

Leszek BORZEMSKI, Łukasz CICHOCKI, Ziemowit NOWAK

Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów

Wing – SYSTEM DO POMIARU WYDAJNOŚCI USŁUGI WWW PO STRONIE KLIENTA

Streszczenie. Powodem spadania wydajności usługi WWW jest niewątpliwie jej ogromna i wciąż rosnąca popularność. Aby odnaleźć źródła problemów, należy poddać analizie zjawiska zachodzące podczas transakcji webowych. W pracy przedstawiono autorski system Wing, którego głównym zadaniem jest pomiar charakterystyk czasowych ruchu HTTP obserwowanego po stronie użytkownika. Analiza zarejestrowanych danych pozwala na diagnozowanie problemów wydajnościowych usługi WWW w sieci Internet.

Słowa kluczowe: WWW, ocena wydajności, pomiary w sieci.

Wing – A SYSTEM FOR USER-PERCEIVED WEB PERFORMANCE MEASUREMENTS

Summary. It is needed to make possible to measure user-perceived web performance. In this paper the design of the system for such purposes called Wing is presented. Using this system we are able to analyse how web transactions are performed and what are the main time delays in web access time.

Keywords: WWW, performance evaluation, network measurements.

1. Wprowadzenie. Specyfika usługi WWW

Internet swoją niezwykłą popularność zawdzięcza usłudze World Wide Web (WWW). Jej twórca, Tim Berners-Lee, zakładał w 1989 roku, że ma ona służyć użytkownikom do pobierania informacji udostępnionych na komputerach połączonych siecią. Współcześnie uważa się, że WWW to sieciowa usługa łącząca użytkowników i serwisy rozproszone na komputerach całego świata [4]. Sukces usługi WWW pociąga, niestety, za sobą degradację wydajno-

ści sieci Internet. Aby dotrzeć do źródeł problemów wydajnościowych, należy prześledzić zjawiska zachodzące podczas transakcji webowych.

Usługa WWW opiera się na modelu klient/serwer, który zakłada, że klient¹ jest zawsze inicjatorem transmisji, tzn. wysyła do serwera wiadomość zawierającą żądanie, natomiast serwer zwraca wiadomość zawierającą odpowiedź. Wielkość (długość) żądania i odpowiedzi liczona jest w bajtach. Zwykle żądanie jest znacznie krótsze niż odpowiedź, gdyż ta ostatnia zawiera żądany przez klienta zasób serwera².

Do przekazywania żądań i odpowiedzi w usłudze WWW wykorzystywany jest protokół HTTP (HyperText Transfer Protocol). Ponieważ HTTP jest protokołem bezstanowym, kolejne pary żądań i odpowiedzi są od siebie niezależne. Obecnie obowiązującym standardem jest wersja HTTP/1.1, jednak wciąż używana jest poprzednia wersja HTTP/1.0. Zasadniczą różnicą obu wersji protokołów jest to, iż pierwszy z nich dla każdej pary żądanie-odpowiedź pomiędzy klientem a serwerem domyślnie zestawia osobne połączenie na poziomie protokołu TCP/IP (Transfer Control Protocol/Internet Protocol). Natomiast HTTP/1.1 podtrzymuje przez pewien czas połączenie TCP/IP tak, aby kolejne pary żądanie-odpowiedź występowały w zestawionym już połączeniu (persistent connection). Protokół HTTP/1.1 umożliwia ponadto klientowi wysyłanie w jednej wiadomości kilku żądań (pipelining). Istotne jest jednak, aby serwer zwrócił odpowiedzi w tej samej kolejności, w jakiej zostały wysłane żądania. Doświadczenia autorów wskazują na małą popularność pipeliningu. Można to tłumaczyć między innymi tym, iż konsekwencją błędu w potoku żądań lub odpowiedzi jest konieczność powtórzenia całej procedury.

Jak wykazały eksperymenty – m.in. [1] – wymienione własności protokołu HTTP/1.1 przyczyniają się do efektywniejszej obsługi pobierania stron WWW. Jest to istotne osiągnięcie, gdyż jak wykazują badania [8], zdecydowana większość użytkowników ma najwięcej zastrzeżeń do wydajności działania usługi WWW (do długiego czasu oczekiwania na informację). Z tego też powodu złośliwi tłumaczą skrót WWW jako World Wide Wait.

