High-throughput all-atom molecular dynamics simulations using distributed computing

I. Buch,[†] M. J. Harvey,[‡] T. Giorgino,[†] D. P. Anderson,[¶] and G. De Fabritiis[*,†]

Computational Biochemistry and Biophysics Lab (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain, High Performance Computing Service, Information and Communications Technologies, Imperial College London, South Kensington, London, SW7 2AZ, UK, and Space Sciences Laboratory, University of California, Berkeley CA 94720, USA

E-mail: gianni.defabritiis@upf.edu

Abstract

Although molecular dynamics simulation methods are useful in the modeling of macromolecular systems, they remain computationally expensive, with production work requiring costly high-performance computing (HPC) resources. We review recent innovations in accelerating molecular dynamics on graphics processing units (GPUs), and we describe GPUGRID, a volunteer computing project that uses the GPU resources of non-dedicated desktop and workstation computers. In particular, we demonstrate the capability of simulating thousands of all-atom molecular trajectories generated at an average of 20 ns/day each (for systems of ≈30,000-80,000 atoms). In conjunction with a potential of mean force (PMF) protocol for computing binding free energies, we demonstrate the use of GPUGRID in the computation of accurate binding affinities of the Src SH2 domain/pYEEI ligand complex by reconstructing the PMF over 373 umbrella sampling windows of 55 ns each (20.5 μs of total data). We obtain a standard free energy of binding of -8.7 ± 0.4 kcal/mol within 0.7 kcal/mol from experimental results. This infrastructure will provide the basis for a robust system for high-throughput accurate binding affinity prediction.

Introduction

Bridging the gap from the molecular-atomistic (femtosecond) to biological (micro-millisecond) timescales remains an unsolved problem in computational biology. In the past 20 years, molecular modeling has advanced from simulating 300 atoms to the routine modeling of entire proteins in solution with lipids and explicit water (30,000-100,000 atoms). This remarkable achievement has been in large part due to the use of high-performance computing (HPC). However, simulations are still far from realizing the full potential of computational molecular biology, due to the limited timescale (usually sub $1\mu s$) that they can practically achieve. For example, the ability to rapidly compute thermodynamic estimations of free energies would be of great use in the *in silico* drug discovery process for virtual screening and lead optimization[1] but exact thermodynamic methods[2] using MD require long-running simulations, making the total computation time uncompetitive with direct experimental measurements. In order to be a practical complement to experimental techniques, a compu-

[*]To whom correspondence should be addressed

[†]Computational Biochemistry and Biophysics Lab (GRIB-IMIM), Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain

[‡]High Performance Computing Service, Information and Communications Technologies, Imperial College London, South Kensington, London, SW7 2AZ, UK

[¶]Space Sciences Laboratory, University of California, Berkeley CA 94720, USA

tational method that reduces the time-to-answer to the order of a few days is needed.

The recent evolution of commodity graphics processing units (GPUs) into general-purpose, fully-programmable, high-performance processors represents an important technological innovation which may realize the full potential of atomistic molecular modeling and simulation. GPUs can currently deliver over an order of magnitude more floating-point operations per second (FLOPS) than standard processors, roughly equivalent to a decade of Moore's law growth,[3] with an exponential divergence in performance compared with CPUs since their inception.[4] To exploit the computational power of GPUs it is necessary to redesign and reprogram algorithms to suit the architectures of these devices. In previous work, we have demonstrated a molecular dynamics simulation code, called ACEMD,[5] targeted to run on GPUs. This code achieves a performance equivalent to tens of contemporary processors on a single NVIDIA GTX280 GPU and to that of hundreds of processors on 3 GPUs. This represents a step change in the accessibility of molecular dynamics simulation, making it possible to perform on a workstation simulations that would previously have required dedicated HPC cluster or supercomputing systems. Paradoxically, because GPUs represent a disruptive technological change, few dedicated HPC services have yet incorporated them into their systems, and the majority of GPU devices are sold to the home and desktop market (in large part, the swift development of these devices has been driven by the demands of the computer entertainment market).

This circumstance provided us with a unique opportunity to build a distributed computing system for molecular dynamics simulations which could be used to achieve comparable performance to a typical HPC cluster, but with the capacity and cost efficiency of distributed computing. In general, distributed computing is best-suited to problems which have a high degree of intrinsic parallelism, of the order of at least thousands of jobs. For example, the computing model of SETI@home involves many small jobs; the goal is high throughput rather than low latency. In contrast, we sought to integrate GPUGRID as an experimental tool into the iterative cycle of hypothesis, experiment and analysis and so placed additional emphasis on reducing the time taken to complete the computational experiment phase.

GPUGRID was implemented using the BOINC framework for volunteer computing.[6] To date, the computational capability accessible via GPUGRID has grown to a peak performance in excess of 1 PFLOPS (10^{15} floating-point operations per second), with a consistent sustained performance of 60 TFLOPS, resulting in GPUGRID becoming one of the most computationally powerful volunteer computing projects after only two years of activity.[7]

In this paper we discuss the GPUGRID infrastructure and the challenges encountered in adapting the BOINC middleware to accommodate the novel aspects of GPU use. We also describe the molecular dynamics protocols used by ACEMD and their application to the specific example of calculating the binding affinity of Src homology 2 (SH2) domain/pYEEI ligand complex, which should serve as the basis for future high-throughput free energy calculations.

