Programowanie równoległe i rozproszone

Laboratoria nr 3

Jakub Niewadzi

Kacper Sosnowski

Rozproszone sieciowo procesy

1. Opis działania programu

Program składa się z trzech plików:

- my-server.py – Odpowiada za stworzenie kolejek wejściowej i wyjściowej. Serwer przyjmuje jako argumenty wejściowe adres ip oraz port, na którym serwer jest uruchomiony.

- client.py – Odpowiada za rozdzielenie zadań dla workerów, dzieli macierz wejściową na pojedyncze wiersze i wrzuca je na kolejkę wejściową. Klient pobiera również wyniki częściowe z kolejki wyjściowej, a po pobraniu wszystkich wyników wypisuje macierz wynikową. Klient przyjmuje jako argumenty wejściowe adres ip serwera, port, na którym serwer jest uruchomiony oraz tryb pracy.

- worker.py – Odpowiada za pobranie elementu macierzy z kolejki wejściowej, obliczenie iloczynu i zwrócenie wyniku na kolejkę wyjściową. Workerzy posiadają w sobie zmienną, która oznacza liczbę procesów, jest ona ustalana na podstawie wartości z biblioteki multiprocessing, która pobiera ilość rdzeni dla danego procesora. Worker przyjmuje takie same argumenty wejściowe jak klient.

Jednym z argumentów wejściowych jest tryb pracy. Mówi on po ile elementów jest przekazywanych do kolejki.

Tryb 1:

Do kolejki przykazywany jest tylko po jednym obiekcie klasy QueueObject, który odpowiada pojedynczej wartości wektora wyjściowego.

Tryb 2:

Do kolejki przekazywane jest lista pięćdziesięciu obiektów klasy QueueObject. Zmniejsza to liczbę interakcji z kolejką, która jest bardzo czasochłonną operacją.

Na zajęciach nie udało się wykonać wszystkich pomiarów, z tego powodu zdecydowaliśmy się powtórzyć wszystkie pomiary zdalnie, aby ich porównanie było takie samo dla obu metod rozdziału danych w workerach. Taka konfiguracja została stworzona dzięki połączeniu się do sieci typu GAN ZeroTier. Czasy odpowiedzi w sieci GAN są naturalnie dłuższe niż takie jak na przykład w sieci LAN, w której znajdowały się komputery w sali zajęciowej.

Obliczenia zostały przeprowadzone w dwóch konfiguracjach:

a) Klient, serwer oraz dwóch workerów na jednej maszynie.

W tym wypadku wszystko było przeprowadzane w obrębie sieci pętli zwrotnej, w której czasy przekazania danych są bardzo krótkie, co wpłynęło pozytywnie na czas wykonania.

b) Klient oraz serwer na jednej maszynie, po jednym workerze na maszynach zewnętrznych

2. Pomiary dla trybu 1:

Wyniki pomiarów:

Konfiguracja	Czas wykonania [s]
Klient, serwer i dwóch workerów na jednej maszynie	2,51359129
	2,368112087
	3,122713327
	2,170364857
	2,275924206
Klient i serwer na jednej maszynie, po jednym workerze na oddzielnych maszynach	31,26
	32,97836018
	34,99115014
	31,11525393
	30,6838212

3. Pomiary dla trybu 2:

W tym trybie działania programu przeprowadziliśmy pomiary czasu dla dwóch konfiguracji z poprzedniego punktu.

Wyniki pomiarów:

Konfiguracja	Czas wykonania [s]
Klient, serwer i dwóch workerów na jednej maszynie	0,40
	0,395970821
	0,39439702
	0,439607859
	0,395722151
Klient i serwer na jednej maszynie, po jednym workerze na oddzielnych maszynach	13,36094618
	11,12768888
	11,49362636
	15,75383067
	14,24855375

Obliczenia ponownie były przeprowadzone w dwóch konfiguracjach, jednak w trybie drugim czasy wykonania zadania znacząco spadły dla obu konfiguracji.

4. Wnioski

Spadek czasów wykonania jest najprawdopodobniej spowodowany redukcją interakcji z kolejką. Z tego można wywnioskować, że takowe interakcje są najbardziej kosztownymi operacjami, co najprawdopodobniej jest związane z aspektem sieciowym.

W kontekście pamięci rozproszonej, kluczowe jest efektywne zarządzanie danymi oraz minimalizacja opóźnień związanych z komunikacją między węzłami. W przypadku pracy na maszynach zewnętrznych, gdzie klient oraz serwer są umieszczone na jednej maszynie, a workerzy na innych komputerach, interakcje sieciowe mogą wprowadzać znaczące opóźnienia w przekazywaniu danych pomiędzy węzłami, co z kolei może negatywnie wpływać na wydajność całego systemu. Optymalizacja sposobu przekazywania danych oraz wybór odpowiedniej infrastruktury sieciowej mogą przynieść korzyści w postaci skrócenia czasów wykonania i poprawy efektywności obliczeń.