Maciej WIELGOSZ [2], Dominik ŻUREK [1], Marcin PIETROŃ [1],
Agnieszka DĄBROWSKA-BORUCH [2], Kazimierz WIATR [2]
[1] ACK-CYFRONET AGH, Nawojki 11, 30-950 Kraków
[2] AKADEMIA GÓRNICZO-HUTNICZA, Al. A. Mickiewicza 30, 30-059 Kraków

Równoległa implementacja algorytmu winnowing dla operacji strumieniowej analizy tekstu

Dr inż. Maciej WIELGOSZ

Ukończył studia na AGH (2005), wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki na kierunku Elektronika i Telekomunikacja. Obronił pracę doktorska w 2010 roku. Obecnie jest pracownikiem Katedry Elektroniki AGH i bierze czynny udział w pracach badawczych realizowanych w zespole rekonfigurowalnych systemów obliczeniowych. Jego zainteresowania naukowe dotyczą sprzętowej akceleracji obliczeń, strumieniowej analizy treści oraz architektur sprzętowych dla algorytmów sztucznej inteligencji.

e-mail: wielgosz@agh.edu.pl

Mgr inż. Dominik ŻUREK

Ukończył studia na AGH (2011),wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki na kierunku Elektronika i Telekomunikacja. Pracuje z ACK Cyfronet AGH z działem zespołu Akceleracji Obliczeń. Jego zainteresowania naukowe dotyczą przetwarzania obrazów, tekstu oraz akceleracji obliczeń.

e-mail: dominik.zurek@cyfronet.krakow.pl

Dr inż. Marcin PIETROŃ

Ukończył studia na AGH (2003), wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki na kierunku Elektronika i Telekomunikacja oraz na kierunku Informatyka (2005). Obronił pracę doktorską na Wydziale Informatyki, Elektroniki i Telekomunikacji. Obecnie jest pracownikiem Akademickiego Centrum Komputerowego Cyfronet AGH. Jego zainteresowania naukowe dotyczą obliczeń i algorytmów równoległych oraz data-mining'u .

e-mail: pietron@agh.edu.pl

Dr inż. Agnieszka DĄBROWSKA-BORUCH

Absolwentka kierunku Elektronika i Telekomunikacja Wydziału EAIiE AGH (2002), dr nauk technicznych (2007). Obecnie jest adiunktem w Katedrze Elektroniki AGH oraz członkiem Zespołu Akceleracji Obliczeń ACK CYFRONET AGH. Jej zainteresowania naukowe to kompresja obrazu, systemy czasu rzeczywistego, układy programowalne oraz rekonfigurowalne.

e-mail: adabrow@agh.edu.pl

Prof. dr hab. inż. Kazimierz WIATR

Studia AGH Kraków (1980), dr nauk technicznych (1987), dr habilitowany (1999) i profesor (2002). Profesor zwyczajny na Akademii Górniczo-Hutniczej oraz Dyrektor Akademickiego Centrum Komputerowego Cyfronet AGH. Prowadzone prace badawcze dotyczą komputerowego sterowania procesami, systemów wizyjnych, systemów wieloprocesorowych, układów programowalnych, rekonfigurowalnych systemów obliczeniowych i sprzętowych metod akceleracji obliczeń.

e-mail: wiatr@agh.edu.pl

Streszczenie

W ramach praca przeprowadzona została analiza możliwości wykorzystania algorytmu winnowing do strumieniowego przetwarzania informacji tekstowej. W szczególności nacisk został położony na operacje generacji odcisku jako jej zredukowanej reprezentacji wiadomości tekstowej. Autorzy przeprowadzili szereg eksperymentów, w celu określenia efektywności działania algorytmu oraz możliwego do uzyskania przyspieszenia obliczeń, z wykorzystaniem węzła procesorów Intel Xeon E5645 2.40GHz oraz karty GPU Nvidia Tesla m2090.

Słowa kluczowe: n-gramowy model, eksploracja danych, przetwarzanie strumieniowe, GPGPU.

