ZESZYTY NAUKOWE UNIWERSYTETU SZCZECIŃSKIEGO

NR 740 STUDIA INFORMATICA NR 31 2012

RAFAŁ GOŁĄB, ŁUKASZ MADEJ
Akademia Górniczo-Hutnicza w Krakowie

OPRACOWANIE WYDAJNEGO FRAMEWORKU DO METODY AUTOMATÓW KOMÓRKOWYCH Z WYKORZYSTANIEM OBLICZEŃ RÓWNOLEGŁYCH I ROZPROSZONYCH

Streszczenie

W ostatnich latach powstało bardzo dużo modeli do przewidywania rozwoju mikrostruktury w materiałach metalicznych podczas przeróbki plastycznej, bazujących na metodzie automatów komórkowych. Rozwój modeli materiałów jest nierozerwalnie związany ze wzrostem wydajności obliczeniowej współczesnych komputerów, w szczególności w zakresie nowoczesnych architektur równoległych i rozproszonych. To pociąga za sobą zapotrzebowanie przemysłu na coraz wydajniejsze i dokładniejsze modele rozwoju mikrostruktury. Równocześnie ze względu na duży stopień złożoności numerycznej takich modeli konieczne jest opracowanie uniwersalnego i efektywnego narzędzia wspomagającego tworzenie bardzo skomplikowanych modeli i dającego jednocześnie możliwość ich testowania. W niniejszym artykule przedstawiono jedno z możliwych rozwiązań, jakim jest framework dedykowany do obliczeń przy użyciu metody automatów komórkowych na komputerach z pamięcią wspólną i rozproszoną. Przedstawione rozwiązanie oparto na technologii C++ i bibliotece przesyłania komunikatów MPI (Message Passing Interface).

Główny nacisk podczas pracy położono na opracowanie metodologii podziału przestrzeni automatów komórkowych na architektury równoległe i rozproszone w celu pełnego wykorzystania mocy obliczeniowej dostępnych komputerów.

Wprowadzenie

Każde przedsiębiorstwo przemysłowe dąży do opracowania takiej technologii, aby jego wyroby były konkurencyjne na rynku. Jednak opracowanie nowych technologii przy użyciu prób przemysłowych jest bardzo kosztowne i czasochłonne, co powoduje zbyt wolne dostosowywanie się przedsiębiorstw do wymagań szybko zmieniającego się rynku. Rozwiązaniem tego problemu mogą być obliczenia numeryczne i symulacje komputerowe, dzięki którym można precyzyjnie modelować zmiany na poziomie mikrostruktury, a zatem zastąpić realizacje rzeczywistych prób laboratoryjnych / przemysłowych. To powoduje możliwość zaoszczędzenia cennego czasu i pieniędzy. Jednak aby skorzystać z zalet komputerowego projektowania technologii, konieczny jest dostęp do odpowiedniego sprzętu i oprogramowania.

Ciągły rozwój sprzętu komputerowego i nowych technologii pozwala na powstawanie coraz bardziej złożonego oprogramowania. Jednak ze względu na dużą złożoność niektórych zjawisk fizycznych czas obliczeń na komputerach osobistych nieustannie wzrasta. W większości przypadków czas obliczeń można w dużym stopniu zredukować – dzięki zastosowaniu odpowiednich architektur komputerowych (takich jak architektury równoległe czy rozproszone), a także przy użyciu odpowiednio stworzonego oprogramowania. W tym przypadku umiejętne wykorzystanie technik programowania staje się kluczowym elementem do osiągnięcia zamierzonego celu, czyli zredukowania czasu obliczeń.

W ciągu ostatnich lat powstało bardzo dużo modeli rozwoju mikrostruktury w materiałach metalicznych bazujących na metodzie automatów komórkowych[1]. Jednakże opracowywane modele rozwoju mikrostruktury bazujące na

