

A Cloud Computing Service Architecture of a Parallel Algorithm Oriented to Scientific Computing with CUDA and Monte Carlo

Ji Yimu^{1,2}, Kuang Zizhuo¹, Pan Qiao Yu¹, Sun Yanpeng¹,
Kang Jiangbang¹, Huang Wei³

¹ College of Computer, Nanjing University of Posts and Telecommunication, Jiangsu, China

² Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, China

³ Institute of Materials, Nanjing University of Posts and Telecommunication, Jiangsu, China
E-mail: jim4njupt@gmail.com

Abstract. *The GPGPU (General Purpose Graphics Processing Units) have become a whole new area for research due to the fast development of GPU hardware and programming tools, such as CUDA (Compute Unified Device Architecture). Here we have made a research on CUDA and its applications in the field of scientific computation, as organ electronics. We propose one solution of the parallel computation in global optimization of the physical characteristics in organ electronics with Monte Carlo, and one cloud service architecture for parallel computation of organ electronics was designed. Finally, one case of computing the organ molecule moving orbit was implemented based on the above solution and architecture, and has got a good display by the cloud service.*

Keywords: *CUDA, GPU, Monte Carlo, cloud computing, scientific computing.*

1. Introduction

Compute Unified Device Architecture (CUDA) [19] is a parallel computing platform and a programming model invented by NVIDIA. It is a general purpose parallel computing platform and a programming model that leverages the parallel compute engine in NVIDIA GPUs to solve many complex computational problems in a more efficient way than CPU. It enables dramatic increases in the computing

performance by harnessing the power of the graphics processing unit. With the development of the graphics card, the computing ability of GPU has exceeded the counterpart of CPU. It is a waste of resource to be just a graphics card for GPU. So CUDA is introduced to satisfy the other needs for the graphics card besides the image computation. Currently CUDA can be only used on the *N* card of G80 platform. The core set of tools is a C language compiler.

Meanwhile, cloud computing is a new method of sharing infrastructure, the resources can be connected into a huge pool to provide various IT services. Roughly, cloud computing describes highly scalable computing resources provided as an external service via Internet on a pay-as-you-go basis. The cloud is simply a metaphor for Internet. Many scientific areas have taken use of cloud computing, however even so, they still show a huge demand for computing. So we can try to develop a new cloud computing model in GPU domains. On one hand, we can make use of the powerful processing capabilities provided by the GPU cluster, on the other hand, we can enjoy the convenience and low-cost brought by cloud computing.

2. The background of CUDA and Monte Carlo

2.1. The hardware model and the software model

CUDA is a new infrastructure, which can use GPU to solve the problems in commerce, industry and scientific complex computing. It is a complete plan of solving the GPGPU [3,16]. It provides direct access to the hardware interface instead of depending on API graphics interface to achieve the access of GPU as a traditional approach. CUDA adopts a totally new computing architecture to use the hardware resources provided by GPU in order to offer a stronger computing ability to the massive data computing applications.

The software model of CUDA can be divided into three parts: the software hierarchical model, the thread hierarchical model and the memory hierarchical model.

The software hierarchical model is combined with three parts, CUDA Library, CUDA Runtime and CUDA Driver. The CUDA Library is provided by CUDA technology. The CUDA Runtime provides application development interfaces and runtime components, including the definition of basic data types, all kinds of calculation, type conversion, memory management, device access and so on. The code developed on the basis of CUDA is divided into two types when executed. One is the host code running on CPU, the other is the device code running on GPU. Different types of codes can access different resources due to their different locations. CUDA Driver can be understood as the Device Abstract Layer of GPU (CUDA-ENABLE), providing an abstract access interface of hardware. Currently, the application developed on the basis of CUDA must have the hardware support of NIVIA CUDA-enable. But in the future, CUDA will develop into a global standard interface of GPGPU, which is compatible with GPU products by different manufactures.

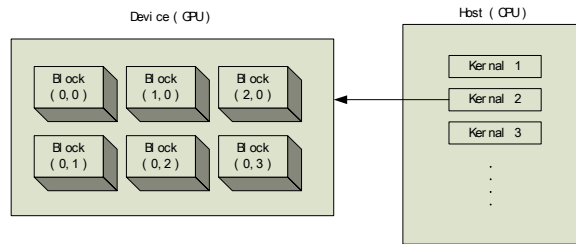


Fig. 1. The thread hierarchical model

Fig. 1 shows the thread hierarchical model. CUDA provides fine-grained data parallelism and thread parallelism, nested within coarse-grained data parallelism and task parallelism. They guide the programmer to partition the problem into coarse sub-problems that can be solved independently in parallel by blocks of threads, and each sub-problem into finer pieces that can be solved cooperatively in parallel by all threads within the block.