Parallel Winnowing Implementation for text stream analysis

Abstract

There are several models available for information retrieval and text analysis but the two are considered to be the dominant ones, namely Boolean and the vector space model (VSM). A model maps the existing words or text into a new representation space. This paper presents a boolean n-gram-based algorithm - winnowing for fast text search and comparison of documents with main focus on its implementation and performance analysis. The algorithm is used to generate fingerprints (i.e. a set of hashes) of the analyzed documents. A dedicated test framework was designed and implemented to handle the task of the algorithm evaluation which utilizes PAN test corpus and programming environment. Several tests were conducted in order to determine the comparison quality of the obfuscated and not obfuscated text for the winnowing algorithm and different window and n-gram size. The tests revealed interesting properties of the algorithms with respect to comparison of documents as well as defied the limits of their applicability. The n-gram-based algorithms due to their simplicity are well suited for hardware implementation. Thus, the authors implemented computationally demanding part of both fingerprint generation both on CPU and GPU. Performance measurements for Intel Xeon E5645, 2.40GHz and Nvidia Tesla m2090 implementation of Ngram-based algorithm show approximately 14x computational speedup.

Keywords: n-gram-based model, document comparison, GPU, information retrieval.

1. Wstęp

Obecnie generowana i przysyłana jest ogromna ilość danych w ogólnodostępnych sieciach teleinformatycznych. Szacuje się, że ilość wszystkich danych zgromadzonych w 2012 roku w zasobach Internetu sięgnęła 2,7 ZB, co stanowi 48 % wzrost w porównaniu z rokiem 2011, pod koniec roku 2013 liczba ta osiągnie 4 ZB [1] [2]. Jednocześnie wzrasta ilość danych przesyłanych, w 2010 roku było to 14 EB, w 2011 - 20 EB, w 2012 - 31 EB miesięcznie w sieciach niemobilnych [3]. Podobne tempo wzrostu ma miejsce w infrastrukturze mobilnej: rok 2010 - 256 PB, 2011 - 597 PB i 2012 - 885 PB miesięcznie [3][4]. Należy oczekiwać, że w nadchodzących latach będzie następować dalszy wzrost liczby urządzeń mobilnych oraz innych źródeł danych, co powodować będzie dalszy szybki wzrost liczby generowanych danych. Już dziś można zauważyć, że przechowywanie wszystkich danych (ang. raw data) wymaga ogromnej ilości przestrzeni dyskowej. Dodatkowo wraz z rozwojem infrastruktury sieciowej i wzrostem ilości przesyłanych danych, rośnie znaczenie szybkości pozyskania precyzyjnej informacji, oraz pojawia się coraz bardziej znacząca potrzeba szybkiej analizy treści. Przykładowo, dla firmy działającej w branży finansowej (np. Banku) informacja, która ma 3 godziny

posiada znacznie mniejsze znaczenie niż ta, która pojawiła się przed minutą.

W konsekwencji pojawia się potrzeba opracowania systemów, które będą w stanie dokonywać operacji ekstrakcji wiedzy z wielu szybko napływających strumieni danych. Systemy takie, aby skutecznie działać powinny być wyposażone w algorytmy pozwalające na modelowanie wybranego obszaru wiedzy w czasie rzeczywistym (dodawania nowych i usuwania starych nieaktualnych już struktur) oraz odpowiednią dedykowaną infrastrukturę sprzętową.

2. Strumieniowa analiza tekstu

Typowy system ekstrakcji informacji składa się z elementu realizującego wstępne przetwarzanie danych, elementu ekstrakcji wybranych cech oraz sekcji klasyfikatora lub zestawu klasyfikatorów, która generuje wynik. Do kluczowych zabiegów procesu projektowania systemów strumieniowej analizy treści należy zaliczyć wybór odpowiednich cech, które są ekstrahowane z danych w trakcie pracy systemu. Ten krok jest istotny, gdyż determinuje on bezpośrednio dwa najważniejsze aspekty działania systemu i algorytmu, jego skuteczność oraz szybkości.

Bardzo ważnym etapem jest również wstępne przetwarzanie tekstu, gdyż jego wynik rzutuje bezpośrednio na jakość informacji tekstowej przekazywaną do dalszych etapów przetwarzania. Do operacji stosowanych na tym etapie możemy zaliczyć np. lematyzację, stemming, stop-listę, start-listę lub word-net.

