

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# Wydajne witryny internetowe. Przyspieszanie działania serwisów WWW

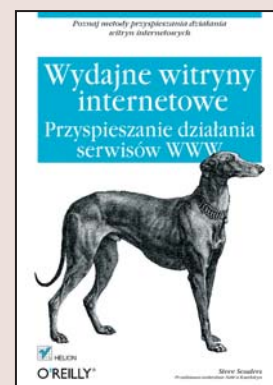
Autor: Steve Souders

Tłumaczenie: Robert Górczyński

ISBN: 978-83-246-1518-6

Tytuł oryginału: [High Performance Web Sites: Essential Knowledge for Front-End Engineers](#)

Format: 168x237, stron: 160



### Poznaj metody przyspieszania działania witryn internetowych

- Jak zredukować ilość żądań HTTP?
- W jaki sposób skompresować przesyłane dane?
- Jak zoptymalizować interfejsy w technologii Ajax?

Tempo rozwoju internetu przerosło najśmielsze oczekiwania. Każdego dnia w sieci pojawiają się nowe witryny WWW i aplikacje internetowe, przyciągające rzesze użytkowników. Niestety, nadal największym problemem internautów jest niska przepustowość łącza, która często skutecznie zniechęca do korzystania z bardziej rozbudowanych serwisów internetowych. Przed projektantami witryn WWW oraz twórcami interfejsów użytkownika dla aplikacji sieciowych stoi poważne wyzwanie – trzeba zoptymalizować i przyspieszyć funkcjonowanie tego, co użytkownik widzi w przeglądarce. W ciągu ostatnich kilku lat pojawiło się sporo ciekawych rozwiązań, znacznie ułatwiających ten proces, lecz stworzenie naprawdę wydajnego przeglądarkowego interfejsu użytkownika wymaga opanowania wielu zagadnień.

Książka „Wydajne witryny internetowe. Przyspieszanie działania serwisów WWW” to kompleksowy przegląd i omówienie czynników składających się na optymalny interfejs użytkownika aplikacji internetowej. Czytając ją, dowiesz się, jak przyspieszyć działanie witryn WWW poprzez ograniczanie ilości przesyłanych danych, buforowanie ich i kompresowanie, optymalne rozmieszczenie elementów graficznych i zdefiniowanie stylów CSS. Nauczysz się także ograniczać rozmiary skryptów, korzystać z technologii Ajax i przyspieszać działanie elementów utworzonych za pomocą programu Adobe Flash.

- Ograniczanie ilości żądań HTTP
- Korzystanie z nagłówka Expires
- Stosowanie kompresji GZIP
- Odpowiednie umieszczanie definicji stylów oraz kodów skryptów w dokumentach HTML
- Optymalne korzystanie z CSS
- Stosowanie zewnętrznych plików ze skryptami
- Redukcja ilości zapytań DNS
- Buforowanie zapytań Ajax

**Twórz wydajne i szybko działające witryny internetowe**

Wydawnictwo Helion  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 032 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)



---

# Spis treści

<b>Przedmowa .....</b>	<b>9</b>
<b>Wprowadzenie.....</b>	<b>11</b>
<b>A Waga wydajności interfejsu .....</b>	<b>17</b>
Określanie wydajności strony internetowej	17
Co dzieje się z czasem udzielania odpowiedzi?	19
Złota reguła wydajności	19
<b>B Ogólny opis protokołu HTTP .....</b>	<b>23</b>
Kompresja	24
Warunkowe żądania GET	24
Nagłówek Expires	25
Funkcja Keep-Alive	25
To tylko wprowadzenie	26
<b>1. Reguła 1. Wykonywanie mniejszej liczby żądań HTTP .....</b>	<b>27</b>
Mapy obrazkowe	27
Obrazy CSS Sprite	28
Osadzone obrazy	30
Łączenie skryptów i arkuszy stylów	32
Podsumowanie	33
<b>2. Reguła 2. Używanie Content Delivery Network .....</b>	<b>35</b>
Content Delivery Network	36
Oszczędności	37
<b>3. Reguła 3. Używanie nagłówka Expires .....</b>	<b>39</b>
Nagłówek Expires	39
Dyrektywa max-age oraz moduł mod_expires	40
Bufor pusty kontra zapełniony	41
Nie tylko obrazy	43
Zmiana nazw plików	44
Przykłady	45