Methods

Middleware support for GPU computing

We sought to exploit the GPUs installed not only in our own institutional computers, but in those of the volunteers from the general public. To this end, we developed the GPUGRID project to allow computer owners to volunteer their resources for use by our research. This approach, called "volunteer computing", has been used by many other projects over the last 10 years. Currently roughly 500,000 people and 1 million computers participate in volunteer computing projects.

We chose to base GPUGRID on BOINC, a mature middleware platform for volunteer computing which is used by about 50 projects, including SETI@home,[8] IBM World Community Grid,[9] Einstein@home, Rosetta@home, and Climateprediction.net. Folding@home is another project which in fact pioneered the use of accelerator processors, Cell processors and GPUs,[10] in a volunteer computing context; however, it is based on an

internally developed middleware. To contribute to a BOINC project, volunteers must download and run a client program (available for all common operating systems) and "attach" it to the desired project. BOINC allows and encourages volunteers to participate in multiple projects. This feature was particularly attractive as the GPUGRID application uses the GPU almost exclusively and does not require much CPU time. Thus, a client machine may run GPUGRID jobs on its GPUs while simultaneously processing CPU-bound jobs for other projects.

Significant changes to BOINC were necessary to accommodate the needs of GPUGRID. BOINC originally supported only single-process CPU applications. PS3GRID, our previous project harnessing the Cell processor,[11] used this version of BOINC to distribute applications that ran on the Playstation's Cell processor. This arrangement worked only because no other BOINC projects had applications for the PS3. However, as other BOINC projects besides GPUGRID were preparing GPU-based applications, GPUs had to be made "first-class citizens" in BOINC.

We sought to extend BOINC to handle GPUs and, in the future, to handle other types of accelerator processors, multithreaded CPU applications, and applications using arbitrary combinations of accelerator processors and CPUs. This required major changes to the BOINC client, the server, and the protocol by which they communicate.

In BOINC, an application may have versions for different platforms (e.g., Win32, Mac OS X, Linux/Intel32, etc.). When a job is submitted it is associated with an application, not with a particular version. We generalized this model by allowing multiple versions for a single platform. For a given platform, for example, there might be one version for a GPU, another for a multicore system using multiple threads, and a third optimized for uniprocessors. BOINC may run several different versions on a host to fully utilize its resources.

The BOINC client was modified to detect the presence of GPUs, together with their hardware characteristics (clock rate, video RAM, number of processors) and their driver software version; this information is reported to the server. In addition, two parts of the BOINC client were rewritten to accommodate GPU applications. The first is the mechanism that decides when to fetch work and which project to fetch it from. This mechanism now estimates the queue length for each processing resource type (CPU, GPU, etc.) and tracks which projects have work for which resource types. When the queue length for a resource falls below a certain level, work for that resource is requested from a project that is likely to have it. The second affected part is the client's job scheduling policy, which selects a set of jobs to run. It now gives priority to GPU jobs, schedules them in a way that minimizes preemption, and passes them a command-line argument indicating which GPU devices to use. It runs GPU jobs at normal operating system priority; the CPU portion of a GPU job typically uses little time, but if it runs at low priority the entire job runs inefficiently. By default, GPU jobs are not run while the computer is in use; this is necessary to avoid impacting the performance of GUI interactions.

Similarly, the BOINC server was modified to handle GPU jobs. In handling a request, the server scans a list of available jobs and selects a set which will use the requested amount of time and which meet various other criteria (e.g., they are likely to complete by the job's deadline, the client has sufficient RAM and disk, etc.). As part of this decision, the server must decide which application version to use for each job, and must determine the resource usage (number of GPUs and CPU cores used) and the estimated FLOPS performance. This decision may be complex and application-specific, so its logic is placed in a project-supplied "planning function" that is linked with the BOINC scheduler. This function takes as input a host description and an application version. It decides whether the application version can run on the host, and if so, what CPU and coprocessor resources it will use, how many FLOPS it will achieve, and what command-line argument should be passed to it. For GPU applications, this logic can enforces constraints on GPU speed, video RAM size, or driver version number. The BOINC server, to decide what version to use for a given job, calls the application planning function for each available version, and chooses the one for which the predicted FLOPS is greatest. A version is skipped if it uses coprocessor resources for which the client is not requesting work. If a client

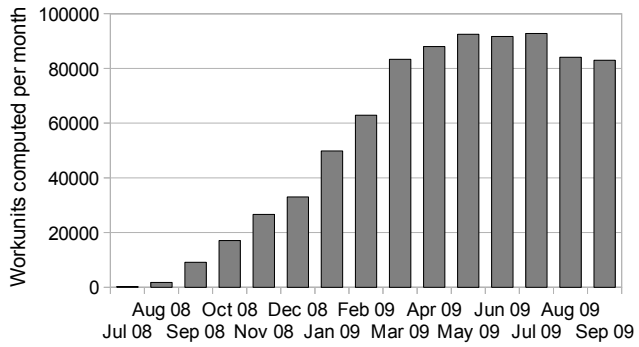

Figure 1: Number of jobs per month computed by GPUGRID from July 2008 until September 2009. For systems of $\sim$ 50,000 atoms, one job corresponds to approximately 2.3 ns of simulated time. Therefore, GPUGRID currently samples approximately 6 μs per day.

requests both CPU and GPU work, and the GPU part of the request has been satisfied, the scheduler will associate subsequent jobs with a CPU version.

