ZESZYTY NAUKOWE UNIWERSYTETU SZCZECIŃSKIEGO

NR 476 STUDIA INFORMATICA NR 21 2007

JAKUB SWACHA
ARTUR KULPA
ROMAN BUDZOWSKI

USPRAWNIENIE INTERNETOWEGO SYSTEMU MONITORUJĄCEGO PRZEZ ZASTOSOWANIE KOMPRESJI DANYCH

Wstęp

W erze informacji ogólnie przyjętą zasadą postępowania jest powiększanie wiedzy, gdy tylko pojawia się sposobność ku temu. Zasady tej szczególnie łatwo przestrzegać w przypadku Internetu, gdzie koszt zbierania informacji jest niewielki. Jedną z podstawowych technik zbierania informacji w Internecie jest monitorowanie zachowań użytkowników w zautomatyzowany sposób, a jego rezultaty mogą dostarczyć cennych informacji zarówno o samym monitorowanym serwisie, jak i jego użytkowniku.

Wiele danych użytecznych z punktu widzenia handlu elektronicznego można pozyskać już za pomocą prostych systemów monitorujących jedynie przejścia między stronami serwisu, takich jak stat.pl[1]. Znacznie bogatszą w detale informację o zachowaniu użytkownika i sposobie wykorzystywania przez niego interfejsu można było dotychczas uzyskać dzięki specjalnemu oprogramowaniu działającym po stronie klienta[2]. *Novum* w tej dziedzinie jest charakteryzujący się nieinwazyjnością i lekkością autorski system działający w środowisku skryptowym przeglądarki, w którym program monitorujący jest dołączony do zawartości

[1] www.stat.pl

[2] K. Fenstermacher, M. Ginsburg: *Client-Side Monitoring for Web Mining.* „Journal of the American Society of Information Science and Technology" 2003, No 54(7), s. 625–637.

dokumentów pobieranych z serwera[3]. System ten, szerzej omówiony w artykule, pozwala bez uciążliwości dla użytkownika zbierać bardzo szczegółowe dane, dokumentujące wykonywane przez niego czynności, które następnie są przesyłane na serwer w celu analizy i dalszego wykorzystania. Naturalną konsekwencją objętość przesyłanych danych jest stosunkowo duża. Ma to niewielkie znaczenie dla użytkowników korzystających z szybkich łącz szerokopasmowych, lecz użytkownik korzystający z powolnego łącza, wykorzystującego na przykład linię telefoniczną, zauważy wyraźny spadek efektywnej szybkości połączenia. Niedogodność tę można usunąć przez kompresję przeznaczonych do przesłania danych. Sposób, w jaki wykorzystano ją w tym systemie, omówiono w dalszej części artykułu, a praktyczne konsekwencje implementacji kompresji przedstawiono na przykładzie wyników uzyskanych dla jednego z monitorowanych serwisów internetowych.

1. Internetowy system monitorujący – zasada działania i problemy praktyczne

System monitorowania służy do rejestrowania aktywności użytkownika podczas interakcji z dokumentami WWW. Podstawowym założeniem działania systemu jest umożliwienie śledzenia działalności użytkownika systemu, w odróżnieniu od standardowych systemów monitorowania po stronie przeglądarki internetowej. Dzięki takiemu podejściu możliwe jest zebranie bardzo dokładnych danych na temat współpracy na linii interfejs użytkownika-użytkownik. Klasyczne systemy monitorujące w systemach WWW działają po stronie serwera, co umożliwia jedynie nadzorowanie, jakich dokumentów WWW zażądał klient, i rejestrowanie informacji z nagłówków żądań.

System zapewnia niezależność od platformy i komfort użytkownikowi, gdyż nie ma potrzeby instalowania jakiegokolwiek dodatkowego oprogramowania u klienta systemu WWW. Konieczne jest jedynie używanie przeglądarki internetowej obsługującej język JavaScript[4] i mechanizmu cookie.

Jak wspomniano, system monitorujący jest oparty na możliwościach języka JavaScirpt, obsługiwanego przez przeglądarkę internetową. Dodatkowym wy-

[3] A. Kulpa, J. Swacha, R. Budzowski: *Script-Based System for Monitoring Client-Side Activity.* „Lecture Notes in Informatics, 85, Proc. Business Information Systems" 2006, s. 573–582.

[4] D. Goodman, M. Morrison: *JavaScript Bible*. Wiley Publishing, Indianapolis 2004.

maganiem wobec przeglądarki, które musi zostać spełnione, jest implementacja interfejsu modelu DOM[5] dla dokumentów HTML.

