

Prof. dr hab. Lech Polkowski  
Mickiewicza 66/100, 01650 Warszawa

## RECENZJA O ROZPRAWIE DOKTORSKIEJ

zatytułowanej

*Sterowanie w programach równoległych  
oparte na spójnych stanach aplikacji*

przedstawionej przez  
mgr Janusza Borkowskiego  
Senatowi PJWSTK

1. Recenzowana rozprawa jest poświęcona problemom sterowania w programach równoległych i szczegółowej analizie w tym obszarze problemów i metod synchronizacji procesów.

Klasyczne metody synchronizacji w programach równoległych wykorzystują ideę kontrolowanego z pomocą reguł dostępu korzystania z zasobów dostępnych programowi; dla realizacji tej idei, powstało wiele koncepcji krótko omawianych przez autora rozprawy (zwanego dalej autorem) w rozdz. 1.1, jak koncepcje arbitra pamięci, test&set, semaforu, monitoru, zdalnego dostępu do pamięci RDMA, rozproszonej pamięci dzielonej (DSM), wspólnej przestrzeni danych.

Synchronizacja dotyczy również tempa wykonywania procesów (rozdz. 1.1.1.1.2), sterowania przepływem danych (rozdz. 1.1.1.1.3).

Od opisu sposobów synchronizowania w programach równoległych, przechodzi autor do poziomu analizy mechanizmów synchronizacji z uwagi na sposób przekazywania procesom informacji o stanie innych procesów (rozdz. 1.1.2.1), sposób koordynacji i powiązania procesów (rozdz. 1.1.2.2), złożoność proponowanych metod (rozdz. 1.1.2.3), sposób komunikowania się procesów z danymi (rozdz. 1.1.2.4), możliwości rozszerzenia na systemy rozproszone (rozdz. 1.1.2.5).

Analiza mechanizmów synchronizacji oparta o wspomniane konkretne idee, prowadzi autora do tego wniosku, że interesującą, a jednocześnie ważką w aspekcie zastosowań do współczesnych masywnie rozproszonych systemów ideą synchronizacji jest wprowadzenie do środowiska programu równoległego nowego obiektu którego zadaniem byłoby analizowanie globalnego stanu układu i komunikowanie procesom decyzji sterujących.

Jest to wstępny cel rozprawy i jest on realizowany w specyficznych aspektach o których poniżej w dalszym jej toku. Realizacja ta wymaga wprowadzenia formalnego aparatu dla opisu, oraz analizy stanów globalnych układu w którym biegnie program równoległy, tj. modelu obliczenia równoległego.

W rozdz. 1.2 autor podaje model systemu równoległego jako systemu *SR* złożonego z procesów sekwencyjnych  $P_1, \dots, P_N$ , komunikujących się między sobą kanałami asynchronicznymi tworzącymi silnie spójny graf połączeń, w czasie losowym, w sposób niezawodny. Zdarzenia (events) jakie mogą zaistnieć dla procesu, to *wysłanie komunikatu*

(send(message)), odebranie komunikatu (recv.(message)), oraz zdarzenia wewnętrzne, niezależne od komunikacji.

Obliczenie wykonywane przez dany proces  $P(i)$  jest to ciąg  $e(i,0), \dots, e(i, m)$ , zdarzeń; stan lokalny procesu, jest określany jako odcinek początkowy  $s(i,k) = e(i,0), \dots, e(i,k)$  obliczenia.

Stan komunikacji jest opisany w tym modelu jako zbiór komunikatów wysłanych i nie odebranych tj. dla procesów  $P(i), P(j)$  w stanach  $s(i,k), s(j,l)$ ,  $com(i,j,k,l) = \{m: send(m) \text{ jest wyrazem ciągu } s(i,k) \text{ oraz } recv.(m) \text{ nie jest wyrazem ciągu } s(j,l)\}$ .

Aspekt temporalny modelu wyrażony jest z pomocą relacji poprzedzania PS, tj., formuła  $PS(e(i), e(j))$  oznacza, że zdarzenie  $e(i)$  poprzedza zdarzenie  $e(j)$ . Zdarzenia nieporównywalne względem relacji PS są nazywane *współbieżnymi* (symbol  $\parallel$ ).

