УДК 519.254

OPTIMAL PARALLELING FOR SOLVING COMBINATORIAL MODELLING PROBLEMS USING GRAPHICS PROCESSING UNITS

S.M. Yefimenko

International Research and Training Centre of Information Technologies and Systems of the NAS and MES of Ukraine

syefim@ukr.net

У роботі розглядається можливість підвищення ефективності комбінаторного алгоритму МГУА за допомогою розпаралелювання обчислень на графічному процесорі. Запропоновано схему алгоритму з послідовним ускладненням структур, яка забезпечує рівномірне навантаження на всі мультипроцесори графічної карти.
Ключові слова: індуктивне моделювання, комбінаторний алгоритм МГУА, паралельні обчислення, графічний процесор

The paper considers possibility of combinatorial GMDH algorithm computational speedup by means of programs paralleling for computing on graphics processing units. The scheme of algorithm with successive complication of structures providing the uniform loading on all multiprocessors of graphic card is proposed.
Keywords: i*nductive modelling, combinatorial GMDH algorithm, parallel computing, GPU*

В работе рассматривается возможность повышения эффективности комбинаторного алгоритма МГУА с помощью распараллеливания вычислений на графическом процессоре. Предложена схема алгоритма с последовательным усложнением структур, обеспечивающая равномерную нагрузку на все мультипроцессоры графической карты.
Ключевые слова: индуктивное моделирование, комбинаторный алгоритм МГУА, параллельные вычисления, графический процессор

1 Introduction

A problem of modeling by experimental data consists in dependence construction between input and output variables of an object or process being modeling. In case of combinatorial algorithms use the stage of parameters estimation requires high-powered computational resources, because of exponential dependence of computational complexity on arguments amount. That is why available computer resources have to be applied completely. In this case it is advisable to use:

- the high-speed methods of parameters estimation based on the recurrent algorithms for solving of linear equations systems [1];
- paralleling of computing using multiprocessing cluster systems [2].

In the paper we will consider the possibility of using GPU-based parallel computing for solving combinatorial modelling problems.

2 GPU-based parallel computing

Computations on graphics processing units (GPU) are enjoying wide popularity lately. It is related to that computational capabilities of GPU have been racing up. State-of-the-art graphics processors possess the enormous productivity that substantially exceeds productivity of central processing units (CPU). Their availability to be used for solving computational problems is being more and more catching in this connection. For an example comparisons between computer systems performance using Intel processors and video cards of ATI [3] are showed on figures 1 and 2.

The reason for better performance of systems on the bases of GPU is that GPU is specialized for compute-intensive, highly parallel computation. Combinatorial algorithms are known to be well-parallelized and therefore can be effectively executed on GPU.

