STUDIA INFORMATICA 2009
Volume 30 Number 2B (84)

Artur WILCZEK, Karol WOŹNIAK
Politechnika Wrocławska, Instytut Informatyki

WYMAGANIA DLA STRUMIENIOWYCH SYSTEMÓW MONITORUJĄCYCH RUCH NA SERWERACH WWW

Streszczenie. Pojawienie się komercyjnych systemów strumieniowych baz danych ogólnego przeznaczenia otwiera możliwości badań nad ich zastosowaniami w różnorodnych dziedzinach. W niniejszej pracy omówiono wymagania dla systemu monitorującego serwery WWW w czasie rzeczywistym przy zastosowaniu strumieniowej bazy danych.

Słowa kluczowe: strumieniowe bazy danych, monitorowanie ruchu na serwerze WWW

REQUIREMENTS FOR STREAM SYSTEMS IN WWW SERVER TRAFFIC MONITORING

Summary. Emerging general purpose data stream management systems made it possible to conduct research focused on various applications. This chapter presents requirements for an application of data stream management systems in real-time web server traffic monitoring..

Keywords: data stream management systems, web server traffic monitoring

1. Wprowadzenie

Strumieniowe bazy danych były i są tematem badań licznych zespołów badawczych [1, 6]. Badania te otworzyły drogę do stworzenia systemów zarządzania bazami danych nowej klasy, operujących na strumieniach danych, czyli ciągach krotek, którym przypisane są znaczniki czasowe – systemów przetwarzania strumieni danych (ang. *Stream Processing Engine*) [1]. Próbę ogólnego scharakteryzowania tych systemów podjęto w pracy *The 8 Requirements of Real-Time Stream Processing* [11].

Jednym z efektów tych badań jest pojawienie się pierwszych komercyjnych systemów przetwarzania strumieni danych, takich jak StreamBase [12]. Systemy takie w założeniach mają umożliwić szybką implementację systemów monitorujących o dużej wydajności i krótkim czasie reakcji. Można oczekiwać, że w miarę wprowadzania różnorodnych cyfrowych technologii komunikacyjnych do masowego użytku wzrastać będzie zapotrzebowanie na aplikacje przetwarzające duże ilości danych [11].

Serwisy WWW stały się obecnie jednym z podstawowych mediów. Za ich pośrednictwem świadczy się usługi służące zarówno rozrywce, jak i pracy, a obsługiwany przez nie ruch wzrasta wykładniczo. Wiążą się z tym liczne i złożone wyzwania związane chociażby z zapewnieniem bezpieczeństwa i niezawodności usług. W związku z tym pojawia się zainteresowanie sposobami efektywnej analizy tego ruchu, a także automatycznego reagowania na pojawiające się zdarzenia.

2. Monitorowanie ruchu na serwerze WWW

Serwis internetowy (ang. *Website*) jest zbiorem stron internetowych (dokumentów) znajdujących się pod określonym adresem, połączonych odnośnikami (linkami) i powiązanych tematycznie. W epoce powszechnej informatyzacji serwisy internetowe coraz częściej odgrywają podstawową rolę dla firm i organizacji, które je utrzymują. Stale rośnie liczba podmiotów, dla których głównym źródłem dochodów jest świadczenie usług lub sprzedaż produktów za pośrednictwem Internetu. Analiza i przetwarzanie danych o ruchu w serwisie WWW ma dla nich podstawowe znaczenie. Pojawia się więc potrzeba opracowania metod efektywnego monitorowania[1] serwisów WWW.

Serwerem WWW nazywamy program komputerowy obsługujący protokół HTTP lub komputer, na którym taki program jest uruchomiony. Serwer taki udostępnia zasoby programom klienckim. Zasoby te są popularnie nazywane stronami internetowymi i najczęściej mają postać dokumentów w języku HTML i plików obrazów. Programami klienckimi najczęściej są przeglądarki internetowe. Duże serwisy internetowe z reguły posiadają własny serwer WWW działający na przeznaczonym do tego celu komputerze (węźle) lub grupie komputerów tworzących klaster.

