I. TEMATY

1. Mnożenie macierzy

Rozważmy problem mnożenia macierzy *C* = *A.B* , gdzie *A* , *B* i *C* są macierzami wymiaru *N* x *N.* Liczba operacji niezbędna do wykonania takiego mnożenia jest proporcjonalna do $N^{3.}$ Poszczególne elementy macierzy C_{ij} oblicza się z poniższego wzoru.

$$C_{ij} = \sum_{k=0}^{N-1} A_{ik} B_{kj}$$

Jako że elementy C_{ij} mogą być obliczane niezależnie, problem nadaje się do zrównoleglenia. Należy przeprowadzić analizę możliwości zrównoleglenia problemu N wymiarowego gdy do dyspozycji jest P maszyn (P. < N) połączonych kanałem komunikacyjnym lub pracujących na wspólnej pamięci. Następnie zaimplementować algorytm w wybranym środowisku i porównać metodę procesów komunikujących się poprzez komunikaty z metodą opartą na komunikacji przez pamięć współdzieloną.
Referencje [1], [2]

Temat 1.1 Mnożenie macierzy - model procesów komunikujących się poprzez komunikaty i pamięć współdzieloną

2. Sortowanie liczb – metoda bąbelkowa

Dany jest ciąg N liczb Z(N) = $\{x_1,x_2,...,x_N\}$. Zadanie polega na posortowaniu tego zbioru liczb. Zadanie to daje się zrównoleglić na P maszyn. Ciąg Z(N) można podzielić na P podciągów $Z_1(N_1)$, ...,$Z_p(N_p)$ gdzie $N_1+N_2+...N_p=N$ które należy posortować na oddzielnych maszynach lub procesorach. Następnie posortowane podciągi podlegają procesowi scalania w wynikowy ciąg posortowany. Należy przeprowadzić analizę problemu, zaprojektować algorytm równoległy oraz sposób implementacji w wybranym środowisku. Jako algorytm sortowania należy użyć sortowania bąbelkowego.

Temat 2.1 Sortowanie bąbelkowe - model procesów komunikujących się poprzez komunikaty

3. Równoległe wyszukiwanie informacji (wyszukiwarka internetowa)

Problem polega na wyszukiwaniu pewnego wzorca (łańcucha) w zbiorze plików tekstowych (bibliotece). Przykładem może być wyszukiwarka internetowa lub dostępny w Uniksie program `grep` który przeszukuje pliki zawarte w pewnym katalogu. Jego zadaniem jest znalezienie danego łańcucha w pliku tekstowym. Jeżeli łańcuch taki w pliku zostanie znaleziony, należy wypisać na konsolę ścieżkę do pliku, nazwę pliku, numer linii w której znaleziono łańcuch i zawartość tej linii. Należy zaprojektować równoległą wersję tego programu.

Temat 3.1 Równoległe wyszukiwanie informacji

4. Rozwiązanie problemu komiwojażera

Problem komiwojażera (*ang. TSP - Travel Salesman Problem*), jest to zagadnienie z teorii grafów, polegające na znalezieniu minimalnego cyklu Hamiltona w grafie. Nazwa pochodzi od typowej ilustracji problemu przedstawiającej go z punktu widzenia wędrownego sprzedawcy (komiwojażera). Dane jest N miast, które komiwojażer ma odwiedzić oraz odległość pomiędzy każdą parą miast. Należy znaleźć najkrótszą trasę wychodzącą z miasta początkowego i przechodzącą jednokrotnie przez wszystkie pozostałe miasta bez powtórzeń, po czym należy wrócić z ostatniego odwiedzonego miasta z powrotem do miasta wyjściowego.

Dane:

n - liczba miast, $n \in \mathrm{Z}^+$,

c_{ji}, $i, j \in \{1, \ldots, n\}, i \neq j$ - odległość między miastem i a miastem j, $c_{ji} = c_{ij}, c_{ji} \in \mathrm{R}^+$.

Zadanie:

Znaleźć permutację miast π minimalizującą kryterium $\sum_{j=1}^{n-1} c_{\pi(j)\pi(j+1)} + c_{\pi(n)\pi(1)}$.