Kolejnym aspektem projektowania systemów ekstrakcji wiedzy jest dobór odpowiednich struktur danych, które pozwolą z jednej strony na wiarygodną reprezentację gromadzonych i przetwarzanych informacji, z drugiej strony będzie je można efektywnie przetwarzać oraz przechowywać w pamięci maszyny, na której zaimplementowany został algorytm. W obszarze przetwarzania informacji tekstowej dominują obecnie dwa modele reprezentacji informacji. Model typu boolowskiego (ang. boolean) oraz wektorowy. Zastosowanie modelu boolowskiego wiąże się z wykorzystaniem operacji n-gramowych, które mają mniejsze możliwości analityczne niż operacje realizowane w przestrzeni wektorowej.

Dlatego model oparty na przestrzeni wektorowej jest dominujący, nazywany jest on czasem również modelem worka słów (ang. „bag of words"). Model ten mapuje istniejące słowa lub też większe fragmenty tekstu do przestrzeni wektorowej i odwzorowuje każdy dokument, lub fragment tekstu jako wektor w tej przestrzeni. Porównywanie dokumentów sprowadza się do znalezienia zależności pomiędzy wektorami, do czego wykorzystywane są różne miary podobieństwa (np. iloczyn skalarny wektorów). Wymiarowość modelu wektorowego bardzo szybko rośnie wraz ze wzrostem ilości nowych dokumentów lub fragmentów tekstu wprowadzanych do modelu. W konsekwencji dla bardzo dużego modelu pojawia się zjawisko noszące miano „przekleństwa wymiarowości" (ang. curse of dimensionality), które przejawia się wzrostem wymiarowości oraz rzadkości przestrzeni w jakiej prowadzone są obliczenia. W konsekwencji, stosowany jest szereg metod redukcji wymiarowości danych (np. SVD, PCA).

Metody te są niestety złożone obliczeniowo, silnie sekwencyjne i składają się z operacji algebry macierzowych (np. mnożenie macierzy, obliczanie macierzy kowariancji oraz wartości i wektorów własnych). Dlatego autorzy pracy postanowili wykorzystać model boolowski do strumieniowej analizy tekstu. W pracy zbadane zostały granice możliwości jego zastosowania na przykładzie operacji porównywania dokumentów lub ciągów tekstu.

3. Algorytmy oparte na n-gramach

N-gram jest to ciąg liter o długości n, który najczęściej stanowi fragment większego ciągu znaków lub dokumentu tekstowego. Istnieją n-gramy oparte na wyrazach, które reprezentują ciąg następujących po sobie wyrazów, jak również takie, które przedstawiają ciąg liter. W pracy tej wykorzystywane będą tylko n-gramy bazujące na literach. Należy zauważyć, że ilość wszystkich n-gramów w dokumencie (wybranym fragmencie tekstu) jest prawie równa ilości liter i wyraża się następującą formułą [7]:

$$N = (n - k + 1), \tag{1}$$

gdzie: N- ilość wszystkich n-gramów generowanych z danego fragmentu tekstu, n - liczba wszystkich liter w analizowanym ciągu, k - rozmiar n-gramu.

Wykorzystanie wszystkich wygenerowanych n-gramów wymagałoby ogromnej ilości obliczeń. Analiza tekstu przedstawionego w ten sposób nie przyniosłaby zysku większego niż ta przeprowadzona bezpośrednio na niemodyfikowanym tekście. Dlatego w rzeczywistych aplikacjach wykorzystywana jest tyko część wszystkich n-gramów, co prowadzi to ważnego zagadnienia wyboru ich zestawu czyli tzw. odcisku. Odcisk dokumentu stanowi zbiór wybranych n-gramów, które najlepiej reprezentują dokument.

Ma to kluczowe znaczenie dla działania samego algorytmu i jego skuteczności, gdyż wybrane fragmenty stanowią reprezentację całego tekstu w procesie jego przetwarzania. Istnieje kilka popularnych rozwiązań w tym obszarze [5, 6] takich jak 0 mod p lub też wybór największej bądź najmniejszej wielkość z danego zakresu skrótów (ang. hash) n-gramów. Sztywne określenie ilości n-gramów niezależnej od wielkości dokumentu, pozwala na lepszą skalowalność całego modułu obliczeniowego, ale jednocześnie powoduje, że ciągi tekstu o różnych rozmiarach nie mogą być porównane wiarygodnie. Dlatego dobrze jest zapewnić równomierny wybór n-gramów w całej przestrzeni analizowanego tekstu, co realizuje algorytm winnowing [5].

