the limitations described in the previous paragraph. Furthermore, they can be readily integrated with different types of experimental data to validate the computer-generated predictions. The limit with MD simulations is generally the computational power available, which defines the largest molecular weight and longest simulation times that can be tackled. Clearly, the aspects are inversely correlated: given the computational infrastructure, shorter simulations can be achieved for larger systems, and vice versa. However, the biological problem that one wants to address will require a minimum length of the simulation, so that being able to produce longer and longer simulations effectively unlocks a wider range of biological aspects that MD can provide insight into. At the same time, it is now generally accepted that proteins and nucleic acids in the cell do not function in isolation. Thus, the focus of research is more and more shifting towards the analysis of multi-macromolecular complexes and so-called macromolecular machines, which have larger size than individual macromolecules. Here we additionally tackled a specific application of MD in the field of macromolecular structure determination, namely the refinement of 3D structural models against experimental data. In these applications, energetic pseudo-potentials describing the experimental information are integrated into the overall potential of the simulation in order to steer the system toward conformations that are in agreement with the data and at the same time feature optimal geometric properties and inter-atomic contacts.

In the last few years, many research teams using MD have focused on accelerated computing as an efficient, cost-effective resource to push the limits of their simulations. In this context, the structural biology community has been particularly interested in GPGPUs (General-Purpose computation on Graphics Processing Units). MoBrain has endeavoured to demonstrate the usefulness of GPGPUs for MD applications. In this document, we report on benchmarking efforts for popular MD software tools under typical, real application settings. We also extended our effort to benchmark two other recent Structural biology tools, to demonstrate the broader value of using GPGPUs in this area of science. To do this, we implemented a submission mechanism for GPGPU-based computations with gLite to GPGPU-supporting CREAM-CE. In addition, we tested the performance of GPGPU clouds, which allowed us to conclude, by comparison with published benchmarks, that the cloud environment does not impose a significant overhead on MD calculations.

1.1 Infrastructures used and their relationship with JRA2.

For the calculations described in Sections 2 and 4, we leveraged the work done in the context of JRA2 to enable GPGPU support in the EGI HTC platform (see Deliverable D4.6). For this, we implemented at the CIRMMP data centre a test-bed with three nodes (2x Intel™ Xeon™ E5-2620v2) each harbouring two NVIDIA™ Tesla™ K20m GPGPU cards. The test-bed was managed by the Torque 4.2.10 Local Resource Management System (LRMS, compiled with NVIDIA NVML libraries) with the Maui 3.3.1 scheduler, and was installed with CUDA 5.5. The last version of the EM13 CREAM-CE was finally installed on top of the LRMS to enable remote grid access to enmr.eu VO members.

The first prototype of GPGPU-enabled CREAM-CE featured two new JDL attributes: GPUNumber and GPUMode. The glite-ce-submit client could successfully submit AMBER jobs remotely to this prototype. Further tests involved the use of the Maui scheduler, which failed to define the NVIDIA compute mode. This issue was present also for other popular LRMSes like LSF, Slurm, SGE and Condor. Therefore, the final GPGPU-enabled CREAM-CE release did not include the GPUMode JDL attribute any longer. This CREAM-CE version deployed on the CIRMMP test-bed was used to support the DisVis and PowerFit applications, encapsulated in Docker containers exploiting the GPGPU cards. Note that AMBER does not allow GPU contention, so we did not investigate this aspect.

For GPGPU virtualization, we run calculations at the IISAS-GPUCloud - NGI_SK EGI federated cloud site. The predefined “gpu.large” virtual machine flavour was used, mapping to Intel™ E5-2650v2 @ 2.6 GHz CPU, 8 GB RAM, and NVIDIA™ Tesla™ K20m GPGPU. The EGI-supported virtual machine image used Ubuntu 14.04 and had the necessary CUDA version 7.5 development and runtime. This is essentially the configuration described in Deliverable D4.6.

2 Implementation and benchmarking of AMBER

We installed the PMEMD tool of the AMBER package¹ on the GPGPU test-bed described in Section 1.1.

We decided to benchmark two types MD simulations that are often performed with the AMBER package:

1. Restrained MD (rMD), i.e. including experimental data that define the conformational space available to the molecule
2. Unrestrained, also called free, MD simulations.

Besides their specific research purpose, the two types of simulations differ in the input data required. For free MDs only the atomic coordinates of the initial 3D structural model of the biological system of interest are needed. For rMD simulations, the experimental data are required as an additional input, in a suitable format. rMD simulations are enabled by the AMPS-NMR portal which was developed in the context of the WeNMR project [1;2]. AMPS-NMR uses experimental NMR data as its input. For the rMD benchmark, it was necessary to modify the source code of PMEMD in order to harmonize the treatment of NMR experimental data to that implemented in AMPS-NMR. No modifications in the code were instead needed for the free MD benchmark. In the benchmarks, we referred to CPU systems using either AMD™ Opteron™ 6366-HE or Intel™ Xeon™ CPU E5-2620 v2 processors.

2.1 rMD benchmark with the AMPS-NMR portal

The present benchmark exploited publicly available data for the homeodomain fragment of the human stem cell transcription factor Nanog. An overview of the input data is given below:

Protein structure

- Protein size (MW): 10,069 Da
- Protein size (amino acids): 84
- PDB ID: 2KT0

NMR experimental data

- Distance restraints: 889
- Torsion angle restraints: 832
- BMRB ID: 16680

¹ <http://ambermd.org/>

The rMD calculation was performed with the same simulation protocol [2]. It is a fixed-time simulation, where the solvated protein is initially heated, then allowed to move for a predefined number of steps and then cooled again to 0 K. The rationale is to optimize the energetics of the protein while maintaining agreement with the restraints. The AMPS-NMR portal was used to set up the simulation parameters, and thus only the executable had to be defined differently for the comparison.

