

Częstochowa, dn. 5.08.2018

Prof. dr hab. inż. Roman Wyrzykowski
Instytut Informatyki Teoretycznej i Stosowanej
Politechnika Częstochowska
ul. Dąbrowskiego 69
42-201 Częstochowa

RECENZJA
ROZPRAWY DOKTORSKIEJ
Mgr Magdaleny Ryczkowskiej
„Opracowanie w modelu PGAS wybranych równoległych algorytmów
grafowych i ich implementacja przy użyciu języka Java”

Promotor: prof. dr hab. Piotr Bała

Interdyscyplinarne Centrum Modelowania
Matematycznego i Komputerowego
Uniwersytetu Warszawskiego

Niniejsza recenzja przygotowana została na zlecenie Z-cy Dyrektora ds. Naukowych Instytutu Podstaw Informatyki PAN w Warszawie, Pana prof. dra hab. inż. Wojciecha Penczka - pismo z dnia 6 czerwca 2018 r.

1. Ocena wyboru tematu i tezy (celu) rozprawy

Tematyka przedłożonej do recenzji rozprawy doktorskiej mgr Magdaleny Ryczkowskiej dotyczy zagadnień programowania równoległego, obszaru niezwykle istotnego i dynamicznie rozwijającego się od lat. Wśród czynników które temu sprzyjały należy wymienić wykrystalizowania się architektur wielordzeniowych jako podstawowego sposobu zwiększenia wydajności procesorów, a także rozwój obliczeń wielkiej skali z wykorzystaniem systemów pozwalających na uzyskanie wydajności wielu Petaflops, a ostatnio zmierzających do przekroczenia „progu marzeń” – czyli 1 Exaflops. Jednocześnie, skuteczne wykorzystanie olbrzymiego potencjału takich systemów, charakteryzujących się znacznym kosztem zarówno ich budowy, jak i utrzymania, wymaga opracowania, niekiedy od podstaw, wielu nowych technologii i podjęcia szeregu wyzwań badawczych. Zakres tych wyzwań rozciąga się od dogłębnego przeformułowania modeli matematycznych i związanych z nim algorytmów rozwiązywania zagadnień aplikacyjnych po opracowanie efektywniejszych narzędzi i środowisk programistycznych, wraz z obiektywną weryfikacją ich przydatności.

W tym kontekście bardzo pozytywnie oceniam wybór tematyki rozprawy zmierzającej do wykazania dużych możliwości wiążących się z zastosowaniem języka programowania Java oraz biblioteki PCJ (*Parallel Computation in Java*) jako metody umożliwiającej efektywne wdrożenie nowoczesnego modelu PGAS (ang. *Partitioned Global Address Space*) programowania równoległego do wydajnej realizacji ważnej klasy algorytmów, którą tworzą algorytmy grafowe. Algorytmy te wykorzystywane są, na przykład, w zagadnieniach eksploracji danych czy też do analizy grafów semantycznych czy serwisów społecznościowych, a charakterystyki złożonościowe tych obszarów zastosowań decydują o potencjalnie bardzo dużej złożoności praktycznej implementacji algorytmów grafowych.

Jako rozwiązanie umożliwiające stworzenie wspólnej wirtualnej przestrzeni adresowej dostępnej dla bardzo wielu współdziałających wątków, model PGAS wraz ze wspierającym go API zapewnia twórcom oprogramowania równoległego (rozproszonego) w pełni transparentny dostęp zarówno do pamięci lokalnej, jak i zdalnej (rozproszonej) współczesnych systemów HPC. Oczywiście, nie jest jedyna implementacja modelu PGAS dostępna w chwili obecnej, gdyż istnieje cała gama rozwiązań alternatywnych, takich jak języki X10, Chapel, UPC czy też interfejs GASPI (*Global Address Space Programming Interface*). Wyróżnia ją jednak jedna niezwykle istotna zaleta. Jest nią możliwość głębokiej integracji tradycyjnej dziedziny aplikacji HPC, obejmującej głównie zagadnienia obliczeniowe, symulacyjne i optymalizacyjne, z innymi niezwykle ważnymi dziedzinami, takimi jak zagadnienia przetwarzania bardzo dużych zbiorów danych, kryjące się pod nośnym określeniem Big Data, czy klasyczne już obszary programowania aplikacji internetowych oraz aplikacji mobilnych.

