JOURNAL OF MECHANICAL AND TRANSPORT ENGINEERING
Vol. 67, no.4 2015
DOI 10.21008/j.2449-920X.2015.67.4.06

Wojciech SZELIGA[*,**],
Marek MORZYŃSKI[**],
Witold STANKIEWICZ[**],
Krzysztof KOTECKI[**]

SCALABILITY TESTS OF THE DIRECT NUMERICAL SIMULATION SOLVER UNS3

In this paper analysis of scalability of the solver UNS3, dedicated to direct numerical simulation (DNS) of Navier-Stokes equations, is presented. Efficiency of parallel computations has been examined with the use of a PC cluster built by the Division of Virtual Engineering. Tests have been carried out on a different number of partitions, in the range of 1÷80. The test case was steady flow around a wall-mounted circular cylinder with Reynolds number set to the value of Re = 10. The research included the measurement of preparatory time, calculation time, communication time, speedup, core hours and efficiency.

Keywords: UNS3, parallel computations, scalability, partitioning, CFD, DNS, MPI, cluster

1. INTRODUCTION

Numerical reconstruction of physical phenomena in fluid flow is still an issue that makes high requirements of computational power and storage space. It is associated with necessity of precise discretization of investigated domain, which results in huge systems of equations to be solved. CFD (computational fluid dynamics) calculations on a single computer, even if possible due to memory requirements, are very time-consuming. Nowadays, to shorten the computational time, the calculations are performed in parallel environment, on many processors or processor cores simultaneously.

[*] Poznan Supercomputing and Networking Center.
[**]Institute of Combustion Engines and Transport, Poznan University of Technology.

To apply parallelization succesfully, the CFD solver has to be characterized by good scalability, defined as the ability to shorten the calculation time by usage of many processors. Information about behavior of CFD applications depending on the number of used cores is extremely important for code developers constantly improving product performance, which is tested with appropriate benchmarks [9].

The goal of this study is the testing of the solver UNS3 applied to CFD calculations, to examine it's ability to effective parallelization. The computations are executed on a PC cluster built by the Division of Virtual Engineering, equipped with two types of processors.

2. DESCRIPTION OF THE UNS3 SOLVER

2.1. Solving governing equations

The UNS3 solver [13, 14] is suitable to perform various analysis of viscous, incompressible flows. It allows global stability analysis using subspace iteration method and direct numerical simulation of Navier-Stokes momentum equation (1), supplemented by continuity equation (2):

$$\partial_t u + \nabla \cdot (u \otimes u) + \nabla p - \frac{1}{\mathrm{Re}} \Delta u = 0 , \tag{1}$$

$$\nabla \cdot u = 0 , \tag{2}$$

where u – velocity vector, p – pressure, Re – Reynolds number.

Governing equations are discretized with the finite element method using the Galerkin projection. As those equations are nonlinear, classic Newton-Rhapson method is used to perform iterations. To define the Jacobian matrix and to set the residuum, the integration of the volumes utilizing symbolic manipulation program is used.

2.2. Elimination of the pressure – the penalty method

The pressure p is eliminated from the equation (1) with usage of the penalty method [1, 3]. The LBB condition (Ladyzhenskaya, Babuska and Brezzi) [2, 5, 6] results in the application of lower order shape function for pressure than for velocity.

In the penalty method there is introduced artificial, small compressibility into incompressible form of Navier-Stokes equation which changes the character of global stiffness matrix. It influences the conditioning of the matrix, so it is important for the solving procedure [14]. The pressure is replaced with the velocity gradient by usage of ε coefficient, as it is stated in the formula (3):

$$\frac{1}{\varepsilon} p = \nabla u \,. \tag{3}$$

The value of the parameter ε is usually set as a large number. In the range of 10^3 to 10^7 it has imperceptible influence onto solving steady solution, as shown in numerical experiments [14]. In described tests the value of mentioned parameter was set to $\varepsilon = 10^5$. For lower values of ε it is possible to control the speed of disturbance propagation in the domain during initial iterations. This approach shortens the time of receiving convergence, as the conditioning of the Jacobian matrix improves [14].

2.3. Multithreading implementation

In-house solver UNS3 allows parallel solution of governing equations on the distributed- and shared-memory systems. Communication between nodes of the cluster is realized by MPI (Message Passing Interface) library [4]. The partitioning of the domain (Fig. 2.1) is done by the calls to METIS [10, 11] software.

