Computers and Structures 133 (2014) 30–38

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Super linear speedup in a local parallel meshless solution of thermo-fluid problems

G. Kosec *, M. Depolli, A. Rashkovska, R. Trobec

Department of Communication Systems, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 15 May 2013
Accepted 27 November 2013
Available online 25 December 2013

Keywords:
Multicore
Cache
OpenMP
Superlinear speedup
Meshless
Natural convection

ABSTRACT

The performance of the parallel implementation of the local meshless numerical method in solving system of coupled partial differential equations is explored. Presented numerical approach makes the computation convenient for parallel implementation using OpenMP based parallelisation. The numerical experiments are performed on the de Vahl Davis natural convection case, with superlinear computational speedup regime identified. The phenomenon is further investigated through measurements of the central processing unit cache hit rates. It is demonstrated that the accumulation of L3 caches governs the superlinear speedup. Considering the presented analyses, basic rules for effective computation strategy regarding the multicore computations are suggested.

1. Introduction

Numerical analysis and computer modelling are becoming basic tools for technological and scientific research. Numerous problems, e.g. fluid flow, various transport phenomena, weather dynamic, etc., require adequate discretization techniques to be addressed. In the majority of numerical simulations, the Finite Volume Method (FVM) [1], the Finite Difference Method (FDM) [2], the Boundary Element Method (BEM) [3] or the Finite Element Method (FEM) [4] are used. However, in the last few years, new class of numerical methods, referred to as the meshless methods [5], is becoming popular as an alternative. The treatment of complex geometries is within the meshless framework much simplified since no topological relations between computational nodes are needed. Several different meshless methods exist [6–8] and this work is focused on one of the simplest among them – the point interpolation [9] Local Radial Basis Function Collocation Method (LRBFCM) [10]. The main advantage of the local numerical method is that the system matrix remains sparse or banded, which simplifies the solution procedure. In contrast, a global approach [11] might become unstable for increasing number of discretization points, demands a lot of computational resources, and complicates the computer program implementation. Besides the simpler formulation, the local solution procedure also enables higher parallel efficiency. From the computation point of view, the localisation reduces inter-processor communication, which is often a bottleneck of parallel algorithms [12]. The computation time is an important factor in numerical simulations and it is often not addressed adequately. An important part of the numerical approach is thus the effective implementation of the solution procedure on modern computer architectures. The developments in the technology of the computer architectures are nowadays extremely vivid. The processing power can be increased either by increasing the processor's clock frequency or by increasing the number of processing units. The clock frequencies are approaching their physical limits; therefore the second option – increased number of processing units – is becoming more attractive. Parallel computers, available today in most desktop computers or computer servers, can compensate for the lack of performance of a single computer, but only in cases where an efficient parallelization of the computational method is known. Various application programming interfaces (APIs) for parallel programming are used to maximise the performance of parallel systems. Nowadays, the most widely used APIs for parallel programming are MPI for distributed-memory systems, and Ptreads and OpenMP for shared-memory systems [13]. Moreover, using graphical processing units (GPUs) for solving parallel problems is widely spreading. APIs that support parallel programming on GPUs are becoming more and more popular, like CUDA and OpenCL [14,15]. There are several publications regarding the parallelization of different numerical schemes for various applied problems [12,16–18], mostly based on MPI parallelization, but only a few numerical studies tackle the influence of the cache memory effects on the performance of parallel computations [19].

* Corresponding author. Tel.: +386 1 477 3827.
E-mail addresses: gkosec@ijs.si (G. Kosec), matjaz.depolli@ijs.si (M. Depolli), aleksandra.rashkovska@ijs.si (A. Rashkovska), roman.trobec@ijs.si (R. Trobec).

0045-7949/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2013.11.016

In this paper, we demonstrate the efficiency of an OpenMP [20] based parallel implementation of the completely local meshless solution of the classical de Vahl Davis benchmark test case [21] on a multicore multiprocessor architecture. The paper contributes two basic messages. First, the parallelization of the proposed meshless based numerical scheme is straightforward on shared-memory systems. A minor amount of effort and expertise are required to apply OpenMP parallelization if the sequential code is ready, thus the approach is interesting for engineering computations. On the other hand, the method offers several convenient features, like ease of implementation, stability, accuracy, and good convergent behaviour [22] that have been already successfully proved on several demanding non-linear coupled problems [23–25].

Second, the efficiency of a parallel implementation can be gravely affected by the memory architecture of a computational system. It is demonstrated that extreme superlinear speedup can be achieved if appropriate architecture is used for a specific problem size. The effect is explained by measurements of the central processing unit (CPU) counters through execution of a simulation. It is clearly shown that accumulating L3 caches govern the effect. In other words, it is not only the power of CPU that matters in intense simulations; communication speed is equally important.

The rest of the paper is organised as follows. In Section 2, the test problem is described. Next, the meshless solution methodology and LBRFCM are briefly presented, followed by description of the parallel program implementation. Section 5 is devoted to the analysis and interpretation of the obtained experimental results. Concluding section summarises the results and provides suggestions for the users dealing with complex realistic numerical problems.

