



prof. dr hab. Adam Roman
Kierownik Katedry Inżynierii Oprogramowania
Instytut Informatyki i Matematyki Komputerowej
Wydział Matematyki i Informatyki
Uniwersytet Jagielloński
Łojasiewicza 6, 30-348 Kraków
adam.roman@uj.edu.pl, www.ii.uj.edu.pl/~roman

Kraków, 22.05.2026

RECENZJA

osiągnięcia naukowego dr. Marka Nowickiego w związku z postępowaniem w sprawie nadania stopnia doktora habilitowanego

Niniejsza recenzja została wykonana na prośbę dr hab. Agnieszki Mykowieckiej, prof. IPI PAN, z-cy Dyrektora ds. naukowych IPI PAN, zawartą w piśmie NR.5120.1.2025.15 z dnia 7 kwietnia 2026 r.

Zgodnie z art. 219 ust 1 pkt. 1-3 ustawy Prawo o szkolnictwie wyższym i nauce, stopień doktora habilitowanego nadaje się osobie, która posiada stopień doktora, posiada w dorobku osiągnięcia naukowe lub artystyczne stanowiące znaczny wkład w rozwój określonej dyscypliny oraz wykazuje się istotną aktywnością naukową albo artystyczną realizowaną w więcej niż jednej uczelni, instytucji naukowej lub instytucji kultury, w szczególności zagranicznej. Dlatego niniejszą recenzję podzieliłem na trzy główne części, odpowiadające powyższym trzem wymogom ustawowym.

1. Posiadanie stopnia doktora (art. 219 ust. 1 pkt 1)

Dr Marek Nowicki posiada stopień doktora, nadany uchwałą Rady Wydziału Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego w dniu 18 czerwca 2015 r. na podstawie rozprawy „Opracowanie nowych metod programowania równoległego w Javie w oparciu o paradygmat PGAS (Partitioned Global Address Space)”. Spełnia zatem formalną przesłankę art. 219 ust. 1 pkt 1 ustawy Prawo o szkolnictwie wyższym i nauce.

2. Ocena osiągnięcia naukowego (art. 219 ust. 1 pkt 2)

2.1. Charakterystyka osiągnięcia naukowego

Jako osiągnięcie naukowe będące przedmiotem oceny w postępowaniu o nadanie stopnia doktora habilitowanego dr Nowicki wskazał cykl tematycznie powiązanych publikacji pt. „Paradygmat PGAS programowania równoległego w języku Java: rozwój i analiza biblioteki PCJ na przykładzie wybranych algorytmów, w tym bioinformatycznych”. W skład osiągnięcia wchodzi siedem prac opublikowanych w latach 2018-2025 (numeracja wg autoreferatu; w nawiasach podaję tzw. liczbę punktów ministerialnych czasopisma lub rangę konferencji wg Core Ranking):

- [1] – opublikowana w Journal of Computational Biology (70 pkt)
- [2] – z konferencji HPCS (70 pkt wg wykazu MNiSW, obecnie brak w Core Ranking, rank B w 2018 r.)
- [3] – samodzielna praca opublikowana w Journal of Big Data (140 pkt)
- [4] – opublikowana w Concurrency and Computation – Practice and Experience (100 pkt)
- [5] – opublikowana w Journal of Big Data (140 pkt)
- [6] – z konferencji ISGC (20 pkt, rank C)
- [7] – opublikowana w Briefings in Bioinformatics (140 pkt)

Prace [3] i [5] opublikowane są w bardzo dobrym czasopiśmie Journal of Big Data (IF = 6.4), a praca [7] w bardzo dobrym czasopiśmie bioinformatycznym Briefings in Bioinformatics, (IF = 7.7). Pozostałe cztery prace są opublikowane w czasopismach i wygłoszone na konferencjach niższej rangi. Praca [2] ma raczej walor dydaktyczny niż naukowy (patrz niżej).

Dr Nowicki złożył wniosek o przeprowadzenie postępowania w naukach technicznych, w dyscyplinie informatyka techniczna i telekomunikacja. Nie ulega wątpliwości, że dorobek naukowy oraz zakres i stosowana metodologia badań Habilitanta mieszczą się w obszarze nauk inżyneryjno-technicznych w zakresie informatyki technicznej (programowanie równoległe).