<b>4. Reguła 4. Kompresja gzip .....</b>	<b>47</b>
W jaki sposób działa kompresja?	47
Jakie dane kompresować?	48
Oszczędności	49
Konfiguracja	49
Buforowanie proxy	51
Przypadki skrajne	52
Kompresja gzip w działaniu	53
<b>5. Reguła 5. Umieszczanie arkuszy stylów na początku dokumentu .....</b>	<b>55</b>
Progresywne generowanie	55
Narzędzie sleep.cgi	56
Pusty biały ekran	57
Chwilowe wyświetlenie treści strony bez zastosowanych stylów	60
Co może zrobić programista interfejsu?	61
<b>6. Reguła 6. Umieszczanie skryptów na końcu dokumentu .....</b>	<b>63</b>
Problemy związane ze skryptami	63
Równoległe pobieranie elementów strony	64
Zablokowanie przez skrypty możliwości pobierania innych elementów strony	66
Najgorsze rozwiązanie: skrypty na początku strony	67
Najlepsze rozwiązanie: skrypty na końcu dokumentu	67
Spojrzenie w przyszłość	68
<b>7. Reguła 7. Unikanie wyrażeń CSS .....</b>	<b>69</b>
Uaktualnianie wyrażeń	69
Rozwiązanie problemu	70
Podsumowanie	71
<b>8. Reguła 8. Używanie zewnętrznych plików JavaScript i CSS .....</b>	<b>73</b>
Kod na stronie kontra w zewnętrznym pliku	73
Typowe wyniki	76
Strona główna	76
Połączenie zalet obu rozwiązań	77
<b>9. Reguła 9. Redukcja liczby zapytań DNS .....</b>	<b>81</b>
Buforowanie zapytań DNS oraz wartość TTL	81
Perspektywa przeglądarki internetowej	84
Zmniejszanie liczby zapytań DNS	86
<b>10. Reguła 10. Zmniejszanie objętości kodu JavaScript .....</b>	<b>87</b>
Zmniejszanie objętości kodu	87
Zaciemnianie kodu	87

Oszczędności	89
Przykłady	90
Dodatkowe rozwiązania	91
<b>11. Reguła 11. Unikanie przekierowań .....</b>	<b>95</b>
Rodzaje przekierowań	95
W jaki sposób przekierowanie zmniejsza wydajność?	96
Alternatywa dla przekierowania	98
<b>12. Reguła 12. Usuwanie duplikujących się skryptów .....</b>	<b>105</b>
Powtarzające się skrypty — to się zdarza	105
Duplikujące się skrypty zmniejszają wydajność	106
Unikanie powtarzających się skryptów	107
<b>13. Reguła 13. Konfiguracja nagłówka ETag .....</b>	<b>109</b>
Czym jest ETag?	109
Problem związany z nagłówkiem ETag	111
Nagłówek ETag — używaj go lub zapomnij o nim	113
Używanie nagłówka ETag w praktyce	114
<b>14. Reguła 14. Buforowanie zapytań Ajax .....</b>	<b>117</b>
Web 2.0, DHTML oraz Ajax	117
Asynchroniczny = natychmiastowy?	119
Optymalizacja zapytań Ajax	120
Buforowanie zapytań Ajax w rzeczywistych aplikacjach	120
<b>15. Analiza dziesięciu witryn internetowych .....</b>	<b>125</b>
Wielkość strony, czas udzielania odpowiedzi i ocena w skali YSlow	125
W jaki sposób zostały przeprowadzone testy?	127
Amazon	128
AOL	130
CNN	133
eBay	135
Google	137
MSN	141
MySpace	143
Wikipedia	146
Yahoo!	148
YouTube	150
<b>Skorowidz .....</b>	<b>153</b>