4. Algorytm Winnowing

Autorzy algorytmu Winnowing [5] zaproponowali sposób wyboru odcisków tekstu, który gwarantuje wykrycie ciągu znaków o określonej długości, a zarazem zbiór odcisków jest o wiele mniej liczebny niż cały zbioru n-gramów (rys. 1). Okno wyboru odcisków W jest wyrażone następującą formułą:

$$W = (T - N + 1), \tag{2}$$

Mając zbiór skrótów n-gramów danego dokumentu $h_1,h_2,h_3 \ldots h_k$ dla każdego podzbioru o szerokości okna $h_i \ldots h_{i+W-1}$ gdzie $1 \le i \le k - W + 1$ wybieramy jedną wartość skrótu. Gwarantuje nam to znalezienie ciągów znaków o długości równej co najmniej T.

W każdym oknie wybieramy najmniejszą wartość i zapisujemy ją w zbiorze odcisku danego ciągu tekstowego. Jeżeli w oknie są dwie lub więcej takich samych wartości, to skrajnie prawa jest wybierana. Jest duża szansa, że w kolejnym oknie (po przesunięciu o 1 w prawo) najmniejsza wartość będzie tą samą wartością. Dlatego taki wybór gwarantuje stosunkowo niewielki rozmiar zbioru odcisku dokumentu.

Przykładowo mając następujący zbiór skrótów: [26 122 19 46 88 42 19 47 111 64 28 64 65 28 38 11 17 110 112], okno o wielkości 5 (W = 5) - dla każdego podzbioru wybierana jest jedna wartość według opisanej powyżej zasady i zapisywana jako odcisk dokumentu.

(26 122 **19** 46 88) (122 19 46 88 42) (19 46 88 42 **19**)
(46 88 42 19 47) (88 42 19 47 111) (42 19 47 111 64)
(19 47 111 64 28) (47 111 64 **28** 64) (111 64 28 64 65)
(64 28 64 65 **28**) (28 64 65 28 38) (64 65 28 38 **11**)
(65 28 38 11 17) (28 38 11 17 110) (38 11 17 110 112)

W konsekwencji otrzymujemy następujący odcisk dokumentu: [19 19 28 28 11].

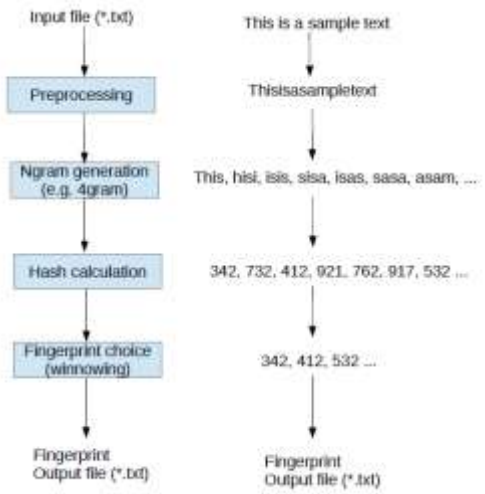
Kluczowym zagadnieniem przy porównywaniu dokumentów za pomocą algorytmów N –gramowych jest odpowiedni wybór wielkości n-gramu, wielkości okna – W (z czym wiąże się parametr T) oraz wartości progowej, która określa jak wiele wspólnych skrótów muszą mieć dwa dokumenty, aby sklasyfikować je jako podobne.

Przykładowa implementacja procedury winnowing w języku Python została przedstawiona poniżej:

```
FingerPrint = []
tmp = []

for i in range(0, (len(HashedNgrams) - WindowSize + 1)):
        b = HashedNgrams[i:(WindowSize+i)]
        if (min(b) != tmp) or (b[WindowSize - 1] == tmp):
                tmp = min(b)
                FingerPrint.append(tmp)
```

Schemat generacji odcisku fragmentu strumienia tekstu został przedstawiony na rys. 1.



