ELSEVIER

The impact of accelerator processors for high-throughput molecular modeling and simulation

G. Giupponi[1], M.J. Harvey[2] and G. De Fabritiis[1]

[1] Computational Biochemistry and Biophysics Lab, GRID IMIM Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C/ Doctor Aiguader 88, 08003 Barcelona, Spain
[2] Information and Communications Technologies, Imperial College London, South Kensington, London SW7 2AZ, UK

The recent introduction of cost-effective accelerator processors (APs), such as the IBM Cell processor and Nvidia's graphics processing units (GPUs), represents an important technological innovation which promises to unleash the full potential of atomistic molecular modeling and simulation for the biotechnology industry. Present APs can deliver over an order of magnitude more floating-point operations per second (flops) than standard processors, broadly equivalent to a decade of Moore's law growth, and significantly reduce the cost of current atom-based molecular simulations. In conjunction with distributed and grid-computing solutions, accelerated molecular simulations may finally be used to extend current *in silico* protocols by the use of accurate thermodynamic calculations instead of approximate methods and simulate hundreds of protein–ligand complexes with full molecular specificity, a crucial requirement of *in silico* drug discovery workflows.

Bringing a new drug to market is a long and expensive process [1] encompassing theoretical modeling, chemical synthesis and experimental and clinical trials. Despite the considerable growth of biotechnologies in the past ten years, the practical consequences of these techniques on the number of approved drugs have failed to meet expectation [2]. Nonetheless, the discovery process greatly benefits from the use of computational modeling [3], at least in the initial stages of compound discovery, screening and optimization [4].

Among the techniques available, force-based molecular modeling methods (briefly, molecular modeling) such as molecular dynamics are particularly useful in studying molecular processes at the atomistic level, providing accurate information on macromolecular dynamics and thermodynamic properties. Bridging from the molecular-atomistic (femtosecond) to biological (micro-millisecond) timescales is still an unaccomplished feat in computational biology: the complexity of the modeling impedes sufficient sampling of the evolution of the system, even on expensive high performance computing (HPC) resources. For instance, cost and sampling constraints have so far limited a routine molecular dynamics run over a PC cluster to a single protein system for tens of nanoseconds, barely sufficient to compute the binding free energy using an exact thermodynamic method [5]. This information is of great importance in the discovery process for virtual screening, lead optimization and *in silico* drug discovery [4], but needs to be computed for hundreds of protein–ligand systems efficiently and economically to open the way for molecular simulations to become a routine tool in the discovery workflows used in the biotechnology industry.

In light of the significant changes that have occurred lately in microprocessor design, we review the first cost-effective accelerator processors (APs) – the Cell processor and Nvidia graphics processing units (GPUs) – and examine results of their early adoption in stand-alone and grid-computing scenarios in the context of high-throughput molecular simulation for biomolecular applications.

Accelerator processors

Historically, microprocessor performance has improved primarily through the raising of clock speeds, make possible by the development of ever-finer fabrication processes. In recent years, it has

Corresponding author: Giupponi, G. (giovanni.giupponi@upf.edu), Harvey, M.J. (m.j.harvey@imperial.ac.uk), De Fabritiis, G. (gianni.defabritiis@upf.edu)

