Hybrid CPU-GPU calculations – a promising future for computational biology

Pydiura Nikolay, Karpov Pavel, Blume Yaroslav

Institute of Food Biotechnology and Genomics NAS of Ukraine, 2A Osipovskogo str., Kyiv, Ukraine

pydiura@gmail.com, karpov.p.a@gmail.com, cellbio@cellbio.freenet.viaduk.net

Abstract. *In this study we analysed the GPU support and parallelisation features of Gromacs 4.6.3, evaluated the performance gain provided by hybrid CPU-GPU calculations as compared to CPU only calculations. We compared compute performance value of the graphics cards available on the market and analysed the possible benefit of the use of CPU-GPU computational nodes in grid environment. All the studies and computations were performed in the framework of our virtual organisation CSLabGrid. Comparative performance analysis was based on molecular dynamics performance results obtained with different hardware systems of the Ukrainian National Grid and data from Folding@Home AnandTech GPU test.*

Keywords

Molecular dynamics, Gromacs, hybrid, stochastic, computational biology, GPU, GPGPU

1 Introduction

Calculations using graphics processing units (GPU) are ideally applicable for the molecular dynamics (MD) simulation tasks due to their intrinsic parallelism.[1, 2] One graphics card contains several hundreds of processor cores executing either a single instruction or a small set of divergent directives in parallel in thousands of threads. To make GPUs programmable as CPUs specific frameworks are required.

One of such frameworks is CUDA (Compute Unified Device Architecture) – a parallel computing platform and programming model designed by NVidia for its cards (http://developer.nvidia.com/category/zone/cuda-zone). CUDA gives developers access to the virtual instruction set and memory of the parallel computational elements in GPUs. A set of programming features supported by NVidia GPU type can be determined by compute capability version value assigned to it (http://developer.nvidia.com/cuda-gpus).[3]

Apart from CUDA, another known API for GPGPU calculations is an Open Computing Language (OpenCL) framework. The key feature of OpenCL is portability. Instead of direct access to the hardware specific technologies, as CUDA does for NVidia GPUs, it provides the developer with abstracted memory and execution models. OpenCL is a general framework programming for heterogeneous platforms. Unlike CUDA, OpenCL is an open industry standard and runs on AMD and Intel CPUs, NVIDIA and AMD-ATI GPUs, and other heterogeneous platforms (processors for mobile devices, industry, etc.).

Both CUDA and OpenCL are low-level libraries and require time consuming programming. In terms of performance both CUDA and OpenCL frameworks are capable to fully utilize the hardware resources. The difference betwen their performance is negligible (http://blog.accelereyes.com/blog/2012/02/20/cuda-and-opencl-benchmarks/) and the end performance mainly depends on the user code optimization.

Currently CUDA remains a *de-facto* standard for GPU calculations. For example the number of CUDA related posts on a developers site StackOverflow exceeds the number of OpenCL posts almost three times (14,374 vs 5,094). Google trends are even more pronounced – 58 vs 7 for CUDA. This is because since first CUDA release in 2007 NVidia has invested a lot in the establishing of GPU calculations and its CUDA. Still, due to its versatility OpenCL can have a promising future.

Gromacs version 4.5 in the year 2010 for the first time provided acceptable speed of MD calculation in Gromacs on CUDA GPUs with OpenMM library.[4] However, this first implementation was very limited in terms of supported

algorithms and MD options and had many unchangeable hard-coded parameters. A noticeable boost from GPU utilisation could be observed then only for implicit solvent calculations.

Gromacs 4.6 introduces a native Gromacs GPU implementation which supports a wider range of algorithms: the Verlet cut-off scheme, PME, reaction-field, and plain cut-off electrostatics making it applicable to a wider number of biological MD tasks. Also, it provides a multi-level hybrid parallelization (MPI + OpenMP + CUDA). The minimum compute capability version of the GPU for Gromacs 4.6 is 2.0. Native support of GPU acceleration using CUDA allows making use of hybrid acceleration i.e. by using GPUs to carry out non-bonded force calculations while bonded forces and PME electrostatics are calculated simultaneously on CPUs.[5] Parallelisation of hybrid calculations both in a single node (with OpenMP) as well as across multiple nodes (with MPI) is supported using domain-decomposition. As a single GPU is assigned to calculate non-bonded data of a domain, the number of GPUs should correspond to the number of processes (or MPI threads). Available CPU cores are apportioned among these processes and a set of cores with a GPU do the calculations on the respective domain.