Istnieje wiele algorytmów rozwiązania tego problemu. Najprostszym jest metoda przeglądu zupełnego. Znane jest też wiele algorytmów przybliżonych. Należy przedstawić metodę rozwiązania problemu na komputerze równoległym posługując się wybranym algorytmem.
Literatura:
[1] http://www.mini.pw.edu.pl/miniwyklady/grafy/prob-komiw.html

Temat 4.1 Problem komiwojażera

5. Łamanie szyfru metodą brutalnej siły

Do zaszyfrowania hasła dostępu użyto metody szyfrowania z kluczem. Znany jest wynik szyfrowania dla pewnego hasła zaszyfrowanego danym kluczem. Należy znaleźć klucz szyfrujący. Jako metodę szyfrującą można użyć funkcji crypt (metoda szyfrowania hasła w Unixie), lub jeszcze metody. Metoda łamania szyfru polega na przeglądzie zupełnym zbioru kluczy. Dla pewnego klucza otrzymamy znany wynik szyfrowania. Poszukiwanie może być prowadzone równolegle. Przeprowadź analizę przypadku i zaprojektuj równoległy algorytm obliczeń.

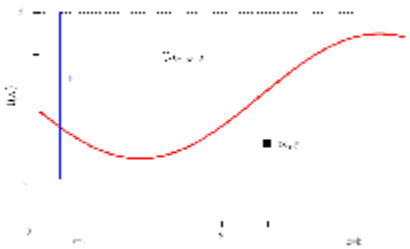
http://www.cypherix.co.uk/cryptainerle/index.htm
http://sun.iinf.polsl.gliwice.pl/~sdeor/students/srpp/problem2.pdf
http://www.bezpieczenstwoit.pl/Kryptografia.html
http://members.value.com.au/christie/usenix91.htm

Temat 5.1 Łamanie szyfru - model programowania z pamięcią współdzieloną

Temat 5.2 Łamanie szyfru - model komunikujących się procesów

6. Całkowanie metodą Monte Carlo

Jest stosowana do modelowania matematycznego procesów zbyt złożonych (obliczanie całek, łańcuchy procesów statystycznych), aby można było przewidzieć ich wyniki za pomocą podejścia analitycznego. Istotną rolę w metodzie MC odgrywa losowanie (wybór przypadkowy) wielkości charakteryzujących proces.



Chcemy policzyć całkę z funkcji *f(x)* w przedziale *[a, b],* czyli

$$I = \int_a^b f(x)dx \ . \qquad (1)$$

Niech *f(x)* w tym przedziale spełnia nierówności: c < *f(x)* < *d.* Za pomocą generatora liczb losowych wyznaczamy *N* par niezależnych od siebie przypadkowych liczb *(xi,yi)* takich, że spełnione są następujące nierówności: *a* < *xi* < *b* oraz *c* < *yi* < *d.* Następnie wyznaczamy liczbę par N_p spełniających warunek: *yi < f(xi)* Oszacowaniem I_N Monte Carlo całki jest liczba :

$$I_N = \frac{N_p}{N} A \,, \tag{3}$$

gdzie A = (b — a)(d — c). Innymi słowy – całka z funkcji f(x) czyli I_N równa jest iloczynowi pola prostokąta A oraz stosunku liczby punktów które znalazły się pod wykresem f(x) do całkowitej liczby punktów wylosowanych w obszarze prostokąta A. Zgodnie z twierdzeniem o wartości średniej :

$$I = (b-a)\langle f \rangle \tag{4}$$

gdzie <f> jest wartością średnią funkcji *f(x).* Średnią liczymy zgodnie z metodą MC jako

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

gdzie *xi (i = 1 , . . . , N)* jest ciągiem *N* liczb przypadkowych jednorodnie rozłożonych w przedziale [a, b]. Dla funkcji o d zmiennych f (X 1 , . . . X d) , przy czym każda spośród zmiennych xj przyjmuje wartości z przedziału [a,j,bj]. Całka d-wymiarowa ma postać:

$$I = \int_{a_1}^{b_1} dx_1 \ \ldots \int_{a_d}^{b_d} dx_d \ f(x_1, \ldots, x_d) \,. \tag{6}$$

Oszacowaniem Monte Carlo powyższej całki jest:

$$I_N = \frac{\Omega}{N} \sum_{i=1}^{N} f(x_{1i}, \ldots, x_{di}) \tag{7}$$

gdzie Ω = (b1 - a1) x ... x (bd - ad) jest d-wymiarową objętością, a ($x1_i$ ..., xdi) są wybrane losowo według rozkładu jednorodnego z przedziałów [a_n, b_n], przy czym *n* = 1,…, *d.*

Należy opracować równoległą wersję algorytmu obliczania masy bryły trójwymiarowej, zaimplementować ją i przetestować.

