USING OF NEW POSSIBILITIES OF FERMI ARCHITECTURE BY DEVELOPMENT OF GPGPU PROGRAMS

V.A. Dudnik,[*] *V.I. Kudryavtsev, S.A. Us, M.V. Shestakov*

National Science Center "Kharkov Institute of Physics and Technology", 61108, Kharkov, Ukraine

(Received December 11, 2012)

Description of additional functions of hardware and software, which are presented in the structure of new architecture of FERMI graphic processors made by company NVIDIA, was given. Recommendations of their use within the realization of algorithms of scientific and technical calculations by means of the graphic processors were given. Application of the new possibilities of FERMI architecture and CUDA technologies (Compute Unified Device Architecture - unified hardware-software decision for parallel calculations on GPU) of NVIDIA Company was described. It was done for time reduction of applications' development which is using possibilities of GPGPU for acceleration of data processing.

PACS: 89.80.+h, 89.70.+c, 01.10.Hx

1. INTRODUCTION

CUDA architecture, which has appeared several years ago, provided an opportunity to use for programming of calculation tasks for GPU the same program tools as for usual CPU: C-language, Fortran, Open CL or similar. Use of GPU computing powers for the scientific and technical calculations has ceased to be a complete exotic, but still some difficulties both in realization of effective computing algorithms on GPU and in their using are remained. Major lack for GPU use as fast calculators was an absence of calculations' support with double accuracy and finding and correction of memory's errors mechanisms.

However the greatest difficulties have arisen during the debugging of programs using GPU. CUDA architecture has given conventional debugging tools for MS Visual Studio: the task of points of interruption, viewing of content of memory areas, status of parallel flows, etc. by means of familiar Locals windows, Watch, Memory and Breakpoints, but it is only in a mode of emulation of GPU by the means of CPU. Search of synchronization's errors, analysis of various exclusive situations (division by zero, overflow, etc.) was considerably complicated also by the fact that parallel sites of GPU program in the emulator were carried out consistently, besides operations of calculation with a floating point on GPU and on CPU were carried out a little differently.

In 2009 NVIDIA developers have presented the further development of CUDA platform - FERMI architecture. Fermi architecture initially implies using of graphic processors not only for processing of computer graphics. NVIDIA positioned the new architecture mainly on the market of high performance computing that assumed both the high speed of settlement operations and the high reliability with the high convenience of programming.

2. NEW HARDWARE FUNCTIONS OF FERMI ARCHITECTURE

Supporting of calculations of double precision floating point realized in the new architecture was one of key requirement of the market of high-efficiency calculations. It is necessary to notice that the graphic processor of previous generation GT200 also could be used for the calculations of double precision floating point but its productivity need to be better for such operations.

Besides in FERMI architecture has been realized the mechanism of errors' finding and correction in operative memory and subsystems of a cache-memory (ECC Error Correcting Code is a special technology of operative memory which allows to find out and correct incorrect value of one bit on every 8 bits of transferred data). It has allowed achieving comparable with CPU fault tolerance and reliability of the work of computing algorithms. Usual graphic processors did not require these functions and were satisfied with the calculations of single precision floating point.

An essential obstacle for the realization on GPU complex computing algorithms was the structure of GP200 memory which did not allow organizing the effective internal manager of memory and using of data allocation as required in the process of calculations. As a rule it was necessary to reserve all GPU

[*] Corresponding author. E-mail address: vladimir-1953@mail.ru

memory within start of GPGPU-a thread of the program, and to release only after its end. Emergence in FERMI architecture of the common L2-cache has considerably accelerated (in ten times) atomic operations with the memory. It gave the opportunity to organize fast enough internal manager of memory allocating memory of each thread during work. It has allowed realizing functions of management of memory **malloc** and **free** of programming C language that has essentially accelerated performance of the algorithms using dynamic data structures.

Access to a global memory also has been improved in Fermi. The global video memory has been divided into six banks in G200. It was necessary to seek compliance with the various banks for achieving of the maximal speed and paralleling of performance of reading inquiries. This restriction is removed in FERMI and the access to the local memory of a multiprocessor is accelerated. The local memory of the multiprocessor is divided into the banks. The common address space for all memory of CPU and GPU realized in FERMI architecture allows uniting in one address space all visible memory for the thread (memory of the multiprocessors and visible global memory (Fig. 1)).

