Wydział Fizyki i Informatyki Stosowanej

Systemy równoległe i rozproszone

Drzewa wszystkich najkrótszych ścieżek - algorytm Dijkstry Aplikacja PGAS

Arkadiusz Kasprzak
Aleksandra Poręba

2 czerwca 2020

Spis treści

1 Wstęp

Niniejszy dokument stanowi dokumentację drugiego projektu wykonanego w ramach przedmiotu *Systemy równoległe i rozproszone*. Jego tematem było stworzenie, z wykorzystaniem technologii *Unified Parallel C*, aplikacji implementującej algorytm Dijkstry.

1.1 Algorytm Dijkstry

Zaimplementowany w ramach projektu algorytm Dijkstry jest algorytmem poszukiwania najkrótszych ścieżek z wybranego wierzchołka do wszystkich pozostałych w grafie (skierowanym lub nieskierowanym) o nieujemnych wagach krawędzi. Algorytm ten oparty jest na metodzie zachłannej: w każdym kroku wybierany jest wierzchołek, do którego dotarcie wiąże się z najmniejszym kosztem.

tutaj mozna jeszcze cos dopisac i ewentualnie pseudokod

1.2 Zastosowane technologie

Przygotowana aplikacja napisana została w języku programowania C, z wykorzystaniem technologii UPC (*Unified Parallel C*).

tutaj jakies szczegoly dopisac

2 Struktura projektu

3 Działanie projektu

4 Obsługa programu

Niniejsza część dokumentacji przedstawia informacje związane z kompilacją i uruchomieniem projektu.

4.1 Obsługa projektu na pracowni WFiIS

W celu ułatwienia kompilacji i uruchomienia projektu na pracowni Wydziału Fizyki i Informatyki Stosowanej przygotowane zostały dedykowane pliki `Makefile` oraz skrypty do obsługi programu. Znajdują się one w katalogu `build_uni`. Aby dokonać koniecznej konfiguracji środowiska oraz, następnie, kompilacji programu należy wykonać następujące polecenia:

```
cd build_uni
source setup.sh
make
```

Plik `setup.sh` pozwala na skonfigurowanie zmiennych środowiskowych oraz produkuje plik `nodes`. Następnie, aby uruchomić program należy użyć polecenia:

```
make run VERTEX=V FILE=F PROC_COUNT=P HOSTS=H
```

gdzie `V` oznacza wierzchołek startowy (domyślnie: 0), `F` to ścieżka do pliku z danymi (domyślnie: `../data/graph.dat`), `P` to liczba procesów, które zostaną użyte do wykonania algorytmu (domyślnie: 1) a `H` to ścieżka do pliku z węzłami (domyślnie: `nodes`). Inne opcje udostępnione przez plik `Makefile` to:

- `make build` - wykonanie procesu kompilacji - tożsame z poleceniem `make` bez argumentów
- `make run` - uruchomienie programu z domyślnymi opcjami
- `make clean` - przywraca projekt do stanu początkowego
- `make docs` - tworzy dokumentację za pomocą narzędzia Doxygen
- `make install` - kopiuje plik wykonywalny do aktualnego katalogu

Wynik działania programu zapisywany jest do pliku `resultsUPC.txt`.

4.2 Dane wejściowe

Program jako jedną z danych wejściowych przyjmuje ścieżkę do pliku z grafem zapisanym w postaci macierzy sąsiedztwa. W pierwszej linii pliku powinna znajdować się liczba wierzchołków kodowanego grafu. Kolejne linie powinny odpowiadać wierszom macierzy sąsiedztwa. Wagi powinny być zapisane w formie dodatnich liczb rzeczywistych.

Listing 1: Przykładowy plik wejściowy.

```
7
0.0000 5.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 7.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 1.0000 0.0000 0.0000 0.0000
0.0000 4.0000 0.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 2.0000 0.0000 0.0000 0.0000 0.0000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 10.0000
0.0000 0.0000 0.0000 0.0000 0.0000 2.0000 0.0000
```

4.3 Format pliku wynikowego

Wyniki wygenerowane przez program zapisywane są do pliku tekstowego o nazwie `resultsUPC.txt`. Wyprodukowany plik ma następujący format:

Listing 2: Przykładowy plik z wynikami

```
============ RESULTS ============
Distance from vertex 0 to 0: 0.00
Distance from vertex 0 to 1: 5.00
Distance from vertex 0 to 2: 12.00
Distance from vertex 0 to 3: 13.00
Distance from vertex 0 to 4: inf
Distance from vertex 0 to 5: inf
Distance from vertex 0 to 6: inf
============= PATHS =============
0,
0, 1,
0, 1, 2,
0, 1, 2, 3,
Vertex 4 unreachable from source vertex.
Vertex 5 unreachable from source vertex.
Vertex 6 unreachable from source vertex.
```

W pierwszej części pliku znajdują się wyliczone koszty dotarcia z wierzchołka źródłowego do pozostałych wierzchołków w grafie. Jeśli dotarcie nie było możliwe, zapisywana jest wartość `inf`. W drugiej części pliku znajdują się wyliczone ścieżki - ponownie, jeśli wierzchołek był nieosiągalny, jest to odnotowywane.

Literatura

[1] dr inż. Piotr Gronek - wykład z przedmiotu *Systemy równoległe i rozproszone* prowadzony na Wydziale Fizyki i Informatyki Stosowanej AGH

[2] Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar *Introduction to Parallel Computing, Second Edition.* Addison-Wesley, 2003.

[3] Tarek El-Ghazawi, William Carlson, Thomas Sterling, Katherine Yelick *UPC: Distributed Shared Memory Programming*

[4] Dokumentacja projektu *Berkeley UPC* - `https://upc.lbl.gov/docs/` (dostęp: 01.06.2020)

[5] UPC Consortium - *UPC Required Library Specifications Version 1.3*