[1] S. Das, E.J. Palmiere, I.C. Howard, *CAFE: a tool for modelling thermomechanical processes*, Proc. Thermomech. Processing: Mechanics, Microstructure & Control, red. E.J. Palmiere, M. Mahfouf, C. Pinna, Sheffield 2002, s. 296–301; C.H.J. Davies, *Growth of nuclei in a cellular automaton simulation of recrystalization*, „Scripta Materialia", 36, 1997, s. 35–40; J. Gawąd, M. Pietrzyk, *Application of CAFE coupled model to description of microstructure development*

metodzie automatów komórkowych były dostosowane do rozwiązywania konkretnego problemu, np. symulacji rekrystalizacji statycznej lub dynamicznej. Jednak w przypadku tworzenia nowego modelu, np. symulującego przemianę fazową austenit – ferryt, pracę od strony algorytmicznej i implementacyjnej należało rozpoczynać od nowa, co znacząco wydłużało czas tworzenia modelu. Aby rozwiązać ten problem, autorzy zdecydowali się opracować uniwersalny framework dla metody automatów komórkowych, w którym podstawowe rozwiązania algorytmiczne automatów komórkowych są dostępne dla każdego użytkownika. Dzięki temu główny ciężar pracy skupia się na opracowywaniu właściwych reguł tranzycji odwzorowujących mechanizmy mikrostrukturalne, a nie na aspektach implementacyjnych. W rezultacie wykorzystania takiego frameworku, opracowanie modelu numerycznego będzie możliwe dla naukowców nieposiadających rozległej wiedzy z zakresu programowania i rozwiązań algorytmicznych. Dodatkowo w celu zwiększenia wydajności obliczeń z wykorzystaniem wymienionego wyżej rozwiązania możliwe jest skorzystanie ze wspomnianych zalet oferowanych przez architektury rozproszone i równoległe. W rezultacie powstanie efektywne narzędzie do tworzenia modeli rozwoju mikrostruktury, ukrywające warstwę implementacyjną przed użytkownikiem końcowym.

Głównym celem niniejszej pracy jest opracowanie metodologii podziału przestrzeni automatów komórkowych w opracowywanym frameworku tak, aby umożliwić efektywne wykorzystanie zalet wynikających z obliczeń równoległych.

1. Metoda automatów komórkowych

Automat komórkowy, AK (ang. Cellular Automaton, CA) jest dyskretnym modelem matematycznym składającym się z regularnej przestrzeni komórek, z których każda może przyjąć jeden ze skończonej liczby stanów. Idea automatów komórkowych znajduje swój początek w pracach von Neumanna,

during dynamic recrystallization, „Archives of Metallurgy and Materials", 52, 2007, s. 257–266; Ł. Madej, P.D. Hodgson, M. Pietrzyk, *Development of the multi-scale analysis model to simulate strain localization occurring during material processing*, „Archives of Computational Methods in Engineering", 16, 2009, 287–318.

dotyczących automatów zdolnych do samoreplikacji[2]. Późniejsze rozwinięcie tej koncepcji znalazło zastosowania w symulacji i modelowaniu systemów dyskretnych[3]. W przeciwieństwie do numerycznych rozwiązań równań różniczkowych, automaty komórkowe służą do modelowania systemów, których stan opisywany jest za pomocą wartości dyskretnych. Kolejną zasadniczą różnicą jest metodyka konstruowania modelu: celem AK nie jest opisanie skomplikowanego systemu za pomocą skomplikowanych równań, ale opisanie systemu za pomocą zestawu relatywnie prostych reguł i interakcji pomiędzy składowymi tego systemu (komórkami). Automat komórkowy w większości zastosowań zbudowany jest z następujących elementów:

– siatki, zazwyczaj identycznych komórek, których stan może być opisywany zarówno za pomocą zmiennych dyskretnych, jak i ciągłych,
– relacji sąsiedztwa, określającej logiczne powiązanie pomiędzy komórkami,
– zestawu reguł, za pomocą których realizowana jest tranzycja pomiędzy stanami komórek.

Najistotniejszą częścią automatu komórkowego jest wspomniany wyżej zestaw reguł tranzycji. Za jego pomocą realizowana jest dynamika systemu. Zmiana stanu komórki dokonywana jest na podstawie stanu komórki oraz stanu komórek należących do sąsiedztwa z poprzedniego kroku czasowego. Możliwe jest także uwzględnienie stochastycznych własności systemu, głównie poprzez wprowadzanie niedeterministycznych reguł tranzycji i definicji sąsiedztwa. W związku z tym tego typu modele znalazły szerokie zastosowanie w zagadnieniach modelowania rozwoju mikrostruktury w materiałach metalicznych[4].