2. Governing equations

The most standard free fluid flow benchmark test is the well-known de Vahl Davis natural convection test [21]. There are several numerical solutions published in the literature [24,26,27] that make the tests convenient for benchmarking purposes. The problem domain is a closed air-filled square-shaped cavity with differentially heated vertical walls with temperature difference ΔT and insulated horizontal walls. Non-permeable and no-slip velocity boundaries are assumed. The problem dynamics is described by three coupled partial differential equations (PDEs) equations: mass (1), momentum (2) and energy conservation (3) equations, where all material properties are considered to be constant. The Boussinesq approximation (4) is used for the treatment of the body force in the momentum equation. The natural convection is thus described by the following system of equations

$$\nabla \cdot \mathbf{v} = 0, \tag{1}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \nabla \cdot (\mathbf{v}\mathbf{v}) = -\nabla P + \nabla \cdot (\mu \nabla \mathbf{v}) + \mathbf{b}, \tag{2}$$

$$\rho \frac{\partial (c_p T)}{\partial t} + \rho \nabla \cdot (c_p T \mathbf{v}) = \nabla \cdot (\lambda \nabla T), \tag{3}$$

$$\mathbf{b} = \rho[1 - \beta_T (T - T_{\mathrm{ref}})]\mathbf{g}, \tag{4}$$

with $\mathbf{v}(u, v)$, P, T, λ, c_p, $\mathbf{g}$, ρ, β_T, T_{ref}, μ and $\mathbf{b}$ standing for velocity, pressure, temperature, thermal conductivity, specific heat, gravitational acceleration, density, coefficient of thermal expansion, reference temperature for Boussinesq approximation, viscosity and body force, respectively. The thermo-physical properties are assumed constant in the de Vahl Davis case. The case is characterised by two dimensionless values

$$\mathrm{Ra} = \frac{|\mathbf{g}|\beta_T \Delta T \Omega_H^3 \rho^2 c_p}{\lambda \mu}, \tag{5}$$

$$\mathrm{Pr} = \frac{\mu c_p}{\lambda}, \tag{6}$$

referred as Rayleigh and Prandtl numbers, respectively. $\Omega_{H,W}$ stands for the domain dimension (Fig. 1).

3. Solution procedure

In this work, we focus on a local meshless numerical method with a local pressure–velocity coupling. The general idea behind the method is the use of local sub clusters of discretization nodes termed as local support domains (Fig. 2). Within a selected support domain, an arbitrary field is approximated as a linear combination of weighted basis functions

$$\theta(\mathbf{p}) = \sum_{n=1}^{N} \alpha_n \Psi_n(\mathbf{p}), \tag{7}$$

where θ, N, α_n and Ψ_n, $\mathbf{p}(p_x, p_y)$ stand for the approximation function, the number of basis functions, the approximation coefficients, the basis functions and the position vector, respectively. Such an approximation function is created in each discretization point. Considering the analysis from Franke [28], we use Hardy's Multiquadrics (MQs) for the basis functions. We use the collocation approach, i.e. the number of support points is the same as the number of the basis functions. After the solution of local systems, i.e. determination of unknown coefficients $\boldsymbol{\alpha}$, the arbitrary spatial differential operation L can be evaluated (7)

$$L\theta(\mathbf{p}) = \sum_{n=1}^{N} \alpha_n L\Psi_n(\mathbf{p}). \tag{8}$$

The computation of the coefficients and the evaluation of the differential operators can be combined in a single operation. The differential operator $\boldsymbol{\chi}^L$ vector is introduced as

$$\chi_m^L(\mathbf{p}) = \sum_{n=1}^{N} \Psi_{nm}^{-1} L(\Psi_n(\mathbf{p})) \tag{9}$$

and a differential operation is thus simplified to

$$L\theta(\mathbf{p}) = \sum_{n=1}^{N} \chi_m^L(\mathbf{p}) \theta(\mathbf{p}_n). \tag{10}$$

The structured formulation is convenient for implementation since most of the complex and CPU demanding operations are performed in the pre-process phase. The Neumann boundary conditions are computed directly by Eq. (10) while the Dirichlet conditions are explicitly set. More details about the presented spatial discretization can be found in [29].

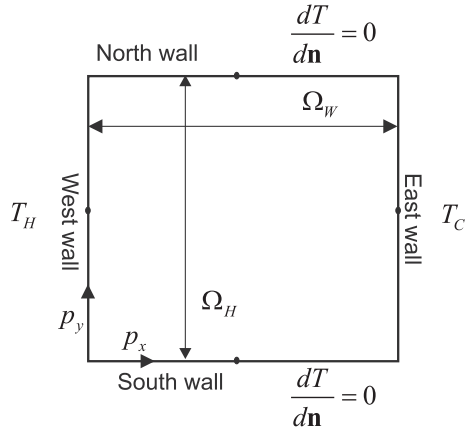

Fig. 1. The geometry and boundary conditions of natural convection benchmark test.