

Dr hab inż. Marek Tudruj
ul. Raławicka 8
02-601 Warszawa
Profesor w Polsko-Japońskiej Wyższej Szkole
Technik Komputerowych

Warszawa, 24 stycznia 2009

Recenzja pracy doktorskiej Piotra Tronczyka pt: "Optymalizacja numerycznej analizy dużych układów liniowych równań algebraicznych metodami programowania równoległego"

Przedstawiona mi do recenzji praca doktorska zawiera 127 stron i składa się ze wstępu, 9 zasadniczych rozdziałów oraz podsumowania i bibliografii.

We wstępie nakreślono zakres niniejszej rozprawy na tle metod modelowania propagacji fali elektromagnetycznej, z których większość sprowadza się do rozwiązania układów liniowych równań algebraicznych.

W rozdziale 1 przedstawiono wprowadzenie do algorytmów rozwiązywania dużych układów równań liniowych, które można przedstawić w postaci macierzowej jako $\mathbf{K} \mathbf{x} = \mathbf{b}$, gdzie \mathbf{x} jest wektorem niewiadomych, \mathbf{K} jest kwadratową macierzą współczynników przy niewiadomych w równaniach układu, \mathbf{b} jest wektorem wyrazów wolnych w tych równaniach. Wprowadzenie to przedstawia podstawy teoretyczne dla problemów postawionych w pracy. Omówiono metody iteracyjne oraz metody bezpośrednie, wśród których przedstawiono algorytm wykorzystujący tzw. dekompozycję LU macierzy, stanowiący przedmiot badań wykonanych w niniejszej pracy. W wyniku dekompozycji LU macierz \mathbf{K} zostaje przedstawiona jako iloczyn macierzy \mathbf{L} oraz \mathbf{U} , zgodnie z równaniem $\mathbf{K} = \mathbf{L} \mathbf{U}$, gdzie macierz \mathbf{L} (od angielskiego Lower) jest tzw. macierzą trójkątną dolną, zawierającą jedynki na głównej przekątnej i wszystkie elementy powyżej tej przekątnej równe zero a macierz \mathbf{U} (od angielskiego Upper) - tzw. macierz trójkątną górną, w której wszystkie elementy poniżej głównej przekątnej są równe zero. Rozłożenie macierzy \mathbf{K} na \mathbf{L} oraz \mathbf{U} pozwala na zastąpienie równania $\mathbf{K} \mathbf{x} = \mathbf{b}$ przez dwa równania $\mathbf{L} \mathbf{y} = \mathbf{b}$ oraz $\mathbf{U} \mathbf{x} = \mathbf{y}$, gdzie \mathbf{y} jest wektorem zmiennych pomocniczych, które dają się rozwiązać metodą eliminacji zmiennych w bardzo prosty sposób, ze względu na to, że macierze \mathbf{L} oraz \mathbf{U} są trójkątne. Wyznaczenie macierzy \mathbf{L} oraz \mathbf{U} odbywa się również w stosunkowo prosty sposób, poprzez stopniowe wyliczanie elementów tych macierzy w odpowiedniej kolejności za pomocą prostych formuł arytmetycznych na współczynnikach macierzy \mathbf{K} i znanych już elementach macierzy \mathbf{L} oraz \mathbf{U} . W części tej przedstawiono metodę realizacji dekompozycji macierzy na \mathbf{L} i \mathbf{U} w oparciu o statyczna dekompozycje macierzy \mathbf{K} oraz zarys realizacji równoległej wykorzystującej statyczny przydział danych procesorom. Przedstawiono też zarys analizy błędów metody LU. Szybkie równoległe metody rozwiązywania układów równań liniowych metodą dekompozycji LU są bardzo użyteczne praktycznie dla przypadków, gdy mamy wielokrotnie rozwiązać dane układy równań liniowych dla tych samych zmiennych i ich współczynników lecz dla różnych wartości wyrazów wolnych. Tak więc podjęty w pracy problem ma duże znaczenie praktyczne.