Molecular simulations in GPUGRID

With molecular dynamics simulation methods one can model molecular systems so that all atoms are treated as point charges which interact with their neighbors according to the forces defined by a force field. The force field is carefully parametrized to reflect the chemical environment of the particles within the molecules. Commonly used force fields such as CHARMM[13] and AMBER,[14] which are designed specifically to model biomolecular systems, contain parametrizations for specific bond, angle and dihedral terms. In addition to these bonded terms and in order to correctly capture the dynamics of a system, it is necessary to model the long-range electrostatic and van der Waals (vdW) interactions. Various techniques, such as the particle-mesh Ewald method[15] exist for approximating the contribution of long-range forces beyond a specific cutoff but, nevertheless, the evaluation of the non-bonded force terms constitutes almost the entire computational cost of biomolecular MD simulation. Because MD uses an explicit time integration scheme[16] with a time step of less than 4 fs, a 10 ns simulation – typical of production work – may still take a day to complete on high-performance computing resources. ACEMD[5] permits such production simulations to be performed on a single GPU within a day. GPUGRID is being used so far in the study of six molecular systems, shown in Table 1.

In order to efficiently use a distributed computational infrastructure, the use of suitable simulation protocols is crucial. As previously described, GPUGRID's computing nodes are personal computers that are managed by volunteers, and have no direct communication with other nodes. These facts impose constraints on how simulations are run in GPUGRID. First, a simulation protocol that is able to deploy enough runs in parallel is needed to efficiently exploit the computational resources. Second, to automate as much as possible the daily management and analysis of tens of gigabytes of data, proper workflows, involving as little human intervention as possible, are needed.

A single computational experiment for binding affinities is composed of hundreds of shorter simulations (1-100 ns) that are distributed to the GPUGRID nodes. We will refer to these individual simulations as 'jobs'. When the computation of a job finishes, the results are returned to the server and the job can either terminate or continue. When a job reaches the target amount of simulated time it finishes and is ready for result retrieval. If the simulation needs to be run for a longer time than the set up for single jobs, the returned job is continued by resubmission to a different host. This strategy allows us to use longer simulated times, up to 100 ns for system sizes of even 80,000 atoms, without burdening each user with longer computing times per job.

During the simulation, several files are exchanged between the client and the project's data server, over the Internet. For molecular simulations, files downloaded by the clients typically include PDB files, force field parameters, coordinates and velocities of intermediate configurations, and any other parameters related to the experimental conditions. If the job is finished successfully, the results are uploaded to the server; results typically include coordinates and velocities (used to continue the simulation on other hosts), log files (for analysis) and, if required, trajectories of individual atoms. The transfer of these files

Table 1: Simulations running on GPUGRID (as of October 2009). Note that data sampled refers to cumulative simulations run for the specific system.

Date	System	PDB code	No. of atoms	Data sampled	Description
2008	Gramicidin A	1JNO	29,042	$\sim 22\ \mu$s	K^+ ion translocation through a transmembrane Gramicidin-A channel[12]
2008	Triosephosphate Isomerase	1NEY	83,999	$\sim 11\ \mu$s	Conformational differences arising when TPI enzymes undergo tyrosine nitration
2008	GPCR Dopamine receptor D2	(Homology)	60,733	$\sim 69\ \mu$s	Mobility of Na^+ ions and its effect on the dynamic properties of the D2 receptor under physiological ionic strength conditions
2009	HIV-1 Protease	1HHP	55,342	$\sim 230\ \mu$s	Maturation and flexibility mechanisms for protease in its dimer form
2009	hERG K^+ channel	(Homology)	43,874	$\sim 31\ \mu$s	Potassium channel blockage by cardiotoxic drugs
2009	SH2-pYEEI complex	1LKK	38,655	$\sim 905\ \mu$s	Calculation of free energy of protein-ligand binding

taxes the users' Internet connections individually, and the aggregate amount of data exchanged requires that the institution hosting the server provides a suitable bandwidth. For GPUGRID, daily bandwidth usage at the server averages between 0.5-1 MB/s.

GPUGRID performance

We quantified the performance of the GPUGRID system, characterizing the hosts that have been computing for us at the time of writing. We took into account only *active* hosts, defined as hosts which computed and returned at least one result in the period of time of two weeks.

During the period considered there were 1,370 active hosts. Figure 2 shows the distribution of GPU processing power installed per host. The peak GFLOPS figure was computed for each card considering the number of processing cores and its clock frequency. The majority of the hosts had an installed GPU capacity of approximately 500 GFLOPs, consistent with the current distribution of consumer cards. At the time of writing, the fastest GPU available had a peak processing rate of approximately 1 TFLOPS. Some of the hosts show a higher installed computing power, which is achieved by mounting multiple cards in the same machine (inset of Figure 2).