Czas, jaki upływa od wysłania przez klienta żądania do otrzymania przez niego kompletnej odpowiedzi od serwera, zależy przede wszystkim od następujących czynników:

- wielkości żądania,
- szybkości transmisji na łączu w kierunku od klienta do serwera,
- czasu przetwarzania żądania (generowania odpowiedzi) przez serwer,
- wielkości odpowiedzi,
- szybkości transmisji na łączu w kierunku od serwera do klienta.

¹ Klient, serwer – komputer z zainstalowanym oprogramowaniem klienta, serwera.

² Nie ma jednak przeszkód, żeby proporcje były odwrotne. Klient może w żądaniu wysłać swój zasób na serwer, a ten w odpowiedzi wyśle jedynie potwierdzenie przyjęcia.

Analiza przebiegu transakcji WWW³ wykazuje jeszcze jeden istotny czynnik wydłużający czas oczekiwania użytkownika. Jest nim czas zamiany zawartej w URL nazwy domeno-
wej komputera na jego adres IP.

Należy sobie zdawać sprawę, że w przypadku usługi WWW, wyspecyfikowanie przez użytkownika interesującego go zasobu⁴ pociąga najczęściej za sobą serię żądań klienta. Wynika to z właściwości języka składu dokumentów – HTML (HyperText Meta Language). Język ten zakłada możliwość osadzania w jednym zasobie innych zasobów – zwykle obiektów multimedialnych. Gdy przeglądarka WWW⁵ stwierdza, że otrzymała od serwera zasób zapisany w języku HTML, dokonuje jego analizy i w przypadku wykrycia odpowiednich poleceń, automatycznie żąda osadzonych zasobów. Należy podkreślić, iż obiekty osadzone w dokumencie HTML mogą pochodzić z różnych serwerów WWW.

Aby klient mógł zestawić połączenie z serwerem na poziomie protokołu TCP/IP, musi znać następujące dane:

- własny adres IP,
- własny numer portu TCP,
- adres IP serwera,
- numer portu TCP serwera.

Zarówno własne dane, jak i numer portu serwera⁶ są dla klienta oczywiste. Aby uzyskać adres IP serwera WWW, klient wysyła żądanie do serwera DNS (Domain Name System), którego adres IP ustalany jest podczas konfiguracji środowiska sieciowego klienta. Zwykle jest to tzw. lokalny serwer DNS, który jeśli nie zna odpowiedzi, przekazuje żądanie do innych, odpowiednich serwerów DNS.

Aby zminimalizować ruch generowany w sieci przez system DNS, zarówno klient WWW, jak i serwer DNS przez określony czas TTL (Time To Live) przechowują odwzorowania nazw w adresy IP. Powoduje to, iż czas oczekiwania klienta WWW na odpowiedź systemu DNS jest zmienny. Warto odnotować, że żądania i odpowiedzi DNS przekazywane są bezpołączeniowym protokołem UDP (User Datagram Protocol) działającym na bazie IP.

³ Termin transakcja będzie stosowany do określenia serii żądań klienta i odpowiedzi serwera związanych z uzyskaniem przez klienta kompletnego dokumentu WWW.

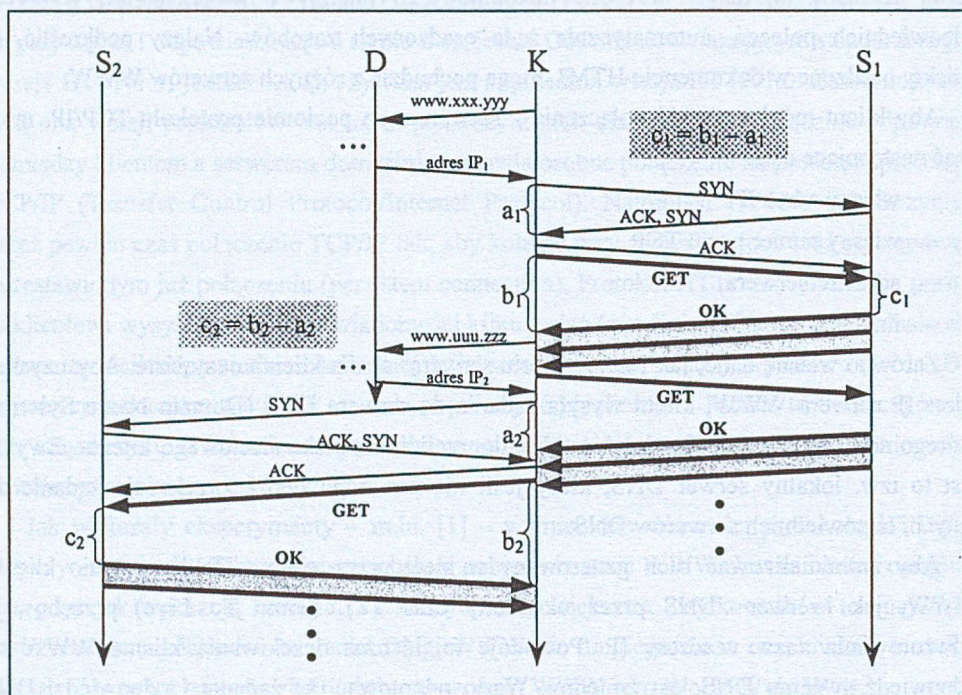
⁴ Poprzez podanie URL (Universal Resource Locator).