[1] Monitorowanie – stała obserwacja i kontrola jakichś procesów lub zjawisk (Słownik Języka Polskiego, PWN, 2005)

2.1. Ruch w serwisie WWW

Protokół HTTP jest protokołem komunikacyjnym działającym na zasadzie żądanie – odpowiedź. Współczesne serwery WWW wykorzystują wersję 1.1 tego protokołu, która opisana jest w dokumencie RFC 2616 [7]. Zgodne ze specyfikacją, żądanie zawiera, między innymi: metodę (akcję), którą serwer ma wykonać, identyfikator zasobu, którego ta akcja dotyczy oraz, opcjonalnie, nagłówki (ang. *headers*), zawierające dodatkowe informacje, na przykład o typie przeglądarki i systemu operacyjnego. Szczególnie istotne są dane z nagłówka *Referer*. Jeśli użytkownik trafił na daną stronę za pomocą odnośnika nagłówek ten będzie zawierał adres (URI), na której znajdował się odnośnik. Kolejnym istotnym z punktu widzenia tej pracy nagłówkiem HTTP jest nagłówek *User agent*. Zawiera on informacje o systemie operacyjnym użytkownika oraz używanej przez niego przeglądarce.

Odpowiedź serwera HTTP rozpoczyna się od trzycyfrowego kodu sygnalizującego poprawne wykonanie żądania lub błąd, tak zwanego kodu stanu (ang. *Status code*). Następnie przekazywane są nagłówki odpowiedzi i jej ciało - właściwa zawartość.

Całość danych przesyłanych pomiędzy serwerem WWW a klientami określa się jako *ruch w serwisie www* (ang. *Web traffic*). Duże serwisy internetowe muszą często obsługiwać dziesiątki tysięcy żądań w ciągu sekundy. Przykładowo, amerykański serwis MySpace obsługuje około miliarda trafień (żądań pojedynczych dokumentów) w ciągu doby, to znaczy średnio 10 tysięcy w ciągu sekundy. Należy jednak zwrócić uwagę, że aktywność użytkowników nie jest rozłożona równomiernie w czasie i o pewnych porach ruch jest znaczenie większy od wartości średniej. Takie ilości napływających informacji w pełni uzasadniają wykorzystanie strumieniowych baz danych w zadaniach związanych z monitorowaniem ruchu.

Serwery WWW obsługujące duże serwisy często współpracują z innymi systemami, takimi jak serwery baz danych, serwery aplikacji oraz serwery buforujące (PROXY), stanowią więc tylko fragment infrastruktury obsługującej serwis. Monitorowanie ruchu w serwisie opiera się jednak na monitorowaniu aktywności serwerów WWW obsługujących ten serwis, gdyż właśnie one zarządzają całością przetwarzania żądań HTTP.

2.2. Korzyści płynące z monitorowania ruchu w serwisie WWW

Wdrożenie odpowiedniego systemu monitorującego serwis WWW może przynieść instytucji utrzymującej go różnorakie korzyści zarówno na płaszczyźnie biznesowej, jak i technicznej. Przede wszystkim można tą drogą uzyskać dane o działaniach użytkowników. Dane te są przydatne w działaniach marketingowych oraz w ocenie, jak dobrze dana strona spełnia swoje zadania biznesowe. Dzięki nim można analizować na przykład, jak użytkownicy przemieszczają się po stronach serwisu, jak szybko odnajdują potrzebne informacje, kiedy podejmują decyzję o zakupie produktu i tak dalej. Dzięki zawartym w nagłówkach HTTP informa-

cjom można określić, skąd dany użytkownik trafił na stronę, a przez to ocenić skuteczność reklamy internetowej prowadzonego serwisu oraz sprawdzić, czy można do niego łatwo trafić za pomocą wyszukiwarek internetowych. W warunkach ostrej konkurencji panującej w sieci prowadzenie takich badań jest bardzo istotne.

Na płaszczyźnie technicznej wprowadzenie systemu monitorowania ułatwia utrzymanie bezpieczeństwa, niezawodności i wydajności serwisu. Dzięki prowadzeniu skrupulatnego monitorowania możliwe jest wykrywanie nieprawidłowości w działaniu serwisu i likwidowanie ich. System monitorowania może ułatwiać wykrywanie błędów w oprogramowaniu serwisu i diagnozowanie ich przyczyn. Może też informować o pojawiających się problemach z wydajnością i ułatwiać ich usuwanie.