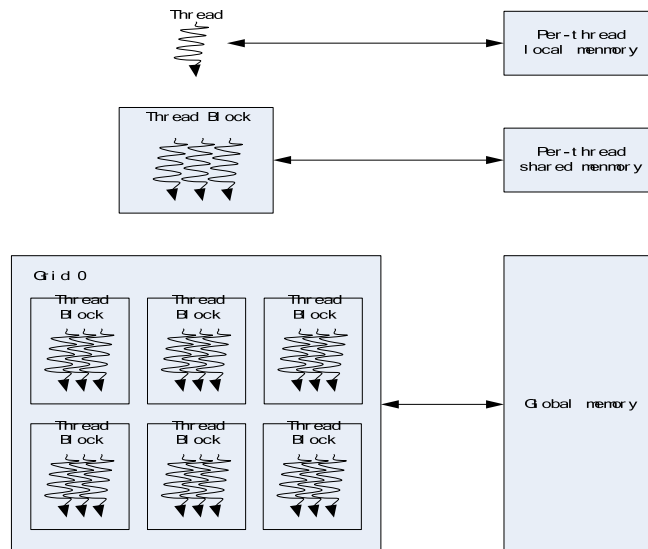


Fig. 2. The memory hierarchical model

Fig. 2 shows the memory hierarchical model. CUDA threads may access data from multiple memory spaces during their execution as illustrated in Fig. 3. Each thread has private local memory. Each thread block has shared memory visible to all threads of the block and with the same lifetime as the block. All threads have access to the same global memory.

There are also two additional read-only memory spaces accessible by all threads: constant and texture memory spaces. The global, constant, and texture memory spaces are optimized for different memory usages. Texture memory also offers different addressing modes, as well as data filtering, for some specific data formats.

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

2.2. Functions of CUDA

CUDA is an architecture of parallel computing, which can handle enormous calculation and sharply improve the system performance by making use of hundreds of GPU stream processors. So, CUDA is used in image and video editing, biology and chemistry calculation, simulation of fluid mechanics, the reforming image of CT, earthquake analysis, ray tracing and many other areas by more and more software developers. There has been different software starting to support CUDA technology currently and many of our commonly used applications has already applied CUDA to speed up. For example, CUDA can improve the display effect of PDF and shorten the open time of a file when there are many pictures in it; CUDA can also be used in media player to speed up hardware decoding when watching a film on PC.

By the promotion of the hardware manufacturers and software developers, there will be more and more applications supporting CUDA. The area CUDA reaches will be wider, and the computer use will be more efficient. The development of science and technology is really anticipated.

2.3. Monte Carlo method

Monte Carlo method is a non-quantum calculation method mainly used in huge systems. Monte Carlo method combines the random motion of the particle (atom or molecule) with the probability distribution principle of statistical mechanics to get statistical systems and thermodynamic data. Proper probability model will be built according to the changing regularity of the problem and then massive experiments will be done to assure that some statistical parameters of the model are exactly the answers of the problem. Energy transferring in organic solar cells is a random event, therefore, we can make computing simulation based on Monte Carlo method, analyze the computing results and then predict the results of the experiments. In this way we can further optimize the selection of the organic material, the design of the device structure and the efficiency of light-to-electricity conversion. The procedure of the traditional computing simulation algorithm based on Monte Carlo method is shown as Algorithm 1.

Algorithm 1. The simulation computing procedure of Monte Carlo method

- | |
|--|
| <p>Step 1. Input, iterations of Monte Carlo, the Monte Carlo probability of random events.</p> <p>Step 2. Determine the implementation steps through random function and the probability of random events, such as the energy migration direction.</p> <p>Step 3. Repeat Step 2 until the iterations end.</p> <p>Step 4. Output, statistical average, analyze and predict the results.</p> |
|--|

How to improve the parallel computing efficiency of computationally intensive tasks based on Monte Carlo method? Its iterations can be parallelized and its efficiency of the computing tasks can be improved based on CUDA. CUDA does not need API graphics, but it exploits programs in C language. Meanwhile, CUDA adopts the unified processing framework, which reduces the difficulty of programming and helps NVIDIA catch up from behind compared with AMD/ATI. The NVIDIA GPU imports a shared memory chip to improve the efficiency

compared with the AMD GPU. As above mentioned, the code developed on the basis of CUDA is divided into two types when executed. One is the host code running on CPU, the other is the device code running on GPU. Different types of a code can access different resources due to their different locations. Besides taking use of the parallel computing ability of CUDA, we will also develop and design the corresponding cloud service based on Monte Carlo method with distributed parallel computing.

3. Applications based on CUDA

CUDA is widely applied to the areas of data mining, telecommunication, finance data analysis, statistical analysis, biological medicine, military simulation, etc. The calculations of these areas belong to mass data intensive computing. There is every reason to believe GPU can build its new age in these areas with the drive of CUDA. The following texts will discuss the CUDA-based applications in some areas.

