Dr hab. Romuald Kotowski prof. PJATK
Polsko-Japońska Akademia Technik Komputerowych
ul. Koszykowa 86
02-008 Warszawa

RECENZJA ROZPRAWY DOKTORSKIEJ

Software Framework
for High-Performance Inter-Process Communication
Based on Shared-Memory
for Large-Scale Data Acquisition Systems

Autor rozprawy: Rolando Inglés Chávez

Promotor rozprawy: prof. dr hab. Andrzej Napieralski
Promotor pomocniczy rozprawy: dr inż. Mariusz Orlikowski

1. Cel, zakres i charakter rozprawy

Celem pracy było zaprojektowanie struktury oprogramowania obejmującej na niskim poziomie cały mechanizm związany z alokacją współużytkowanej pamięci, transferem danych i synchronizacją procesów. Zadaniem projektu była pomoc w opracowaniu spersonalizowanej aplikacji dostosowanej do wymagań użytkownika systemu ITER-CODAC[1], w którym wymagany jest ekstremalnie szybki transfer danych ze względu na charakter układu sterowania przeznaczonego do monitorowania działania agregatu prądotwórczego TOKAMAK wykorzystującego reakcje fuzji termojądrowej.
Praca ma charakter praktyczny, gdyż proponuje rozwiązania programistyczne mogące być podstawą do tworzenia systemów komputerowych również w innych obszrach niż wspomniany powyżej.

2. Zawartość rozprawy

Rozprawa doktorska Rolando Inglés Cháveza liczy 118 numerowanych stron i składa się z 6 rozdziałów oraz streszczenia, spisu tabel (10 pozycji), spisu rysunków (60 pozycji), spisu skrótów (40 pozycji), podziękowań oraz spisu literatury (89 pozycji). Praca jest napisana w języku angielskim.

[1] ITER – International Thermonuclear Experimental Reactor (Międzynarodowy Eksperymentalny Reaktor Termonuklearny); CODAC – Control, Data Access and Communication System (System Sterowania, Dostępu do Danych i Komunikacja).

W Rozdziale 1 (Introduction) przedstawiono główny cel rozprawy, czyli zaprojektowanie nowej struktury informatycznej służącej komunikacji między procesami i ich synchronizacji umieszczonych we współdzielonej pamięci (*shared-memory*). Dane są pozyskiwane z wielu źródeł i muszą być przetwarzane z największą możliwą do uzyskania prędkością. O wadze problemu świadczy fakt, że zaproponowana technologia jest stosowana w sterowaniu przebiegu kontrolowanej fuzji termojądrowej.

Obecnie od wielu lat prowadzone są intensywne prace badawcze dotyczące zastosowania urządzenia umożliwiającego wykorzystywanie kontrolowanych reakcji termojądrowych na skalę przemysłową. Głównym problemem jest osiągniecie zysku energetycznego, czyli uzyskanie z urządzenia więcej mocy od mocy dostarczonej i koniecznej do zajścia reakcji termojądrowej. Obecny rekord (jeszcze z roku 1997) został uzyskany w JET TOKAMAK (Joint European Torus TOKAMAK[2]) gdzie wkład mocy dostarczonej urządzeniu wynosił 24MW, a uzyskano zaledwie 16MW, czyli wydajność Q wyniosła zaledwie 67%.
Najbardziej ambitnym projektem na skalę światową jest obecnie projekt ITER, w którym zakłada się, że pierwsza plazma (czyli gaz zjonizowanych atomów) zostanie wygenerowana nie wcześniej niż w roku 2019.
W dalszym ciągu rozdziału omówione zostały podstawy reakcji termojądrowej, czyli fuzja lekkich atomów (lub ich izotopów) doprowadzonych do bardzo wysokiej temperatury (100 000 000 °C) oraz zalety takiej procedury.
Kolejny podrozdział (1.1.2) poświęcony został opisaniu urządzeniu ITER-TOKAMAK. Wyliczono jego najważniejsze składowe, ich parametry oraz architekturę. Tworzą one razem strukturę ITER Instrumentation and Control (I & C), składająca się z ponad 160 podukładów.

CODAC grupuje różne układy, a każdy z nich dysponuje własnym system pozyskiwania danych z własnymi czujnikami i aktuatorami (urządzeniami uruchamiającymi) oraz oprogramowaniem sterującym i monitorującym przebieg procesu.