3. Istniejące rozwiązania w zakresie monitorowania ruchu

Monitorowanie serwerów WWW, podobnie zresztą jak wszelkich innych usług sieciowych, można prowadzić na dwa sposoby. Monitorowanie aktywne (lub syntetyczne) charakteryzuje się tym, że system monitorujący samodzielnie generuje żądania i sprawdza, jak serwer na nie odpowiada. Z natury rzeczy, liczba takich żądań jest zazwyczaj dość ograniczona, aby uniknąć nadmiernego obciążenia systemu monitorowanego. Monitorowanie pasywne polega natomiast na rejestrowaniu całości ruchu i jego analizie. W tej pracy skoncentrowano się na systemach monitorowania pasywnego, czyli rejestrujących całość ruchu na serwerze WWW – systemy aktywne nie wymagają zazwyczaj przetwarzania danych z taką szybkością, aby uzasadnione było zastosowanie strumieniowych baz danych.

3.1. Analiza dziennika serwera WWW

Większość powszechnie używanych serwerów WWW zapisuje dane o odwiedzinach użytkowników w dziennikach (logach), najczęściej mających postać plików płaskich. Dzienniki są w wysokim stopniu konfigurowalne i zawierają dane o wszystkich żądaniach obsłużonych przez serwer, a także różnego rodzaju informacje diagnostyczne [2]. Większość systemów monitorowania pasywnego serwerów WWW opiera się na analizie dzienników. Istnieje wiele programów przeznaczonych do tego celu, począwszy od prostych skryptów bazujących na wyrażeniach regularnych, takich jak AWStats [3], a skończywszy na kompleksowych rozwiązaniach, wykorzystujących zaawansowane techniki Data Mining jak SAS Web Analytics [10]. Podstawowe informacje dostarczane przez najprostsze z tych programów to:

- informacje ilościowe o ruchu na serwerze w czasie: liczbę żądań w jednostce czasu, ilość wysłanych i odebranych bajtów itp.;

- informacje o użytkownikach: numer IP, system operacyjny, typ i wersja przeglądarki internetowej. Dodatkowo, dzięki odpowiednim usługom sieciowym można w przybliżeniu ustalić kraj i region, z którego nastąpiło połączenie;
- podstawowe informacje o wizytach użytkowników – ile czasu spędzają w serwisie, z jakiej strony do niego trafili i tym podobne;
- podsumowanie wszystkich błędów, jakie wystąpiły podczas przetwarzania żądań.

Tego typu informacje są najczęściej wykorzystywane przez administratorów zajmujących się technicznym utrzymaniem strony. Są też przydatne programistom tworzącym aplikacje internetowe w procesie wykrywania błędów w tych aplikacjach.

Zaawansowane narzędzia analizujące ruch koncentrują się na rozszerzonej analizie działań użytkowników na stronie WWW. Analizy takie mogą przykładowo obejmować: korelacje między promocjami a zakupami, klasteryzację (ang. *clustering*) odwiedzających, stwierdzenie, które z podstron serwisu najlepiej zachęcają klientów do dokonania zakupu i tym podobne [10]. Wyniki tych analiz mają podstawowe znaczenie dla osób zajmujących się zagadnieniami marketingu, sprzedaży oraz specjalistów od reklamy, a także kadry zarządzającej zainteresowanej efektywnością biznesową serwisu internetowego.

3.2. Integracja systemu monitorującego z serwerem WWW

Systemy monitorujące bazujące na analizie dzienników praktycznie nie nadają się do zastosowania w sytuacjach, w których trzeba szybko reagować na zachodzące zdarzenia. Przetwarzanie plików dzienników odbywa się niezależnie od serwera WWW, często na innej maszynie fizycznej. Dzięki temu nie wpływają one na wydajność monitorowanego serwera, ale też nie nadają się do zastosowania w sytuacji, kiedy od systemu monitorującego wymaga się szybkiej, autonomicznej reakcji na zachodzące zdarzenia. Ponadto, nie wszystkie informacje na temat żądania mogą zostać zapisane w logach. W związku z tym w niektórych sytuacjach zachodzi potrzeba integracji systemu monitorującego z serwerem WWW.