Jednakże, tak jak wcześniej wspomniano, każdy nowo powstały model rozwoju mikrostruktury, oparty na metodzie automatów komórkowych, aby mógł znaleźć praktyczne zastosowanie, musiał zostać zaimplementowany od podstaw. To wymagało od autora wiedzy nie tylko z dziedziny jego nauki,

[2] J. Von Neumann, *Theory of self reproducing automata*, red. A.W. Bamk, University of Illinois, Urbana 1966.

[3] S. Wolfram, *Statistical mechanics of cellular automata*, „Reviews of Modern Physics", 55 , 1983, s. 601–644.

[4] S. Das, E.J. Palmiere, I.C. Howard, dz. cyt.; C.H.J. Davies, dz. cyt., J. Gawąd, M. Pietrzyk, dz. cyt.; Ł. Madej, P.D. Hodgson, M. Pietrzyk, dz. cyt.

np. metalurgii czy inżynierii materiałowej, ale także specjalistycznej wiedzy informatycznej z zakresu programowania.

Z uwagi na fakt, że wszystkie modele oparte na metodzie automatów komórkowych mają ze sobą pewne części wspólne, takie jak przestrzeń, typ sąsiedztwa, ściśle zdefiniowany stan, warunki brzegowe czy reguły przejścia, możliwa jest ich enkapsulacja w postaci uniwersalnego oprogramowania – frameworku[5]. Taka aplikacja umożliwia w prosty sposób definiowanie parametrów wspólnych dla każdego automatu, a także własnych reguł przejścia, odciążając użytkownika od zagadnień implementacyjnych. Takie podejście umożliwia tworzenie modeli rozwoju mikrostruktury szerokiemu gronu odbiorców.

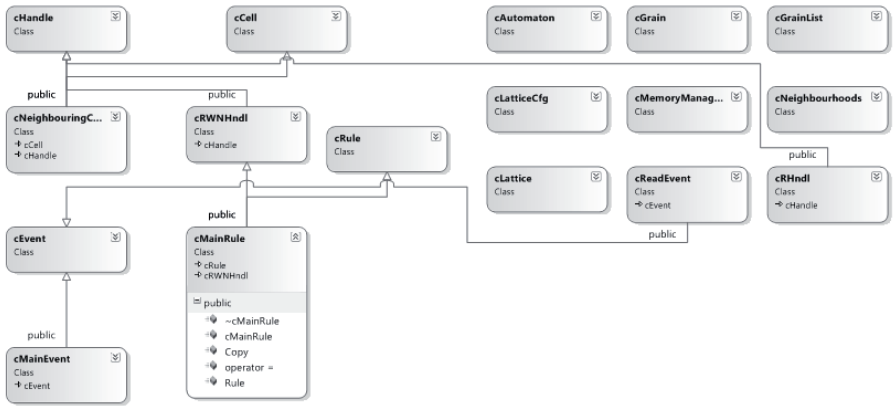
2. Framework CA

Podstawowe założenia projektowe dotyczące frameworku CA to łatwość obsługi, sprawne i wydajne działanie, a także możliwość budowania złożonych modeli fizycznych. Jeżeli chodzi o łatwość obsługi, na obecnym etapie aplikacja bazuje na wykorzystaniu zdefiniowanego interfejsu programistycznego (ang. Application Programming Interface, API), co pozwala na szybkie tworzenie gotowych aplikacji i ponowne wykorzystanie kodu. Jednak docelowo framework będzie umożliwiać tworzenie modeli za pomocą pseudojęzyka – w łatwy sposób zrozumiałego dla technologów nieposiadających wiedzy na temat języków programowania.

Framework CA zrealizowany został przy wykorzystaniu technologii obiektowej (rys. 1), tak aby istniała możliwość udostępnienia go w formie dynamicznie linkowanej biblioteki dla innych programistów. Dzięki takiemu rozwiązaniu framework CA stanowi narzędzie służące do ujednoliconego definiowania nowych algorytmów na podstawie metody automatów komórkowych. Podstawowe klasy zaimplementowane we frameworku obejmują obsługę: przestrzeni (cLattice), sąsiedztwa (cNeighbourhoods), komórek (cCell) oraz reguł przejścia (cRule). Podczas projektowania frameworku szczególny nacisk został położony na algorytmy związane z symulacją zjawisk mikrostrukturalnych,

[5] Ł. Rauch, Ł. Madej, P. Spytkowski, *Development of the cellular automata framework for modelling of material microstructures*, CMS'09 Computer Method and Systems, Kraków 2009, s. 446.

co zapoczątkowało powstanie dodatkowej klasy obejmującej operacje na ziarnach (cGrain).