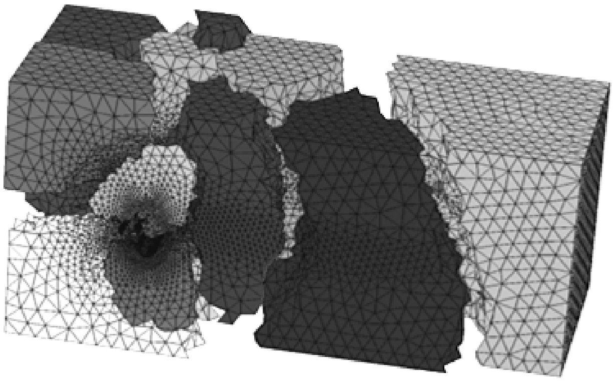

Fig. 2.1. Example of mesh partitioning for 16 parts [13] with usage of METIS

3. RESEARCH METHODOLOGY

3.1. Cluster structure

Cluster used for the test consists of 18 multi-core nodes. For parallel computations 17 of them are used, including:

– 10 nodes containing quad core processors Intel® Core$^{\mathrm{TM}}$ 2 Quad Q9550 2.83 GHz and 8 GB of RAM each,

– 7 nodes containing quad core processors Intel® Xeon® E31240 3.30 GHz, with physical cores each divided to 2 logical ones with the usage of the Intel® Hyper-Threading Technology [7] and 16GB of RAM each.

One unused node, equipped with 24 computational cores and 48 GB of RAM, is reserved for cluster management and modal analysis of the results.

The cluster works under control of the openSUSE 11.2 operating system. Data exchange is realized with the InfiniBand™ interface [8], allowing 10 Gbit transfer with single fiber-optic cable connection and possible 20 GBit transfer with double fiber-optic cables used.

3.2. Examined configurations

To define operating characteristics of UNS3 solver on multiple CPUs, tests in the following configurations have been run (cores × nodes):

– Intel® Core™ 2 Quad nodes: 1 × 1, 1 × 2, 4 × 1, 4 × 2, 4 × 3, 4 × 4, 4 × 5, 4 × 6, 4 × 7, 4 × 8, 4 × 9, 4 × 10.

– Intel® Xeon® nodes: 1 × 1, 1 × 2, 1 × 4, 8 × 1, 8 × 2, 8 × 3, 8 × 4, 8 × 5, 8 × 6.

– mixed arrangement: 4 × 2 + 8 × 1, 4 × 4 + 8 × 2, 4 × 6 + 8 × 3, 4 × 8 + 8 × 4, 4 × 10 + 8 × 5.

3.3. Obtained parameters

For every configuration the following set of solver operating parameters has been obtained:

– preparation time t_{set} [s],

– communication time t_{comm} [s],

– solver operating time (includes t_{comm}) t_{solve} [s],

– work ratio *WR* [%], coefficient that specifies the level of usage operating time for actual computations, as described below (4):

$$WR = \frac{t_{solve} - t_{comm}}{t_{solve}} \cdot 100[\%], \tag{4}$$

– core hours *CH* [h], time of calculations multiplied with amount of partitions *p*

– speedup S_p, speeding of calculations related to single thread according to the formula (5):

$$S_p = \frac{T_1}{T_p}, \tag{5}$$

where T_1 – time of serial computations, T_p – time of calculations with p threads

– efficiency E_p, effectiveness of resources usage as described in (6):

$$E_p = \frac{S_p}{p}. \tag{6}$$

3.4. Test case

A steady flow of incompressible fluid past wall mounted circular cylinder has been chosen as the test case. In engineering applications, finite in length, cylinder-shaped objects are used to simulate the flow past chimneys, stacks, high-rise buildings or legs of oil platforms [15]. This simple geometry, in general, is a source of complex phenomena like horseshoe vertices, von Karman vertex street and trailing vertices [12]. Although, to provide simple and roust test case, steady wake for the Reynolds number Re = 10 has been investigated using direct numerical simulation of Navier-Stokes equations. The value of Re was chosen as a compromise between a scale of observed phenomenon and computational effort for all the tested configurations. Furthermore, numerical stability is easy to maintain, what makes this test case easy to use in case of testing any other code. The problem has been non-dimensionalized using characteristic dimension, cylinder diameter D, equal to unit. The proportions of the domain have been set to: 25D × 8D × 10D. Cylinder was placed in the distance of $l = 5D$ from the inlet surface. The fluid domain has been discretized using second order tetrahedral elements. Resulting finite-element mesh (Fig. 3.1) consisted of 1 154 928 nodes.

