Politechnika Poznańska
Wydział Informatyki
Instytut Informatyki

Praca dyplomowa inżynierska

Implementacja algorytmu eksploracji danych z użyciem CUDA API

Marcin Jabłoński
Łukasz Kosiak
Piotr Kurzawa
Marek Rydlewski

Promotor:
dr inż. Witold Andrzejewski

Poznań, 2017 r.

„Coś się popsuło“
Zbigniew Stonoga

Spis treści

1 Wstęp

1.1 Wprowadzenie

Informatyzacja życia codziennego, jaka dokonała się w ostatnich latach sprawiła, że każdego dnia często nieświadomie zostawiamy po sobie wiele informacji na swój temat. Nawet z pozoru niewinne dane o naszych przyzwyczajeniach typu "z której półki bierzemy bułki w sklepie" są zapisywane w nieznanych nam systemach informatycznych. Dodając do tego inne usługi świadomie przez nas wykorzystywane - chociażby zapisywanie naszej lokalizacji przez prywatny telefon komórkowy - uzyskujemy dość ponury obraz tego, co jesteśmy w stanie po sobie zostawić. Co gorsza, chcąc czy nie chcąc, musimy się pogodzić z faktem, że dane te mogą zostać wykorzystane w różnym celu. Czy mamy jednak czego się obawiać?

Wbrew pozorom, taka błaha na pierwszy rzut oka informacja może mieć jednak istotne znaczenie dla funkcjonowania przemysłu piekarskiego. Przecież takich informacji codziennie my, klienci, zostawiamy ogromne ilości. Nic nie szkodzi na przeszkodzie, aby spróbować z tych danych odczytać preferencje bądź przyzwyczajenia przeciętnego Kowalskiego na temat jego codziennych zakupów, które mogą w przyszłości zaprocentować - zarówno dla właściciela, jak i klienta. Jest to oczywiście tylko przykład, ale oddaje doskonale fakt przydatności z pozoru nie mających znaczenia prostych czynności człowieka, jakie często przypadkiem rejestrują działające wokół nas systemy.

Pozostaje jednak problem przetworzenia takich danych w celu otrzymania interesującej nas informacji, która byłaby potencjalnie użyteczna. Trzeba pamiętać, że rozmiar takich danych nierzadko sięga terabajtów i w praktyce skuteczna analiza takich danych przez człowieka nie jest możliwa. Musi on zatem w tym celu skorzystać z dobrodziejstw, jakie przynosi mu współczesna technologia.

Problem efektywnego przetwarzania zdążył urosnąć do rangi oddzielnego działu w informatyce. W pracy [1] zasugerowano utworzenie nowej dyscypliny mającej na celu opracowanie technik obliczeniowych rozwiązujących takie problemy, zwanej roboczo odkrywaniem wiedzy w bazach danych (ang. KDD – *Knowledge Discovery in Databases*). Techniki te mają na celu odnajdywanie prawidłowych i potencjalnie użytecznych wzorców w dużych zbiorach danych.

Wspominane wyżej techniki w dużej mierze zależą od rodzaju bazy, a ściślej mówiąc - charakteru danych występujących w niej. W przypadku danych zawierających informację o położeniu zazwyczaj mowa jest o odkrywaniu wiedzy w bazach danych przestrzennych (ang. *spatial data mining*). Takie systemy mogą zawierać atrybut lokalizacji obiektu w danym obszarze, jego opis w formie geometrycznej (np. w postaci wielokątów), a także inne atrybuty nieprzestrzenne. Okazuje się, że tradycyjne metody analizy danych przestrzennych zazwyczaj nie radzą sobie z nimi na tyle efektywnie, by było opłacalne ich użycie w praktyce [3], dlatego też zaczęto szukać nowych sposobów na odkrywanie wiedzy w takich bazach.

W pracy [4] zaproponowano odkrywanie *wzorców kolokacji przestrzennych* (lub krócej: *kolokacji*), czyli zbioru cech przestrzennych występujących w niewielkiej odległości od siebie. Łatwo to można sobie wyobrazić na przykładzie przyrody, gdzie osobniki (gatunki) o podobnych cechach zazwyczaj trzymają się razem. Rozumowanie to działa również dla bliższych współczesnemu człowiekowi cech przestrzennych, np. punktach o podobnej funkcji - stacje, kina, piekarnie, itd. Wraz z rosnącą popularnością obliczeń na kartach graficznych (w dużej mierze spowodowana wprowa-

dzeniem technologii *CUDA* autorstwa firmy NVIDIA) pojawiło się wiele gotowych rozwiązań, pozwalających na efektywne wyszukiwanie kolokacji nawet w bardzo rozbudowanych bazach danych. Przegląd niektórych z nich można znaleźć w pracy [5].

Ostatni rok przyniósł kolejną metodę efektywnego przeszukiwania baz danych w celu odnalezienia kolokacji [6]. Wykorzystuje ona autorski algorytm wyszukiwania maksymalnych klik w grafie rzadkim oraz skondensowane drzewa instancji przechowywaczce kliki instancji dla każdego kandydata do kolokacji (patrz Rozdział 2) w celu zmniejszenia czasu obliczeń oraz ograniczenia wymagań co do pamięci operacyjnej. Algorytm ten jest przedmiotem badań niniejszej pracy zbiorowej.