Instalacja oprogramowania działającego po stronie przeglądarki internetowej polega na włączeniu do dokumentu HTML pliku zawierającego kod systemu monitorującego, przez umieszczenie w nim następującego elementu:

<script type=”text/javascript” language=”JavaScript” src=”eventlog.js”></script>.

Może się to odbyć jednorazowo, za pomocą skryptu, który umieści ten element we wszystkich dokumentach HTML na serwerze systemu monitorowanego, lub „w locie”, przy każdej okazji wysyłania przez serwer WWW pliku HTML do przeglądarki internetowej.

System działa na zasadzie wychwytywania określonych typów zdarzeń inicjowanych przez użytkownika. Aby było możliwe ich śledzenie, niezbędne jest podłączenie procedur monitorowania do wszystkich tych elementów dokumentu HTML, które mogą być przedmiotem zdarzenia. Analizowanie całej struktury dokumentu HTML i dołączanie odpowiedniej obsługi do wszystkich elementów dokumentu może znacznie spowolnić działanie przeglądarki internetowej. Obciążenie to byłoby proporcjonalne do liczby elementów w dokumencie, dlatego podczas ładowania dokumentu do przeglądarki internetowej obsługa zdarzeń zostaje włączona tylko do elementu <BODY>. Dzięki mechanizmowi „bubbles”, informacja o zdarzeniach, które podlegają tej procedurze, jest przekazywana do wszystkich elementów nadrzędnych. Ponieważ <BODY> jest elementem nadrzędnym dla wszystkich innych wyświetlanych w przeglądarce, informacja o zdarzeniach trafia ostatecznie do niego wraz z informacją o elemencie, który faktycznie spowodował zdarzenie.

Większość czynności dokonywanych na stronie internetowej powoduje stworzenie w przeglądarce internetowej obiektu zdarzenia, który zawiera wiele informacji na jego temat. Dostęp do jego własności jest możliwy za pomocą skryptu JavaScript za pośrednictwem modelu DOM. Skrypt zbiera istotne informacje o zdarzeniach i gromadzi je w buforze systemu monitorującego. Po wypełnieniu buforu informacje o zdarzeniach są wysyłane do części serwerowej

[5] J. Robie, S. Byrne, P. le Hégaret, A. le Hors, L. Wood, G. Nicol, M. Champion: *Document Object Model (DOM) Level 3 Core Specification*. April 2004, http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-200404077.

systemu monitorującego za pomocą mechanizmu AJAX[6], gdzie trafiają do bazy danych. Podczas badań zauważono, że wysyłanie informacji o zdarzeniu do serwera za każdym razem, gdy ono zajdzie, nadmiernie obciąża przeglądarkę internetową, dlatego dane o zdarzeniach są grupowane w pakiety (po 20 zdarzeń każdy) i dopiero w takiej postaci wysyłane do serwera systemu monitorującego. Po stronie serwera zdarzenia i paczki są zapisywane do specjalnie zaprojektowanej bazy danych za pomocą procedur napisanych w języku PHP5[7]. Do utrzymywania bazy danych wykorzystano system PostgreSQL[8].

Chcąc uzyskać sekwencję zdarzeń w systemie monitorowanym, dla konkretnego użytkownika zaimplementowano mechanizm sesji w systemie nadzorczym. W momencie, gdy żądany dokument zostanie załadowany do przeglądarki internetowej, zakładane są dwa obiekty *cookie*. Każdy z nich zawiera unikatowy numer sesji. Pierwsze *cookie* jest tymczasowe, ma termin ważności do momentu przejścia do innej strony lub zamknięcia przeglądarki internetowej, w zależności od tego, co nastąpi wcześniej. Sesja tymczasowa pozwala uzyskać sekwencję zdarzeń wywołanych przez użytkownika podczas jego wizyty na stronie. Drugie *cookie* ma 30-dniowy termin ważności. Pozwala na identyfikację użytkowników, którzy powrócili w tym okresie do określonej aplikacji WWW.

Akcje użytkownika implikują powstanie różnych typów obiektu zdarzenia. System monitorujący może, w zależności od potrzeby, monitorować 25 rodzajów zdarzeń, które można podzielić na następujące kategorie:

a) zdarzenia związane z elementem (*onclick*, *onfocusin*, *onfocusout*, *onselectstart*, *ondblclick, oncontextmenu*);
b) zdarzenia związane z aktywnością urządzenia wskaźnikowego (*onmouseover*, *onmouseout*, *onmousedown*, *onmousewheel, onmouseup*);
c) zdarzenia mechanizmu „przesuń i upuść" (*ondrag*, *ondragenter*, *ondrop*, *ondragend*, *ondragleave, ondragstart*);
d) zdarzenia związane z klawiaturą komputerową (*onkeydown*, *onkeyup, onkeypress*);
e) zdarzenia związane ze schowkiem systemowym (*oncopy*, *onpaste, oncut*);
f) zdarzenia wejścia/wyjścia do/z dokumentu (*onload*, *onunload*).