Rozdział 2 zawiera wprowadzenie do metodologii obliczeń równoległych. Omówiono tam klasyfikację systemów komputerowych, charakterystykę obecnie

dostępnych systemów do obliczeń równoległych i rozproszonych oraz miary wydajności wykonania programów równoległych.

W rozdziale 4 przedstawiono metody realizacji dekompozycji LU w systemie wieloprocessorowym typu "master-slave" ze stałą wielkością bloków macierzy \mathbf{K} przydzielanych procesorom. Metody te podzielono na metody ze statycznym przydziałem danych do procesorów liczących przed rozpoczęciem obliczeń i metody obliczeń ze stopniowym dynamicznym przydzielaniem danych o stałej wielkości.

W rozdziale 5, stanowiącym najbardziej istotną część rozprawy, przedstawiono zaproponowane algorytmy sterowania przebiegiem dynamicznej równoległej wersji dekompozycji macierzy LU. Wyróżniono tu 3 ogólne wersje strategii realizacji sterowania w systemie "master-slave": scentralizowana strategia z nadzorem stanu obciążeń i ich wyrównywania przez procesor "master", zdecentralizowana strategia z rozgłaszaniem przez procesory "slave" informacji o niedociążeniu oraz scentralizowana strategia z dynamiczną optymalizacją przez procesor "master" rozmiarów bloków danych przydzielanych procesorom liczącym w funkcji parametrów wydajnościowych sprzętu wykonawczego (moc obliczeniowa procesorów oraz przepustowość łączy komunikacyjnych). Parametry te wraz z grafem systemu wykonawczego są generowane przed rozpoczęciem obliczeń metodą wykonania obliczeń na wzorcowym układzie równań. Dalsze badania przeprowadzone w rozprawie skupiają się na tej trzeciej strategii. Zasadniczym elementem obliczeń jest transformacja macierzy współczynników przy niewiadomych \mathbf{K} na dwie podmacierze \mathbf{L} i \mathbf{U} . Obliczenia zorganizowane są w ten sposób, że kolejne elementy w macierzy \mathbf{K} są stopniowo transformowane na elementy podmacierzy \mathbf{L} oraz \mathbf{U} i zastępują elementy macierzy \mathbf{K} . Transformacja LU macierzy \mathbf{K} odbywa się w cyklach obliczeń wykonywanych równoległe przez zespół procesorów „slave”, sterowanych przez procesor „master”, który dynamicznie określa wielkość danych rozdzielanych między procesory „slave”. W każdym cyklu procesory wykonawcze otrzymują do przetworzenia podmacierze macierzy \mathbf{K} stanowiące warstwy zawierające liczbę wierszy dopasowaną do mocy obliczeniowej procesora i przepustowości jego łączy komunikacyjnych. W danym cyklu procesor przetwarza sekwencyjnie kolejne wiersze przydzielonej podmacierzy.

Dla wyznaczenia optymalnego rozmiaru danych dla każdego procesora, który ma uczestniczyć w transformacji LU podmacierzy \mathbf{K} doktorant zaproponował krokowy cykliczny heurystyczny algorytm, wyznaczający analitycznie takie rozmiary podmacierzy przetwarzanych równoległe przez procesory w każdego kroku, które zapewnią zbliżone czasy wykonania kroku dla wszystkich współpracujących procesorów. Jednocześnie doktorant założył zbliżony do paradygmatu BSP (Bulk-Synchronous-Parallel, L. Valiant) model algorytmu równoległego, polegający na tym, że w trakcie wykonywania kroku obliczeniowego nie ma komunikacji między równoległe pracującymi procesorami - wszelka niezbędna komunikacja związana z przekazaniem dotychczas wyznaczonych elementów macierzy \mathbf{L} oraz \mathbf{U} odbywa się równocześnie między krokami algorytmu. Zaproponowany algorytm daje pełne zrównoważenie obciążeń współpracujących procesorów i eliminuje straty wydajności procesorów związane z oczekiwaniem na wzajemną asynchroniczną wymianę danych. Dla zbudowania krokowej formy algorytmu dekompozycji LU, wykorzystuje się własność postawionego problemu dekompozycji LU, która polega na tym, że pierwsza (górna) podmacierz macierzy \mathbf{K} , licząc od elementu o najmniejszych współczynnikach na przekątnej, może być stopniowo całkowicie przetransformowana na elementy macierzy \mathbf{L} i \mathbf{U} przez jeden procesor, używając wyłącznie danych z tej

