

**EGI-Engage**

Implementation and evaluation of AMBER and/or GROMACS

|  |  |
| --- | --- |
| **Date** | 11 March 2016 |
| **Activity** | SA2 Knowledge Commons – Task SA2.5 MoBrain |
| **Lead Partner** | CIRMMP |
| **Document Status** | DRAFT |
| **Document Link** | https://documents.egi.eu/document/XXX |

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 as high as a 180 factor in the speed of MD calculations with respect to a single core. 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 16-core CPUs, the acceleration afforded by GPGPUs ranges from a 2-fold (for DisVis) up to a 10-fold for MD.

**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**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Name*** | ***Partner/Activity*** | ***Date*** |
| **From:** | Antonio Rosato | CIRMMP | 11/03/2016 |
| **Moderated by:** |  |  |  |
| **Reviewed by** |  |  |  |
| **Approved by:** |  |  |  |

**DOCUMENT LOG**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Issue*** | ***Date*** | ***Comment*** | ***Author/Partner*** |
| **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 Verlato /INFN |
| **v.4** | 10/03/2016 | Final draft | Antonio Rosato/CIRMMP |

**TERMINOLOGY**

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

**Contents**

[1 Introduction 5](#_Toc445130914)

[2 Implementation and benchmarking of AMBER 7](#_Toc445130915)

[2.1 rMD benchmark with the AMPS-NMR portal 7](#_Toc445130916)

[2.2 Unrestrained MD benchmark for a large macromolecular adduct 9](#_Toc445130917)

[3 Implementation and benchmarking of GROMACS 12](#_Toc445130918)

[3.1 Software and experiment setup 12](#_Toc445130919)

[3.2 Unrestrained MD benchmark 12](#_Toc445130920)

[4 Other software tools for Structural Biology 15](#_Toc445130921)

[4.1 PowerFit 15](#_Toc445130922)

[4.2 DisVis 16](#_Toc445130923)

[5 References 20](#_Toc445130924)

**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 as high as a 180 factor in the speed of calculations with respect to a single core 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 16-core CPUs, the acceleration afforded by GPGPUs ranges from a 2-fold (for DisVis) up to a 10-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.

# 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 typically described by reporting the molecular weight of the system, expressed in Daltons (Da). 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. Specifically, the focus has been on accelerators cards, mainly GPGPUs (General-Purpose computation on Graphics Processing Units). MoBrain has endeavored to demonstrate the usefulness of GPGPUs for the types of applications described in this section, and contributed also to the activity carried out to enable GPGPU support in the EGI HTC platform (see Deliverable D4.6). 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. And finally, we demonstrate that GPGPU-based computations can be successfully submitted with gLite to GPGPU supporting CREAM-CE.

# Implementation and benchmarking of AMBER

We installed the PMEMD tool of the AMBER package (<http://ambermd.org/>) on the GPGPU test-bed available at the CIRMMP data centre. The test bed has three nodes (2x Intel Xeon E5-2620v2) with two NVIDIA Tesla K20m GPUs per node. The test-bed was managed by the Torque 4.2.10 Local Resource Management System (LRMS, source compiled with NVIDIA NVML libraries) with the Maui 3.3.1 scheduler, and was installed with CUDA 5.5. The same test-bed has been used to develop a solution enabling GPGPU support in CREAM-CE.

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

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.

## 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

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.

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.