The graph in figure 1 indicates the different times (logarithmic scale) required for the simulations, per each conformer of the family:

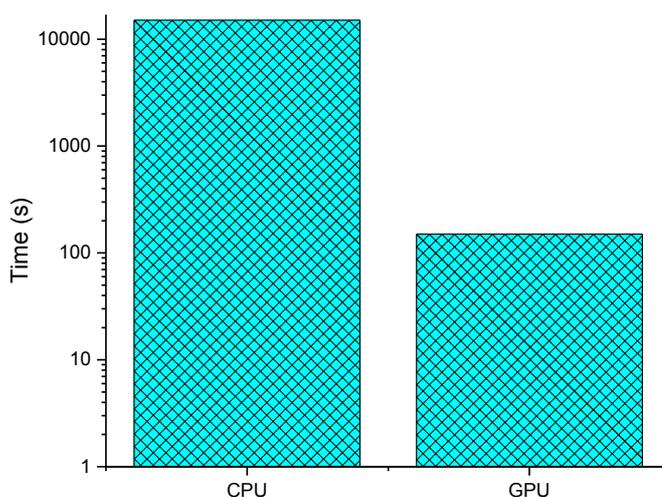


Figure 1 – Calculation times for rMD simulations with the AMPS-NMR portal for the Nanog protein on a single-core CPU (AMD™ Opteron™ 6366-HE) vs one GPU card (NVIDIA™ Tesla™ K20). Note the logarithmic scale of the y axis.

The **acceleration factor** under this real case scenario is thus about **100** (15,060 s/150 s). The reference calculation was done against a single-core CPU; rMD simulations do not scale well with the number of cores because of the way the restraint potential is coded in the software. Note that the AMBER code does not rely on the CPU to enhance performance while running on a GPU, and in particular it involves very little CPU to GPU communication. In practice, the CPU server does not affect the GPU performance.

The agreement between the refined structural models and the experimental restraint data in the two simulations, as measured by the residual deviations of the back-calculated data with respect to the input, was essentially identical. This demonstrates that the use of GPUs does not introduce numerical errors due to single precision.

2.2 Unrestrained MD benchmark for a large macromolecular adduct

To benchmark unrestrained MD simulations with AMBER we used a more challenging macromolecular system than the system we used in 2.1, owing to its large total mass (see below). The system of choice was the M homopolymer ferritin from bullfrog. This system is composed by 24 identical protein chains, forming a single macromolecular adduct.

Protein structure

- System size (MW): 499,008 Da
- Protein size (amino acids): 4,200 (24 x 175) aa
- PDB ID: 4DAS

The following image displays the architecture of ferritin, where each protein chain is coloured by a different colour. Ferritin forms a hollow sphere, and is a biological cargo/storage system

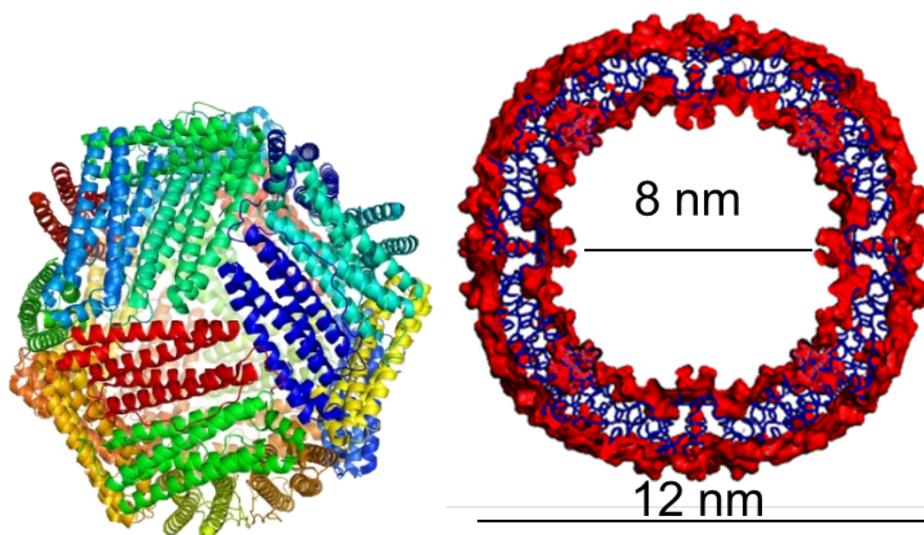


Figure 2. The architecture of ferritin. Left panel: ribbon representation in which each protein chain is coloured by a different colour. Right panel: the internal cavity of ferritin

The fully solvated system that serves as the starting point for the simulation contains as many as about 180,000 atoms (with respect to 1,440 atoms in the Nanog protein of section 2.1). Again, we set up the same simulation parameters for both the CPU and GPU computational infrastructures. Because unrestricted MD simulations are run for as long as possible, the meaningful parameter to monitor is not total simulation time. Instead, one can monitor the wall time required to compute a

given number of steps (i.e. to calculation motions over a given simulation time) or, conversely, the number of steps (simulation time) computed in a fixed wall time interval (i.e. nanoseconds of simulation computed per day). This is shown in figure 3 (note the logarithmic scale in the right panel).