Rys. 1. Diagram klas obejmujący podstawową funkcjonalność frameworku CA

Działanie frameworku opiera się na dwóch głównych klasach cEvent oraz cRule, odpowiedzialnych za operacje na zdarzeniach oraz regułach tranzycji. Podział na zdarzenia i reguły pozwala na stosowanie kilku reguł przejścia podczas jednego kroku czasowego automatu. Każde zdarzenie może posiadać kilka reguł przejścia. Np. zdarzenie rozrost ziarna będzie miało dwie reguły: zarodkowanie i rozrost.

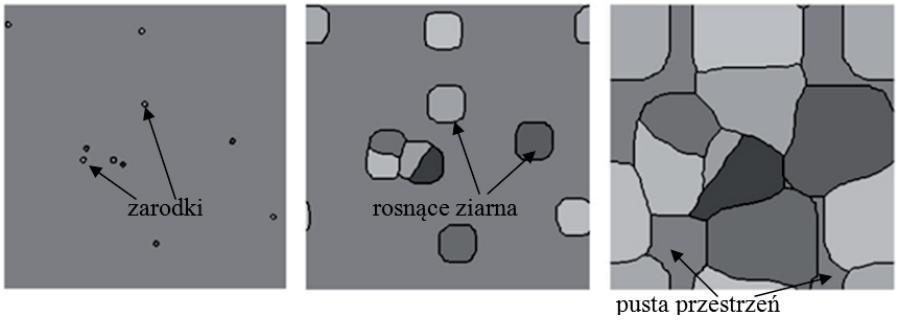
Tworzone algorytmy bazujące na automatach komórkowych mają niejednokrotnie bardzo dużą złożoność obliczeniową, przez co znacząco wydłużony zostaje czas ich wykonania. Dlatego też zaproponowano rozwiązanie pozwalające na zrównoleglenie problemu z wykorzystaniem maszyn wieloprocesorowych i procesorów wielordzeniowych. W tym celu zaprojektowano klasę cMemoryManager, która zajmuje się podziałem przestrzeni automatu na części, które następnie przekazywane są do odrębnych wątków obliczeniowych. Dodatkowo zaproponowana została klasa cMPI dedykowana obliczeniom na maszynach wielowęzłowych (klastrach obliczeniowych).

Dla uzyskania przyspieszenia obliczeń konieczne jest jednak precyzyjne określenie sposobu podziału analizowanej przestrzeni. Analiza tego zagadnienia jest głównym celem niniejszej pracy.

3. Metodologia podziału przestrzeni automatów komórkowych

W celu przeprowadzenia badań została stworzona aplikacja na bazie opracowanego frameworku przy użyciu technologii C++ i biblioteki przesyłania komunikatów MPI. Skorzystano z biblioteki MPI, ponieważ jest ona obecnie dominującym modelem wykorzystywanym w klastrach komputerów oraz w superkomputerach. Autorzy zdecydowali się również na skorzystanie z biblioteki MPI ze względu na fakt, iż udostępnia ona hermetyczny interfejs programistyczny, pozwalający na skupienie się na samej komunikacji, bez wnikania w szczegóły implementacji biblioteki i obsługi błędów.

Opracowany na potrzeby pracy testowy model swobodnego rozrostu ziaren jest schematycznie pokazany na rys. 2. Model ten zastosowano w celu wykazania wpływu sposobu podziału przestrzeni na wydajność realizowanych obliczeń. Program tworzy prosty automat komórkowy o zdefiniowanej przez użytkownika przestrzeni 3D. Każda komórka automatu ma dwie zmienne wewnętrzne – jedną id-ziarna typu int i jedną id-test typu double. Program na początku nadaje centralnej komórce wartość int równą 1, a także dla tej komórki losową wartość typu double. Na tym etapie wartości typu double są dodane tylko w celu analizy możliwości przesyłu różnych struktur danych pomiędzy wykorzystywanymi do obliczeń maszynami. Natomiast zmienne typu int w komórce oznaczają numery ziaren, do których dana komórka przynależy.