At the time of this analysis, the total theoretical peak performance for the GPUGRID project is 1.021 PFLOPS. This performance would be achieved if all the active hosts were computing at the same time; in practice, the figure should be adjusted by the fraction of time that each card is devoted to the project. This factor may be influenced by two reasons: (a) some users are sharing the card with other BOINC GPU projects; and (b) some users do not share the resource for part of the time, because the PC is turned off, or they require the GPU for other purposes.

Latency problems. During operation of GPUGRID, it was observed that the effective throughput was often affected by delays in the processing of some jobs, as shown in the distribution in Figure 3. This is attributed to the intrinsic uncertain availability, and varying capabilities, of the client machines (Figure 2). To compensate for this, GPUGRID supports a load balancing algorithm which re-assigns the results that fall behind schedule to more reliable hosts. By checking the progress of each task against a projection,

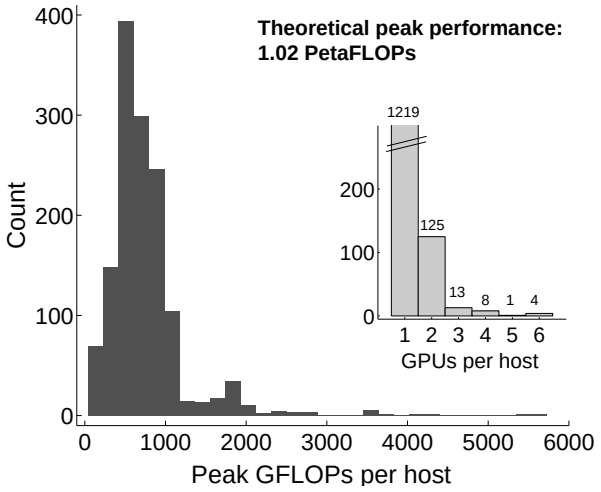

Figure 2: Distribution of GPU peak performance installed in active hosts of GPUGRID (main plot) and GPUs installed per host (inset). During the period considered, 1,370 active hosts provided a theoretical processing power (to be adjusted by the fraction of resource shared) of 1.02 PetaFLOPs.

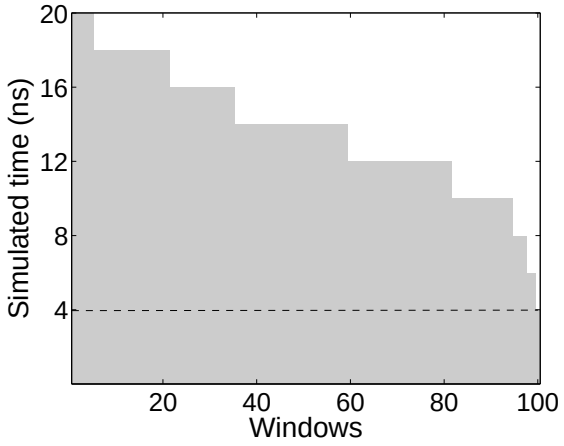

Figure 3: A snapshot of the results distribution taken during an umbrella sampling simulation. "Windows" (horizontal axis) are independent simulations, each running until 20 ns of simulated time. Due to the distribution of jobs (2 ns each) to heterogeneous machines, some windows would complete faster than others unless a load-balancing algorithm is enabled. Analysis is only possible up to a time that all windows have computed (dashed line).

any tasks which are deemed to be running behind schedule are rerun with high priority on client machines which have a high-specification GPU and judged to be reliable, based on their recent processing history.

Failures. It is inevitable that any distributed system, particularly one comprising non-dedicated components, will suffer failures. The application would fail non-deterministically during execution, typically issuing a catchall "unknown error in kernel execution" error. Upon inspection of the configuration of these hosts, it was found that the GPUs were "overclocked" *i.e.* the clock frequencies of the processors had been increased beyond the manufacturer's recommendations in order to improve performance. On many occasions this had been done intentionally by the systems' owner but it was noted that some vendors sold "factory-overclocked" parts. Owners of such devices reported that they operated stably under normal (*i.e.* non-computational) use. Lastly, a fraction of failures could be attributed to the driver prioritization of memory allocation for graphics over computation. A graphics mode change could lead to memory allocated to the GPUGRID application being silently reallocated to the frame-buffer, leading to sudden corruption and application crash. Similarly, the GPUGRID application would be unable to start if a graphical application had already reserved all available GPU memory. Throughout the open testing phase, communication with and feedback from the user community was essential and was significantly facilitated by the web forums provided by the BOINC website software.