3.1. Computational chemistry oriented application

Computational chemistry is based on the principle of computational science to solve chemistry problems. The architecture of GPU has the basic requirements to enter the field of scientific computation. The first time that GPU is imaged to be used in numerical calculation can be traced back to 20 years ago [17]. Despite various efforts and practice hanging by a thread [1, 12], there was no breakthrough because of the lack of scientific computing programming tools. CUDA emerged as the times required [4]. A chemical package called BigDFT [10] should be mentioned here as it uses CUDA to do programming on GPU and improves the speed by 6 times than on a CPU. The algorithms proposed to speed up the calculation of quantum chemistry on GPU these years are all making use of the existing software packages to improve a part of themselves, except BigDFT. However, the Terachem program [23-24] produced by 2010, takes complete use of CUDA and overwrites the basic procedure of quantitative calculation. L i u is the first person to use CUDA to simplify MD calculation [18]. B a o, F e n g, Y u j i a n g u o [2] summarize the progress in algorithms and the corresponding software with regard to computational chemistry using GPU including quantum chemistry and molecular dynamics simulations in details. They introduce and explore the newly developed TeraChem program, which is unique quantum chemical software and they discuss the algorithms, implementations and functionality of the program. GPU also accelerated molecular dynamics simulation of thermal conductivities [25]. MD method is very time-consuming and with the help of GPU, molecular dynamics simulations have become a powerful tool for elucidating complex physical phenomena.

3.2. Other CUDA-based applications

Using computer science to analyze the data in biology can discover the potential biological knowledge and provide reliable prediction before the experiments. It can

also provide guidance for the design of the experiments. Y u [26] discussed the realization and the optimization of the Voting algorithm [8] and the Gibbs [5] algorithm on CUDA platform, which is used in DNA motif discovering [21]. As a result, with the increasing of the problem scale and computing tasks, the acceleration ratio is enlarging. The algorithms on CUDA have better accuracy and faster execution speed. The voting algorithm has little mathematical calculation and large expense of memory, so its optimization is difficult and finally gets a maximum of 4.7 times acceleration by improving the address computing. The Gibbs algorithm is mainly iterated and finally gets a maximum of 6.6 times acceleration by running multi sampling process simultaneously in a thread block.

The first science area that CUDA approaches, is the astrophysics. The developers make use of GPU to simulate millions of particles. The data set of a geographic information system has become more and more complicated and related to a lot of gigabyte interactive maps. Under the pressure of this, they introduced the product which can fastly and efficiently deal with the data set. With the help of CUDA, the precious calculation which needs 20 minutes can be done within 30 seconds [13].

CUDA is a convenient use of GPU for general computing. L e i, L i u [11] have researched several vector dot product algorithms based on CUDA. They also made a comparison and analysis of their performances and causes. The experiment shows that the best algorithm is about 7 times faster than the CPU algorithm.

4. Cloud computing and scientific computing

Cloud computing [6] is a service model. It is the newest trend in IT industry, which is widely concerned by the industry and academic institutions [7, 9]. The cloud may provide an alternative to traditional computational tools in the high performance scientific computing area. The availability of these large and virtualized pools of computing resources raises the possibility of a new computing paradigm for scientific research with many advantages: For research groups, cloud computing provides convenient access to reliable, high performance clusters and storage without the need to purchase and maintain sophisticated hardware. For developers, virtualization allows the scientific codes to be optimized and pre-installed into machine images, facilitating the control over the computational environment. The following will introduce the cloud service system and the current status of the scientific computing cloud.

4.1. Cloud service layered system

In cloud computing, according to the type of the services provided by its service set, the entire cloud computing service set is divided into four layers: the application layer, platform layer, infrastructure layer and the virtualization layer. Each of these four layers is corresponding to a sub-set of service. Fig. 3 shows the cloud computing service levels.

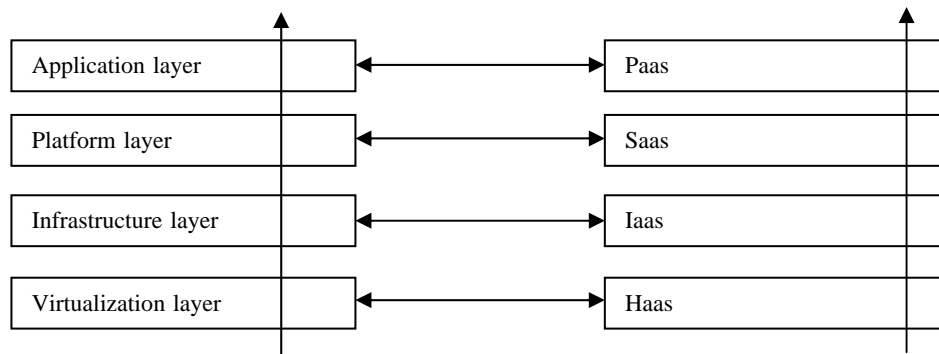


Fig. 3. Cloud computing service level

Cloud computing service level is different from our familiar computer network level. In a computer network, each layer has its certain functions and there is a certain relationship among the layers. However, the level of cloud computing architecture is split, which means each layer can complete the user's requirements independently and does not need the necessary support and service of other layers.

