

Częstochowa, dn. 19.10.2017

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

**RECENZJA  
ROZPRAWY DOKTORSKIEJ**

**mgr Łukasza Górskiego**

**„Adaptacja wybranych równoległych algorytmów do modelu PGAS i ich implementacja w języku Java z wykorzystaniem biblioteki PCJ”**

**Promotor: prof. dr hab. Piotr Bała**

**Interdyscyplinarne Centrum Modelowania  
Matematycznego i Komputerowego  
Uniwersytetu Warszawskiego**

**1. Obszar problemowy rozprawy**

Tematyka przedłożonej do recenzji rozprawy doktorskiej mgr Łukasza Górskiego dotyczy zagadnień programowania równoległego, obszaru niezwykle istotnego i dynamicznie rozwijającego się od lat. Szczególnego znaczenia tematyka ta nabrała, z jednej strony, w momencie wykrystalizowania się architektur wielordzeniowych jako podstawowego sposobu zwiększenia wydajności procesorów, a drugiej strony – w związku z rozwojem obliczeń wielkiej skali z wykorzystaniem systemów pozwalających na uzyskanie wydajności wielu Petaflops, a ostatnio zmierzających do przekroczenia „progu marzeń” – czyli 1 Exaflops. Jednakże skuteczne wykorzystanie olbrzymich możliwości drzemących w takich systemach wymaga stworzenia, niekiedy od podstaw, wielu nowych technologii i podjęcia szeregu wyzwań badawczych, począwszy od dogłębnego przeformułowania modeli matematycznych i związanych z nim algorytmów rozwiązywania zagadnień aplikacyjnych, a skończywszy (jeśli można się tak wyrazić) - na opracowaniu bardziej efektywnych narzędzi i środowisk programistycznych, w tym odpowiednich języków i bibliotek.

W tym kontekście bardzo pozytywnie oceniam wybór tematyki rozprawy zmierzający do pokazania dużych możliwości wiążących się z zastosowaniem języka programowania Java oraz biblioteki PCJ jako metody umożliwiającej

efektywne wdrożenie nowoczesnego modelu PGAS (ang. Partitioned Global Address Space) programowania równoległego. Jako rozwiązanie umożliwiające stworzenie wspólnej wirtualnej przestrzeni adresowej dostępnej dla bardzo wielu współdziałających wątków, model ten 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, język Java wraz z biblioteką PCJ to nie jedyna implementacja modelu PGAS dostępna w chwili obecnej. Spośród innych implementacji, udostępnianych np. przez języki X10, Chapel czy UPC, wyróżnia ją jedna, moim zdaniem niezwykle istotna zaleta. Jest nią możliwość głębokiej integracji tradycyjnej dziedziny aplikacji HPC, obejmującej głównie zagadnienia obliczeniowe, symulacyjne i optymalizacyjne, z inną niezwykle ważną dziedziną, które tworzą zagadnienia przetwarzania bardzo dużych zbiorów danych, kryjące się pod nośnym określeniem Big Data. W tej drugiej dziedzinie język Java zajmuje bowiem pozycję wręcz dominującą.

Podsumowując ten punkt recenzji, pragnę stwierdzić, iż wybór tematyki rozprawy uważam za bardzo trafny, a rozważane w niej problemy są istotne i aktualne, zarówno dla teorii, jak i przede wszystkim praktyki współczesnego programowania równoległego.

## **2. Koncepcja i redakcja rozprawy**

Recenzowana praca doktorska obejmuje formalnie krótki wstęp, 4 rozdziały i wnioski końcowe, a także bibliografię zawierającą 162 pozycje. Praca liczy łącznie 159 stron.

W *rozdziale pierwszym* Autor zawarł wprowadzenie do zagadnień programowania równoległego i rozproszonego, omawiając paradygmaty i modele programowania równoległego, w tym model PGAS, oraz charakteryzując język Java i bibliotekę PCJ jako narzędzie programowania równoległego.

