Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

HPC Methodologies for PharmScreen

Aleksey Kondratyev[a], Thomas Ponweiser[a]*, Enric Gibert[b], Enric Herrero[b]

[a]*RISC Software GmbH, Softwarepark 35, 4232 Hagenberg, Austria*
[b]*Pharmacelera S.L., Pl. Pau Vila, 1, Edifici Palau de Mar, 08039 Barcelona, Spain*

Abstract

Pharmacelera is a company that develops hardware and software solutions for drug discovery. Among other products, Pharmacelera offers PharmScreen, a revolutionary software tool for ligand-based drug design. PharmScreen scans compound databases consisting of hundreds of thousands of molecules and identifies potential hits by comparing molecules using a full 3D representation of its interaction fields. In the course of the SHAPE project "HPC methodologies for PharmScreen", Pharmacelera has been exploring the potential of heterogeneous HPC platforms for PharmScreen and, in cooperation with RISC Software, has been working on (1) performance profiling and analysis, (2) porting computation kernels to OpenCL, enabling the usage GPU, Xeon Phi and other accelerator platforms, (3) porting the application to MareNostrum, (4) static code analysis and refactoring, and (5) improving the testing methodology.

1. Introduction

In Computer-Aided Drug Design, chemists have high interest in accurate computational algorithms for finding relevant candidate molecules that have higher success probabilities in later drug discovery stages. At the same time, computational costs have to be kept under control for avoiding a negative impact on the drug time-to-market. Pharmacelera, a Spanish SME located in Barcelona, has developed PharmScreen, a revolutionary software tool for ligand-based drug design. PharmScreen scans compound databases consisting of hundreds of thousands of molecules and identifies potential hits by comparing molecules using a full 3D representation of its interaction fields. In collaboration with the Applied Medicine Research Center (Universidad de Navarra), Pharmacelera has shown that PharmScreen finds up to 94% of the active molecules for a given set of compounds where the hit rate of current state of the art software is only at about 26%. The results of a poll to potential customers of Pharmacelera indicates that 2/3 of them would be interested in using PharmScreen if its time-to-solution is reduced.

In the course of the SHAPE project "HPC methodologies for PharmScreen", Pharmacelera has been exploring the use of heterogeneous HPC platforms (CPUs, GPUs, FPGAs) in order to reduce PharmScreen's execution time and to develop integrated hardware/software solutions that are more appealing to their customers. In particular Pharmacelera pursued the following four main goals:

- Defining a parallelisation strategy for PharmScreen using a combination of traditional CPUs, GPUs and highly parallel vector machines and setting up an appropriate testing methodology for such a HPC solution.
- Implementing a first parallel version of PharmScreen beyond OpenMP (which has already been available).
- Exploring HPC alternatives for Pharmacelera's hardware platforms PharmaBox and PharmaBlade.
- Assess the performance and accuracy of PharmScreen by doing a sensitivity analysis of its different parameters.

* Corresponding author. *E-mail address*: thomas.ponweiser@risc-software.at

1.1. Structure of the Document

The rest of this document is structured as follows: In section 2, we report on scalability analyses and parameter studies for PharmScreen which have been carried out on the MareNostrum supercomputer of the Barcelona Supercomputing Center (BSC). In section 3, we report on activities and results related to code profiling, performance optimization and porting PharmScreen to accelerator hardware. Other activities, like improving the testing methodology and quality of the code are briefly summarized in section 4. Section 5 gives an overall conclusion on the project and its outcomes.

2. Scalability Analysis and Parameter Studies

The scalability of the OpenMP-based (i.e. shared-memory) parallelization of PharmScreen has been evaluated on the MareNostrum supercomputer at the Barcelona Supercomputing Center (BSC). Several applications have been tested with a variable number of threads and with different input set size in order to assess the influence of initialization and data gathering stages. Figure 1 shows the speedups of PharmScreen showing a very nice scalability for the evaluated datasets.