Lokując się w nakreślonej wyżej tematyce, recenzowana praca mgr Ryczkowskiej skupia się na wszechstronnej analizie możliwości opracowania metod efektywnej implementacji w modelu PGAS wybranych, równoległych algorytmów grafowych w kontekście ich implementacji w języku Java. Zaproponowanemu w pracy podejściu do rozwiązania tego problemu towarzyszy zbadanie wydajności i skalowalności opracowanych algorytmów jako kroku niezbędnego do eksperymentalnej weryfikacji przydatności modelu PGAS w oparciu o język Java i bibliotekę PCJ, korzystnie świadcząc o przygotowaniu Autora do działalności badawczej. Chciałbym również zauważyć, iż recenzowana rozprawa nawiązuje wyraźnie do wcześniejszej rozprawy doktorskiej Pana Łukasza Górskiego z tego samego zespołu, a poświęconej adaptacji wybranych zagadnień numerycznych oraz optymalizacyjnych do modelu PGAS z wykorzystaniem biblioteki PCJ.

Podsumowując ten punkt recenzji, pragnę stwierdzić, iż wybór tematyki rozprawy Pani mgr Ryczkowskiej uważam za trafny, a zaproponowany kierunek i cel badań oceniam zdecydowanie pozytywnie. Rozważane w rozprawie problemy są istotne i aktualne, zarówno dla teorii, jak i przede wszystkim praktyki współczesnego programowania równoległego. Lokują się one korzystnie w nakreślonej wyżej tematyce współczesnej informatyki, dotyczącej aspektów teoretycznych organizacji obliczeń w systemach HPC oraz zastosowań praktycznych tych systemów dla szerokiej gamy aplikacji, definiując teoretyczny i jednocześnie technologiczny charakter rozprawy.

2. Koncepcja i redakcja rozprawy

Recenzowana praca doktorska obejmuje formalnie krótkie wprowadzenie, 5 rozdziałów i podsumowanie, a także bibliografię zawierającą 97 pozycji. Praca liczy łącznie 134 strony.

W *rozdziale pierwszym* Autor zawarł stosunkowo krótkie do zagadnień przetwarzania grafów, koncentrując się na przedstawieniu niektórych sfer praktycznych zastosowań oraz podstawowych modeli przetwarzania grafów. W szczególności, zaprezentowano metody macierzowe i wprowadzono podstawowe macierzowe sposoby reprezentacji grafów, co pozwala przetrząść pomost pomiędzy równoległymi technikami obliczeń na macierzach rzadkich a równoległymi algorytmami grafowymi.

Rozdział drugi poświęcono zwięzłej charakterystyce paradygmatów i modeli programowania równoległego, w tym modelu PGAS oraz biblioteki PCJ jako narzędzia programowania wykorzystanego w recenzowanej rozprawie do równoległej implementacji opracowanych algorytmów grafowych. Wskazano przy tym na ważne praktyczne zalety tego narzędzia związane z jego samodzielnością, tj. niezależnością od dodatkowych bibliotek i wykorzystaniem bez żadnych modyfikacji języka Java.

Rozdział trzeci, wraz z kolejnymi dwoma rozdziałami, zawiera autorską część pracy ukierunkowaną na opracowanie algorytmów umożliwiających dostosowanie powszechnie wykorzystywanego benchmarku Graph 500 do modelu PGAS i biblioteki PCJ. W rozdziale tym skoncentrowano się na adaptacji dwóch testów obliczeniowych (tzw. kerneli) powyższego benchmarku, a mianowicie:

(ii) przeszukiwanie grafu wszerz, znane pod akronimem BFS (ang. *breadth-first search*).

(ii) transformacji wygenerowanego wcześniej grafu nieskierowanego z postaci listy krawędzi do bardziej zoptymalizowanej struktury danych wykorzystywanych

przez poprzedni kernel – w pracy przyjęto tutaj klasyczną reprezentację CSR (ang. *Compressed Sparse Row*) wykorzystywaną do przechowywania macierzy rzadkich.

Mając na uwadze uzyskanie możliwie najwyższej wydajności oraz skalowalności rozwiązania, zastosowano techniki pozwalające na ograniczenie wymiany informacji, jak również nałożenie obliczeń i komunikacji, co było możliwe dzięki wykorzystaniu komunikacji jednostronnej charakterystycznej dla modelu PGAS. Zbadano także i uwzględniono właściwości wykorzystywanych struktur danych, biorąc pod uwagę specyfikę języka Java.