Technologia użyta do przesyłu dużych ilości danych i wykorzystująca współdzieloną pamięć to POSIX[3]. Została ona wykorzystana w aplikacjach pisanych w obiektowo zorientowanym

[2] TOKAMAK – (ros.:*Toroidalnaja Kamiera s Magnitnymi Katuszkami*) – toroidalna komora z cewkami magnetycznymi.

[3] **POSIX** (ang. *Portable Operating System Interface for Linux*) – przenoszalny interfejs dla systemu operacyjnego Linux.

języku programowania C++. POSIX zapewnia obsługę kolizji zapisu danych, co dla tego typu zadania jest bardzo istotnym elementem projektu.

Tezy pracy, jakich Autor dysertacji postanowił dowieść, zostały sformułowane w następujący sposób[4]:

1. Zastosowanie obiektów synchronizujących umieszczonych w pamięci współdzielonej i użytej do komunikacji między procesami w dużo-skalowym systemie pozyskiwania danych daje porównywalne wyniki jakie mogą być otrzymane w wielowątkowym modelu programowania.
2. Możliwe jest zaprojektowanie struktury programistycznej wykorzystującej pamięć współdzieloną do tworzenia niezależnych wielo-procesowych aplikacji komputerowych w schemacie operacyjnym producent/konsument i ich zastosowanie w systemach pozyskiwania i przesyłania wielkich ilości danych z dużą wydajnością.

Rozdział 2 (Background and Related Work) jest najbardziej obszernym rozdziałem pracy i obejmuje 34 strony (strony do 27 do 61). Poświęcony został przypomnieniu pojęć używanych przez Autora w dalszej części pracy. W szczególności podał on definicję systemu pozyskiwania danych DAQ (ang.: *Data Aquisition System*) jako połączenie urządzeń odbierających sygnały, np. w postaci obrazów z kamery, urządzeń przetwarzających je w sygnały elektryczne, odpowiedniego hardware'u przetwarzającego sygnały elektryczne w sygnały cyfrowe, a następnie odpowiednie oprogramowanie komputerowe przekazujące te dane do wyspecjalizowanych aplikacji komputerowych w celu odpowiedniego sterowania systemem.

---

[4] Wersja oryginalna:
Thesis 1
Using synchronization objects placed in shared memory, which can be used for inter-process communication in large-scale data acquisition systems, produces comparable results to those that can be obtained using multi-threading programming model.
Thesis 2
It is possible to design a software-programming framework based on shared-memory for the development of independent multi-process applications with the producer/consumer operation schema to be used in large-scale data acquisition systems with high-rate data streams throughput.

I tak, np. w celu obserwacji plazmy wewnątrz TOKAMAK-u w projekcie ITER używany był moduł TAMC641.
W podrozdziale 2.3.2 omówione zostało pokrótce programowanie wielowątkowe, co w przypadku prezentowanego projektu ogrywa kluczową rolę. Omówiono również różnice pomiędzy wątkami a procesami.
Podrozdział 2.4 poświęcony jest omówieniu komunikacji pomiędzy procesami (ang. *Inter-Process Communication* IPC). Kluczowym elementem jest tu współdzielona pamięć, co pozwala na znaczne przyspieszenie prowadzonych obliczeń. Z drugiej strony pojawiła się potrzeba unikania dostępu do wspólnych obszarów pamięci przez współbieżne procesy, co wymusiło wykorzystywanie zmiennych warunkowych, mutexów[5] i semaforów. W Tabeli 2.1 przedstawiono zalety współpracy procesów.

W podrozdziale 2.5 omówione zostały problemy związane ze współbieżnością obliczeń, a w podrozdziale 2.6 ze współdzieloną pamięcią. Do zarządzania współdzieloną pamięcią został użyty system komputerowy NUMA (ang. *Non-uniform Memory Access*). Podrozdział 2.7 poświęcony został omówieniu struktur informatycznych FastFlow i ACE (ang. *Adaptive Communication Environment*) wspierających programowanie równoległe. Z tych dwu Autor wyżej ceni ACE, ale jego główną wadą jest fakt, że wykorzystuje on system operacyjny Microsoft Windows, a nie Linux. Rozdział 2 zamyka omówienie różnych podejść do dynamicznego przydziału pamięci współdzielonej (DSMA): podejście polegające na stronicowaniu wykorzystującemu pamięć wirtualną (R. Benosman), wykorzysującemu standardowe biblioteki C++ (M. Ronell), wykorzystującemu w pamięci dzielonej odwzorowania, wektory i listy (G. Ketema) oraz CxxDisruptor, będący implementacją LMAX Disruptor, czyli wysoce wydajnego systemu wymiany komunikatów pomiędzy wątkami (H. Baastrup).