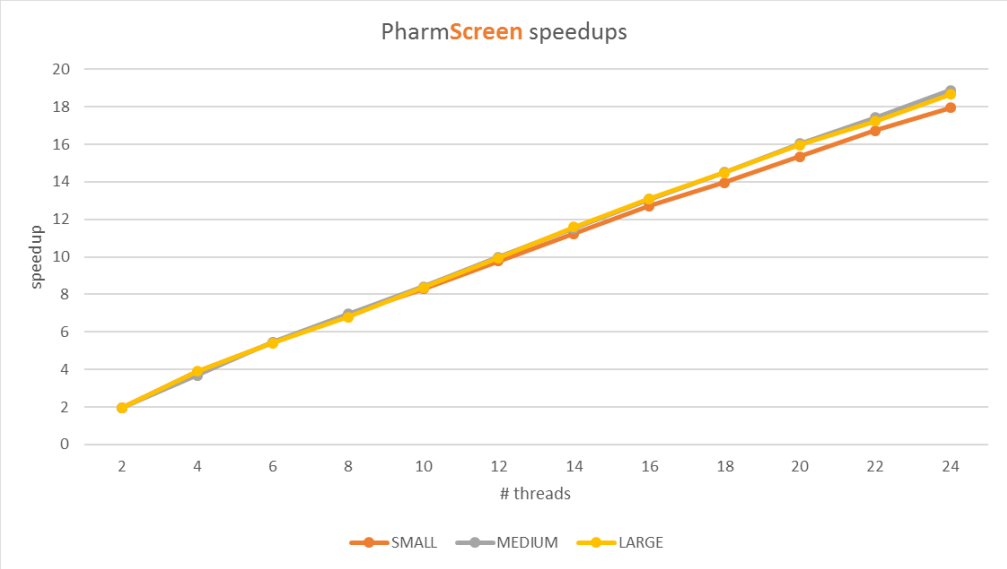

Figure 1: Speedups of PharmScreen execution for a variable number of threads and different input sizes (small, medium and large)

Moreover, the availability of the MareNostrum supercomputer allowed to perform multiple sensitivity studies of PharmScreen configuration parameters. Among them, the influence of grid spacing in the quality and simulation time was performed.

Grid spacing influence (Quality: From 0 to 1)						
	PreAlign grid spacing (Å)					
Final align grid spacing (Å)	0,5	1	1,5	2	3	4
0,5	0,815	0,838	0,839	0,853	0,837	0,816
1	0,826	0,8258	0,84	0,854	0,838	0,827
1,5	0,8255	0,824	0,843	0,8541	0,837	0,826

Table 1: Influence of the grid spacing parameters (for the pre-align and final align phases of the computation) on result quality. Results with best quality are highlighted in green.

Grid spacing influence (Simulation time h)						
	PreAlign grid spacing (Å)					
Final align grid spacing (Å)	0,5	1	1,5	2	3	4
0,5	5,783	3,162	2,911	2,915	2,914	2,914
1	3,714	1,137	0,900	0,862	0,743	0,738
1,5	3,373	0,821	0,732	0,569	0,498	0,492

Table 2: Influence of the grid spacing parameters (for the pre-align and final align phases of the computation) on simulation time. Fastest configurations are highlighted in green.

Very interesting conclusions were drawn from this study since it provided results that were not expected. Simulation time for different grid values was reduced as the spacing was higher, however, quality of results did not improve after a certain distance reduction. Performed sensitivity studies have allowed Pharmacelera to optimize PharmScreen and increase its quality.

3. Performance Optimization

For improving the performance of PharmScreen, potentials for optimizing the CPU variant of the code and for porting it to accelerator hardware, primarily GPUs, but also Intel Xeon Phi (through OpenCL), have been investigated. For this purpose, the following tasks have been carried out:

- Definition of benchmarks
- Profiling the base CPU version, bottleneck identification and optimization
- Identification of code regions suitable for usage of accelerator hardware (main focus on GPU)
- Porting identified computation kernels to CUDA (GPU) and OpenCL (GPU and Xeon Phi)
- Benchmarking of the hybrid code versions

3.1. CPU: Profiling, Optimization and Results

Code profiling with Intel VTune revealed that more than 75% of the overall runtime of the CPU variant of PharmScreen was consumed by loops over a regular three-dimensional grid. The body of those loops consisted mainly of simple algebraic operations in vector form.

