

dr hab. inż. Paweł Czarnul
Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

Gdańsk, 17 lipca 2017

**RECENZJA ROZPRAWY DOKTORSKIEJ
DLA
INSTYTUTU PODSTAW INFORMATYKI
POLSKIEJ AKADEMII NAUK**

**Autor rozprawy: mgr Łukasz Górski
Instytut Podstaw Informatyki
Polskiej Akademii Nauk**

Tytuł rozprawy: Adaptacja wybranych równoległych algorytmów do modelu PGAS i ich implementacja w języku Java z wykorzystaniem biblioteki PCJ

Wstęp

Praca, napisana w języku polskim, zawiera krótki wstęp (3 strony), 4 obszernie rozdziały w tym wprowadzenie do zagadnienia, opis metodologii, analizowane schematy obliczeniowe, opis zaproponowanej równoległej metaheurystyki, wnioski oraz bibliografię. Praca liczy 159 stron z bibliografią. Wstęp pracy określa cele, temat i problemy badawcze, które podjęte zostały przez autora rozprawy, w szczególności wykazanie stosowalności modelu PGAS (Partitioned Global Address Space) w języku Java dzięki implementacji równoległych wersji wybranych algorytmów. Wstęp zawiera krótkie streszczenie pracy a także listę 5 publikacji, których współautorem jest autor rozprawy. Rozdział pierwszy zawiera wprowadzenie do zagadnienia, w tym podstawowe pojęcia związane z równoległością i obliczeniami równoległymi, podstawowe czynniki pozwalające na zwiększenie wydajności w systemach równoległych, omówienie podstawowych paradygmatów programowania równoległego, w tym, co istotne z punktu widzenia pracy, modelu PGAS wraz z przykładami realizacji niektórych konstrukcji. Autor omówił wybrane możliwości obsługi równoległości z poziomu języka Java, w tym biblioteki PCJ pozwalającej na programowanie

w paradygmacie PGAS. Rozdział drugi zawiera opis podstawowych metryk oceny wydajności i efektywności obliczeń równoległych, opis wybranych benchmarków, w tym istotnego z punktu widzenia pracy HPC Challenge Benchmark, a także omówienie środowiska testowego wraz ze specyfikacją systemów, w których autor przeprowadził testy. Z kolei rozdział trzeci zawiera opis podstawowych schematów komunikacyjnych, a także aplikacje testowych wykorzystywanych dalej. Autor omówił rozgłoszenie każdy do każdego wraz z różnymi implementacjami z wykorzystaniem PCJ, wraz z pomiarami czasów wykonania w zależności od różnej liczby wątków, w różnych systemach. Następnie przeanalizowany został test dostępu do pamięci. W dalszej kolejności autor opisał implementacje i wyniki testów wydajnościowych dla jednowymiarowej, szybkiej dyskretnej transformaty Fouriera jak i trójwymiarowej transformaty Fouriera, a także implementację i wyniki testu HPL w środowisku rozproszonym, z porównaniem wersji PCJ do HPCC. Rozdział czwarty obejmuje charakterystykę zrównoleglenia algorytmu ewolucyjnego wraz z implementacją z wykorzystaniem biblioteki PCJ, z różnymi strategiami migracji osobników i oceną w wybranych systemach. Mgr Łukasz Górski zaprezentował także implementację wraz z wynikami symulacji konektomu *Caenorhabditis elegans*, dla różnej liczby wątków. Autor rozpoznał także problem niezrównoważenia obciążenia i przetestował różne strategie pozwalające na zminimalizowanie problemu. Bibliografia zawiera 162 pozycje.

Jaki jest problem naukowy (teza) rozprawy i czy został on trafnie i jasno sformułowany?

Praca zorientowana jest na zbadanie przydatności i możliwości wykorzystania modelu PGAS oraz implementacji tego modelu w języku Java do tworzenia naukowych aplikacji równoległych. Jako tezę pracy (we wstępie) można odczytać wykazanie stosowalności modelu PGAS w Javie poprzez implementację zrównoleglonych wersji wybranych algorytmów. Autor, jako problemy badawcze, wymienia także opracowanie równoległych wersji algorytmów, które wykorzystują komunikację asynchroniczną oraz nakładanie obliczeń i komunikacji, a także zbadanie skalowalności i wydajności zaimplementowanych algorytmów. Problem został jasno sformułowany.

