Laboratorium Programowanie Równoległe i Rozproszone

Projekt - sprawozdanie końcowe
Rozwiązywanie układów równań liniowych w MPI4py

Jacek Kozieja, Piotr Piwoński

15 czerwca 2016

Spis treści

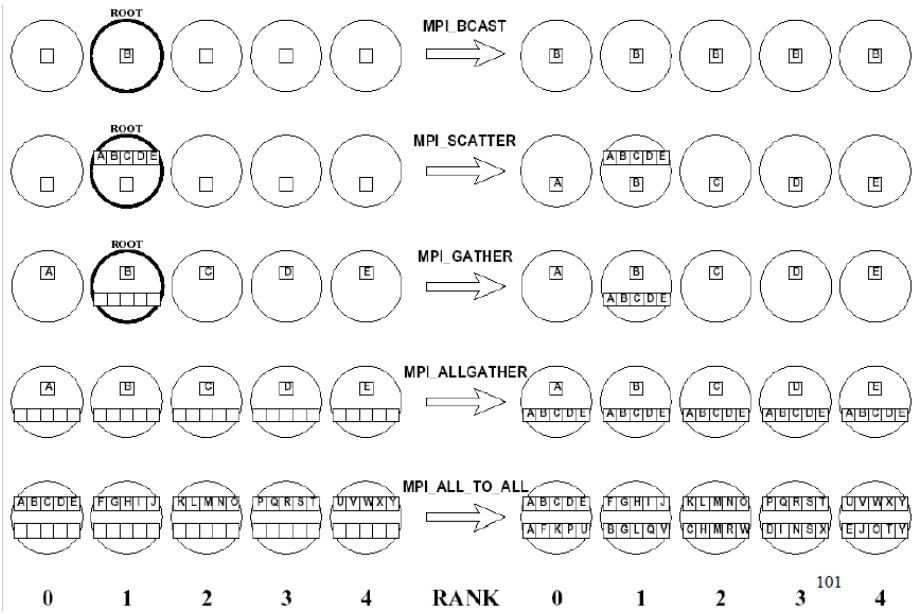
1 Opis ogólny

1.1 Poruszany problem

Zadaniem programów było rozwiązywanie rozbudowanych układów równań liniowych. WYkorzystana została do tego metoda eliminacji Gaussa, oraz Doolittle'a. W celu zwiększenia wydajności dla dużych zestawów danych, algorytmy są wykonywane wielowątkowo.

1.2 MPI4py

MPI4py to implementacja standardu MPI dla języka Python, wykorzystana przy wykonaniu projektu. Bazuje ona na standardowej implementacji MPI dla C++, jednak zmienia składnię i semantykę na tą charakterystyczną dla języka Python. Interfejs MPI ma na celu dostarczenie wirtualnej topologii, synchronizacji oraz zapewnienie komunikacji pomiędzy zestawem procesów (które zostały przypisane do węzłów/serwerów/instancji komputerów) w niezależny od języka sposób, przy podobnej do niego składni. Programy MPI zawsze współpracują z procesami, aczkolwiek programiści powszechnie odnoszą się do procesów jako procesorów. Zazwyczaj dla uzyskania maksymalnej wydajności, każdy CPU (lub rdzeń w wielordzeniowym procesorze) ma przypisany pojedynczy proces. Przypisanie to ma miejsce w czasie rzeczywistym przez agenta, który uruchamia program MPI (zazwyczaj nazywany mpirun lub mpiexec).
Kolektywne sposoby komunikacji w MPI:



2 Opis funkcjonalności

Zasadniczą funkcjonalnością programów jest obliczenie rozwiązań układu równań i wyświetlenie na ekranie. Jako argumenty wywołania przyjmują nazwę pliku z macierzą.