2.2. Ocena osiągnięcia naukowego

Osiągnięcie naukowe Habilitanta dotyczy biblioteki PCJ – otwartoźródłowej biblioteki dla języka Java służącej do wykonywania obliczeń równoległych w środowisku wielowęzłowym i wielordzeniowym, w ramach paradygmatu PGAS (Partitioned Global Access Space). Bibliotekę PCJ Habilitant rozwija nieprzerwanie od 2011 roku. Pierwsze trzy wersje biblioteki powstały przed uzyskaniem stopnia doktora i związane były z implementacją istotnych elementów architektonicznych.

W wersji 4 biblioteki Autor usunął własną ładowarkę klas. Praca [1] wykorzystuje bibliotekę w tej wersji do implementacji algorytmu BLAST dopasowywania sekwencji biologicznych, przy czym autorzy wykorzystali oryginalny algorytm NCBI-BLAST, a twórczy wkład Habilitanta polegał na jego zrównolegleniu przy użyciu biblioteki PCJ opierając się na rozdzieleniu sekwencji zapytań pomiędzy wykorzystywane węzły.

W wersji 5 biblioteki Autor przebudował mechanizm komunikacji między węzłami, opisując to jako „rewolucyjną” modyfikację. Ta wersja była wykorzystana w pracy [2], która jest tzw. „tutorial paper” na użytek warsztatów dla uczestników konferencji Int. Conf. on High Performance Computing & Simulation. Nie jest to więc publikacja stricte o charakterze naukowym, a bardziej dydaktycznym. Stanowi opis biblioteki PCJ na potrzeby szkolenia, natomiast warto zaznaczyć, że wspomniany warsztat odbył się na zaproszenie organizatorów konferencji.

W pracy [3] (jedynej samodzielnej w zgłoszonym cyklu publikacji) Autor przeprowadza badania w sposób podobny do tych w pracy [1]. Wykorzystuje bibliotekę PCJ do problemu sortowania wielkich zbiorów danych w środowisku wielowęzłowym. Tu z kolei PCJ porównywana jest z podejściem MapReduce w Hadoop. Eksperymenty zawierają szczegółową analizę czasu poświęconego na wykonywanie poszczególnych kroków wykorzystanego w pracy algorytmu sortującego TeraSort. W porównaniu z rozwiązaniem Hadoop podejście oparte na PCJ zarówno w trybie wewnętrznym (tj. gdy węzły obliczeniowe są umieszczone w tym samym miejscu co węzły z danymi) jak i w trybie zewnętrznym (tj. gdy węzły obliczeniowe są umieszczone w innym miejscu niż te z danymi) działa szybciej dla mniejszej liczby wątków, natomiast wyniki stają się zbliżone wraz ze wzrostem liczby wątków. Ponadto, oba rozwiązania dobrze się skalują, choć wraz ze wzrostem wolumenu danych uwidacznia się przewaga rozwiązania opartego na Hadoop.

Praca [4] opisuje wykorzystanie biblioteki PCJ w rozwiązaniu chmurowym (AWS EC2 oraz Linaro Developer Cloud) i porównuje jej działanie z rozwiązaniem typu HPC uzyskanym na superkomputerze Cray XC40. Porównanie obejmowało testy dla trzech problemów charakteryzujących się różnymi profilami działania: intensywny obliczeniowo algorytm szyfrowania DES, intensywny komunikacyjnie algorytm szybkiej transformaty Fouriera oraz przetwarzający duże zbiory danych algorytm WordCount.

Praca [5] jest podobna do prac [1], [3] i [4] w swojej zasadniczej idei: zbadać, jak biblioteka PCJ działa dla różnego typu problemów/architektur. Tym razem autorzy zbadali wykorzystanie PCJ porównując je z bibliotekami takimi jak APGAS, C/MPI, Java/MPI czy Horovod w problemach z obszaru *machine learning* i uczenia głębokiego (problem modelowania konektomu nicienia *C. Elegans* za pomocą algorytmu ewolucji różnicowej oraz rozproszone uczenie sieci neuronowej), a także w problemach testowanych wcześniej w pracy [4], mianowicie FFT i WordCount.