Definicja obliczenia równoległego w prezentowanym modelu jest podana na str. 19 rozprawy i jest postaci (SR, PS), tj., jest to zbiór procesów z relacją poprzedzania.

Uporządkowanie częściowe stanów procesów określa się jako antyleksykograficzne uporządkowanie stanów względem relacji PS. Podobnie jak dla zdarzeń, określa się dla stanów relacje współbieżności.

Są te relacje bardzo istotne dla formalnej strony rozprawy gdyż na ich podstawie określa się *spójne stany globalne* (Consistent Global States = CGS); są to zbiory stanów o tej własności, że jeśli dla komunikatu  $m$ , pewien stan zawiera zdarzenie send ( $m$ ) to żaden stan z tego zbioru nie zawiera zdarzenia recv( $m$ ).

Każde obliczenie  $O$  indukuje zbiór  $CGS(O)$ , *spójnych stanów globalnych* dla tego obliczenia; zbiory postaci CGS można też uporządkować częściowo z pomocą relacji przedłużenia (poprzednik zawiera stany – poprzedniki stanów w następniku).

Teraz autor może już podać definicję pojęcia *bezpośredniej osiągalności* między zbiorami stanów: relacja ta zachodzi między zbiorami  $S$  i  $T$  gdy zbiór  $T$  zawiera dokładnie jeden stan lokalny więcej dokładnie jednego procesu; domknięcie tranzytywne tej relacji to *relacja osiągalności* na zbiorach stanów.

Powyższy schemat definicyjny jest znany w literaturze i zgodny z ogólnym schematem definiowania modelu obliczeń sięgającym wstecz do prac Alana Turinga i Emila Posta; autor cytuje opracowania źródłowe.

Odrębny problem – wyznaczania CGS'ów w konkretnych realizacjach programów równoległych jest omawiany w rozdz. 1.2.3. Wspomina tu autor metodę zegarów logicznych pochodzącą od Lamporta, algorytm migawki (snapshot) podany przez Lamporta i Chandy'ego oraz metodę logicznych zegarów wektorowych.

Opis procesu równoległego w przyjętym modelu, prowadzony jest zazwyczaj z punktu widzenia spełniania pewnych warunków definiowanych jako formuły logiczne na zbiorach stanów (predykaty logiczne).

Autor podaje w rozdz. 1.3 przegląd tej problematyki, wyodrębniając aspekt modalny: możliwość spełnienia predykatu oraz konieczność jego spełnienia; jak można oczekiwać, ogólny problem spełnialności jest tu również NP.-complete. Zatem badanie heurystyk dla różnych istotnych modalności otwiera pole do szczegółowych i nowatorskich badań.

Badania przeprowadzone przez autora i przedstawione w rozprawie, miały za zadanie sprawdzenie hipotezy, że sterowanie programem równoległym oparte o analizę stanów globalnych daje dla wybranych klas problemów wyniki obliczeniowo co najmniej nie gorsze od sterowania opartego na klasycznym mechanizmie komunikacji.

Konkretne zadania jakie autor postawił sobie w rozprawie obejmowały :

- Wybór stosownej modalności,
- opracowanie algorytmów wyznaczania CGS'ów,
- opracowanie symulatora modelu obliczeniowego,
- opracowanie systemu czasu rzeczywistego dla sterowania z pomocą predykatów na stanach globalnych oraz analiza efektywności zaproponowanych metod.

Pozostałe rozdziały rozprawy są poświęcone realizacji tych zadań.

2. W rozdziale 2., autor omawia problematykę detekcji CGS'ów z pomocą zegarów czasu.

W rozdz. 2.2, omawia autor szczegółowo algorytm detekcji CGS'ów o szczególnej postaci: SCGS'ów ; zauważmy, że algorytmy dla SCGS mają złożoność wielomianową wskutek małej mocy zbioru SCGS'ów. Algorytm ten, przytoczony jako Algorithm 1 (str. 39), pochodzi od Stollera jako adaptacja algorytmu Fromentina – Raynala i ma złożoność  $O(EN \log N)$ , gdzie  $N$ =liczba procesów,  $E$ =maksymalna długość procesu.

Szczegółowość opisu Alg. 1 jest zasadna, gdyż autor następnie podaje opracowaną przez siebie modyfikację tego algorytmu (Algorithm 2, str. 42), która gwarantuje, że po znalezieniu stanu SCGS, algorytm poszukuje kolejnego takiego stanu. Poprawność algorytmu jest udowodniona w Theorem 1.