⁵ Najczęściej wykorzystywany klient WWW.

⁶ W przypadku usługi WWW domyślnie jest to port o numerze 80. Numer portu może być jawnie wyspecyfikowany w URL.

2. Przebieg transakcji WWW

Rys. 1 przedstawia przebieg przykładowej transakcji WWW. Przeglądarka pobiera dokument HTML, w którym osadzone są dwa obiekty: jeden na tym samym serwerze, drugi na innym. Osie czasu dotyczą kolejno: drugiego serwera WWW (S_2), lokalnego serwera DNS (D), przeglądarki – klienta WWW (K) oraz pierwszego serwera WWW (S_1). Czas „biegnie” z góry na dół. Wektory oznaczają transmisję poszczególnych pakietów poprzez sieć. Pionowa składowa wektora jest czasem transmisji, który zależy od efektywnej przepustowości sieci.



Rys.1. Schemat przebiegu transakcji webowej

Fig. 1. Time diagram of web transaction

Po podaniu (kliknięciu) przez użytkownika URL-a odpowiadającego dokumentowi HTML K żąda od D adresu IP S_1 . Kropki na wysokości odcinka czasu pomiędzy nadejściem żądania a wysłaniem odpowiedzi oznaczają, iż D przekazywał żądanie dalej. Po otrzymaniu przez K adresu IP S_1 , K wysyła do S_1 pakiet z ustawioną flagą `SYN` w celu zestawienia połączenia w kierunku od K do S_1 . W odpowiedzi S_1 wysyła pakiet z ustawionymi flagami: `ACK` – potwierdzającą zestawienie połączenia w jednym kierunku oraz `SYN` – żądającą zestawienia połączenia w przeciwnym kierunku. Następnie K wysyła do S_1 pakiet `ACK` potwierdzając w ten sposób zestawienie połączenia w przeciwnym kierunku. Kolejny pakiet zawiera pierwsze żądanie HTTP. Jak widać, przygotowanie odpowiedzi jest czasochłonne dla S_1 . Jej prze-

kazanie wymaga użycia trzech pakietów. Natychmiast po otrzymaniu pierwszego pakietu, K rozpoczyna analizę dokumentu HTML. W jej wyniku okazuje się, iż należy zażądać obiektu, który znajduje się na S_2 . K rozpoczyna wykonywanie takich samych czynności, jakie były wykonane względem S_1 . Po nadejściu od S_1 trzeciego pakietu, dalsza analiza dokumentu HTML przez K wykazuje konieczność zażądania obiektu, który znajduje się na S_1 . Ponieważ K do komunikacji z S_1 i S_2 używa protokołu HTTP/1.1, żądanie jest wykonywane na zestawionym już połączeniu z S_1 . Należy dodać, iż po upływie określonego czasu od przeprowadzenia ostatniej transmisji następuje zamknięcie połączeń TCP/IP pomiędzy klientem i serwerami.

Rys. 1 jest schematem poglądowym. W rzeczywistych warunkach trudne jest, a zarazem niecelowe, jednoczesne odtwarzanie interwałów czasowych po stronie klienta i serwera. Do analizy problemów, z jakimi spotykają się użytkownicy usługi WWW, wystarczająca jest znajomość współrzędnych punktów na osi czasowej po stronie klienta. Poza tym chcąc, np. wyznaczyć przybliżone opóźnienie wprowadzane przez serwer, można posłużyć się zależnościami przedstawionymi na rys. 1.