Depending on the workstation hardware configuration and concrete MD task the load of CPUs and GPUs of the hybrid system may vary. With PME electrostatics, Gromacs 4.6 mdrun program supports automated CPU-GPU load-balancing by redistributing workload from the CPU PME mesh calculations to the particle-particle non-bonded calculations, done on the GPU.[5] So, for maximum performance at the beginning of each run for several hundred iterations scaling of the electrostatics cut-off and PME grid spacing is carried out. The values of these parameters that give optimal CPU-GPU load balance are calculated and applied on the fly.

In this study we analysed the GPU support and parallelisation features of Gromacs 4.6.3, evaluated the performance gain provided by hybrid CPU-GPU calculations as compared to CPU only calculations, compared compute performance value of the graphics cards available on the market and analysed the possible benefit of the use of CPU-GPU computational nodes in grid environment. Studies and computations were performed in the framework of our virtual organisation (VO) CSLabGrid (http://ifbg.org.ua/uk/cslabgrid; http://infrastructure.kiev.ua/ru/monitoring/47/).

2 Platform configuration and test system preparation

Hybrid calculations were carried out on a workstation of the VO CSLabGrid. The hardware configuration was Intel Core2 Quad Q9400 @ 2.66GHz + NVidia GeForce GTX 480 graphics card. The compute capability version of GTX 480 GPU is 2.0. The software configuration used for the analysis was FFTW v3.3.3 (http://www.fftw.org), Gromacs 4.6.3 (http://www.gromacs.org) and NVidia CUDA 5.5 software.

As a test model was used a previously constructed homology model of *Arabidopsis thaliana* alpha beta tubulin dimer. Using Gromacs software package the model was solvated. Na^+ and Cl^- ions were added to simulate the physiological ionic strength (0.15 M NaCl). After that, an energy minimization and a short position restrained run were performed. The final system configuration contains 146,101 atoms in 45,016 residues. The volume of the system box is 1444 nm^3 and the number of the SOL molecules is 44,119 (figure 1A).

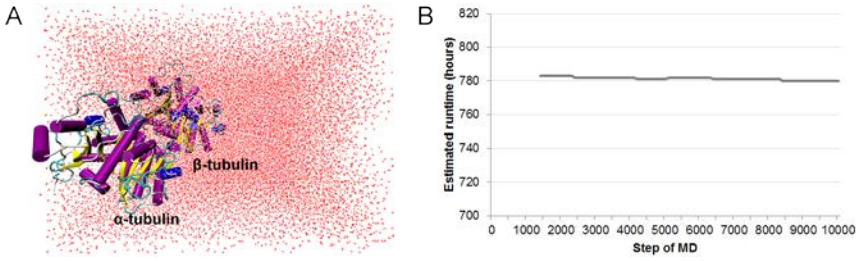


Fig. 1. (A) – the homology model of *Arabidopsis thaliana* alpha beta tubulin dimer. (B) – the plot of the Gromacs time estimation for the 100ns MD simulation of the test system during the initial 10000 steps. Time step was set to 2fs and thus the total number of steps to calculate 100ns equals to 50,000,000. The estimation is calculated at each step basing on the current computational speed. The estimate time of this run is 780 hours.

Amber 99SB force field (http://ffamber.cnsm.csulb.edu) was used for the preparation of the topology. Parameters of the simulation included PME coulomb type (rcoulomb=1nm), Verlet cutoff-scheme (cut-off distance for the short-range neighbor list – rlist=1nm) and vdw-type – cut-off (cut-off distance rvdw=0.9nm as it required to be equal to rlist with such scheme). To analyse the performance of the hardware configurations we used the Gromacs mdrun program runtime estimation data written to the standard output with the mdrun –v option. PME grid spacing adjustment took first 1300 steps. The initial PME grid was 119x119x119, and the adjusted PME grid was 74x74x74. The initial and adjusted nstlist values were 10 and 40 respectively. Runtime estimate time equilibrated during the first 5000 steps of MD around the value of 780 hours (figure 1B).

We analysed the performance of the hybrid calculations on the Q9400 CPU + GTX 480 GPU system and compared it with the performance on the Q9400 CPU alone, and the performance of the same on two different nodes of the Ukrainian National Grid (http://ung.in.ua). The first node was equipped with 2xE5-2650 CPU and located on the cluster of the Institute of Food Biotechnology and Genomics NAS of Ukraine (http://grid.ifbg.org.ua) and the other – with 2xX5650 Intel CPU on the cluster of the Institute of Molecular Biology and Genetics of NAS of Ukraine (http://grid.imbg.org.ua).