## 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

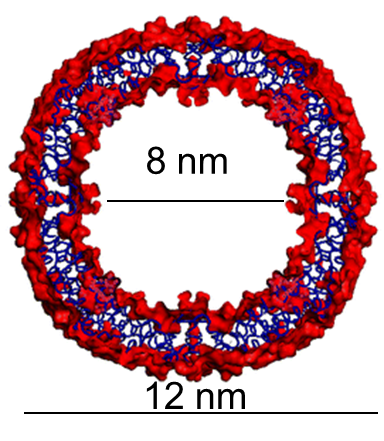
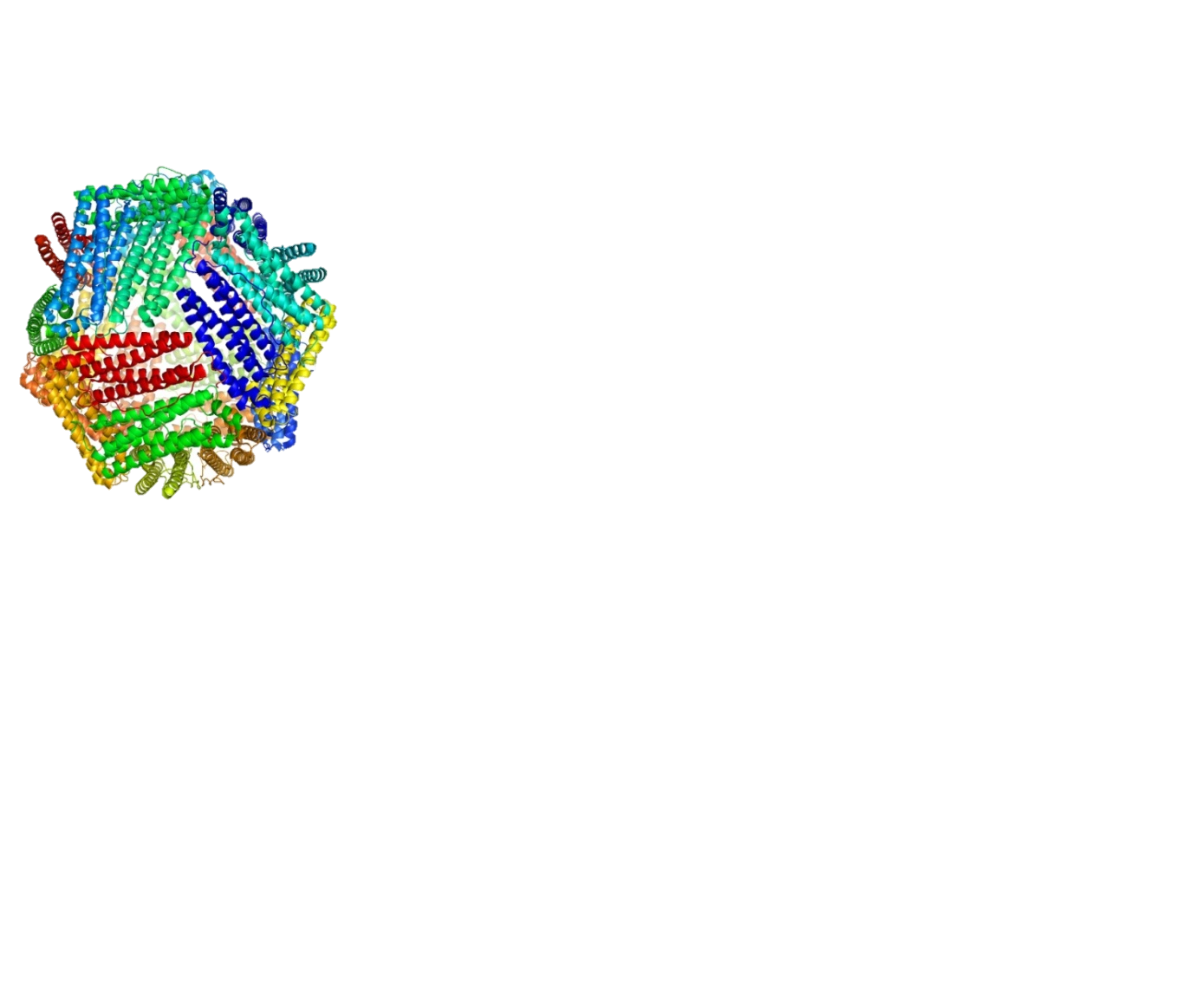
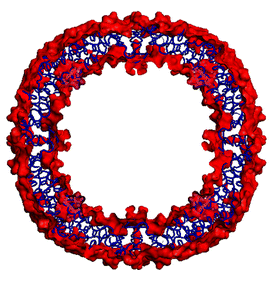


Figure 2. The architecture of ferritin. Left panel: ribbon representation in which each protein chain is coloured by a different colour. Right panel: the internal cavity of ferritin



8 nm

12 nm

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, but 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. 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 0,17 on 4 a 4-core CPU and 0.04 for a single core system. The time required to produce one ns of simulation was respectively of 1.15·104 s, 5.0·105 s and 1.93·106 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 = 167:3.86:1.00**. 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 ratio of the achieved simulation length as a function of the number of cores used is 1:5.1:9.6, against an expectation of 1:6:12; for the AMD™ CPU the corresponding data are 1:11.2:32 against an expectation of 1:16:64. Thus, using a full 64 core CPU 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 CPU is 4.7**.

# Implementation and benchmarking of GROMACS

Gromacs [3], <http://www.gromacs.org/> 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 computation. More details are given at <http://www.gromacs.org/GPU_acceleration>. 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.