Praca [6] ma charakter inżynierski, opisuje bowiem implementację w PCJ tzw. operacji kolektywnych pozwalających na łatwe i niewymagające pisania dużej ilości kodu operacje koordynujące grupową wymianę danych między wieloma procesami. Poza już istniejącymi w bibliotece PCJ operacjami kolektywnymi *barrier*, *broadcast* i *scatter*, Habilitant zaimplementował operacje *reduce* i *collect* (z nazwą zmienioną na *gather* aby uniknąć kolizji ze standardową funkcją *collect* w Javie). Autorzy przeprowadzili również testy wydajnościowe (na wykorzystywanym już wcześniej problemie WordCount) dla nowo zaimplementowanych operacji, które pokazały poprawę wydajności dzięki wykorzystaniu w implementacji zrównoważonego drzewa binarnego optymalizacji polegającej na tzw. przesunięciu węzłów w drzewie komunikacji.

Ostatnia praca z cyklu, [7], opisuje rozwiązanie o nazwie HPV-KITE. Jest to aplikacja umożliwiająca szybkie przeszukiwanie zbiorów danych genomowych w celu identyfikacji w nich sekwencji wirusa HPV. Została ona dostosowana do wyników metod sekwencjonowania nowej generacji (NGS). Równoległość została zaimplementowana przy pomocy biblioteki PCJ. Rozwiązanie porównano ze standardowymi narzędziami stosowanymi do tego celu (Kraken2, Diamond), a wyniki świadczą na korzyść PCJ.

2.3. Szczegółowe uwagi polemiczne i pytania

W pracy [1] wykorzystano identyczne kopie wejścia aby zapewnić idealny load balancing, ale to niejako ułatwiło przeprowadzenie eksperymentu. Autorzy nie dyskutują sytuacji, w której wejścia mogą być dowolne i niewykluczone, że wtedy load balancing może stać się bardzo ważnym elementem przyczyniającym się do efektywności rozwiązania. Autorzy porównali swoje rozwiązanie z rozwiązaniem bazującym na podziale bazy referencyjnej¹ wykazując dwukrotnie lepszy czas w przypadku ich rozwiązania. Autorzy wyjaśniają to wykorzystaniem „dynamic load balancing mechanisms”, ale w pracy nie znalazłem żadnej informacji o tym, jakie konkretnie mechanizmy load balancingu zostały zaimplementowane. W pracy nie ma też porównania z innymi, analogicznymi do PCJ bibliotekami opartymi na paradygmacie PGAS. Nie jest to oczywiście wada metodologiczna pracy (wszak chodziło w

¹ Mikailov, M., Luo, F.-J., Barkley, S., et al. 2017. Scaling bioinformatics applications on HPC. BMC Bioinformatics 18, 501.

niej przede wszystkim o efektywne rozwiązanie konkretnego problemu bioinformatycznego), ale w kontekście wniosku habilitacyjnego byłoby nieco bardziej wartościowe poznawczo.

Autor wspomina, że w wersji 5 biblioteki wprowadził, jak sam twierdzi, „rewolucyjne” zmiany dotyczące przebudowy mechanizmu komunikacji między węzłami, jednak nie znalazłem ani w autoreferacie, ani w żadnej z załączonych publikacji, opisu tych zmian. W kontekście inżynierii oprogramowania oraz faktu, iż przedmiotem rozprawy habilitacyjnej jest oprogramowanie, jest to dosyć istotna kwestia i powinna być przedyskutowana. Autor wspomina co prawda o ujednoliceniu zachowania podstawowych metod biblioteki („usynchronizowaniu” ich), ale czyni to w kontekście opisu zmian w API (a przynajmniej tak wynika z logiki tekstu w autoreferacie), więc nie jestem pewien, czy to są te „rewolucyjne” zmiany, czy też polegają one jednak na czymś innym.