[6] J.J. Garrett: *Ajax. A New Approach to Web Applications*. *Adaptive Path*. February 2005, http://www.adaptivepath.com/publications/essays/archives/000385.php.

[7] J. Coggeshall: *PHP Unleashed*. Sams Publishing, Indianapolis 2004.

[8] www.postgresql.org.

2. Możliwość redukcji obciążenia sieci poprzez użycie kompresji danych

Spotykane w praktyce dane, w tym także pochodzące z internetowego systemu monitorującego, zawierają wiele informacji nieistotnych lub powielonych. Informacja nieistotna obejmuje wszystkie te składniki przekazu, które mogą być z niego usunięte bez straty dla jego znaczenia z punktu widzenia odbiorcy. Informacja powielona to te składniki przekazu, które występują w nim wielokrotnie. Odjęcie obu rodzajów niepotrzebnych składników może radykalnie zmniejszyć ilość przesłanych danych.

Kompresja danych polega na przekształceniu reprezentacji danych tak, aby długość ich zapisu uległa skróceniu[9]. Kompresja daje szczególnie dobre rezultaty wówczas, gdy projektuje się ją dla ściśle określonego zastosowania, jakim w tym przypadku są dane zebrane przez internetowy system monitorujący. Zastosowanie takie wymaga algorytmu kompresji, który będzie działał szybko bez wielu zasobów systemowych. Wymaganie to spełniają algorytmy substytucyjne, których istotą jest zamiana długich, powtarzalnych sekwencji danych na krótki identyfikator elementu słownika, który jednoznacznie identyfikuje tę sekwencję[10]. Jeżeli zawartość słownika jest z góry znana (słownik jest statyczny), nie ma potrzeby dołączania go do skompresowanych danych. Jest tak w przypadku wielu danych zbieranych przez internetowy system monitorujący, takich jak nazwy elementów HTML, zdarzeń JavaScript itp.

3. Implementacja kompresji danych w internetowym systemie monitorującym

Jednym z podstawowych problemów działania systemu monitorującego po stronie przeglądarki internetowej jest sposób przekazania zebranych informacji do części serwerowej tego systemu. Problem ten wynika z wielkości generowanego ruchu sieciowego. W opisywanym systemie monitorującym co dwadzieścia zdarzeń do serwera WWW jest przekazywana paczka z przeglądarki internetowej. Wpływ na generowanie wielkości ruchu sieciowego ma rozmiar paczek i częstotliwość ich transferu. Niestety, nie można wpłynąć na częstotliwość wysyłania paczki bez straty informacji lub wzrostu jej wielkości. Krytyczną sprawą jest zatem zmniejszenie rozmiaru paczki i zachowanie dotychczasowych informacji na temat aktywności użytkownika na stronie internetowej. Idea ta polega

[9] J. Swacha: *Popularne standardy kompresji danych*. „Pro Dialog" 1999, nr 9, s. 23–32.

[10] *Ibidem*.

na zmniejszeniu narzutów stałych na każde zdarzenie i cała paczkę. Na wielkość paczki duży wpływ mają następujące narzuty:

– nazwy parametrów pliku zapisującego paczkę na serwerze,
– nazwy znaczników HTML,
– nazwy zdarzeń w przeglądarce internetowej,
– łańcuchy znaków określające datę i czas wygenerowania zdarzenia,
– wartość zdarzenia opisująca dodatkowe informacje o zdarzeniu.

Pierwszym i najprostszym krokiem w kierunku zmniejszenia rozmiaru paczki było skrócenie nazw parametrów skryptu zapisu. Paczki są wysyłane do serwera systemu monitorującego w postaci żądania przeglądarki internetowej za pomocą metody POST. Długość nazwy parametru ma wpływ na wielkość paczki. W tabeli 1 przedstawiono nazwy parametrów przed zmianą i po zmianie oraz wynikające z niej skrócenie długości zapisu (oszczędność pamięci).

Tabela 2

Oszczędności stałe z kodowania nazw parametrów

Lp.	Nazwa parametru		Oszczędność [B]	Oszczędność na paczkę [B] (po 20 zdarzeń)
	przed	po		
1.	parDateTimeKlient	D	16	320
2.	parRodzajElementu	R	16	320
3.	parZdarzenie	Z	11	220
4.	parTypElementu	likwidacja	14	280
5.	parWartosc	W	9	180
6.	parSkrypt	S	8	8
7.	EVL_tmpCookieName	A	16	16
8.	EVL_tmpCookieValue	B	17	17
9.	EVL_persCookieName	C	17	17
10.	EVL_persCookieValue	E	18	18
	Razem			1328

Źródło: opracowanie własne.

