STUDIA INFORMATICA 2003
Volume 24 Number 2B (54)

Grzegorz HRYŃ
Politechnika Śląska, Instytut Informatyki

MOŻLIWOŚCI POPRAWY WYDAJNOŚCI POŁĄCZENIA POMIĘDZY KLIENTEM I SERWEREM WWW

Streszczenie. Wydajność jest jednym z najważniejszych czynników wpływających na popularność poszczególnych serwisów WWW. Artykuł stanowi przegląd badań mających na celu poprawę jakości połączenia pomiędzy klientem i serwerem WWW. Omawiane są sposoby klasyfikacji klientów w zależności od jakości łącza oraz możliwości reakcji serwera na żądania przychodzące od różnych grup klientów.

Słowa kluczowe: wydajność serwera WWW, charakterystyka klienta, postrzegana wydajność połączenia, protokół HTTP.

POSSIBILITIES OF IMPROVING END-TO-END WEB PERFORMANCE

Summary. Performance is the one of the most important factors that influences the popularity of a particular web service. This paper presents an overview of current research in the area of improvement of connectivity between client and server. There are also described client categorization methods depending on link quality and possible server actions in respond to requests coming from different client groups.

Keywords: Web server performance, client characterization, perceived connectivity performance, HTTP protocol.

1. Wstęp

Na przestrzeni kilku ostatnich lat wraz z rozwojem usług internetowych rosnąca liczba użytkowników sieci i serwerów usług WWW zaczęła powodować narastanie problemów związanych z wydajnością sieci komputerowych i serwerów. W związku z tym jednym z najważniejszych zadań stojących przed administratorami systemów informatycznych stało się zapewnienie odpowiedniej szybkości i jakości usług. Pierwszym krokiem do osiągnięcia tego celu jest analiza całego kanału komunikacyjnego pomiędzy klientem a serwerem WWW

i określenie jego parametrów w celu odpowiedniego dostrojenia go do potrzeb klienta. Ważnym elementem mającym wpływ na to, czy użytkownik będzie korzystał z serwisu WWW, jest pierwszy kontakt, a więc między innymi subiektywny czas otwarcia strony w przeglądarce klienta. Stąd w badaniach kładzie się szczególny nacisk nie tylko na wydajność samego serwera, ale również na badanie innych czynników wpływających na postrzeganą wydajność połączenia.

Problemem zasadniczym jest fakt, że chociaż współczesne serwery są w stanie obsłużyć wielką liczbę żądań na sekundę, rozkład tych żądań pochodzących od jednego klienta ma charakter impulsowy: użytkownik przeglądający zasoby sieci otwiera strony WWW, generuje duży chwilowy ruch, a następnie czyta i analizuje zawartość, nie obciąża serwera. Gdy nawarstwi się wiele takich okresów aktywności pochodzących od różnych użytkowników, to pomimo iż wydajność serwera jest odpowiednio obliczona dla średniego obciążenia, w momentach szczytowego natężenia powoduje zwiększenie czasu odpowiedzi serwera powyżej akceptowalnej przez klienta wartości.

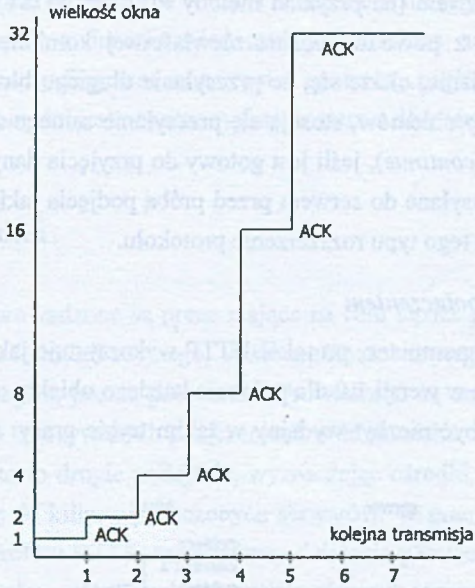
W przypadku badania całości systemu należy jednak wziąć pod uwagę nie tylko serwer, ale wszystkie elementy połączenia, a zatem klienta (przeglądarka internetowa), jakość łącza, protokół komunikacyjny czy też użycie serwera proxy. Jeśli w przypadku serwera usług administrator ma pewną możliwość usprawnienia jego działania, począwszy od optymalizacji parametrów do fizycznego ulepszenia poprzez wymianę dysków, pamięci, procesora czy dodanie kart sieciowych, to w przypadku kanału komunikacyjnego i przeglądarki internetowej serwer może tylko zareagować poprzez zmianę zawartości stron dostarczanych do klienta lub sposobu dostarczania.