Pomiary czasów mogą być prowadzone w formie eksperymentu aktywnego lub pasywnego [4, 5]. W eksperymencie pasywnym wykorzystuje się ruch wytwarzany przez użytkowników usługi WWW [2, 10]. Zaletą takiego rozwiązania jest brak konieczności generowania dodatkowego ruchu w sieci, wadą – skomplikowany sposób pozyskiwania danych oraz brak możliwości powtórzenia pomiaru w określonym momencie. Eksperyment aktywny wymaga umyślnego generowania żądań [6, 3]. Zaletą jego jest łatwość pozyskiwania danych pomiarowych oraz powtarzalność pomiaru, wadą – wytwarzanie dodatkowego ruchu w sieci.

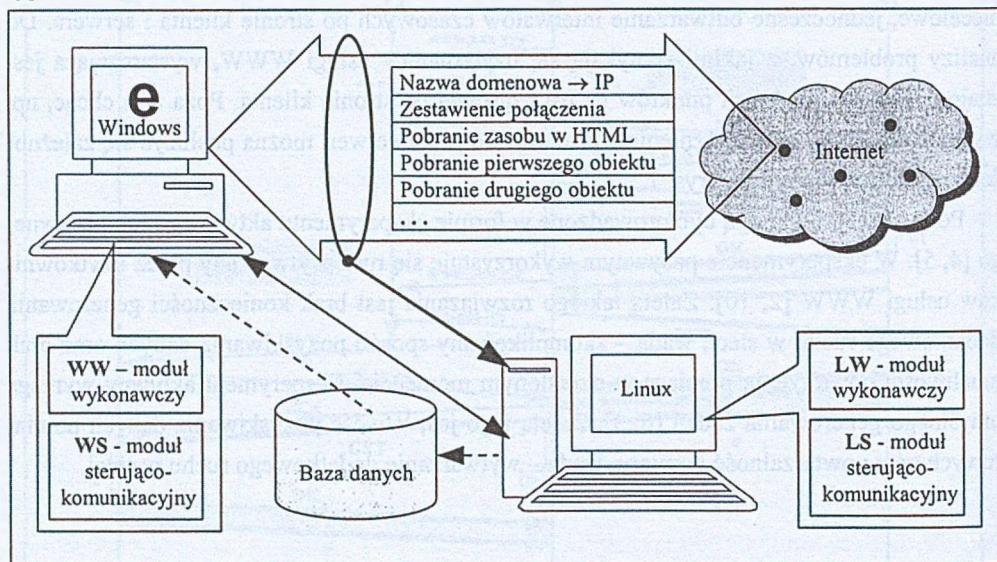
3. System pomiarowy Wing

W Zakładzie Rozproszonych Systemów Komputerowych Instytutu Sterowania i Techniki Systemów na Wydziale Informatyki i Zarządzania Politechniki Wrocławskiej skonstruowano system pomiarowy, który wykorzystując eksperyment aktywny umożliwia analizę interwałów czasowych usługi WWW obserwowanych po stronie klienta. Nazwa systemu – **Wing** – jest kalamburem słów *web* i *ping*. Znaczenie pierwszego jest oczywiste, natomiast *ping* jest nazwą usługi stosowanej szeroko w Internecie do aktywnego pomiaru opóźnienia wprowadzanego przez sieć.

Projekt zakładał jak najwierniejsze odtworzenie sytuacji doświadczanych przez maksymalnie dużą grupę użytkowników przeglądarek WWW. Jak wykazują statystyki [9], 96% użytkowników przeglądarek posługuje się aplikacją MS Internet Explorer (MS IE) pracującą pod kontrolą systemu operacyjnego z rodziny Windows. Według najlepszej wiedzy

autorów, dotychczasowe projekty o podobnym charakterze, zarówno komercyjne [6], jak i darmowe [7, 11], nie analizują transakcji przeprowadzanych za pomocą protokołu HTTP/1.1, a w szczególności zachowań przeglądarki MS IE.

Zasada działania systemu została pokazana na rys. 2. System **Wing** wymaga zaangażowania dwóch komputerów klasy PC. Wstępne eksperymenty na pojedynczej maszynie wykazały występowanie zakłóceń w pracy przeglądarki podczas jednoczesnego rejestrowania danych pomiarowych. Poza tym system Windows, niezbędny jako platforma działania przeglądarki MS IE, uniemożliwiał wykorzystanie sprawdzonych i dobrze udokumentowanych narzędzi do pozyskiwania danych działających jedynie w środowisku systemów operacyjnych typu Unix.