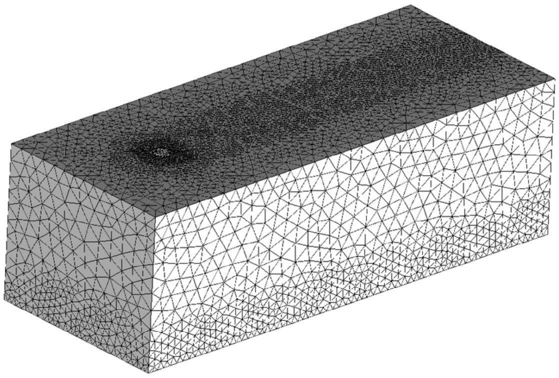

Fig. 3.1. Test case mesh

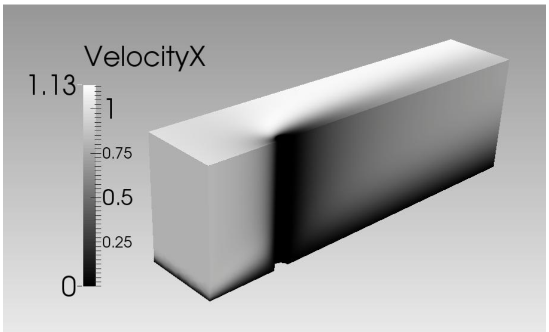

Fig. 3.2. X component of velocity, cross-section on the middle *xz* surface

For every tested hardware configuration, 4 iterations of the solution of Navier–Stokes equations have been performed. Each one consisted of 3 thousand of iterations solving a nonlinear system of equations with Newton–Raphson method. The result of the analysis is shown in Fig. 3.2.

4. COMPARISON AND ANALYSIS OF THE RESULTS

Obtained values of operating time t_{solve} are shown in Fig. 4.1. For every of 3 groups of the sets, the more processors are used, the shorter operating time is. In the range of $p < 8$ (one CPU) calculations were completed faster with usage of processors Intel® Xeon®. Eventual lack of RAM in mentioned range was excluded, as the whole test case occupied ~5GB of RAM, which is the amount not exceeding resources of single node. With the growth of processors number, operating time is similar for both CPU types. For mixed arrangement time t_{solve} is disproportionately longer.

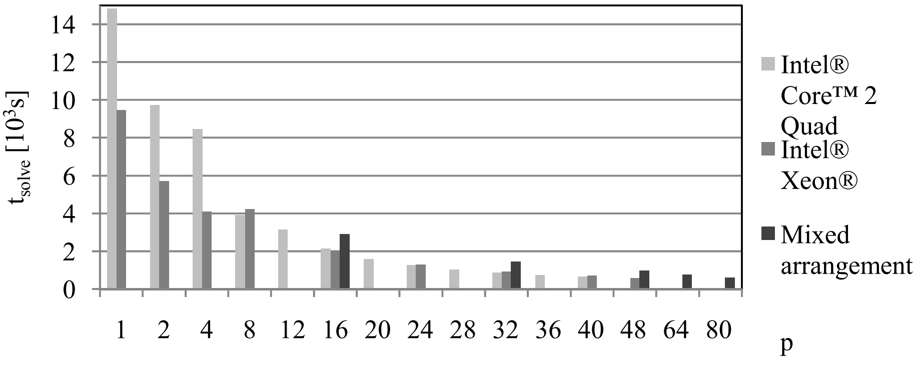

Fig. 4.1. Operating time t_{solve} depending on the p value

The growth of operating time in the case of mixed arrangement is associated with the elongation of communication time t_{comm}. According to the Fig. 4.2., it is longer by one order of magnitude, when CPU types are mixed. Apart from the slowdown caused by unequal performance of both CPU types, it is also due to hardware differences of nodes.