W rozdziale czwartym zaprezentowano testy wydajnościowe opracowanych rozwiązań dla obydwu kerneli na trzech platformach sprzętowych, które przedstawiały sobą klastry obliczeniowe wyposażone w procesory o różnej architekturze, począwszy od 6-rdzeniowych procesorów AMD Opteron, a skończywszy na 12-rdzeniowych (a nie 24-rdzeniowych, jak błędnie wskazano w pracy) procesorach Intel Xeon o architekturze Haswell. W przeciwieństwie do pierwszego klastra, którego węzły komunikowały się za pośrednictwem stosunkowo mało wydajnej sieci 10Gb Ethernet, węzły tego ostatniego klastra były połączone wysokowydajną siecią o topologii Dragonfly, opracowaną przez firmę Cray. Wyniki porównano z rozwiązaniami zaimplementowanymi w języku C z wykorzystaniem interfejsu MPI.

Przedmiot **rozdziału piątego** stanowiło natomiast porównanie wydajności opracowanej implementacji z rozwiązaniem wykorzystującym alternatywny model MapReduce programowania równoległego, oparty o platformę Hadoop. Węzły testowanego klastra połączone były siecią InfiniBand, przy czym nie podano szczegółowego typu tej sieci.

Rozprawę kończy zwięzłe podsumowanie pracy, a także wskazanie kierunków dalszych badań w obszarze tematyki rozprawy.

3. Wkład Autora i zaprezentowana wiedza

Uwzględniając powyższe omówienie zawartości pracy oraz ogólną pozytywną ocenę jej zawartości merytorycznej, uważam, że za bezsporne osiągnięcia Autora należy uznać następujące rezultaty:

1. Podstawowym wynikiem o ogólnym charakterze jest wykazanie możliwości rozszerzenia zakresu stosowalności modelu PGAS w języku Java o bardzo ważną klasę algorytmów jaką stanowią algorytmy grafowe, charakteryzujące się wysoce nieregularną strukturą i dużymi narzutami na komunikację. Wynik ten, rozpatrywany łącznie z rezultatami wcześniejszej rozprawy Pana Łukasza Górskiego, rozszerza spektrum alternatywnych rozwiązań dla języków i środowisk programowania aplikacji równoległych wykorzystujących współczesne platformy

HPC. Dzięki wykorzystaniu niezmodyfikowanego języka Java, dane osiągnięcie stanowi także ważny krok w kierunku integracji obszaru aplikacji internetowych i mobilnych oraz tradycyjnej dziedziny aplikacji HPC, skupionej na zagadnieniach numerycznych, symulacyjnych i optymalizacyjnych, z innymi ważnymi obszarami zastosowań, takimi jak zagadnienia wydajnego przetwarzania i analizy dużych zbiorów danych, w tym szczególnie danych grafowych, znanych jako BigData.

2. Wykazanie stosowalności modelu PGAS i biblioteki PCJ do zrównoleglenia algorytmów grafowych zrealizowano poprzez opracowanie, a następnie eksperymentalne zbadanie wydajności i skalowalności równoległych wersji dwóch kerneli obliczeniowych wchodzących w skład reprezentatywnego benchmarku Graph500, a mianowicie, algorytmu przeszukiwania grafu wszerz (BFS) oraz transformacji grafu nieskierowanego do zoptymalizowanej struktury danych wykorzystywanej przez poprzedni kernel. Skuteczność zrealizowanego dostosowania wynikała nie tylko z uwzględnienia właściwości wykorzystywanych struktur danych w języku Java, lecz przede wszystkim z zastosowania technik pozwalających na ograniczenie wymiany informacji, jak również nałożenie obliczeń i komunikacji, co było możliwe dzięki wykorzystaniu komunikacji jednostronnej charakterystycznej dla modelu PGAS.
3. Istotne znaczenie nie tylko ściśle praktyczne, lecz również poznawcze ma dokonane w pracy eksperymentalne porównanie wydajności i skalowalności opracowanych implementacji dla modelu PGAS i języka Java z wersjami analizowanych algorytmów wykorzystującymi, z jednej strony, niskopoziomowy interfejs MPI i język C, a z drugiej – modelu MapReduce i platformy Hadoop dedykowanych do przetwarzania równoległego dużych zbiorów danych. Zrealizowane badania wykazały znaczną konkurencyjność opracowanych implementacji w stosunku do niskopoziomowej kombinacji C/MPI i dużą przewagę w porównaniu do połączenia MapReduce/Hadoop.

Uzyskane wyniki zostały opublikowane w siedmiu pracach w języku angielskim, które opublikowano w materiałach reprezentatywnych, specjalistycznych konferencji o zasięgu ogólnosiwiatowym, poświęconych problematyce przetwarzania równoległego oraz rozproszonego. Świadczy to pozytywnie o stopniu weryfikacji uzyskanych rezultatów przez międzynarodową społeczność specjalistów zajmujących się rozpatrywaną dziedziną.