Rys.2. Zasada działania systemu Wing

Fig. 2. Wing functioning

Zarówno *windowsowa*, jak i *linuxowa* część systemu **Wing** składa się z dwóch modułów: sterująco-komunikacyjnego i wykonawczego⁷. Komunikacja pomiędzy modułami WS i LS zorganizowana została na bazie protokołu TCP/IP i wykorzystuje stworzony na potrzeby systemu zbiór komend. Inicjacja systemu następuje zawsze z modułu WS poprzez wysłanie w kierunku LS komendy INIT⁸. Komenda INIT przekazuje również LS adres serwera BD, jeśli dane pomiarowe należy gromadzić w bazie danych. Od tego momentu **Wing** znajduje się w stanie gotowości do prowadzenia pomiarów.

⁷ Stąd odpowiednie nazwy modułów: WS, WW, LS i LW.

⁸ Moduł LS jest swego rodzaju serwerem sterującym.

Aktywacja badania następuje poprzez wyspecyfikowanie w module WS identyfikatora zasobu (URL) oraz parametrów definiujących tryb pracy modułu WW. Następnie *windowso*-wy moduł sterujący za pomocą mechanizmu OLE (Object Linking Embedded) wywołuje moduł WW oraz wysyła do modułu LS komendę START. Odebranie przez *linuxowy* moduł sterujący komendy START powoduje uruchomienie modułu LW odpowiedzialnego w tej części badania za rejestrację danych. Aby uniknąć zniekształceń czasów pobierania obiektów, WW usuwa pliki pamięci podręcznej przeglądarki IE, po czym uruchamia ją przekazując wyspecyfikowany w module WS URL. Przeglądarka wykonuje te same czynności, które wykonywałaby w przypadku wybrania URL-a przez użytkownika. Proces pobierania kompletnego dokumentu WWW rejestrowany jest przez moduł LW.

Linuxowy moduł wykonawczy zbudowany został na bazie kodu aplikacji znanej pod nazwą TCPdump. Odpowiednia jej funkcja przełącza kartę sieciową maszyny *linuxowej* w tzw. tryb promiscuous, pozwalający przechwytywać wszystkie pakiety pojawiające się w medium transmisyjnym, do którego jest podłączona. Tak więc, aby było możliwe przeprowadzenie badania, pakiety przepływające pomiędzy maszyną *windowso*wą a serwerem WWW muszą docierać również do karty sieciowej maszyny *linuxowej*. **Wing** wykorzystuje również funkcję TCPdump-a umożliwiającą filtrowanie pakietów według zadanego kryterium. Zapobiega to gromadzeniu się w buforze LW zbędnych pakietów, gdy w nasłuchiwanym segmencie sieci pracują jeszcze inne komputery.

Kiedy WW wykryje, iż przeglądarka zakończyła pobieranie kompletnego dokumentu, w zależności od wyspecyfikowanych w WS parametrów, ulega samodestrukcji lub przechodzi w stan uśpienia. Reakcją na powyższe zachowanie modułu WW jest wysłanie przez WS do *linuxowego* modułu sterującego komendy STOP. Powoduje to przełączenie modułu LW w tryb analizy treści pakietów zebranych w buforze.

Jest to najbardziej skomplikowana czynność w procesie pomiarowym. Należy bowiem zrekonstruować cały przebieg transakcji WWW, realizowanej zwykle za pomocą kilku połączeń równolegle. Wykorzystywane są w tym celu adresy IP, porty TCP i numery sekwencyjne pakietów. Podczas analizy należy uwzględnić fakt, że w wyniku opóźnień i retransmisji na skutek błędów, pakiety mogą być ułożone w innej kolejności niż zostały wysłane oraz zdublowane. Aby identyfikować obiekty osadzone w dokumentach webowych, należy sięgać do „wnętrz” pakietów w celu identyfikacji nagłówków wiadomości przesyłanych na poziomie protokołu HTTP. W przypadku zasobów zapisanych w języku HTML należy dokonać analizy ich kodu w celu ustalenia, które z obiektów są w nich osadzone. Niebanalnym problemem jest śledzenie automatycznych przekierowań wykonywanych przez przeglądarkę na poziomie protokołu HTTP⁹ oraz rekonstrukcja dokumentów HTML składających się z tzw. ramek.