Rys. 2. Schemat obrazujący regułę przejścia dla zastosowanego modelu rozrostu ziaren

Rozrost ziaren w automacie następuje na podstawie zdefiniowanej reguły przejścia, która zakłada, że komórka, która ma wartość id-ziarna równą 0 zmienia numer ziarna do którego przynależy, jeżeli w sąsiedztwie ma komórki różne – o id-ziarna różnym od 0. W przypadku wystąpienia kilku różnych wartości zmiennej id-ziarna w sąsiedztwie badanej komórki algorytm wybiera tę wartość, której w sąsiedztwie jest najwięcej. Jeśli zaistnieje sytuacja, że w sąsiedztwie jest identyczna liczba sąsiadów o innych wartościach id-ziarna, to analizowana komórka w sposób losowy wybiera id-ziarna spośród dostępnych. Opracowaną regułę przejścia dotyczącą rozrostu można przedstawić w następującej formie:

$$Y_{n,m}\left(t_{i+1}\right)=\begin{cases} id_{ziarna} \quad dla \quad G\left[id_{ziarna}\right]_{\max}\left(t_i\right) \quad i \quad Y_{n,m}\left(t_i\right)=0 \\ Y_{n,m}\left(t_i\right) \quad dla \quad Y_{n,m}\left(t_i\right)\neq 0 \end{cases}$$

gdzie:

$Y_{n,m}(t_{i+1})$ – stan komórki $n\times m$ w przestrzeni dla kolejnego kroku czasowego,

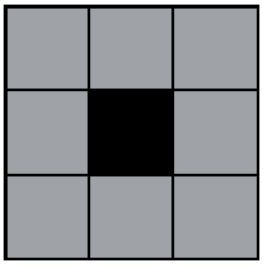
$Y_{n,m}(t_i)$ – stan komórki $n\times m$ w przestrzeni dla badanego kroku czasowego,

id_{ziarna} – numer ziarna,

G – tablica z ilością sąsiadów danego typu,

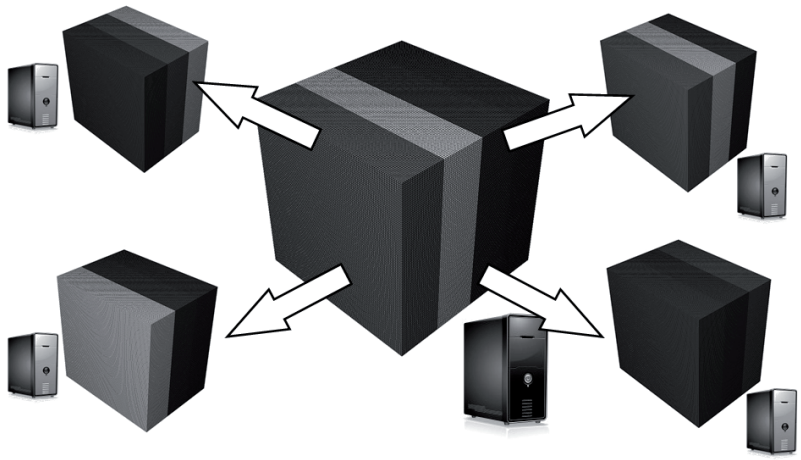
$G[id_{ziarna}]_{\max}(t_i)$ – największa suma sąsiadów z takim samym id_{ziarna} dla badanego kroku czasowego.

Na potrzeby testów zaimplementowano sąsiedztwo Moore'a o 8 sąsiadach, które dla przestrzeni dwuwymiarowej pokazano na rys. 3. W trakcie obliczeń zastosowano również periodyczne warunki brzegowe, aby zapewnić ciągłość przestrzeni obliczeniowej.