1359-6446/06/$ - see front matter doi:10.1016/j.drudis.2008.08.001

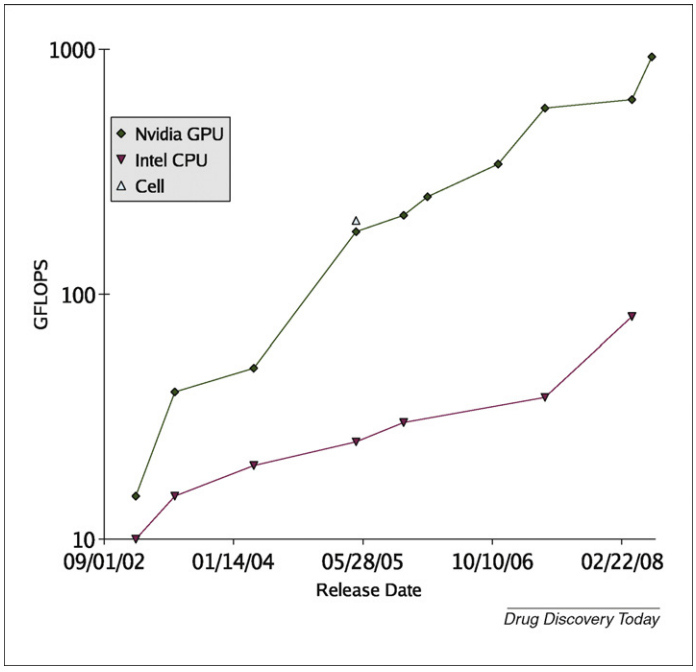

FIGURE 1

The theoretical peak performance trend of Nvidia and Cell processors in comparison to contemporary Intel processors. Gflops measures for Nvidia and Intel devices are taken from [11]. Since 2002, Nvidia GPU performance has increased by ≈100%/annum while that of Intel CPUs by ≈50%/annum. Nvidia GPU devices gained the programmable flexibility necessary for general purpose computation with the release of the G80 core in November 2006.

become increasingly difficult to keep increasing clock speeds because of fundamental limits in the process technology and power consumption. Performance has also been limited by the increasing relative cost of accessing main memory, the speed of which has increased at a slower rate than CPUs. Despite this, Moore's Law [6] (Fig. 1), the empirical observation that the density of transistors on an integrated circuit doubles every 18–24 months has continued to hold true. Manufacturers have been forced to reconsider their 'single fast core' design and have used the greater transistor counts to build CPUs containing multiple independent processing cores. Even so, a large fraction of the transistors on a modern CPU remain dedicated to providing a very fast cache memory that is used to hide the cost of communicating to the much larger but slower main system memory.

As a result, although the aggregate performance of multicore CPUs has continued to increase, it is no longer possible for serial (single-threaded) programs to take advantage of the increased processing capability, as the computing cores are independent. Instead, it is necessary for codes to be parallelized: adapted to perform computation concurrently on multiple cores. In the case of molecular dynamics (MD) modeling, parallel codes such as NAMD [7] may be used to distribute simulations across multiple processors. Low latency, high bandwidth interconnections between processors are needed for good scalability and performance is ultimately limited by the size of the simulated system which must increase with processor count to maintain parallel efficiency.

An alternative approach is to employ special purpose hardware specifically tailored for molecular dynamics simulation, such as MD-GRAPE [8] or Anton [9]. These can yield an improvement in performance of several orders of magnitude over commodity processors, however special hardware is expensive to buy, develop and – without continued development – Moore's Law ensures that gains in performance relative to general purpose computing hardware are rapidly eroded.

Recently, hardware manufacturers have introduced a third class of devices, which we term accelerator processors. These vary in architectural details but share the common trend of having many, comparatively simple processing cores in a single package. These cores are generally optimized to have much higher floating-point arithmetic performance than conventional CPUs (Fig. 1), at the expense of being able to execute complex branching programs efficiently, for example reduced control hardware (Fig. 2). At present, each vendor is independently developing its solutions, with Sony–Toshiba–IBM (STI) Cell processor (Fig. 2b) and Nvidia GPUs (Fig. 2c) already adopted by a large user-base. Cell and GPU devices are able to achieve an order of magnitude more floating-point operations per second (flops) than conventional processors at a similar price and, for some numerically intensive applications, promise a speed-up of almost 100 times. Furthermore, because these devices are designed for the mass market, they are also substantially cheaper than previous special purpose hardware.

The main specifications of currently available devices are as follows:

- *The Cell processor architecture*: Sony–Toshiba–IBM released in 2006 the first general-purpose processor (Cell) [10] that implements a multicore architecture (nine inhomogeneous cores) with features specifically designed to mitigate the effects of memory access latency. At more than 230 Gflops (230 billion flops) in single precision floating-point, the Cell provides ten times more flops than a CPU at similar cost (Box 1).
- *The GPU architecture*: the introduction of general-purpose programmable capabilities to GPUs has marked a drastic change in hardware and HPC architectures, as graphic cards can now be used for the computing intensive parts of a calculation. The Nvidia G80 architecture, introduced in 2007, is