Results

In order to discuss the practical application of GPUGRID, we shall describe the SH2 domain/pYEEI free energy of binding experiment in more detail. Calculation of free energies is considered one of the most important applications of biomolecular simulations although it is renowned for being computationally expensive.[17] However, it can be achieved by using many independent MD simulations, making it an ideal candidate for distributed computing. The computational protocol employed to compute the free energy of binding for the SH2 domain/pYEEI complex is based on

the potential of mean force (PMF) method.[18,19] We employed the protocol of Doudou *et al*[19] as it is the one which is better suited for automation. In fact, it requires only the calculation of the PMF along a distance (z) separating the ligand and the protein and one free energy perturbation calculation (which could be omitted depending on the restrain used). The PMF is reconstructed along the one-dimensional reaction coordinate by a stratification of umbrella sampling (US)[20] simulations at different positions in z, then reconstructed with the WHAM protocol.[21,22]

Free energy calculations on the Src SH2 domain/pYEEI ligand system

The SH2 (Src homology 2) domains were first identified in cytoplasmic protein-tyrosine kinases of the *src* family, an oncogenic protein of the Rous sarcoma virus. This non-catalytic domain is composed of approximately 100 amino acids and binds to short peptidic sequences containing phosphorylated tyrosine residues regulating signal transduction pathways.[24] It is also well studied experimentally and computationally,[18,25–27] where several variants of the binding ligands have been studied.

System preparation. The input model is based on the bound crystallographic structure of the complex of the human p56^{lck} domain and the peptide phosphotyrosine-Glu-Glu-Ile (pYEEI for simplicity) (PDB:1LKK) using the CHARMM27[13] forcefield. The phosphotyrosine residue was assumed to be in its charged form Y-$PO_3{}^{2-}$ as experimentally determined.[25] Neutral acetylated N-terminus (ACE) and amidated C-terminus (CT2) residues were used to cap the peptide. The complex was solvated in a TIP3P[28] water box of a size of $65 \times 62 \times 93$ Å^3, being the z axis larger to allow for the generation of several US initial configurations with the ligand at different distances from the protein (Figure 4a). The system was solvated at the ionic strength of 0.15 M by adding 36 Na^+ and 33 Cl^- ions for 12280 water molecules. The final system comprised 38,655 atoms. The reaction coordinate was determined by the cross product of the vector defined by the center of masses of Cα atoms in residue Asn188 and Asn173 in the SH2 domain and the vector defined by the center of masses of the Cα atoms in residue pTyr252 and Ile255 in the pYEEI ligand. The model was rotated to align the reaction coordinate vector along the z-axis. In order to reduce the water molecules required to solvate the system, an additional rotation of 45 degrees with respect to the x-y plane was performed.

The system was minimized and equilibrated under NPT conditions at 1 atm and 298K using NAMD2.6[29] on a standard CPU cluster using a timestep of 2 fs, cutoff of 9 Å, rigid bonds and PME for long range electrostatic with a grid of $64 \times 64 \times 96$. During minimization and equilibration, the heavy protein atoms were restrained by a 10 kcal/mol/Å^2 spring constant. Two rounds of velocity re-initialization for 2 ps were performed under NVT conditions. The magnitude of the restraining spring constant was then reduced to 1 kcal/mol/Å^2 during 10 ps of NVT before the barostat was switched on at 1 atm for a further 10 ps of NPT simulation. A final 40 ps of NPT simulation was conducted with a restraint constant of 0.05 kcal/mol/Å^2. Finally, the volume was allowed to relax for 10 ns under NPT conditions. During this run, only Cα atoms of the complex were restrained with a 1 kcal/mol/Å^2 constant in order to prevent the system to reorient.

Umbrella sampling preparation. All the production simulations were run using ACEMD[5] over GPUGRID.net with the same parameters used for the equilibration but a timestep of 4 fs thanks to the use of the hydrogen mass repartition scheme implemented in ACEMD.[30] Note that individual atom masses do not appear explicitly in the equilibrium distribution, therefore changing them only affects the dynamic properties of the system (marginally) but not the equilibrium distribution. Umbrella sampling was performed starting from 381 initial configurations. Obtaining proper starting configurations for the US is crucial to reduce the time to system equilibration for the PMF calculation.

The reaction coordinate extended from $z = -3$ Å to $z = 35$ Å with the bound configuration at position $z = 0$ Å. US windows were spaced at 0.1 Å intervals from each other. For windows between $z = -3$ Å and $z = 0$ Å in the reaction coordinate (31 initial configurations) the bound structure was used as initial US configuration. For windows in

the positive range of the reaction coordinate $z = 0$ Å and $z = 35$ Å (350 initial configurations), an MD simulation displacing the ligand towards the bulk was performed. The displacement of the ligand was carried out for 35 Å applying a linear force $F = -k_d(z - vt)$ to all carbons of the ligand, where $k_d = 0.5$ kcal/mol/Å^2 and $v = 5$ Å/ns. During the pulling trajectory, snapshots of the system coordinates (Figure 4b) were saved at constant intervals corresponding to 0.1 Å. These system coordinates snapshots were used as the US initial configurations. In order to prevent ligand diffusion during its displacement and during the US run, an harmonic biasing restraint of $k = 0.1$ kcal/mol/Å^2 was applied to the center of mass of the ligand to restrain to the *xy* plane (with respect to the initial bound position of the ligand). Similarly, by means of preventing protein rotation and translation while preserving binding pocket flexibility during ligand separation as well as during US runs, a further harmonic restraint of $k = 0.5$ kcal/mol/Å^2 was applied to every Cα atom residing in an α-helix or β-sheet of the protein further than 9 Å from the ligand.

