Janusz Borkowski
asystent w PJWSTK
janb@pjwstk.edu.pl

Warszawa, 2004-01-14

Sterowanie w programach równoległych oparte na spójnych stanach aplikacji

Omówienie pracy doktorskiej

Praca dotyczy metodologii programowania równoległego, a ściślej zagadnień sterowania i synchronizacji w programach równoległych. Dziedzina ta jest istotna z punktu widzenia zarówno praktyki jak i teorii. Istniejące metody programowania równoległego dają do dyspozycji programiście jedynie podstawowe prymitywy sterujące wykonaniem programu. Prymitywy te są identyczne z prymitywami używanymi w programowaniu sekwencyjnym (if, while, case), opartymi o badanie stanu lokalnego procesu lub są prostymi prymitywami synchronizującymi typu bariera. Bardziej rozwinięte sterowanie powinno brać pod uwagę całą równoległą aplikację w oparciu o uogolnione funkcje sterujące. Obecnie programista musi sam uzupełnić niedostatki modelu i zatroszczyć się o to, aby pojedynczy proces posiadał dostateczne informacje o całości programu. W istniejących systemach rozwiązaniem jest przekazywanie danych między procesami. Stosowane jest ono ad-hoc, a kod odpowiedzialny za sterowanie i synchronizacje jest zmieszany z kodem specyfikującym obliczenia. W tej sytuacji wskazane wydaje się rozszerzenie modelu sterowania w programach równoległych. Skoro w programach sekwencyjnych prymitywy sterujące badają stan lokalny procesu, to w naturalny sposób prymitywy sterujące w programach równoległych mogą badać stany globalne aplikacji. Rozszerzony model powinien zawierać mechanizmy sterujące biorące pod uwagę i obejmujące działaniem całość programu równoległego. Programista mógłby wtedy oprzeć logikę działania w swych programach o dostarczone gotowe prymitywy sterujące i skoncentrować się na algorytmie aplikacji, a nie na realizacji potrzebnych metod sterowania. Pojęcie stanów globalnych i w szczególności stanów spójnych w systemach rozproszonych doczekały się pokaźnej liczby opracowań. Dotychczasowa teoria i praktyka koncentrowała się jednak na zastosowaniu tych pojęć do biernego monitorowania programów równoległych, często w połączeniu z problematyką rozproszonych programów uruchamiających (debuggerów). Na stanach spójnych określa się warunki logiczne (predykaty globalne) i za ich pomocą bada się poprawność działania aplikacji. Wiodącym pomysłem niniejszej pracy jest zastosowanie predykatów globalnych do aktywnego sterowania na bieżąco wykonaniem programu.

Proponowana idea jest dobrze umotywowana. Programista zyskuje silny, elastyczny i dobrze określony model sterowania w programach równoległych. Dotychczasowe pomieszanie funkcji synchronizujących/sterujących z czystym przesyłaniem danych, powszechne w standardowych modelach programowania równoległego, nie ma tu miejsca. Warunki sterujące są wyodrębnione w postaci predykatów, zapisane w wydzielonych określonych miejscach w programie, a zatem łatwe do zrozumienia, korekty i modyfikacji. Proponowany mechanizm sterujący może być użyty bezpośrednio także do weryfikacji programów, analogicznie jak w przypadku rozproszonych debuggerów, też korzystających z predykatów globalnych. Co więcej, sterowania oparte wprost na predykatach globalnych precyzujących warunki poprawności aplikacji może uczynić programy od razu poprawnymi przez konstrukcję.

Rozdział funkcjonalny sterowania programem od przesyłania danych umożliwia wykorzystanie zaawansowanych architektur systemów równoległych dysponujących niezależnymi sieciami do przesyłania informacji sterującej oraz do przesyłania danych. Każda z tych sieci powinna mieć inne charakterystyki, najlepiej dostosowane do pełnionych funkcji. Dzięki temu wydajność systemu komputerowego może być lepsza niż dla systemów dysponujących jedną uniwersalną siecią o kompromisowych charakterystykach.