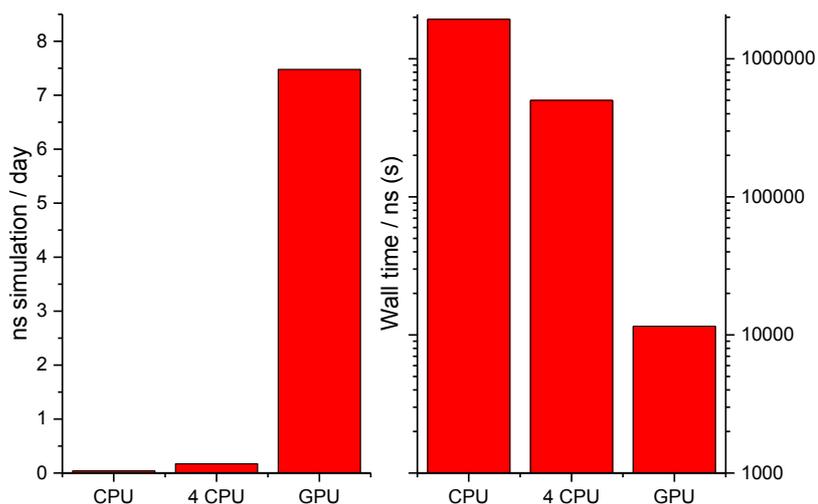


Figure 3. Comparison of the performance achieved using single/multi-core CPUs (AMD™ Opteron™ 6366-HE) vs one GPU card (NVIDIA™ Tesla™ K20). Left: nanoseconds of simulation that can be computed in one day of wall time; Right: Wall time required to compute 1 nanosecond of simulation (logarithmic scale)

For the ferritin system, the calculation on one GPU card achieved about 7,5 ns of simulation per day, with respect to 0.18 on a 4-core CPU and 0.05 for a single core system. The time required to produce one ns of simulation was respectively of $1.15 \cdot 10^4$ s, $5.0 \cdot 10^5$ s and $1.93 \cdot 10^6$ s. Thus, the GPU card achieved two orders of magnitude improvement over a single CPU core (right panel of the figure above). **The comparative performance was GPU:4 cores:1 core = 150:3.6:1.00.** Note that the code scales almost linearly with the number of cores. On this basis, we decided to extend our initial benchmark to include a larger number of cores in the comparison. In addition, we used two different CPU types (Figure 4).

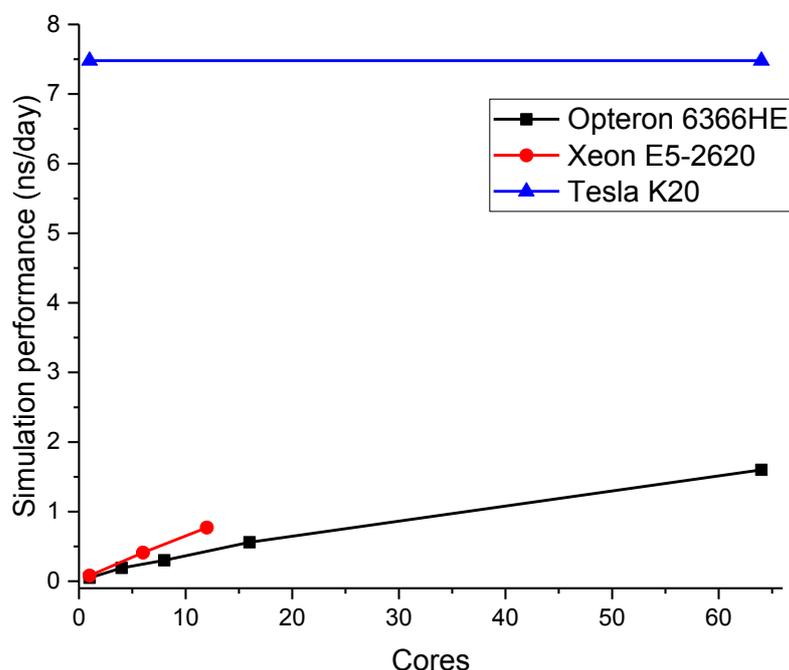


Figure 4. Extended comparison of the performance achieved using single/multi-core CPUs (Intel™ Xeon™ CPU E5-2620 and AMD™ Opteron™ 6366-HE) vs one GPU card (NVIDIA™ Tesla™ K20). The graph reports the nanoseconds of simulation that can be computed in one day as a function of the number of cores used on each CPU type.

It appears that the correlation with the number of cores is actually less than optimal on either CPU type. For the Intel™ CPU, the simulation performance (ns/day of simulation) as a function of the number of cores used is 1:5.1:9.6, against a ratio of 1:6:12 cores; for the AMD™ CPU the corresponding data are 1:11.2:32 against a ratio of 1:16:64 cores (Table 1, compare the first and third columns). Thus, using a 64-core system constituted by four AMD™ Opteron™ 6366-HE CPUs (1.8GHz, 16C, cache L2 16MB/L3 16MB, 32x4GB RDIMM LV dual rank memory) on a single blade provides about 50% of the expected increase in simulation length that can be computed per day with respect to a single core. **The acceleration provided by a single GPU card with respect to the full 64 core system is 4.7** (Table 1, last column).