---

# Reguła 9. Redukcja liczby zapytań DNS

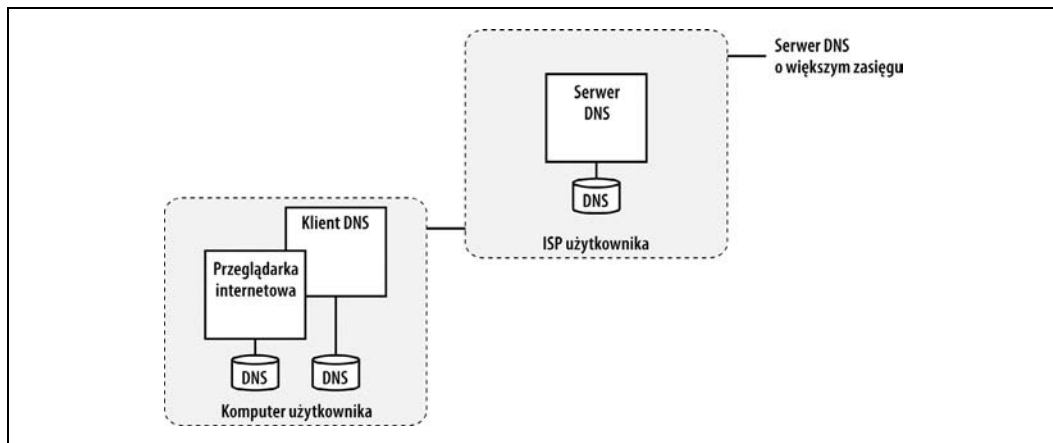
Internet jest siecią bazującą na znajdowaniu serwerów poprzez ich adresy IP. Adresy IP są trudne do zapamiętania; nazwę serwera zawiera zwykle adres URL. Jednak adres IP nadal pozostaje niezbędny w celu wykonania żądania przez przeglądarkę internetową. Określenie tego adresu jest zadaniem serwera DNS (Domain Name System). Wymieniony serwer DNS zmienia nazwy serwerów WWW na adresy IP, podobnie jak książka telefoniczna, w której dla danego nazwiska lub instytucji można odszukać numer telefonu. Po wpisaniu w przeglądarce internetowej adresu *www.yahoo.com*, serwer DNS łączy się z przeglądarką i zwraca jej adres IP wskazanego serwera WWW.

Powyższe objaśnienie przedstawia dodatkową zaletę serwera DNS: stanowi on warstwę pośrednią między adresem URL i rzeczywistym serwerem WWW. Jeżeli wymieniony serwer WWW zostanie zastąpiony przez inny z odmiennym adresem IP, wówczas DNS i tak umożliwi użytkownikowi używanie tej samej nazwy w celu uzyskania dostępu do nowego serwera. Ewentualnie, podobnie jak w przypadku witryny *www.yahoo.com*, z nazwą może być powiązana większa liczba adresów IP, które gwarantują większą wydajność witryny internetowej.

Jednak używanie serwera DNS wiąże się z kosztami. Wyszukanie adresu IP dla podanego adresu URL zwykle zajmuje przeglądarce internetowej od 20 do 120 milisekund. Do chwili zakończenia zapytania DNS przeglądarka nie może pobierać żadnych elementów z podanego serwera WWW. Czas udzielania odpowiedzi na żądanie użytkownika zależy więc także od szybkości działania serwera DNS (zwykle zapewnianego przez dostawcę usług internetowych — ISP), obciążenia serwera, odległości od niego oraz szybkości połączenia z internetem. Po opisie działania DNS z punktu widzenia przeglądarki internetowej zostanie opisana technika zmniejszenia ilości czasu poświęcanego na wykonywanie zapytań DNS.

## Buforowanie zapytań DNS oraz wartość TTL

Zapytania DNS są buforowane w celu zwiększenia wydajności. Proces buforowania może występować na specjalnych serwerach buforujących obsługiwanych przez ISP bądź w sieci lokalnej, choć wówczas buforowanie będzie obejmowało poszczególne komputery użytkowników. Jak pokazano na rysunku 9.1, po żądaniu przez użytkownika nazwy serwera WWW informacje dotyczące DNS pozostają w buforze DNS systemu operacyjnego (w systemie Microsoft Windows jest to usługa o nazwie Klient DNS). Dalsze żądania dostępu do danego serwera nie wymagają więc wykonywania kolejnych zapytań DNS, przynajmniej przez pewien czas.