Przykładem takiego zintegrowanego systemu jest ModSecurity – moduł (rozszerzenie) serwera Apache pełniący funkcję ściany ogniowej poziomu aplikacji (ang. *Web Application Firewall*) [8]. Oznacza to, że moduł ten monitoruje całość żądań obsługiwanych przez serwer i poszukuje w nich cech charakterystycznych dla różnych zagrożeń, jak ataki przeciwko aplikacjom działającym na serwerze, aktywność programów automatycznie skanujących strony internetowe (ang. *Web crawlers*) czy próby połączenia z programami koni trojańskich. Poszukiwane wzorce są zapisywane w plikach zasad (ang. *rule files*). Pliki te zawierają też specyfikację akcji, które moduł ma podjąć w przypadku napotkania określonego wzorca. Akcje mogą obejmować między innymi: odrzucenie bądź przekierowanie żądania, uruchomienie ze-

wnętrznej aplikacji i przekazanie jej danych na temat żądania, zapisanie/odstąpienie od zapisania informacji o zdarzeniu w dzienniku i tym podobne [8].

Dzięki temu, że ModSecurity działa jako moduł w obrębie serwera WWW Apache jest on łatwy w instalacji i ma dostęp do wszystkich danych na temat żądania posiadanych przez serwer. Umożliwia też bieżące reagowanie na zdarzenia. Jednakże, taka architektura systemu monitorującego wiąże się z istotnymi ograniczeniami, jeśli chodzi o dostępny czas procesora i ilość pamięci operacyjnej, a przede wszystkim o wykonywanie operacji dyskowych. Starsze wersje omawianego modułu w ogóle nie posiadały możliwości trwałego składowania danych. Dopiero w jego najnowszej wersji (2.1) wprowadzono możliwość przechowywania danych o sesjach i użytkownikach w postaci tak zwanych kolekcji (ang. *collections*). Jednakże, możliwości korzystania z pamięci trwałej są ograniczone względami wydajnościowymi – duża liczba operacji dyskowych albo wykonywanie bardziej skomplikowanych obliczeń szybko spowodowałoby znaczne obciążenie serwera WWW i przez to duże opóźnienie obsługi żądań.

Wadą ograniczającą zastosowanie modułu ModSecurity jest fakt, że język używany do definiowania reguł jest nader skomplikowany i nie przypomina innych, powszechnie znanych języków programowania. Wiele zadań wykonać można wyłącznie stosując różnego typu sztuczki programistyczne, właściwe wyłącznie dla tego języka. W związku z tym tworzenie i wprowadzanie zmiany w regułach jest skomplikowane i drogie. Implementacja bardziej złożonego systemu monitorującego z użyciem tego modułu byłaby więc nieopłacalna.

4. System strumieniowy w monitorowaniu ruchu na serwerze WWW

Systemy zarządzania strumieniami danych wydają się być obiecującym narzędziem do monitorowania serwerów WWW. Charakteryzują się one zarówno niskim czasem reakcji, jak i zdolnością do przetwarzania znacznych ilości danych, przez co pozbawione są ograniczeń tradycyjnych systemów monitorujących. Wydaje się, że system monitorujący oparty na strumieniowej bazie danych może spełniać jednocześnie funkcję analizatora dzienników serwera i wbudowanego systemu monitorującego aktywnie reagującego na zdarzenia, łącząc zalety obu tych rozwiązań.

4.1. Korzyści wynikające z zastosowania systemów strumieniowych

Wspomniano wcześniej, że zintegrowane systemy monitorujące, które mają w założeniu wpływać na działanie monitorowanie serwera WWW, mogą powodować istotne dodatkowe obciążenie tego serwera. W związku z tym w niniejszej pracy proponuje się przeniesienie

kosztownych obliczeń na przeznaczony do tego celu oddzielny serwer strumieniowych baz danych. Jednocześnie, na serwerze WWW należy zainstalować komponent umożliwiający łączenie się z systemem strumieniowym i przekazywanie do niego danych na temat ruchu. Oczywiste jest, że aby ograniczyć straty wydajności spowodowane zastosowaniem systemu monitorującego, komponent przekazujący dane powinien być zoptymalizowany pod kątem maksymalnej wydajności.

Dodatkową zaletą wynikającą z zaprojektowania systemu w opisany wyżej sposób jest to, że dzięki temu system łatwo będzie można przystosować do działania, w sytuacji gdy serwis działa na kilku serwerach WWW. W takim przypadku serwery te mogą niezależnie przekazywać dane na temat ruchu do centralnego systemu monitorującego, nawet jeśli znajdują się w różnych lokalizacjach fizycznych.

