

prof. dr hab. inż. Tadeusz Czachórski
Instytut Informatyki Teoretycznej i Stosowanej PAN
ul. Bałtycka 5, 44-100 Gliwice

Gliwice, 15 maj 2015 r.

Recenzja pracy doktorskiej mgr Joanny Potiopy

**p.t. „Algorytmy rozwiązywania wybranych typów układów równań liniowych
o macierzach wąskopasmowych na komputerach z procesorami wielordzeniowymi”.**

Kontekst rozprawy

Powstanie architektur umożliwiających efektywne zrównoleglenie obliczeń przy zachowaniu relatywnie niskiej ceny stosowanego sprzętu pozwala tworzyć coraz bardziej złożone modele, które wykorzystują coraz większe zbiory danych pomiarowych. Powstanie kart GPU, współpracujących z nimi środowisk typu CUDA i dostosowanie tych kart do obliczeń w podwójnej dokładności, jak również rozwój systemów heterogenicznych zawierających wielordzeniowe procesory CPU i wielordzeniowe karty GPU ma w tym procesie szczególne znaczenie i stymuluje prace nad metodami numerycznymi, które potrafią dobrze wykorzystać stworzone przez nowe architektury możliwości.

Praca dotyczy jednego z trudnych problemów tej dziedziny, jakim jest rozwiązywanie układów równań liniowych o macierzach wąskopasmowych, występujących m.in. przy rozwiązywaniu równań różniczkowych zwyczajnych, numerycznym wyznaczaniu interpolujących funkcji sklepanych i generacji ciągów liczb pseudolosowych. Stosowane w tym problemie algorytmy sekwencyjne trudno dostatecznie zrównoleglić, by wykorzystać potencjał architektur wieloprocesorowych i wielordzeniowych.

Omawiana praca proponuje wykorzystać w rozwiązywaniu wspomnianych równań liniowych metodę *dziel i zwyciężaj* – znaną metodę projektowania algorytmów w informatyce, często prowadzącą do bardzo efektywnych rozwiązań, np. dla sortowania przez scalanie (mergesort), sortowania szybkiego (quicksort), wyszukiwania binarnego, algorytmu Cooleya-Tukeya szybkiej transformacji Fouriera, czy graficznego algorytmu Warnocka. Zasadą tej metody jest rekurencyjny podział na dwa lub więcej mniejszych podproblemów tego samego lub podobnego typu tak długo, aż fragmenty staną się wystarczająco proste do bezpośredniego rozwiązania. Następnie rozwiązania otrzymane dla podproblemów scala się, uzyskując rozwiązanie całego zadania.

Teza pracy brzmi: *Efektywne rozwiązywanie układów równań liniowych o macierzach wąskopasmowych na współczesnych komputerach wymaga zastosowania specjalnych algorytmów typu dziel i zwyciężaj, które niosą w sobie potencjalną równoległość obliczeń.*

Słuszność tej tezy jest pokazana w pracy poprzez sformułowanie algorytmów dla wybranych układów równań liniowych o macierzach wąskopasmowych i ich implementację biorąc pod uwagę właściwy dobór struktur danych i sposób dostępu do pamięci oraz eksperymentalne zbadanie efektywności w wybranych zagadnieniach obliczeniowych. Badania dotyczą również wpływu podwyższonej precyzji obliczeń na dokładność rozwiązania. Układ pracy jest następujący.

Struktura rozprawy

Praca składa się z 6 rozdziałów i liczy 115 stron.

Rozdział 1 wprowadza podstawowe pojęcia wykorzystywane w pracy (typy macierzy, operacje na macierzach, macierze specjalne) i opisuje wybrane problemy prowadzące do rozwiązywania układów równań liniowych o macierzach wąskopasmowych, wśród nich interpolację funkcjami sklejanymi (numeryczne wyznaczanie funkcji sklepanych sześciennymi, wybór warunków brzegowych), zagadnienie brzegowe, generatory liczb pseudolosowych, w tym generatory bazujące na rekursji.

Rozdział 2 jest przeglądem współczesnych architektur wielordzeniowych, przedstawia architektury klastrów i procesorów GPU wykorzystywane w chwili obecnej do obliczeń o dużej wydajności. Omawia też biblioteki opracowane dla programowania problemów algebry liniowej: oprogramowanie numeryczne według standardu BLAS (Basic Linear Algebra Subprograms) oraz stanowiącą jej nadbudowę bibliotekę LAPACK (Linear Algebra Package) i jej wersje, jak również standardy przesyłu komunikatów w pamięci rozproszonej (np. PVM – Parallel Virtual Machine, MPI – Message Passing Interface), mające duży wpływ na efektywność obliczeń w algorytmach równoległych. Podane są też informacje o specjalistycznej bibliotece BLACS (Basic Linear Algebra Communication Subroutines) wykorzystującej wymienione standardy przesyłu komunikatów. W przypadku komputerów równoległych odpowiednikami bibliotek BLAS i LAPACK są omówione następnie PBLAS (Parallel BLAS) i ScaLAPACK (Scalable LAPACK).