In the cloud computing service architecture, each layer has its corresponding cloud products. The application layer is corresponding to Saas, such as Google APPS, SoftWare Services; The platform layer is corresponding to Paas, such as IBM IT Factory, Google APPEngine, Force.com. The infrastructure layer is corresponding to Iaas, such as Amazo EC2, IBM Blue Cloud, Sun Grid. The virtualization layer is corresponding to Haas, combining Paas to provide hardware service including a server cluster, hardware detection and so on.

4.2. The current status of scientific computing cloud

With deepening of the scientific research, the environment of scientific computing needs to keep developing to satisfy the various needs of applications. The scientific cloud computing cannot simply use a commercial cloud computing model. It should take the scientific computing needs and characters as an objective. With expansion of the scientific research, the people need to gather massive data to solve complex scientific problems involving physics, chemistry, bio-science, medical science, economics, meteorology, earthquake, gene secrets and so on, which present a greater challenge to data processing in computing technology.

Cloud computing technology has become a hotspot in the development of computer science in recent years. Many companies spend a lot of manpower and material resources to intervene in this area. The application of cloud computing has almost covered the whole Internet, providing new chances and challenges for the development of Internet. Scientists and engineers rent the resources in the large data center [20] to solve the needs of high-performance scientific computing. However, the characteristics of cloud computing provided are not consistent with the requirements of scientific computing. Actually, the most concerned problem is the performance problem.

Performance is undoubtedly the most important problem in scientific computing, many scientists and engineers spare no efforts to improve the

computing ability to meet the increasing requirements of calculation, such as the weather report, the nuclear physics and so on. The developers are searching new computing methods to improve the computing ability. Currently, the main cloud computing model applied to scientific computing is Iaas, such as EC2 cloud, provided by Amzon [14, 15]. It provides the users with applications on the basis of virtualization. However, Z h a n g [27] tested the performance of cloud computing platform through EC2. He constructed a private cloud platform and presented the application methods to use scientific computing software under the environment of cloud computing, called PET/SPECT. The tests and results showed that cloud computing did not bring substantial breakthroughs in performance. That is because on one hand, the cloud computing nowadays is for business operation, on the other hand, it is enslaved to the platform. Admittedly, we can see the cloud computing bringing a new application model, making the scientific computing more convenient. Perhaps in the near future, everybody, not only the scientists and engineers, but the common people can gain larger computing performance from the high-performance cloud, comparing with a supercomputer. Meanwhile, the technology of cloud computing becomes increasingly mature and with the constant development of the corresponding platform software in resource utilization and load balancing, the scientific computing will meet new chances.

Cloud computing shows great advantages in improving resource utilization efficiency, flexible scalability and management ability. Many scientific areas have begun to test and apply this technology and it is believed that cloud computing will make a lot of contributions to scientific computing.

4.3. The cloud service framework of parallel computing based on Monte Carlo

The research on physical modeling and computing simulation of an electron and exciton in an organic solar cell belongs to the computing simulation in material science. The research plan is shown in Fig. 4. Firstly, the mathematical models will be built on exciton transfer, exciton dissociation and charge transfer in the procedure of light-to-electricity conversion, which also realizes multi-objective optimization. The model will then be simulated after optimization, using the Monte Carlo method. At last, we will deliver the procedure of light-to-electricity conversion in an organic solar to the cloud platform for remote clients to access or transfer through Iaas (Eucalyptus), Paas (Hadoop) and Saas (Java+Web Service).

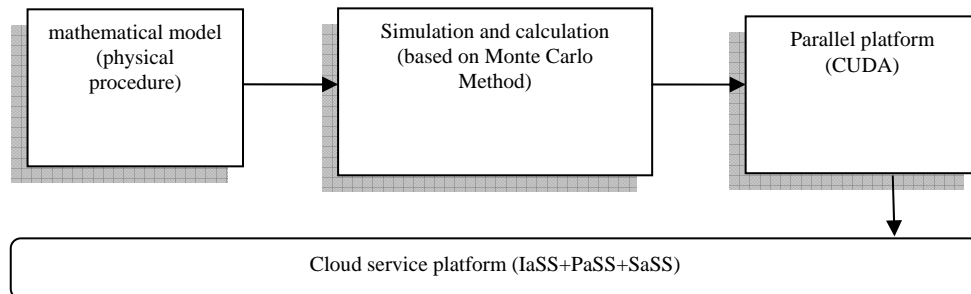


Fig. 4. The computing simulation in light-to-electricity on a cloud service platform

The Monte Carlo method is one of the commonly used methods for solving the organic solar cells' exciton transfer and charge transfer model. It provides possibilities for parallel computation, as Monte Carlo method is computation-intensive to solve the above problems and the computing tasks are independent for multiple exciton or charge.