For optimizing the performance of the identified critical code sections, the underlying hierarchical data structure has been reorganized to plain contiguous arrays which are well suited for concurrent vector operations. Two vectorization approaches were investigated by RISC Software: 1) Usage of BLAS routines and 2) Usage of Vector Math functions, both provided by Intel MKL. In both cases, a moderate speedup of about 10% has been achieved.

3.2. Accelerator Hardware: Porting, Profiling and Results

Previously identified critical code sections (see 3.1) also turned out to be good candidates for the usage of accelerator hardware. For this purpose, two approaches have been investigated: Pharmacelera developed a CUDA-based implementation of the computation kernels for running PharmScreen on Nvidia GPUs and RISC Software provided an OpenCL-based version, enabling the use of a broader class of accelerators, as for example Intel Xeon Phi.

For the CUDA version, the code was parallelised in 5 different ways making a different usage of the GPU resources in order to assess the optimal configuration. In each configuration, the problem is dimensioned in different ways, either by configuring different sizes for grids and grid points assigned to each GPU block, giving more or less work to threads or making use of the different levels of memory.

Figure 2 shows the speedups obtained in this study depending on the number of 3D grid points and the configuration used. This study enabled finding the optimal GPU parallelization technique in order to use it in the PharmScreen software. Note that the underlying hardware for this study is a workstation with Intel i7 4790 CPU, 8 GB RAM, SSD hard drive and an Nvidia Quadro M4000 GPU.

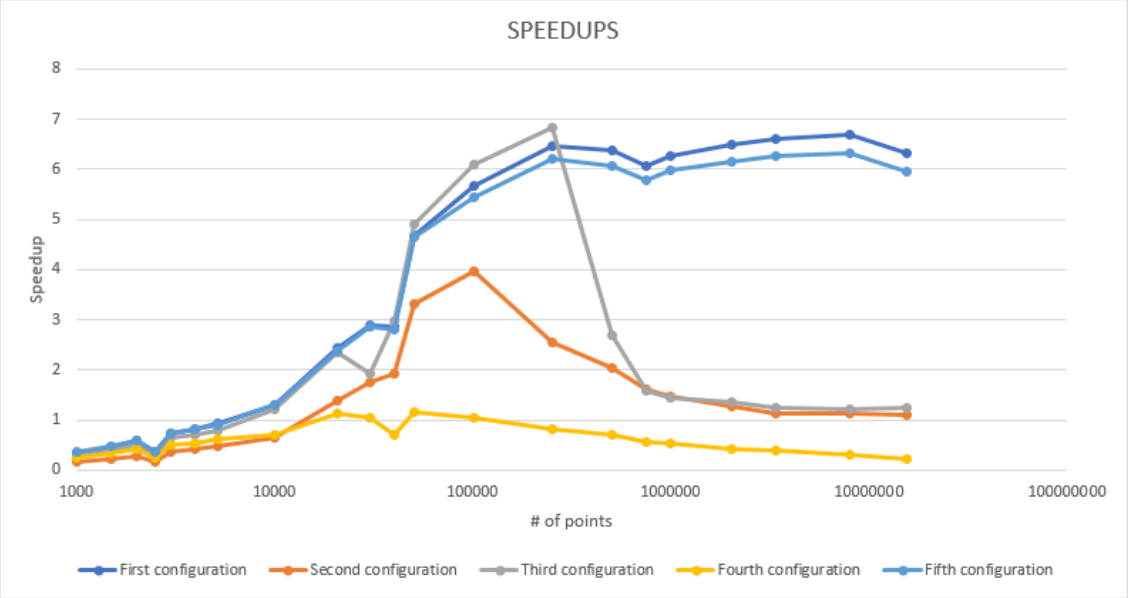

Figure 2: Speedups of the CUDA version for a variable number of points and different implementations