Istotną z punktu widzenia ekonomicznego i organizacyjnego cechą takich systemów jest to, że strumieniowe bazy danych są oparte na powszechnie znanych standardach tradycyjnych baz danych, takich jak język SQL. Dzięki temu programista znający SQL będzie mógł zostać szybko przeszkolony do tworzenia aplikacji strumieniowych, co powinno znacznie ograniczyć koszty budowy i utrzymania aplikacji monitorujących. Jest to szczególnie istotne w świecie dynamiczne rozwijających się usług internetowych, w którym wymagania stawiane aplikacjom mogą się często zmieniać.

4.2. Wymagania dla systemu monitorującego

Aby zbadać możliwość zastosowania systemu strumieniowego do monitorowania serwisu WWW dokonano próby implementacji prostego systemu monitorującego. System służy do monitorowania ruchu na serwerze Apache. W implementacji wykorzystano komercyjny system strumieniowy StreamBase firmy StreamBase Inc. Na potrzeby tworzonego rozwiązania wybrano zestaw funkcji, które system taki powinien implementować. W wyborze tych funkcji uwzględniono także funkcje oferowane w istniejących systemach. Należą do nich:

- Rozkład obciążenia systemu w czasie – system powinien monitorować obciążenie serwera w czasie. Obciążenie serwera będzie wyrażane w następujących jednostkach: liczba żądań w ciągu minuty oraz liczba wysłanych bajtów w ciągu minuty.
- Wykrywanie błędów w przetwarzaniu żądań – system powinien monitorować kody odpowiedzi HTTP i wyszukiwać w nich oznak błędów. Kody błędów protokołu HTTP mają postać 4XX i 5XX, gdzie X oznacza dowolną cyfrę. Z punktu widzenia programisty aplikacji najistotniejsze jest tu wykrywanie błędów o kodzie 503, które często powodowane są przez oprogramowanie.
- Wykrywanie „martwych linków” – w przypadku dużych i dynamicznie rozbudowywanych serwisów, takich jak portale informacyjne zdarza się, że odnośniki wewnątrz danego

serwisu są wadliwie skonstruowane i wskazują na nieistniejącą zawartość. Oznacza to, że kiedy użytkownik kliknie na taki odnośnik, w odpowiedzi otrzyma komunikat HTTP z kodem błędu 404 – nie znaleziono. Takie sytuacje można w prosty sposób wykryć, monitorując nagłówek Referer żądań, które zakończyły się zwróceniem tego kodu błędu.

- Monitorowanie adresów IP zachowujących się podejrzanie – system monitorujący może zbierać dane o adresach IP generujących żądania, które są klasyfikowane jako podejrzane. Zestaw podejrzanych działań określić można na podstawie analizy wybranych reguł programu ModSecurity [8].
- Identyfikacja sesji – system może dokonywać na bieżąco identyfikacji sesji użytkowników. Zapisane w sposób trwały wyniki tej identyfikacji mogą być następnie przekazane do dalszego przetwarzania za pomocą wybranych narzędzi do analizy działań użytkowników. Metody identyfikacji sesji użytkownika można ogólnie podzielić na oparte na czasie (ang. time based) i oparte na kontekście (ang. context based) [9]. Metody oparte na czasie koncentrują się na analizie czasu, jaki upływa pomiędzy żądaniami kolejnych dokumentów. Metody oparte na kontekście wyodrębniają sesje na podstawie pewnych logicznych ciągów działań, które użytkownik wykonuje w jakimś celu. Takim ciągiem działań może być wypełnienie kilkuetapowego formularza w celu rejestracji na stronie lub dodanie towaru do koszyka, a następnie jego zakup.