CUDA, as a parallel computing system, improves the concurrency of computing tasks. CUDA supports a large-scale thread level parallel, and dynamically creates, calls and executes these threads. These operations are heavyweight in CPU, but lightweight in CUDA. We can ignore the cost in creating and calling the threads. The CUDA programming model considers CPU as a host, controlling the overall serial logic and task scheduling of the program, while it takes the GPU as the coprocessor or the device, running parts of parallel data which can be highly threaded. The technological realization framework of Monte Carlo parallel computing based on CUDA is shown in Fig. 5.

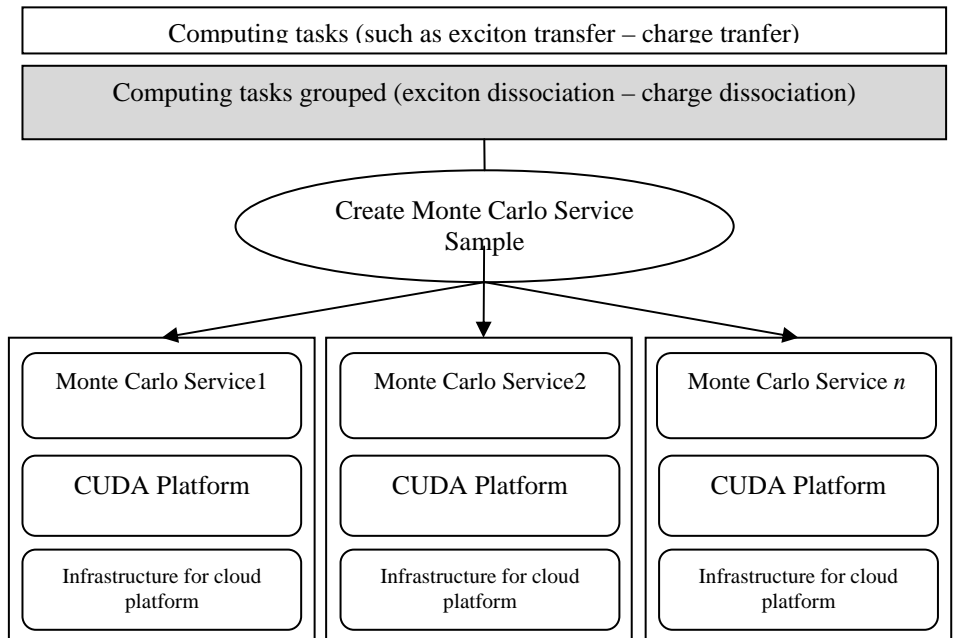


Fig. 5. The Monte Carlo parallel cloud service framework based on CUDA

5. Scientific computing oriented cloud computing service architecture based on CUDA

5.1. Scientific computing oriented cloud computing service architecture based on CUDA

If we combine cloud computing and CUDA into a new service model, there will be many traits we can take advantage of. On one hand, we can make use of the powerful processing capabilities provided by GPU cluster, on the other hand, we can enjoy the convenience and low-cost brought by cloud computing.

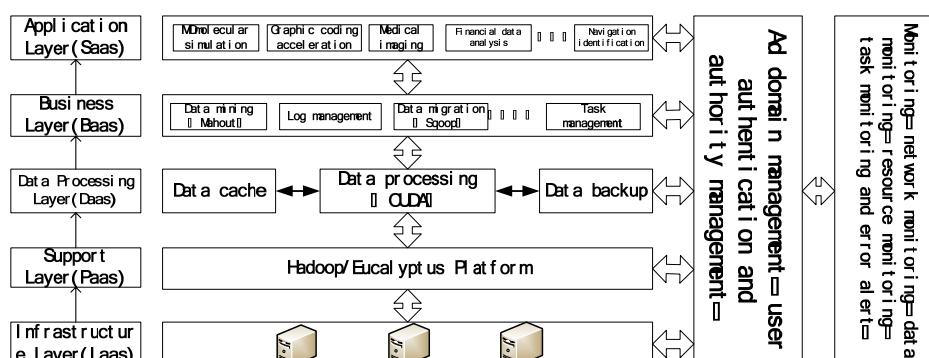


Fig. 6. Scientific computing oriented cloud computing service architecture based on CUDA

Fig. 6 is the scientific computing oriented cloud service infrastructure based on CUDA. The infrastructure is combined in five layers (IaaS, Paas, Daas, Baas, Paas). The infrastructure layer is responsible for the normal operation, maintenance and storage of the physical machines. The support layer provides the platform, such as Eucalyptus or Hadoop, which must be constructed on the machines of IaaS. The data processing layer mainly does mass data operation, which includes data cache and data backup. The business layer mainly takes the responsibility of data migration and data mining. It also manages the tasks and diaries. The application layer mainly provides the interfaces of various kinds of services to the customers, such as navigation recognition, data mining, telecommunication, finance data analysis, statistical analysis, biological medicine, military simulation, etc. The AD domain management takes control of the whole local network. The monitoring function includes network monitoring, data monitoring, resource monitoring, mission monitoring and fault debugging.