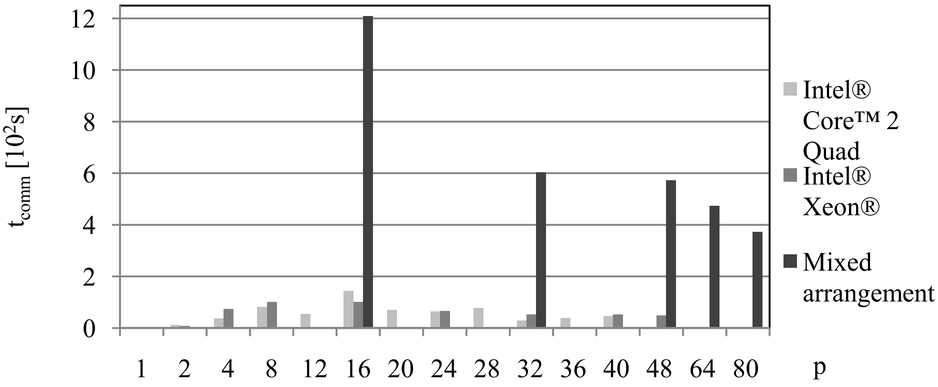

Fig. 4.2. Communication time t_{comm} depending on the p value

Quantitative relationship of solver's operating time spent to actual numerical calculations to all the t_{solve} time is stated as *WR* coefficient in the Fig. 4.3. In the case of the same processor type usage, this parameter does not decrease under 91.56%. It turned out that this coefficient is not relevant for proper processors quantity choice.

Work ratio is significantly decreasing for mixed arrangement. This situation is caused by aforementioned growth of communication time t_{comm}. For p = 16 it has the value of 58.27%, decreasing to 37.85% for p = 64.

Parameter *CH* is quantitative determinant of computational resources usage. Results stated in Fig. 4.4. show that for homogeneous configurations the *CH* parameter is the smallest for $p = 1$ and increases to maximum for $p \approx 12$. After that the value slowly decrease with the growth of the p value. The range below $p \approx 12$ is not in focus despite the smallest values of resources usage, because operating time is relatively long in comparison to bigger p values. It means that, getting into account also t_{solve}, it is worth to partition case to as many parts as possible.

For mixed arrangement *CH* value is much bigger, leading to the conclusion that this is uneconomical configuration.

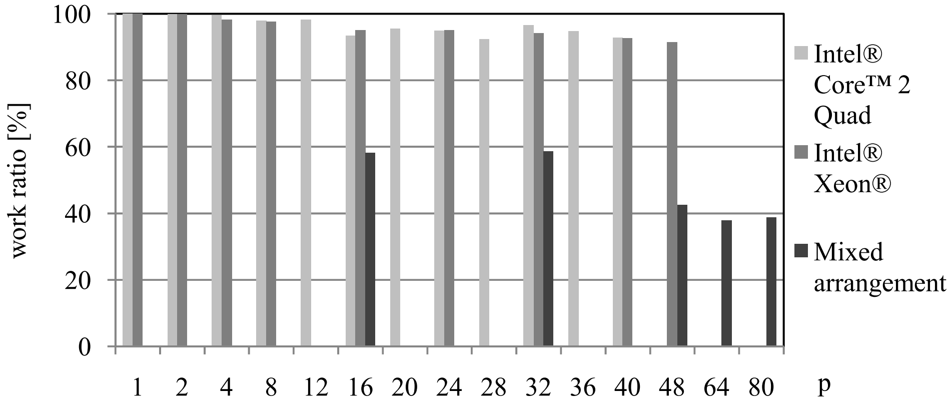

Fig. 4.3. Work ratio *WR* depending on the *p* value

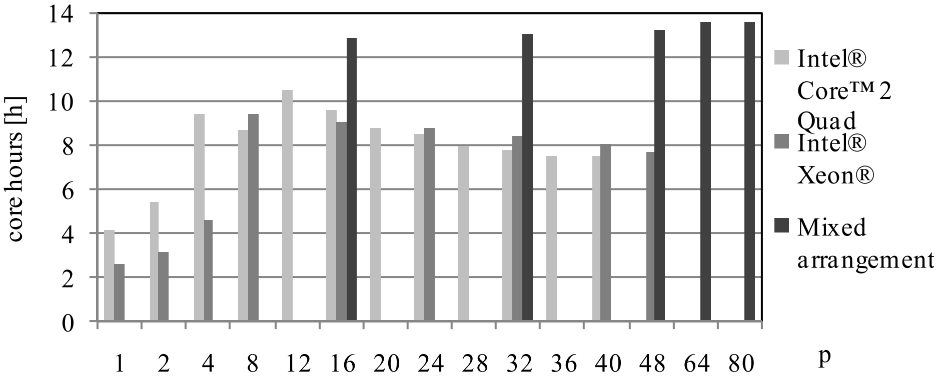

Fig. 4.4. Core hours *CH* depending on the *p* value

The relation of single thread calculations time to t_{solve} of other configurations is shown in the Fig. 4.5. For the comparison there is also stated line determining ideal speedup, meant by speedup equal to the *p* value. Processors Intel® Xeon® offer lower values of speedup. However, it has to be taken into account, that the value of *p* in the case of Intel® Xeon® CPUs represents the amount of logical cores, not physical ones.