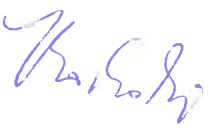
Sterowanie wykonaniem programu na bieżąco wymaga metody szybkiej konstrukcji stanów spójnych dającej mały narzut czasowy. Dodatkowo podejmowane decyzje sterujące muszą bazować na informacji pewnej, a nie tylko prawdopodobnej, dostępnej możliwie natychmiast i dostatecznie pełnej. W niniejszej pracy zanalizuję te kwestie. Zostaną opracowane specjalizowane efektywne algorytmy detekcji stanów spójnych. Algorytmy te trzeba będzie poddać ewaluacji pod kątem wprowadzanego obciążenia oraz możliwej do uzyskania szybkości detekcji stanów spójnych i co za tym idzie szybkości reakcji systemu sterowania. Dodatkowe procesy nadzorujące wykonanie programu (synchronizatory) będą otrzymywały od procesów aplikacyjnych syntetyczną informacje o ich stanie i za pomocą opracowanego wydajnego algorytmu będą obserwowały stany aplikacji. Decyzje sterujące wykonaniem programu będą oparte o predykaty globalne zdefiniowane zgodnie z wymaganiami aplikacji przez programistę w synchronizatorze. Decyzje te będą propagowane do procesów aplikacyjnych w celu ich realizacji. Istnieje kilka możliwości ich realizacji, będą one omówione.

Najbardziej obiecującym sposobem reagowania jest asynchroniczna aktywacje kodu. Sygnał od synchronizatora niosący informację o spełnieniu określonego predykatu globalnego prowadzi do możliwie natychmiastowego uruchomienia wskazanej procedury. Taki sposób reakcji pozwala unikać biernego oczekiwania na decyzję sterującą, co jest powszechnym problemem w klasycznych modelach sterowania (np. oczekiwanie na komunikat niosący wartość od której zależeć ma decyzja). Promowane jest pełne wykorzystanie procesora. Drugim interesującym sposobem reagowania jest zaprzestanie obliczeń. Sygnał od kontrolera niosący informację o spełnieniu określonego predykatu powoduje porzucenie bieżącej fazy obliczeń. Obliczenia te z globalnego punktu widzenia są już niepotrzebne, a zatem ich kontynuowanie byłoby marnowaniem czasu procesora.

Proponowany model sterowania zostanie przebadany symulacyjnie. Przebadany zostanie wpływ parametrów sprzętu i systemu na efektywność funkcjonowania modelu. Zostaną też przetestowane poszczególne warianty algorytmów detekcji stanów spójnych i architektury systemu komputerowego. W szczególności sprawdzona będzie opracowana metoda hierarchicznej detekcji stanów spójnych i hierarchicznego podejmowania decyzji sterujących za pomocą hierarchii synchronizatorów. Wreszcie będzie też można stwierdzić jakie aplikacje (ich cechy, parametry) są szczególnie predestynowane do efektywnej realizacji w proponowanym modelu sterowania.

Prezentowane idee znajdą odzwierciedlenie w systemie graficznego programowania równoległego GRADE. GRADE to system praktycznie działający, pozwalający na tworzenie aplikacji równoległych metodą przekazywania komunikatów, jednakze bez proponowanych w pracy mechanizmy. Testowe aplikacje opracowane za pomocą wariantu GRADE'a rozszerzonego o proponowane mechanizmy sterujące będą porównane z analogicznymi aplikacjami stworzonymi w zwykłym środowisku GRADE.

Głównym rezultatem pracy będzie zaawansowany model sterowania w programach równoległych oraz metoda realizacji tego sterowania. Przeprowadzone badania pozwolą na ustalenie warunków potrzebnych aby proponowany model działał efektywnie. Uzyskane doświadczenie pozwoli stwierdzić które elementy w opisywanym modelu i w jego realizacji wymagają zmian lub uzupełnień. Przewidywanym dalszym kierunkiem prac będzie rozwijanie metodologii tworzenia algorytmów równoległych efektywnie wykorzystujących proponowane mechanizmy sterujące.



Janusz Borkowski
asystent w PJWSTK
janb@pjwstk.edu.pl

Warszawa, 2004-01-14

Sterowanie w programach równoległych oparte na spójnych stanach aplikacji

Plan pracy doktorskiej