The advent of CUDA provides an efficient and low-cost acceleration method, making the computing performance to be well accelerated on a common computer. The CUDA equipment can also be applied to build large-scale computing servers, providing higher performance and lower waste of power than the ordinary cluster. In recent years, CUDA is driving the vigorous development of GPU. More scientific areas will apply CUDA in the research. At the same time, NIVIA will produce GPUs with larger capacity and faster speed. The price of GPU will be lower and CUDA will be expanded by new functions. The ATI Stream Technology produced by AMD will be a strong competitor to CUDA. Application of GPU in scientific computing will show blooms.

5.2. The physical process and the model of organic solar cells and the model

Through analyzing the physical process of organic solar cells, the mathematical models should be prior built on each procedure (exciton transfer, exciton dissociation and charge transfer) in order to build an integrated model. Each procedure has its corresponding theoretical basis. We take each procedure as the target function. The input data, condition setting and target optimization included in the procedure are exhibited in details in Fig. 7.

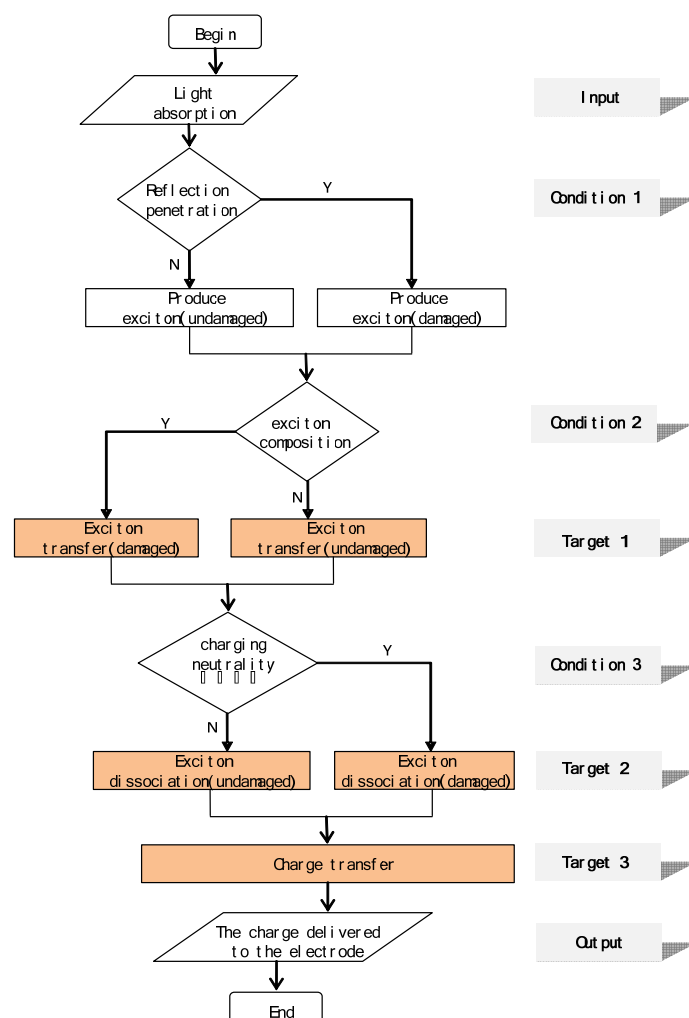


Fig. 7. The light-to-electricity procedure and modeling of polymer organic solar cell

The research on physical procedure modelling of exciton transfer, exciton dissociation and charge transfer in organic solar cell and the optimization of its computing on the basis of improving the energy conversion efficiency is the hotspot in recent years. This article combines current theoretical basis (such as band theory, MO theory, Schrödinger equation, Förster Theory, DFT, Coulombic principle, Marcus theory and the first principle) with computing methods (Monte Carlo method) to put forward a model which could optimize the procedure of light-to-electricity conversion. It also proposes the corresponding parallel computing methods for each specific model during each step, which can improve the computational efficiency and accuracy. In order to achieve this goal, we have done some useful research which can be shown in three aspects: (1) One of the three key links in the physical procedure of an organic solar cell – the integrated multi-objective optimization model can be realized. (2) The computing efficiency of light-to-electricity conversion in an organic solar cell will be improved. (3) We can build

our own distributed parallel computing platform for the physical procedure of an organic solar cell, making traditional operations transfer from a command mode to Web mode. The access mode is improving from the traditional in-lab mode to the Internet mode, making the traditional single computing mode shift to the parallel computing mode (Fig. 8).