Czy autor rozwiązał postawiony problem i czy użył do tego właściwych metod dowodząc, że posiadał umiejętności związane z metodyką i metodologią prowadzenia badań naukowych?

Autor rozwiązał postawiony problem (tj. wykazanie stosowalności modelu PGAS w Javie wymienione powyżej) poprzez:

1. Omówienie podstaw programowania z wykorzystaniem biblioteki PCJ realizującej paradygmat PGAS dla aplikacji w języku Java a także analizę możliwości tego modelu.
2. Wskazanie różnych wariantów implementacji (z wykorzystaniem PCJ) podstawowych schematów wykorzystywanych dalej, w tym rozgłoszenia każdy do każdego wraz z pomiarami czasów wykonania w zależności od różnej liczby wątków, w różnych systemach. Autor wykazał, że czasy wybranych wersji w niektórych systemach są porównywalne do implementacji MPI (za wyjątkiem systemu XC40 ze specjalnie zoptymalizowaną implementacją MPI). Gorsze wyniki dla PCJ uzyskano w teście RandomAccess, ale jednocześnie o wzrastających wartościach przyspieszenia dla zwiększanej liczby wątków, z możliwością uruchomienia testu dla 2048 wątków.
3. Przeprowadzenie testów wydajnościowych kodu jednowymiarowej transformaty Fouriera, z różnymi wersjami maszyn wirtualnych i rodzajami komunikacji względem referencyjnych implementacji wykorzystujących MPI (oraz także OpenMP w jednym przypadku). Autor wykazał, że dla systemu westmere implementacje wykorzystujące PCJ mogą być konkurencyjne do większości innych rozpatrywanych rozwiązań, w szczególności dla liczby wątków przekraczającej 100. Dla systemu XC40 osiągnięto wyniki gorsze niż HPCC, ale o podobnym tempie wzrostu względem liczby wątków. Autor wykazał także dobrą skalowalność kodu PCJ dla trójwymiarowej transformaty Fouriera dla systemu westmere.
4. Stworzenie implementacji rozwiązania ewolucyjnego dla symulacji konektomu *C. elegans* z wykazaniem bardzo dobrej skalowalności względem liczby wątków a także polepszenia osiągniętego wyniku (rozpatrywano minimum funkcji). Autor przeanalizował także problem nierównoważenia obciążenia proponując różne strategie rozwiązania.

Wyniki zamieszczone w pracy zostały także opublikowane w ramach artykułów indeksowanych w bazie Web of Science.

Czy tematyka rozprawy jest aktualna lub dostatecznie ważna?

Tematyka rozprawy, w kontekście modeli programistycznych, a w tym przypadku modelu PGAS i jego stosowalności do programowania równoległego, jest z całą pewnością aktualna i ważna. Wynika to z faktu, że obecnie wzrost wydajności systemów równoległych następuje głównie poprzez zwiększanie liczby rdzeni - zarówno w ramach procesorów, koprocessorów, akceleratorów jak i poprzez zwiększanie liczby węzłów w klastrach obliczeniowych – takich jak te wykorzystywane do testów w ramach rozprawy (Sekcja 2.2). Oznacza to, że łatwość programowania jak i osiągnięta wydajność dla aplikacji uruchamianych w takich systemach są

niezwykle ważne. Model PGAS z implementacją PCJ stanowić może tutaj alternatywę dla tradycyjnych API programistycznych jak MPI czy też MPI+OpenMP. Model PGAS z implementacją PCJ pozwala na programowanie aplikacji wielowątkowej w środowisku wielowęzłowym z udostępnianiem wybranych zmiennych innym wątkom. Ważność podejmowanej tematyki i rozwiązanie zostały także docenione w postaci przyznania nagrody HPC Challenge Benchmark w kategorii produktywności (HPC Challenge Benchmark Class 2, Most Elegant Award) – <http://hpcchallenge.org>.

