

Prof. dr hab. inż. Marek Tudruj
Instytut Podstaw Informatyki PAN
Polsko-Japońska Akademia Technik Komputerowych
tudruj@ipipan.waw.pl

Warszawa, 22 listopada 2016

Recenzja rozprawy doktorskiej

Autor: mgr inż. Konrad Siek

Tytuł: Distributed Pessimistic Transactional Memory: Algorithms and Properties

Przedstawiona mi do recenzji rozprawa doktorska mgr inż. Konrada Sieka zawiera 268 stron i składa się ze Streszczenia w języku angielskim, Spisu treści, Listy publikacji, Tabeli symboli, Wstępu, 8 podstawowych rozdziałów, Wniosków, Streszczenia w języku polskim, Bibliografii oraz 2 Dodatków. Dodatki zawierają dowody zawartych w rozprawie lematów i twierdzeń oraz pseudokody opracowanych algorytmów. Do rozprawy dołączono dysk CD, na którym zapisano tekst rozprawy.

1. Problem badawczy i jego znaczenie

Niniejsza rozprawa dotyczy problematyki pamięci transakcyjnej, stanowiącej specyficzne podejście do problemu automatyzacji i synchronizacji w programowaniu współbieżnym opartym na wykorzystaniu dostępu do danych przechowywanych w pamięciach współdzielonych. Rozdział 2 rozprawy wprowadza definicje podstawowych pojęć używanych w rozprawie. Przy wykorzystaniu pamięci transakcyjnej komponenty programu współbieżnego (wątki) realizują wzajemną komunikację oraz synchronizację poprzez wykonywanie transakcji na danych współdzielonych przechowywanych w pamięci. Transakcja dla pamięci transakcyjnej stanowi zbiór operacji na danych współdzielonych, który spełnia własności atomowości, spójności, izolacji i trwałości (w języku angielskim ACID od Atomicity, Consistency, Isolation and Durability) w sensie podobnym do ekwiwalentnych własności zdefiniowanych w dziedzinie baz danych. Transakcja stanowi więc niepodzielny zespół operacji, który może zakończyć się sukcesem (ang. commit) albo niepowodzeniem (ang. abort), stwierdzonym w wyniku wykonanego zatwierdzenia wyników (ang. commitment). Wyniki transakcji zakończonych sukcesem stają się widoczne dla współużytkowników danej pamięci transakcyjnej, natomiast w przypadku niepowodzenia są one pomijane a stąd niewidoczne. Współbieżne wykonywanie transakcji stanowi więc pewien atrakcyjny paradygmat realizacji aplikacji współbieżnych wykonywanych na współdzielonych strukturach danych we współczesnych systemach rozproszonych, pozwalający na uproszczenie programowania. Opracowanie wysoko wydajnych metod realizacji rozproszonych pamięci transakcyjnych stanowi istotny niebanalny problem naukowy o dużym znaczeniu praktycznym.

We współczesnej metodologii rozproszonej pamięci transakcyjnej wyróżnia się dwa podejścia do sterowania współbieżnością wykonywania transakcji. Pierwsze podejście to tzw. optymistyczne sterowanie współbieżnością. Polega ono na tym, że zezwala się na współbieżne spekulatywne wykonywanie transakcji na współdzielonych strukturach danych, nie analizując wstępnie ich wewnętrznej wzajemnej poprawności, pozostawiając tę analizę fazie zatwierdzania transakcji. W przypadku konfliktowego wykonania, naruszającego reguły poprawności transakcji współbieżnych, transakcja niepoprawna jest wycofywana i jej wyniki są usuwane. Jest to zdecydowana wada podejścia optymistycznego, która bardzo negatywnie wpływa na wydajność obliczeń. Drugim podejściem do sterowania transakcjami współbieżnymi jest podejście pesymistyczne, przy którym warunki poprawności wykonywania transakcji współbieżnych na współdzielonych danych są analizowane przed wykonaniem operacji transakcji. W przypadku konfliktów odnośnie poprawności dostępu do zmiennych/obiektów dzielonych, operacje są opóźniane do momentu gdy przyczyny konfliktu przestają istnieć. Dzięki temu możliwe jest zastosowanie tzw. wczesnego uwalniania zmiennych, które polega na tym, że dostęp do zmiennej współdzielonej jest umożliwiany danej transakcji przed zatwierdzeniem innej transakcji, która tę zmienną modyfikuje. Wynikiem jest możliwość zwiększenia wydajności pesymistycznego modelu sterowania w porównaniu do modelu, w którym dostęp byłby umożliwiany dopiero po zatwierdzeniu całej transakcji lub, jak jest w przypadku modelu optymistycznego pamięci transakcyjnej, gdy wykonywanie transakcji jest obarczone ryzykiem wycofywania niepoprawnych transakcji spekulatywnych.