Kolejnym rozważanym problemem jest ten odnoszący się do dokładności synchronizacji zegarów; autor proponuje tu ideę lokalnej synchronizacji wybranych procesów co prowadzi do większej dokładności w wykrywaniu SCGS'ów. Algorytm ten (Algorithm 3, str. 45) wraz z procedurą Algorithm 4, str. 48, działa w czasie  $O(EN^2 \log N)$  przy złożoności pamięciowej  $O(N^2)$ .

W rozdz. 2.4, autor podejmuje problem wyznaczania SCGS'ów in statu nascendi, tj., z użyciem stanów lokalnych nie zakończonych. Proponuje w tym celu modyfikację Alg. 2 (str. 51).

Kolejnym aspektem jest zależność detekcji SCGS'ów od maksymalnego czasu transmisji komunikatów; w rozdz. 2.4.1, podaje autor algorytm (Algorithm 5, str. 55), który realizuje ten cel. Analogiczny problem detekcji przy znanym minimalnym czasie trwania stanu lokalnego jest rozwiązany w rozdz. 2.4.2 (Algorithm 6, str. 58).

Kolejną modyfikacją i ulepszeniem jakie proponuje autor jest zastąpienie jednego centralnego monitora procesów przez hierarchię monitorów o strukturze drzewa, w której monitory danego poziomu obserwują grupy procesów i konstruują dla nich stany spójne podczas gdy monitor rodzic buduje stany spójne dla obserwowanych grup z informacji uzyskanych od dzieci-monitorów.

Hierarchiczną wersję algorytmu standardowego 1, przedstawia autor jako Algorithm 7, str. 62 i dowodzi (loc.cit), że koszt obliczeniowy przy rozproszeniu monitorów jest równy kosztowi obliczenia scentralizowanego. Modyfikacje (Algorithm 8) Alg. 7 z uwzględnieniem stanów nie zakończonych podaje autor na str. 65.

3. W kolejnym rozdziale 3, autor omawia problem sterowania w programach równoległych z wykorzystaniem predykatów określonych na zbiorach stanów globalnych.

Rozpoczyna więc od analizy modalności dostępnych do implementacji z użyciem zegarów.

Wyróżnia tu znane już nam modalności *Possibly* (możliwość) oraz *Definitely*

(konieczność) wraz z modalnościami *Currently*, *Properly* i *Instantly*. Modalność *Instantly* waluuje predykaty na SCGS'ach, a więc ją właśnie jako adekwatną do wyników analizy teoretycznej w rozdz. 2 wybiera autor jako właściwą dla kontekstu rozprawy.

Autor przeprowadza analizę problemu korelacji między spełnianiem predykatu a reakcją procesów na ten fakt. Możliwe są tu różnorakie schematy korelacji i autor analizując ten problem obiera jako właściwy schemat w którym predykaty są waluowane modulo procesy oraz każdy proces otrzymuje dodatkową informację o aktualnym stanie globalnym .

Dokładna analiza problemu reakcji na predykaty jest przeprowadzona w rozdz. 3.3 i na koniec w rozdz. 3.4, autor podaje model sterowania wraz z wizualizacją graficzną.

#### 4. Efektywność proponowanych rozwiązań jest przedmiotem analizy zawartej w rozdziale 4.

Autor przedstawia tu konstrukcje symulatora z użyciem specjalistycznego pakietu OMNeT++ dla tworzenia symulacji zdarzeń dyskretnych.

Ekspertymenty symulacyjne przeprowadził autor z użyciem czterech algorytmów detekcji SCGS'ów, opisanych w rozdz. 2.: Algorithms 2, 5,6, oraz algorytm z rozdz. 2.4.2.1, wykorzystujący znany minimalny czas trwania stanu lokalnego.

Jako miar jakości sterowania, użył autor następujących parametrów:

*Opóźnienie sterowania*: czas od zaistnienia stanu globalnego S do zaistnienia reakcji wybranego procesu  $P_i$  wskutek spełnienia pewnego predykatu w S;

*Częstotliwość sterowania*: średnia liczba stanów spójnych postrzeganych przez synchronizator w jednej sekundzie = średnia liczba decyzji sterujących na jedną sekundę.;

*Opóźnienie monitorowania*: czas od zaistnienia stanu S do wykrycia tego stanu przez synchronizator;

*Aktualność stanu*: globalny czas po procesach będący sumą czasów dla procesów od zaistnienia stanu lokalnego dla danego procesu do postrzeżenia stanu globalnego przez synchronizator.