Rysunek 9.1. Buforowanie DNS z punktu widzenia przeglądarki internetowej

Wystarczająco proste? Chwila, moment — przecież większość przeglądarek internetowych posiada własne bufory, zupełnie niezależne od bufora systemu operacyjnego. Dopóki przeglądarka przechowuje rekord DNS we własnym buforze, nie musi angażować systemu operacyjnego podczas żądania dostępu do rekordu. Dopiero po usunięciu rekordu z bufora przeglądarki następuje zapytanie kierowane do systemu operacyjnego. W takiej sytuacji system operacyjny albo będzie posiadał i dostarczy odpowiedni rekord z bufora, albo wyśle żądanie do zdalnego serwera, co spowoduje wystąpienie potencjalnego opóźnienia w czasie udzielania odpowiedzi na żądanie użytkownika.

Proces staje się jeszcze bardziej skomplikowany po uwzględnieniu faktu, że programiści zdają sobie sprawę, iż adres IP ulega zmianie i zużywa pamięć bufora. Dlatego też rekordy DNS są okresowo usuwane z pamięci, a kilka różnych ustawień konfiguracyjnych określa sposób i częstotliwość ich usuwania.

## Czynniki wpływające na buforowanie DNS

W pierwszej kolejności serwer powinien określić czas buforowania rekordów. Zwracany jako wynik zapytania rekord DNS posiada tak zwaną wartość TTL (*Time-To-Live*) określającą maksymalny czas ważności pakietu w sieci. Wymieniona wartość informuje klienta o maksymalnym czasie buforowania danego rekordu.

Chociaż bufor systemu operacyjnego honoruje wartość TTL, to przeglądarki internetowe ignorują ją i ustalają własne ograniczenia czasowe. Co więcej, przedstawiona w rozdziale B funkcja `Keep-Alive` protokołu HTTP może nadpisać zarówno wartość TTL, jak i ograniczenia czasowe przeglądarki. Innymi słowy, dopóki przeglądarka internetowa i serwer WWW utrzymują komunikację oraz otwarte połączenie TCP, nie ma żadnego powodu wykonywania kolejnych zapytań DNS.

Przeglądarki internetowe stosują własne ograniczenia dotyczące liczby buforowanych rekordów DNS, niezależnie od czasu, w jakim rekordy znajdują się w buforze. Jeżeli w krótkim czasie użytkownik odwiedza wiele różnych witryn posiadających odmienne nazwy domen, wówczas wcześniejsze rekordy DNS zostaną usunięte z bufora i zajdzie konieczność przeprowadzenia kolejnego zapytania DNS.

Należy pamiętać, że po usunięciu rekordu DNS z bufora przeglądarki system operacyjny wciąż może zawierać go we własnym buforze. W takim przypadku uniknie się potrzeby wykonywania zapytania sieciowego, które może wyraźnie wydłużyć czas udzielania odpowiedzi na żądanie użytkownika.

## Wartości TTL

Maksymalne wartości TTL dla dziesięciu najczęściej odwiedzanych witryn w USA wahają się od minuty do jednej godziny, jak przedstawiono w tabeli 9.1.