Z omówienia treści rozprawy, które przytoczono w punkcie 2 niniejszej recenzji, wynika, iż wstęp do pracy oraz dwa pierwsze rozdziały rozprawy poświęcone są głównie krytycznemu przedstawieniu stanu wiedzy w zakresie tematyki pracy, potwierdzając w ten sposób ogólny stan wiedzy w zakresie dyscypliny Informatyka, ze szczególnym uwzględnieniem zagadnień programowania równoległego i

rozproszonego, w tym: roli grafów we współczesnych aplikacjach komputerowych i wykorzystywanych w tym obszarze podstaw przetwarzania grafów, modeli programowania równoległego wraz z odpowiednimi standardami i środowiskami programistycznymi. Jakość tych rozdziałów nie budzi moich zastrzeżeń. Świadczą one o dużej wiedzy Autora w zakresie tematyki badań, popartej szerokim doświadczeniem praktycznym związanym z wdrożeniem i wykorzystaniem różnych środowisk programowania równoległego, co zademonstrowano w rozdziałach czwartym i piątym.

Również moja opinia o bibliografii wykorzystanej w pracy oraz jej kompletności jest generalnie pozytywna, co nie wyklucza szczegółowych uwag krytycznych.

4. Poprawność pracy i uwagi krytyczne

Poprawność treści pracy nie wzbudza moich istotnych zastrzeżeń, a stwierdzenia w niej zawarte wydają się być godne zaufania, co wynika w szczególności z dosyć szczegółowych uzasadnień, popartych wynikami przeprowadzonych badań eksperymentalnych, w tym również testów porównawczych przeprowadzonych dla biblioteki PCJ, standardu MPI oraz modelu MapReduce. Generalnie sposób i jakość przeprowadzenia badań eksperymentalnych stanowi bardzo wartościowy element pracy i zasługuje na podkreślenie. W szczególności, dotyczy to dostatecznie wyczerpujących eksperymentów przeprowadzonych w środowiskach równoległych udostępnianych przez ogólnopolski projekt PL-Grid. Z doświadczeń moich współpracowników wiem, z jakimi trudnościami trzeba się niekiedy zmierzyć, aby dostosować tego typu infrastrukturę o charakterze publicznym do określonych wymagań i wręcz zmusić ją do stabilnego funkcjonowania dla niestandardowych zastosowań i aplikacji.

Jednocześnie Autor nie ustrzegł się pewnych braków i słabości. Wśród uwag o charakterze krytycznym, a po trosze dyskusyjnym, wymienić należy:

1. Algorytmy zaproponowane w pracy mają na celu zapewnienie jak najwyższej wydajności i skalowalności obliczeń równoległych z uwzględnieniem np. ograniczeń na wydajność sieci komunikacyjnej czy objętość pamięci. W celu potwierdzenia skuteczności swoich propozycji Autor stosuje wyłącznie badania eksperymentalne i nie próbuje w ogóle budować jakichkolwiek modeli, aby później wykorzystać je do analizy wydajności i skalowalności w sposób mniej lub bardziej ogólny.
2. Generalnie, oparcie się wyłącznie na podejściu eksperymentalnym jako jedynym stosowanym w pracy sposobie weryfikacji zaproponowanych algorytmów stanowi jej słabość. Brak rozważań o charakterze bardziej teoretycznym, z wykorzystaniem aparatu formalnego, pozbawia Autora możliwości głębszego zbadania charakterystyk procesów obliczeniowych, w szczególności z uwzględnieniem właściwości wykorzystywanych architektur, co z pewnością

umożliwiłyby wyciągnięcia na tej podstawie szeregu istotnych konkluzji i sformułowanie przydatnych rekomendacji. Na przykład, rekomendacje te mogłyby pomóc twórcy aplikacji wybrać te a nie inne konstrukcje biblioteki PCJ bez każdorazowego przeprowadzania testów porównawczych.