*Rozdział drugi* poświęcono przedstawieniu metodologii przeprowadzonych w pracy badań, ze szczególnym uwzględnieniem metod oceny oprogramowania równoległego oraz sposobów pomiaru wydajności w praktyce z wykorzystaniem zestawów programów testowych (tzw. benchmarków). Rozdział ten kończy omówienie środowiska testowego opartego głównie na wykorzystaniu zasobów udostępnianych w ramach projektu PL-Grid.

**Rozdział trzeci**, wraz z kolejnym rozdziałem, zawiera autorską część pracy ukierunkowaną na opracowanie algorytmów umożliwiających dostosowanie reprezentatywnych zagadnień programowania równoległego do modelu PGAS i biblioteki PCJ. I tak w rozdziale tym skoncentrowano się na następujących problemach: dostępie swobodnym do pamięci, szybkiej transformacie Fouriera (jedno- i trójwymiarowej) oraz teście Linpack czyli rozwiązywaniu układów liniowych w oparciu o algorytm eliminacji Gaussa i rozkład macierzy współczynników do postaci LU. **W rozdziale czwartym** zaproponowano natomiast oryginalne rozwiązanie dla równoległej implementacji jednego z algorytmów ewolucyjnych czyli ewolucji różnicowej. Interesujące wydaje się również przedstawione opracowanie wykorzystania tego rozwiązania w pracach badawczych nad modelowaniem konektomu nicienia.

We rozdziale z **wnioskami** dokonano zwięzłego podsumowania pracy, a także wskazano kierunki 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 stosowalności modelu PGAS w języku Java. Rozszerza to spektrum alternatywnych rozwiązań dla programowania aplikacji równoległych wykorzystujących współczesne platformy HPC, ale również stanowi istotny krok w kierunku uzasadnienia możliwości głębokiej integracji tradycyjnej dziedziny aplikacji HPC, obejmującej głównie zagadnienia obliczeniowe, symulacyjne i optymalizacyjne, z inną niezwykle ważną dziedziną, które tworzą zagadnienia przetwarzania bardzo dużych zbiorów danych.
2. Wykazanie stosowalności modelu PGAS i biblioteki PCJ zrealizowano poprzez opracowanie, a następnie eksperymentalne zbadanie wydajności i skalowalności równoległych wersji algorytmów umożliwiających dostosowanie do tego modelu reprezentatywnych zagadnień programowania równoległego, takich jak: rozgłaszanie każdy z każdym, dostęp swobodny do pamięci, szybka transformata Fouriera w wersji jedno- oraz trójwymiarowej, a także zagadnienie optymalizacji w oparciu o algorytm ewolucji różnicowej. Skuteczność zrealizowanego dostosowania wynikała w szczególności z szerokiego zastosowania komunikacji asynchronicznej oraz nakładania obliczeń i komunikacji.

3. Na podstawie opracowanego algorytmu ewolucji różnicowej, dla potrzeb nauk biologicznych zaprojektowano i zaimplementowano dostatecznie złożone oprogramowanie autorskie przeznaczone do optymalizacji modelu konektomu. Wymagało to w szczególności opracowania i przebadania algorytmu równoważenia obciążenia w algorytmie ewolucyjnym.

Uzyskane wyniki zostały opublikowane w sześciu pracach w języku angielskim, które opublikowano w materiałach reprezentatywnych konferencji o zasięgu ogólnoświatowym, 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 pracy, które przytoczono w punkcie 2 niniejszej recenzji, wynika, iż 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: sprzętowych uwarunkowań wydajności i skalowalności oprogramowania równoległego, modeli programowania równoległego wraz z odpowiednimi standardami i środowiskami programistycznymi oraz wykorzystania benchmarków do oceny wydajności aplikacji równoległych. 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 aplikacji równoległych. Tę pozytywną ocenę wspiera również poziom wiedzy Autora w zakresie optymalizacji z wykorzystaniem meta-heurystyk, którą zaprezentował w rozdziale czwartym. Również bibliografia zawarta i wykorzystana w pracy nie budzi zastrzeżeń, a moja opinia o jej kompletności jest pozytywna.

