UDC 629.7.05:519.6:681.322 (045)

S.S. Tovkach, Cand. Of Sci.(Engineering)
(National Aviation University, Ukraine)
E.A. Shkvar, Dr. of Sci. (Engineering)
(Zhejiang Normal University, China)

CUDA-based massively parallel computing application for improving the efficiency of turbulent flows modeling and methods of their control

A perspective approach of increasing the computing performance properties of turbulent flows, based on productivity of Graphics Processing Units utilization has presented. It can be effectively applied for development of new principles of adaptive turbulent flow control strategies.

Introduction. Turbulent flow control is the direction of increasing the efficiency and competitiveness of high-speed transport vehicles due to drag reduction and, as a result, fuel consumption decreasing and environment saving. In order to make flow control of the streamlined surface better, the technique of turbulence modeling and improving the computing data performance has been widely used for many years as a tool for the drag reduction of aircraft and has shown its advantages in many aspects [1].

Modern perspective methods of flows modelling with abilities to predict high-resolution features of structure and dynamics of turbulent vortex formation such as Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) provide integration of equations on a spatial grid with high scaling and with very fine pitch by the time variable. This requires high costs of computational resources and a considerable time, which is necessary in the research and engineering activity. So, in researches [2] with serial and multithreaded LES calculations on the base of Symmetric Multiprocessing (SMP) computer with two quad-core processors the parallel solution of Poisson equation had to use about 45% of the total calculation time. Therefore, it is necessary to find more effective methods of parallelization than SMP technology based on computational scaling of solving problem by existing number of Central Processing Units (CPU) or their cores. One of the perspective ways is a translating the most resource demanding elements of solving computational problem to the Compute Unified Device Architecture (CUDA) technology algorithms with further scaling of achieved growth by use in the computations much more powerful Graphics Processing Units (GPU) in comparison with CPU productivity.

The goal of this research is to analyse the perspective ways of improving the efficiency of scaling the process of parallel calculations in turbulent flows modeling on CUDA based graphics accelerators.

Turbulence modeling in general is the meaning to calculate the so-called eddy viscosity; and is taken into account in the viscous flow in the system of the Navier-Stokes equations:

$$\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial x_i}(\rho u_i)=0$$

$$\frac{\partial}{\partial t}(\rho u_i)+\frac{\partial}{\partial x_j}(\rho u_i u_j)=-\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\overline{p}\delta_{ij}\right]+\frac{\partial}{\partial x_j}\left(-\rho\overline{u_i' u_j'}\right) \tag{1.1}$$

The left side of second equation (transient member) describes the change of the of chosen liquid volume momentum due to the change in time as averaging velocity component. This change is compensated in the right part by averaging external forces $\frac{\partial p}{\partial x_i}$, by averaging pressure forces $\overline{p}\delta_{ij}$, viscous forces $\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)$ and turbulent stresses $\left(-\rho\overline{u_i' u_j'}\right)$ taking into account the additional losses and redistribution of energy in turbulent flow.

The method of turbulence modeling with LES for Navier-Stokes equations has following steps:

Filtering procedure: the separation of the vortices in the "large" (more than certain size) and "small":

$$\overline{w}(x)=\int_D w(x')\varphi(x,x')dx' \tag{1.2}$$

where D is the fluid domain, and φ is the filter function that determines the scale of the resolved eddies;

Filtered equations: the construction of such a system of equations, whereby large vortices are resolved accurately (the dependent variable are now filtered quantities rather than mean quantities, and the expressions for the turbulent stresses differ);

Subgrid-scale model: description of "small" vortices and their interaction with large ones that are modelled directly.

Boundary conditions: on the sides of computational area, perpendicular to the longitudinal X-axis, the periodic boundary conditions are traditionally used. It allows to research the three-dimensional effects, which is caused by the internal flow instability.

The construction of the parallel algorithm, as the basic computer operation, is a calculation of the velocity gradients in cells on three spatial directions. It is necessary for both algebraic and differential models of turbulence. In the case of a vertex-centered scheme usage for this calculation node surgery gradients computed at each node by summing the gradients on all grid elements containing the node ([1,3], where the same operation is used for reconstruction of a high-order). The element-centered case for the calculation of gradient applies the method of least squares on the adjacent cells (coefficients computed during the initialization phase when you start).

Computing power performance. Testing the computing performance of turbulence modeling has been made with paralleling technologies: OpenMP and CUDA [2,4]. The first one (OpenMP) focused at multi-threaded programming and is

effective in multiprocessor or multicore systems. The second approach – CUDA is effective for computing performance increasing due to the use of GPU.