BOX 1

The Cell processor

The present version of the Cell processor comprises one general purpose PowerPC processing element (PPE) which runs the operating system and acts as a standard processor and eight independent, specialized, synergistic processing elements (SPEs) (Fig. 2) that are simple vector processors. The essential characteristics are as follows:

- Main memory can be accessed only by the PPE core: each SPE must use its limited in-chip local memory (local store) of 256 kB. This memory is accessed directly without any intermediate caching.
- Optimized for single precision in current release.
- Each core (PPE or SPE) features a single instruction multiple data (SIMD) vector unit.
- SPEs are vector processors designed for very fast floating-point operation, at the expense of ability to run complex programs.
- Peak performance is 230 Gflops.

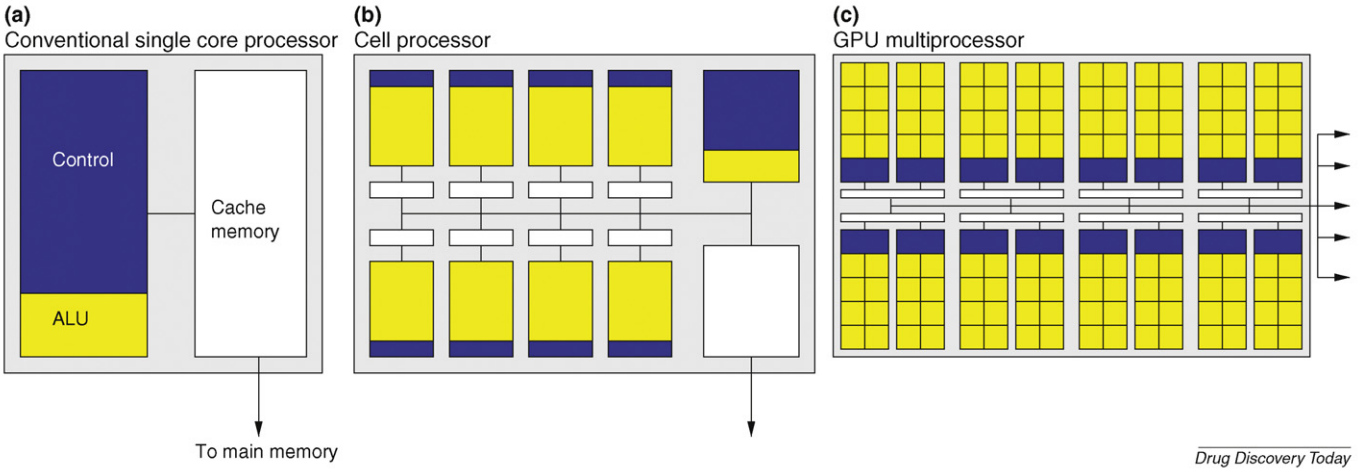

FIGURE 2

Illustration of the relative differences between CPU, Cell and GPU designs. **(a)** A conventional CPU dedicates a relatively large fraction of its transistors to complex control logic, to maximize performance on a mixed workload of serial code. A large cache memory is required to disguise the cost of accessing slow main memory. **(b)** The Cell processor contains eight synergetic processing elements (SPEs) designed to maximize arithmetic throughput at the expense of the ability to run complex programs. A conventional CPU core is used to control the SPEs and supply them with data. **(c)** Nvidia GPU devices have a very large number of cores, all of which may talk to main memory directly. There is no cache for main memory accesses, the cost of which is hidden by overlapping many computational threads. The overall program execution is controlled by code running on the CPU of the host system.

designed to be well suited to data-parallel computation, in which the same program is executed on many data elements in parallel [11]. Current products based on this architecture are able to achieve up to 648 Gflops (GeForce 9800GTX) (Box 2).

By comparison, a dual core Intel Xeon 5160 is capable of achieving approximately 38 Gflops with Intel's fastest quad-core processor, the Xeon 5472 achieves 81 Gflops, but at a considerable cost premium [12]. Other major manufacturers such as AMD-ATI and Intel have announced their intention to enter the AP market, but a systematic evaluation of architecture and performance for their products is at the moment not possible.