Wydaje mi się, że wyniki pracy [4] nie są specjalnie odkrywcze, ponieważ rozwiązania HPC dla problemów wymagających intensywnej wymiany danych (zwłaszcza takich jak algorytm FFT) będą lepsze niż rozwiązania chmurowe. Wyniki pokazują, że w przypadku wszystkich trzech użytych w pracy algorytmów wykorzystanie biblioteki PCJ w rozwiązaniach typu HPC były lepsze niż w przypadku rozwiązań chmurowych. Byłoby interesujące pokazać przykłady problemów, w których PCJ lepiej poradzi sobie w rozwiązaniach chmurowych. Ale, co ważniejsze, nie wiem czy przeprowadzone badania mówią coś o samej bibliotece PCJ; wydaje mi się, że raczej pokazują po prostu różnice w architekturze rozwiązań typu HPC oraz chmurowych (np. komunikacja między węzłami w chmurze będzie zawsze wolniejsza niż w przypadku pojedynczego superkomputera, gdyż wynika to po prostu ze świadomych decyzji architektonicznych obu tych rozwiązań, co wpływa na znaczące różnice w kwestii latencji czy przepustowości). Sama biblioteka PCJ wydaje się być w kontekście tych badań irrelevantna. Pytanie zatem, jaki tak naprawdę problem badawczy rozwiązywany jest w pracy [4], poza technicznym zadaniem zaprojektowania i zaimplementowania przez Autora mechanizmu uruchamiania zadań równoległych wykorzystujących bibliotekę PCJ na obu chmurach obliczeniowych?

W pracy [5] podano wyniki wydajnościowe dla problemu modelowania konektomu nicienia, ale nie znalazłem w niej wyników dotyczących *dokładności* znalezionej modelu, a to – jak rozumiem – było podstawowym problemem z punktu widzenia eksperta biznesowego (biofizyka).

W kilku miejscach pojawia się informacja, że biblioteka PCJ zdobyła w 2014 roku nagrodę „HPC Challenge award”, jednak dla porządku warto zaznaczyć, że biblioteka nie wygrała w kategorii „best performance”, lecz „most elegant”.

Habilitant wspomina, że istnieją rozwiązania bazujące na paradygmacie PGAS i cytuje w autoreferacie szereg takich rozwiązań (np. biblioteki SHMEM, Global Arrays, Charm++, APGAS). W pracach wchodzących w skład rozprawy również pojawiają się obszernie omówienia tych rozwiązań (np. w pracy [4]). Niestety, w żadnej z publikacji cyklu (poza jedynym porównaniem z APGAS w pracy [5]) nie znalazłem dokładniejszych rozważań, które porównywałyby proponowane przez Habilitanta rozwiązanie z istniejącymi, analogicznymi bibliotekami. Nie wiadomo zatem, czy jest ona efektywniejsza niż istniejące rozwiązania. Nie wiadomo też, na czym polegają istotne różnice w stosunku do istniejących rozwiązań, ani co było motywacją do przyjęcia takiej, a nie innej architektury biblioteki PCJ.

Wyniki eksperymentów opisanych w różnych pracach cyklu pokazują przewagę tej biblioteki nad innymi rozwiązaniami, co tym bardziej skłania do tego typu pytań.

2.4. Ocena znaczącego wkładu osiągnięcia w rozwój dyscypliny

Autor utrzymuje i rozwija bibliotekę przez bardzo długi czas – ponad 15 lat. Ta wytrwałość i konsekwencja jest godna podziwu. Repozytorium na GitHub dla projektu PCJ ma 22 gwiazdki, co nie jest może wartością imponującą w ogólnym przypadku, ale należy wziąć pod uwagę, że jest to projekt specjalistyczny, o charakterze naukowym, skierowany do określonej grupy użytkowników. Ponadto, z historii commitów wynika, że prawie wszystkie (z ponad 400) wykonał Habilitant, co ponad wszelką wątpliwość świadczy o tym, że jest głównym autorem rozwiązania. Biblioteka PCJ jest wykorzystywana nie tylko w badaniach naukowych, ale także w procesie dydaktycznym, m.in. na studiach Inżynierii Obliczeniowej w ICMMiK UW w ramach przedmiotu „Programowanie równoległe” oraz na studiach informatycznych na WMil UMK w ramach przedmiotu „Wprowadzenie do obliczeń równoległych”.