2. Wydajność połączenia

Na wydajność połączenia ma wpływ między innymi wybór protokołu komunikacyjnego. W chwili obecnej są używane dwie wersje protokołu HTTP: 1.0 i 1.1. Standard w wersji 1.0 powstał jako następca wersji 0.9, dziś już praktycznie niespotykanej. Grupa Internet Engineering Task Force opracowała dokument RFC 1945 (1996), opisujący sposób użycia protokołu 1.0, ale nie definiujący standardu. Ponieważ jednak zmienił się nieco sposób publikowania serwisów WWW oraz liczba elementów potrzebnych do pobrania i wyświetlenia kompletnej strony a także wymagania stawiane protokołowi HTTP, więc po czterech latach prac powstała modyfikacja – wersja 1.1 – opisująca rozszerzenia tego protokołu.

2.1. Wady protokołu HTTP/1.0

Podstawową wadą tej wersji jest fakt, iż każda operacja pobrania elementu z serwera wymagała otwarcia nowego połączenia TCP, a następnie jego zamknięcia. Ponieważ jednak większość elementów strony ma wielkość poniżej 5000B ([1], a nawet 80% jak podaje [9]), w praktyce oznacza to, że większa część przesyłanych pakietów dla elementów tej wielkości to pakiety kontrolne TCP. Ponadto kiedy po raz pierwszy jest otwierane połączenie, TCP używa algorytmu nazywanego *slow start*, który polega na tym, że kilka pierwszych pakietów jest używanych do określenia optymalnych parametrów transmisji (rys. 1.). Przy małych elementach może się okazać, że zanim algorytm *slow start* zostanie zakończony, element strony może być już w całości przesłany, a zatem HTTP/1.0 wykorzystuje protokół TCP w sposób nieefektywny, co może w skrajnych przypadkach prowadzić do przeciążenia łącza nadmiarowymi pakietami, co zostało omówione w [2] i [3].



Rys. 1. Ilustracja algorytmu *slow start*
Fig. 1. Illustration of *slow start* algorithm

2.2. Protokół HTTP w wersji 1.1

W celu wyeliminowania niedogodności poprzedniej wersji, po czterech latach od opublikowania RFC 1945, powstała wersja 1.1 wprowadzająca, między innymi, kilka mechanizmów, mających wpływ na poprawę wydajności, omawianych w pracy [8]:

- optymalizacja przepustowości,

- zarządzanie połączeniem,
- kontrola transmisji.

