

Advantages of GPU technology in DFT calculations of intercalated graphene

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2014 Phys. Scr. 2014 014027

(<http://iopscience.iop.org/1402-4896/2014/T162/014027>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

This content was downloaded by: pesicj

IP Address: 147.91.1.43

This content was downloaded on 17/10/2014 at 10:41

Please note that [terms and conditions apply](#).

Advantages of GPU technology in DFT calculations of intercalated graphene

J Pešić and R Gajić

Center for Solid State Physics and New Materials, Institute of Physics, University of Belgrade, Pregrevica 118, 11080 Belgrade, Serbia

E-mail: yelena@ipb.ac.rs

Received 19 September 2013

Accepted for publication 10 February 2014

Published 19 September 2014

Abstract

Over the past few years, the expansion of general-purpose graphic-processing unit (GPGPU) technology has had a great impact on computational science. GPGPU is the utilization of a graphics-processing unit (GPU) to perform calculations in applications usually handled by the central processing unit (CPU). Use of GPGPUs as a way to increase computational power in the material sciences has significantly decreased computational costs in already highly demanding calculations. A level of the acceleration and parallelization depends on the problem itself. Some problems can benefit from GPU acceleration and parallelization, such as the finite-difference time-domain algorithm (FDTD) and density-functional theory (DFT), while others cannot take advantage of these modern technologies. A number of GPU-supported applications had emerged in the past several years (www.nvidia.com/object/gpu-applications.html). Quantum Espresso (QE) is reported as an integrated suite of open source computer codes for electronic-structure calculations and materials modeling at the nano-scale. It is based on DFT, the use of a plane-waves basis and a pseudopotential approach. Since the QE 5.0 version, it has been implemented as a plug-in component for standard QE packages that allows exploiting the capabilities of Nvidia GPU graphic cards (www.qe-forge.org/gf/proj). In this study, we have examined the impact of the usage of GPU acceleration and parallelization on the numerical performance of DFT calculations. Graphene has been attracting attention worldwide and has already shown some remarkable properties. We have studied an intercalated graphene, using the QE package PHonon, which employs GPU. The term ‘intercalation’ refers to a process whereby foreign atoms are inserted onto a graphene lattice. In addition, by intercalating different atoms between graphene layers, it is possible to tune their physical properties. Our experiments have shown there are benefits from using GPUs, and we reached an acceleration of several times compared to standard CPU calculations.

Keywords: GPU, DFT, graphene, intercalation, first principle

(Some figures may appear in colour only in the online journal)

Introduction

A development of general-purpose graphic-processing unit (GPGPU) technology has had great impact on computational science, and it is shown to be a valuable resource when it comes to the high-resource-demanding calculations of solid-state physics. GPGPUs use a graphic-processing unit to perform calculations usually handled by the central-processing unit (CPU).

We used Quantum Espresso code and examined the impact of GPU acceleration and parallelization on the density functional theory (DFT) calculations of graphene.

Calculation method

Although GPUs have been an essential part of any computer for decades, it became much more important during the last

decade of twentieth century. The era of three-dimensional (3D) graphics in the gaming industry started in the mid-1990s, but for scientific use, GPUs showed their true potential in the first years of 2000s when GPU programmability was introduced.

High-performance computing (HPC) became widely available thanks to GPU computing and the fact that it brought the performance of the most powerful supercomputers of ten years ago to today's affordable workstations [1]. GPUs nowadays are an important platform for general-purpose computing.

A GPU is a massively parallel processor with a large memory bandwidth. Its memory is hierarchically organized, and the transitions between various memory levels are explicitly defined and organized by the programmer.

One or several GPUs can be easily included in a common workstation and supercomputer nodes as well. Hybrid combinations of one or more GPUs with aCPU are one of the main choices of HPC laboratories worldwide because of their indisputable combination of high performance and low price.

GPUs are valuable tools for calculations and numerical simulations, but it is very important to emphasize that not all problems can benefit equally from using a GPU. Certain problems are native for parallelization and GPU acceleration. However, the DFT calculation method is shown to have great success in exploiting the advances of GPU technology.

DFT is a method for *ab-initio* electronic structure calculations based on Kohn-Sham equations. For solving Kohn-Sham equations, there are several numerical approaches and approximations, each based on discretization of equations and the treatment of core electrons [2]. An iterative procedure, known as the self-consistent field (SCF) method, is used to find solutions to eigenproblems.

Quite a few codes and programs have been developed for DFT calculations. In the past few years, certain codes have been reported to support GPU technology. The main idea behind porting existing DFT code to GPUs is to discover methods for improving the computationally expensive parts of the SCF loop and re-implementing them with GPUs [2]. This is achieved by replacing common computational libraries with GPU versions, such as compute unified basic linear algebra subroutines (CUBLAS), compute unified fast Fourier transform (CUFFT), among others, and by writing custom kernels for GPUs. The final goal is performance speedup by more than several times while minimizing the poor transfer time of data between a CPU and a GPU, which is one of the biggest bottlenecks.