We Wstępie rozprawy określono bezpośrednio tezę, która w przetłumaczeniu na język polski jest następująca:

„Możliwe jest zaproponowanie pesymistycznego algorytmu sterowania współbieżnością dla rozproszonej pamięci transakcyjnej, który jednocześnie:

- a) osiąga wysoką wydajność,
- b) spełnia silne własności bezpieczeństwa, żywotności i postępu,
- c) gwarantuje poprawne wykonanie operacji niewycofywalnych,
- d) jest stosowalny w ogólnych modelach systemów.”

Teza dotyczy dobrze określonego problemu naukowego i dobrze określa cele do osiągnięcia. W szczególności w rozprawie zaproponowano rozwiązania, które mają duże znaczenie praktyczne, gdyż eliminują wady optymistycznego podejścia do zarządzania pamięcią transakcyjną.

2. Wkład autora

Rozprawa ma częściowo charakter teoretyczny jak i praktyczny. Podobnego typu jest przedstawiony w niej wkład Doktoranta do rozwoju metodologii rozproszonej pamięci transakcyjnej. Wkład teoretyczny autora polega na opracowaniu szeregu aspektów teorii i algorytmów sterowania, udoskonalających technikę realizacji rozproszonych pamięci transakcyjnych. Wkład praktyczny polega na implementacji zaproponowanych algorytmów i systemów programowych oraz ich zbadanie.

Pierwszym z aspektów teoretycznych jest przedstawiona w rozdziałach 3 i 4 analiza stosowalności gwarancji bezpieczeństwa pamięci transakcyjnych, wynikających z bazodanowego dziedzictwa podejścia transakcyjnego oraz analiza stosowalności warunków spójności do operacji w rozproszonych pamięciach transakcyjnych z wczesnym zwalnianiem zmiennych (zasobów). Stosowalność tych gwarancji i warunków spójności została zweryfikowana przez wykonanie analizy implementacji wybranych pamięci transakcyjnych z pesymistycznymi strategiami sterowania z wczesnym zwalnianiem zasobów. Pozwoliło to na określenie parametrów i warunków bezpieczeństwa poprawnego stosowania tych strategii w pamięciach transakcyjnych. Dodatkowym efektem wykonanej analizy było wytypowanie algorytmów i metod użytecznych dla implementacji pesymistycznych rozproszonych pamięci transakcyjnych. Rozdział 3 zawiera analizę znanych z literatury własności baz danych i pamięci transakcyjnych dotyczących problemu bezpieczeństwa ich działania (serializability, commitment order preservation, recoverability, cascadelessness, strictness, opacity, markability, rigorousness, virtual word consistency, live opacity, elastic opacity). Rozdział kończy się podsumowaniem, w którym przedstawiono w postaci tabeli czy badane własności dotyczą baz danych czy pamięci transakcyjnych oraz czy wspierają wczesne zwalnianie zasobów, nadpisywanie i przerywanie wczesnego zwalniania jak również czy są one równie skuteczne jak serializability.