## Software and experiment setup

All the experiments were run at the IISAS-GPUCloud - NGI\_SK EGI federated cloud site. The predefined “gpu.large” virtual machine flavour was used, mapping to 4 cores of E5-2650v2 @ 2.6 GHz, 8 GB RAM, and Nvidia Tesla K20m GPU.

We started with the EGI supported virtual machine image Ubuntu 14.04, brought up to date with the distribution repository, and installed the necessary Cuda version 7.5 development and runtime.

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 obviously.

## Unrestrained MD benchmark

We ran the benchmark with the standard GPU benchmark inputs available with GROMACS at <http://www.gromacs.org/GPU_acceleration>. 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 biggerprotein (1408 residues in 4 chains) which should generate enough workload.
* **Ferritin –** see section 2.2

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

Table 1 and Figure 5 summarize 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 1. 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Simulation performance in ns/day** | | | | **GPU Acceleration** | |
| **Dataset** | **Protein size (aa)** | **1 core** | **4 cores** | **1 core + GPU** | **4 cores + GPU** | **1 core** | **4 cores** |
| **Villin** | 35 | 61 | 193 | 264 | 550 | 4.3x | 2.8x |
| **RNAse** | 126 | 19 | 64 | 87 | 185 | 4.6x | 2.9x |
| **ADH** | 1,408 | 3.1 | 11 | 13 | 38 | 4.2x | 3.5x |
| **Ferritin** | 4,200 | - | 1.1 | - | 4.1 | - | 3.7x |



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 computation setup.

The measurements are consistent among one another. The acceleration is larger for single-core than four-core systems. For single-cores it does not depend significantly on the protein size. In the more realistic case where four cores are used, we get approx. an **acceleration factor of 2.9** for the small molecules and **3.5-3.7** for the larger ADH and ferritin systems. Notably, with four cores the acceleration increases with increasing molecular size, owing to the increased complexity of the calculation. With more atom-atom interactions to take into account, the GPU contribution to the simulation is enhanced [5].

The acceleration factors correspond well to those reported by GROMACS (despite the absolute performance is different because of using different CPU and GPU). 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.

# Other software tools for Structural Biology

## 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 files. It is available at [www.github.com/haddocking/powerfit](http://www.github.com/haddocking/powerfit). The execution of PowerFit as a container for the Docker software with GPGPUs has been successfully demonstrated on the CIRMMP test-bed.

