Implementation of parallel version of cellular automata static recrystallization algorithm

Mateusz Sitko[1*] **and Lukasz Madej**[2]

[1] *Faculty of Metals Engineering and Industrial Computer Science, AGH University of Science and Technology*
al. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: msitko@agh.edu.pl

[2] *Faculty of Metals Engineering and Industrial Computer Science, AGH University of Science and Technology*
al. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: lmadej@agh.edu.pl

Abstract

Implementation of parallel version of the cellular automata (CA) static recrystallization algorithm is the subject of the present paper. First, a short description of the major assumptions of the sequential version of the CA algorithm are presented. Then division of the cellular automata computational domain between subsequent computational nodes based on an equal rectangular domains scheme is discussed. Finally, major implementation details of investigated parallelization approach based on the Message Passing Interface (MPI) standard as well examples of obtained speed ups for two dimensional case studies are presented.

Keywords: cellular automata, parallelism, MPI standard, static recrystallization

1. Introduction

Sophisticated in-use properties of metallic products are directly related to complicated thermo-mechanical treatment operations. There are two important phenomena, that allow controlling microstructure evolution during hot deformation conditions: recovery and recrystallization, respectively [2]. Thus, there is a lot of research carried on across the globe to understand in detail influence of these phenomena on properties of new materials [1]. Numerical modelling of microstructure evolution became recently a very important part of that investigation as it can directly reduce research costs. As a result, modern material models with improved predictive capabilities are being intensively developed. These numerical models are often based on the discrete techniques like cellular automata (CA) or Monte Carlo (MC) because these approaches take into account not only kinetics of the process but also microstructure morphology changes in an explicit manner [3,7]. However, major weakness of these advanced methods is computational time, that increases with the increase in predictive capabilities [6]. Especially in 3D, computing times are an enormous obstacle for practical application of developed sophisticated models.

One of the methods to reduce computation time of the cellular automata calculations, which is investigated within the work, is application of code parallelization schemes. The cellular automata static recrystallization model developed in [7] is selected as a case study.

2. Cellular automata model of static recrystallization

In the static recrystallization (SRX) model based on the CA method two major phases control progress of simulation: nucleation and grain growth, respectively. During the simulation each CA cell can be in one of the two defined states: unrecrystallized or recrystallized. Each CA cell in the unrecrystallized state can become a nuclei of a recrystallized grain during nucleation stage. This transition is controlled with the probability calculated according to:

$$p = StM_N \exp\left(-\frac{Q_a}{RT}\right) \tag{1}$$

where: S – volume in which nucleation can appear, t – time, M_N – mobility coefficient, Q_a – activation energy for nucleation, R – universal gas constant, T – temperature.

When the CA cell changes state to a nuclei, then the grain growth stage can take place. This process is driven by two driving forces: the stored energy due to plastic deformation or the grain boundary curvature. The governing equation controlling the process of recrystallization front movement is calculated as:

$$v = MP = M\frac{\varepsilon}{a\varepsilon + b}\gamma_{LAGB} \tag{2}$$

where: M – grain boundary mobility, P – net pressure on the grain boundary, ε – equivalent strain, γ_{LAGB} – low angle grain boundary energy, a, b – coefficients,.

Finally, based on velocity and dimensions of the CA cell, the cell coverage by the recrystallization front is calculated as:

$$RX_{i,t}^{fraction} = RX_{i,t-1}^{fraction} + \sum_{j=1}^{rx}\left(\frac{v_j t_{step}}{c_s}\right) \tag{3}$$

where: $RX_{i,t-1}^{fraction}$ - the level of coverage of the *i–th* cell in the previous (t-1) time step, rx – number of recrystallized neighbours (stored energy driving force) or number of neighbouring cells that belong to recrystallized grains (grain boundary curvature driving force), v_j – velocity of the recrystallization front or recrystallized grain boundary, t_{step} – length of time step, c_s – cellular automata cell size.
When the recrystallized volume fraction reaches critical value, the cell changes its state to a recrystallized. The sequential version of the model has already been validated with experimental data and proved its predictive capabilities [4].

3. Parallelization concept

When parallel execution of the CA algorithm is discussed, first a problem related with efficient division of the CA space have to be addressed. The MPI standard [5] are used in the present work to parallelize the CA static recrystallization code.

* **Acknowledgements**. Financial assistance of the NCN project no 2016/21/N/ST8/00194 is acknowledged.

In the present work a concept of the space division scheme based on equal rectangular computational domains, which are distributed between computing nodes, is investigated as presented in Figure. 1.

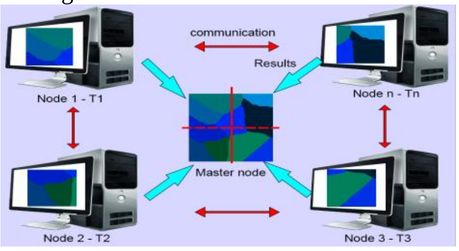

Figure 1. MPI master-slave computational concept.