samej podmacierzy. Ta własność zachodzi dla kolejnych niższych warstw transformowanej macierzy pod warunkiem zakończenia transformacji wyżej położonych warstw transformowanej macierzy. Dla budowy krokowego równoległego algorytmu, wykorzystuje się ponadto fakt, że pozostałe (niższe) warstwowe podmacierze \mathbf{K} mogą być równocześnie częściowo przetransformowane na \mathbf{L} , \mathbf{U} przez równoległe pracujące procesory, używając lokalnych danych z przydzielonych im podmacierzy. Dla pełnej transformacji LU, procesory te potrzebują elementów \mathbf{L} i \mathbf{U} wyznaczonych przez inne procesory w wyższych warstwach transformowanej macierzy, przekazywane im przez procesor "master" przed wykonaniem kolejnego kroku. Zgodnie z opracowanym algorytmem, każdy procesor transformuje przydzieloną mu warstwową podmacierz wiersz po wierszu, wykorzystując elementy macierzy \mathbf{L} oraz \mathbf{U} wyznaczone stopniowo w kolejnych podfazach (przebiegach pętli) kroku. Po zakończeniu każdego kroku, wyniki transformacji LU przesyłane są przez procesory "slave" do procesora "master", który wyznacza dla następnego kroku optymalną dystrybucję nieprzetworzonej do końca części macierzy \mathbf{K} (z częściowo przetworzonymi elementami \mathbf{L} i \mathbf{U}) pomiędzy procesory "slave" oraz dokonuje następnie wszelkich niezbędnych transmisji wyników częściowych. Należy tu podkreślić, że doktorant zakłada że procesory "slave" mają w ogólności różne moce obliczeniowe i różne przepustowości ich łączy komunikacyjnych do procesora "master" oraz że system wykonawczy jest zbudowany w oparciu o rozproszoną pamięć operacyjną.

Algorytm wyznaczania wielkości podmacierzy dla procesorów "slave" zakłada, że znane są charakterystyki czasowe całkowitej sekwencyjnej transformacji LU dla wszystkich procesorów dla podmacierzy o różnych liczbach wierszy. Te charakterystyki są wyznaczane eksperymentalnie przed wykonaniem algorytmu LU dla danego zestawu procesorów wykonawczych. Na podstawie tych charakterystyk procesor "master" wyznacza czas wykonania transformacji połowy nieprzetworzonej macierzy przez najszybszy procesor jako wyznaczaną wielkość czasu kroku obliczeniowego, a następnie określa jak wielkie podmacierze są w stanie przetworzyć w tym czasie pozostałe procesory - jest to osiągalne przez rozwiązanie swego równania kwadratowego dla każdego procesora. Następnie następuje sprawdzenie czy tak określone wielkości podmacierzy mieszczą się w przedziale $(n-\varepsilon, n+\varepsilon)$, gdzie n to liczba nieprzetworzonych do końca wierszy macierzy \mathbf{K} , natomiast ε to założona tolerancja doboru rozmiaru macierzy. Jeśli ta relacja jest spełniona, następuje wyrównanie liczby wierszy do n przez przydział brakujących wierszy do najszybszego procesora lub usunięcie nadmiarowych wierszy z podmacierzy najwolniejszego procesora. Jeśli ta relacja nie jest spełniona następuje modyfikacja wielkości czasu kroku obliczeniowego. Jest on zmniejszany do połowy jeśli przekracza $n+\varepsilon$, albo, gdy jest mniejszy od $n-\varepsilon$, jest on zwiększany o połowę różnicy między czasem transformacji macierzy dla najszybszego i drugiego pod względem szybkości procesora. Rozdział 5 zawiera przykład realizacji algorytmu optymalizacyjnego dla 4 procesorów.