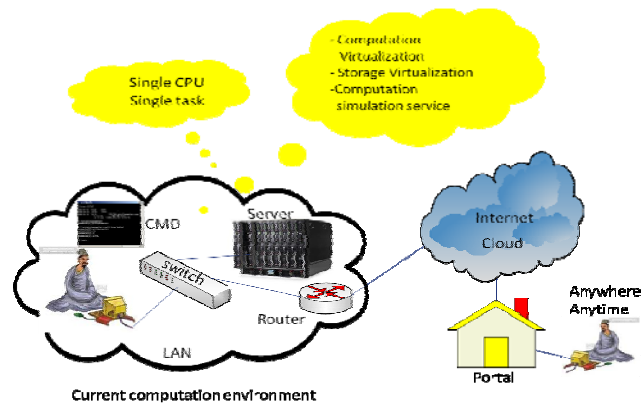


Fig. 8. The shift from the traditional command mode to Web mode

We have initially completed the exciton transfer model and the computing simulation based on Monte Carlo. The computing simulations of a homogeneous migration model and inhomogeneous migration model are shown in Fig. 9.

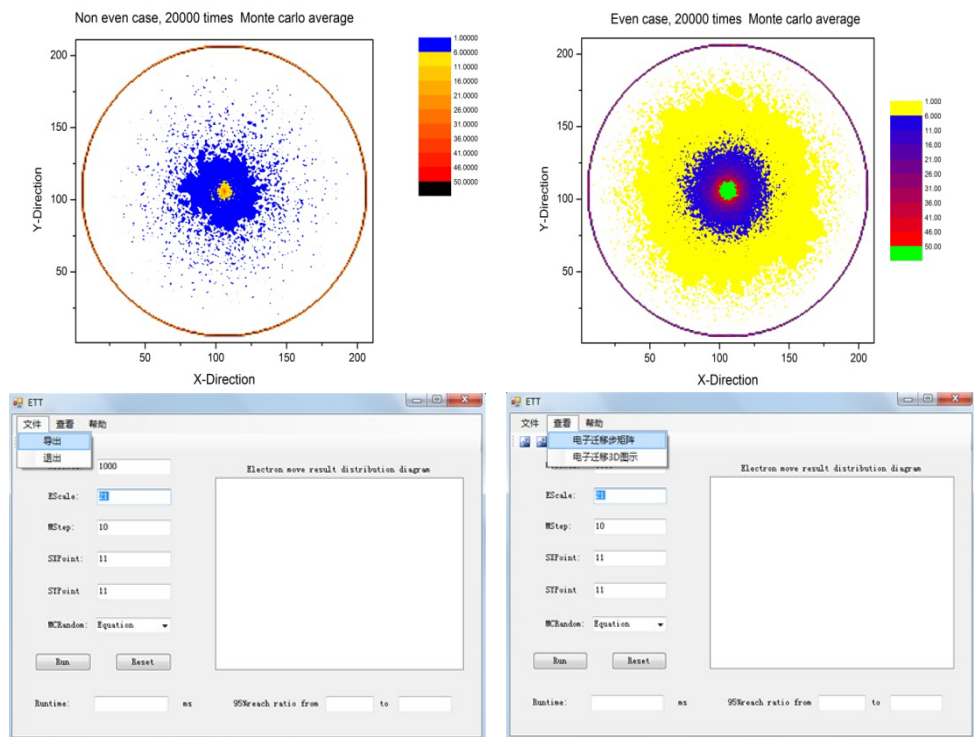


Fig. 9. The computing results of exciton migration based on Monte Carlo and software interface

6. Summary

GPU has always been applied to graphic processing, especially 3D image rendering. GPU is based on a parallel architecture so that the computing ability is greater than CPU. The article first introduced CUDA in three aspects: the hardware design, the software model and the programming language interface. CUDA is used in many scientific areas to do a large amount of calculations, such as data mining, telecommunication, finance data analysis, statistical analysis, biological medicine, military simulation, and so on. Cloud computing has become a hot spot in the development of computer science and the applications of cloud computing have almost covered all the applications in Internet. It provides more chances and meanwhile, more challenges, such as cloud security [22]. The application of scientific computing in cloud computing is a new direction produced by the popularity of cloud computing. If we change the model of traditional high-performance scientific computing and make it in accordance with the model of cloud computing, then we can make full use of high-performance computing service and enjoy all kinds of advantages brought by cloud computing, such as the lower cost. At last, the architecture of the cloud computing services based on CUDA is introduced. However, the CUDA-based applications still have many gaps to fill in and much value to research.

Predictably, the scientists, the engineers and even the ordinary people can easily access the cloud computing resources with high performance and it would be an important direction.

Acknowledgments: This work was partly supported by Projects of JiangSu Industry Supporting (BE2011189, BE2010057), JiangSu Natural Science of Young Foundation (SJ213030) JiangSu High School Natural Science Research (11KJB520013), and National Post-doctoral Foundation (SBH13011).