Moja zasadnicza wątpliwość wobec przedstawionego do oceny osiągnięcia dotyczy pewnej powtarzalności cyklu prac. Wchodzące w jego skład publikacje, zwłaszcza [1], [3], [4], [5], [7], w dużej mierze opierają się na podobnym schemacie badawczym: wykorzystaniu biblioteki PCJ i jej ewaluacji na różnych typach problemów oraz w odmiennych architekturach obliczeniowych. Zgodnie z ustawową definicją, osiągnięcie habilitacyjne powinno stanowić „znaczący wkład w rozwój dyscypliny”. W tym kontekście pojawia się pytanie, w jakim stopniu zestaw eksperymentów – choć niewątpliwie zróżnicowanych, rozbudowanych i szczegółowo opisanych – odnoszących się do jednej biblioteki i prowadzonych według podobnej metodologii, spełnia to kryterium. Opisane w kolejnych pracach modyfikacje i usprawnienia poszczególnych wersji biblioteki wydają się wynikać przede wszystkim z praktycznych obserwacji inżynierskich oraz stosowania dobrych praktyk w projektowaniu oprogramowania, takich jak mechanizmy fail-safe, obsługa wyjątków, ujednolicone interfejsy, dbanie o niskie sprzężenie komponentów, czy wykorzystanie wzorców projektowych. W przedstawionym cyklu prac nie jest wyraźnie widoczne, aby wnioski z przeprowadzonych eksperymentów w sposób bezpośredni przekładały się na istotne, koncepcyjne modyfikacje samej biblioteki PCJ. Jej rozwój można w dużej mierze postrzegać jako rezultat konsekwentnego stosowania uznanych zasad inżynierii oprogramowania. Dorobek Autora niewątpliwie potwierdza wysokie kompetencje w tym zakresie, a także w zakresie praktycznego wykorzystania narzędzi programistycznych. Jednocześnie można odnieść wrażenie, że prace polegające na wielokrotnym badaniu tej samej biblioteki w różnych scenariuszach testowych mają ograniczony potencjał generowania nowych treści poznawczych. Rodzi to pytanie, w jakim stopniu takie badania przyczyniają się do poszerzenia wiedzy w obszarze np. obliczeń równoległych, zwłaszcza jeśli podobne analizy można by przeprowadzać dla wielu innych problemów i konfiguracji systemowych.

Rozważając wszystkie argumenty za i przeciw, wziąłem pod uwagę w szczególności to, że dr Nowicki ponad wszelką wątpliwość jest głównym autorem biblioteki, jest to jego autorski projekt, rozwija go sukcesywnie od wielu lat, dotyczy on ważnych i praktycznych zagadnień (efektywność obliczeniowa), wyniki eksperymentalne pokazują niewątpliwie zalety jego rozwiązania nad innymi istniejącymi rozwiązaniami, a trzy spośród prac wchodzących w skład cyklu opublikowane zostały w bardzo dobrych

czasopismach; ponadto, pozostały dorobek dr. Nowickiego, mimo, że nie jest ulokowany w tak dobrych miejscach (patrz pkt 2.5 niżej), to jednak jest obszerny i świadczy o stabilnym, nieprzerwanym rozwoju i aktywności naukowej Habilitanta. Dodatkowo, należy docenić stały rozwój biblioteki, w którym widać też dbałość o stronę architektoniczną rozwiązania i stosowanie dobrych praktyk i zasad inżynierii oprogramowania w projektowaniu architektury.

Te argumenty przeważają na korzyść pozytywnej oceny co do „znaczącego wkładu w rozwój dyscypliny”, aczkolwiek chciałbym, aby podczas kolokwium habilitacyjnego Habilitant odniósł się do mojej powyższej uwagi dotyczącej „monotonii” wyników.

2.5. Ocena pozostałego dorobku naukowego

Z autoreferatu wynika, że poza cyklem publikacji przedłożonych jako osiągnięcie habilitacyjne dr Nowicki jest autorem 6 prac przed uzyskaniem stopnia doktora, 19 prac opublikowanych po uzyskaniu stopnia doktora oraz 12 prac o charakterze dydaktycznym. Wśród prac opublikowanych po uzyskaniu stopnia doktora są prace: [A2] z konferencji ICA3PP (core rank C), [A3] z EuroPar (core rank B) oraz [A19] opublikowana w European Journal of Human Genetics (100 pkt MNiSW). Mimo obszernej liczby prac, poza wspomnianymi trzema nie udało mi się wśród nich znaleźć innych znajdujących się na liście konferencji Core Rank lub w czasopismach indeksowanych na tzw. „liście ministerialnej”. Oznacza to, że – wyłączając cykl habilitacyjny – Habilitant zdaje się nie publikować swoich prac na dobrych konferencjach, które zwiększałyby szansę dostrzeżenia jego dorobku wśród społeczności ekspertów z obszaru programowania równoległego czy high performance computing.