Rozdział 4 zawiera przegląd znanych z literatury algorytmów sterowania współbieżnością poprzez wykorzystanie pamięci transakcyjnych. Dla pesymistycznej rozproszonej pamięci transakcyjnej omówiono szereg odmian algorytmu 2PL - Two-Phase Locking opartego na wykorzystaniu zamków; dwie odmiany algorytmu wersjonowania: Basic – BVA oraz Supremum – SVA, kontrolujące spójność wartości współdzielonych zmiennych i obiektów, poprzez przypisywanie im indeksów wersji przyjmowanych wartości, z możliwością wczesnego uwalniania danych oraz z wiedzą o górnej granicy liczby dostępów (Supremum SVA). Dla optymistycznej rozproszonej pamięci transakcyjnej opisano najpierw algorytm DTL2 - Distributed Transactional Locking II, stanowiący rozproszoną wersję algorytmu TL2, charakteryzujący się użyciem zamków dla synchronizacji, wersjonowania danych oraz globalnego zegara dla określania ważności operacji w transakcjach. Następnie opisano algorytm TFA – Transaction Forwarding Algorithm, dedykowany dla modelu sterowanego przepływem danych, w którym operacje na współdzielonych obiektach są delegowane do sztywno alokowanych transakcji. Kolejne opisane algorytmy dotyczą pesymistycznych nierozproszonych pamięci transakcyjnych. Omówiono tu algorytm MS-PTM: Pessimistic Transaction Machine by Matveev and Shavit., oparty na DTL2, stanowiący pamięć transakcyjną bez odwoływania i z zatwierdzaniem w fazie commit. Następnie omówiono algorytmy PLE – Pessimistic Lock Elision, opierający się na elizji ochrony wersjonowanego dostępu do starych i nowych wersji w buforach zmiennych współdzielonych oraz SemanticTM – w pełni pesymistyczny algorytm pamięci transakcyjnej dla dostępu do zmiennych przy równoległości na poziomie instrukcji. Końcowym algorytmem opisanym w rozdziale 4 jest DATM – Dependency Aware TM stanowiący optymistyczny algorytm dla nierozproszonej pamięci transakcyjnej z wykorzystaniem zatwierdzania w fazie commit

ale rozszerzony o wczesne zwalnianie zasobów. Rozdział kończy się podsumowaniem, w którym przedstawiono w postaci tabeli zbiorcze porównanie przedstawionych w nim algorytmów ze względu na 10 wyróżnionych cech. Ponieważ różnice w zachowaniu się poszczególnych algorytmów nie są wyrażone w kategoriach spełnianych w tych algorytmach własności bezpieczeństwa, Doktorant sformułował wniosek, że brak jest dobrych ogólnych własności bezpieczeństwa dla pamięci transakcyjnych z wczesnym zwalnianiem zasobów. Innym aspektem dodatkowo ocenionym jest stosowalność przedstawionych algorytmów w systemach rozproszonych. Ogólny wniosek z wykonanej analizy jest taki, że poszczególne algorytmy w bardzo różnym stopniu spełniają powyższe wymaganie oraz, że spośród tych, które je spełniają tylko algorytmy wersjonowania oraz oparte na wykorzystaniu zamków wspierają obecność w transakcjach operacji nieodwracalnych.

Następnym aspektem teoretycznym, stanowiącym oryginalny wkład doktoranta, w metodologii pamięci transakcyjnej są nowe własności bezpieczeństwa działania pamięci transakcyjnych z wczesnym zwalnianiem zasobów. Są to własności: Last Use Opacity - nieprzezroczystość do ostatniego użycia oraz Strong Last Use Opacity - silna nieprzezroczystość do ostatniego użycia. Są one przedstawione w rozdziale 5 rozprawy, który jest jednym z podstawowych rozdziałów, zawierającym podstawy teoretyczne dla zaproponowanej przez Doktoranta metody realizacji pesymistycznej rozproszonej pamięci transakcyjnej. Zbadane dwie wersje nieprzezroczystości do ostatniego użycia umożliwiające odczyt danych współdzielonych zapisanych przez niezatwierdzone transakcje pod warunkiem, że zapis był ostatnim zapisem na danej zmiennej dokonany przez transakcje współdzielące dane. Obydwie wersje umożliwiają odczyty współdzielonych danych z wczesnym ich zwalnianiem przed zatwierdzeniem transakcji ale po dokonaniu najpóźniejszego zapisu tych danych. Podana własność nie zabezpiecza jednak przed tym, że po dokonaniu odczycie transakcja, która dokonała zapisu może być odwołana wprowadzając stan niespójności w realizacji współbieżnych transakcji na współdzielonych danych. Dlatego wprowadzono wersję silnej nieprzezroczystości do ostatniego zapisu, która wymaga aby po odczycie zmiennej współdzielonej bezpiecznej pod względem kolejności zapisów spełniona była dodatkowa własność braku odwołania transakcji, która dokonała ostatniego zapisu. Zaproponowana silna własność eliminuje niespójne stany odczytywanych współdzielonych danych w pamięci. Daje ona pełną gwarancję bezpieczeństwa dla wczesnego zwalniania zasobów pozwalającego na redukcję czasu realizacji transakcji w pamięci.