In Figure 2 it can be seen how the speedup provided by the GPU implementation varies depending on the number of grid points. It is clear from the figure that the GPU implementation only makes sense when the number of points is 50.000 or more. Additionally, it can be seen that some configurations are not able to use the GPU efficiently, others scale up to a certain number of points and others keep a constant speedup of 6-6.5x. Differences among GPU configurations performance are mostly due to how the data is allocated in the GPU, which in some cases generates more overheads or in other forces under-utilization of GPU resources. Those configurations that divide the problem in sizes that better fit GPU memory structures are those that are able to execute faster. In configuration 3, for example, data is partitioned in a much lower number of blocks handling a higher amount of points. Since the data does not fit the Streaming Multiprocessor (SM) it forces a reduction in the number of active threads which results in a lower occupancy (38.5% vs 73% of configurations 1st and 5th), resulting in a much lower performance. From the obtained results, it is clear that the first and fifth configurations are the most interesting to be used by PharmScreen.

The performance results for the OpenCL-based version turned out to be good for GPUs (speedup of factor 5, comparing CPU and GPU version of PharmScreen running on a workstation with Core i7-965XE CPU and Nvidia GTX 780ti GPU), but quite unsatisfactory for Intel Xeon Phi (which in fact turned out to be slower than the OpenCL-based CPU version of the code).

4. Code analysis and enhancements

In addition to topics related to performance optimization and porting to accelerator hardware, RISC Software also supported Pharmacelera in further improving the software quality and stability of PharmScreen.

Firstly, various code analysis techniques and tools were employed to PharmScreen. This includes analyzing and eliminating warnings emitted by different compiler suites (such as gcc and clang), as well as using the code analysis tools clang-tidy and cppcheck (both for static code analysis) as well as the valgrind memory checker (dynamic code analysis). In this way, several issues (most of them minor) have been detected and successfully resolved.

Secondly, also the testing methodology has been improved by migrating parts of the testing code to the Google Test unit testing framework. Among other features and benefits, Google Test provides parameterized tests, which is a mechanism to implement a test template once and instantiate it in multiple ways. In this way, the amount of boilerplate testing code could be reduced significantly, having an overall positive effect on the maintainability of the testing code base.

5. Conclusion

Partners from RISC Software supported the SME with (1) performance profiling and analysis, (2) porting computation kernels to OpenCL, enabling the usage GPU, Xeon Phi and other accelerator platforms, (3) porting

the application to MareNostrum, (4) static code analysis and refactoring, and (5) improving the testing methodology. Computational resources for carrying out scalability and sensitivity analyses were provided in the course of the equally named PRACE Preparatory Access Type B project 2010PA3391 by the Barcelona Supercomputing Center (BSC; MareNostrum supercomputer).

The SME has benefited of the HPC expertise of the PRACE partners in order to adapt its proprietary software to run on HPC infrastructures. Moreover, PRACE has provided them with computing hours that have allowed Pharmacelera to optimize the tool and make it more efficient.

The collaboration between Pharmacelera and RISC Software is considered unproblematic and fruitful by both parties. Initially, Pharmacelera expressed preference for a support expert located near to their site. However in the end, fortunately neither spatial distance nor any language barriers did affect the quality of the collaboration.

Key for the success of the project were regular Skype meetings as well as the responsiveness and commitment of both parties. In addition, the well-structured cloud-based document sharing and source version control provided by Pharmacelera were the basis for a frictionless collaboration. To conlcude, Pharmacelera is very satisfied with the project outcomes and both parties hope for the opportunity to collaborate again in the future.

Acknowledgements

This work was financially supported by the PRACE project funded in part by the EU's Horizon 2020 research and innovation programme (2014-2020) under grant agreement 653838. We acknowledge that the results in this paper have been achieved using the PRACE Research Infrastructure resource MareNostrum at Barcelona Supercomputing Center (BSC), Spain.