BOX 2

The CUDA compatible GPU device

The recently introduced G80 [43] series GPU from Nvidia [44] represents the first GPU that compatible with the Nvidia Compute Unified Device Architecture (CUDA) [11] platform for programming this and subsequent GPUs. Key points are as follows:

- Highly parallel with up to 240 cores.
- The G80 architecture includes texture units, which are capable of performing linear or bi-linear interpolation of arrays of floating point data.
- The current G80 device is capable only of giving single precision, IEEE-754 floating point arithmetic, double precision arithmetic is expected to reach the market in 2008.
- The processor also has special hardware support for reciprocal square root, exponentiation and trigonometric functions, allowing these functions to be computed with very low latency at the expense of reduced precision.
- Peak performance is 933 Gflops.

Costs, benefits and risks

Software re-engineering

Although APs offer high peak computational performance, exploiting this efficiently comes at the cost of ease of use: codes must be refactored as highly parallelized programs. Code redesign and redevelopment has a very high cost that, when weighted against the traditional 'free' performance increase provided by the next iteration of conventional hardware, has in the past significantly limited the appeal of special-purpose hardware on the high-performance computing market. In the next few years, this re-engineering cost will become less important as all serial codes will have to be redesigned to take advantage of standard multicore processors. Therefore, the possibility of realizing a 100 times improvement in application performance on an AP, broadly equivalent to a decade of Moore's Law growth, makes code redevelopment a fully justified proposition to speed up current workflows or perform totally new applications that could not be achieved on single CPUs within the next decade.

With wide and pervasive availability of APs across the market, from high-end HPC [13] to commodity Nvidia graphics cards, we contend that applications programmers must eventually adopt this technology, with the risk of being seriously outperformed otherwise, however embracing APs requires the acquisition of new know-how and a commitment to develop further application enhancements on such architectures. Porting an application to either the Cell or Nvidia GPUs requires refactoring code to conform the platform's programming model. The Cell processor can be programmed as a multicore chip using standard ANSI C and relying on libraries from the IBM system development kit (SDK) [14] to handle communication, synchronization and Single Instruction Multiple Data (SIMD) computation. The SIMD model is similar to that found on some commodity CPUs, such as Altivec

(PowerPC) and SSE (Intel). The Nvidia's Compute Unified Device Architecture (CUDA) SDK [11] reduces the difficulty of programming GPU devices by providing a minimal set of extensions to the standard C programming language. Code functions written to be executed on the GPU are known as *kernels* and are executed in *blocks*. Each block consists of multiple instances of the kernel, called *threads*, that are run concurrently on a single multiprocessor. The CUDA run-time is responsible for efficiently scheduling the execution of blocks on available GPU hardware. As the CUDA programming model abstracts the implementation details of the GPU, the programmer may easily write code that is portable between current and future Nvidia GPUs. Furthermore, it is expected that future CUDA SDKs will provide support for targeting Intel and AMD multicore CPUs, although obviously without the performance boost provided by GPU hardware.

Hardware costs and energy savings

APs are already available as consumer products: the Sony PlayStation 3 (PS3) features a Cell processor while Nvidia's entire range of GeForce graphics cards support CUDA at varying levels of performance. Professional-grade variants of these products are also available in the form of IBM Cell QS20 blade servers and Nvidia's Tesla GPU range. Mass production assures low unit price, better vendor support and a broad market. Millions of PS3s have been sold and can be trivially converted to running workstation operating systems such as Linux [15]. GPUs can be bought and added to most PCs as a simple end-user upgrade. APs cost a few hundred US dollars per piece and approach the computational power of small (tens of machines) PC clusters. Additionally, adopting APs may markedly reduce setup, support and running costs in comparison to a conventional cluster. It has been noted that the PS3 is one of the most efficient hardware architectures per dollar for molecular dynamics [16] and fluid dynamics.

APs are also very energy-efficient computing resources. For example, it is claimed that the recently announced ATI Firestream GPU board will achieve 5 Gflops/W [17], a figure to be matched by Nvidia's next generation of Tesla products [18]. As for the Cell processor, a PS3 has a peak power rating of 280 W [19], delivering at least 0.8 Gflops/W, and more professional solutions such as IBM QS21 blades featuring two Cell processors deliver 2.09 Gflops/W [20]. Furthermore, IBM has recently fabricated the Cell processor on a 45 nm process, estimating a 40% less power consumption. As of July 2008, the most energy-efficient supercomputer delivers 0.35 Gflops/W [21], therefore standard computer clusters are at least one order of magnitude less efficient than APs. These figures sum up the significant impact that APs will have not only on performance, but also on green computing.