Rys. 3. Sąsiedztwo Moore'a dla przestrzeni dwuwymiarowej

Opracowane rozwiązanie przetestowano na różnej ilości węzłów obliczeniowych. Tak jak wspomniano, istotnym aspektem jest dokonanie właściwego podziału przestrzeni AK pomiędzy kolejne węzły. W ramach niniejszej pracy analizie poddano podział przestrzeni na wzajemnie równoległe warstwy, co schematycznie przestawiono na rys. 4. W tym schemacie w przypadku pracy z przestrzenią 100×100×100 komórek i przy założeniu dostępności np. 5 węzłów program podzieli przestrzeń na 4 równe warstwy, czyli każdy z 4 węzłów liczących otrzyma do obliczeń przestrzeń 100×100×27. Wymiar 27 związany jest bezpośrednio z definicją metody automatów komórkowych, gdzie każda z komórek musi mieć dostęp do swojego bezpośredniego sąsiedztwa (rys. 4). W analizowanym przykładzie węzeł nr 5 pełni funkcję mastera, który zajmuje się dystrybucją danych do poszczególnych węzłów, jak również ich aglomeracją na koniec każdego kroku obliczeniowego. Warunkiem zatrzymania działania automatu jest pełne rozrośnięcie się mikrostruktury w analizowanym obszarze AK.



Rys. 4. Sposób podziału przestrzeni na równe warstwy pomiędzy węzły liczące

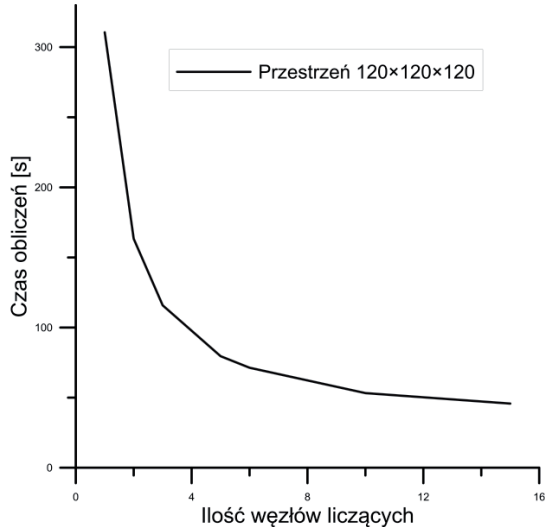
Dla oceny wydajności zaproponowanego sposobu podziału przestrzeni AK przeprowadzono serię testów obejmujących 7 rozmiarów przestrzeni od 30×30×30 do 210×210×210 komórek. Analizę przeprowadzono dla maksymalnej liczby węzłów równej 15. Tabelaryczne zestawienie wyników badań dla różnej ilości węzłów i różnych rozmiarów przestrzeni automatu komórkowego

przedstawiono w tabeli 1. Dodatkowo dla załączonych wyników tabelarycznych przygotowano wykresy zależności czasu obliczeń od ilości węzłów, a także wykresy zależności funkcji speed up od ilości węzłów liczących. Przykłady obu wykresów przedstawiono dla przestrzeni 180×180×180 na rys. 5 i rys. 6. Z wykresu pokazanego na rys. 5 wyraźnie widać, że wraz ze wzrostem ilości węzłów liczących czas obliczeń ulega skróceniu, co przekłada się bezpośrednio na przyspieszenie obliczeń. Na rys. 6 linią (wyżej) oznaczony został idealny wzrost prędkości obliczeń (tzw. liniowy speed up). Następnie na rys. 7 przedstawiono wykres przyspieszenia obliczeń wraz ze wzrostem ilości węzłów liczących dla różnych rozmiarów przestrzeni automatu komórkowego.