Quantum Espresso (QE) is an integrated suite of open-source computer codes for electronic-structure calculations and materials modeling at the nano-scale [3] and is based on DFT, the use of a plane-waves basis and a pseudopotential approach. Since version 5.0, certain packages of QE have been able to exploit the capabilities of Nvidia GPUs by porting the most computationally expensive parts of the SCF cycle to run on GPUs. QE uses CUFFT and the Magma computational software, as well as phiGEMM numerical library operations, which are parallel hybrid replacements for GEMM [4]. It has been shown that the best practice is to

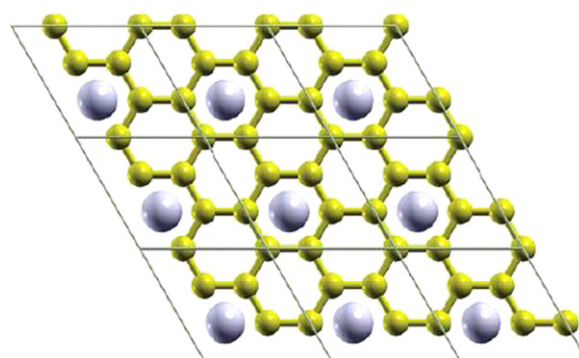


Figure 1. Crystal structure of monolayer graphene with lithium adatoms. Small yellow spheres represent carbon atoms, whereas larger violet spheres represent lithium adatoms placed in the hollow sites of the graphene layer at H-sight.

Table 1. Optimized structural parameters, lattice constant (*a*) and adatom-graphene distance (*h*) in Å.

	<i>a</i>	<i>h</i>
LiC ₆	4.27	1.73

employ OpenMP parallelizations within one node, MPI parallelizations across multiple nodes, and GPU acceleration where possible for the studied problem.

Graphene

Graphene has been attracting attention worldwide and has shown some remarkable properties [5]. Our goal is to show the advantages of applying GPU technology to DFT calculations for graphene. We chose the QE package PHonon, which employs GPU technology. It implements density-functional perturbation theory (DFPT) to calculate second- and third-order derivatives of energies with respect to atomic displacement and electric fields [6].

In this study, we examined intercalated graphene. The term 'intercalation' refers to a process where foreign adatoms are inserted onto a graphene lattice. Studies have shown it is possible to tune properties of the graphene by doping the surface with alkaline metal adatoms [6]. This is analogous to graphite-intercalated compounds (GIC).

Results

Methods

The reported results were obtained from first-principles DFT in a local density approximation (LDA). We used QE with a norm-conserving pseudopotential and the plane-wave cutoff energy of 65Ry. A monolayer system was simulated in $\sqrt{3} \times \sqrt{3} R30^\circ$ in-plane unit cell with one lithium adatom per unit cell and 6 Å of vacuum between layers (to simulate a monolayer) (figure 1).

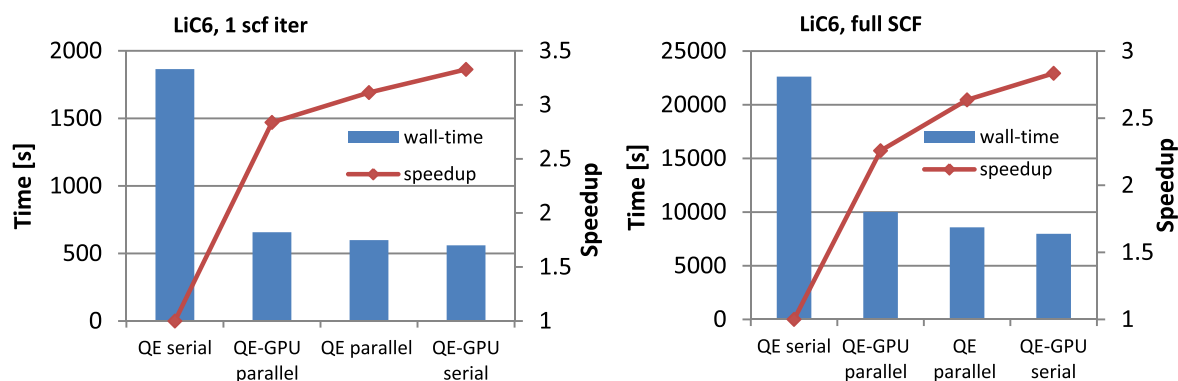


Figure 2. QE-PWscf calculation on Intel Core i7-3930K at 3.8 GHz with Nvidia Tesla K20 (a) single SCF iteration and (b) full SCF calculation.