Czy rozprawa świadczy o dostatecznej wiedzy autora, wiedzy na zaawansowanym poziomie, o charakterze podstawowym dla dziedziny nauk technicznych oraz o charakterze szczegółowym, odpowiadającym obszarowi badań naukowych?

Czy rozprawa obejmuje najnowsze osiągnięcia nauki i świadczy o znajomości współczesnej literatury z dyscypliny naukowej, której dotyczy?

Autor wykazał znajomość zagadnień programowania równoległego (zarówno na poziomie podstawowym jak i szczegółowym) poprzez analizę:

1. Sprzętowych elementów wpływających na zwiększanie wydajności aplikacji równoległych takich jak, najistotniejsze dzisiaj: wielordzeniowość, wykorzystanie pamięci cache.
2. Modeli programowania równoległego wraz z reprezentatywnymi API pozwalającymi na, w szczególności: programowanie w systemach z pamięcią współdzieloną (OpenMP), programowanie w systemach z pamięcią rozproszoną jak i współdzieloną (MPI), a także modeli aktorów, programowania funkcyjnego etc.
3. Metryk wydajności i efektywności aplikacji równoległych wraz z omówieniem podstawowych benchmarków i ich charakterystyką pod kątem lokalności przestrzennej i czasowej.
4. Możliwości wykorzystania modelu PGAS i biblioteki PCJ.

Przedstawione wyżej elementy, dotyczące zarówno sprzętowej jak i programowej strony aplikacji równoległych, opatrzone zostały licznymi cytatami z literatury, także z roku 2016 lub takich, do których autor uzyskał dostęp w roku 2017. Bogata literatura liczy 162 pozycje.

Na czym polega oryginalny dorobek autora i jakie jest jego znaczenie poznawcze?

Jakie są najważniejsze osiągnięcia rozprawy?

Oryginalny dorobek autora i osiągnięcia przedstawione w pracy postrzegam w opracowaniu:

1. Propozycji i implementacji schematów jak rozgłoszenie każdy z każdym, dostępu do pamięci, implementacji algorytmów jednowymiarowej transformaty Fouriera jak i trójwymiarowej dyskretnej transformaty Fouriera z wykorzystaniem biblioteki PCJ, wraz z przeprowadzeniem testów porównawczych w stosunku do implementacji bazujących na MPI, w różnych systemach.
2. Projektu i implementacji równoległego algorytmu ewolucyjnego z wykorzystaniem biblioteki PCJ, wraz z różnymi strategiami migracji osobników i oceny ich wydajności w środowisku równoległym.
3. Projektu i implementacji wykorzystującej algorytm ewolucyjny i bibliotekę PCJ do optymalizacji modelu konektomu nicienia *C. elegans*, z demonstracją dobrej skalowalności i udokumentowaniem polepszenia rozwiązania dla zwiększanej liczby wątków.
4. Opracowanie mechanizmów równoważenia obciążenia w algorytmie ewolucyjnym omówionym powyżej.

Wymienione powyżej osiągnięcia stanowią rozwiązanie problemu naukowego rozprawy, określają również możliwości wykorzystania rozwiązań w podobnych, jak rozpatrywane w pracy, systemach równoległych.

Jakie są wady i słabe strony rozprawy?

1. Nie jest jasne dlaczego dane testy wykonano dla wybranych, konkretnych systemów, a nie dla innych. Czy wyniki te mogą w takim układzie zostać uznane za w pełni reprezentatywne, ew. reprezentatywne dla danych konfiguracji sprzętowych z konkretnymi pakietami/wersjami oprogramowania?
2. W kontekście komentarza autora do rys. 3.19 (komentarz na s. 98), warto byłoby przeprowadzić szersze testy dla FFTW dla innych rozmiarów danych. Czasy wykonania dla eksperymentu, dla którego wyniki przedstawiono na rys. 3.19 są niewielkie, wydaje się, że zbyt małe dla wyniesienia w pełni wiarygodnej konkluzji odnośnie porównania implementacji.