W rozdziale 6 pracy opisano biblioteki komunikacyjne wykorzystywane dla programowania równoległego w tym bibliotekę MPI wykorzystaną przy realizacji eksperymentów weryfikujących opracowane algorytmy.

Rozdział 7 opisuje eksperymenty porównawcze zrealizowane dla oszacowania opracowanych algorytmów dekompozycji LU wykorzystując superkomputer SR 2201 znajdujący się w PJWSTK oraz dwurdzeniowy procesor pracujący pod systemem Windows XP. Opracowano dwie wersje programów: ze statycznym przydziałem

danych procesorom oraz z dynamicznym przydziałem zgodnie z zaproponowanym w rozprawie algorytmem. Wykonano testy algorytmów dla różnych rozmiarów macierzy (od 300×300 do 3000×3000) i różnych liczb procesorów w systemie. Uzyskane wyniki: czas wykonania dekompozycji LU i przyspieszenia obliczeń równoległych przedstawiono w tabelach oraz na wykresach. Wyniki eksperymentów świadczą o tym, że zaproponowany algorytm dekompozycji LU z dynamicznym przydziałem danych dla procesorów wykazuje przewagę nad algorytmem ze statycznym przydziałem danych o 20% dla rozmiarów macierzy 100×100 do ponad 30% dla rozmiarów macierzy 3000×3000 . W eksperymentach porównano ponadto zachowanie algorytmu z dynamicznym przydziałem danych w stosunku do nieoptymalizowanego algorytmu Linpack. Opracowany algorytm wykazuje od 2 do 3.5-krotny wzrost wydajności obliczeniowej w zakresie rozmiarów macierzy 600×600 do 3300×3300 .

W rozdziale 8 są opisane trzy zastosowania, które doktorant praktycznie zrealizował wykorzystując opracowany przez siebie algorytm rozwiązywania układów równań liniowych metodą dekompozycji LU. Jest to modelowanie ruchu obiektów wiotkich (tkanin) w polu grawitacyjnym, modelowanie metodą elementów skończonych prądów wirowych indukowanych przez zmienne pole magnetyczne w długich przewodach ferromagnetycznych oraz modelowanie rozkładu temperatury wewnątrz płytki prostokątnej przy pobudzeniu na jej brzegach. Modele matematyczne rozpatrywanych aplikacji to równania bądź układy równań algebraicznych i różniczkowych. Autor przedstawia sposoby sprowadzenia tych równań i układów równań do postaci równań liniowych, które dla zasymulowania przebiegu rozpatrywanych zjawisk w czasie zostały rozwiązane z wykorzystaniem przedstawionych w rozprawie równoległych metod z dekompozycją LU. Następnie przedstawione są graficznie wyniki obserwacji zmian zachodzących podczas symulacji.

Przechodząc do oceny opiniowanej rozprawy stwierdzam, że praca dotyczy aktualnego problemu naukowego, gdyż szybkie metody rozwiązywania układów równań liniowych w systemach równoległych są bardzo ważne dla obliczeń naukowych. Problem naukowy został w rozprawie poprawnie sformułowany. Celem pracy jest przeprowadzenie badań i zaproponowanie nowych rozwiązań algorytmicznych (dynamiczny przydział danych równoległym procesorom składowym) dla zrównoleglenia komputerowego rozwiązywania układów równań liniowych metodą dekompozycji LU i wykazanie, że dynamiczny przydział danych daje przyspieszenie wykonywania obliczeń w porównaniu ze statycznym przydziałem danych. Teza pracy jest sformułowana w sposób jasny i poprawny.