Tabela 9.1. Wartości TTL

Domena	TTL
<a href="http://www.amazon.com">http://www.amazon.com</a>	1 minuta
<a href="http://www.aol.com">http://www.aol.com</a>	1 minuta
<a href="http://www.cnn.com">http://www.cnn.com</a>	10 minut
<a href="http://www.ebay.com">http://www.ebay.com</a>	1 godzina
<a href="http://www.google.com">http://www.google.com</a>	5 minut
<a href="http://www.msn.com">http://www.msn.com</a>	5 minut
<a href="http://www.myspace.com">http://www.myspace.com</a>	1 godzina
<a href="http://www.wikipedia.org">http://www.wikipedia.org</a>	1 godzina
<a href="http://www.yahoo.com">http://www.yahoo.com</a>	1 minuta
<a href="http://www.youtube.com">http://www.youtube.com</a>	5 minut

Dlaczego pomiędzy wymienionymi wartościami występują tak duże różnice? Prawdopodobnie wynika to z połączenia czynników geograficznych i historycznych. Interesujący dokument RFC<sup>1</sup> zawiera szczegółowy opis formatu rekordów DNS oraz najczęściej popełnianych podczas ich konfiguracji błędów. Pierwsza wskazówka w wymienionym dokumencie dotyczy unikania ustalania zbyt małych wartości TTL, a wartość zalecana wynosi jeden dzień!

Wymienione dziesięć najczęściej odwiedzanych witryn w USA posiada ogromną liczbę użytkowników, a ich celem jest poprawne działanie DNS pomimo awarii serwera, wirtualnego adresu IP (VIP) lub kolokacji serwera. Z tego powodu Yahoo! stosuje małą wartość TTL. Z drugiej strony, MySpace używa jednej kolokacji serwera. Poprawne działanie usługi pomimo awarii ma mniej krytyczne znaczenie dla bieżącej topologii sieci, stąd zastosowanie większej wartości TTL, która zmniejsza liczbę zapytań DNS, co z kolei redukuje obciążenie serwerów DNS.

Zalecenia dotyczące konfiguracji DNS są poza zasięgiem tematycznym niniejszej książki. Znacznie istotniejszym zagadnieniem jest wpływ buforowania DNS na wydajność stron internetowych. Zapoznamy się więc z buforowaniem DNS z punktu widzenia przeglądarki internetowej w celu określenia liczby zapytań DNS wykonywanych przez strony internetowe.

Przeciętna wartość TTL otrzymywana przez klienta z rekordu DNS jest połową maksymalnej wartości TTL. Wynika to z faktu, że serwer DNS sam posiada wartość TTL powiązaną z rekordem DNS. Kiedy przeglądarka wykonuje zapytanie DNS, wtedy serwer DNS zwraca ilość czasu pozostałą w TTL dla danego rekordu. Jeżeli maksymalna wartość TTL wynosi 5 minut,

<sup>1</sup> Common DNS Data File Configuration Errors, <http://tools.ietf.org/html/rfc1537>.

wówczas wartość TTL zwracana przez serwer DNS waha się od jednej do trzystu sekund, natomiast wartość przeciętna wynosi 150 sekund. Wartość TTL dla danego serwera WWW jest inna podczas wykonywania każdego zapytania DNS.

## Perspektywa przeglądarki internetowej

Jak wspomniano we wcześniejszym podrozdziale „Czynniki wpływające na buforowanie DNS”, duża liczba czynników ma wpływ na liczbę zdalnych zapytań DNS wykonywanych przez przeglądarkę internetową. Istniejąca specyfikacja DNS (<http://tools.ietf.org/html/rfc1034>) pozwala klientom na elastyczność w kwestii sposobu działania bufora DNS. W podrozdziale skoncentrujemy się na przeglądarkach Internet Explorer oraz Firefox w systemie Microsoft Windows, gdyż są to najpopularniejsze platformy.

Bufor DNS w systemie Microsoft Windows jest zarządzany przez usługę Klient DNS. Użytkownik może wyświetlić i usunąć zawartość wymienionej usługi za pomocą polecenia `ipconfig`:

```
ipconfig /displaydns
ipconfig /flushdns
```

Ponowne uruchomienie systemu również powoduje wyczyszczenie bufora usługi Klient DNS. Oprócz wymienionej usługi przeglądarki Internet Explorer oraz Firefox posiadają własne bufory DNS. Ponowne uruchomienie przeglądarki internetowej powoduje wyczyszczenie bufora przeglądarki, ale nie bufora usługi Klient DNS.