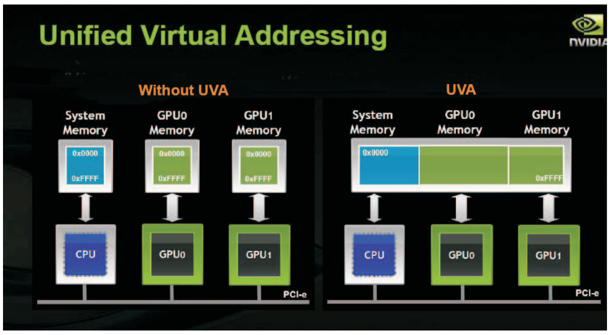

Fig. 1. *Unified virtual addressing.*

Management of the access to the global memory is improved. Cache inquiries about the memory are united for warp (32 strings), go on 128 bites with caching in L1 within sequential data access in the global memory and included L1. Thus data for the following warp will be already in L1 with a high probability that allows to minimize a quantity of manipulations to the global memory (Fig. 2).

2 transactions by 128B

Fig. 2. *Management of the access to the global memory (caching is off)*

The cache inquiries to the global memory always go on 32 bites at switched off L1 that it is more preferable to unloaded access to the memory (for example, at realization of table functions)(Fig. 3).

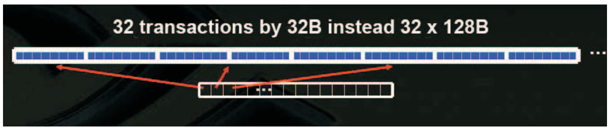

Fig. 3. *Management of the access to the global memory (caching is on)*

The opportunity of recursive functions' using, appeared in FERMI architecture, has allowed to carry more complex calculations on GPU and to reduce the quantity of rather slow operations of exchange with the operative memory on trunk PCI Express.

Two interfaces for copying allow accelerating practically twice the data exchange due to simultaneous performance of data copying from the memory of the multiprocessors CPU in GPU and from GPU into the CPU memory. The simultaneous optimized performance of several kernels realized in FERMI architecture allows to organize the simultaneous performance some of CUDA-functions of one application if one CUDA-function cannot completely load the computing capacities of the GPU-device. It allows loading more optimally GPU (Fig. 4), for example, it allows compensating long time of data transmission on trunk PCI due to overlapping operations of data loading of one task and calculations performance for another.

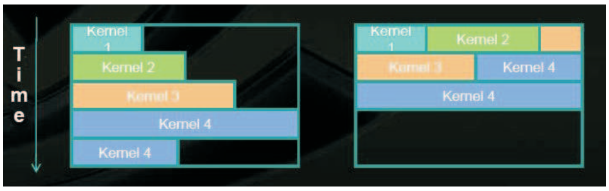

Fig. 4. *Several kernels realized in FERMI architecture.*

3. POSSIBILITIES OF FERMI ARCHITECTURE FOR DEVELOPMENT AND DEBUGGING OF APPLICATIONS

Hardware possibilities of FERMI architecture have allowed to NVIDIA experts to develop the new program decision for perfection of the processes of creation and debugging of applications CUDA – the NVIDIA NEXUS.

Supporting of C language ++ became the important innovation in NVIDIA NEXUS. Earlier a code for CUDA can be written only on C. Besides, the interaction between mechanisms of graphics processing and means for the performance of calculations of a general purpose is improved. Now libraries Direct3D 9, 10 and 11, and also library OpenGL can be used for data processing through the CUDA mechanism.

The opportunity of use for the applications' development of OpenCL means (the open standard of calculations with use of the graphic processors) is added. Perhaps, the most essential innovations realized in

NVIDIA NEXUS are the possibilities of high-grade debugging of applications on GPU's hardware. Earlier it was necessary to use for debugging the GPU program emulator. Using NVIDIA NEXUS for Microsoft Visual Studio will allow avoiding the majority of existed problems of debugging and due to this to increase the speed of applications' development. We consider the possibilities of debugging appeared in NEXUS in more details.

Nexus Debugger supports debugging of the code on CUDA C and HLSL directly on GPU equipment in working space Visual Studio 2008 and includes the following functions:

- **The Information page on CUDA** - gives the full information about the status of CUDA starts in the user application. Users can filter and receive the detailed information about exclusive situations, points of interruption, facts added into the database, errors of MMU. It is easy to be switched for debugging a problem.
- **CUDA Warp Watch**provides more effective way of navigation on resident streams and visualization of a status of the streams at the place of deformation.
- **Supports graphics and GPU computing.**Simple debugging of shaders or programs of scientific and technical calculations directly on GPU.
- **Parallel-aware**– debugging of applications using thousands of the data processing streams or graphic primitives.
- **Source breakpoints** – points of interruption at any place (with using of a hardware estimation of conditions).
- **Memory inspection**– the direct control and display of GPU memory using Visual Studio Memory Window.
- **Data breakpoints** – breakpoint on record at any place of memory.
- **Memory Checker**– breakpoint out of limits of the allocated memory.
- **Trace** – recording of actions and events executed on CPU and GPU on chosen correlated line. Includes:
  CUDA C, DX10, Open GL and Cg API calls;
  GPU - Host memory transfers;
  GPU workload executions;
  CPU core, thread and process events;
  Custom user events - Mark custom events or time ranges using a C API.