References

1. B o h n, C. A. Joint Conference on Intelligent Systems 1999. (JCIS'98), 1998, 2, 64.
2. B a o, J., X. F e n g, Y u j i a n g u o. GPU Triggered Revolution in Computational Chemistry [J]. – Acta Phys.-Chim. Sin., Vol. **27**, 2011, No 9, 2019-2026.
3. C h e n, G., P. L i et. al. High Performance Computing Via a GPU. – In: Information Science and Engineering (ICISE), 1st International Conference, 2009, 238-241.
4. CUDA: Santa Clara, CA (accessed April 13, 2011).
http://www.nvidia.com/object/cuda_home_new.html
5. L a w r e n c e, C. E., S. E. A l t s c h u l, M. S. B o g u s k i et al. Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment. – Science, Vol. **262**, 1993, 208-214.
6. C h e n, K., W. M. Z h e n g. Cloud Computing: System Instances and Current Research. – Journal of Software, Vol. **20**, 2009, No 5, 1337-1348 (in Chinese with English Abstract).
[doi: 10.3724/SP.J.1001.2009.03493]
<http://www.jos.org.cn/1000-9825/3493.htm>
7. D a s h, D., V. K a n t e r e, A. A i l a m a k i. An Economic Model for Self-Tuned Cloud Caching. – In: Y. E. Ioannidis, D. L. Lee, R. T. Ng, Eds. Proc. of 25th Int'l Conf. on Data Engineering (ICDE'2009), New York, IEEE Computer Society Press, 2009, 1687-1693.
[doi: 10.1109/ICDE.2009.143]

8. Chin, F. Y. L., H. C. M. Leung. Voting Algorithms for Discovering Long Motifs. – In: Proc. of Third Asia-Pacific Bioinformatics Conf. (APBC'05), 2005, 261-271.
9. Feng, D. G., M. Zhang, Y. Zhang, Z. Xu. Study on Cloud Computing Security. – Journal of Software, Vol. **22**, 2011, No 1, 71-83 (in Chinese with English Abstract). [doi: 10.3724/SP.J.1001.2011.03958]
<http://www.jos.org.cn/1000-9825/3958.htm>
10. Genovese, L., M. Ospici, T. Deutsch, J. F. Mehaut, A. Neelov, S. J. Goedecker. – Chem. Phys., 2009, No 131, 34103.
11. Lei, G., J. Liu. Performance Analysis of Vector Dot Product on GPU. – Computer Engineering and Applications, Vol. **48**, 2012, No 2, 201-202.
12. Hoff, K. E., I. I. Culver, T. Keyser, J. Ming, L. Manocha. Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware. – In: Proceeding of SIGGRAPH'99, Danvers, 8-13 August 1999, Assison-Wssley Publishing Company, 1999, 277-286.
13. <http://tech.sina.com.cn/h/2008-08-18/1539771950.shtml>
14. <http://aws.amazon.com/eeZ>
15. <http://aws.amazon.com/s3/>
16. Anderson, J. A., C. D. Lorenz, A. Traveset. General Purpose Molecular Dynamics Simulation Fully Implemented on Graphics Proeessing Units. – J. Chem. P5, Vol. **227**, 2008, No 10, 5342-5359.
17. Reichert, L. J., M. Donald, B. R. Greenberg. Comput Graph. 1990, No 24, 327.
18. Liu, W., B. Schmidt, G. Voss, W. Müller-Wittig. – In: High Performance Computing-HiPC 2007, Lecture.
19. NVIDIA CUDA. Compute Unified Device Architecture Programming Guide Version 3.0. (accessed March 6, 2010).
http://www.nvidia.com/object/cuda_develop.html
20. Rehr, J. J., J. P. Gardner, M. Prange, L. Svec, F. Vila. Scientific Computing in the Cloud. CoRRabs/0901.0029.
21. Sinha, S., M. Tompa. A Statistical Method for Finding Transcription Factor Binding Site. – In: Proceedings of the Eighth International Conference on Intelligent Systems on Molecular Biology, San Diego, CA, 2000, 344-354.
22. Securing Cloud Computing is Industry Responsibility. – Infosecurity, Vol. **7**, March-April 2010, Issue 2, p. 11.
23. Chem, T. Notes in Computer Science. – S. Aluru, M. Parashar, R. Badrinath, V. K. Prasanna, Eds. Berlin, Heidelberg, Springer, 2007, Vol. **4873**, p. 185 (accessed March 6, 2010).
<http://www.petachem.com>
24. Ufimtsev, I. S., T. J. J. Martinez. Chem. Theory Comput., 2009, Vol. **5**, p. 2619.
25. Xu, D., M. J. Williamson, R. C. Walker. Annual Reports in Computational Chemistry. – Elsevier, Amsterdam, Vol. **6**, 2010, p. 1.
26. Yu, L. The Research of CUDA-Based High-Performance Computing and the Application in Bioinformatics [D], 2009, p. 501.
27. Zhang, S. Research of Application of Cloud Computing in Cloud [D]. 2010, p. 5.