## Internet Explorer

Bufor DNS przeglądarki Internet Explorer jest nadzorowany przez trzy ustawienia rejestru: `DnsCacheTimeout`, `KeepAliveTimeout` oraz `ServerInfoTimeOut` tworzone w następującym kluczu rejestru:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\
```

Na witrynie Microsoftu znajdują się dwa artykuły opisujące wpływ powyższych ustawień rejestru na bufor DNS<sup>2</sup>. Te artykuły wskazują następujące wartości domyślne wymienionych powyżej ustawień rejestru:

- `DnsCacheTimeout`: 30 minut,
- `KeepAliveTimeout`: 1 minuta,
- `ServerInfoTimeOut`: 2 minuty.

Warto w tym miejscu wspomnieć (ponieważ nie zostało to dobrze wyjaśnione w artykułach Microsoftu), że wartości TTL serwera DNS mniejsze niż 30 minut mają mały wpływ na częstotliwość zapytań DNS wykonywanych przez przeglądarkę. Gdy przeglądarka umieści w buforze rekord DNS, będzie on używany przez 30 minut. Jeżeli wystąpi błąd, wtedy zapytanie DNS zostanie wykonane jeszcze przed upływem tego czasu. W normalnych warunkach mała wartość TTL (poniżej trzydziestu minut) nie spowoduje zwiększenia liczby zapytań DNS wykonywanych przez przeglądarkę Internet Explorer.

<sup>2</sup> *How Internet Explorer uses the cache for DNS host entries*, <http://support.microsoft.com/default.aspx?scid=KB;en-us;263558>.

*How to change the default keep-alive time-out value in Internet Explorer*, <http://support.microsoft.com/kb/813827>.



Rola funkcji Keep-Alive również jest ważna. Domyślnie trwałe połączenie TCP jest używane, jeśli okres bezczynności osiągnie jedną minutę. Ponieważ połączenie jest utrzymywane, wykonanie zapytania DNS nie jest wymagane (zalety funkcji Keep-Alive zostały omówione w rozdziale B). Istnieje jeszcze dodatkowa korzyść — funkcja Keep-Alive pozwala uniknąć powtarzania zapytań DNS poprzez ponowne wykorzystanie nawiązanego połączenia.

Wartość dwóch minut ustawienia `ServerInfoTimeout` oznacza, że jeśli nawet bez stosowania funkcji Keep-Alive nazwa serwera WWW zostanie ponownie użyta przed upływem dwóch minut, to zapytanie DNS będzie niepotrzebne. W testach z wykorzystaniem przeglądarki Internet Explorer okazało się, że jeśli nazwa serwera WWW była używana co najmniej raz na dwie minuty, wtedy przez ponad 30 minut nie były wykonywane żadne zapytania DNS (przy założeniu, że nie wystąpiły problemy z dostępem do podanego adresu IP).

To jest ważna informacja dla centrów operacji sieciowych próbujących rozproszyć ruch sieciowy poprzez zmiany wartości DNS. Podczas rozpraszania takiego ruchu użytkownicy przeglądarki Internet Explorer przez co najmniej 30 minut będą korzystali ze starego rekordu DNS, zanim nastąpi jego aktualizacja. Użytkownicy aktywnie korzystający z witryny (przynajmniej raz na dwie minuty) nadal będą używali poprzedniego adresu IP, a aktualizacja rekordu DNS nie będzie miała miejsca aż do chwili wystąpienia awarii.

## Firefox

Przeglądarka internetowa Firefox jest znacznie prostsza w zakresie obsługi DNS i posiada trzy następujące ustawienia konfiguracyjne:

- `network.dnsCacheExpiration`: 1 minuta,
- `network.dnsCacheEntries`: 20,
- `network.http.keep-alive.timeout`: 5 minut.

Rekordy DNS są buforowane przez okres jednej minuty poza ich wartość TTL. Ponieważ jest to mała ilość czasu, to ustawienie niskiej wartości TTL (mniej niż jedna godzina) prawdopodobnie spowoduje zwiększenie liczby wykonywanych zapytań DNS wymaganych przez strony internetowe wyświetlane w przeglądarce Firefox.