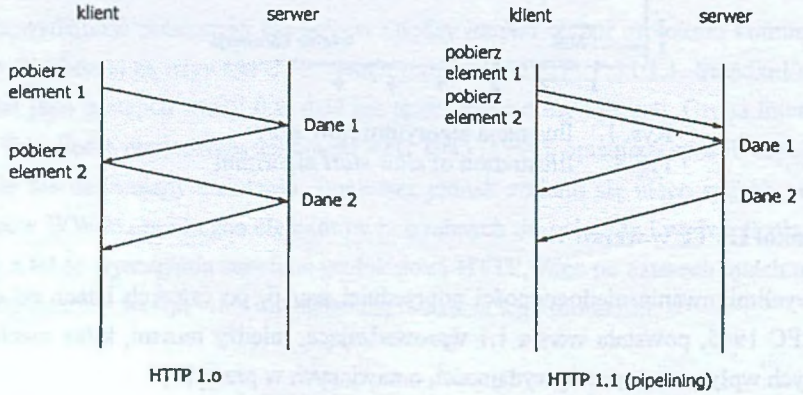
2.2.1. *Optymalizacja przepustowości*

W protokole HTTP/1.1 wprowadzono kilka rozwiązań, które pomogły uelastycznić jego zachowanie w stosunku do poprzedniej wersji, powodując znaczne oszczędności, jeśli chodzi o ilość przesyłanych danych. Przykładem takiego rozwiązania jest mechanizm *range request*. Klient nie zawsze potrzebuje całego zasobu, przykładowo przy przeglądaniu dużych dokumentów może chcieć przeglądać tylko początkowe akapity, aby zorientować się w ich treści przed pobraniem całości dużego dokumentu. Takie podejście może zaoszczędzić czas i zajętość łącza komunikacyjnego. Innym zastosowaniem jest wznowienie przerwanej transmisji, od miejsca w którym została ona przerwana.

Innymi rozwiązaniami tego typu są *expect* i *continue*. Znajdują one zastosowanie podczas przesyłania żądań do serwera (na przykład metody PUT czy POST), których długość może być duża. Jeśli serwer z powodu podania niewłaściwej kombinacji nazwy użytkownika i hasła odrzuci takie żądanie, okaże się, że przesyłanie długiego bloku danych było zbędne. Aby uniknąć tego typu problemów, stosuje się przesyłanie samego żądania, na które serwer odpowiada kodem 100 (*continue*), jeśli jest gotowy do przyjęcia danych (ciała) tego żądania. Polecenie *expect* jest wysyłane do serwera przed próbą podjęcia takiej transmisji, aby upewnić się, czy obsługuje on tego typu rozszerzenie protokołu.

2.2.2. *Zarządzanie połączeniem*

Jak już to zostało wspomniane, protokół HTTP wykorzystuje jako medium transportowe protokół TCP. Ponieważ w wersji 1.0 dla pobrania każdego obiektu otwierano nowe połączenie, TCP okazywał się być niezbyt wydajny w takim trybie pracy: nawiązanie i zamykanie



Rys. 2. Przetwarzanie potokowe
Fig. 2. Pipelining

połączenia stanowiło duży narzut czasowy. Aby zminimalizować wpływ tej niedogodności, wprowadzono dwa mechanizmy: stałych połączeń (*persistent conntections*) i przetwarzania potokowego (*pipelining*). Pierwszy z nich powoduje, że połączenie jest utrzymywane podczas całej komunikacji pomiędzy klientem a serwerem, a nie tylko na czas przesłania jednego obiektu. Oczywiście obie strony mogą zamknąć sesję w dowolnym momencie. Drugi mechanizm umożliwia wysłanie kilku żądań bez oczekiwania po każdym z nich na odpowiedź ze strony serwera. Mechanizm ten ilustruje rys. 2.

2.2.3. Kontrola transmisji

Coraz większa część danych przesyłanych przez serwer jest generowana w sposób dynamiczny, w związku z czym serwer nie może użyć nagłówka *Content-Length*, ponieważ przed wysłaniem całości danych nie zna ich wielkości. Rozwiązaniem może być zamknięcie sesji, podobnie jak w przypadku protokołu w wersji 1.0, ale nie jest to optymalne rozwiązanie. Zaproponowano następujące rozwiązanie: wiadomość jest dzielona na kawałki o stałej wielkości, a co za tym idzie każdy fragment może zostać opatrzony nagłówkiem definiującym jego wielkość. Koniec wiadomości jest oznaczony fragmentem o długości 0.