1.2 Cel i zakres pracy

Celem niniejszej pracy jest analiza wydajności zaproponowanych w pracy [6] rozwiązań z zakresu odkrywania kolokacji przestrzennych dla GPU i CPU.

Zakres pracy obejmuje następujące zadania szczegółowe:

1. **Zapoznanie się z literaturą.** Zapoznanie się z podstawowymi pojęciami dotyczącymi odkrywania danych w bazach danych przestrzennych oraz wyszukiwania wzorców kolokacji przestrzennych jest niezbędne do stworzenia działającej implementacji powyższego algorytmu. Dodatkowo należy zwrócić uwagę na dodatkowe zagadnienia związane z teorią grafów.

2. **Opracowanie wersji równoległej algorytmu eksploracji danych.** Konieczne jest przemyślenie wykorzystania algorytmów pomocniczych dla poszczególnych kroków całego rozwiązania oraz zaproponowanie możliwie najkorzystniejszego rozwiązania biorąc pod uwagę dostępną pamięć operacyjną, czas przetwarzania i przesyłania danych między pamięcią operacyjną a pamięcią karty graficznej.

3. **Implementacja wersji sekwencyjnej i równoległej ww. algorytmu.** Rozwiązanie podane w punkcie drugim powinno zostać zaimplementowane w technologii NVIDIA CUDA dla wersji GPU oraz biblioteki OpenMPI w przypadku odmiany dla CPU.

4. **Przeprowadzenie eksperymentów wydajnościowych.** Analiza wyników testów wydajnościowych implementacji z punktu 3 jest głównym celem tej pracy. Należy zbadać efektywność obu rozwiązań pod względem czasu wykonywania oraz zapotrzebowania na dostępną pamięć.

1.3 Charakterystyka źródeł

Jak już wspomniano, niniejsza praca w dużej mierze opiera się o algorytm zaprezentowany w dokumencie [6]. Do jej opracowania była wymagana wiedza zawarta w innych źródłach, często również o charakterze naukowym.

Głównym źródłem wiedzy na temat kolokacji przestrzennych była rozprawda doktorska dr inż. Pawła Boińskiego [5], która w dużym przekroju omawia ideę kolokacji zaprezentowaną przez Shakara i Huanga w pracy [4], a także prezentuje najpopularniejsze techniki ich odkrywania (metody *Co-location Miner*, *iCPI-tree*). Część rozwiązań wykorzystanych w tych technikach została wykorzystana w trakcie realizacji algorytmu.

Oddzielną kwestią jest literatura książkowa, wykorzystana do zapoznania się z technologią CUDA oraz przyjęcia dobrych praktyk optymalizacyjnych i programistycznych. Tutaj szczególnie należy wymienić popularną pozycję *CUDA w przykładach* autorstwa Shane'a Cooke'a [7], a także *Professional CUDA C Programming* [8] będącą również podstawą do wstępu teoretycznego w rozdziale drugim.

1.4 Struktura pracy

W pracy przedstawiono główne pojęcia związane z wyszukiwaniem kolokacji przestrzennych oraz programowaniem równoległym na procesory graficzne i zawarto je w rozdziale 2. Rozdział 3 poświęcony jest algorytmowi będącemu głównym tematem pracy. Rozdział 4 opisuje implementację tego algorytmu w technologii CUDA, natomiast rozdział 5 prezentuje wyniki przeprowadzonych testów.

W tym miejscu zasadniczo będzie można napisać więcej, jeżeli już te rozdziały zostaną ustalone bądź wstępnie uzupełnione. Poza tym należy ustalić, czy w ogóle potrzebujemy takiego działu dla tak małej pracy. Z drugiej strony, zawsze to jednak te pół strony więcej spamu - borewicz

2 Podstawy teoretyczne

Więcej informacji można znaleźć w książce [2].

3 Algorytm

4 Implementacja

5 Testy efektywnościowe

6 Zakończenie

Bibliografia

[1] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17:37–54, 1996.

[2] Ł. Stanisławowski. *Bogactwo i nędza narodów.* O'reilly, 2013.

[3] Harvey J. Miller and Jiawei Han. Geographic Data Mining and Knowledge Discovery. Taylor & Francis, Inc., Bristol, PA, USA, 2001

[4] S. Shekhar and Y. Huang. Discovering Spatial Co-location Patterns: A Summary of Results. In SSTD 2001, pages 236–256, 2001.

[5] Przetwarzanie zbiorów przestrzennych zapytan neksploracyjnych w srodowiskachzograniczonym rozmiarem pamiecioperacyjnej

[6] A fast space-saving algorithm for maximal co-location pattern mining

[7] CUDA by Example: An Introduction to General-Purpose GPU Programming, Jason Sanders, Edward Kandrot

[8] Professional CUDA C Programming, John Cheng, Max Grossman, Ty McKerche

A Dodatek A

B Dodatek B