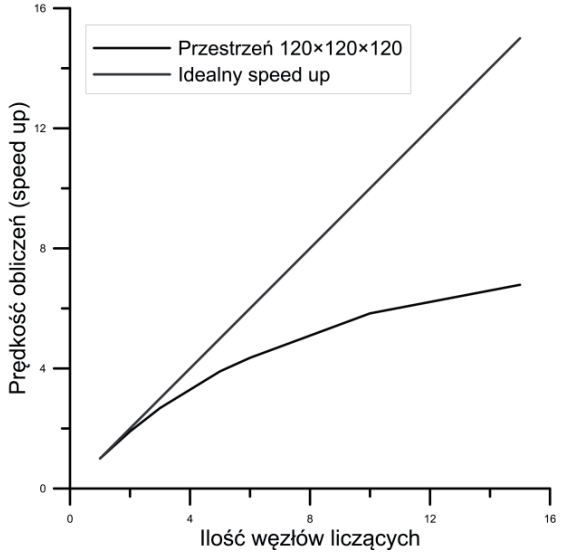
Tabela 1

Wyniki obliczeń dla różnych rozmiarów przestrzeni automatu komórkowego

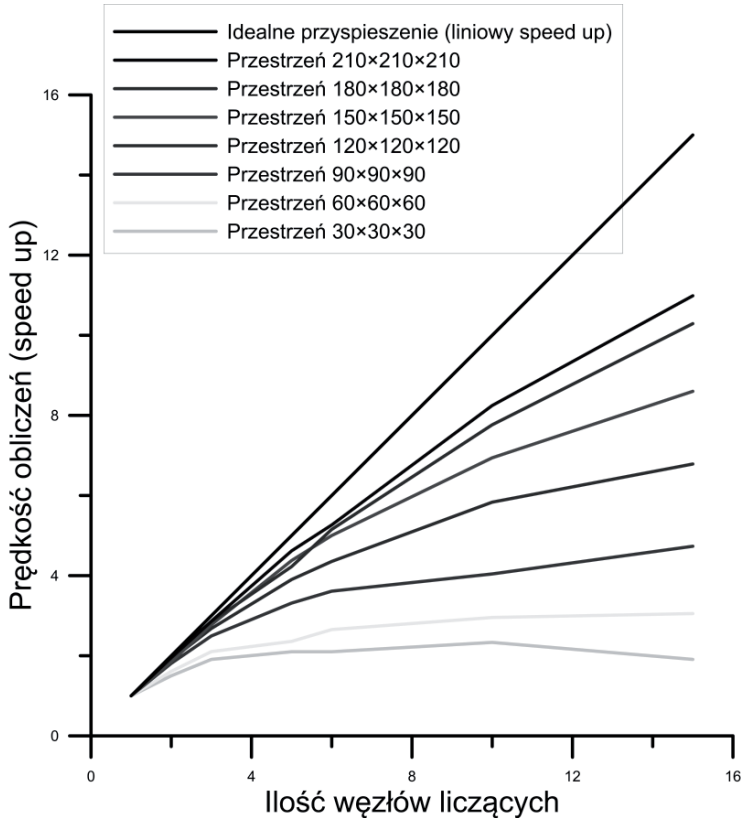
30×30×30			**60×60×60**			**90×90×90**		
Ilość węzłów	Czas [s]	SpeedUp	Ilość węzłów	Czas [s]	SpeedUp	Ilość węzłów	Czas [s]	SpeedUp
1	0.21	1.00	1	4.67	1.00	1	49.56	1.00
2	0.14	1.50	2	2.9	1.61	2	27.38	1.81
3	0.11	1.91	3	2.22	2.10	3	19.87	2.49
5	0.1	2.10	5	1.98	2.36	5	14.95	3.32
6	0.1	2.10	6	1.76	2.65	6	13.72	3.61
10	0.09	2.33	10	1.58	2.96	10	12.26	4.04
15	0.11	1.91	15	1.53	3.05	15	10.47	4.73
120×120×120			**150×150×150**			**180×180×180**		
Ilość węzłów	Czas [s]	SpeedUp	Ilość węzłów	Czas [s]	SpeedUp	Ilość węzłów	Czas [s]	SpeedUp
1	310.56	1.00	1	1371.14	1.00	1	4742.87	1.00
2	163.33	1.90	2	740.03	1.85	2	2443.8	1.94
3	115.86	2.68	3	494.39	2.77	3	1664.61	2.85
5	79.56	3.90	5	313.24	4.38	5	1122.47	4.23
6	71.32	4.35	6	274.14	5.00	6	920.18	5.15
10	53.23	5.83	10	197.53	6.94	10	610.74	7.77
15	45.76	6.79	15	159.41	8.60	15	460.89	10.29
210x210x210			**300x300x300**					
Ilość węzłów	Czas [s]	SpeedUp	Ilość węzłów	Czas [s]	SpeedUp			
1	13714.2	1.00	1	153170	1.00			
2	7009.03	1.96	2	57232.9	2.68			
3	4787.48	2.86	3	38752.3	3.95			
5	2969.5	4.62	5	24408.4	6.28			
6	2604.54	5.27	6	21165.8	7.24			
10	1662.89	8.25	10	13372.8	11.45			
15	1248.25	10.99	15	9704.27	15.78			