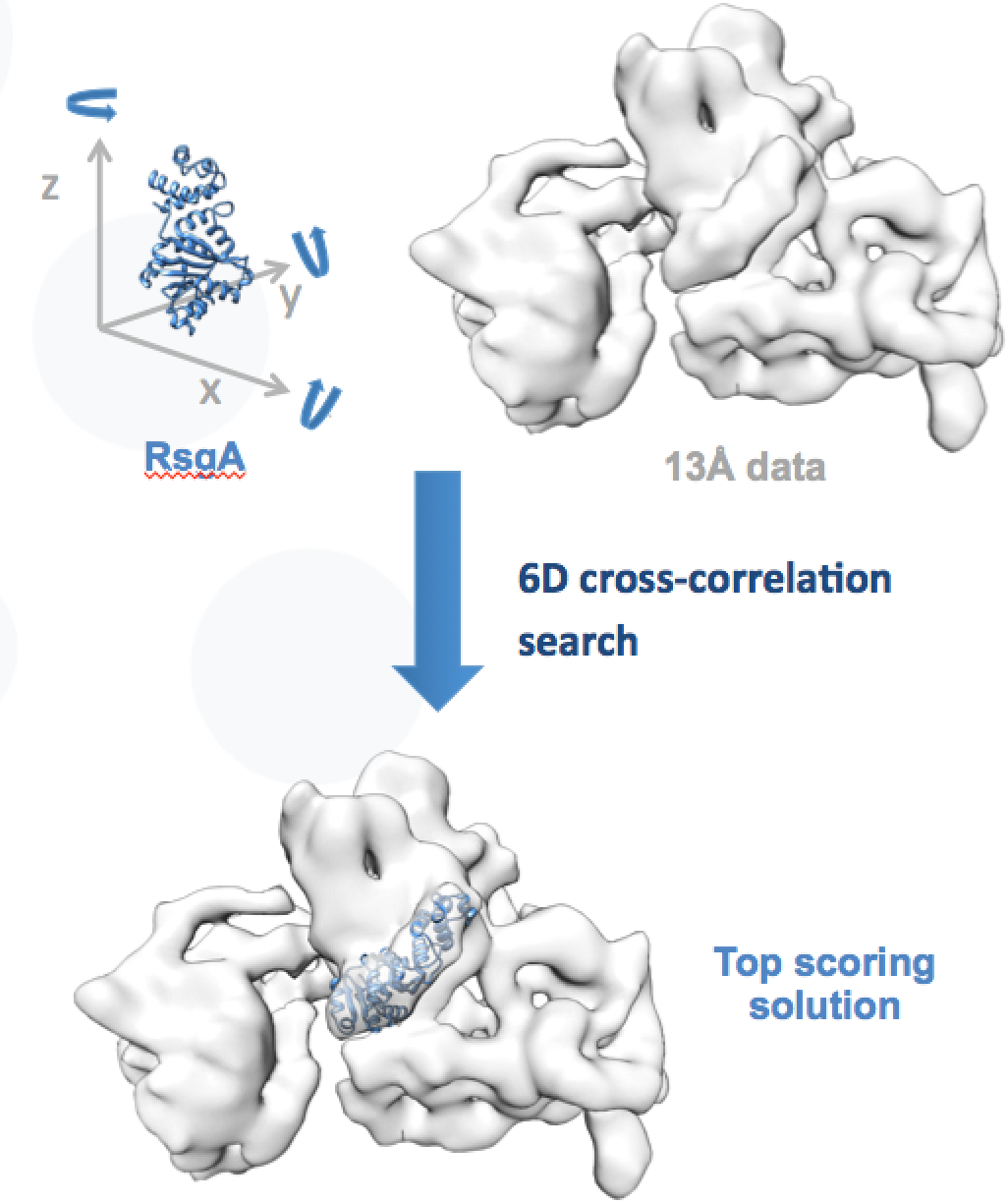
**

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 and commands used for this can be found on the MoBrain website (<https://mobrain.egi.eu/technical)>. The results are given in Table 2 and figure 7, for two different molecular systems.

***Table 2 – 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.

Thus for both molecular systems analysed the **acceleration was about a 20-fold**.

## 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 8) [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 files. It is available at [www.github.com/haddocking/disvis](http://www.github.com/haddocking/disvis).