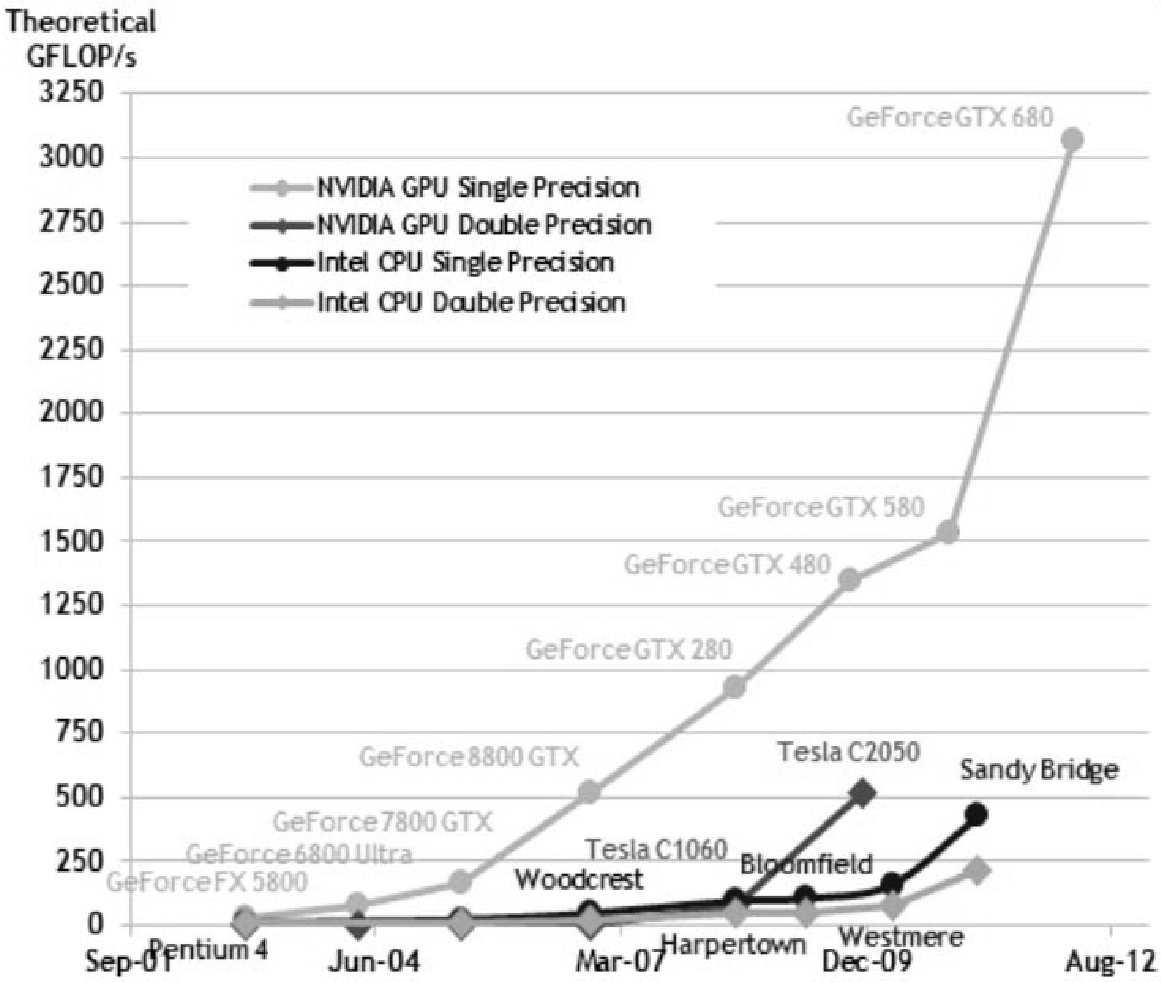

Fig.1. Floating-point operations per second for the CPU and GPU

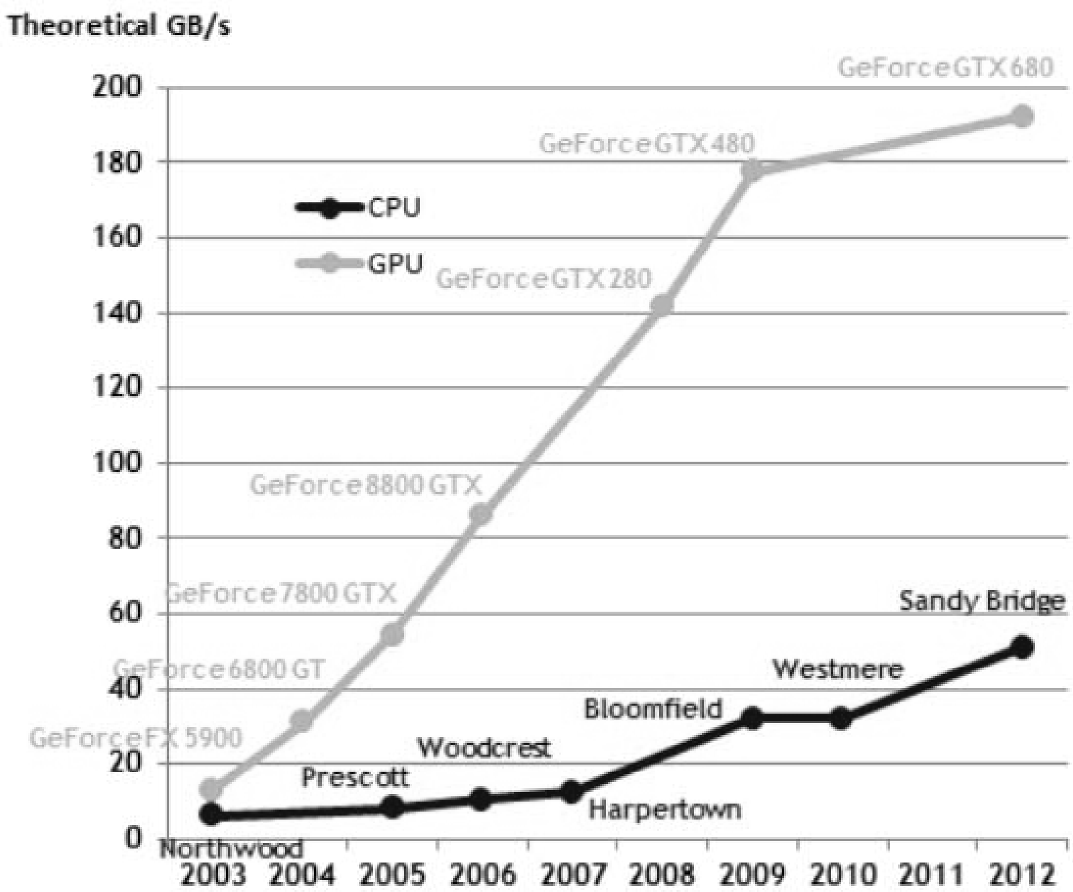

Fig.2. Memory bandwidth for the CPU and GPU

3 GMDH combinatorial algorithm

Let us consider the variants of computations organization in the combinatorial generator. There are many schemes for change of binary structural vector, specifying the sequence of regressors including in the model. Two of them (generally used) are described below.