⁹ Często informacja o przekierowaniu znajduje się w kodzie HTML.

Nierzadko zdarza się, że treść kodu wysyłana przez serwer jest uprzednio kompresowana. W takich przypadkach, po zestawieniu odpowiednich pakietów, treść wiadomości należy zdekompresować.

W wyniku analizy zgromadzonych danych powstaje drzewiasta struktura obiektów, zawierająca takie dane, jak:

- czasy nadejścia każdego z pakietów należącego do obiektu,
- liczby poprawnych i niepoprawnych (wymagających retransmisji) pakietów,
- rozmiary każdego obiektu ustalone poprzez zliczenie bajtów przynależących do ciał obiektów,
- rozmiary obiektów pobrane z nagłówek HTTP,
- kolejności wystąpień podczas pobierania z uwzględnieniem równoległego pobierania obiektów na wielu portach przez przeglądarkę.

Dla każdego z obiektów moduł LW generuje szczegółowe ciągi wartości. Są to m.in.:

- *DateTime* – data i czas przeprowadzonego badania;
- *Measurement* – całkowity czas pobierania obiektu;
- *DNS* – czas potrzebny na „rozwiniecie” nazwy komputera do postaci adresu IP;
- *Connect* – czas potrzebny do ustanowienia połączenia z serwerem WWW na bazie protokołu TCP/IP;
- *First_Byte* – czas, jaki upłynął od wysłania żądania HTTP do momentu nadejścia pierwszego pakietu z odpowiedzią;
- *IndexFile* – czas potrzebny na pobranie obiektu zakodowanego w HTML;
- *Content* – sumaryczny czas potrzebny na załadowanie osadzonych obiektów;
- *Start_MS* – czas, jaki upłynął od rozpoczęcia badania do momentu wysłania pakietu związanego z danym obiektem;
- *URL* – identyfikator URL obiektu;
- *DNS2SYN* – czas, jaki upłynął pomiędzy otrzymaniem przez klienta ostatniego pakietu związanego z obsługą DNS a wysłaniem pakietu inicjującego połączenie TCP/IP;
- *ACK2GET* – czas, jaki upłynął od momentu otrzymania przez klienta pakietu potwierdzającego zestawienie połączenia do momentu wysłania pakietu z żądaniem HTTP danego obiektu.

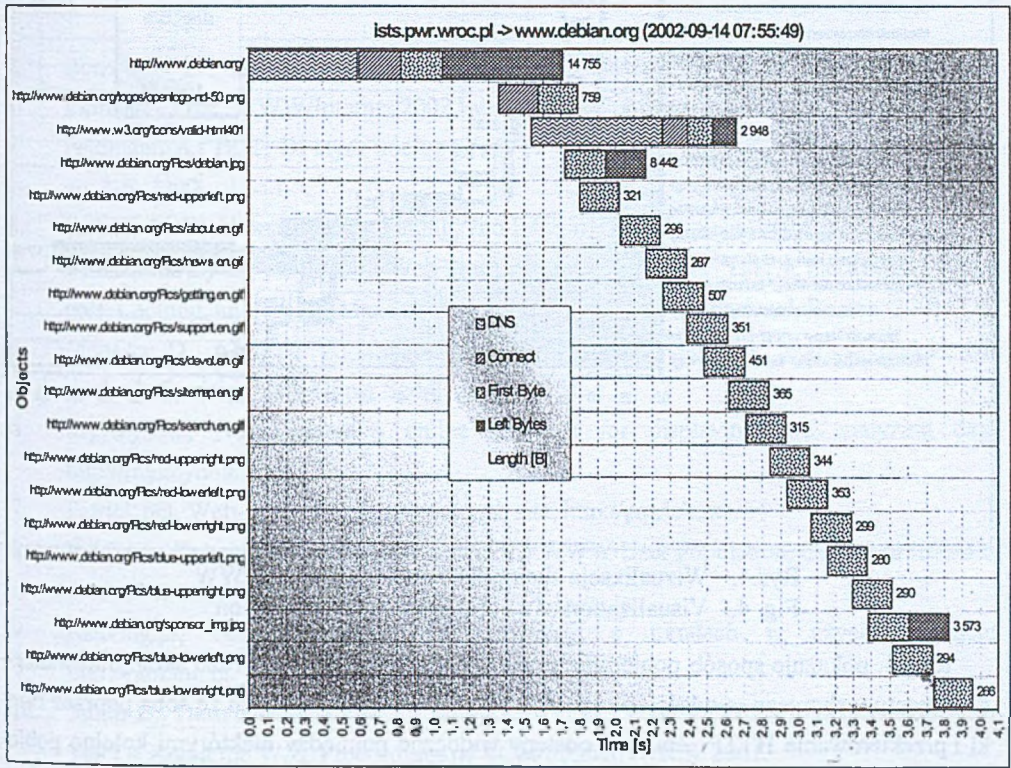
W zależności od parametru przekazanego komendą INIT, powyższe ciągi wartości są przesyłane w formie rekordów do bazy danych lub jedynie na standardowe wyjście maszyny *linuxowej*. Za zapis struktury do bazy odpowiedzialny jest moduł LS, który otrzymuje dane od modułu LW za pośrednictwem potoku.