Kolejnym aspektem teoretycznym stanowiącym wkład Doktoranta do rozwoju metodologii pamięci transakcyjnej są opracowane przez niego nowe algorytmy sterowania współpracą elementów składowych programów współbieżnych, dla rozproszonej pamięci transakcyjnej z pesymistycznym podejściem do współbieżnej realizacji transakcji. Te algorytmy są przedstawione w rozdziale 6 rozprawy. Bazują one na nowych własnościach dla bezpieczeństwa współbieżnego dostępu do danych w pamięciach transakcyjnych wprowadzonych w rozdziale 5. Opracowano szereg nowych zoptymalizowanych algorytmów rozszerzających algorytmy wersjonowania (SVA) o wykorzystanie podejścia silnej nieprzezroczystości do ostatniego użycia. W podrozdziale 6.2 wprowadzono

rozszerzenie algorytmów z wersjonowaniem w kierunku uwzględnienia uzasadnionej potrzeby programowanego wycofywania transakcji. W podrozdziale 6.3 przedstawiono wymienione poniżej algorytmy dla heterogenicznego modelu systemu dla danych typu obiekt-zmienna oparte na rozszerzonym algorytmie SVA. Są to: algorytm (SVA+R - Supremum Versioning Algorithm with Rollback) uwzględniający kontrolowane wycofywanie transakcji przy obecności odtwarzania obiektów, kontrolowanie porządku zatwierdzania transakcji oraz kaskadowe wycofywanie transakcji; algorytm (Reluctant Supremum Versioning Algorithm with Rollback - RSVA+R) z kontrolowaniem transakcji z operacjami opornymi na kaskadowe wycofywanie; algorytmy (Optimized Supremum Versioning Algorithm with Rollback - Opt SVA +R, R Opt SVA +R) z buforowaniem efektów operacji o specjalnej semantyce niewidocznej na zewnątrz transakcji (uwzględniające transakcje tylko do odczytu, odróżnianie supremów dla zapisu i odczytu, opóźnione synchronizacje przy pierwszym odczycie, wczesne zwalnianie przy ostatnim zapisie). W podrozdziale 6.4 przedstawiono algorytm dla rozproszonych pamięci transakcyjnych dla modelu programowego sterowanego przepływem sterowania (control flow driven) o nazwie (Optimized Supremum Versioning Algorithm with Rollback - Opt SVA - CF +R) przeznaczony dla bardziej uniwersalnego modelu systemu, gdzie nie jest znana dokładna charakterystyka obiektów, na których transakcje operują, w szczególności obiektów zawierających wiele zmiennych, o złożonych interfejsach dostępu. Aby zachować możliwie ogólny charakter rozważań, charakterystykę obiektów ograniczono do trzech typów operacji stanowiących wykonanie pewnych programów: uogólnionego odczytu – zwracającego wartość ale nie modyfikującego stanu obiektu, uogólnionego zapisu – wykonującego zmianę stanu obiektu bez jego czytania i zwracania wartości oraz aktualizacji - mogącej odczytywać i modyfikować stan obiektu i zwracać wartość. Do tak zdefiniowanych operacji zaadaptowano realizację algorytmów przedstawionych w podrozdziale 6.3 oraz wykonywane optymalizacje.

Wykonano badania porównawcze charakterystycznych cech 9 algorytmów opisanych w rozdziale 6, przedstawiając wyniki porównania w postaci odpowiedniej tabeli.

Kolejnym aspektem teoretycznym opracowanym przez Doktoranta są podane w rozdziale 7 dowody własności opracowanych nowych algorytmów przedstawionych w rozdziałach 5 i 6. Dowody te są oparte na wykorzystaniu użytych przez Doktoranta właściwych metod dowodzenia. Pozwalają one na analizę bezpieczeństwa działania opracowanych nowych algorytmów w sensie nieprzezroczystości i nieprzezroczystości do ostatniego użycia.