The prices of the computer parts in Ukraine were obtained using hotline.ua web-site (as of 10.09.2013). Prices of the nodes are approximate as the final price much depends on the particular hardware configuration and vendor.

3 Results

3.1 Performance analysis

The results show that Q9400 + GTX 480 hybrid system showed 62% better performance than Q9400 CPU alone (figure 2 A,B). At the same time, the performance of the Q9400 + GTX 480 system was 55% slower than the 2xE5-2650 CPU node. Thus, taking into a count a high workload o f computational resources of the Ukrainian National Grid, an outdate Q9400 workstation equipped with proper graphics card can be a good option for shorter MD calculations.

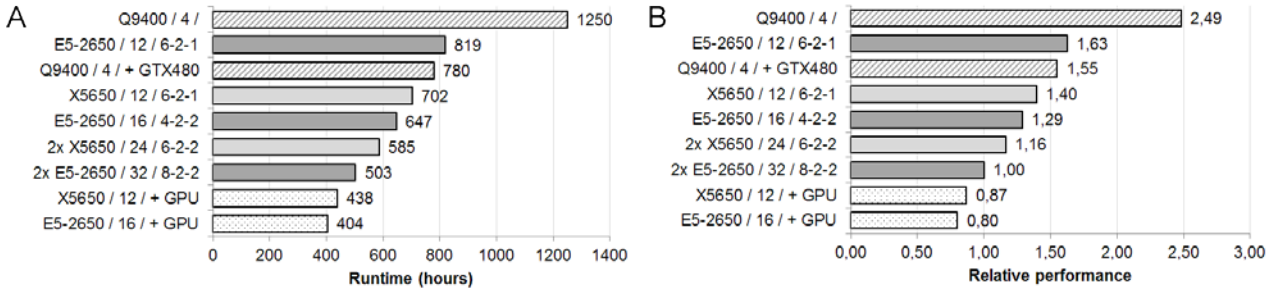


Fig. 2. MD performance results obtained with different hardware systems. Format: CPU / number of cores used / domain decomposition. (A) – absolute execution time (hours). (B) – execution time relatively to the 2xE5-2650 CPU configuration.

Knowing the performance of E5-2650 and X5650 CPUs in this MD task and taking the same value of the performance gain we have also estimated the possible performance of these CPUs in combination with a mid/high level graphics card. The estimated performance of the E5-2650 + mid-level GPU would be 20% faster than that of a 2xE5-2650 CPU node (figure 2 A,B).

3.2 Methods of utilisation of GPUs for scientific calculations

The shift of the most compute-intensive part of simulations to GPUs can allow reaching a noticeable productivity boost. A 62% increase of performance observed for our configuration is impressive but far from performance boost of 200% and even 300% reported for the latest NVidia graphics cards (http://www.nvidia.com/docs/IO/122634/GROMACS-benchmark-report.pdf). Taking into account low cost of the graphics cards as compared to the price of an extra node, equipment of the computational nodes with mid- and high-level GPUs could be very economically attractive. For example, if 2xE5-2650 CPU nodes (5900$) were equipped with at least 2 mid-level graphics cards (500$ for both) this would increase the computational power of this node in MD tasks about two-fold.

There are at least three distinct computational system types that can be built basing on GPUs. The possibility of the utilisation of a concrete graphics card for each of the schemes is determined by the supported programming features expressed as compute capability version of the GPU.

I) GPU only scheme, where calculations are run on GPU only and CPU stands idle. In such a scheme usually only one GPU per node is used. If it is necessary to scale system beyond one GPU the speed of exchange GPU1<->CPU/RAM<->GPU2 becomes a bottleneck. GPUs with any compute capability version can be used for this scheme. GPUs with compute capability <2.0 can be efficiently used only in this way and allow GPU calculations only of simple highly parallel tasks. Such cards can still be applied now for such tasks as docking procedures where simple evaluation of millions of protein-ligand orientations is required.