Pojawienie się i wzrost wykorzystania wielordzeniowych architektur spowodowały opracowanie nowych wersji, też omówionych w tym rozdziale, bibliotek: PLASMA (Parallel Linear Algebra Software for Multi-core Architectures) i przeznaczonej dla heterogenicznych systemów hybrydowych (CPU + GPU) MAGMA (Matrix Algebra on GPU and Multicore Architectures). Istnieją też biblioteki dostarczane przez twórców nowych technologii – firma NVIDIA oferuje zestaw narzędzi CUDA Toolkit, a w nim bibliotekę CURAND algorytmów dla generowania liczb pseudolosowych i ciągów quasi-losowych.

Następne rozdziały dotyczą już oryginalnych wyników pracy.

Rozdział 3 zawiera opis oprogramowania równoległego dla problemu tworzenia funkcji ciągłej poprzez interpolację funkcjami sklejanymi sześciennymi. Znając wartości funkcji w węzłach, w każdym z przedziałów między węzłami musimy wyznaczyć wielomian trzeciego stopnia, tak aby funkcja powstała przez sklejenie tych wielomianów była w węzłach ciągła i miała ciągłą pierwszą i drugą pochodną. Problem wyznaczania współczynników tych wielomianów prowadzi do rozwiązania układu równań o macierzy trójdzielnej, które zwykle rozwiązuje się metodą eliminacji Gaussa

lub rozkładem LU. W pracy podano algorytm sekwencyjny wykorzystujący podprogram z biblioteki LAPACK, a następnie przedstawiono jego równoległą wersję, wykorzystującą podprogram z biblioteki ScaLAPACK. Obie metody, sekwencyjną i równoległą, implemowano w Fortranie 95 i poddano testom wydajnościowym na dwu platformach: klastrze 20 procesorów oraz superkomputerze z procesorami wektorowymi Cray X1, prowadząc obliczenia dla różnych rozmiarów problemu i różnej liczby procesorów, wyznaczono przyspieszenie metody równoległej w stosunku do sekwencyjnej. Testy zostały uzupełnione obliczeniami w nowszej architekturze klastra z procesorami Intel Xeon bez zmian algorytmu konkludując, że zmiana architektury nie miała wpływu na skalowalność algorytmu, który jest uniwersalny – skalowalny tak ze względu na rozmiar danych, jak liczbę węzłów.

Rozdział 4 korzysta z faktu, że rozwiązywanie wybranych typów równań różniczkowych można sprowadzić do rozwiązywania układu równań liniowych o macierzy trójdzielnej zbliżonej do macierzy Toeplitza (macierz mająca te same wartości na poszczególnych przekątnych) i pokazuje jak zastosować w tym przypadku metodę *dziel i zwyciężaj* i efektywnie ją implementować na procesorach wielordzeniowych. Testy były w tym przypadku prowadzone na dwu różnych zestawach komputerowych, wykorzystujących odpowiednio pierwszą architekturę CUDA i jej późniejszą wersję Fermi. Obliczenia wykonywano w pojedynczej i podwójnej precyzji, dokumentując czas obliczeń i uzyskaną dokładność.

Rozdział 5 pokazuje, jak można zaprojektować szybkie równoległe algorytmy dla generatorów liczb pseudolosowych: liniowego generatora kongruentnego (LCG) i generatora Fibonacciego (LFG). Nowe algorytmy wykorzystują podejście *dziel i zwyciężaj* i mogą być implementowane w systemach hybrydowych z akceleratorami GPU wykorzystującymi CUDA.

Zastosowane podejście polega na zrównolegleniu rekurencyjnych relacji dla LCG i LFG, dzięki czemu statystyczne własności równoległych generatorów są takie same, jak w przypadku generatorów sekwencyjnych i nie trzeba przeprowadzać od nowa testów statystycznych. Jeszcze raz problem sprowadza się do rozwiązania układu równań liniowych o macierzach wąskopasmowych. W implementacji algorytmów duże, dające się zrównoleglić zadania są wykonywane na GPU, a drobniejsze, sekwencyjne, są realizowane na CPU. Implementacja jest przenośna, będzie dobrze działać także na nowych architekturach GPU.

Testy przeprowadzone na komputerze z kartą graficzną GPU w architekturze Fermi pokazują dobre przyspieszenie tych algorytmów w stosunku do algorytmów sekwencyjnych.

Rozdział 6 to podsumowanie i wnioski. Autorka wskazuje też dalsze możliwości rozwijania tematu.

Ocena pracy

Do oryginalnych wyników pracy można zaliczyć

- sformułowanie równoległego algorytmu wyznaczania funkcji sklepanych sześciennych na kla-

stry komputerowe, jego implementacja i eksperymentalna ocena efektywności wykonana na kilku komputerach o różnej architekturze, podwierdzająca uniwersalność proponowanego rozwiązania.