At the lowest level of their accelerated-computing infrastructure, a company could use a locally deployed BOINC installation – similar to that used by gpugrid.net – to run simulations at zero cost using GPU hardware already available in office desktop computers. A scenario in which the simulations are mission crucial might involve the use of Tesla units (approximately US$ 5000 per blade with four teraflops per blade) or properly engineered GeForce clusters (approximately US$ 2000 for four teraflops). Obtaining the same computing power on a cluster of CPUs would cost over an order of magnitude more in hardware and substantially more to run in terms of power and maintenance.

Accelerated modeling

Despite the very recent introduction of APs into the market, a variety of applications targeting different industrial and scientific fields have already appeared. Excellent performance speed-ups have been reported for such diverse cases as computational finance [22], fluid dynamics [23], sequence alignment [24] and quantum chemistry [25]. These notable achievements for such a variety of algorithms highlight the potential of APs for computational science in general.

Accelerators processors will prove to be beneficial where the application has potential for a large degree of parallelism. In this sense, they should be appropriate for most of the computational biology tools used in the industry (network building, cell and organ simulations) and for multidimensional modeling of properties like ADME/Tox alongside bioactivity [26]. These tools are yet to be ported on APs, a task which may require significant re-engineering. We would expect that docking programs will be one of the first applications which will be available on the new hardware, with molecular dynamics simulations currently being the most active sector of development. For quantum chemistry codes, which are computationally very expensive, APs could play a major role. A current limitation for this kind of application is the lack of double precision support of current GPUs. With new cards coming out late in the year featuring double precision, it is probable that quantum mechanics code developers will be looking at this hardware technology. Software vendors will require some time to perform this migration, but the different exponential growth of GPU versus CPU shown in Fig. 1 will ultimately force this process.

We review here results that are directly relevant to extending current molecular modeling capabilities in more detail. Several groups have started to implement MD routines to APs. van Meel *et al.* [27] describe a CUDA implementation of Lennard–Jones MD which achieves a net speed-up of up to 40 times over a conventional CPU. Although suitable for coarse-grained simulations, the lack of support for an atomistic, biomolecular force field limits the applicability of the code. Stone *et al.* [28] demonstrate the GPU-accelerated computation of the electrostatic and van der Waals forces obtaining 10–100 times speed-up compared to heavily optimized CPU-based implementations, but because the remainder of the MD simulation remains performed by the host CPU, the net speed-up is reduced to around 5 times. Preliminary results for a fully atomistic MD program called aceMD [29] show a 50 times speed-up measured at the peak performance of a parallel multi-GPU MD code. Stone *et al.* also describe performance improvement for other algorithms in the visualization software VMD [30] such as Coulomb-based ion placement and time-averaged potentials calculations obtaining respectively up to 470 and 110 performance speed-up. On the Cell processor, a sustained performance of 30 Gflops was achieved for a fully atomistic molecular dynamics program, more than an order of magnitude faster compared to the same application running on a single processor core [16]. We note here that all performance speed-ups in these studies have been obtained using single precision floating-point, which is adequate for MD simulations. Nevertheless, it is expected that double precision arithmetic will be supported in the next iterations of the Cell processor and GPUs.

Distributed computing in the accelerated era

Computational grids enable scientists to distribute simulations across a pool of machines to benefit from their aggregate power, and have already proved useful for fast calculations of binding affinities using MD techniques [31]. Owing to the computational cost of molecular simulations, the grid is usually composed of very expensive parallel-processing HPC resources, the costs of which limit the applicability of the approach. When distributed computing is combined with AP-equipped hardware, however, a single computational node is sufficiently powerful (equivalent to tens of CPUs) to simulate molecular dynamics trajectories of reasonable length in a day for a molecular system of the order of 50,000 atoms. This speed-up, compared to a couple of weeks for the same run on a single PC, enables the effective use of loosely-coupled computational grids of AP-equipped machines for molecular dynamics simulations.