Autorzy zwracają uwagę na fakt, iż gromadzenie wyników pomiarów w bazie danych stwarza możliwość dogłębnej ich analizy porównawczej w późniejszym czasie. Dotyczy to

zarówno miejsc, z których były pobierane zasoby, jak i momentów czasowych ich pobierania. Aby umożliwić cykliczne badanie wydajności wielu miejsc w Internecie, moduł WS został dodatkowo wyposażony w programator. Funkcja ta umożliwia „ręczne” zadanie listy URL-i do przebadania lub zaimportowanie ich z bazy danych. Możliwe jest również ustalenie częstotliwości powtarzania badań.

4. Przykłady pomiarów

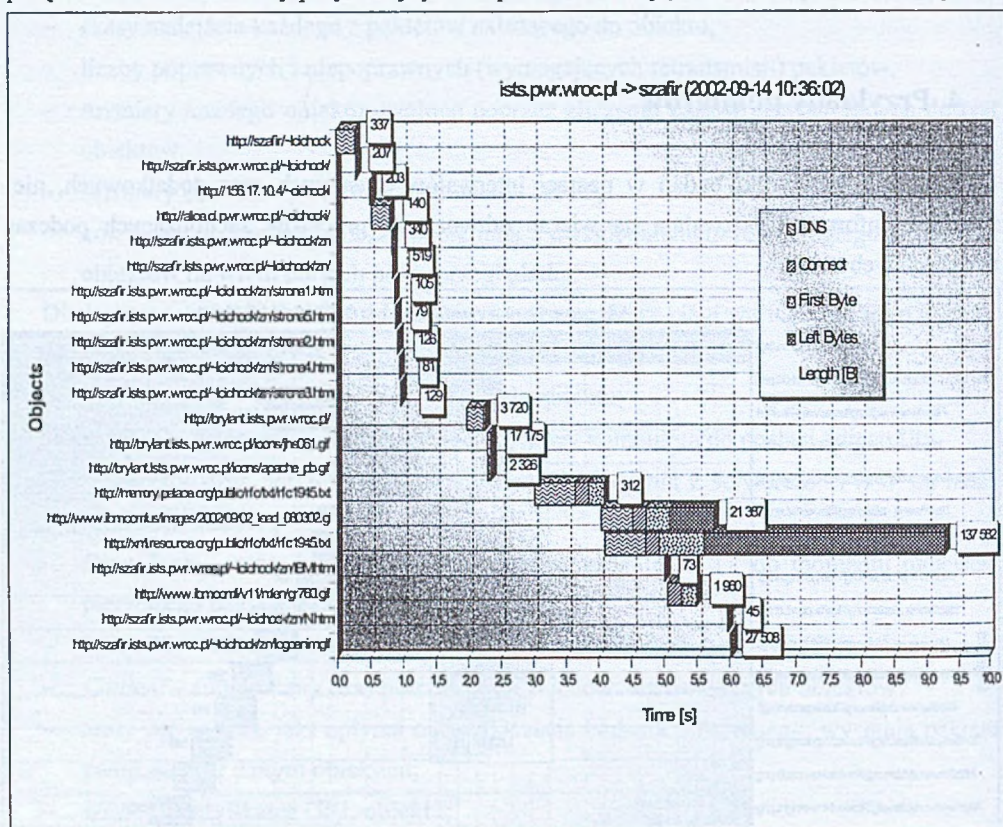
Szczegółowe wyniki badań w postaci interwałów czasowych oraz dodatkowych, niezbędnych informacji pozwalają na wiernie odtworzenie procesów zachodzących podczas transakcji webowej.