3. Opis wykresów 4.2 i 4.3 nie jest do końca wyczerpujący. Ponadto, na stronie 115 można znaleźć zdanie: „Brak synchronizacji nie wpłynął również negatywnie na osiągnięte wyniki optymalizacji (poziom energii na wykresach, rozumiany jako najmniejsza obliczona wartość funkcji celu)”. Jak rozumieć „poziom energii na wykresach”?
4. Brakuje nieco wniosków z wyników zamieszczonych w sekcji 3.5. Autor sam sugeruje przeprowadzenie testów ponownie co stawia pod znakiem zapytania celowość umieszczenia wyników dla wersji Java.
5. W kontekście stosowalności modelu PGAS z implementacją PCJ, można by przeprowadzić szerszą ocenę łatwości programowania równoległego z wykorzystaniem tego ostatniego w porównaniu do klasycznego API MPI+OpenMP lub też MPI+Pthreads, w szczególności w kontekście uwagi autora o relatywnie łatwej implementacji oprogramowania (s. 139) i komentarzu zawartym we wnioskach.

Uwagi szczegółowe i redakcyjne

Praca zawiera błędy tzw. „literówki”, poniżej podano właściwą formę:

1. s. 50 „W kontekście oprogramowania o charakterze naukowym (...)”
2. s. 53 „Referencyjne implementacje wykorzystują język C i bibliotekę MPI (...)”
3. s. 55 „Konkurs HPC Challenge Award przyznaje dwie kategorie nagród.”
4. s. 57 „Na użytek niniejszej pracy oprócz testów stanowiących część zestawu HPC Challenge Benchmark”
5. s. 86 „(...) (rozwiązanie jest oparte o sześciokrokowy algorytm FFT, stanowiący rozwinięcie czterokrokowego zaprezentowanego w pracy jako algorytm 5)”
6. s. 100 „W celu sfaktoryzowania bieżącego panelu wykonywane są: wybór elementu dzielącego, (...)”
7. s. 105 „Sama nazwa „metaheurystyka” może być myląca (...)”
8. s. 110 „Oprócz możliwości doboru wartości podstawowych zmiennych (...)”
9. s. 113 „Uruchomiono 40000 iteracji ewolucji różnicowej (...)”

10. s. 123 „Na przykład, w wypadku kodu związanego z modelowaniem konektomu nicienia, większość z kilkudziesięciu (...)”, ponadto brakuje właściwej kontynuacji tego zdania.

Ponadto, tekst na stronie 84 oraz na stronie 86 wychodzi na prawy margines.

W podpisie rysunku 3.11 (s. 87) mowa jest o dwóch wątkach, w opisie o czterech.

Byłoby lepiej, aby wyniki dla tych samych rozwiązań były oznaczone analogicznie na odpowiednich wykresach – np. na rys. 3.7 i rys. 3.8.

Po lekturze pracy nasuwa się pytanie czy możliwa jest rekomendacja rozwiązań, w szczególności PCJ, dla konkretnych klas programów, dla konkretnych środowisk? Czy wymaga to każdorazowo porównawczych testów wydajnościowych?

Podsumowanie i ocena rozprawy

W rozprawie doktorskiej mgr Łukasz Górski wykazał stosowalność modelu PGAS w implementacji w języku Java do realizacji wybranych równoległych aplikacji opisanych powyżej – szczególnie w kontekście wygodnego modelu programowania wielowątkowego w języku Java pozwalającego na uzyskanie relatywnie dobrej wydajności i skalowalności. Tematyka pracy jest istotna, aktualna i o dużym znaczeniu dla wytwarzania aplikacji równoległych uruchamianych w systemach obliczeniowych z wieloma rdzeniami.

Podsumowując, w moim przekonaniu rozprawa spełnia wymagania postawione rozprawom doktorskim przez Ustawę z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz stopniach i tytule w zakresie sztuki, Dz.U. 2003 nr 65 poz. 595 z późniejszymi zmianami, w dziedzinie nauk technicznych w dyscyplinie informatyka i wnioskuję o dopuszczenie rozprawy do publicznej obrony.

Do której kategorii recenzent zalicza rozprawę?

Rozprawę doktorską pana magistra Łukasza Górskiego zaliczam do kategorii:

zadowolająco spełniająca wymagania



dr hab. inż. Paweł Czarnul