Inevitably, making efficient use of a collection of nondedicated machines presents new challenges. As the machines might not be dedicated to computing during office hours, the pool of machines available must be treated as transient: the owner of the GPU-equipped PC may choose to power it down at any moment, for example. The distributed infrastructure must accommodate this and be able to correct for the loss of results arising from an incomplete simulation. Software like the Berkeley Open Infrastructure for Network Computing (BOINC) framework [32] or Condor [33] distributed computing solutions have reached, over many years, the maturity and stability required by these types of applications (Seti@HOME [34]). For instance in the ps3grid.net and gpugrid.net projects [35] a BOINC server produces hundreds of calculations a day with very little human intervention and very high reliability on a diverse range of hardware including PlayStation3 and PC-based Nvidia GPUs scattered across the globe. Similarly, a dedicated in-house distributed grid of Nvidia graphics cards, commonly already in use in most modern PC, could reach throughput only possible with use of the largest HPC resources, but at fraction of the cost.

A simple benchmark of the impact of distributed computing using APs is given by the first numerical experiment of ps3grid.net which used steered molecular dynamics to compute the free energy of translocation of a potassium ion across a transmembrane pore [36]. ps3grid.net uses an MD code that is optimized for the Cell processor [16] and runs a single MD simulation per client. The trajectories produced by this ensemble of simulations are subjected to statistical analysis [36] to recover the free-energy profile of the ion translocation. The setup of the computational protocol was first performed using standard supercomputing hardware with a few iterations of over 50 runs, each lasting half day on 32 processors and amounting to 19,000 CPU hours and around 40 ns of simulation time [36]. An extended set of numerical experiments run on ps3grid.net produced 5000 trajectories, 4 μs of simulated time, over 200 years of CPU time with a daily output of 100 ns and 5 GB of data. This numerical experiment produced several pullings which is at least one order of magnitude closer to single-molecule pulling experiments performed using optical tweezers [37].

Future outlook for medium-throughput molecular modeling

There is great interest in methods for supporting and optimizing experimental high-throughput screening [4,38] to identify, characterize and optimize possible leads for a given target out of the vast number of viable chemical compounds. It is, however, very difficult for such methods to account correctly for the many phenomena involved in complex formation, such as the subtle interplay between entropy and enthalpy, conformational changes of the ligand or the substrate, presence of water molecules in the binding sites and limited resolution of structures [39,40]. We advocate that high-throughput molecular-based methods aimed at a detailed, systematic and physically sound quantitative description of the binding site can now be used thanks to APs.

Calculating the chemical potential of water [41] in the binding site already provides a cheap but useful way to understand the mechanism of ligand–protein binding. Thermodynamics integration (TI) [31], steered molecular dynamics [36], umbrella sampling [5] and linear interaction energy (LIE) approaches [42] can be applied to several hundred protein structures at once. Furthermore, standard molecular dynamics could be of great help to the understanding of a plethora of molecular and cellular processes. High-throughput docking using new techniques or improved scoring functions would become treatable on distributed accelerated solutions even if such methods require substantially more computing power.

Methods using atomistic force-fields and dynamical molecular simulations can provide enough level of detail to achieve accurate virtual screening performance [39]. In fact, such approaches potentially take into account system-specific interactions involved in complex formation. We note that as larger timescales and extended sampling become routinely available the limitation of force-fields will start to appear more evidently. Cases where this limitation is already demonstrated are known as in [36] for a gramicidin A pore, where an error of several kcal/mol of the energetic barrier of translocation is consistently found, possibly owing to lack of polarization in the water molecules within the pore. Nevertheless, current force-fields have shown also encouraging degrees of accuracy. Increasing sampling capabilities will move the bottleneck from computers back into the modeling of molecular structures and classical force-fields with the possibility of developing revised versions which will accommodates the new timescales.

As an example, it is common for a biotechnology company to screen potential targets using docking ranking methods. The availability of accelerated software for molecular dynamics simulations would allow them to perform more accurate thermodynamic free-energy protocols as umbrella sampling which are expected to produce much higher enrichment than ranking methods. Of course, as there are several possible compounds to test the problem quickly becomes combinatorially too expensive to treat experimentally. Supplementing the experimental work with free-energy calculations performed across an in-house distributed GPU computing solution using quantitative thermodynamic calculations could significantly reduce the cost and time of the research provided that the thermodynamic screening can be performed on at least hundreds of protein–ligand complexes, rather than just few as it is for current standard CPU hardware solutions. The key factor introduced by accelerators is to shift the modeling from single binding affinity predictions to high-throughput affinity predictions.