Rys. 1. Przykładowy tor generacji odcisku dokumentu
Fig. 1. Sample n-gram generation scheme (4-gram)

W celu porównania dwóch dokumentów na podstawie ich odcisków wykorzystywana jest następująca zależność:

$$sim(str0, str1) = \frac{|H1 \cap H2|}{\max(|H1|, |H2|)}, \tag{3}$$

gdzie: $H1$ oraz $H2$ są odciskami dokumentów (strumieni tekstu) *str0* oraz *str1*.

Im więcej wspólnych n-gramów występuje w badanych fragmentach tekstu tym większa jest wartość współczynnika *sim*, dla identycznych dokumentów wartość ta wynosi 1.

5. Eksperymenty oraz dyskusja

Przeprowadzony został szereg eksperymentów w celu oceny efektywności działania algorytmu dla różnych rodzajów danych tekstowych na przykładzie badania wzajemnego podobieństwa dokumentów. Do badania skuteczności algorytmów wyszukujących podobieństwa między dokumentami potrzebna jest baza danych zawierająca dokumenty zarówno źródłowe, jak i te które zawierają w sobie fragmenty źródeł. Im większa baza danych tym bardziej wiarygodne wyniki testów możemy uzyskać.

Uniwesytet Bauhaus w Weimarze od kilku lat organizuje zawody międzyuczelniane, na których zespoły startują w kilku konkurencjach dotyczących NLP (Natural Language Processing), między innymi jednym z zadań jest detekcja duplikacji dokumentów. Na stronie internetowej tych zawodów jest zamieszczona obszerna baza danych zawierająca różnego rodzaju duplikacje wraz z opisującymi je dokumentami w formacie *.XML. Wyróżnione są w niej cztery typy duplikacji:

- Bez zaciemnienia (No Obfuscation) – dokładnie skopiowany fragment tekstu,
- Z zaciemnieniem losowym (Random Obfuscation) – niektóre wyrazy w skopiowanym fragmencie są zamienione kolejnością,
- Z zaciemnieniem tłumaczeniowym (Translation obfuscation) – za pomocą translatora fragmenty tekstu są tłumaczone na inny język, a następnie ponownie na język angielski. Powoduje to, że niektóre wyrazy zastąpione są bliskoznacznymi, zmieniana jest też składnia zdań,
- „Summary obfuscation" - plagiat będący streszczeniem źródłowego tekstu własnymi słowami.

W skład testowej bazy danych wchodzi :

- 1827 dokumentów podejrzanych o duplikację
- 3230 dokumentów źródłowych

Do porównania udostępnione są pary dokumentów podejrzany –źródłowy między którymi jest przypadek danego rodzaju duplikacji, oraz dokumenty XML-owe zawierające informacje o tym, który fragment tekstu został skopiowany.

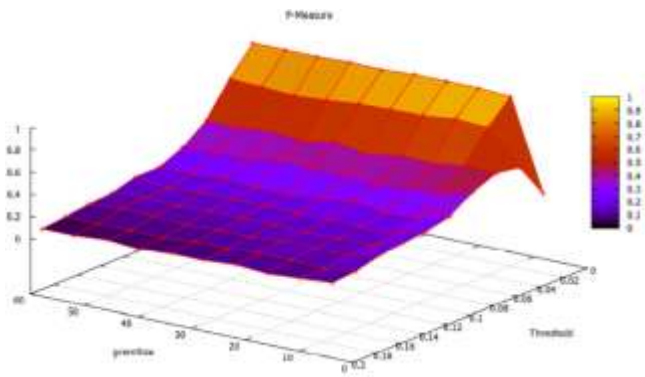
W pracy zastosowany został zestaw trzech podstawowych popularnych metryk do pomiaru skuteczności działania algorytmu [8]:

$$precision = \frac{relevant \cap retrieved}{retrieved}, \tag{4}$$

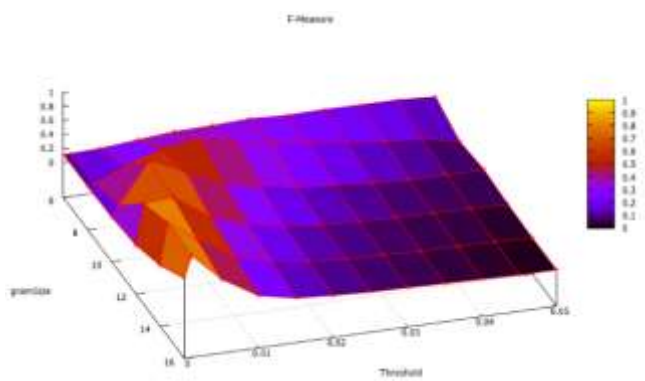
$$recall = \frac{relevant \cap retrieve}{relavant}, \tag{5}$$

$$F - measure = 2x \frac{precison \cap recall}{precision + recall}, \tag{6}$$

gdzie: *relevant* określa zbiór skrótów dokumentu, *retrieved* - zbiór zaklasyfikowanych przez moduł jako poprawne.



Rys. 2. Winnowing, brak zaciemnienia. Zależność *f-measure* od wielkości n-gramu i wysokości progu
Fig. 2. Winnowing, no obfuscation. *F-measure* as a function of n-gram and threshold



Rys. 3. Winnowing, zaciemnienie. Zależność *f-measure* od wielkości n-gramu i wysokości progu.
Fig. 3. Winnowing, obfuscation. *F-measure* as a function of n-gram and threshold

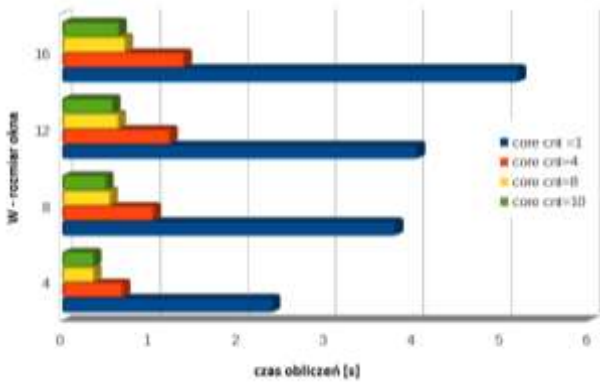
Rys. 2-3 prezentują zależności skuteczności wykrycia duplikacji, czyli efektywnego podobieństwa dokumentów w funkcji wielkości n-gramu oraz wysokości progu. Można zauważyć, obniżenie progu ma znaczący wpływ na wykrywalność podobieństwa dokumentów, przy jednoczesnym wzroście nieprawidłowej klasyfikacji ciągów tekstowych (spadek *precision*). W przypadku stosowania zaciemnienia znaczący wpływ na wynik porównania ma również wielkość n-gramu.

6. Przyspieszenie obliczeń

Operacja generacji odcisku realizowana jest dla wszystkich wiadomości pojawiających się w strumieniu danych. Czas jej realizacji ma duże znaczenie dla szybkiego przetwarzania informacji. Dlatego autorzy postanowili ją zaimplementować z wykorzystaniem węzła dwuprocesorowego Intel Xeon E5645 oraz GPU Nvidia Tesla m2090.

Tab. 1. Czas generacji odcisku dokumentu. Obliczenia na procesorze Intel Xeon E5645, 2.40GHz oraz karcie GPU Nvidia Tesla m2090 dla stałego rozmiaru okna i wielkości n-gram (N=W=4)
Tab. 1. Fingerprint generation time. Intel Xeon E5645, 2.40GHz and Nvidia Tesla m2090 were used for the computations for fixed window and n-gram size (W=N=4)

Rozmiar odcisku dokumentu [32bit skrót (ang.hash)]	Ilość bloków GPU	Czas obliczeń GPU, ms	Czas obliczeń CPU, ms	Przyspieszenie, x
512	4	0.078	0.041	0.526
701	8	0.084	0.132	1.571
2048	16	0.097	0.254	2.619
2771	32	0.116	0.496	4.276
5521	64	0.159	0.986	6.201
11157	128	0.253	2.037	8.051
22367	256	0.456	3.904	8.561
44753	512	0.811	7.82	9.642
174292	2000	2.842	30.798	10.837
699473	8000	9.948	147.87	14.864
1400736	16000	19.528	282.056	14.444
2676067	30000	36.16	472.107	13.056
5267371	60000	71	970.896	13.675