Pozycje od 1 do 5 to parametry tablicowe przesyłające informacje o zdarzeniu, dlatego oszczędność jest przemnożona przez standardową liczbę zdarzeń w paczce. Pozostałe parametry występują raz na cała paczkę. Należy zauważyć, że parametr parTypElementu, który służył do przesłania ewentualnego typu dla znacznika INPUT, został zlikwidowany, gdyż jego role przejął parametr parRodzajElementu.

Dużym narzutem opisu zdarzenia charakteryzowały się informacje na temat rodzaju znacznika, który brał udział w zdarzeniu, a także same nazwy zdarzeń. Udział tych danych w wielkości paczki został zwielokrotniony przez liczbę zdarzeń w paczce. Aby zmniejszyć tę porcję danych, wprowadzono kodowanie substytucyjne nazw poszczególnych elementów HTML i zdarzeń. Dzięki temu wszystkie nazwy zdarzeń i większość nazw znaczników uzyskały kody jednobajtowe, a część rzadko używanych – dwubajtowe. Przykładowo, znacznik TABLE zastąpiono literą D, a zdarzenie *onmouseover* –3.

Kolejnym sposobem zmniejszenia wielkości paczki są łańcuchy znaków określające datę i czas zajścia zdarzenia. Łańcuch taki ma następujący format: YYMMDDhhmmss, określający rok w formacie dwucyfrowym, miesiąc, dzień, godzinę, minuty i sekundy. Aby przesłać moment zajścia zdarzenia, należało zużyć 12 znaków. Dla zmniejszenia tego formatu bez straty informacji tylko pierwsze (najwcześniejsze) zdarzenie w paczce ma pełną informację o dacie i czasie. Następne zdarzenia w danej paczce ma tylko znacznik interwału od momentu pierwszego zdarzenia dla paczki. Przykładowe działanie przedstawiono w tabeli 3.

Tabela 3

Przydzielenie czasu do zdarzenia

Paczka	Zdarzenie	Zapis daty i czasu
1	1	060705013055
	2	1
	...	...
	20	15
2	1	060705013422
	2	2
	...	...
	20	45
...	...	...

Źródło: opracowanie własne.

Ostatnim punktem, w którym uzyskano zmniejszenie długości, była zawartość parametru wartości. Jest to parametr przekazujący do serwera WWW wszelkie dodatkowe dane o zdarzeniu i elemencie uczestniczącym w zdarzeniu. Dla zdarzenia *onmouseover* są to przykładowo współrzędne kursora urządzenia wskazującego względem całego ekranu lub obszaru roboczego przeglądar-

ki internetowej. Każdy składnik parametru wartości zawiera w sobie znaczniki określające charakter tej części wartości. Ulepszenie polegało tu na skróceniu nazw tych znaczników i nieużywanie znaków, które podczas przesyłania do serwera WWW zostaną przekształcone do bezpiecznej postaci trzyznakowej. Także zawartość poszczególnych składników, jeśli to możliwe, przekształcono tak, aby nie pojawiały się w niej znaki podlegające transformacji podczas transferu. Zmniejszono też liczbę składników, aby zaoszczędzić przestrzeń na objętości ich nazw. Przykładowo, dla zdarzenia *onmouseover* wystąpiły składniki w następującej formie:

EventParentX==**XX**;;EventParentY==**YY**;;EventScreenX==**XX**;;EventScreenY==**YY**,
EventOffsetX==**XX**;;EventOffsetY==**YY**;;EventClientX==**XX**;;EventClientY==**YY**.

Po przekształceniu otrzymano następującą formę:

X=**XX**p**YY**l**XX**p**YY**l**XX**p**YY**l**XX**p**YY**.

Dzięki temu wyłączono nadmiarowe znaki = oraz ;, które zostałyby przekształcone podczas przesyłania za pomocą protokołu HTTP do postaci szesnastkowego kodu ASCII. Przykładowo, dla znaku ; byłoby to %3B. Odrzucono również długie nazwy składników. Przy zapisie przykładowego zdarzenia *onmouseover* zaoszczędzono w ten sposób 178 bajtów.