II) Hybrid CPU-GPU scheme where CPU generates tasks for GPU and does the balancing. GPUs with compute capability >2.0 can be used for hybrid CPU-GPU calculations. Load balancing is made between CPU-GPU partitions

and each such partition works on a relatively separate part of the system. So, no GPU-GPU interaction is needed and thus, such scheme allows efficient parallelisation on one node or between different nodes. But still, for optimal performance with multiple GPUs, especially with parallelisation on multiple nodes, it is preferable to use identical hardware. Depending on the hardware and tasks specificity, one GPU per several cores of CPU may be installed in such systems. Such scheme very promising for MD simulations. We found configurations with one mid/high end GPU per 12-16 CPU cores reasonable for such tasks.

III) "GPU-only clusters" where GPUs generate tasks for GPUs and do the load balancing. Such scheme becomes possible with compute capability 3.5 which introduces "dynamic parallelism" for GPUs. GPUs can be located on one or different nodes. Since during the computations the CPUs are mostly idle and to avoid data exchange delays it is necessary to minimize the number of nodes by maximizing the number of GPUs per node. These systems can be built on special platforms that support multiple GPUs – 4 and more. There are no extra requirements for CPUs and RAM size because GPUs and video memory are mainly used. GPU clusters are intended for tasks that require unprecedented performance.

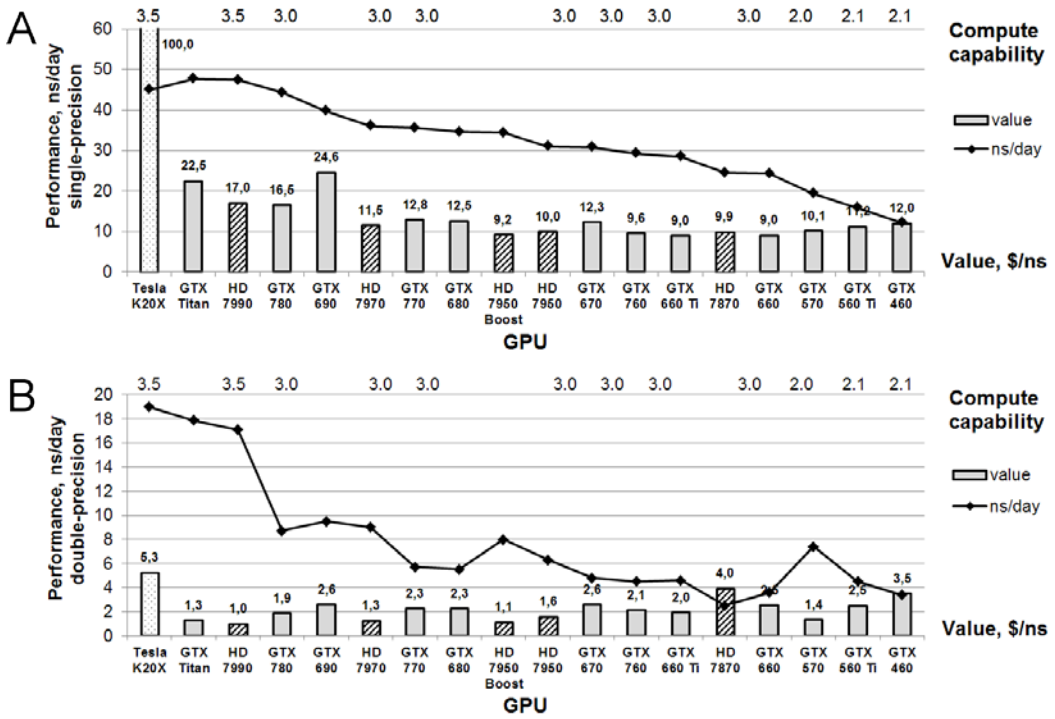
Starting from the first cards with compute capability 3.x (Kepler GPUs), NVidia is splitting graphics and compute into two distinct product lines, meaning that mainstream graphics cards will no longer fit for computational tasks and the compute cards are becoming more and more expensive. Currently, for all spheres of scientific computing most in demand are nodes capable of running hybrid CPU-GPU calculations. Such systems can be built both on the basis of new nodes as well as on the basis of already existing nodes and workstations by equipping them with graphics cards of a corresponding productivity. Equipment of the older workstations with high end graphics cards of the previous generations which support compute capability 2.0 and higher can boost the performance of these nodes in hybrid calculations by 60% and more at a low cost. Taking this into account, it is important to analyse the compute performance value of the available NVidia graphics cards.