3. Rozumiem oczywiście, że uwzględnienie dwóch powyższych uwag jest zadaniem dostatecznie trudnym, biorąc pod uwagę chociażby nieregularność struktury zależności pomiędzy operacjami algorytmów grafowych. Zagadnienie to może stanowić ciekawy temat do przyszłych prac. Skoro więc Autorka skupiła się na podejściu eksperymentalnym do problemu analizy wydajności i skalowalności opracowanych algorytmów, pewien niedosyt budzi np. fakt, iż w rozdziale czwartym nie dokonano eksperymentalnej analizy wpływu poszczególnych technik ukierunkowanych na skuteczną adaptację tych algorytmów do modelu PGAS i języka Java (patrz punkt 2 w poprzednim rozdziale mojej recenzji). W tej chwili np. można się tylko domyślać, która z tych technik okazała się najistotniejsza z punktu widzenia wydajności i/lub skalowalności.
4. Przeprowadzone badania pokazały kluczowe znaczenie sieci komunikacyjnych w klastrze, pozwalających na zapewnienie szybkiego łącza między węzłami. W tym kontekście pewien niedosyt z punktu widzenia interpretacji wyników budzi bardzo pobieżne przedstawienie wykorzystywanych sieci komunikacyjnych. Jest to szczególnie istotne w przypadku sieci wykorzystujących mieszankę szybkiej technologii Infiniband i niezbyt wydajnej sieci 1Gb Ethernet. W związku z tym nasuwa się szereg pytań. Jaka jest topologia sieci ? Na ile wydajna jest wynikowa sieć ? Jak można scharakteryzować jej wydajność i mam na myśli nie tylko przepustowość, ale przede wszystkim parametr opóźnienia w przesyłaniu komunikatów ? Prawdopodobnie ten drugi parametr (o czym nie wspomniano w pracy) ma znaczenie decydujące ?
5. Przy wyznaczaniu słabej skalowalności (rysunek 4.8) istotne jest zapewnienie stałej ilości pracy na watek. W pracy nie przytoczono żadnych wyliczeń na ten temat. Nie bardzo nawet wiadomo czemu odpowiadają punkty pośrednie na tym rysunku (dla 64-512 wątków).
6. Dodatkowo chciałem zapytać co spowodowało, że w pracy nie zamieszczono w ogóle wykresów dla przyspieszenia przy stałym rozmiarze problemu (silna skalowalność). Parametr przyspieszenia lepiej jednak charakteryzuje skalowalność niż wnioskowanie o niej na podstawie czasu wykonania algorytmu przy stałym rozmiarze problemu.
7. Rozdział czwarty poświęcony opisowi wyników badań wydajnościowych należałoby zakończyć przedstawieniem wynikających z nich wniosków. Mogłyby one stanowić wartościowe rekomendacje dla innych badaczy, gdyby chcieli np. przystąpić do zrównoleglenia innych algorytmów grafowych.

8. Sam również chciałbym zadać pytanie Autorce o ewentualne doświadczenia z równoległą implementacją w modelu PGAS innych algorytmów grafowych czy aplikacji wykorzystujących grafy.
9. W rozdziale piątym Autorka porównuje implementacje algorytmu BFS w modelach PGAS oraz MapReduce. Do realizacji wersji w MapReduce wykorzystano rozproszoną platformę Hadoop. Alternatywą dla platformy Hadoop, która bazuje na rozproszonym systemie plików HDFS, stanowi platforma Spark, w której operacje wykonywane są w pamięci operacyjnej, co potencjalnie zapewnia większą wydajność. Jaka jest opinia Autorki na temat tej alternatywy i czy była ona rozpatrywana? Podejrzewam, że w tym przypadku nie będzie już tak dużej przewagi PCJ nad Spark, jak ma to miejsce dla Hadoop.
10. Moim zdaniem, celowe byłoby uzupełnienie części pracy zawierającej *state-of-the-art* o chociażby krótki podrozdział poświęcony podstawowym metrykom wydajności i skalowalności obliczeń równoległych. Dwa króciutkie akapity na początku rozdziału czwartego to rozwiązanie niesatysfakcjonujące.

5. Podsumowanie

Przytoczone wyżej uwagi krytyczne i dyskusyjne nie umniejszają zasług Autorki ani nie kwestionują jej osiągnięć. Recenzowana praca zasługuje na pozytywną ocenę merytoryczną i wnosi oryginalny wkład w rozwój metod i algorytmów organizacji obliczeń równoległych z wykorzystaniem nowoczesnych języków i narzędzi programistycznych, co pozwala ją jednoznacznie zakwalifikować do **dyscypliny Informatyka**. Podjęta tematyka dobrze wpisuje się we współczesny nurt badań w tym zakresie. Podstawowe cele i zadania pracy zostały zrealizowane.

W podsumowaniu recenzji stwierdzam zatem, że opiniowana rozprawa Pani mgr Magdaleny Ryczkowskiej pt. „Opracowanie w modelu PGAS wybranych równoległych algorytmów grafowych i ich implementacja przy użyciu języka Java” zawiera samodzielne rozwiązanie przez doktorantkę ważnego problemu naukowego i spełnia wymagania przewidziane dla rozpraw doktorskich w aktualnie obowiązującej Ustawie o Tytule Naukowym i Stopniach Naukowych. Stawiam więc wniosek o dopuszczenie jej Autorki do dalszych etapów przewodu doktorskiego.



Prof. dr hab. inż. Roman Wyrzykowski