2.6. Uwagi do autoreferatu

Autoreferat jest napisany poprawnie, przejrzysty, w sposób czytelny i zrozumiały. Z recenzenckiego obowiązku odnotowuję zauważone drobne błędy w autoreferacie (s. 4: udana realizacji -> udana realizacja, s. 8: rozwiązanie (...) wykazał -> rozwiązanie (...) wykazało, s. 19: rozwiązania oparte o bibliotekę -> rozwiązania oparte na bibliotece, s. 20: uruchomienie (...) na 200,000 wątków -> uruchomienie (...) na 200,000 wątkach, s. 26: dla rozmiaru (...) wynoszącym -> dla rozmiaru (...) wynoszącego).

Błędy te nie w żaden sposób nie wpływają na moją merytoryczną ocenę osiągnięcia.

3. Ocena aktywności naukowej (art. 219 ust. 1 pkt 3)

Dr Nowicki poza pracą na Uniwersytecie Mikołaja Kopernika jest również zatrudniony na Uniwersytecie Warszawskim. Dodatkowo, odbył także dwa staże zagraniczne: w ośrodku superkomputerowym HLRS w University of Stuttgart w Niemczech (czerwiec-lipiec 2017), oraz dłuższy, kilkumiesięczny, przed uzyskaniem stopnia doktora, w IBM Zurich Research Laboratory (maj-listopad 2013). Dr Nowicki brał także udział w czterech międzynarodowych projektach badawczych. Dwa z nich miały miejsce przed uzyskaniem stopnia doktora. Po uzyskaniu stopnia doktora był wykonawcą w grantie NCN oraz kierownikiem grantu w ramach projektu Horizon 2020. Był uczestnikiem wielu konferencji krajowych i zagranicznych, w tym kilkakrotnie jako *invited speaker*.

W ramach działalności organizacyjnej pełnił funkcję członka komitetu naukowego warsztatów Workshop on PGAS Programming (2019) oraz członka komitetu organizacyjnego konferencji Petri Nets 2016. Dr Nowicki pełnił funkcję promotora pomocniczego w jednym przewodzie doktorskim oraz recenzenta dla czasopism Bioinformatics, BMC Bioinformatics, Journal of Big Data, Software X. Prowadził wiele specjalistycznych zajęć dydaktycznych (takie jak inżynieria wsteczna oprogramowania, wprowadzenie do obliczeń równoległych, programowanie równoległe i rozproszone), był promotorem kilkunastu prac inżynierskich i magisterskich, prowadził warsztaty popularyzujące naukę na festiwalach naukowych, był zastępcą trenera drużyny na Akademickich Mistrzostwach Polski w Programowaniu Zespołowym, pełnił funkcję sędziego w różnorakich konkursach, hackatonach i olimpiadzie informatycznej. Jest laureatem kilku indywidualnych nagród Rektora UMK oraz zwycięzcą konkursu IBM Great Minds student internship.

Powyższe fakty świadczą o tym, że dr Nowicki jest aktywny naukowo i w mojej ocenie z nadwyżką spełnia wymagania w zakresie działalności organizacyjnej, dydaktycznej i popularizatorskiej.

Nie ulega zatem wątpliwości, że dr Nowicki spełnia przesłanki art. 219 ust. 1 pkt 3 ustawy.

4. Konkluzja

Podsumowując, całość dorobku dr. Marka Nowickiego oceniam pozytywnie. Mimo wyrażonych pewnych wątpliwości, biorąc pod uwagę wszystkie przedstawione powyżej argumenty uważam, że **osiągnięcia Habilitanta odpowiadają wymaganiom określonym w art. 219 ust. 1 pkt 2 ustawy i w związku z tym wnioskuję do Rady Naukowej Instytutu Podstaw Informatyki PAN o dopuszczenie dr. Nowickiego do dalszych etapów postępowania o nadanie stopnia doktora habilitowanego.**

Adam Roman



Podpisany elektronicznie przez
Adam Roman
22.05.2026
18:50:59 +02'00'