Rys.3. Wizualizacja prostej transakcji WWW
Fig. 3. Visualization of a simple WWW transaction

Rys. 3 obrazuje przebieg pobierania głównej strony serwisu www.debian.org przez przeglądarkę MS IE. Jak widać, zajęło to przeglądarce 4 sekundy. Interwał *Left Bytes* – czas pomiędzy nadejściem pierwszego i ostatniego pakietu związanego z danym obiektem – został wyznaczony poprzez przekształcenia odpowiednich innych interwałów. Jego rzadkie wystę-

powanie świadczy o tym, iż obiekty mieściły się zwykle w pojedynczych pakietach. Pojawienie się tylko w górnej części wykresu interwału *Connect* wskazuje, iż do transmisji wykorzystywany był protokół HTTP/1.1. Ze sposobu pobierania kolejnych obiektów z serwera www.debian.org można wywnioskować, że przeglądarka wykorzystywała jednocześnie dwa połączenia TCP/IP. Liczby przy kolejnych słupkach oznaczają wielkości obiektów w bajtach.



Rys.4. Wizualizacja skomplikowanej transakcji WWW

Fig. 4. Visualization of a complex WWW transaction

Rys. 4. pokazuje sposób pobierania przez przeglądarkę MS IE bardzo skomplikowanego dokumentu webowego, a właściwie szeregu dokumentów powiązanych ze sobą poprzez ramki i przekierowania HTTP. Znaczne odstępy widoczne pomiędzy niektórymi kolejno pobieranymi obiektami wynikają z opóźnień zadanych w kodzie HTML. Ze względu na duże zróżnicowanie czasów, do wizualizacji transakcji zastosowano trójwymiarowe słupki.

5. Uwagi końcowe

W pracy omówiono procesy zachodzące podczas wszechobecnych w Internecie transakcji webowych oraz wskazano źródła opóźnień mających wpływ na wydajność usługi WWW. Autorzy zaprezentowali zbudowany przez nich system **Wing** służący do pomiaru charakterystyk czasowych transakcji webowych. Kierunkiem dalszych prac będzie analiza danych pomiarowych zgromadzonych w bazie. Ze względu na znaczną ich ilość, oprócz standardowych metod statystycznych obiecujące wydaje się zastosowanie metod określanych angielskim terminem *data mining*.

LITERATURA

1. Borzemski L.: The impact of cluster-based web systems design on user-perceived performance. Proc. WWW/Internet 2002 International Conference, Lisbon, Nov. 2002.
2. Feldmann A.: BLT: Bi-layer tracing of HTTP and TCP/IP, Computer Networks, vol. 33, no. 1-6, 2000.
3. Killelea P.: Web Performance Tuning, 2nd Edition, O'Reilly, Sebastopol 2002.
4. Krishnamurthy B., Rexford J.: Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement, Addison Wesley Professional, Boston. 2001.
5. Menascé D., Almeida V.: Capacity Planning for Web Services: metrics, models, and methods, Prentice Hall, Upper Saddle River, 2001.
6. MyKeynote, Next-generation online interface for displaying and analyzing data, <http://mykeynote.com>.
7. Patrick.net, Web-based set of tools for web site, <http://patrick.net>.
8. Pitkow J., Kehoe C.: Emerging Trends in the WWW User Population, Communications of the ACM, vol. 39, no. 6, 1996.
9. Ranking.pl, Serwis dostarczający informacji o trendach w użyciu Internetu, <http://ranking.pl>.
10. Smith D., Hernandez-Campos F., Jeffay K., Ott D.: What TCP/IP Protocol Headers Can Tell Us About the Web, Proceedings of ACM SIGMETRICS, 2001.
11. WebPerf.org, Web performace monitoring, <http://webperf.org>.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 7 kwietnia 2003 r.

Abstract

This paper presents the design of the **Wing** system developed for the performance measurements of web transactions as they are seen by the users. The system monitors all actions performed by the web browser and collects the information concerning times events occurred during an individual web page transaction. The information collected can be visualized and processed in further analysis, e.g. using data mining tools.

Adresy

Leszek BORZEMSKI, Łukasz CICHOCKI, Ziemowit NOWAK: Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Polska, leszek@ists.pwr.wroc.pl, lcichocki@ists.pwr.wroc.pl, znowak@ists.pwr.wroc.pl.