

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

<b>1</b>	<b>Wstęp</b>	<b>3</b>
1.1	Wprowadzenie . . . . .	3
1.2	Cel i zakres pracy . . . . .	4
1.3	Charakterystyka źródeł . . . . .	4
1.4	Struktura pracy . . . . .	5
1.5	Podział pracy . . . . .	5
<b>2</b>	<b>Podstawy teoretyczne</b>	<b>6</b>
2.1	Charakterystyka danych przestrzennych . . . . .	6
2.1.1	Modelowanie danych przestrzennych . . . . .	6
2.1.2	Źródła danych przestrzennych . . . . .	6
2.1.3	Relacje . . . . .	6
2.2	Metody eksploracji danych przestrzennych . . . . .	6
2.2.1	Grupowanie przestrzenne . . . . .	6
2.2.2	Klasyfikacja przestrzenna . . . . .	6
2.2.3	Odkrywanie trendów . . . . .	6
2.2.4	Asocjacje przestrzenne . . . . .	6
2.2.5	Kolokacje przestrzenne . . . . .	6
2.3	Odkrywanie kolokacji przestrzennych . . . . .	6
2.3.1	Cecha przestrzenna . . . . .	6
2.3.2	Podstawowe definicje . . . . .	6
2.3.3	Miary kolokacji . . . . .	7
2.3.4	Definicja problemu odkrywania wzorców kolokacji przestrzen- nych . . . . .	7
2.3.5	Przegląd algorytmów odkrywania wzorców kolokacji przestrzen- nych . . . . .	8
<b>3</b>	<b>Algorytm</b>	<b>9</b>
3.1	Konstrukcja tabeli instancji o rozmiarze 2 . . . . .	9
3.2	Obliczanie miary powszechności . . . . .	9
3.3	Generowanie kandydatów na kolokacje maksymalne . . . . .	9
3.4	Proces odcinania . . . . .	9
<b>4</b>	<b>Implementacja</b>	<b>9</b>
<b>5</b>	<b>Testy efektywnościowe</b>	<b>9</b>
<b>6</b>	<b>Zakończenie</b>	<b>9</b>
	<b>Bibliografia</b>	<b>9</b>
<b>A</b>	<b>Dodatek A</b>	<b>9</b>
<b>B</b>	<b>Dodatek B</b>	<b>9</b>

# 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 nieznanym 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 błaża 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 przechowywujące 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 rozprawa doktorska dr inż. Pawła Boińskiego [5], która w dużym przekroju omawia ideę kolokacji zaprezentowaną przez Shakara i Huangą 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*

## 1.5 Podział pracy

**Marcin Jabłoński** w ramach niniejszej pracy wykonał projekt tego i tego, opracował .....

**Łukasz Kosiak** wykonał ....., itd.

## 2 Podstawy teoretyczne

### 2.1 Charakterystyka danych przestrzennych

#### 2.1.1 Modelowanie danych przestrzennych

#### 2.1.2 Źródła danych przestrzennych

#### 2.1.3 Relacje

### 2.2 Metody eksploracji danych przestrzennych

#### 2.2.1 Grupowanie przestrzenne

#### 2.2.2 Klasyfikacja przestrzenna

#### 2.2.3 Odkrywanie trendów

...

#### 2.2.4 Asocjacje przestrzenne

#### 2.2.5 Kolokacje przestrzenne

### 2.3 Odkrywanie kolokacji przestrzennych

Niniejszy rozdział zawiera opisy i definicje pojęć niezbędnych do zrozumienia algorytmu zawartego w rozdziale X.

#### 2.3.1 Cecha przestrzenna

Kluczową kwestią w procesie odkrywania kolokacji jest odpowiednia klasyfikacja obiektów występujących w bazie danych. Każdy zbiór danych przestrzennych, oprócz informacji o lokalizacji obiektu i opisujących go danych nieprzestrzennych powinien zawierać także właściwość pozwalającą na sklasyfikowanie danego obiektu do określonej klasy. Takie przypisanie nazywane jest cechą przestrzenną (ang. spatial feature) lub rzadziej klasą obiektu (ang. object class).

Jako typowy przykład cechy przestrzennej można podać etykietę przypisaną do obiektu na mapie (np. kościół, szkoła, strzelnica). Pozwala ona na jednoznaczne określenie własności przestrzeni w punkcie, gdzie znajduje się obiekt.

#### 2.3.2 Podstawowe definicje

**Definicja 1 (Instancja cechy przestrzennej)** *Niech  $f$  będzie cechą przestrzenną. Mówimy, że obiekt  $x$  jest instancją cechy przestrzennej  $f$ , wtedy i tylko wtedy, gdy obiekt  $x$  jest typu  $f$  oraz jest opisany przez lokalizację i identyfikator.*

**Definicja 2 (Wzorzec i instancja kolokacji)** *Załóżmy  $F$  jako zbiór cech przestrzennych  $F = \{f_1, f_2, \dots, f_m\}$ , a  $FI = FI^{f_1} \cup FI^{f_2} \cup \dots \cup FI^{f_m}$  niech będzie zbiorem ich instancji. Niech  $>_F$  oznacza dowolną relację porządku zdefiniowaną dla zbioru  $F$ . Niech  $f_i$  oznacza  $i$ -tą cechę przestrzenną (ze względu na relację  $>_F$ ), zatem  $\forall i, j \in 1, \dots, m$   $f_i <_F f_j \Leftrightarrow i < j \wedge f_i, f_j \in F$ . Mając daną relację sąsiedztwa  $R$*

(zwrotną i przechodnią) mówimy, że wzorzec kolokacji przestrzennej (w skrócie "kolokacja") jest podzbiorem cech przestrzennych  $c \subseteq F$ , których instancje  $I \subseteq FI$  tworzą klikę ze względu na relację  $R$ . Zbiór wszystkich instancji kolokacji przestrzennej  $c$  jest oznaczany przez  $CI^c$ . Przez długość kolokacji należy rozumieć liczbę elementów w zbiorze cech przestrzennych, który tworzy tę kolokację.

