

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

PARALELNÍ ARCHITEKTURY POČÍTAČŮ

TECHNICKÁ ZPRÁVA

Hledání nejkratších cest v grafu

Autoři:

Vojtěch MYSLIVEC

Zdeněk NOVÝ

8. dubna 2015



Abstrakt

Účelem této práce je sumarizovat výsledky měření řešení problému hledání nejkratších cest v grafu (NCG). Práce se zaměřuje na řešení problému Dijkstrovým a Floyd-Warshallovým algoritmem a porovnání sekvenční a několika paralelních implementací.

Klíčová slova Dijkstra, Floyd, Warshall, nejkratší cesty, NCG, OpenMP, Cuda

1 Úvod

Tato práce se zabývá implementací dvou algoritmů hledání nejkratších cest v grafu. Jedná se o implementaci sekvenčním algoritmem, který je poté paralelizován pro procesor a pro grafickou kartu. Pro jednotlivé algoritmy je provedeno měření, které si klade za cíl určit zrychlení paralelních algoritmů proti sekvenčnímu.

2 Hledání nejkratších cest v grafu

2.1 Definice

Hledání nejkratších cest v grafu je NP-úplná grafová úloha, jejímž cílem je nalézt v zadaném grafu nejkratší cesty mezi všemi možnými dvojicemi uzlů A a B [3].

2.2 Algoritmy

2.2.1 Dijkstrův algoritmus

Dijkstrův algoritmus slouží k nalezení všech nejkratších cest ze zadaného uzlu do všech ostatních uzlů grafu. Graf nesmí obsahovat hrany se zápornou délkou [1].

Princip Dijkstrův algoritmus je zobecněné prohledávání grafu do šířky, při kterém se vlna šíří na základě vzdálenosti od zdrojového uzlu. K uchovávání uzlů slouží prioritní fronta, která je řazena podle vzrůstající vzdálenosti od zdroje. V každém kroku algoritmu je vybrán uzel s nejmenší vzdáleností a pro každého souseda je vypočítána jeho vzdálenost od zdrojového uzlu [1].

2.2.2 Floyd-Warshallův algoritmus

Floyd-Warshallův algoritmus slouží k nalezení nejkratších cest mezi všemi dvojicemi uzlů v grafu. Graf může obsahovat hrany, ale nikoliv cykly, záporné délky [2].

Princip Floyd-Warshallův algoritmus pracuje s maticí sousednosti, kde hrana je ohodnocena vahou. Na počátku tato matice obsahuje pouze vzdálenosti dvou uzlů, mezi kterými je vedena hrana. V každém kroku je vybrán jeden uzel jako prostředník. Prvek matice sousednosti se přepočítá, pokud je vzdálenost z počátečního do koncového uzlu kratší přes nového prostředníka než bez něj [2].

3 Sekvenční algoritmus

3.1 Společná implementace

Oba algoritmy vycházejí z obecného principu, který je popsán v kapitole 2.2.2. Algoritmy pracují s grafem, který je programu předložen jako soubor, ve kterém je graf ve formě matice sousednosti. Společnou částí je tedy načítání vstupu a jeho kontrola.

3.2 Dijkstrův algoritmus

Protože Dijkstrův algoritmus slouží k hledání nejkratších cest od jednoho zdrojového uzlu, je nutné jej spouštět pro každý uzel grafu. To zajišťuje funkce *dijkstraNtoN*.

3.2.1 Dijkstrův algoritmus z jednoho zdroje

Pro výpočet Dijkstrova algoritmu z jednoho zdrojového uzlu se alokují tři pole o velikosti počtu uzlů. V jednom je uložena vzdálenost daného uzlu od zdrojového, ve druhém předchozí uzel v nalezené nejkratší cestě. Třetí pole určuje, jestli je už uzel uzavřený pro výpočty.

Algoritmus prochází postupně, podle nejmenší vzdálenosti, všechny uzly, které se nacházejí v prioritní frontě. Z daného uzlu vypočítá pro každého svého souseda novou cestu, která by vedla přes uzel samotný a porovná ji s dosavadní vzdáleností souseda. Menší vzdálenost je zapsána do pole vzdáleností a algoritmus pokračuje.

3.3 Floyd-Warshallův algoritmus

Floyd-Warshallův algoritmus obsahuje tři vnořené for cykly a funguje na principu popsaném v 2.2.2. Jako datové struktury používá čtyři pole o veli-

kosti počtu uzlů. Pro každý uzel si algoritmus udržuje aktuální vzdálenosti ke všem uzlům a navíc vzdálenosti z předchozí iterace. Další dvě pole obsahují předchozí uzel v nalezené cestě.

3.4 Implementace

Aktuální implementace je k nahlédnutí i ke stažení na adrese <https://github.com/VojtechMyslivec/PAP-NCG>. Sekvenční algoritmus se nachází ve složce *01_sekvencni*.

4 Paralelní algoritmus pomocí knihovny OpenMP

4.1 Dijkstrův algoritmus

4.1.1 Úprava algoritmu

4.1.2 Vektorizace

4.1.3 Optimalizace

4.2 Floyd-Warshallův algoritmus

4.2.1 Úprava algoritmu

4.2.2 Vektorizace

4.2.3 Optimalizace

4.3 Měření

4.3.1 Analýza

4.3.2 Zhodnocení

5 Paralelní algoritmus pomocí technologie CUDA

5.1 Dijkstrův algoritmus

5.1.1 Úprava algoritmu

5.1.2 Optimalizace

5.2 Floyd-Warshallův algoritmus

5.2.1 Úprava algoritmu

5.2.2 Optimalizace

5.3 Měření

5.3.1 Analýza

5.3.2 Zhodnocení

6 Závěr

Reference

- [1] Mička, P.: Dijkstrův algoritmus. [cit. 2015-04-08].
URL <http://www.algoritmy.net/article/5108/Dijkstruv-algoritmus>
- [2] Mička, P.: Floyd-Warshallův algoritmus. [cit. 2015-04-08].
URL <http://www.algoritmy.net/article/5207/Floyd-Warshalluv-algoritmus>
- [3] Mička, P.: Problém nejkratší cesty. [cit. 2015-04-08].
URL <http://www.algoritmy.net/article/36597/Nejkratsi-cesta>