Bardzo istotnym aspektem praktycznym rozprawy jest implementacja programowa opracowanych algorytmów omówiona w rozdziale 8. Implementacja dotyczyła przede wszystkim dwu ogólnych rodzajów algorytmu wersjonowania: SVA+R oraz Opt SVA – CF + R z uwzględnieniem instancji dotyczących transakcji z operacjami opornymi na kaskadowe wycofywanie transakcji. Podane są opisy pseudokodów algorytmów, interfejsu programisty oraz specjalnych mechanizmów zastosowanych w tych algorytmach, aby zrealizować zaproponowane optymalizacje.

Dla implementacji zaproponowanych algorytmów użyto dwu opracowanych z udziałem Doktoranta środowisk programowania transakcyjnego w języku Java: Atomic RMI (dla SVA + R) oraz Atomic RMI 2 (dla Opt SVA – CF + R), opierających się na wykorzystaniu mechanizmów Java RMI – Remote Method Invocation. Każdy algorytm został porównany z innymi rozwiązaniami mechanizmów sterowania współbieżnością w systemach rozproszonych. Mierzono wydajność wymienionych wyżej algorytmów sterowania pamięcią transakcyjną jako uzyskiwaną liczbę operacji na sekundę w funkcji liczby klientów pamięci transakcyjnej, liczby węzłów w systemie wykonawczym oraz procentowego udziału operacji zapisu i odczytu w transakcjach. Wyniki przedstawiono graficznie w postaci wykresów. Wyniki uzyskane w środowiskach Atomic RMI oraz Atomic RMI 2 porównano z wynikami pochodzącymi przede wszystkim z optymistycznego systemu rozproszonej pamięci transakcyjnej HYFlow2 o wysokiej jakości działania. Dla eksperymentów z algorytmem SVA + R i środowiskiem Atomic RMI użyto szeregu mieszanek (distributed hash table, bank, loan and vacation) różniących się stopniem złożoności transakcji oraz użyto 10-węzłowego klastra zawierającego w węźle 2 czterordzeniowe procesory Intel Xeon L3260. Dodatkowo do HyFlow DTL2 wyniki algorytmu SVA + R porównano z wynikami standardowego Java RMI dla algorytmów z wzajemnie wykluczającymi się zamkami oraz z prywatnymi zamkami czytaj/zapisz, dla zawartości odczytów/zapisów w transakcji w proporcjach 20/80% i 80/20%. Dla eksperymentów z algorytmem Opt SVA – CF + R i środowiskiem Atomic RMI2 użyto mieszanki EigenBench typowej dla testowania rozproszonych pamięci transakcyjnych oraz 16-węzłowego klastra z procesorami Intel Xeon L3260. Programowano w języku Groovy, stanowiącym obiektowy język skryptowy wzorowany na składni Java z wykonaniem na maszynach wirtualnych Java. Wyniki algorytmu Opt SVA – CF + R porównano z wynikami algorytmów z operacjami nierozróżnialnymi lub czytaj/zapisz z dwufazowym blokowaniem zrealizowanym na zamkach lokalnych oraz z wynikami uzyskanymi przy zastosowaniu zamków globalnych. Dla eksperymentów ze środowiskiem Atomic RMI zachowanie algorytmu w funkcji liczby węzłów systemu silnie zależało od testowanej mieszanki. Największą przewagę wydajności SVA + R w środowisku Atomic RMI uzyskano dla mieszanki loan, bank oraz vacation. Najlepszą wydajność Opt SVA – CF + R uzyskano dla wszystkich wykonań w środowisku Atomic RMI2, najgorszą - dla wykonań z użyciem globalnych zamków. Wyniki uzyskiwane dla środowiska Atomic RMI2 były zawsze lepsze od wyników pochodzących z wykonań w optymistycznej pamięci transakcyjnej. Świadczy to o przewadze opracowanych algorytmów pesymistycznych nad algorytmami optymistycznymi, popartej zbadaniem ich rzeczywistych implementacji.