3. Kierunki badań

W chwili obecnej prowadzone są prace mające na celu lepsze poznanie wpływu wszystkich elementów kanału komunikacyjnego, w szczególności użytego protokołu, na efektywność działania i postrzeganą jakość połączenia. Prowadzone są one zarówno w środowisku laboratoryjnym, jak i na rzeczywistych połączeniach w środowisku sieci Internet.

W pracy [6] przyjęto to drugie podejście, wyznaczając ośrodki, w których badano klientów sieci, łączących się do kilku wyznaczonych serwerów. W pracy brano pod uwagę różne opcje obu wersji protokołów HTTP, mechanizmy działania pamięci podręcznej (*cache*), budowę strony WWW z elementów pochodzących z różnych serwerów, jak również mechanizm pobierania fragmentów zasobu (*byte range requests*). Przedstawione wyniki potwierdzają, że nowe mechanizmy wprowadzone w HTTP/1.1 znacząco poprawiają postrzeganą jakość połączenia. Zwrócono również uwagę, że zarządzanie połączeniem powinno być bardziej deterministyczne, jeśli chodzi o zamykanie połączenia – nie powinno być ono zamykane podczas procesu pobierania pojedynczej strony WWW.

W opracowaniu [1] skupiono się na przygotowaniu modelu ruchu, odbywającego się pomiędzy klientem a serwerem, na poziomie sesji. Zanalizowano wpływ różnych parametrów na proces otrzymywania żądań HTTP przez serwer oraz rozkład przybywających pakietów i zaproponowano model kolejkowy do analizy obciążenia serwera Apache, gdy używany jest

mechanizm stałych połączeń (*persistent connections*). Okazało się, że przy dużym obciążeniu serwera zalety wersji 1.1 protokołu HTTP mogą stać się wadami przy niewłaściwym doborze parametrów konfiguracyjnych serwera i zachłannej polityce klienta.

4. Problem klasyfikacji klientów

Ponieważ postrzegana jakość połączenia jest jednym z najważniejszych elementów mających wpływ na zadowolenie użytkownika, rozpoczęto badania nad możliwością dostosowania pracy serwera do potrzeb klienta, związanych z jakością kanału komunikacyjnego. Pierwszym krokiem, który w tym celu należy zrobić, jest klasyfikacja klientów, będąca podstawą do dalszych działań.

Podstawową metodą przyjmowaną podczas klasyfikacji jest analiza dzienników (*logs*) serwera WWW [4]. W idealnym przypadku do klasyfikacji potrzebny byłby czas przejścia RTT (*round trip time*), przepustowość łącza oraz obciążenie łącza. Istnieje również możliwość wysłania przez klienta specyfikacji, ale nie jest to wymuszone i nieformalne z punktu widzenia zaleceń dotyczących protokołu HTTP, a zatem nie można się opierać na tej metodzie. Najczęściej zatem serwer musi zadowolić się przybliżoną oceną RTT, która jest dokonywana na podstawie pomiaru czasu między odpowiedzią serwera a następnym żądaniem klienta. Badanie parametru RTT prowadzone jest na podstawie dzienników serwerów WWW, ponieważ wydaje się, że aktywne mierzenie może stanowić zbyt duży narzut na komunikację.

W pracy [7] przedstawiono metodologię i wyniki badań przeprowadzonych na podstawie rzeczywistych danych z kilku serwerów i zaproponowano sposób klasyfikacji oraz podział na trzy klasy klientów.

5. Możliwości działania ze strony serwera

Po przeprowadzeniu klasyfikacji klienta serwer WWW może podjąć różne działania w celu dostosowania zawartości przesyłanych stron w zależności od „jakości" klienta. Są to, między innymi:

- zmiana zawartości strony, która może sprowadzać się do zmiany grafiki na „uboższą" (mniejszą objętościowo), pominięcia niektórych elementów graficznych, takich jak animacje, reklamy,
- wybór innej repliki serwisu WWW (np. położonej topologicznie bliżej),
- zmiany znaczników meta, powodujące inne zachowanie serwerów proxy,
- zmiana metody dostarczania zawartości: kompresja, szyfrowanie,

- szybsze zamykanie sesji w zależności od jakości klienta i ustalenie odpowiednich priorytetów. W szczególności ma to duże znaczenie w przypadku klientów łączących się przez modem, gdzie ważne jest utrzymywanie sesji ze względu na duży narzut komunikacyjny podczas zamykania i otwierania sesji.