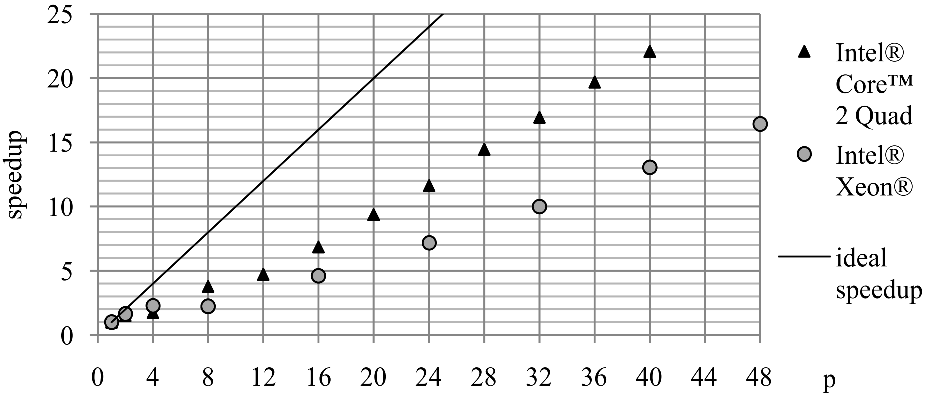

Fig. 4.5. Speedup S_p depending on the p value

There was noted similarity of shapes of characteristics of E_p (Fig. 4.6.) and trends of *CH* presented on Fig. 4.4.

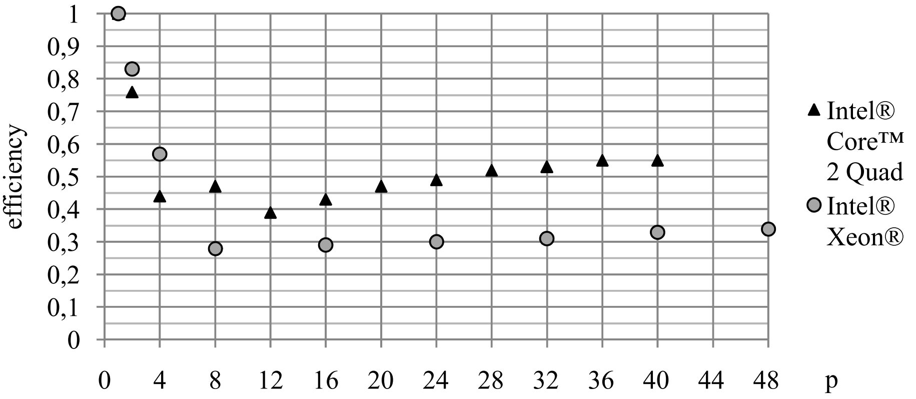

Fig. 4.6. Efficiency E_p depending on the p value

The most beneficial values are assigned to the smallest p values, below p = 4. As mentioned before, these are not valuable configurations because of big t_{solve} parameter value, so more relevant is the growth of the efficiency taking place with increasing p. For number of partitions over p = 4, processors Intel® Core$^{\text{TM}}$ 2 Quad offer higher efficiency than other CPUs. As in the case of speedup there has to be taken into account the way of obtaining p value in case of Intel® Xeon® processors.

5. SUMMARY

In performed tests it was proven that for parallel computations in the range of examined configurations (without taking into account mixed arrangement), communication time t_{comm} is small enough to be neglected as a parameter that affects determination of proper number of partitions of computational domain.

The experiments with mixed CPU types resulted in dramatic increase of communication time. Mentioned behavior, probably caused by different node architectures, makes mixed configuration the uneconomical one.

It has been observed, that for the number of threads p smaller than 8, Intel® Xeon® processors offer shorter operating time than Intel® Core^{TM} 2 Quad. For $p \geq 8$, Intel® Core^{TM} 2 Quad processors offer better efficiency of computational resources usage.

It was demonstrated that the more processors are involved, the bigger decrease of core hours is seen, what means that in the case of queuing system usage, the best solution is to divide the case to as many partitions as possible, without taking into account configurations related to single cluster node.

For processors Intel® Core^{TM} 2 Quad there were observed bigger values of speedup in comparison to Intel® Xeon® CPUs, exception were configurations with $p = 4$.