Rozdział 9 rozprawy zawiera opis dodatkowych narzędzi wykorzystanych w badaniach opracowanych algorytmów zapewniających automatyzację pozyskiwania parametrów związanych z wykonywaniem transakcji pod kontrolą algorytmów z wersjonowaniem. Są to: prekompilator kodu transakcji oraz moduł generujący instrumentację kodu transakcji dla wykonania z użyciem supremów liczb dostępu do obiektów. Prekompilator statycznie analizuje kod transakcji używając podejścia sterowanego przepływem danych i określa liczbę obiektów dostępowanych w transakcjach oraz

supremum liczby dostępów wykonywanych na tych obiektach. W trakcie analizy wykorzystywany jest język pośredni Jimple w środowisku Soot. Następnie kod transakcji jest transformowany dla wykonań pod kontrolą algorytmów typu SVA.

Podsumowując można stwierdzić, że cele naukowe rozprawy zarówno teoretyczne jak i praktyczne zostały w pełni osiągnięte. Wyniki eksperymentalne rozprawy stanowią potwierdzenie, że możliwe jest opracowanie rozproszonej pamięci transakcyjnej opartej na pesymistycznym podejściu do algorytmów sterowania, które zapewniają lepszą wydajność wykonywania transakcji niż podejście optymistyczne, skuteczny algorytm dla bezpiecznego wykonywania transakcji sterowanych tym algorytmem bez zakleszczeń, z zachowaniem żywotności i progresywności dzięki zastosowaniu własności nieprzezroczystości do ostatecznego użycia. Podejście to eliminuje wycofywanie transakcji, rozwiązując problem poprawności wykonania transakcji niewycofywalnych, zapewnia działanie na niejednorodnym modelu obiektowym, obsługiwanym z użyciem informacji o profilu wykonania transakcji, generowanych przed ich wykonaniem.

3. Poprawność

Nie zauważyłem w pracy błędów merytorycznych w zakresie proponowanych koncepcji algorytmów sterowania pamięciami transakcyjnymi. Poprawność zawartych w pracy koncepcji jest w przeważającej części uzasadniona przez zamieszczenie odpowiednich dowodów. Nie widzę też większych wad w konstrukcji i prezentacji rozprawy. Struktura rozprawy jest skonstruowana z dużą dbałością o łatwość jej odbioru. Rozprawa jest napisana dobrym i poprawnym językiem. Terminologia używana w tekście angielskim jest zgodna z literaturą problemu. Uważam więc rozprawę za bardzo cenne opracowanie anglojęzyczne w dziedzinie metodologii pamięci transakcyjnych.

Bardzo ważne jest zamieszczone w rozprawie obszerne Streszczenie w języku polskim zawierające 32 strony. Podano w nim omówienie zawartości merytorycznej rozprawy, przyczyniając się do tworzenia polskiej terminologii fachowej i naukowej w dziedzinie pamięci transakcyjnych. Nieco problemów dostarcza polska terminologia zawarta w tym streszczeniu. W szczególności dotyczy to centralnego pojęcia rozprawy, które jest określone angielskim słowem „opacity”. Według słownika Jana Stanisławskiego słowo „opacity” może być tłumaczone na język polski w różny sposób: jako nieprzezroczystość w odniesieniu do przepuszczania światła przez różne materiały; jako mętność (niejasność) w odniesieniu np. do tekstu lub płynu oraz tępota w odniesieniu np. do umysłu. W technice programowania „opacity” jest przeciwieństwem do „transparency”, która może oznaczać niewidoczność pewnych operacji z punktu widzenia programisty pomimo ich istnienia w rzeczywistym systemie, osiągalną np. poprzez virtualizację użytych zasobów. W kontekście niniejszej rozprawy, „opacity” mogłoby być tłumaczone w sposób bardziej samo wyjaśniający na „nieokreśloność” lub „niejasność” w odniesieniu do bezpieczeństwa odczytu współdzielonych zmiennych. Oczywiście ta propozycja powinna być skonfrontowana z zaszłościami i dotychczasową

tradycją dla terminologii w języku polskim w dziedzinie metod obliczeń współbieżnych i pamięci transakcyjnej. Dlatego niniejsza uwaga ma charakter dyskusyjny i nie umniejsza wartości merytorycznych rozprawy.