3.1 Scheme of algorithm with the use of binary numbers generator

The scheme uses generation of binary numbers, corresponding to sequential decimal numbers. Complication of partial models changes from 1 to the maximal number of m.

The scheme is simple enough and effective in case of uniprocessor system. However it is not quite suitable for paralleling of combinatorial algorithm, because it does not provide the even loading on every multiprocessor. The least computational loading (larger common amount of arguments at the identical amount of models) will fall on multiprocessors with a less sequence number, and most loading – on multiprocessors with most sequence number, where models will be with the amount of arguments, near to the number of m. It is related to the feature of structural vector forming on principle of binary counter.

3.2 Scheme of algorithm with successive complication of structures

This scheme uses such sequence of binary numbers generation when at first all connections appear with one unit in a structural vector ($C_m^1 = m$ possible variants), then – with two units ($C_m^2 = \frac{m(m-1)}{2}$ possible variants), and etc to one possible variant $C_m^m = 1$ of including in the model of all arguments.

Such scheme can be easily enough applied for parallelization of combinatorial algorithm. Idea of the equal apportionment of common amount of models and arguments on all processors of the cluster system consists in the following. Amount of models $C_m^i = \frac{m!}{(m-i)! \times i!}$ complexity of i, $i = \overline{1, m}$, is evenly distributed between all processors of p (i.e. every processor "handles" $\frac{m!}{p \times (m-i)! \times i!}$ models). Thus, it is necessary to define an initial point (first structural vector) for every processor.

4 Paralleling of GMDH combinatorial algorithm for GPU

We will consider the scheme of algorithm with successive complication of structures of binary numbers generator. The corresponding scheme for paralleling on CPU was constructed in [4].

The scheme of algorithm paralleling for determination of the initial state of binary structural vector by position for every multiprocessor is proposed later. Paralleling of other parts for GMDH combinatorial algorithm using GPU, as well as using cluster systems [2] presents no substantial difficulties.

Lets we have m arguments and κ multiprocessors within GPU. We will write down the sequence of operations for the models of complication i, $i = \overline{1, m}$:

1. Calculation of amount of combinations – $C_m^i - 1$.

2. Determination of the initial state of binary vector d for every multiprocessor j, $j=\overline{1,k}$, as a decimal number $\left[\frac{C_m^i-1}{k}\right](j-1)+1$.

3. Conversion from the decimal number to appropriate binary number for every multiprocessor:

$$\text{position} = \left[\frac{C_m^i-1}{k}\right](j-1)+1;$$

u=i-1, d=m-1, $C=C_d^u$;

Cycle on l, $l=\overline{1,m}$

if position<=C then $b[l]$=1, u= u -1, d= d -1, $C=C_d^u$;

else $b[i]$=0, position = position – C, u= u -1, $C=C_d^u$.

Conclusion

The principle possibility for the use of graphics processing units for solving combinatorial modelling problems is shown. Scheme of operations paralleling using GPU in a combinatorial algorithm on principle of binary counter is proposed. The scheme provides the uniform loading on all multiprocessors of GPU. Further investigations will be devoted to comparative testing of combinatorial algorithm paralleling using GPU and cluster systems.

References

1. V. S. Stepashko, and S. N. Efimenko. Sequential Estimation of the Parameters of Regression Model // Cybernetics and Systems Analysis, Springer New York, July, 2005, Vol. 41, Num. 4, pp.631-634.

2. Stepashko V.S., Yefimenko S.M., Rozenblat O.P., Chernyack A.I. On application of parallel computations in tasks of modelling on the basis of inductive approach // Problems in programming. – 2006. – № 2-3. – PP. 170-176.

3. NVIDIA CUDA C Programming Guide, [Online]. Available: http://developer.nvidia.com/nvidia-gpu-computing-documentation.

4. V. S. Stepashko, S. N. Efimenko. Optimal paralleling for solving combinatorial modelling problems // Proceedings of the 2nd International Conference on Inductive Modelling ICIM 2008. – Kyiv, 2008. – P. 172-175.