Petro STAKHIV[1], Iryna STRUBYTSKA[2], Yuriy KOZAK[1]

Lviv Polytechnic National University(1), Ternopil National Economic University (2),

Parallelization of calculations using GPU in optimization approach for macromodels construction

Abstract. *Construction of mathematical models for nonlinear dynamical systems using optimization requires significant computation efforts to solve the optimization task. The most CPU time is required by optimization procedure for goal function calculations, which is repeated many times for different model parameters. This allows to use processors with SIMD architecture or calculation parallelization. The effectiveness of such parallelization is the subject of investigation in this paper.*

Keywords: macromodels, optimization, parallelization, GPU, SIMD.

Problem Statement

Let's consider some system, which need to be simulated (see fig. 1). Here $\vec{x}$ - variables which describe system state, $\vec{v}$ - variables which describe external influences, and $\vec{y}$ - variables which describe output values.

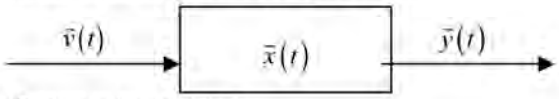

Fig.1 System to be simulated

Due to we are considering dynamical systems all three mentioned vectors are functions of time. This can be either continuous functions or sets of discrete values. Not limiting the generality let's concentrate on analysis of discrete values case, because discrete values is more useful for computers. We will use the discrete state variables form:

$$(1)\qquad \begin{cases} \vec{\mathbf{x}}^{(k+1)} = \mathbf{F}\cdot\vec{\mathbf{x}}^{(k)} + \mathbf{G}\cdot\vec{\mathbf{v}}^{(k)} + \mathbf{\Phi}(\vec{\mathbf{x}}^{(k)}, \vec{\mathbf{v}}^{(k)}) \\ \vec{\mathbf{y}}^{(k+1)} = \mathbf{C}\cdot\vec{\mathbf{x}}^{(k+1)} + \mathbf{D}\cdot\vec{\mathbf{v}}^{(k)} \end{cases}$$

where F, G, C, D – some matrixes with unknown coefficients, which need to be found during model construction, $\mathbf{\Phi}(\vec{\mathbf{x}}^{(k)}, \vec{\mathbf{v}}^{(k)})$ - some vector-function, the form and coefficients of which should also be found.

The mathematical form (1) we selected for our system simulation as well as any other form, has some set of unknown coefficients $\vec{\lambda}$. In our case it includes elements of matrixes F, G, C, D and coefficients of vector-function $\mathbf{\Phi}(\vec{\mathbf{x}}^{(k)}, \vec{\mathbf{v}}^{(k)})$.

Let's also introduce some criterion for model precision measuring $Q(\vec{\lambda}) > 0$, which depicts an inaccuracy of system simulation using the model for known time intervals.

Function $Q(\vec{\lambda})$, also called a goal function is generally calculated as a root-mean-square error of system simulation using constructed model.

Therefore model construction can be reduced to finding a value of vector $\vec{\lambda}$, for which function Q will reach it's minimum. This is done using optimization algorithm.

The task of finding a valley point of nonlinear function Q is a complex task, which is considered in many writings. Our goal function is hard to calculate in terms of needed CPU time for one calculation. The time needed for single calculation of goal function depends linearly from the number of samples, because the goal function is a sum of deviations between real values and values simulated using constructed model for all samples. For construction of a high-quality model the number of samples used for goal function calculation will be significant, thus CPU time needed for single calculation of goal function will also be significant.

Practical usage of described method requires a lot of CPU power, so it is worthwhile to consider using parallelization. Taking into account, that goal function calculation for different values of vector $\vec{\lambda}$ is independent and is done using same algorithm we can state, that processors with SIMD architecture can be effectively used for this task. Such processors allow to execute same instructions for many data sets. SIMD systems includes many identical processor units executing same instruction sets but using independent data sets. Most common device, build on SIMD architecture is GPU (Graphics Processing Unit).

This approach was implemented by authors using CUDA technology. Achieved parallelization effectiveness depends on the number of points for which the goal function is calculated at one step of optimization algorithm. This dependency is shown on fig. 2.

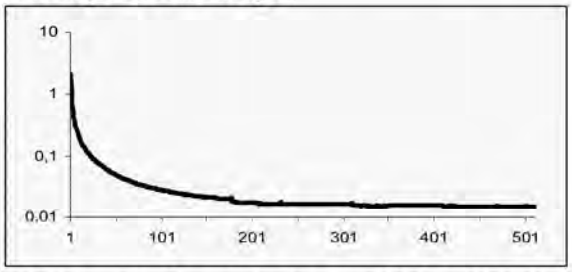

Fig.2 Effectiveness of parallelization using GPU (NVIDIA GeForce GTS250, 1024 Mb) in comarision with Main CPU (Core2Duo E8400, 3GHz).

Conclusion

The usage of GPU allows to spead-up the optimization process. One of the simplest GPU cards allowed to speed-up the computation for more then 5 times comparing to relatively new main CPU.

REFERENCES

[1] Liliana Bychkovska-Lipinska L., Stakhiv P., Kozak Y., Parallelization of Calculations in Construction of Mathematical Models using Optimization. Proceedings of XIII INTERNATIONAL CONFERENCE - SYSTEM MODELLING and CONTROL SMC'2009, ISBN 978-83-927875-0-1

Authors: *Prof. Liliana Byczkowska-Lipińska – Wyższa skola informatyki w Łódzi, lipip@ics.p.lodz.pl; Prof. Petro Stakhiv - Institute of Computer Sciences, Technical University of Łódz, ul. Wolczanska 215, Łódz, 90-924, Poland, petro.stakhiv@p.lodz.pl; Iryna Strubytska – Department of Computer Scince, Institute of Computer Informational Technology, Ternopil National Economic University, Yunosti Str. 9, 46020 Ternopil, Ukraine, iryna.str@gmail.com; Assoc. Prof. Yuriy Kozak - Department of Theoretical Electrical Engineering, Institute of Electric Power and Control Systems, Lviv Polytechnic National University, 12 S. Bandera str. Lviv. 79013. Ukraine ykozak@mail.ru.*