Table 1. Benchmark results for AMBER, on unrestrained MD simulations for the ferritin system. The table reports the performance achieved using only single/multi-core CPUs (Intel™ Xeon™ CPU E5-2620 and AMD™ Opteron™ 6366-HE) vs the same cores plus one GPU card (NVIDIA™ Tesla™ K20m), measured by the nanoseconds of simulation that can be computed in one day for

the various hardware configurations. The table also reports the scale factor resulting from the use of multiple cores vs. a single core as well as acceleration provided by the GPU card.

Number of cores	Simulation performance in ns/day	Ratio vs. single-core	GPU acceleration
AMD Opteron			
1	0.05	1.00	150x
4	0.18	3.6	41.7x
8	0.30	6.0	25.0x
16	0.56	11.2	13.4x
64	1.6	32	4.69x
Intel™ Xeon™ CPU E5-2620			
1	0.08	1.00	93.7x
6	0.41	5.1	18.3x
12	0.77	9.6	9.74x
GPU card			
1	7.5	-	-

3 Implementation and benchmarking of GROMACS

Gromacs² [3] is an open-source molecular dynamics package designed for simulations of biomolecules, and developed with specific focus on computational performance allowing long simulations of large-scale systems.

In particular, GPU acceleration was introduced in version 4.5 and extensively redesigned in 4.6. The essential idea is offloading the expensive non-bonded force calculation to GPU while keeping the bonded forces and PME summation on CPU. In this way the optimal load balancing can be achieved. However, thorough optimization of this load split enforces introduction of a new *verlet* scheme of distance cutoff computation³. At variance with the AMBER package, GROMACS cannot be run solely on GPU cards. Within the WeNMR project [1], we have previously developed a grid-based web portal to enable users to setup and run in simple and user-friendly manner MD simulations using Gromacs [4]. The version of Gromacs implemented in the portal runs on the grid using multi-threading capabilities requiring typically six CPU cores. The portal could greatly benefit in the future from GPGPUs once they become available in the EGI production infrastructure.

3.1 Software and experiment setup

We decided to exploit the GROMACS benchmark to obtain information also on the usefulness of using GPGPUs in a cloud environment, instead of via the grid.

The current version 5.1 of Gromacs was compiled with gcc-4.8, using internal build of FFTW library, the best SIMD configuration for the machine (AVX_256), and GPU support.

3.2 Unrestrained MD benchmark

We ran the benchmark with the standard GPU benchmark inputs⁴. This suite consists of four molecules of different size:

- **Villin headpiece** – tiny protein fragment of just 35 residues. In general, MD of small molecules is difficult to accelerate significantly due to modest workload for the GPU.
- **Ribonuclease ZF-1A** – moderate size protein of 126 residues
- **Alcohol dehydrogenase (1YKF)** – a bigger protein (1408 residues in 4 chains) which should generate enough workload.
- **Ferritin** – see section 2.2

All the simulations were run in explicit solvent, using a dodecahedron box and PME electrostatics. They all ran for 10,000 steps of 5 fs.

² <http://www.gromacs.org/>

³ http://www.gromacs.org/GPU_acceleration

⁴ http://www.gromacs.org/GPU_acceleration

Table2 and Figure 5 summarize the performance of the computation (expressed in ns/day) in different CPU-only and CPU + GPU setups. All were repeated 3 times and the best figures were taken (this is a common method to eliminate interference with uncontrollable activities like disk or network background load of the physical machine).

Table 2. Benchmark results for GROMACS, on four different protein systems of increasing size. The table reports the performance achieved using only single/multi-core CPUs (Intel™ Xeon™ CPU E5-2650v2 @ 2.6 GHz) vs the same cores plus one GPU card (NVIDIA™ Tesla™ K20m), measured by the nanoseconds of simulation that can be computed in one day for the various computation setups. The acceleration provided by the GPU card is also reported.

Dataset	Protein size (aa)	Simulation performance in ns/day						GPU Acceleration		
		1 core	4 cores	8 cores	1 core + GPU	4 cores + GPU	8 cores + GPU	1 core	4 cores	8 cores
Villin	35	61	193	301	264	550	650	4.3x	2.8x	2.2x
RNAse	126	19	64	105	87	185	257	4.6x	2.9x	2.4x
ADH	1,408	3.1	11	18	13	38	50	4.2x	3.5x	2.8x
Ferritin	4,200	-	1.1	2.1	-	4.1	5.9	-	3.7x	2.8x

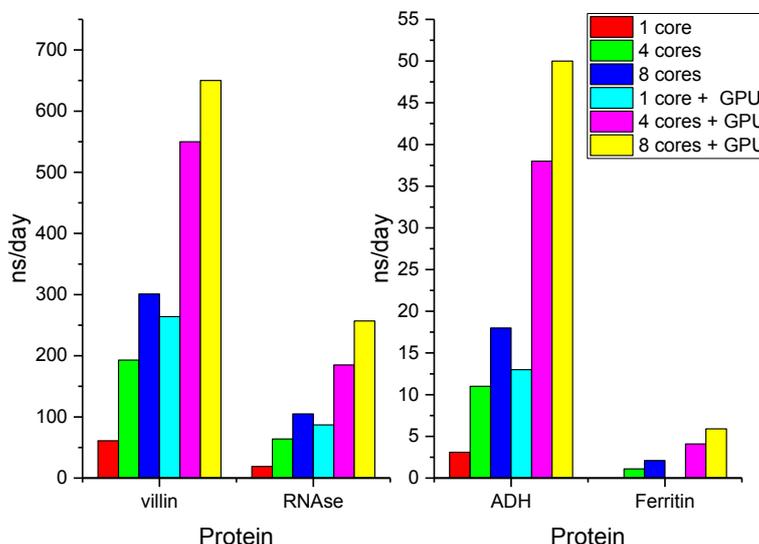


Figure 5. Comparison of the performance achieved using only single/multi-core CPUs (Intel™ Xeon™ CPU E5-2650v2 @ 2.6 GHz) vs the same cores plus one GPU card (NVIDIA™ Tesla™ K20m). Four different protein systems have been tested (see also Table 1). The graph reports the nanoseconds of simulation that can be computed in one day as a function of the setup.

