

Чланови групе: Кристина Кнежевић и Тамара Елез

Имплементација за прве двије теме је планирана као конзолна апликација кодирана у програмском језику C, уз паралелизацију реализовану помоћу OpenMP технологије, а за трећу као WFA апликација која користи System.Threading библиотеку за рад са нитима у .NET окружењу.

Теме:

1) Упоредивање алгоритама за сортирање

Упоредићемо Merge sort и Quick sort алгоритме за сортирање који су засновани на принципу "подијели и владај" и оба имају временску сложеност $O(n \log n)$. То их чини једнима од најбржих алгоритама у пракси и популарним за сортирање великих низова података.

Основна идеја иза Merge sort –а је дијелити низ на пола све док се не дође до појединачних елемената (једноелементних низова), а затим комбиновати их у сортирани низ. Паралелизација се може извршити додјељивањем свакој нити за сортирање по један дио низа.

Основна идеја Quick sort –а: Прво се одабере елемент из низа, назван "пивот" (може бити било који елемент низа, а најчешће се узима први, последњи или елемент у средини низа). Низ се подијели на два подниза, тако да се елементи мањи од пивота премјештају лијево од њега, док се елементи већи од пивота премјештају десно од њега. На крају овог корака, пивот се налази на својој коначној позицији у сортираном низу. Поступак се рекурзивно понавља за поднизове лијево и десно од пивота, све док поднизови не постану довољно мали да се сортирају без додатних рекурзивних позива.

Паралелизација се може извршити за рекурентне позиве алогиртма или при подјели елемената на веће и мање од пивота.

Циљ апликације је да видимо да ли ће паралелизација бити корисна за ове алгоритме који се већ сматрају брзим у секвенцијалном извршавању.

Рад конзолне апликације:

Корисник уноси дужину низа који треба да се сортира. Низ се попуњава рандом цијелим позитивним бројевима. Прво се врши секвенцијално сортирање и испис чланова низа уз испис утрошеног времена. Затим се процес понавља након паралелизације. Упорјеђују се времена извршавања, с циљем откривања ефикаснијег начина извршавања и који је од ова два алгоритма бржи.

2) Флојд-Варшалов алгоритам (Floyd-Warshall)

У рачунарству Флојд-Варшалов алгоритам је алгоритам анализе графова за налажење најкраћих путања у тежинском, усмјереном графу и у рачунарству се може користити за рутирање.

Улаз Флојд-Варшаловог алгоритма је матрица тежина и у њој се налазе нуле и позитивни бројеви који представљају јединице удаљености између чворова (нпр. километри, метри, временска јединице). Ако се у матрици налази вриједност 0, то значи да постоји изравна веза (грана) између та два чвора. Претпостављамо да је граф потпуно повезан, тако да у матрици нема бесконачних вриједности које би јавиле у случају да нема везе између нека два чвора. Након примјене алгоритма, резултат ће бити матрица најкраћих удаљености.

Паралелизација овог алгоритма је корисна за графове са великим бројем чворова. Умјесто да се итерира по сваком чвору и ажурира матрицу путања, може се паралелизовати тако да свака нит обрађује различите чворове, чиме се убрзава израчунавање најкраћих путева. Када су сви парови обрађени, резултати се могу комбиновати како би се добила коначна матрица најкраћих путања.

Рад конзолне апликације:

Корисник уноси број чворова графа (n). Формира се квадратна матрица $n \times n$. Матрица се попуни нулама и рандом позитивним вриједностима.

Конзолна апликација омогућава секвенцијално одређивање матрице најкраћих растојања и испис колико времена је потребно за такво извршавање, а затим паралелно извршавање уз испис утрошеног времена и колико времена је свака од нити кориштена.

3) n – ти члан низа простих бројева

WFA апликација у којој корисник уноси два цијела позитивна броја. Ако нису унесени адекватни бројеви, захтјева поновни унос. Врши се пребројавање и испис свих простих бројева који се налазе између та два броја, секвенцијално и паралелно и упоређује се вријеме извршавања у оба случаја. Прво се рачуна величина интервала, односно разлика између унесених бројева, да би одредили максималан број могућих простих бројева, затим се величина интервала дијели са бројем нити ради добијања мањих интервала који се додјељују за обраду појединачним нитима. Уколико нема простих бројева у задатом интервалу, корисник се обавјештава поруком.