#### **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 i standardu MPI. 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 dostępu do 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 i konkretnych klas programów, co z pewnością umożliwiłoby 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. W rozdziale trzecim Autor dostatecznie wyczerpująco zbadał zagadnienie dostosowania takich zagadnień jak rozgłoszenie każdy-do-każdego, dostęp swobodny do pamięci czy szybka transformata Fouriera. Na tym tle spory niedosyt pozostawiają studia nad adaptacją (jeśli w ogóle można użyć tego pojęcia akurat w tym przypadku) zagadnienia sprowadzania macierzy do postaci LU. Zagadnienie to należy do podstawowych w obliczeniach HPC i nie sprawia jakichś specjalnych trudności z punktu widzenia osiągnięcia wysokiej wydajności i skalowalności. Z tego powodu wykorzystywane jest z sukcesem do demonstracji możliwości współczesnych superkomputerów w teście Linpack. W ostatnim okresie test ten wręcz spotyka się z zarzutem, że jest zbyt mało skomplikowany i stawia architekturom współczesnych systemów zbyt małe wymagania, nie odpowiadające skali wyzwań jakie przynoszą inne zagadnienia, bliższe wielu krytycznym aplikacjom

użytkowników systemów HPC. Tymczasem wyniki Autora dla rozkładu LU są bardzo mało zachęcające, demonstrując skalowalność w zakresie do 4 czy 8 wątków. Nie byłoby to nawet tak niepokojące, jeśliby nie fakt, iż Autor nie próbuje nawet wyjaśnić takiego zachowania swoich implementacji i podać sposobów przezwyciężenia napotkanych trudności.

4. Autor dosyć szeroko stosuje pojęcia silnej i słabej skalowalności (*strong and weak scalability*). W związku z powyższym wydaje się, że w części pracy zawierającej *state-of-the-art* należałoby możliwie precyzyjnie zdefiniować oba te kluczowe pojęcia.
5. Na stronie 109, gdzie rozważane są możliwe podejścia do zrównoleglenia algorytmu ewolucji różnicowej, zaraz po przytoczeniu wad kilku znanych podejść Autor stwierdza, iż wprowadzony wcześniej Algorytm 6 pozwala przezwyciężyć napotkane przez innych trudności. Nie bardzo jednak wiadomo (Autor w każdym razie tego nie wyjaśnia), dzięki czemu udało się to osiągnąć. Samo wprowadzenie tego algorytmu w rozprawie nie zostało dokonane w najszcześniejszy sposób, gdyż algorytm odwołuje się do szeregu równań, które pojawiają się dopiero w dalszej części tekstu.

## 5. Podsumowanie

Przytoczone wyżej uwagi krytyczne nie umniejszają wysokiej wartości merytorycznej pracy, która stanowi istotny i oryginalny wkład Autora w rozwój metod i algorytmów organizacji obliczeń równoległych z wykorzystaniem nowoczesnych języków i narzędzi programistycznych, co pozwala ją jednoznacznie zakwalifikować do **dyscypliny Informatyka**.

Podsumowując recenzję, stwierdzam więc, że moja generalna opinia o pracy „Adaptacja wybranych równoległych algorytmów do modelu PGAS i ich implementacja w języku Java z wykorzystaniem biblioteki PCJ” jest zdecydowanie pozytywna. Uważam, że przedstawiona mi do recenzji praca zawiera samodzielne rozwiązanie przez doktoranta ważnego i trudnego problemu naukowego, co w pełni odpowiada wymaganiom stawianym rozprawom doktorskim przez odnośną ustawę o tytule i stopniach naukowych. Na tej podstawie wnioskuję o dopuszczenie pracy do publicznej obrony celem uzyskania przez Autora stopnia doktora nauk technicznych w zakresie informatyki.

Prof. dr hab. inż. Roman Wyrzykowski