Literatura:
[1] R. Zieliński, R. Wieczorkowski, *Komputerowe generatory liczb losowych"*, WNT 1997.
[2] R. Wit, *Metody Monte Carlo – wykłady"*, Wydawnictwo Politechniki Częstochowskiej, 2004.

Temat 6.1 Obliczanie masy bryły metodą Monte Carlo

7. Testowanie pakietu MPI

Pakiet MPI jest narzędziem do tworzenia środowiska obliczeń równoległych opartego o model komunikujących się procesów w środowisku systemu Linux lub Windows. Warunkiem realizacji projektu jest dostęp do dwóch połączonych siecią lokalną komputerów pracujących pod kontrolą systemu Linux lub Windows. Należy skonfigurować pakiet, uruchomić go i przeprowadzić kilka testów (na przykład znajdowanie liczb pierwszych w przedziale) porównując czas wykonania testu w środowisku MPI na 2 lub więcej maszynach z czasem wykonania na jednej maszynie.
Literatura: Strona domowa projektu :http://www-unix.mcs.anl.gov/mpi/

Temat 7.1 Testowanie pakietu MPI

8. Wykrywanie konturów w obrazie

Wykrywanie konturów w obrazie może być przeprowadzone z wykorzystaniem operatora Sobela.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Jasność pixela w punkcie oblicza się według wzoru: $|G| = \sqrt{Gx^2 + Gy^2}$ lub w przybliżeniu |G| = |Gx| + |Gy|. Zapis ten oznacza że aby otrzymać jasność danego punktu wykonuje się sumowanie jasność punktów sąsiednich pomnożonych przez współczynniki macierzy filtra. Aby otrzymać obraz po konturowaniu należy przekształcenie zastosować do kolejnych pixeli obrazu. Łatwo dokonać dekompozycji obszaru obrazu dzieląc go na pewną liczbę podobszarów dla których obliczenia można przeprowadzić równolegle.
Literatura: na stronie projektu

Temat 8.1 Konturowanie obrazu

9. Wygładzanie obrazu

Celem wygładzania obrazu jest usunięcie z niego szumów. Algorytm realizuje filtr dolnoprzepustowy. Polega on na zastąpieniu każdego punktu wartością średnią z punktów sąsiednich i punktu bieżącego. Procedura powtarzana jest N razy. Dekompozycji obszaru obrazu dokonać można dzieląc go na pewną liczbę podobszarów dla których obliczenia można przeprowadzić równolegle.

Temat 9.1 Wygładzanie obrazu

10. Rozwiązywanie układu równań liniowych

Rozwiązać duży układ równań liniowych metodą eliminacji Gaussa lub Gausa-Jordana.
Metoda eliminacji Gaussa polega na przekształceniu danego poniżej układu równań liniowych do postaci trójkątnej.

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 && a`_{11}x_1 + a`_{12}x_2 + \ldots + a`_{1n}x_n = b`_1 \quad (2)\\ &a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \quad \text{Þ} && a`_{22}x_2 + \ldots + a`_{2n}x_n = b`_2\\ &\ldots && \ldots\\ &a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n && a`_{nn}x_n = b` \end{aligned}$$

Dla pierwszej kolumny przekształcenie wykonuje się dzieląc pierwszy wiersz przez a_{11}. Następnie dla każdego kolejnego wiersza i dzielimy wiersz przez a_{i1} i odejmujemy go od wierszy 2,3,...n. Postępowanie to powtarza się dla kolejnych kolumn j = 2,3,...,n. Większość operacji arytmetycznych które należy wykonać to mnożenie/dzielenie wektora przez liczbę i odejmowanie wektorów. Operacje te można przeprowadzić równolegle.