3.3 Estimation of the performance value and discussion

Graphics card manufacturers offer two distinct product lines. First are consumer level cards which are used for gaming and graphics – the "gaming" cards (NVidia GeForce and AMD-ATI Radeon series). Second are professional cards intended for production and computing (Nvidia Quadro, Nvidia Tesla, and AMD-ATI FirePro series). So, Nvidia Tesla professional series provides GPUDirect RDMA for InfiniBand performance, Hyper-Q for MPI and supports tools for GPU and cluster management. The main advantages that professional graphics cards provide for scientists apart from better clustering capabilities are better double-precision performance and error-correcting code (ECC) option. Since for the tasks solved by consumer cards single precision is sufficient enough, Nvidia and AMD-ATI, in fact, artificially cap the double precision performance on consumer cards by about a factor of 1/8 of that of single precision to justify the prices of professional cards with the same GPU cores where this factor is 1/2 to 1/3 (http://www.nvidia.com/object/why-choose-tesla.html). This also allows lowering the thermal design power (TDP) of consumer cards by turning off double precision units which is important for desktops and portable devices.

The accuracy of calculations provided by double-precision floating point numbers and ECC are indispensable for such tasks as solving of mathematical problems, financial operations, (de)cyphering, economic modeling, etc. On the other hand, MD as well as a considerable amount of other biological and biophysical processes (protein-protein and protein-ligand interactions, ferment reactions, cell cycle, gene expression, population genetics, etc.) are stochastic. Thus, neither double precision nor ECC is required for the modelling of these processes. Gromacs as well as other molecular modelling applications (Gamess, Amber, hmmer, NAMD, etc.) calculations are compute-bound, thus, rough performance of the processing units is more important than memory size and bandwidth. This all justifies the utilisation of the consumer GPUs for scientific calculations of the stochastic processes. Besides Gromacs, similar stochastic algorithms are used for MD simulation in such packages as Amber, Charmm, DL_POLY, LAMMPS, NAMD, etc.

Compute performance shows the capability of GPUs to perform programmed calculations. AnandTech GPU test provides the graphics card compute performance in Gromacs based Folding@Home task (http://www.anandtech.com/bench/GPU13/591) – figure 3. The performance of the Tesla K20X was estimated basing on its comparison with GTX Titan and GTX 680 in other tests on Nvidia official site. The conditions of the test task are explicit solvent and single-, and double-precision. Average prices in Kyiv of the presented in this test GPUs were obtained using hotline.ua web-site. For each graphics card we divided an average price ($) by a daily compute performance ns/day and obtained the value of the graphics cards productivity as $ for ns/day (figure 3). Less compute performance value is better.



C

GPU	Price, $
Tesla K20X	4500
GTX Titan	1073
GTX 780	732
GTX 690	976
GTX 770	457
GTX 680	433
GTX 670	378
GTX 760	280
GTX 660 Ti	256
GTX 660	220
GTX 570	195
GTX 560 Ti	177
GTX 460	146
HD 7990	805
HD 7970	415
HD 7950 Boost	315
HD 7950	309
HD 7870	242

Fig. 3. Single- (A) and double-precision (B) daily compute performance, ns/day of the latest Nvidia and AMD-ATI graphics cards in Gromacs based Folding@Home task from AnandTech GPU test. (C) – Average prices of these graphics cards in Kyiv, Ukraine. Compute performance values are obtained by division of an average price, $ of a graphics card by its daily compute performance, ns/day.

As it can be seen from figure 3, the best single-precision compute performance value among Nvidia graphics cards to date have GTX 660, GTX 660 Ti and GTX 760 graphics cards. The prices of these cards (10.09.2013) are 220$, 256$ and 280$ respectively. All these cards have 2Gb of video RAM and compute capability 3.0. The number of stream processors is 1344@915Mhz in GTX 660 Ti, 960@1020Mhz in GTX 660 and 1152@1006Mhz in GTX 760. All this together makes these cards the best choice today for hybrid calculations. Besides that, GTX 760 has a 256-bit memory interface width while two other cards have a 192-bit interface and thus, may be a more interesting variant for memory-bound tasks. The compute performance of these cards is high enough to pair them with the latest server CPUs (GTX 660 Ti or GTX 760) and desktop CPUs (GTX 660).