Doktorant rozwiązał postawiony problem używając poprawnych metod. Uzyskane w pracy wyniki są oryginalne. Działanie opracowanych algorytmów i rozwiązań sprzętowych zostało zweryfikowane poprzez eksperymenty symulacyjne. Uzyskane wyniki eksperymentów są pozytywne i świadczą o praktycznej użyteczności zaproponowanych rozwiązań. Na podstawie tych eksperymentów można sądzić, że zaproponowane w rozprawie rozwiązania znajdują zastosowanie w praktyce.

Niniejsza rozprawa doktorska świadczy o dobrej wiedzy Doktoranta w zakresie metod numerycznych dla rozwiązywania układów równań algebraicznych i różniczkowych oraz układów równań liniowych. Cytowana i wykorzystana literatura uwzględnia moim zdaniem podstawowe problemy cząstkowe dziedziny pracy.

Uwagi szczegółowe

Nie zauważyłem w pracy istotnych błędów merytorycznych. Natomiast istnieją pewne niejasności i przypadki niepełnego definiowania używanych pojęć i wielkości. Na str 28 do wzoru 2.4.13 powinien być dodany warunek "dla $i < j$ " a do wzoru 2.4.14 powinien być dodany warunek "dla $j \geq i$ ". Pojęcie operacji podstawowej, użyte na str. 32 i wykorzystywane na str. 63, 65 oraz 66, jest wprowadzone w sposób niejasny, gdyż właściwie brak tu jednoznacznej definicji precyzującej podoperacje składowe operacji podstawowej a sprawa jest istotna, gdyż tym operacjom jest przypisany domyślnie identyczny czas wykonania. Dobrze byłoby sformułować założenia architekuralne w układzie wykonawczym, przy których takie domniemanie będzie spełnione, zwłaszcza, gdy te operacje są użyte dla określenia mocy obliczeniowej procesora. Miejscami nadmiernie mnożone są pojęcia lub używane są różne nazwy dla określenia tych samych pojęć: jak np. krok obliczeniowy, faza obliczeniowa, krok optymalizacyjny, przebieg, etap początkowy na str. 63 i 67 czy pętla, etap, przebieg, iteracja na str. 32, 67 i 68. Nie zostały wystarczająco jasno opisane rysunki 2.4.2, 2.4.5 oraz 5.1.2. Nie jest jasne w jaki sposób dokładnie jest obliczany parametr $M(p_i)$, gdyż wskazywany dla tego celu rysunek 5.1.1 prezentuje jedynie sposób wyznaczenia testowego układu równań a nie wyznaczenia tego parametru. Nie wyjaśniony pozostaje również sposób wyznaczania "chwilowej mocy obliczeniowej" procesora, wspomnianej na str. 63. Podane uwagi szczegółowe utrudniają czytanie tekstu ale nie umniejszają ogólnej wartości algorytmów opracowanych w pracy.

Praca jest napisana w olbrzymiej większości poprawnym językiem, jednakże pozostawiono w niej pewne nieliczne błędy pisarskie i stylistyczne jak np. str. 19, 7 wiersz od dołu czy str. 56, 20 wiersz od dołu - rozpoczęcie zdania od małej litery czy str. 56 - podano dziwny tekst "brak wymiany danych nie ma konieczności", itp.

Podsumowując stwierdzam, że Doktorant uzyskał wyniki merytoryczne spełniające wymagania dla rozpraw doktorskich określone w obowiązującej ustawie o stopniach i tytule naukowym, dlatego wnoszę o dopuszczenie niniejszej rozprawy doktorskiej i Doktoranta do dalszych faz przewodu doktorskiego.