**Definicja 3 (Sąsiedztwo)** Mające daną zwrotną i symetryczną relację sąsiedztwa  $R$ , sąsiedztwem lokalizacji  $l$  nazywamy zbiór lokalizacji  $L = \{l_1, l_2, \dots, l_n\}$ , gdzie  $l_i$  jest sąsiadem  $l$ , tzn. zachodzi  $R(l, l_i) \forall i \in 1, \dots, n$ .

Przykład TODO (oprzeć na przykładzie chińczyków?)

### 2.3.3 Miary kolokacji

**Definicja 4 (Współczynnik uczestnictwa)** Współczynnik uczestnictwa (ang. participation ratio) cechy  $f_i$  w kolokacji  $c$  jest równy procentowemu udziałowi wszystkich instancji cechy  $f_i$  w instancjach kolokacji  $c$ :

$$pr(f_i, c) = \frac{|\pi^{f_i}(CI^c)|}{FI_{f_i}} \quad (1)$$

gdzie  $\pi^{f_i}(CI^c)$  oznacza projekcję relacyjną zbioru instancji  $CI^c$  względem cechy  $f_i$  (z usuwaniem duplikatów).

**Definicja 5 (Indeks uczestnictwa)** Indeks uczestnictwa (ang. participation index) kolokacji  $c$  jest równy najmniejszemu ze współczynników uczestnictwa wyznaczonych dla każdej cechy przestrzennej  $f_i \in c$ :

$$pi(c) = \min_{f_i \in c} pr(f_i, c) \quad (2)$$

Indeks uczestnictwa najczęściej określany jest w literaturze mianem miary powszechności lub krótko powszechności kolokacji.

**Definicja 6 (Maksymalny wzorzec kolokacji przestrzennej)** Niech będzie dana wartość  $min\_prev$  oznaczająca pewien minimalny próg powszechności. Jeżeli  $c = \{f_1, \dots, f_m\}$  jest kolokacją powszechną (tzn.  $pi(c) \geq min\_prev$ ) i nie istnieje żaden nadzbiór  $c$  taki, że powszechność dla tego nadzbioru jest równa co najmniej  $min\_prev$ , kolokacja  $c$  nazywana jest kolokacją maksymalną.

### 2.3.4 Definicja problemu odkrywania wzorców kolokacji przestrzennych

**Definicja 7 (Reguła kolokacyjna)** Reguła kolokacyjna to reguła postaci  $c_1 \rightarrow c_2(p, cp)$ , gdzie  $c_1 \subseteq F$ ,  $c_2 \subseteq F$  i  $c_1 \cup c_2 = \emptyset$ . Potencjalna użyteczność reguły może być mierzona przy pomocy jej powszechności  $p$  oraz prawdopodobieństwa warunkowego  $cp$ .

**Definicja 8 (Prawdopodobieństwo warunkowe)** Prawdopodobieństwem warunkowym  $cp(c_1, c_2)$  reguły kolokacyjnej  $c_1 \rightarrow c_2$  nazywamy stosunek liczby instancji wzorca  $c_1$  w sąsiedztwie instancji wzorca  $c_2$  do liczby wszystkich instancji wzorca  $c_1$ :

$$cp(c_1, c_2) = \frac{|\pi^{c_1}(CI^{c_1 \cup c_2})|}{CI_{c_1}} \quad (3)$$

gdzie  $\pi^{c_1}(CI^{c_1 \cup c_2})$  oznacza projekcję relacyjną instancji wzorca  $CI^{c_1 \cup c_2}$  względem wzorca  $c_1$  (z usuwaniem duplikatów).



**Definicja 9 (Problem odkrywania kolokacji)** *Problem odkrywania kolokacji przestrzennych jest zdefiniowany w następujący sposób. Mając dane:*

- *zbiór cech przestrzennych  $F = \{f_1, f_2, \dots, f_m\}$*
- *zbiór obiektów  $FI = FI^{f_1} \cup FI^{f_2} \cup \dots \cup FI^{f_m}$ , gdzie  $FI^{f_i}, (0 < i \leq m)$  jest zbiorem instancji cechy  $f_i$ , przy czym każda instancja jest opisana przez lokalizację i identyfikator,*
- *symetryczną i zwrotną relację sąsiedztwa  $R$ ,*
- *próg minimalnej powszechności  $min\_prev$  oraz próg minimalnego prawdopodobieństwa warunkowego  $min\_cond$ ,*

*znajdź wszystkie poprawne reguły kolokacyjne z powszechnością nie mniejszą niż  $min\_prev$  i prawdopodobieństwem warunkowym nie mniejszym niż  $min\_cond$ .*

### **2.3.5 Przegląd algorytmów odkrywania wzorców kolokacji przestrzennych**

## **3 Algorytm**

### **3.1 Konstrukcja tabeli instancji o rozmiarze 2**

### **3.2 Obliczanie miary powszechności**

### **3.3 Generowanie kandydatów na kolokacje maksymalne**

### **3.4 Proces odcinania**

## **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 neksploacyjnych w środowiskach z ograniczonym rozmiarem pamięci operacyjnej
- [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**