The generation of the initial configurations represented around 7 ns of simulated time which, using ACEMD on a single NVIDIA GeForce GTX275, took around 16 hours to complete. We ran 10 different trajectories in order to generate variability in the US initial configurations set. The final set of configurations was composed of an alternate sequential selection of configurations from the 10 trajectories. All 381 initial US windows, were submitted to GPUGRID.net for execution of the US protocol. All restraints preventing protein rotation and translation as well as ligand diffusion on the *xy* plane, were kept during the US runs. The US bias was fixed to $k = 10$ kcal/mol/Å^2 and it was applied to the center of mass of the ligand. Each US window simulation was 55 ns long divided into 21 successive steps. Each step was run as a separate GPUGRID job. Each job corresponded to about 7-8 hours of continued computation for a typical GPUGRID volunteer computer.

Free-energy calculation. The PMF over the reaction coordinate was reconstructed from time-windows of 5 ns over 373 completed US windows (Figure 5) using the WHAM method with a convergence tolerance of 10^{-4}. Although 381 US windows were submitted for computation, due to the volatility of the grid, some results did not complete in time for the analysis without affecting the total results. Hence, only the successfully returned results were used for PMF reconstruction.

The standard free energy of binding was computed using the expression given in:[19]

$$\Delta G^\circ = \Delta W_R - k_B T \ln\left(\frac{l_b A_{u,R}}{V^\circ}\right) + \Delta G_R, \qquad (1)$$

where ΔW_R is the PMF depth, k_B is the Boltzmann constant, T is the temperature, $l_b = \int_{\text{bound}} \exp\left(-W_R(z)/k_B T\right) dz$ is the integral of the PMF over the bound length, $A_{u,R} = 2\pi k_B T / k_{xy}$ is the area in the x and y directions of the unbound ligand, $V^\circ = 1{,}661$ Å^3 is the standard volume, and ΔG_R is the free energy to remove the orthogonal restraints (on x and y) when the ligand is bound. ΔG_R is obtained via a free energy perturbation approach from the exponential average.[19] The lowest PMF reported in Figure 5 is $\Delta W_R = -10.8$ kcal/mol, the bound distance is $l_b = 0.93$Å and the area explored by the ligand in the *xy* plane $A_{u,R} = 37.07$Å^2. The free energy to remove the constraints have a negligible contribution $\Delta G_R = -0.0124$ kcal/mol due to the low restrain applied. The standard free energy of binding for this PMF is computed from Eq. (1) as $\Delta G^\circ = -8.5$ kcal/mol. The final standard free energy of binding is estimated from the average of the last three computed PMF (15 ns) as they differ by only 0.4 kcal/mol. We obtain an average result of $\Delta G^\circ = -8.7 \pm 0.4$ kcal/mol which compares with the reported experimental value of -8.0 ± 0.1 kcal/mol.[23] The error is taken as the maximum difference between the last three computed PMF depths.

Conclusion

Although the practice of volunteer computing is well established, until recently it has used primarily the CPUs in the volunteered computers. Consequently, it has been useful only for applications that are amenable to being decomposed into jobs small enough to be accommodated by the typical desktop or workstation machines. With the advent of general-purpose graphics cards (GPUs), the computational power accessible within an in-

dividual PC is now up to 100 times that of the CPU alone, with a commensurate speed-up possible for applications designed to exploit them. In the case of molecular dynamics applications, we have shown that the speed-up realized on a GPU is sufficient to allow a simulation previously requiring dozens of processors on a dedicated HPC system to be performed on a single GPU. As a result, it has become feasible to perform non-trivial molecular dynamics simulations in a distributed manner. To this end, we devised GPUGRID, a volunteer computing initiative using BOINC.

For computational scientists, GPUs can represent a way to perform computational protocols which were previously impractical due to the computational cost. For the specific case of Src SH2 domains, we were able to obtain an accurate estimate of the free energy (-8.7 ± 0.4 kcal/mol against an experimental value of -8.0 ± 0.1 kcal/mol), but for a quite high computational cost consisting of 20.5 μs of umbrella sampling data. We also tried to use the most adequate computational protocols in order to increase parallel granularity and semi-automated scripts such that the infrastructure could be made scalable to multiple protein complexes with reasonable human intervention.

At the current level of performance, we expect that this protocol could be refined to be able to produce several protein-ligand binding affinity calculations per day, a challenge that we are currently working on. Also, a similar level of performance could be replicated in-house by a dedicated GPU cluster which need not be as large as GPUGRID due to its dedicated nature. It is conceivable that lead optimization studies will be best suited for this infrastructure, due to the accuracy required by the binding affinities and to easy parallelization of the protocols over multiple ligand structures. This protocol can possibly be optimized in order to obtain better sampling of the initial configurations of the US in order to reduce the sampling time. Over the time, we have performed on GPUGRID an extensive amount of tests for long US simulations totaling more than 900 μs of US data (see Table 1) leading to the presented best protocol. Sampling more different initial configurations seems to be the factor that most affects the convergence of the estimate. We are currently looking at improving the protocol in order to obtain similar accurate binding free energies at lower cost. Future calculations will allow us to determine the binding affinities for other ligands like PQpYEEIPI, PQpYEpYIPI, PQpYIpYIPI, PQpYIpYVPI to evaluate differential standard free energies ($\Delta\Delta G^{\circ}$). Indeed, once the sampling problem is solved, there is the limit of the force fields, a task that would probably be helped by this kind of high-throughput molecular simulations.