4. Wiedza kandydata

Rozprawa zawiera bibliografię zawierającą 105 pozycji. Stanowi ona bardzo dobrą podstawę dla analizy stanu wiedzy w dziedzinie metod realizacji rozproszonych pamięci transakcyjnych. Poważna część rozprawy (rozdziały 2, 3, 4) zawiera bardzo starannie wykonany przegląd stanu wiedzy w dziedzinie rozprawy. Jakość tego przeglądu jest bardzo wysoka. Uwzględnia on moim zdaniem wszystkie istotne aspekty problemu rozpatrywanego w rozprawie, poparte stosownymi pozycjami literatury i analizą zamieszczoną w rozprawie. Wykonany przegląd literatury świadczy o bardzo dobrej wiedzy Doktoranta w dziedzinie rozprawy i stanowi dobre tło dla nowych opracowanych i zbadanych rozwiązań. W całą pewnością można stwierdzić, że Doktorant posiada bardzo dobrą ogólną wiedzę w dyscyplinie informatyka i umiejętność poprawnego prowadzenia badań naukowych.

5. Inne uwagi¹

Bardzo wysoko oceniam przydatność niniejszej rozprawy dla rozwoju metod informatyki. Jest to przykład bardzo rzetelnej pracy naukowej, której wyniki mogą znaleźć szerokie zastosowanie praktyczne w technologii obliczeń współbieżnych. Na szczególne podkreślenie zasługuje fakt, że badania wykonane w ramach niniejszej rozprawy łączą w sobie wykorzystanie elementów o charakterze teoretycznym z elementami praktyki informatycznej.

6. Podsumowanie

Biorąc pod uwagę opinie zaprezentowane w poprzednich punktach i wymagania zdefiniowane przez artykuł 13 Ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym (z późniejszymi zmianami)² moja ocena rozprawy pod względem trzech podstawowych kryteriów jest następująca:

A. Czy rozprawa zawiera oryginalne rozwiązanie problem naukowego? (wybierz jedną opcję stawiając znak X)

Zdecydowanie TAK

Raczej TAK

Trudno powiedzieć

Raczej NIE

Zdecydowanie NIE

B. Czy po przeczytaniu rozprawy zgadzasz się, że kandydat posiada ogólną wiedzę teoretyczną w dyscyplinie Informatyka lub Automatyka i Robotyka?

¹ Opcjonalnie

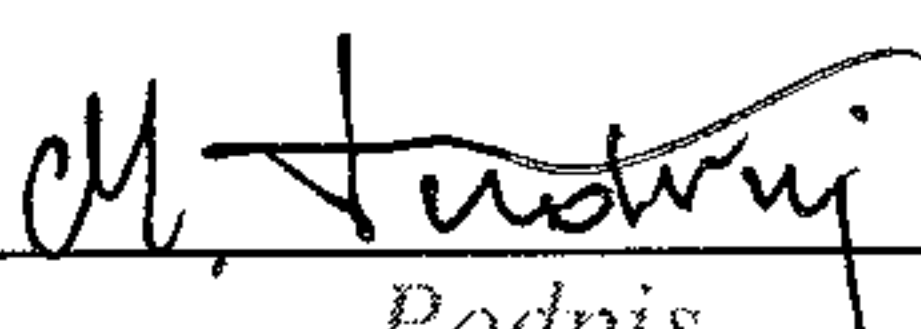
² http://www.nauka.gov.pl/g2/oryginal/2013_05/b26ba540a5785d48bee41aec63403b2c.pdf

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zdecydowa nie TAK	Raczej TAK	Trudno powiedzieć	Raczej NIE	Zdecydowa nie NIE

C. Czy kandydat umiejętność samodzielnego prowadzenia pracy naukowej?

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zdecydowa nie TAK	Raczej TAK	Trudno powiedzieć	Raczej NIE	Zdecydowa nie NIE

Ponadto, biorąc pod uwagę, że rozprawa zawiera istotne wyniki teoretyczne i praktyczne, stanowiące bardzo duży wkład w rozwój metodologii i techniki realizacji pamięci transakcyjnych a przez to istotny wkład w rozwój ogólnej techniki realizacji obliczeń rozproszonych, jak również bardzo staranne i obszerne opracowanie rozprawy oraz poparcie wyników teoretycznych przez realizację praktyczną w postaci działających środowisk systemowych, rekomenduję wyróżnienie niniejszej rozprawy doktorskiej³.


Podpis

³ Oczywiście to zdanie jest opcjonalne.