- opracowanie metody numerycznego rozwiązywania pewnego typu równania różniczkowego zwyczajnego z warunkami brzegowymi, sprowadzającego zagadnienie do rozwiązywania układu równań liniowych o macierzach trójdzielnych, algorytm numerycznego rozwiązywania tego układu równań wykorzystujący metodę dziel i zwyciężaj, analiza złożoności obliczeniowej algorytmu, implementacja algorytmu w systemie GPU i środowisku CUDA biorący pod uwagę strukturę danych odpowiednią dla pamięci karty graficznej, testy efektywności (korzystne dla zaproponowanego algorytmu) w dwu architekturach CUDA, zbadanie dokładności obliczeń w przypadku pojedynczej, podwójnej i mieszanej precyzji,
- opracowanie opartego na podejściu *dziel i zwyciężaj* algorytmu równoległego generowania ciągów liczb pseudolosowych w oparciu o generatory LCG i LFG dla komputerów wyposażonych w karty graficzne, sprowadzającego problem do rozwiązywania układu równań liniowych o macierzy dwudzielnej typu Toeplitza lub dolnotrójkątnej diagonalnie rzadkiej, w zależności od generatora; implementacja i testowanie algorytmów.

Wyniki potwierdzają postawioną tezę o przydatności w badanych problemach strategii *dziel i zwyciężaj*. Autorka potrafiła bardzo fachowo zaprojektować eksperymenty, dobierając architekturę i oprogramowanie, pokazała duże doświadczenie i znajomość problemów dostosowania algorytmów i metod programowania oraz doboru istniejących bibliotek do wymagań złożonych architektur. Przeprowadzona duża liczba testów dostarczyła obszerny materiał, który został umiejętnie zinterpretowany. Uzyskane wyniki i wnioski są oryginalne. Dysertacja napisana jest jasno, Autorka potrafi zachować ścisłość wywodu, a jednocześnie przedstawić problem w zrozumiałych słowach. Wyniki są dobrze udokumentowane.

Wyniki zostały opublikowane w 3 artykułach przedstawionych na specjalistycznych konferencjach i czwartym, opublikowanym w *Journal of Computational Applied Mathematics* (jego 5-letni impact factor wynosi 1.245).

Oprogramowanie, wymagające dłuższej refleksji, często nie nadąża za możliwościami szybko rozwijającego się sprzętu, dlatego rezultaty uzyskane w pracy są interesujące i mają znaczenie praktyczne dla obliczeń numerycznych. Trzeba docenić dotychczasową pracę Doktorantki i zachęcać do dalszej, do której jest znakomicie przygotowana. Rozwój sprzętu nie stoi w miejscu, powstają nowe komputery z wieloma akceleratorami i węzły akcelеровane kartami GPU, dlatego rozwiązania dobre dzisiaj jutro będą wymagały dalszych prac, z czego Doktorantka zdaje sobie dobrze sprawę.

Już teraz można się zastanowić, czy uzyskane wyniki można przenieść do nowych istniejących architektur, takich jak Intel Many Integrated Core (MIC) Architecture, zwanej od 2012 r. Intel Xeon Phi, zawierającej około 60 procesorów core, dwa poziomy pamięci cache i pamięć graficzną,

wszystko połączone bardzo szybkim pierścieniem. Procesory Xeon Phi weszły w skład the Tianhe-2, uznanego w 2013 r. za najszybszy w świecie supercomputer i utrzymującego się do tej pory (po raz czwarty) na czele listy TOP500. Testy efektywności i studium programowalności tych procesorów [Fang, Jianbin; Sips, Henk; Zhang, Lilun; Xu, Chuanfu; Yonggang, Che; Varbanescu, Ana Lucia *Test-Driving Intel Xeon Phi* (2014)] stwierdzają, że uzyskanie wysokiej wydajności Xeon Phi wciąż wymaga pomocy programistów: *“Our experience indicates that a simple data structure and massive parallelism are critical for Xeon Phi to perform well. When compiler-driven parallelization and/or vectorization fails, programming Xeon Phi for performance can become very challenging”*. Autorzy piszą dalej, że badana platforma dostarcza deklarowaną siłę obliczeń, lecz programiści będą musieli znaleźć właściwą strategię zrównoleglania by zapewnić 2400 wątkom urządzeniowym dostateczne obciążenie i znaleźć odpowiednią równowagę pomiędzy podziałem danych i spójnymi odwołaniami do pamięci by zapewnić odpowiednio wysoką przepustowość komunikacji z pamięcią. Dodają, że bez dużego wysiłku ze strony programistów co do paralelizacji, analizy i optymalizacji kodu, liczba aplikacji efektywnie korzystających z możliwości sprzętu pozostanie niewielka.

Pamiętajmy, że znane są założenia drugiej generacji architektury MIC (Knights Landing), a trzecia generacja (Knights Hill) jest już w drodze, stwarzając nowe wyzwania dla programistów.

Podsumowanie:

Uważam, że rozprawa doktorska mgr Joanny Potiopy spełnia w zupełności warunki stawiane rozprawom doktorskim przez ustawę o stopniu i tytułach naukowych. Autorka umiejętnie stawia i rozwiązuje oryginalne i trudne problemy naukowe o dużym znaczeniu praktycznym. Wnioskuje o przyjęcie tej pracy jako rozprawy doktorskiej i dopuszczenie jej do publicznej obrony.