Conclusion

APs have the potential to provide a radical change in scientific and industrial computation. With an effective performance tens of times that of standard computers and doubling each 8–12 months, unprecedented levels of computational power can be put into the hands of scientists and programmers at an affordable cost. Such disruptive technology is already being used by research groups and companies in many different fields, and applications targeting molecular modeling are appearing at a steady pace. The gain of raw performance combined with the possibility of distributed, grid-like deployment, can finally unleash the full potential of advanced molecular modeling methods, routinely allowing virtual experiments previously only achievable on supercomputers. In effect, these loosely-coupled distributed computing solutions can be seen as grids of small computer clusters provided that the software is optimized to run on accelerated processors.

For the case of molecular modeling and simulations, such novel accelerator hardware and technologies are bound to play a significant role in next generation *in silico* drug discovery. APs open up the possibility to increase the molecular detail allowing more refined and computationally expensive methods based on atomistic force fields that cannot presently be considered. APs will therefore allow a fundamentally different approach to modeling and reshape *in silico* drug discovery protocols, by increasing the detail (up to atomistic force-field based level) and using more accurate thermodynamic methods to compute free energies, and by increasing the throughput (more trajectories, longer, on more proteins and ligands) to an extent outside the capability of traditional processors for the next decade.

Conflicts of interest statement

We notify the journal that the authors are scientific consultants and also share holders of Acellera Ltd, a UK-based company selling software solutions for accelerated processors.

Acknowledgements

We gratefully acknowledge support from Barcelona supercomputing center (http://www.bsc.es), Acellera Ltd (http://www.acellera.com), Sony Computer Entertainment Spain (http://www.scee.com) and Nvidia corporation (http://www.nvidia.com). We thank Jordi Mestres and Ferran Sanz for a critical reading of the manuscript. GG acknowledges the Aneurist project (http://www.aneurist.org) for financial support. GDF acknowledges support from the Ramon y Cajal scheme.