In the approach only one node stores information on entire investigated CA space, however at the same time performs computation only on a part of this space. Remaining nodes store information about rectangular part of the computational domain received from the master node and also about additional boundary cells from neighbouring computing nodes.

Analysis of efficiency and scalability was perform to evaluate capabilities and limitations of proposed approach based on equal rectangular computational domains.

4. Results

2D CA space with 1000×1000×1 cells containing information on microstructure morphology and accumulated deformation energy was used as an input data to CA static recrystallization model [6]. To evaluate efficiency of implemented parallelization solutions, calculations were realized on 2xIntel Xeon E5-2420 v2 @2.20GHz processors and 64GB DDR3-1600 memory with different amount of MPI computing nodes - 1 to 20. To evaluate obtained results computation speedup was calculated:

$$S(p)=\frac{T_S}{T(p)} \qquad (4)$$

where: T_S – computation time for best sequential algorithm for one node, $T(p)$ – computation time for parallel algorithm on p nodes (Figure 2).

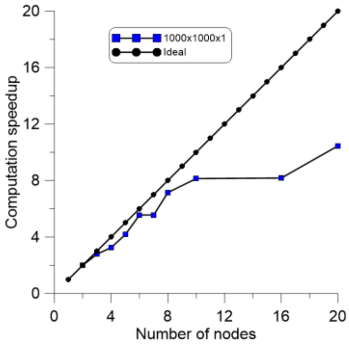

Figure 2. Speed up of single simulation.

As can be seen in Figure 2 simulation performed with 10 computational nodes gives almost ideal speedup. However above this level, speedup dramatically drops down. This behavior is directly attributed with increased amount of data that are exchanged within the master and slave nodes.

The second analysis performed within the work is related with scalability of parallel implementation of the CA code. Set of computational domains with increasing sizes: 1000×1000×1, 2000×1000×1, 4000×1000×1, 8000×1000×1, 16000×1000×1 was used for that investigation. Simulation on 1, 2, 4, 8 and 16 computational nodes was realized to evaluate if computational time scales with increasing domain size and node number based on:

$$S^S(p)=\frac{p*T(1,W_0)}{T(p,pW_0)} \qquad (5)$$

where: $T(1,W_0)$ – computation time for 1 node, $T(p,pW_0)$ – computation time for p-times bigger computational task on p nodes. Obtained results are summarized in Figure 3.

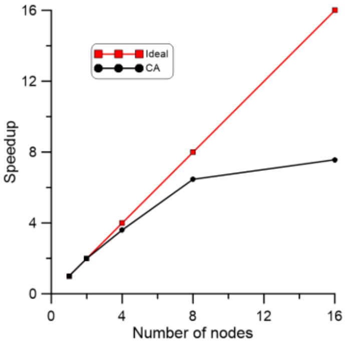

Figure 3. Scalable speedup for different task size on different number of computational nodes.

5. Conclusions and future work

Based on the presented results it can be stated that the MPI is an efficient tool to parallelize computations based on discrete analysis methods like cellular automata. Significant computation speedup with respect to increasing number of computing nodes is observed during simulation at the present stage of research, however some algorithm improvement especially for communication have to be introduced.

References

[1] Fei C., Ke Q., Zhenshan C., Xinmin L. Modeling the dynamic recrystallization in austenitic stainless steel using cellular automaton method, *Comp. Mat. Sci*., 83, pp. 331–340, 2014.

[2] Humphreys M.J. and Hatherly M., *Recrystallization and Related Annealing Phenomena*, 2nd ed. Elsevier, Oxford, 2004.

[3] Madej L., Rauch L., Perzynski K. and Cybulka P., Digital Material Representation as an efficient tool for strain inhomogeneities analysis at the micro scale level, *Arch. Civ. Mech. Eng.*, 11, pp. 661–679, 2011.

[4] Madej L., Sitko M., Radwanski K., Kuziak R., Validation and predictions of coupled finite element and cellular automata model: Influence of the degree of deformation on static recrystallization kinetics case study, *Mater. Chem. Phys.*, 179, pp. 282-294, 2016.

[5] Millan E., Bederian C., Piccoli M., Garino C., Bringa E., Performance analysis of Cellular Automata HPC implementations, *Comput. Electr. Eng.*, 48, pp. 12–24, 2015.

[6] Pietrzyk M., Madej L., Rauch L., Szeliga D., *Achieving High Accuracy and Efficiency in Metals Processing Simulations*, Elsevier, 2015.

[7] Sieradzki L. and Madej L., A perceptive comparison of the cellular automata and Monte Carlo techniques in application to static recrystallization modeling in polycrystalline materials, *Comp. Mat. Sci*., 67, pp. 156–173, 2013.