Rozdział 3 (Framework Review) podsumowuje wymagania, jakie są stawiane strukturze informatycznej przez architekturę ITER-CODAC.
Struktura informatyczna, jaka ma być zaproponowana w dysertacji, ma dwie składowe: przydzielanie pamięci i synchronizację współdzielonych obiektów. Został zaprezentowany schemat blokowy systemu CODAC (omówionego w Rozdziale 1) oraz szkielet struktury informatycznej, w którym oprogramowanie aplikacyjne działa w trybie producenta danych

[5] Mutex (ang. *mutual exclusion*).

zbieranych z urządzeń monitorujących i przekazujących je w procesie konsumenta we współdzielonym buforze danych. Odpowiedni rysunek zilustrował również schemat logiczny działania systemu z elementami zapewniającymi funkcjonalność poszczególnych jego składowych.

W Rozdziale 4 (Framework Design) zaprezentowano, jakie działania muszą być podjęte przez odpowiednią strukturę informatyczną w celu podziału danych pomiędzy nieskorelowane procesy. Wszystkie działania mają miejsce w schemacie *producent – konsument*. Producent jest odpowiedzialny za publikacje danych, a konsument oczekuje na sygnał, że nowe dane nadeszły, a po otrzymaniu takiego sygnału przetwarza te dane i informuje o tym fakcie producenta.

W poszczególnych podrozdziałach zostały omówione szablony (templates) klas Producenta (Producer Class) , Konsumenta (Consumer Class), Procesu (Process Class) i Kanału (Channel Class) będącego abstrakcyjną implementacją sposobu komunikacji pomiędzy procesami używającymi współdzielonej pamięci oraz klasy Współdzielonych Zasobów (Shared Resource Class).

W kolejnych podrozdziałach omówione zostały bardziej szczegółowo implementacje komputerowe poszczególnych zadań: sterowanie dostępem do danych w buforze pierścieniowym i w trybie wsadowym. W obu tych procedurach konsument musi spędzić pewien czas oczekując na nowe zadanie.

Warto zwrócić uwagę, że Autor dysertacji zaproponował zmodyfikowane podejście, w którym wykorzystywane są dwa bufory pierścieniowe: jeden dla danych, a drugi dla sygnalizacji. To oczywiście zwiększa ilość wykorzystywanej pamięci komputera, ale też skutecznie redukuje czas pracy CPU.

Rozdział 5 (Operational Testing) poświęcony został omówieniu wyników przeprowadzonych symulacji komputerowych opracowanego systemu informatycznego. Testowano kolejno efektywność algorytmów zastosowanych w omawianych poprzednio prototypach sposobu przesyłania danych. Jako punkt odniesienia w prędkości przesyłania danych wybrano wielkość opóźnienia przesyłu rzędu 147 μs.

W podejściu *pojedyncza wiadomość* (Per-Message Approach) wykorzystano takie obiekty synchronizujące jak zmienne warunkowe, futexy[6] i zmienne atomowe, w podejściu *bufora pierścieniowego* (Ring-Buffer Approach) użyto tych samych obiektów synchronizujących, ale tu stwierdzono że tylko 60%wiadomości było przesyłanych poniżej granicy 147 μs co nie jest akceptowalne. Nie spełniło oczekiwań również podejście *wsadowego bufora pierścieniowego* (Batch Ring-Buffer Approach) gdzie tylko 512 segmentów (*slots*) miało czas oczekiwania poniżej 147 μs, co było sprzeczne z oczekiwaniami. W *podejściu pojedynczego segmentu* (Per-Slot Approach) wykorzystano pięć mechanizmów komunikacji pomiędzy procesami (IPC): blokada czytaj-zapisz (*read-write lock*), semafor, mutex, wirująca blokada (*spin-lock*) i atomowa blokada czytaj-zapisz (*atomic read-write-lock*).

3. Opinia merytoryczna

Autor rozprawy zaproponował trzy prototypy struktur informatycznych mających za zadanie przesyłanie i analizę olbrzymich ilości danych z jak największą wydajnością, czyli w jak najkrótszym czasie i z ograniczoną wielkością użytej pamięci komputera, a następnie dokonał ich krytycznej analizy na postawie przeprowadzanych testów.