3 Format danych

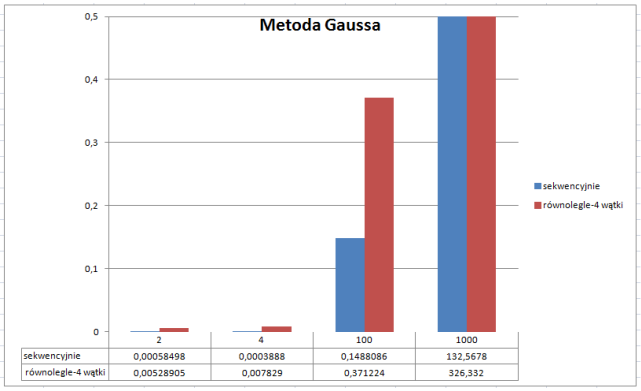
Program jako argument wywołania przyjmuje nazwę pliku z danymi, w którym format zawartych informacji wygląda jak niżej:
11 12 13 14 1
21 22 23 24 2
31 32 33 34 3
41 42 43 44 4
Gdzie po lewej znajdują sie kolejne wiersze i kolumny macierzy A, zaś ostatnia, skrajnie prawa kolumna to wartości macierzy b.

4 Sposób działania

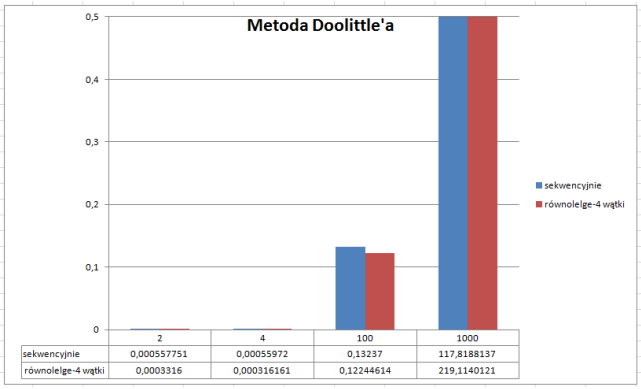
Skrypty gaussP.py i dolitleP.py to wersje algorytmów zrównoleglone w miarę możliwości. Mogą być uruchamiane na dowolnej liczbie wątków. Skrypty gauss i dolitle to wersje sekwencyjne, używane do testów i będące punktem wyjściowym do tworzenia wersji równoległej. Skrypt gen.py służy do generowania losowych danych wejściowych - jako argument wywołania przyjmuje wymiar żądanej macierzy.

5 Testy

Wygenerowane dane w plikach test1.txt do test4.txt to macierze o wymiarach kolejno 2, 4, 100, 1000. Dla kontroli poprawności wyników porównano wyniki wersji zrównoleglonej z sekwencyjną. Testy te wypadły pomyślnie. Następnie wykonano serię testów porównujących wydajność dla różnych wymiarów macierzy. Ich wyniki zamieszczono poniżej.



	2	4	100	1000
sekwencyjnie	0,00058498	0,0003888	0,1488086	132,5678
równolegle-4 wątki	0,00528905	0,007829	0,371224	326,332



	2	4	100	1000
sekwencyjnie	0,000557751	0,00055972	0,13237	117,8188137
równolelge-4 wątki	0,0003316	0,000316161	0,12244614	219,1140121

Odległość między wynikami była zbyt duża by zmieścić je wszystkie na 1 wykresie, dlatego wartości największe zostały obcięte - prawdziwe wyniki można odczytać z tabel pod wykresami.

6 Wnioski

Przy metodzie Gaussa nakład na wymianę danych jest zbyt duży by skutecznie zrównoleglić program. Podstawowe metody komunikacji w MPI jak send/receive czy scatter/gather są w tym wypadku nieskuteczne. Nawet jeśli w niektórych krokach możemy wyliczać coś równolegle to po chwili i tak dane nalezy scalić do pierwotnej formy. Udział części którą da się zrównoleglić jest niewielki, więc zgodnie z prawem Amdahla - możliwości przyspieszenia mocon ograniczone. Jeśli chodzi o metodę Dollitle'a, zrównoleglenie również może dotyczyć wyłącznie 1 wewnętrznej pętli.
Pomimo prostoty użytkowania MPI, rozwiązywanie układów liniowych wydaje się zadaniem nietrywialnym pod kątem potencjalnego zrównoleglania.