The measurements are internally consistent. The acceleration is larger for single-core than four- and eight-core systems. For single-cores it does not depend significantly on the protein size. Instead, when using four and eight cores the acceleration increases with increasing molecular size, owing to the increased complexity of the calculation. At variance with AMBER, which makes use essentially only of the GPGPU card to compute the simulations, in GROMACS it is only the heavy calculation of non-bonded forces (i.e. of interactions between atom pairs) that is performed on the accelerator, while the CPU computes bonded forces and lattice summation (PME) in the same time. The number of atom pairs grows quadratically with increasing molecular size, therefore the larger systems experience an enhanced GPGPU contribution to the simulation [5].

When all the eight cores of the Intel™ Xeon™ CPU E5-2650 CPU are used, we get approx. an **acceleration factor of 2.2** for the small molecules and **2.8** for the larger ADH and ferritin systems. In the simulation using eight cores, the smallest molecule (villin) does not generate enough load (as indicated also by the CPU-only experiments) to fully benefit of the acceleration. Only the calculations on ferritin, the largest molecule in this benchmark, scale up nearly linearly with the number of cores. The relative GPU acceleration becomes lower with increasing number of CPU-cores due to the efficient load-balancing in Gromacs. However, the overall speedup is still apparent.

Our acceleration factors correspond well to those reported on the GROMACS web site (despite the absolute performance is different because of the different hardware used). This allows us not only to state that the present computation meets the expectation, but also that **the cloud environment does not impose a significant overhead**. This is also a very important conclusion of this work.

4 Other software tools for Structural Biology

4.1 PowerFit

PowerFit is a software, developed by the Bonvin group, for automatic rigid body fitting of biomolecular structures in Cryo-Electron Microscopy (Cryo-EM) densities (Figure 6) [6]. PowerFit is a Python package and a simple command-line program, able to make use of single/multiple CPUs or GPUs to accelerate the calculations. The inputs (Protein Data Bank, PDB, Cryo-EM data) and outputs (PDB, log files etc.) for PowerFit are all text files⁵. The execution of PowerFit as a container for the Docker software with GPGPUs has been successfully demonstrated on the CIRMMP test-bed.

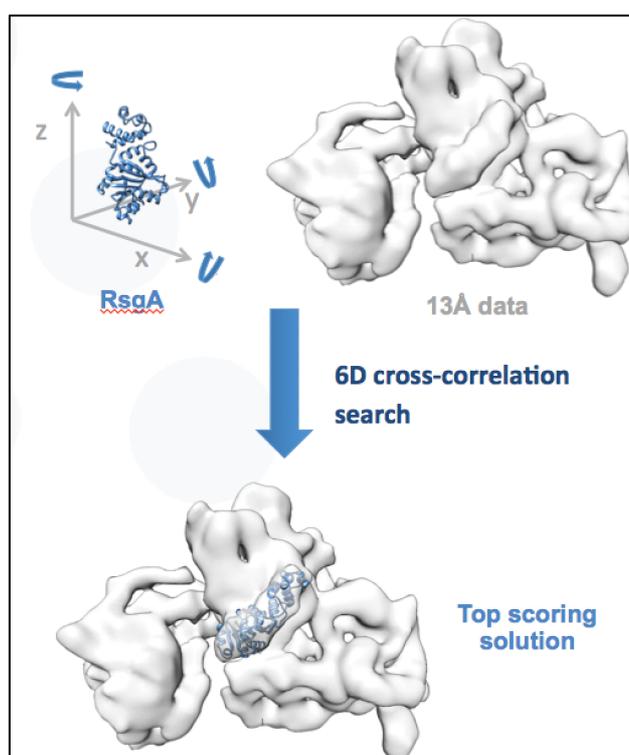


Figure 6. Fitting high resolution X-ray structure into low resolution Cryo-EM data using PowerFit.

The PowerFit tool was benchmarked by comparing the performance on a single-core AMD Opteron™ CPU to that observed on a GPU from the test-bed. For the latter, we exploited **direct submission via gLite to the GPGPU-enable CREAM-CE in Florence**. Technical details on the scripts

⁵ www.github.com/haddock/powertfit

and commands used for this can be found on the MoBrain website⁶. Table 3 and figure 7 show the results obtained for two different molecular systems.