Pierwszy prototyp został zrealizowany po przyjęciu założenia, że największą prędkość w komunikacji pomiędzy procesami osiągnie się tylko wtedy, gdy tylko jedna komórka będzie współdzielona przez producenta i konsumenta i gdy tylko jedna dana będzie przesyłana. Procedura ta jest uzupełniona przez dwa obiekty synchronizujące: jeden że nowa dana została już opublikowana (wiadomość dla konsumenta) i drugi, że nowa dana została odczytana (wiadomość dla producenta).
Drugi prototyp wykorzystywał pierścieniową strukturę bufora danych, polegającą na wydzieleniu ustalonego obszaru współdzielonej pamięci i podzieleniu ich na pojedyncze komórki. Każda komórka lub odpowiedni slab pamięci są logicznie połączone z każdym segmentem bufora pierścieniowego i są przeszukiwane sekwencyjnie.

Końcowy prototyp, będący podsumowaniem przeprowadzonych badań, wykorzystuje prototyp drugi, ale jest stosowany do każdego segmentu danych. Udoskonalenie prototypu polega na tym, że zamiast dwu globalnych obiektów synchronizujących pomiędzy producentem i konsumentem każdy segment danych ma swój odpowiednik obsługujący

[6] futex – ang. *fast user space mutex.*

sygnalizację. To podejście eliminuje czas potrzebny na sprawdzanie czy nowe dane są dostępne w określonym segmencie kiedy konsument nie oczekuje tych danych. Podejście to wymaga wprowadzenia dwu współdzielonych buforów pierścieniowych: jeden dla danych, a drugi dla sygnalizacji. To podejście zwiększa prędkość działania systemu.

Autor dysertacji przeprowadził systematyczne badania dostępnych narzędzi programistycznych w programowaniu równoległym w celu zintegrowania ich w jeden system najlepiej realizujący postawiony sobie cel, czyli efektywnego przetwarzania olbrzymich ilości danych. W przeprowadzonych symulacjach komputerowych ujawnił silne i słabe strony proponowanych rozwiązań, czyli przedstawionych powyżej prototypów, co pozwoliło mu wybrać optymalne rozwiązanie.

4. Poprawność i oryginalność postawionych tez

Obserwowany obecnie gwałtowny rozwój technologii komputerowych stawia przed praktykami coraz więcej wyzwań. Jednym za takich wyzwań jest przetwarzanie i integracja napływających danych z różnych źródeł w celu umożliwienia podjęcia odpowiednich kroków aby założone zadania mogły być zrealizowane.
Autor podjął się realizacji tego ambitnego zadania i zbudował narzędzie, jakie będzie mogło znaleźć zastosowania nie tylko w projekcie ITER-CODAC, będącym inspiracją do rozpoczęcia prac nad tematem będącym przedmiotem dysertacji.
Zadanie zostało sformułowane poprawnie, a tezy dysertacji, czyli zastosowanie obiektów synchronizujących w pamięci współdzielonej i użycie ich do sprawnej komunikacji między procesami, a następnie przyjęcie założenia, że możliwe jest zaprojektowanie struktury programistycznej wykorzystującej pamięć współdzieloną do tworzenia niezależnych wielo-procesowych aplikacji komputerowych pracującej z bardzo dużą wydajnością są oryginalnymi pomysłami Autora dysertacji.

5. Czy tezy rozprawy zostały w rozprawie wykazane

Przeprowadzone przez Autora dysertacji symulacje komputerowe działania zaproponowanych rozwiązań programistycznych i ich dyskusja wykazały, że przyjęte przez niego założenia były trafne i tezy rozprawy zostały wykazane.

6. Analiza źródeł i wiedza autora w danej dyscyplinie naukowej

Autor dysertacji wykazał się głęboką wiedzą dotyczącą skomplikowanych procesów w obliczeniach równoległych i w efektywnym przetwarzaniu danych pochodzących z różnych źródeł i ich synchronizacją w procesie producent-konsument.

Źródła zostały dobrane poprawnie i są adekwatne do treści prezentowanych w rozprawie.

Zauważam z żalem, że brakuje w spisie literatury pozycji odnoszących się do wcześniejszych prac Autora dysertacji.

7. Pozycja rozprawy na tle stanu wiedzy w literaturze

Praca wniesie istotny wkład w zrozumienie złożonych problemów w obliczeniach równoległych i będzie inspiracją do dalszych poszukiwań coraz bardziej skutecznych metod przetwarzania informacji.