Nexus Analyzer– Nexus Analyzer supports tracking and GPU profiling of applications, collecting and the analysis of the information of the performance level of the kernel including hardware counters of productivity. Now traced loadings can consider dependences between processes and stacks of calls that allow to developers to analyze completely GPU loading, working of corresponding API calls and basic code of interested GPU process. Thus, the means of profiling CUDA FERMI allow understanding developer's problems of the productivity on the basis of the analysis of following factors:

- a deviation of flows or branching of the code;
- statistics of manipulations to the memory;
- statistics of the reasons of delays of the performance of the code;
- achieved levels of the performance of data processing in FLOPS.

4. SUMMARY

Use of Fermi architecture allows increasing the productivity of the computing applications using GPU for the acceleration of scientific and technical calculations.

Appeared in structure of software CUDA – NEXUS means of the debugging (allowing to debug the application directly on GPU equipment) sharply reduce time of debugging and reduce probability of occurrence of "floating errors" in already debugged program.

Means of Nexus Analyzer profiling give to the developers of applications the information allowing within the development and debugging of applications to achieve maximal use of computing functions of GPU.

References

1. A. Zubinsky. NVIDIA CUDA: unification schedules and calculations. May, 8-th, 2007 (http://itc.ua/node/27969).
2. D. Luebke. Graphics CPU-not only for graphics, (http://www.osp.ru/os/2007/02/4106864/).
3. D. Luebke, G. Humphreys. How GPUs Work// *IEEE Computer, February 2007.* IEEE Computer Society.
4. V. Dudnik, V. Kudryavtsev, T. Sereda, S. Us, M. Shestakov. Using of opportunities of graphic processors for acceleration of scientific and technical calculations. //*PAST*, 2009, N3, p. 120-123.
5. V. Dudnik, V. Kudryavtsev, T. Sereda, S. Us, M. Shestakov. Application of opportunities of tool system "CUDA" for programming graphic processors in tasks of scientific and technical calculations.//*PAST*, 2009, N5, p. 159-165.
6. V. Dudnik, V. Kudryavtsev, T. Sereda, S. Us, M. Shestakov. Sooftware development tools using GPGPU potentialities.//*PAST*, 2011, N3, p. 99-103.

7. V. Dudnik, V. Kudryavtsev, T. Sereda, S. Us, M. Shestakov. Use of a GPGPU means for the development of search programs of deffects of monochrome half-tone pictures //*PAST*, 2013, N3, p.282.

ИСПОЛЬЗОВАНИЕ НОВЫХ ВОЗМОЖНОСТЕЙ АРХИТЕКТУРЫ FERMI ПРИ РАЗРАБОТКЕ GPGPU ПРОГРАММ

В.А. Дудник, В.И. Кудрявцев, С.А. Ус, М.В. Шестаков

Приведено описание дополнительных возможностей аппаратных и программных средств, представленных в составе новой архитектуры графических процессоров FERMI компании NVIDIA. Даны рекомендации их использования при реализации алгоритмов научно-технических расчётов средствами графических процессоров. Описано применение новых возможностей архитектуры FERMI и технологии CUDA компании NVIDIA (Compute Unified Device Architecture – унифицированного программно-аппаратного решения для параллельных вычислений на GPU) для сокращения времени разработки приложений, использующих возможности GPGPU для ускорения обработки данных.

ВИКОРИСТАННЯ НОВИХ МОЖЛИВОСТЕЙ АРХІТЕКТУРИ FERMI ПРИ РОЗРОБЦІ GPGPU ПРОГРАМ

В.О. Дуднік, В.І. Кудрявцев, С.О. Ус, М.В. Шестаков

Приведено опис додаткових можливостей апаратних і програмних засобів, що представлені у складі нової архітектури графічних процесорів FERMI компанії NVIDIA. Дани рекомендації їх використання при реалізації алгоритмів науково-технічних розрахунків засобами графічних процесорів. Описано застосування нових можливостей архітектури FERMI і технології CUDA компанії NVIDIA (Compute Unified Device Architecture – уніфікованого програмно-апаратного рішення для паралельних обчислень на GPU) для скорочення часу розробки додатків, що використовують можливості GPGPU для прискорення обробки даних.