Jak widać, zdecydowano się na połączenie funkcji zazwyczaj realizowanych przez odrębne systemy monitorujące. Miało to na celu przede wszystkim pokazanie elastyczności systemów strumieniowych. Ponadto, różnorodność działań wykonywanych przez aplikację umożliwia zademonstrowanie różnych możliwości systemu strumieniowego

5. Podsumowanie

Systemy zarządzania strumieniami danych stosowane np. w monitorowaniu sieci telekomunikacyjnych wydają się być obiecującym narzędziem także do monitorowania serwerów WWW. Niski czas reakcji i zdolność do przetwarzania znacznych ilości danych, charakterystyczna dla systemów strumieniowych, eliminuje wiele ograniczeń tradycyjnych systemów monitorujących. System monitorujący oparty na strumieniowej bazie danych może spełniać jednocześnie funkcję analizatora dzienników serwera i wbudowanego systemu monitorującego aktywnie reagującego na zdarzenia, łącząc zalety obu tych rozwiązań.

W pracy przedstawiono propozycję wymagań, które powinien spełniać system monitorujący ruch na serwerze WWW, stanowiący rozwiązanie pośrednie pomiędzy systemem zintegrowanym a analizatorem logów. Ograniczono się do wskazania najistotniejszych wymagań funkcjonalnych, które po zaimplementowaniu w prototypie systemu monitorującego pozwolą

ocenić zalety i ewentualne wady stosowanych systemów strumieniowych w monitorowaniu ruchu na serwerach WWW.

Na podstawie przedstawionych w pracy wymagań zbudowano prototyp systemu monitorującego z wykorzystaniem komercyjnego systemu strumieniowego StreamBase firmy StreamBase Inc. monitorujący ruch na serwerze WWW Apache. Zagadnienia dotyczące implementacji prototypu strumieniowego systemu monitorującego spełniającego wymagania przedstawione w niniejszej pracy są przedmiotem odrębnej publikacji.

BIBLIOGRAFIA

1. Arasu A., Babcock B., Babu S., Datar M., Ito K, Nizhizawa I., Rosenstein J., Widom J.: STREAM: The Stanford Stream Data Manager. In ACM SIGMOD Conference, June 2003.
2. Apache, Apache http Server Documentation, http://httpd.apache.org/docs/2.2/logs.html
3. AWstat, AWStats logfile analyzer 6.9 Documentation, http://awstats.sourceforge.net/docs/index.html.
4. Lavoie B., Frystyk Nielsen H.: Web Characterization Terminology & Definitions Sheet – W3C Working Draft. World Wide Web Consortium, 1999, http://www.w3.org/1999/05/WCA-terms/.
5. Chen Z., Wai-Chee Fu A., Chi-Hung Tong F.: Optimal Algorithms for Finding User Access Sessions from Very Large Web Logs. World Wide Web: Internet and Web Information Systems 6, s. 259÷279, Kluwer Academic Publishers, Holandia, 2003.
6. Carney D., Cetintemel U., Cherniack M., Convey C., Lee S., Seidman G., Stonebraker M., Tatbul N., Zdonik S.: Monitoring Streams: A New Class of Data Management Applications. In proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02), Hong Kong, China, 2002.
7. Fielding R., Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., Berners-Lee T.: Hypertext Transfer Protocol – HTTP/1.1, The Internet Society, 1999.
8. Ristic I.: Apache Security, O'Reilly, 2005.
9. Pierrakos D., Paliouras G., Papatheodorou C., Spyropoulos C.: Web Usage Mining as a Tool for Personalization: A Survey, W: User Modeling and User-Adapted Interaction 13, s. 311÷372, Kluwer Academic Publishers, Holandia, 2003.
10. SAS, Web Analytics, http://www.sas.com/solutions/webanalytics/index.html.
11. Stonebraker M., Çetintemel U., Zdonik S.: The 8 Requrements of Real-Time Stream Processing. ACM SIGMOD Record 34, 4 (Dec 2005), s. 42÷47.

12. StreamBase Inc., StreamBase Documentation, http://streambase.com/developers/docs/latest/

Recenzent: Dr hab inż. Andrzej Chydziński

Wpłynęło do Redakcji 1 lutego 2009 r.

Abstract

This paper presents requirements for web traffic monitoring system located in the middle between log analyzers and fully integrated systems. Specifying requirements stream managements systems were taken into consideration. Monitoring system implementing those requirements was successfully implemented using StreamBase stream management system on Apache www server.

Adresy

Artur WILCZEK: Politechnika Wrocławska, Instytut Informatyki,
ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Polska, artur.wilczek@pwr.wroc.pl.
Karol WOŹNIAK: Politechnika Wrocławska, Instytut Informatyki,
ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Polska, karol.w.wozniak@gmail.com