4. Badania doświadczalne i dyskusja uzyskanych wyników

Zaproponowany schemat kompresji zastosowano do monitorowania zachowania użytkowników serwisu www.konferencja.org. Badanie przeprowadzono w maju 2006 roku. W tym czasie z serwisu skorzystało około 3,5 tys. użytkowników. Wygenerowali oni prawie 40 tys. paczek zdarzeń, co dało w przybliżeniu pół miliona zarejestrowanych zdarzeń.

Dokładna analiza zebranych danych pozwoliła zaobserwować znaczne korzyści z zastosowania zaproponowanej metody kompresji przesyłanych danych. Maksymalny zaobserwowany rozmiar przesłanej paczki w metodzie bez kompresji wyniósł 15385 bajtów, a w rozwiązaniu z kompresją – zaledwie 9321 bajtów, co daje prawie 40-procentowy stopień kompresji, przekładający się na proporcjonalną oszczędność obciążenia łącza. Duże oszczędności wynikają z porównania minimalnie zmierzonych rozmiarów paczek, które wyniosły odpowiednio: bez

kompresji – 519 bajtów, z kompresją – 274 bajty, czyli dały prawie 48-procentową oszczędność. Porównując średnie rozmiary paczek, zauważono, że rozwiązanie z kompresją okazało się aż o prawie 72% lepsze od rozwiązania bez kompresji. Dokładne zestawienie przedstawiono w tabeli 4.

Tabela 4

Porównanie rozmiarów paczek

Rozmiar	Bez kompresji	Z kompresją	Oszczędność (%)
Maksymalny	15385	9321	39,42
Minimalny	519	274	47,21
Średni	8122	2279	71,94
Odchylenie standardowe	1709	473	–

Źródło: opracowanie własne.

Rozmiary paczek mają decydujący wpływ na obciążenie łącza internetowego, zarówno po stronie klienta jak i serwera. W bardzo popularnych serwisach duży rozmiar paczki w połączeniu z dużą liczbą odwiedzających mógłby doprowadzić do przeciążenia serwera. Wyliczone wartości transferów dla serwisu konferencja.org podano w tabeli 5.

Tabela 5

Transfer danych do serwera

	Zmierzony w testach (MB)	Dla 10 tys. odsłon (GB)	Dla 100 tys. odsłon (GB)
Bez kompresji	307,8	2,06	20,62
Z kompresją	86,4	0,58	5,78

Źródło: opracowanie własne.

Rozmiary przesyłanych paczek wpływają na komfort pracy z serwisem. Im mniej czasu zajmuje przesłanie paczki, tym mniej uciążliwe (i zauważalne) jest monitorowanie dla użytkownika. Dla przykładu, nawet na dobrej jakości połączeniu modemowym przesłanie średniej paczki zajmuje ponad sekundę (uwzględniając czas nawiązania połączenia z serwerem). Tak długi czas transferu wykluczałby rozwiązanie monitorujące z powszechnego zastosowania. Zastosowanie kompresji pozwoliło skrócić ten czas do około 0,32 sekundy. Przybliżone czasy przesyłu danych do serwera dla popularnych prędkości łącz internetowych przedstawiono w tabeli 6.

Tabela 6

Przybliżone czasy przesyłu danych do serwera

Łącze	56 kb/s	128 kb/s	256 kb/s	512 kb/s
Bez kompresji	1,13 s	0,5 s	0,25 s	0,12 s
Z kompresją	0,32 s	0,14 s	0,07 s	0,03 s

Źródło: opracowanie własne.

Podsumowanie

Internetowe systemy monitorujące zachowanie użytkowników są bez wątpienia narzędziem przydatnym dla administratorów serwisów WWW. Są tym lepsze, im dostarczają precyzyjniejszych danych. Choć jest już system monitorujący, który pozwala na nieinwazyjne zbieranie szczegółowych danych, problemem pozostała ich ilość, nadmiernie obciążająca łącze.

W artykule przedstawiono techniki kompresji danych, które zastosowano w internetowym systemie monitorującym. Pozwoliły one na znaczną redukcję ilości danych przesyłanych na serwer. Praktyczna implementacja w serwisie www.konferencja.org umożliwiła zredukowanie średniej wielkości pakietu przeciętnie o 72 %.

IMPROVING WEB MONITORING SYSTEM BY IMPLEMENTING DATA COMPRESSION

Summary

Web monitoring systems are very useful in e-commerce, as the record of user actions constitute an important source of information for operational customer relationship management. As client-side monitoring of the low-level user activity requires a significant amount of data to be transmitted to the server, the connection speed can be noticeably decreased. In this paper we show how to solve this problem by compressing data before transmission. We explain the techniques used and present the results of applying compression in the system, positively verifying its usefulness.

Translated by Jakub Swacha