Zaskakujący jest fakt, że domyślnie tylko dwadzieścia rekordów DNS jest buforowanych przez przeglądarkę Firefox. Oznacza to, że w związku z koniecznością wykonania dodatkowych zapytań DNS użytkownicy odwiedzający dużą liczbę witryn z oddzielnymi nazwami domen doświadczą wolniejszego działania, niż gdyby korzystali z przeglądarki Internet Explorer.

Wartość Keep-Alive w przypadku przeglądarki Firefox jest większa niż w przypadku Internet Explorera: pięć minut wobec jednej minuty. Upewnienie się przez programistę, że serwer WWW obsługuje funkcję Keep-Alive, spowoduje zmniejszenie liczby zapytań DNS wymaganych w trakcie odwiedzania tworzonej przez niego witryny.

Fasterfox (<http://fasterfox.mozdev.org>) jest doskonale znanym rozszerzeniem przeglądarki Firefox i służy do pomiaru oraz usprawniania wydajności przeglądarki Firefox. W celu porównania poniżej zostały przedstawione wartości wymienionych wcześniej ustawień DNS zmodyfikowane przez rozszerzenie Fasterfox:

- `network.dnsCacheExpiration: 1 godzina,`
- `network.dnsCacheEntries: 512,`
- `network.http.keep-alive.timeout: 30 sekund.`

## Zmniejszanie liczby zapytań DNS

Kiedy bufor DNS klienta jest pusty (zarówno w przeglądarce internetowej, jak i w systemie operacyjnym), wówczas liczba zapytań DNS odpowiada liczbie unikatowych nazw serwerów WWW używanych przez stronę internetową. Dotyczy to serwerów WWW użytych w adresie URL strony, obrazach, plikach skryptów, arkuszach stylów, obiektach Flash itd. Zmniejszenie liczby unikatowych nazw serwerów WWW powoduje zmniejszenie liczby wykonywanych zapytań DNS. Witryna Google (<http://www.google.com>) jest doskonałym przykładem osiągnięcia właściwego celu — wymaga tylko jednego zapytania DNS.

Zmniejszenie liczby unikatowych nazw serwerów WWW może spowodować zmniejszenie ilości równoległe pobieranych na stronie danych. Uniknięcie wykonywania zapytań DNS skraca czas udzielania odpowiedzi na żądanie użytkownika, ale redukcja ilości równocześnie pobieranych danych może ten czas wydłużyć. Jak opisano w podrozdziale „Równoległe pobieranie elementów strony” znajdującym się w rozdziale 6., większa ilość równocześnie pobieranych danych jest dobrym rozwiązaniem, nawet jeśli ma prowadzić do zwiększenia liczby serwerów WWW. W przypadku witryny Google.com na stronie znajdują się tylko dwa elementy. Ponieważ z jednego serwera WWW są pobierane jednocześnie dwa elementy, to użycie jednego serwera minimalizuje liczbę wykonywanych zapytań DNS przy zachowaniu maksymalnej liczby pobieranych jednocześnie danych.

Większość istniejących obecnie stron internetowych posiada dziesięć lub więcej elementów — nie są tak ascetyczne jak Google. Autor sugeruje rozproszenie elementów między co najmniej dwa, ale nie więcej niż cztery serwery. Osiągnięte wyniki będą dobrym kompromisem między ograniczeniem liczby wykonywanych zapytań DNS i zachowaniem wysokiego wskaźnika liczby pobieranych jednocześnie elementów.

Zaletą używania opisanej w rozdziale B funkcji Keep-Alive jest ponowne użycie istniejącego połączenia, co powoduje skrócenie czasu udzielania odpowiedzi poprzez uniknięcie generowania ruchu TCP/IP. Jak przedstawiono w rozdziale, upewnienie się o obsłudze funkcji Keep-Alive przez serwer WWW także zmniejsza liczbę wykonywanych zapytań DNS, zwłaszcza w przypadku użytkowników przeglądarki Firefox.

**Zredukuj liczbę zapytań DNS poprzez używanie funkcji Keep-Alive i mniejszej liczby domen.**