Rys. 4. Implementacja algorytmu winnowing dla N=4 z wykorzystaniem węzła dwuprocesorowego (12 rdzeni) Intel Xeon E5645, 2.40GHz
Fig. 4. Implementation of n-gram winnowing algorithm for N=4 (constant n-gram size) on dual processor node (12 cores) of Intel Xeon E5645, 2.40GHz

Uzyskane wyniki pokazują dużą skalowalność algorytmu generowania odcisku dokumentu (rys. 4), rozumianą jako proporcjonalny liniowy wzrost przyspieszenia obliczeń wraz ze wzrostem ilości rdzeni. Pełne wykorzystanie jednostki GPU (Tab. 1) pozwala uzyskać przyspieszenie obliczeń wynoszące około 14x. Należy podkreślić, że zwiększanie rozmiaru okna zwiększa proporcjonalnie nakład obliczeniowy (rys. 4). Pozwala to jednocześnie w znaczący sposób zmniejszyć rozmiar odcisku dokumentu, co z kolei wpływa na czas operacji porównania odcisków realizowanej w następnej fazie algorytmu.

7. Podsumowanie

W pracy dokonano analizy możliwości wykorzystania algorytmu winnowing to strumieniowego przetwarzania tekstu. W tym celu przetestowano jego efektywność opierając się na korpusie PAN [8]. Jest ona wysoka dla niskich wartości progów podobieństwa, co jednocześnie wpływa na spadek parametru *precision* i w konsekwencji na konieczność stosowania dwuetapowej weryfikacji [8].

Jednocześnie algorytm winnowing został zaimplementowany na węźle wielordzeniowym i GPU w celu weryfikacji możliwości zrównoleglenia operacji generacji odcisku dokumentu. Wyniki pokazały dużą skalowalność algorytmu zarówno na karcie graficznej jak i na procesorze wielordzeniowym. Maksymalne uzyskane przyspieszenie dla GPU wyniosło 14x.

W ramach dalszych prac zaimplementowana zostanie operacja szybkiego porównywania odcisków oraz stworzony zostanie kompletny tor strumieniowej analizy treści.

Projekt został sfinansowany ze środków Narodowego Centrum Nauki przyznanych na podstawie decyzji numer DEC-2011/01/B/ST6/03024.

8. Literatura

[1] IDC Predicts 2012 Will Be the Year of Mobile and Cloud Platform Wars as IT Vendors Vie for Leadership While the Industry RedefinesItself, http://www.businesswire.com/news/home/20111201005201/en/IDC-Predicts-2012-Year-Mobile-Cloud-Platform [access: 16.01.2014].
[2] Hilbert M.,López, P. :The Worlds Technological Capacity to Store. Science, Vol. 332, no. 6025, s. 60-65, 2011.
[3] Cisco Visual Networking Index: Forecast and Methodology, 2012 2017. Cisco Systems, White paper. [access: 16.01.2014].
[4] Amine A, Elberrichi Z., Simonet M., Malki, M.: WordNet-Based and N-Grams-Based Document Clustering. Proceedings of Third International Conference on Broadband Communications, Information Technology and Biomedical Applications, s.394-401, 2008.
[5] Cavnar W., B. Trenkle J.M.: N-Gram-based text categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, s. 161-175, 1994.
[6] Heintze, N.: Scalable Document Fingerprinting. Proceedings usenix workshop on electronic commerce, s. 191-200, 1996.
[7] Schleimer S., Wilkerson D.S., Aiken A.:Winnowing: local algorithms for document _ngerprinting. Proceeding of SIGMOD '03 Proceedings of the ACM SIG-MOD international conference on Management of data, s. 76-85, 2003.
[8] Potthast M., Stein B., Eiselt A., Barron-cedeno A., Rosso P.: Overview of the 1st International Competition on Plagiarism Detection. Benno Stein, Paolo Rosso, Efstathios Stamatatos, Moshe Koppel, and Eneko Agirre, editors, SE-PLN 09 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09), 2009.

otrzymano / received: 06.02.2014
przyjęto do druku / accepted: 01.04.2014 *artykuł recenzowany / revised paper*