Both speedup and efficiency parameters were, in the majority of cases, more favorable for Intel® Core^{TM} 2 Quad processors, nevertheless it has to be noted, that the number of cores of Intel® Xeon® processors configurations concerned logical cores, not physical ones.

By the time of running described tests, turning off Intel® Hyper-Threading Technology was not possible on used cluster. Another set of scalability tests with mentioned technology turned off with comparison to results stated in this paper is considered valuable and planned in the future when possible.

ACKNOWLEDGEMENTS

This work has been funded by National Centre of Science under research grant no.2011/01/B/ST8/07264 ”Novelmethod of PhysicalModal Basis Generation for Reduced Order Flow Models”.

REFERENCES

[1] Baker A.J., Finite Element Solution Algorithm for Viscous Incompressible Fluid Dynamics. John Wiley and Sons, New York, 1983.

[2] Brezzi F., On existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique, 8(2), 1974, p. 129–151.

[3] Chung T.J., Computational Fluid Dynamics, Cambridge University Press, Cambridge, 2002.

[4] Gabriel E., Fagg G.E., Bosilca G., Angskun T., Dongarra, J.J., Squyres J.M., Sahay V., Kambadur P., Barrett B., Lumsdaine A., Castain R.H., Daniel D.J., Graham R.L. and Woodall T.S., Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation, In Proceedings, 11th European PVM/MPI Users' Group Meeting, Budapest, Hungary, September 2004.

[5] Girault V., Raviart P., Finite Element Methods for Navier-Stokes Equations, Springer Verlag, Berlin, 1986.

[6] Gunzburger M., Finite Element Methods for Viscous Incompressible Flows, Academic, Boston, 1989.

[7] Intel ® Hyper-Threading Technology, http://www.intel.com/content/www/us/en/architecture-and-technology/hyper-threading/hyper-threading-technology.html (access date: 14.01.2014).

[8] Introduction to InfiniBand^{TM} for End Users, InfiniBand Trade Association, April 2010.

[9] Jamshed S., Using HPC for Computational Fluid Dynamics: A guide to High Performance Computing for CFD Engineers, Chapter 6 – HPC Benchmarks for CFD, Amsterdam: Academic Press, 2015, p. 119-155.

[10] Karypis G., Kumar V., A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs, SIAM Journal on Scientific Computing, Vol. 20, Iss. 1.

[11] Karypis G., Kumar V., MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0, University of Minnesota, June 1995.

[12] Kawamura T., Hiwada M., Hibino T., Mabuchi I., Kumada M., Flow around a finite circular cylinder on a at plate (cylinder height greater than turbulent boundary layer thickness), Bulletin of the Japan Society of Mechanical Engineers, 27 (232), p. 2142–2151, 1984.

[13] Morzyński M., Thiele F, 3D FEM Global Stability Analysis of viscous flow, In Lecture Notes in Computer Science, Parallel Processing and Applied Mathematics, Vol. 4967, Springer-Verlag, 2008, p. 1293–1302.

[14] Morzyński M., Thiele F., Finite Element Method for Global Stability Analysis of 3D Flows, In AIAA-Paper, number 2008-3865, 38th AIAA Fluid Dynamics Conference and Exhibition, 23–26th June, Seattle, Washington, USA, 2008.

[15] Wang H.F., Zhou Y., Mi J., Effects of aspect ratio on the drag of a wall-mounted finite-length cylinder in subcritical and critical regimes, Experiments in fluids, 2012, Vol. 53 (2), p. 423–436.

TESTY SKALOWALNOŚCI SOLWERA BEZPOŚREDNIEJ SYMULACJI NUMERYCZNEJ UNS3

Streszczenie

W niniejszym artykule zawarto analizę skalowalności solwera UNS3 służącego do obliczeń CFD (ang. *computational fluid dynamics*) typu DNS (ang. *direct numerical simulation*). Skuteczność wykorzystania wielowątkowości sprawdzano przy użyciu klastra Zakładu Inżynierii Wirtualnej. Badania prowadzono na procesorach typu Intel® Core^{TM} 2 Quad oraz Intel® Xeon® przy ilości partycji w zakresie 1÷80. Za testowe zadanie posłużyły obliczenia stacjonarne opływu cylindra o przekroju kołowym zamocowanego na ścianie, przy liczbie Reynoldsa *Re* = 10. Badano czas obliczeń, czas komunikacji międzywęzłowej, przyspieszenie w wyniku zrównoleglenia, zużycie zasobów oraz efektywność ich wykorzystania.