References

1 Dickson, M. and Gagnon, J. (2004) Key factors in the rising cost of new drug discovery and development. *Nat. Rev. Drug Discov.* 3, 417–429
2 Nightingale, P. and Martin, P. (2004) The myth of the biotech revolution. *Trends Biotechnol.* 22, 564–569
3 Jorgensen, W.L. (2004) The many roles of computation in drug discovery. *Science* 303, 1813–1818
4 Ekins, S. *et al.* (2007) In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling. *Br. J. Pharmacol.* 152, 9–20
5 Smit, B. and Frenkel, D. (2002) *Understanding Molecular Simulation*. Academic, New York
6 Moore, G. (1995) Lithography and the future of Moore's law. *Proc. SPIE* 2438, 2
7 Phillips, J. *et al.* (2005) Scalable molecular dynamics with NAMD. *J. Comput. Chem.* 26, 1781–1802
8 Susukita, R. *et al.* (2003) Hardware accelerator for molecular dynamics: MDGRAPE-2. *Comput. Phys. Commun.* 155, 115–131
9 Shaw, D. *et al.* (2007) Anton, a special-purpose machine for molecular dynamics simulation. In *Proceedings of the 34th Annual International Conference on Computer Architecture* pp. 1–12
10 IBM website on the Cell processor (http://www.research.ibm.com/cell/)
11 Compute Unified Device Architecture, Nvidia (http://developer.nvidia.com/object/cuda.html)
12 Intel Xeon Processor 5000 Sequence (http://www.intel.com/performance/server/xeon/hpcapp.html, accessed 29 April 2008)
13 IBM Roadrunner supercomputer (http://www.lanl.gov/roadrunner)
14 IBM Cell Broadband Engine Software Development Kit (SDK) (http://www.ibm.com/developerworks/power/cell/)
15 Harvey, M. *et al.* (2007) PS3GRID.NET: building a distributed supercomputer using the PlayStation 3. Distributed and grid computing – science made transparent for everyone. In *Principles, Applications and Supporting Communities*
16 De Fabritiis, G. (2007) Performance of the cell processor for biomolecular simulations. *Comp. Phys. Commun.* 176, 660
17 HPCwire (http://www.hpcwire.com/topic/processors/17910894.html)
18 HPCwire newsletter (http://www.hpcwire.com/hpc/2269095.html)
19 PlayStation3 Safety and Support Manual, Version 2.0, Sony document 3-275-578-31(1) (2008) (http://www.playstation.com/manual/pdf/PS3-02_03-1.9_1.pdf)
20 Morrison, J., Turek, D. The new era of supercomputing: insights from the national labs to wall streetIBM (http://www-03.ibm.com/industries/financialservices/doc/content/bin/ibm_lanl_at_sifma_tech_2007_web.pdf)
21 The green 500 list (http://www.green500.org/lists/2008/02/green500.php)
22 Easton, J. *et al.* (2007) Porting financial markets applications to the cell broadband engine architecture. *Tech. Rep*
23 Zaho, Y. (2007) Lattice Boltzmann based PDE solver on the GPU. *Visual Comput.* 24, 323–333
24 Manavski, S. and Valle, G. (2008) CUDA compatible GPU cards as efficient hardware accelerators for Smith–Waterman sequence alignment. *BMC Bioinformat.* 9, S10
25 Ufimtsev, I. and Martnez, T. (2008) Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *J. Chem. Theory Comput.* 4, 222–231
26 Swaan, P. and Ekins, S. (2005) Reengineering the pharmaceutical industry by crash-testing molecules. *Drug Discov. Today* 10, 1191–1200
27 van Meel, J.A. *et al.* (2008) Harvesting graphics power for MD simulations. *Mol. Sim.* 34 (3), 259–266
28 Stone, J.T. (2007) Accelerating molecular modeling applications with graphics processors. *J. Comput. Chem.* 28, 2618–2640
29 AceMD website (http://multiscalelab.org/acemd)
30 Humphrey, W. *et al.* (1996) VMD: visual molecular dynamics. *J. Mol. Graph.* 14, 33–38
31 Fowler, P. *et al.* (2007) Rapid, accurate, and precise calculation of relative binding affinities for the SH2 domain using a computational grid. *J. Chem. Theory Comput.* 3, 1193–1202
32 Berkeley Open Infrastructure for Network Computing (http://www.boinc.berkely.edu)
33 Litzkow, M. *et al.* (1988) Condor – a hunter of idle workstations. In *Proceedings of the 8th International Conference on Distributed Computing Systems, 1988* pp. 104–111
34 Search for ExtraTerrestrial Intelligence at Home (http://www.setiathome.berkeley.edu)
35 PS3GRID project (http://www.ps3grid.net; http://www.gpugrid.net)
36 De Fabritiis, G. *et al.* (2008) Energetics of K+ permeability through Gramicidin A by forward–reverse steered molecular dynamics. *Proteins* 73, 184
37 Collin, D. *et al.* (2005) Verification of the Crooks fluctuation theorem and recovery of RNA folding free energies. *Nature* 437, 231
38 Bajorath, J. (2002) Integration of virtual and high-throughput screening. *Nat. Rev. Drug Discov.* 1, 882–894
39 Gohlke, H. and Klebe, G. (2002) Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors. *Angew. Chem. Int. Ed.* 41, 2644–2676
40 Ladbury, J.E. (1996) Just add water! the effect of water on the specificity of protein–ligand binding sites and its potential application to drug design *Chem. Biol.* 3, 973–980

41 De Fabritiis, G. *et al*. (2008) Insights from the energetics of water binding at the domain–ligand interface of the Src SH2 domain. *Proteins* 72, 1290
42 Stjernschantz, E. *et al*. (2006) Are automated molecular dynamics simulations and binding free energy calculations realistic tools in lead optimization? an evaluation of the linear interaction energy (lie) method. *J. Chem. Inf. Model* 46, 1972–1983
43 Nvidia (2006) GeForce 8800 GPU architecture overview. *Nvidia Technical Report TB-02787-001_v01*
44 Nvidia (http://nvidia.com)