Table 3 – Results of the PowerFit benchmark

System	Map size (voxels)	Rotations sampled	CPU time (s)	GPU time (s)	Acceleration
GroEL-GroES	90 x 72 x 72	70,728	5,340	249	21x
RsgA into ribosome	72 x 80 x 72	70,728	4,560	242	19x

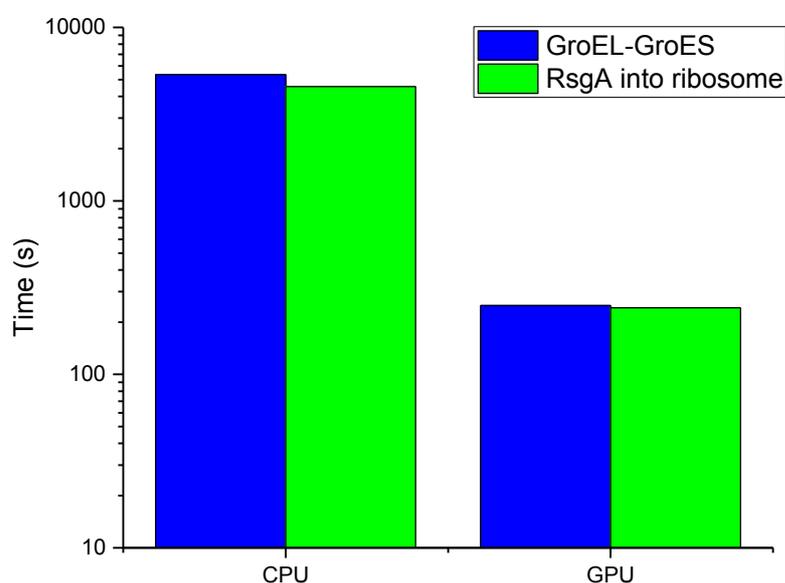


Figure 7 – Calculation times for Powerfit calculations on two model systems on a single-core CPU vs one GPU card. Note the logarithmic scale of the y axis.

We then extended this benchmark further to compare how the performance improves by using multi-core CPUs. The comparison with the performance achieved on the single GPU is reported in Figure 8.

⁶ <https://mobrain.egi.eu/technical>

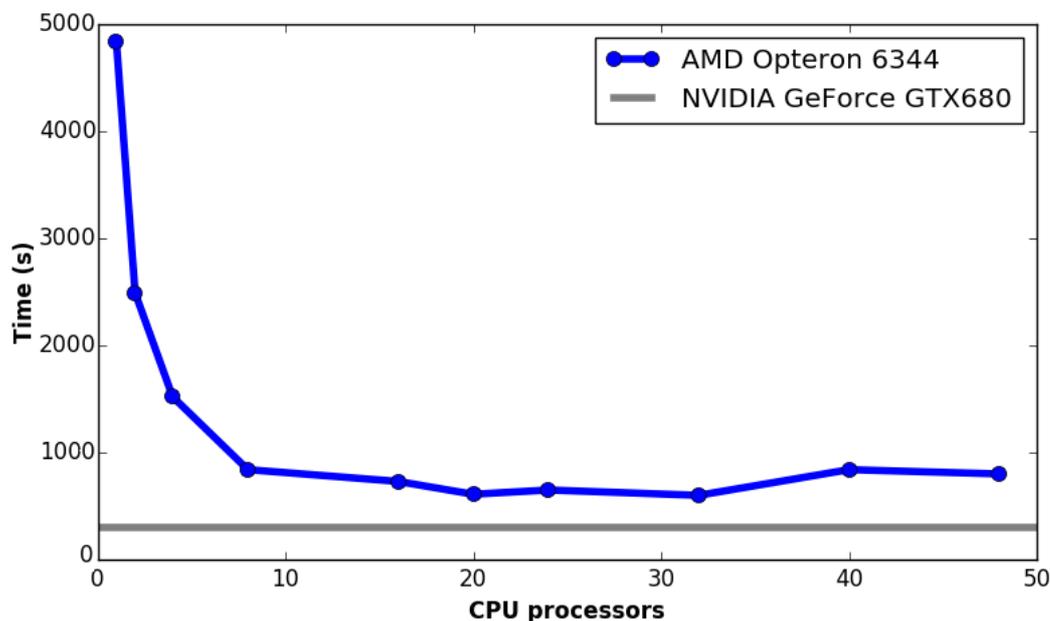


Figure 8 - Performance trend for PowerFit on single/multi-core CPUs (AMD Opteron™ 6344, 48 cores) and GPGPU (NVIDIA GeForce GTX 680), using the RsgA-ribosome test system

We observed that the performance of the software does not improve any more when more than 16 cores are used. Thus for both molecular systems analysed the **acceleration was about a 20-fold with respect to a single core**. We still observed a **2-fold acceleration with respect to 16-32 cores**.

4.2 DisVis

DisVis is a software tool developed by Bonvin group, for the visualization and quantification of the accessible interaction space of distance restrained binary biomolecular complexes (Figure 9) [7]. Like PowerFit, DisVis is a Python package and a simple command-line program, with the ability of harnessing multiple CPUs and GPU. The inputs and outputs for DisVis are all text files⁷.

⁷ www.github.com/haddock/diavis

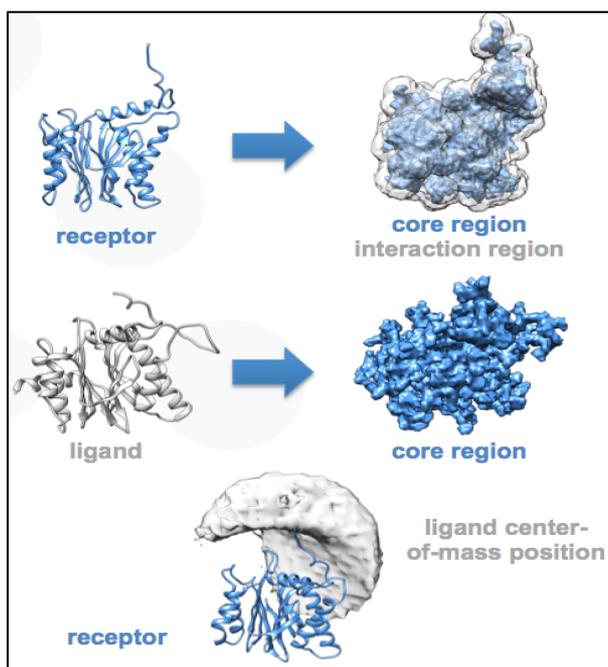


Figure 9. Visualization and quantification of the accessible interaction space for two proteins, using DisVis.