For making the analysis of the features of CUDA paralleling technologies four video cards GeForce GTX 680 with 2 GB video memory of GDDR5 have been used, which installed in a system based on six-core CPU Intel I7-3960x with 16 GB RAM DDR3-1600. Each of GPU software (based on OpenMP technology) treated by the corresponding CPU thread, allowing to process independently each of all grid subdomains by separate GPU on every iteration [4].

The results of acceleration scaling calculations [4,5] demonstrate improved efficiency computing on a system with multiple GPU with increasing dimension of the solved problem (fig. 1.1.). Even in case when number of grid nodes in one direction M is great ($M \geq 400$) and the number of GPU doesn't exceed 3, dependence is so close to linear form, but if we realize computations on the base of the 4-GPU computing system the growth of acceleration is slowing down due to increasing the transmitted information between GPU.

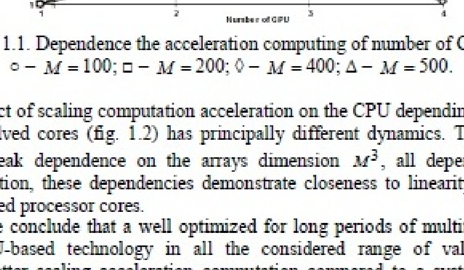

Fig. 1.1. Dependence the acceleration computing of number of GPU
○ – $M=100$; □ – $M=200$; ◊ – $M=400$; Δ – $M=500$.

The effect of scaling computation acceleration on the CPU depending on the number of involved cores (fig. 1.2) has principally different dynamics. Thus, the results show weak dependence on the arrays dimension M^3, all dependencies merged. In addition, these dependencies demonstrate closeness to linearity versus number of all used processor cores.

It can be conclude that a well optimized for long periods of multithreaded computing CPU-based technology in all the considered range of values has demonstrated better scaling acceleration computation compared to a system with multiple GPU. However, if you recalculate the results in absolute figures runtime tasks, priority GPU at $M=400$ becomes 4.76 times.

Conclusion. The efficiency of scaling the parallel process of the most demanding structural elements of LES calculations in turbulent flows modeling on graphics accelerators by using CUDA computational technology has been investigated and analyzed.

The method of turbulence modeling with LES for Navier-Stokes equations has been considered. It helps to better understand the creation of parallel algorithm

to solve governing equations of turbulence by using modern techniques of turbulent flow non-stationary processes analysis.

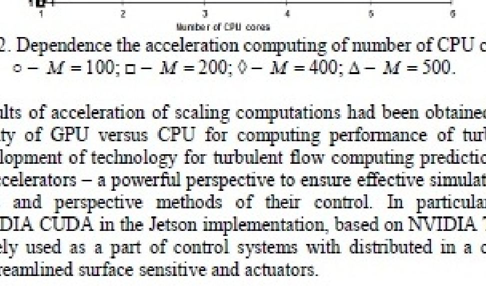

Fig. 1.2. Dependence the acceleration computing of number of CPU cores
○ – $M=100$; □ – $M=200$; ◊ – $M=400$; Δ – $M=500$.

The results of acceleration of scaling computations had been obtained, that show the priority of GPU versus CPU for computing performance of turbulent modeling. Development of technology for turbulent flow computing predictions on with graphics accelerators – a powerful perspective to ensure effective simulation of turbulent flows and perspective methods of their control. In particular, the technology NVIDIA CUDA in the Jetson implementation, based on NVIDIA Tegra, can be effectively used as a part of control systems with distributed in a certain section of the streamlined surface sensitive and actuators.

References

1. Абалакин И.В. Схема на основе реберно-ориентированной квазиодномерной реконструкции переменных для решения задач аэродинамики и аэроакустики на неструктурированных сетках / Математическое моделирование // И.В. Абалкин. – 2013. – Т. 25, № 8. – С. 109 – 136.

2. Шквар Є.О. Гібридний метод паралельних обчислень / Є.О. Шквар // Вісник Черкаського університету: серія Прикладна математика. Інформатика. – Вип. 172. – 2010. – С. 123–136.

3. Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Ra-number up to 1011 // Numerical methods and time-averaged flow / F. X. Trias, A. Gorobets// International Journal of Heat and Mass Transfer. — 2010. — Vol. 53. — Pp. 665–673.

4. Шквар Є.О. Ефективність паралельного розв'язання рівняння Пуассона на обчислювальних системах з кількома графічними прискорювачами / Є.О. Шквар // Вісник НАУ. – 2012. – № 1. – С. 157-166.

5. Шквар Є.О. Інтегрована гібридна технологія паралельних обчислень / Є.О. Шквар // Інтегровані технології та енергозбереження: щоквартальний науково-практичний журнал НТУ (ХПІ). – Харків: НТУ (ХПІ). – 2010. – № 1. – С. 86–99.