6. Podsumowanie

Niniejsze opracowanie ma na celu przedstawienie przeglądu metod służących badaniu połączenia pomiędzy serwerem a klientem WWW, klasyfikacji klientów i możliwych reakcji ze strony serwera podejmowanych na podstawie przydziału do odpowiedniej kategorii. Będzie ono stanowić dla autora podstawę do przeprowadzenia własnych badań w temacie klasyfikacji klientów oraz zmiany zawartości strony WWW w zależności od klasyfikacji klienta żądającego dostępu do zasobu.

LITERATURA

1. Liu Z., Niclausse N., Jalpa-Villanueva C., Barbier S.: Traffic Model and Performance Evaluation of Web Servers. Performance Evaluation, 2001, p. 77-100.
2. Nielsen H., Gettys J., Baird-Smith A., Prud'hommeaux E., Lie H., Lilley C.: Network performance effects of HTTP/1.1, CSS1, and PNG. Proceedings of ACM SIGCOMM'97 Conference, 1997.
3. Touch J., Heidemann J., Obraczka K.: Analysis of HTTP Performance. Raport badawczy University of Southern California, 1998.
4. Srikant R., Yang Y.: Mining Web Logs to Improve Website Organization. ACM Press, Proceedings of the 10th International Conference on World Wide Web, 2001, p. 430-437.
5. Chen X., Mohapatra P., Chen H.: An Admission Control Scheme for Predictable Server Response Time for Web Accesses. ACM Press, Proceedings of the 10th International Conference on World Wide Web, 2001, p. 545-554.
6. Krishnamurthy B., Wills C. E.: Analyzing Factors That Influence End-To-End Web Performance. North-Holland Publishing Co., Proceedings of the 9th international World Wide Web conference on Computer networks, 2000, p. 17-32.
7. Krishnamurthy B., Wills C. E.: Improving Web Performance By Client Characterization Driven Server Adaptation. ACM Press, Proceedings Of The 11th International Conference On World Wide Web, 2002, p. 305-316.

8. Krishnamurthy B., Mogul J. C., Kristol D. M. Key Differences Between HTTP/1.0 And HTTP/1.1. Elsevier North-Holland, Inc., Proceedings Of The 8th International Conference On World Wide Web, 1999, p. 1737-1751.
9. Kruse H., Allman M., Mallasch P.: Network and User-Perceived Performance of Web Page Retrievals. Proceedings of the First International Conference on Telecommunications and Electronic Commerce (ICTEC), 1998.

Recenzent: Dr inż. Arkadiusz Sochan

Wpłynęło do Redakcji 30 marca 2003 r.

Abstract

Performance is the one of the most important factors that influences the popularity of a particular web service. This paper presents an overview of current research in the area of improvement of connectivity between client and server. There are some factors specific to HTTP protocol that may influence connection efficiency: slow start algorithm (Fig. 1.), range requests, *Expect and Continue* method, persistent connection and pipelining (Fig. 2.). This paper also presents current trends in the area of analyzing end-to-end web performace. There are two possible options: using lab environment to isolate chosen factors and preparing appropriate model to perform further analysis or using actual data gathered in real environment of working web serwers.

To improve user-perceived performance, server can take different actions after assigning client to appropriate category: deliver altered content, redirect client to a different replica, change proxy behaviour by altering meta-information, use compression and define adequate connection close policy.

Adres

Grzegorz HRYŃ: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-101 Gliwice, Polska, grzegorz.hryn@polsl.gliwice.pl.