The benchmark for DisVis was computed in the same way as for PowerFit (section 4.1). Table 4 and figure 10 summarize the results.

Table 4 – Results of the DisVis benchmark

System	Number of complexes sampled	CPU time (m)	GPU time (m)	Acceleration
RNA polymerase II	$19 \cdot 10^9$	1,184	56	21x
PRE5-PUP2	$7 \cdot 10^9$	432	15	29x

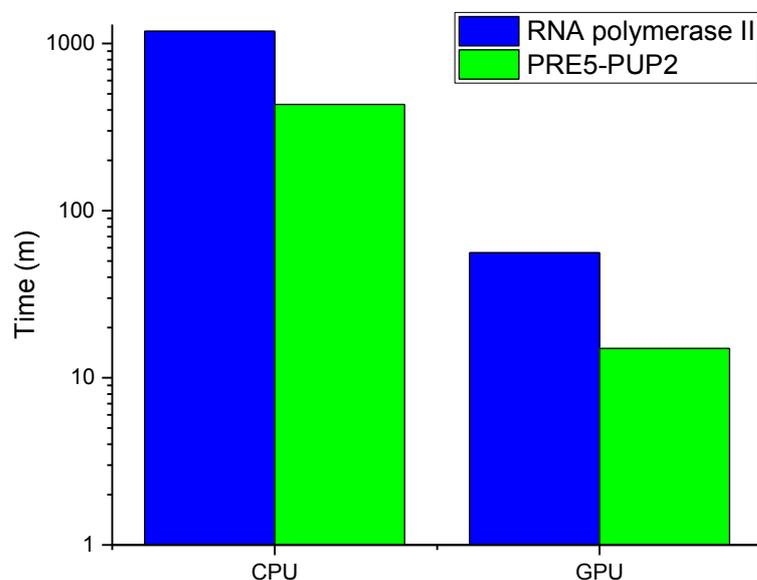


Figure 10 – Calculation times for DisVis calculations on two model systems on a single-core CPU vs one GPU card. Note the logarithmic scale of the y-axis.

The DisVis benchmark was further extended to compare how the performance improves by using multi-core CPUs. Figure 11 compares the performance with a varying number of CPU cores to the performance achieved on the single GPU card. As already observed for PowerFit, by using multi-core CPUs the performance improves only until 16 cores and remains unchanged when more cores are involved. Under these conditions, the **GPU card still provides a 2-fold acceleration**. We expect however that further optimisation of the code for GPUs (both for PowerFit and DisVis) should deliver an increased performance.

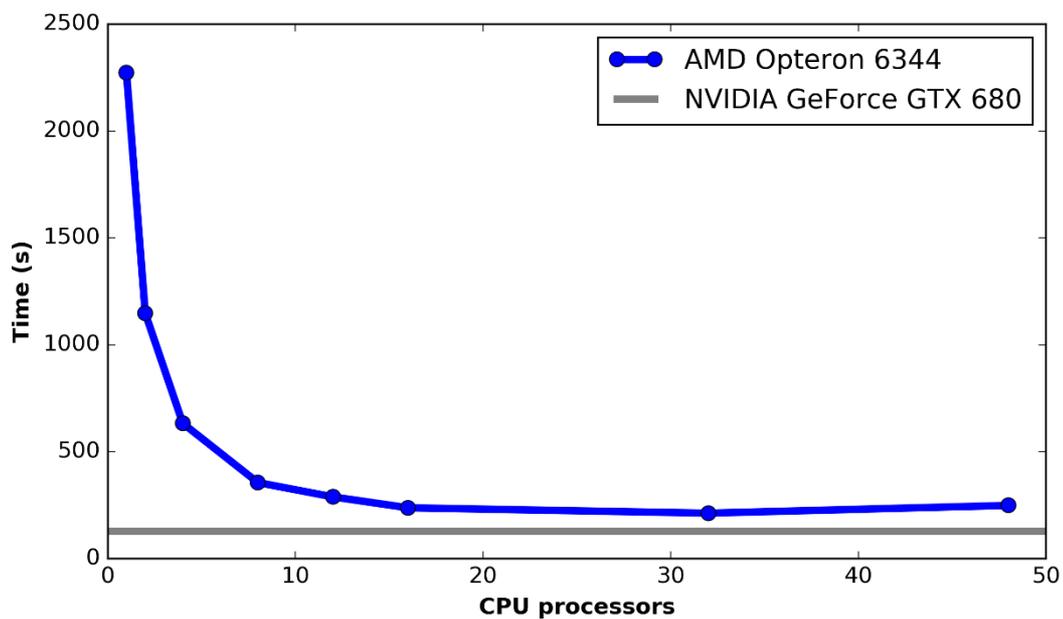


Figure 11 - Performance trend for DisVis on single/multi-core CPUs (AMD Opteron™ 6344, 48 cores) and GPGPU (NVIDIA GeForce GTX 680), using the PRE5-PUP2 complex test system