All structures were relaxed to their minimum energy configuration (in respect to the stress tensor of the unit cell and the internal forces on atoms) (table 1).

GPU acceleration

All QE PH calculations consist of several parts:

1. SCF calculations on a dense grid that consists of all k and $k+q$ grid points.
2. and 3. Normal SCF and phonon dispersion calculations on the coarse k -point grid.

To show differences in calculation wall-times, the same input was used for four different system configurations.

- QE in serial configuration
- QE in parallel configuration
- QE in serial configuration + GPU
- QE in parallel configuration + GPU

Computational resources

We present data obtained using a heterogeneous system equipped with a GPU. All calculations have been executed on the workstation at the GPU Group for Simulations in Solid State Physics at the Center for Solid State Physics and New Materials at the Institute of Physics, with Intel Core i7-3930K at 3.8 GHz with 32 GB RAM. Calculations were performed employing OpenMPI parallelization on a four-core system, with two parallel processes and MKL multi-threading, using four MKL threads.

The GPU-accelerated calculations have been executed on the Nvidia's Tesla K20, which consists of 2496 compute unified device architecture (CUDA) cores¹.

We compared wall-times for QE-PWscf.

Table 2. Wall time and speedup for the QE-SCF calculation on a dense grid for the monolayer graphene with Li adatoms for four different system configurations.

Configuration	Wall time		Wall-time	
	1 SCF iter [s]	Speedup 1 iter	full SCF [s]	Speed up full SCF
QE serial	1864.7	1	22 620	1
QE-GPU parallel	657.1	2.84	10 020	2.26
QE parallel	598.8	3.11	8580	2.64
QE-GPU serial	560.3	3.33	7980	2.83

Conclusion and further study

We have shown that using GPU technologies together with CPU processing can decrease the calculation duration for graphene with adatoms by up to three times (figure 2).

An important fact is that the level of acceleration that can be reached depends significantly on the problem itself. As shown in table 2, for this kind of calculation with the workstation used, the best wall-times are achieved for QE serial with GPU. It is important to emphasize that the problem used in this calculation is rather small, and on larger systems the GPU acceleration would be even more visible [1]. GPUs were designed in a highly parallel structure that allows large sets of data to be processed at the same time; similar computations are being made on data at the same time. This is the reason additional GPU acceleration makes calculation more efficient. CPUs are made to handle requests more linearly, even in parallel mode. For our problem, CPU parallelization also shows good results, but with a larger observed system, the benefit from parallel computation would be also more obvious (we could create more processes for the processor unit).

Here we have an example where improvement in the communication speed and bandwidth between CPU and GPU would be highly beneficial. If the test problem were more complex, the parts of the problem calculated on the GPU would also be more complex, and the speedup achieved

¹ www.nvidia.com/object/nvidia-kepler.html.

would compensate for insufficient CPU–GPU communication speed and bandwidth. In our problem, we have substantial CPU processes and GPU communication for a rather short calculation during the steps in between.

In addition, to improve the speed for this resource-demanding calculation, in further studies the focus should be on exploring better ways to distribute GPU resources on parallel processes while overcoming these components' poor communication speeds.

Acknowledgements

This work is supported by the Serbian Ministry of Education, Science, and Technological Development through Projects OI 171005. Special thanks to Nvidia for donating Tesla K20 to GPU Group for Simulations in Solid State Physics.

References

- [1] Giroto I, Varini N, Spiga F, Cavazzoni C, Ceresoli D, Martin-Samos L and Gorni T 2012 Enabling of Quantum-ESPRESSO to petascale scientific challenges *PRACE Whitepapers*. (www.prace-ri.eu/IMG/pdf/enabling_of_quantum_espresso_to_petascale_scientific_challenges.pdf)
- [2] Harju A, Siro T, Federici-Canova F, Hakala S and Rantalaiho T 2013 Computational physics on graphics processing units *Applied Parallel and Scientific Computing* vol 7782 (Berlin: Springer) pp 3–26
- [3] Giannozzi P *et al* 2009 *J. Phys.:Condens. Matter* **21** 395502
- [4] Spiga F and Giroto I 2012 phiGEMM: a CPU-GPU library for porting Quantum ESPRESSO on hybrid systems *Proc. 20th Euromicro Int. Conf. on Parallel, Distributed and Network Based Computing –Special Session on GPU Computing and Hybrid Computing* pp 368–75
- [5] Novoselov K S, Falko V I, Colombo L, Gellert P R, Schwab M G and Kim K 2012 *Nature* **490** 192
- [6] Profeta G, Calandra M and Mauri F 2012 *Nat. Phys.* **8** 131–4