Acknowledgments We gratefully acknowledge support from NVIDIA corporation. IB acknowledges support from the Obra Social Fundació "La Caixa". MJH acknowledges support from the HPC-Europa2 project (project number 228398). TG acknowledges partial support from the Virtual Physiological Human Network of Excellence (VPH-NoE). GDF acknowledges support from the Ramón y Cajal scheme. BOINC is supported by the National Science Foundation (OCI-0721124). We finally thank all the volunteers of GPUGRID who donate GPU computing time to the project.

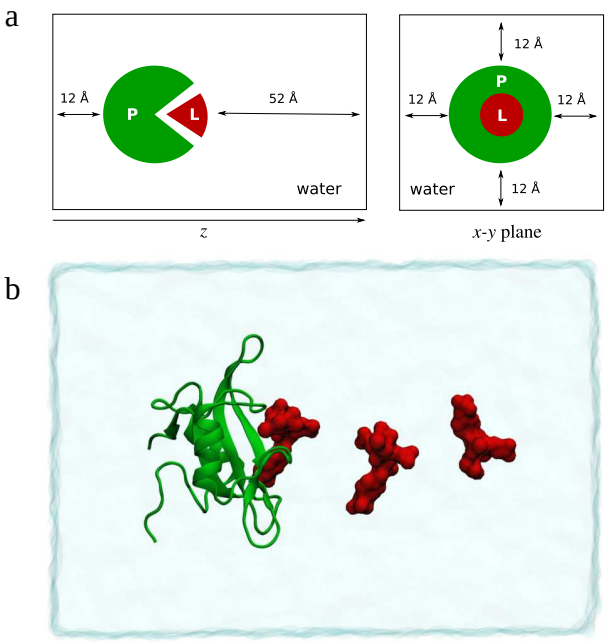

Figure 4: (a) Schematic representation of a system for calculation of free-energy of binding. "P" and "L" are for protein and ligand, respectively. The complex is solvated in a water box with a boundary at least 12 Å from the system in the x and y directions and of 52 Å in the z-direction, to allow for the generation of the US initial configurations. Such orientation is kept fixed during the equilibration protocol by applying restraints on a selection of atoms of the complex. (b) Schematic visualization of the initial configurations for the US of the SH2 domain/pYEEI ligand complex (PDB:1LKK) in the water box. The initial configurations are created by displacing the ligand from the binding pocket by applying a linear force to every carbon atom of the ligand at a 5 Å/ns constant velocity for 35 Å. The protein is kept fixed by an harmonic restrain of 1 kcal/mol/Å^2 constant applied on every Cα atom with secondary structure further than 9 Å from the ligand in its bound state.

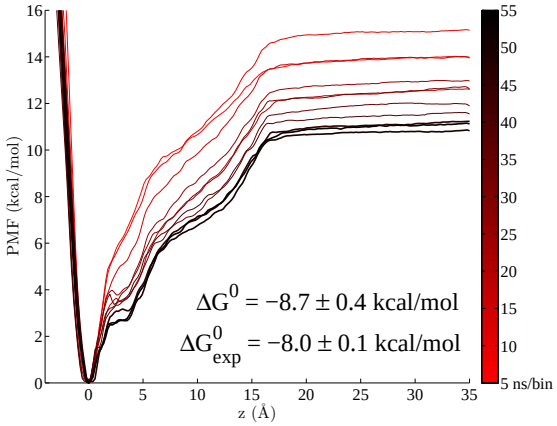

Figure 5: Reconstructed potential of mean force of the SH2 domain/pYEEI ligand complex along the reaction coordinate, calculated from 373 completed US configurations of 55 ns each. The PMF is reconstructed over time windows of 5 ns along the US trajectories (running average) in order to show the long relaxation time of the US simulations. The standard free energy of binding is estimated from the average of the last three computed PMF depths (15 ns) as they differ by only 0.4 kcal/mol. The simulation estimate is $\Delta G^0 = -8.7 \pm 0.4$ kcal/mol, accounting for standard volume and biasing factors of Eq. 1. The error is taken as the maximum difference between the last three computed PMF depths. The experimental value for this system is -8.0 ± 0.1 kcal/mol[23].

References

(1) Ekins, S.; Mestres, J.; Testa, B. In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling. *Brit. J. Pharmacol.* **2007**, *152*, 9–20.

(2) Smit, B.; Frenkel, D. *Understanding Molecular Simulation*; Academic, New York, 2002.

(3) Moore, G. E. Cramming more components onto integrated circuits. *Electronics* **1965**, *8*, 38.

(4) Giupponi, G.; Harvey, M. J.; De Fabritiis, G. The impact of accelerator processors for high-throughput molecular modeling and simulation. *Drug Discov. Today* **2008**.