5 Conclusions

We demonstrated that **MD simulations benefit substantially of GPGPUs for the popular AMBER and GROMACS packages**. Our choice to benchmark these two software for MD was motivated by our long-term experience in using and helping others to use these tools, via the corresponding dedicated web portals [2, 4]. Of course, there are other widespread programs for MD simulations, which we did not address here. We do expect that the results observed here are relevant for all such programs, to an extent that depends subtly on the implementation of the algorithm and on the possible optimization for GPGPUs. This is true, for example, for the NAMD program [8]. Because of the different optimization of the code, the AMBER software has a greater improvement than GROMACS for the same molecular system (approximately 15-fold with respect to 3-fold speedup for ferritin when using eight CPU cores). Other tools, which are devoted to different applications in structural biology, also experienced a significant gain even without any optimization for GPUs. Overall, GPGPUs provided an acceleration with respect to 32-core CPUs ranging from a 2-fold (for DisVis and PowerFit) up to an 8-fold for MD simulations with AMBER.

We also checked whether the calculations run on GPGPUs provided the same results as the calculations on CPUs, as deviations might be caused in principle by the lack of double precision on the former platform. For restrained MD simulations, the comparison to experimental data provides a straightforward, independent validation of the calculation outputs. Exactly the same level of agreement was observed for our benchmark (section 2.1). For unrestrained simulations, experimental data are not available. We therefore need to compare one simulated trajectory to another. All pairs of simulations (GPGPUs vs CPUs) spanned the same range of energetic and structural fluctuations as a function of time, again indicating that there is substantial convergence between the corresponding outputs. For the other tools described in section 4, the comparison can be done by looking at the number of complexes computed and their associated score values. In all cases the differences were of the order of 0.1% or lower, which is completely negligible with respect to the intended use of the tools. Thus, we can conclude that the use of **GPGPUs did not affect the output of the calculations** for all tools and applications benchmarked in this report.

6 Future work

The next step in the MoBrain activity on accelerated computing will be to enable one or more portals for GPGPU usage next to the “traditional” CPUs. At present, we foresee that all calculations will be run on the HTC platform, via gLite. For this, a missing upgrade in the middleware is the implementation of the GLUE2.1 schema that should be supported by the CREAM-CE infoprovider mechanism. The submission mechanism using CREAM-CE that we validated as part of our benchmarking activities should enable a straightforward transition from the test-bed to the GPGPU production infrastructure.

The GPGPU-enabled portals will be accessible also via the MoBrain web portal⁸.

⁸ <http://mobrain.egi.eu/>

7 References

1. T.A. Wassenaar, M. van Dijk, N. Loureiro-Ferreira, G. van der Schot, S.J. de Vries, C. Schmitz, J. van der Zwan, R. Boelens, A. Giachetti, L. Ferella, A. Rosato, I. Bertini, T. Herrmann, H.R.A. Jonker, A. Bagaria, V. Jaravine, P. Guntert, H. Schwalbe, W.F. Vranken, J.F. Doreleijers, G. Vriend, G.W. Vuister, D. Franke, A. Kikhney, D.I. Svergun, R. Fogh, J. Ionides, E.D. Laue, C. Spronk, S. Jurka, M. Verlato, S. Badoer, S. Dal Pra, M. Mazzucato, E. Frizziero and A.M.J.J. Bonvin WeNMR: Structural Biology on the Grid. *J. Grid. Comp.*, 10, 743-767 (2012). doi: 10.1007/s10723-012-9246-z
2. I. Bertini I, D.A. Case, L. Ferella, A. Giachetti, A. Rosato. A Grid-enabled web portal for NMR structure refinement with AMBER. *Bioinformatics*. 27, 2384-2390 (2011). doi: 10.1093/bioinformatics/btr415
3. M.J. Abraham, T. Murtolad, R. Schulz, S. Páll, J.C. Smith, B. Hess, E. Lindahl. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2, 19-25 (2015). doi:10.1016/j.softx.2015.06.001
4. M. van Dijk, T.A. Wassenaar, A.M.J.J. Bonvin. A flexible, grid-enabled web portal for GROMACS molecular dynamics simulations *J. Chem. Theo. Comput.*, 8, 3463-3472 (2012).
5. C. Kutzner, S. Páll, M. Fechner, A. Esztermann, B.L. de Groot, H. Grubmüller. Best bang for your buck: GPU nodes for GROMACS biomolecular simulations. *J. Comp. Chem.* 36, 1990–2008 (2015). doi:10.1002/jcc.24030
6. G.C.P. van Zundert, A.M.J.J. Bonvin. Fast and sensitive rigid-body fitting into cryo-EM density maps with PowerFit. *AIMS Biophysics*. 2, 73-87 (2015). doi: 10.3934/biophy.2015.2.73
7. G.C.P. van Zundert, A.M.J.J. Bonvin. DisVis: Quantifying and visualizing accessible interaction space of distance-restrained biomolecular complexes. *Bioinformatics*. 31, 3222-3224 (2015). doi: 10.1093/bioinformatics/btv333
8. D.E. Tanner, J.C. Phillips and K. Schulten. GPU/CPU Algorithm for Generalized Born/Solvent-Accessible Surface Area Implicit Solvent Calculations. *J. Chem. Theory Comput.* 8, 2521–2530 (2012). doi: 10.1021/ct3003089