Rys. 5. Wykres zależności czasu obliczeń od ilości węzłów liczących dla przestrzeni 120×120×120



Rys. 6. Zależność przyspieszenia obliczeń od ilości węzłów liczących dla przestrzeni 120×120×120



Rys. 7. Wykres zależności speed up od ilości węzłów liczących dla różnych rozmiarów przestrzeni automatu komórkowego

Z rys. 7 wynika, że im większa przestrzeń automatu komórkowego, tym uzyskiwane jest większe przyspieszenie obliczeń. Widać także, że dla relatywnie małej przestrzeni, np. 30×30×30, można zaobserwować spadek predkości obliczeń dla dużej ilości węzłów liczących. Fakt ten związany jest z komunikacją pomiędzy węzłami (dłuższy jest czas przesyłu danych niż czas obliczeń).

Tak jak przedstawiono, wykorzystanie obliczeń wieloprocesorowych z zastosowanym równoległym sposobem podziału przestrzeni AK jest efektywnym sposobem przyspieszenia czasochłonnych modeli.

Podsumowanie

Istnienie dużej ilości skomplikowanych modeli rozwoju mikrostruktury, a także ciągły rozwój nowych technologii i sprzętu pozwoliły autorom pracy na zbudowanie uniwersalnego narzędzia wspomagającego pracę technologów przy projektowaniu nowych technologii. Przeprowadzone badania podziału przestrzeni automatów komórkowych ukazują możliwości optymalizowania czasu obliczeń, a także szerokie pole do dalszych badań. Wykazano, że w przypadku modelu swobodnego rozrostu ziaren im większa przestrzeń automatu komórkowego, tym uzyskiwane jest większe przyspieszenie obliczeń. Równocześnie przy małych rozmiarach analizowanej przestrzeni może nastąpić spadek prędkości obliczeń zwiazany z uwarunkowaniami komunikacji pomiędzy węzłami.

Praca naukowa finansowana w ramach projektu NCN 2011/01/D/ST8/01681.

Literatura

1. Das S., Palmiere E.J., Howard I.C., *CAFE: a tool for modelling thermomechanical processes*, Proc. Thermomech. Processing: Mechanics, Microstructure & Control, red. E.J. Palmiere, M. Mahfouf, C. Pinna, Sheffield 2002.
2. Davies C.H.J., *Growth of nuclei in a cellular automaton simulation of recrystalization*, „Scripta Materialia", 36, 1997.
3. Gawąd J., Pietrzyk M., *Application of CAFE coupled model to description of microstructure development during dynamic recrystallization*, „Archives of Metallurgy and Materials", 52, 2007.
4. Madej Ł., Hodgson P.D., Pietrzyk M., *Development of the multi-scale analysis model to simulate strain localization occurring during material processing*, „Archives of Computational Methods in Engineering", 16, 2009.
5. Rauch Ł., Madej Ł., Spytkowski P., *Development of the cellular automata framework for modelling of material microstructures*, CMS'09 Computer Method and Systems, Kraków 2009.
6. Von Neumann J., *Theory of self reproducing automata*, red. A.W. Bamk, University of Illinois, Urbana 1966.
7. Wolfram S., *Statistical mechanics of cellular automata*, „Reviews of Modern Physics", 55, 1983.

DEVELOPMENT OF EFFECTIVE FRAMEWORK DEDICATED TO CELLULAR AUTOMATA METHOD WITH PARALLEL AND DISTRIBUTED COMPUTATION

Summary

In recent years, a lot of models based on the cellular automata method were developed to predict the microstructure evolution in material during metal forming operations. Development of material models is inextricably linked with increased computational performance of modern computers, particularly in the area of parallel and distributed architectures. This entails with the industry's demand for more efficient and more accurate models of microstructure evolution. At the same time due to the high degree of numerical complexity of such models it is necessary to develop a universal and effective tool to support the creation of those complex models. One possible solution to this problem that is investigated within the present work is a framework dedicated to calculations using the method of cellular automata on computers with shared memory and distributed memory. The solution is based on C++ technology and message-passing library MPI (Message Passing Interface). The main emphasis was put in the present work on development of a methodology for dividing the space of cellular automata on parallel and distributed architectures to fully use the computing power of available computers.

Translated by Rafał Gołąb