(5) Harvey, M. J.; De Fabritiis, G.; Giupponi, G. ACEMD: Accelerated Molecular Dynamics for the microsecond time-scale. *J. Chem. Theory Comp.* **2009**, *5*, 1632–1639.

(6) Anderson, D. P.; Christensen, C.; Allen, B. Designing a Runtime System for Volunteer Computing. *Proc. ACM/IEEE SC 2006 Conference Supercomputing SC '06*, 2006; pp 33–33.

(7) BOINCStats website, URL `http://www.boincstats.com`, online. Accessed: 11 Jan 2010.

(8) SETI@Home website, URL `http://setiathome.berkeley.edu`, online. Accessed: 11 Jan 2010.

(9) World Community Grid website, URL `http://www.worldcommunitygrid.org`, online. Accessed: 11 Jan 2010.

(10) Luttmann, E.; Ensign, D. L.; Vaidyanathan, V.; Houston, M.; Rimon, N.; Øland, J.; Jayachandran, G.; Friedrichs, M.; Pande, V. S. Accelerating molecular dynamic simulation on the cell processor and Playstation 3. *J. Comput. Chem.* **2009**, *30*, 268–274.

(11) Harvey, M.; Giupponi, G.; Villà-Freixa, J.; De Fabritiis, G. PS3GRID.NET: Building a distributed supercomputer using the PlayStation 3. In *Distributed and Grid Computing - Science Made Transparent for Everyone. Principles, Applications and Supporting Communities*; Weber, M. H. W., Ed.; Rechenkraft.net e.V., Marburg, 2007.

(12) De Fabritiis, G.; Coveney, P. V.; Villà-Freixa, J. Energetics of K+ permeability through Gramicidin A by forward-reverse steered molecular dynamics. *Proteins* **2008**, *73*, 185–194.

(13) MacKerell, A. D.; Banavali, N.; Foloppe, N. Development and current status of the CHARMM force field for nucleic acids. *Biopolymers* **2000**, *56*, 257–265.

(14) Duan, Y.; Wu, C.; S., C.; Lee, M. C.; Xiong, G.; Zhang, W.; Yang, R.; Cieplak, P.; Luo, R.; Lee, T.; Caldwell, J.; Wang, J.; Kollman, P. A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *J. Comp. Chem.* **2003**, *24*, 1999–2012.

(15) Darden, T.; York, D.; Pedersen, L. Particle mesh Ewald: An N log(N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089–10092.

(16) Verlet, L. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.* **1967**, *159*, 98.

(17) Deng, Y.; Roux, B. Computations of Standard Binding Free Energies with Molecular Dynamics Simulations. *J. Phys. Chem. B* **2009**, *113*, 2234–2246.

(18) Woo, H. J.; Roux, B. Calculation of absolute protein-ligand binding free energy from computer simulations. *Proc. Natl. Acad. Sci. USA* **2005**, *102*, 6825–6830.

(19) Doudou, S.; Burton, N. A.; Henchman, R. H. Standard Free Energy of Binding from a One-Dimensional Potential of Mean Force. *J. Chem. Theory Comput.* **2009**, *5*, 909–918.

(20) Torrie, G. M.; Valleau, J. P. Nonphysical sampling distributions in Monte Carlo free-energy estimation - Umbrella sampling. *J. Comput. Phys.* **1977**, *23*, 187–199.

(21) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. Multidimensional free-energy calculations using the weighted histogram analysis method. *J. Comput. Chem.* **1995**, *16*, 1339–1350.

(22) Roux, B. The calculation of the potential of mean force using computer simulations. *Comput. Phys. Commun.* **1995**, *91*, 275–282.

(23) Lee, T. R.; Lawrence, D. S. SH2-Directed Ligands of the Lck Tyrosine Kinase. *J. Med. Chem.* **2000**, *43*, 1173–1179.

(24) Sadowski, I.; Stone, J. C.; Pawson, T. A noncatalytic domain conserved among cytoplasmic protein-tyrosine kinases modifies the kinase function and transforming activity of Fujinami sarcoma virus P130gag-fps. *Mol. Cell. Biol.* **1986**, *6*, 4396–4408.

(25) Bradshaw, J. M.; Waksman, G. Calorimetric investigation of the proton linkage by monitoring both the enthalpy and association constant of binding: application of the interaction of the Src SH2 domain with a high-affinity tyrosyl phosphopeptide. *Biochemistry* **1998**, *37*, 15400–15407.

(26) Fowler, P.; Geroult, S.; Jha, S.; Waksman, G.; Coveney, P. Rapid, accurate, and precise calculation of relative binding affinities for the SH2 domain using a computational grid. *J. Chem. Theory Comput.* **2007**, *3*, 1193–1202.

(27) De Fabritiis, G.; Geroult, S.; Coveney, P. V.; Waksman, G. Insights from the energetics of water binding at the domain-ligand interface of the Src SH2 domain. *Proteins* **2008**, *72*, 1290–1297.

(28) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **1983**, *79*, 926–935.

(29) Phillips, J.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R.; Kale, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781–1802.

(30) Hess, K.; Berendsen, H. Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich systems. *J. Comput. Chem.* **1999**, *20*, 786–798.