At present time there are only four Nvidia GPUs with compute capability 3.5 (Tesla K20X, K20, GeForce GTX Titan, GTX 780) and the prices of these graphics cards remain high. For example, the value of GTX 780 compute performance is 85% higher than that of GTX 660 Ti. Besides that, the benefits of these cards are revealed only if several cards (four and more) are used in parallel as a “GPU-only cluster”. As it was mentioned above, these cards should better be located on one node. The hardware for multi GPU platform is very costly and there are only a few tasks that really require “GPU-only clusters”. Tests of Tesla GPUs in hybrid CPU-GPU MD calculations in different software packages (http://www.nvidia.in/object/k20-gpu-test-drive-in.html) show that for a 2xE5-26XX CPU node a configuration with two K20X cards provides on average only 30% performance gain over the same configuration with one K20X card. Thus, for hybrid calculations only one K20X card per 2 CPU node is rational. At the same time, two GTX 660 Ti cards installed on such node could provide the same number of CUDA Cores (1344x2) as one K20X card (2688 CUDA cores) but for 9 times less money (500$ vs 4500$). For the sake of stability, professional cards have a slightly lower GPU clock speeds than “gaming” cards – e.g. 837/876Mhz and 732Mhz in GTX Titan and K20X respectively which are both built on GK110 GPU. Thus, “gaming” GeForce GPUs provide often a better single-precision performance per core than professional cards. Still, professional GPUs provide full clustering options and unprecedented double precision performance and hardware reliability where it is required. On the other hand, for stochastic modelling tasks one can get the most benefits from the use of consumer level graphics cards for hybrid CPU-GPU calculations.

As to the performance of AMD-ATI graphics cards, in single-precision calculations it corresponds to that of Nvidia cards of the same level. Radeon HD 7950 (Boost) has the best price/performance value among top ATI consumer level cards. At the same time, starting from Radeon HD 7890 ATI cards have a lesser single-/double-precision capping (1/3) as compared to Nvidia cards (1/8 except for GTX Titan where it is also 1/3). Thus, HD 7950 Boost on a par with HD 7990 has one of the best double-precision performance values. Radeon cards can be a good choice for entry-level double-precision calculations with OpenCL.

4 Summary

In conclusion, we would like to emphasize a few points:

- though not perfect, hybrid CPU-GPU calculations at present time are a very efficient way to utilize GPUs for scientific calculations in Ukrainian National Grid. Hybrid MD calculations in Gromacs 4.6.3 have some issues with load balancing while running on multiple GPUs which are to be solved;
- any cluster node or workstation intended for scientific calculations should be obligatorily equipped with one or two graphics cards with a corresponding to CPU performance;
- scientists can make a good use of consumer level "gaming" graphics cards for stochastic scientific calculations in hybrid CPU-GPU systems;
- NVidia GTX 760, GTX 660 Ti, GTX 660, and ATI Radeon HD 7950 Boost graphics cards have the best single-precision compute performance value among consumer level graphics cards at present time. Besides that, ATI Radeon HD 7950 Boost on a par with HD 7990 has one of the best double-precision performance values;
- the time of GPU only clusters is still yet to come.

5 Acknowledgments

All molecular computations were carried out with the assistance of The Ukrainian National Grid (UNG) – (http://ung.in.ua/), and we would like to acknowledge Targeted State Scientific and Technological Program on the Development and Implementation of Grid-Technologies for 2009-2013 (http://grid.nas.gov.ua/).

References

[1] Goga N., Marrink S., Cioromela R., Moldoveanu F.: GPU-SD and DPD parallelization for Gromacs tools for molecular dynamics simulations. *IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)*, 251-254, 2012.

[2] Demchuk O., Karpov P., Blume Ya.: Docking Small Ligands to Molecule of the Plant FtsZ Protein: Application of the CUDA Technology for Faster Computations. *Cytology and Genetics*. V. 46, No. 3.:172-179, 2012.

[3] Ng C.M.: Novel Hybrid GPU-CPU Implementation of Parallelized Monte Carlo Parametric Expectation Maximization Estimation Method for Population Pharmacokinetic Data Analysis. *AAPS J.* Sep 4. [Epub ahead of print], 2013.

[4] Pronk S, Páll S, Schulz R, Larsson P, Bjelkmar P, Apostolov R, Shirts MR, Smith JC, Kasson PM, van der Spoel D, Hess B, Lindahl E.: GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics.* 29(7):845-54, 2013.

[5] Apostolov R., Hess B., Laure E.: Improving the Efficiency and Scalability of Life Science Software. *EScience*, (http://www.ci.uchicago.edu/escience2012/pdf/escience2012_submission_188.pdf), 2012.