Temat 10.1 Równoległe rozwiązywanie układu równań liniowych

Literatura
[1] Metoda eliminacji Gaussa
http://apollo.astro.amu.edu.pl/PAD/index.php?n=Dybol.DydaktykaEliminacjaGaussa

II Zawartość projektu

Projekt oddajemy w postaci pisemnej. Projekt powinien zawierać:

1. Analizę problemu ze szczególnym uwzględnieniem wyodrębnienia części dających się przetwarzać równolegle.
2. Opis metody rozwiązania problemu, podział na wykonywane równolegle procesy, komunikacja pomiędzy procesami, przekazywane parametry. Ten etap jest jeszcze abstrakcyjny, jako notację zalecane jest użycie, pseudokodu lub schematów blokowych.
3. Rozwiązanie problemu przy użyciu wybranej metody
4. Analizę czasową problemu – ile czasu zaoszczędzamy wykonując przetwarzanie równoległe przy użyciu p. procesorów, ile tracimy na komunikację.

III Stosowane metody

Metody używające pamięci dzielonej

1. Wątki języka JAVA (klasa thread)
2. Procesy i wątki systemu Linux (biblioteka Pthreads)
3. Procesy i wątki systemu Windows.
4. System OpenMP

Metody komunikujących się procesów:

1. Gniazdka języka JAVA (pakiet java.net)
2. Gniazdka w systemie Windows (np. C++ Builder, Delphi)
3. Biblioteka gniazdek systemu Linux
4. System MPI

Inne metody także mogą być używane.

IV Sprzęt

Jako środowisko sprzętowe wykorzystać można

1. Komputer z procesorem wielordzeniowym
2. Klaster składający się z co najmniej 2 komputerów (procesor jedno lub wielordzeniowy)
3. Kartę graficzną o ile zawiera więcej niż jedną jednostkę przetwarzającą GPU (ang. Grafic Processor Unit).

IV Literatura

Komputery i systemy równoległe:
[1] Ian Foster Designing and Building Parallel Programs, http://www-unix.mcs.anl.gov/dbpp
[2] Obliczenia Równoległe i Rozproszone, praca zbiorowa pod red. Andrzeja Karbowskiego i Ewy Niewiadomskiej-Szynkiewicz http://www.ia.pw.edu.pl/~karbowsk/orr/#materialy
[3] Zbigniew Czech, Wprowadzenie do obliczeń równoległych, PWN Warszawa 2010.

Wątki, biblioteka Pthreads:
[6] M. Mitchel, J. Oldham, A. Samuel, LINUX Programowanie dla zaawansowanych, RM Warszawa 2002.
[7] T. Wagner, Getting Started With POSIX Threads, http://dis.cs.umass.edu/~wagner/threads_html/tutorial.html

Programowanie interfejsu gniazdek:
[8] M. Gabassi, B. Dupouy, Przetwarzanie rozproszone w systemie UNIX, Lupus 1995
[9] K. Haviland, D. Gray, B. Salama; UNIX Programowanie systemowe, RM Warszawa 1999.
[10] W. Richard Stevens, Programowanie zastosowań sieciowych w systemie UNIX, WNT Warszawa 1996.

Środowisko MPI:
[11] Środowisko MPI (Message passing interface) – strona domowa http://www-unix.mcs.anl.gov/mpi/index.html
[12] MPI A Complete Reference http://www.netlib.org/utk/papers/mpi-book/mpi-book.html
[13] Instalation and users Guide for MPICH - http://www-unix.mcs.anl.gov/mpi/mpich/

OpenMP
[16] Strona domowa projektu, http://openmp.org/wp/
[17] Paul Graham, OpenMP A Parallel Programming Model for Shared Memory Architectures http://www.epcc.ed.ac.uk/epcc-tec/documents
[18] OpenMP - http://www.ci.pwr.wroc.pl/~pmazur/publikacje/2003_OpenMP_ProDialog15.pdf

Inne źródła:
[16] [Wikipedia – wolna encyklopedia - http://pl.wikipedia.org/wiki/Sortowanie_szybkie
[17] Witryna projektu OpenMosix, http://openmosix.sourceforge.net/