8. Znaczenie wyników dla dyscypliny naukowej

W pracy *Virtual Tokamak Library Software*, autorzy D. P. Kostomarov, F. S. Zaitsev, A. G. Shishkin, D. Yu. Sychugov, S. V. Stepanov, i E. P. Suchkov piszą (Moscow University Computational Mathematics and Cybernetics, 2011, Vol. 35, No. 4, pp. 201–206):

"Obecnie naukowcy rosyjscy zgromadzili ogromną liczbę matematycznych metod, oprogramowania, różnych podejść informatycznych dotyczących ich programów badawczych. Rozwinęli oni i pomyślnie zastosowali numeryczne programy symulujące ważne procesy w plazmie."

Od czasu publikacji tej pracy minęło już kilka lat i mogłoby się zatem wydawać, że niewiele już zostało w tej dziedzinie do zrobienia. Okazuje się jednak, że projekt ITER-CODAC, który zgromadził naukowców z niemal całego świata i został zaplanowany na jeszcze wiele lat, ciągle potrzebuje nowych rozwiązań, nie tylko w dziedzinie fizyki plazmy, ale również w obszarze informatyki.

Zastosowanie idei zaproponowanych przez Autora dysertacji może być istotnym wkładem w realizację tego projektu, tak ważnego dla przyszłości ludzkości.

9. Umiejętność autora poprawnego przedstawienia wyników

Autor przedstawia wyniki swej pracy w postaci licznych wykresów, co jest oczywiście bardzo dobrym rozwiązaniem. Część pracy omawiająca symulacje komputerowe zaproponowanych rozwiązań programistycznych i dyskutująca otrzymane wyniki zajmuje w rozprawie zaledwie

15 stron, co stanowi około 12% objętości całej dysertacji. Praca zyskałaby istotnie, gdyby ta część pracy była bardziej obszerna.

10. Słabe strony rozprawy i uwagi szczegółowe

Praca mogłaby być zredagowana bardziej staranie. I tak:

1. Po stronie tytułowej praca rozpoczyna się Streszczeniem (Abstract), a strona nosi numer ii (numeracja rzymska). Ostatni numer numeracji rzymskiej to xvi.
2. Rozdział 1 Wstęp (Chapter 1 Introduction) znajduje się na stronie 17 (numeracja arabska). Tradycją jest, aby oba te sposoby numeracji stron były traktowane oddzielnie i rozpoczynały się zawsze od liczby 1 (jeden).
3. Akronim ITER (str. xiii) jest rozwinięty jako The ITER project, czyli niczego nie wyjaśnia, a jest to pojęcie istotne dla całej pracy. Stoi to w kontraście do akronimu SIGKILL (str. xiv) i do nazwy systemu operacyjnego UNIX.
4. Str. 38 Na tej stronie jest tylko podpis Tabeli 2.2, natomiast sama tabela jest na stronie następnej.
5. Str. 54 Pół strony jest puste. Rysunek 2.15 zmieściłby się na tej stronie, gdyby go nieco zmniejszyć.
6. Str. 59 Rysunek jest umieszczony na dole strony, a jego podpis znajduje się w pierwszym wierszu str. 60.
7. Str. 64 W podrozdziale 3.5 Logical Design system informuje o braku odniesienia do rysunku: Error! Reference source not found.
8. Str. 80 niemal cała pusta. Można by na niej umieścić treść ze str. 82.
9. Str. 109 Spis literatury został sformatowany jako wyjustowany. Powoduje to nieeleganckie efekty, np. pozycje [9], [60], [69] i [79].

11. Wniosek końcowy

Niniejszym stwierdzam, że przedstawiona do recenzji rozprawa doktorska Pana mgr. inż. Rolando Inglés Cháveza *Software Framework for High-Performance Inter-Process Communication based on Shared-Memory for Large-Scale Data Acquisition Systems* w dziedzinie informatyka spełnia wymogi stawiane rozprawom doktorskim w Ustawie z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz stopniach i tytule w zakresie sztuki, Dz.U. 2003 nr 65 poz. 595 (tekst jednolity z poprawkami Dz.U. z dnia 15 września 2017 r. poz. 1789).

Na tej podstawie wnioskuję o dopuszczenie Pana mgr. inż. Rolando Inglés Cháveza do publicznej obrony rozprawy doktorskiej przed Radą Wydziału Elektrotechniki, Elektroniki, Informatyki i Automatyki Politechniki Łódzkiej.

Warszawa, 16 lipca 2018 r.