Na stronach 89-99 rozprawy, przedstawione są szczegółowe wyniki eksperymentów. Wskazują one na przydatność wprowadzonych algorytmów w określonych dla nich warunkach stosowania; sterowanie hierarchiczne okazało się korzystne, np., opóźnienie sterowania zmalało dwa razy w porównaniu z sterowaniem centralnym.

Wyniki te wskazują na istotne ulepszenia zastanych metod .

5. W rozdziale 5, autor przedstawia wyniki prac nad przekształceniem systemu P-GRADE dla projektowania aplikacji równoległych, stworzonego w SZTAKI - Instytucie Informatyki Węgierskiej Akademii Nauk w system PS-GRADE będący rozszerzeniem systemu P-GRADE z zwiększonymi możliwościami synchronizacji.

6. W rozdz. 6 , autor przedstawia wyniki eksperymentów na znanych problemach obliczeniowo trudnych : problemie TSP (komiwojażera) i problemie całkowania numerycznego. Wyniki eksperymentów wskazują również na istotne ulepszenie, np. (str. 123), dla problemu TSP, z użyciem wyspecjalizowanej sieci komunikacyjnej, uzyskano przyspieszenie aplikacji średnio 3.4 raza.

Wyniki eksperymentów, które autor przeprowadził bardzo starannie i kompleksowo, dokumentują przekonująco jego istotny wkład do problematyki obliczeń równoległych.

Omówienie treści i wyników rozprawy przeprowadzone w. 1-6 powyżej wskazuje że autor przedstawił bardzo istotny fragment problematyki obliczeń równoległych i uzyskał w tym obszarze bardzo istotne wyniki ulepszające znane dotychczas metody w wielu istotnych aspektach.

Należy więc uznać, że wyniki przedstawione w rozprawie stanowią istotnie nowy i ważki wkład do problematyki obliczeń maszynowych a fortiori więc do informatyki. Z tego punktu widzenia recenzowana rozprawa spełnia wymagania Ustawy o stopniach naukowych i tytule naukowym z dnia 14 marca 2003 roku z późniejszymi uzupełnieniami i zmianami.

Rozprawa jest przygotowana bardzo starannie; oceny tej nie umniejszają zauważalne usterki w postaci literówek oraz brakującej strony 51. Rozprawa jest napisana bardzo przejrzysto i z dbałością i czytelność, zawiera bardzo obszerną bibliografię przedmiotu. Na podstawie jej lektury można stwierdzić z pewnością, że autor bardzo dobrze opanował tę dziedzinę informatyki i porusza się w niej swobodnie i kompetentnie.

Mgr Janusz Borkowski ma w swoim dorobku 16 publikacji wymienionych w Bibliografii, własnych i z współautorami, w tym w pracach wielu konferencji międzynarodowych. Jego dorobek jest więc znaczący i pozwala na stwierdzenie, że wniósł on liczący się wkład w rozwój teorii i praktyki obliczeń równoległych.

W konkluzji powyżej stwierdzonych faktów oraz pozytywnej oceny merytorycznej zawartości przedłożonej rozprawy, stwierdzam, że rozprawa ta spełnia warunki stawiane rozprawom doktorskim przez Ustawę o stopniach naukowych i tytule naukowym w pełni, w szczególności w zakresie wymagania wniesienia istotnego wkładu w rozwój odnośnej dyscypliny naukowej w tym przypadku, informatyki i wnoszę o jej przyjęcie.

Wnoszę o dopuszczenie mgr Janusza Borkowskiego do dalszych przewidzianych w Ustawie etapów postępowania w przewodzie doktorskim .

Chciałbym też wnieść o wyróżnienie rozprawy doktorskiej Pana mgr Janusza Borkowskiego z uwagi na jej zawartość merytoryczną i dorobek naukowy jej autora, które w mojej opinii, znacznie przekraczają wymagania stawiane z reguły rozprawom doktorskim.

Warszawa, 6 stycznia 2007 roku

