

Michał Podpora

Wydział Elektrotechniki, Automatyki i Informatyki

Instytut Automatyki i Informatyki

Politechnika Opolska

O PRZYSPIESZANIU KLASTRA STERUJĄCEGO ROBOTEM MOBILNYM

Abstract: In this paper author describes some ideas of speeding up a computer cluster used in a specific machine vision approach. He explains some ideas of minimization of communication time and other methods of speeding up the algorithm.

Robot mobilny powinien być urządzeniem lekkim, posiadać niewielkie rozmiary, a ponadto dysponować „umiejętnością” orientacji w terenie. Realizacja praktyczna napotyka jednak na trudności. Zaimplementowanie algorytmów sztucznej inteligencji w oparciu o wejście z systemu wizyjnego wymaga sporej mocy obliczeniowej, co skutkuje tym, że komputer sterujący robotem (będąc „potężną maszyną”) może pobierać zbyt dużo prądu lub zajmować zbyt wiele miejsca. Warto zastanowić się wówczas nad zastosowaniem w zamian np. beznapędowego laptopa, przerzucając ciężar obliczeń na zdalny komputer.

W niektórych implementacjach nawet po zastosowaniu nowoczesnej maszyny wieloprocesorowej jako odpowiedzialnej za „inteligencję” robota, moc obliczeniowa jaką dysponujemy jest zbyt mała. W przypadku klastra komputerowego moc nie jest problemem. Skomplikowanym zadaniem jest natomiast umiejętne napisanie aplikacji wykorzystującej ową moc, optymalny podział zadań między węzły klastra (dobór tzw. granulacji zadań), poprawny algorytm komunikacji stworzony w oparciu o wybraną topologię, itd. Specyfika programowania jest też zupełnie inna niż w przypadku „zwykłego” programu.

Stwierdzenie, że klastr zbudowany z czterech komputerów zwróci wynik cztery razy szybciej niż pojedynczy komputer, jest dalekie od rzeczywistości. Wynika to m.in. z opóźnień wprowadzanych przez algorytm synchronizujący realizację zadań i koordynujący pracę klastra, ale przede wszystkim z tzw. czasu komunikacji - dane naszej aplikacji nie będą „poruszały się” jedynie w obrębie stosunkowo szybkiej magistrali wewnętrznej komputera, ale będą przekazywane pomiędzy maszynami poprzez interfejs sieciowy. Dla porównania: moduły pamięci DDR3 PC-12700 mają przepustowość 12,79 Gb/s, podczas gdy osprzęt sieciowy zgodny ze standardem 1000 Base-T dysponuje przepływnością równą 1 Gb/s. Może się więc zdążyć, że program „na klastrze”

działa wolniej niż na pojedynczym komputerze. Trudności jest wiele, ale i efekt obiecujący.

Klaster komputerowy, użyty jako jednostka zdalnego zarządzania sterowaniem robota mobilnego, posiada pewne specyficzne cechy, które muszą koniecznie być wzięte pod uwagę już w fazie projektowania aplikacji. Przykładem jest fakt, że nad wejściem (kamera i czujniki) i wyjściem (silniczki sterujące ruchem) kontrolę sprawuje jeden określony komputer (wspomniany we wstępie laptop) – co powoduje ułatwienie wyboru [2] topologii wirtualnej klastra (czyli logicznej organizacji komunikacji pomiędzy węzłami klastra) spośród sześciu podstawowych topologii [1] zostają do dyspozycji [2] pierścień, gwiazda i drzewo. Nie bez znaczenia jest również to, że w przypadku awarii tego właśnie komputera robot byłby unieruchomiony. Niejako korzystając z tego faktu, programista może dodatkowo użyć tego komputera jako statycznie określonego węzła koordynującego całą komunikację w klastrze. Zagadnienie wykrywania problemów komunikacji z koordynatorem i elekcji nowego koordynatora po prostu nie będzie tu miało zastosowania.

W przypadku robota mobilnego wyposażonego w kamerę, specyfika danych przetwarzanych przez system (gro operacji to analiza i przekształcenia poszczególnych klatek sekwencji wideo) powoduje, że kluczowym zagadnieniem staje się właśnie minimalizacja czasu komunikacji. Jednym z rozwiązań jest redukcja objętości przesyłanych danych – w większości przypadków nie jest wymagane przesyłanie „kompletnej” bitmapy w najwyższej palecie barw, może wystarczyć wycinek obrazu, w odpowiednio dobranym formacie monochromatycznym. Tak samo jak człowiek nie obserwuje równocześnie wszystkich szczegółów pomieszczenia w którym się znajduje, robot mógłby „skupić uwagę” na fragmencie obrazu z kamery [3].

Metod na przyspieszenie jest wiele – w zależności od topologii klastra, działania aplikacji i pomysłowości programisty. Przewaga skuteczności jednych metod nad innymi to odrębne zagadnienie, wymagające wykonania rzetelnych pomiarów.

LITERATURA

- [1] Karbowski A., Niewiadomska-Szynkiewicz E., Obliczenia Równoległe i Rozproszone, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2001,
- [2] Podpora M., Decentralizacja sterowania robota mobilnego przy wykorzystaniu klastra komputerowego, I SSNE'07 – materiały konferencyjne, przed publikacją
- [3] Podpora M., Computer vision in parallel computing, ISTET'07 – materiały konferencyjne, przyjęte do publikacji