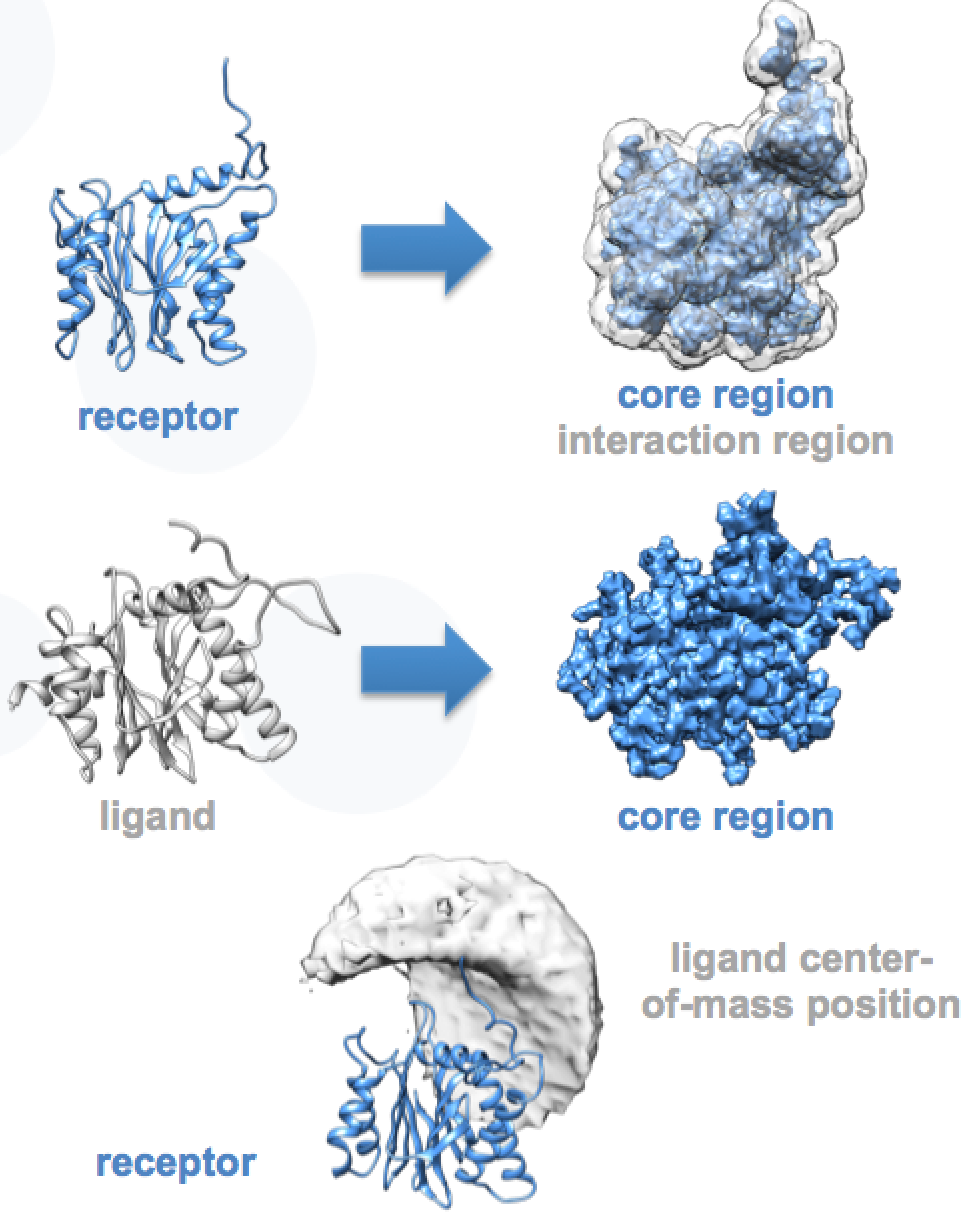


Figure 8. Visualization and quantification of the accessible interaction space for two proteins, using DisVis.

The benchmark for DisVis was computed in the same way as for PowerFit (section 4.1). The results are given in Table 3 and figure 9.

***Table 3 – Results of the DisVis benchmark***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | ***Number of complexes sampled*** | ***CPU time (m)*** | ***GPU time (m)*** | ***Acceleration*** |
| **RNA polymerase II** | 19·109 | 1,184 | 56 | 21x |
| **PRE5-PUP2** | 7·109 | 432 | 15 | 29x |



***Figure 9 – 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. The comparison with the performance achieved on the single GPU is reported in Figure 10.

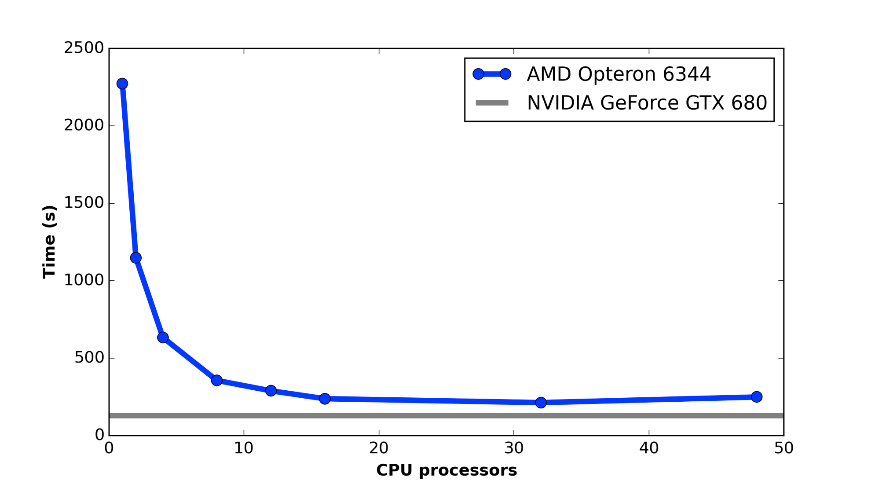


Figure 10 - Performance trend for DisVis on single/multi-core CPUs (AMD Opteron™ 6344, 48 cores) and GPGPU (NVIDIA GeForce GTX 680), using the PRE5-PUP2 complex test system

For both molecular systems analysed using DisVis, the acceleration achieved using a single GPU card with respect to a single-core CPU was **between 20- and 30-fold**. By using multi-core CPUs a performance improvement is observed up to 16 cores. Under these conditions, the GPU card still provides a **two-fold acceleration**. We expect however that further optimisation of the code for GPUs (both for PowerFit and DisVis) should deliver an increased performance.

# 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).
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 and 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 and A.M.J.J. Bonvin. Fast and sensitive rigid-body fitting into cryo-EM density maps with PowerFit. AIMS Biophysics. 2, 73-87 (2015).
7. G.C.P. van Zundert and A.M.J.J. Bonvin. DisVis: Quantifying and visualizing accessible interaction space of distance-restrained biomolecular complexes. Bioinformatics. 31, 3222-3224 (2015).