

Analiza technik buforowania w aplikacjach internetowych

Maciej Zakrzewicz, Marek Wojciechowski
Politechnika Poznańska
Instytut Informatyki
{mzakrz,marek}@cs.put.poznan.pl

Abstrakt. Buforowanie obiektów WWW jest jedną z najsilniej rozwijających się technik poprawiania efektywności aplikacji internetowych. Zadaniem buforowania jest przechowywanie kopii obiektów poza miejscem ich publikowania, w celu obsługi żądań użytkowników bez konieczności odwoływania się do źródłowego serwera WWW. Algorytmy zarządzania buforem muszą rozwiązywać szereg problemów technicznych związanych m.in. z aktualizacją kopii, wielowersyjnością i ziarnistością buforowania. W artykule przedstawiono problematykę buforowania WWW oraz dokonano przeglądu architektur i metod obsługi buforów.

1. Wstęp

World Wide Web (WWW) jest usługą sieciową przeznaczoną do rozpowszechniania dokumentów hipertekstowych. Dokumenty, zapisane najczęściej w formacie HTML, XML lub WML, są przesyłane przez serwer WWW na żądanie użytkownika posługującego się przeglądarką WWW. Użytkownik wyraża swoje żądanie w formie adresu URL, zawierającego m.in. adres serwera WWW i ścieżkę dostępu do pliku dokumentu. Dwukierunkowy przesył danych pomiędzy przeglądarką WWW a serwerem WWW odbywa się przy użyciu protokołu HTTP [2]. W komunikacji pomiędzy przeglądarką WWW a serwerem WWW mogą uczestniczyć pośredniczące węzły transportowe, nazywane serwerami proxy.

Dokumenty hipertekstowe mogą posiadać charakter statyczny lub dynamiczny. Dokumenty statyczne to takie, których pełna treść zapisana jest w pliku lub plikach systemu operacyjnego. Dokumenty dynamiczne to dokumenty, których treść jest każdorazowo generowana przez aplikacje współpracujące z serwerem WWW (aplikacje WWW). Najpopularniejsze technologie tworzenia aplikacji WWW to CGI, PHP, PSP, serwlety Java, JSP. Działanie aplikacji WWW może być parametryzowane przy użyciu parametrów przekazywanych przez użytkownika (np. poprzez formularz HTML) bądź zmiennych Cookies (zapisywanych przez serwer WWW na dysku użytkownika). Aplikacje WWW często wykorzystują systemy baz danych jako źródła treści zamieszczonej wewnątrz generowanych dokumentów. Ponieważ dynamiczne dokumenty hipertekstowe wymagają dodatkowego przetwarzania po stronie serwera WWW, to ich udostępnianie wiąże się z powstaniem większych opóźnień niż w przypadku dokumentów statycznych.

Obecne zastosowania aplikacji WWW, obejmujące m.in. spersonalizowaną sprzedaż internetową, narzucają rygorystyczne wymagania w zakresie wydajności obsługi żądań użytkowników. Coraz częściej wymagane jest przetwarzanie setek czy tysięcy żądań przesłania dokumentów dynamicznych na sekundę z czasem odpowiedzi nie przekraczającym dziesięciu sekund. Najprostszym narzędziem, służącym do spełniania tego typu wymagań, było zwykle skalowanie architektury sprzętowej poprzez dodawanie do systemu kolejnych węzłów obliczeniowych. Podstawową wadą takiego podejścia są ogromne koszty inwestycyjne i koszty utrzymania nierzadko setek węzłów.

Alternatywnym sposobem na poprawę wydajności aplikacji WWW jest buforowanie najczęściej rozpowszechnianych dokumentów. Buforowanie polega na zapisywaniu treści dokumentów statycznych lub dynamicznych poza miejscem ich pochodzenia: na komputerze użytkownika lub na serwerze proxy. Dzięki buforowaniu, wyrażone przez użytkownika żądanie pobrania dokumentu

może być zrealizowane poprzez odczyt gotowego pliku, znajdującego się fizycznie bliżej użytkownika, co pozwala zredukować opóźnienia i natężenie transmisji. Buforowanie pozwala zdecydowanie niższym kosztem osiągnąć podobną poprawę efektywności jak w wyniku skalowania architektury sprzętowej.

Implementacja buforowania obiektów WWW napotyka na szereg problemów technicznych, obejmujących m.in. kwestie spójności (unieważniania) zawartości, ziarnistości oraz rozróżnialności. Wymaganie utrzymania spójności zawartości bufora oznacza, że przechowywane w nim obiekty są aktualnymi kopiami obiektów oryginalnych. Drugi problem - ziarnistości bufora – wiąże się z umożliwieniem oddzielnego buforowania pojedynczych komponentów dokumentów złożonych. Natomiast problem rozróżnialności wynika z tego, że identyczne adresy URL mogą w rzeczywistości odpowiadać różnym treściom dokumentów.

2. Architektury buforowania

Buforowanie dokumentów hipertekstowych może posiadać charakter **pasyny** lub **aktywny**. Buforowanie pasywnie reprezentuje podejście tradycyjne: w buforze zapisywany jest dokument hipertekstowy, którego jawnie zażądał klient. Buforowanie aktywne polega na zapisaniu w buforze zarówno żadanego dokumentu, jak i dokumentów powiązanych z nim poprzez łączniki (np. znaczniki w HTML), dzięki czemu bufor przygotowuje się już do obsługi przewidywanego następnego kroku użytkownika. Te dodatkowe dokumenty są pobierane w czasie niskiej aktywności przeglądarki lub serwera proxy.

Bufor WWW zlokalizowany może być w różnych punktach architektury sieciowej: po stronie przeglądarki WWW (jeden klient – wiele serwerów), po stronie serwera WWW (wielu klientów - jeden serwer) lub na serwerze proxy (wielu klientów – wiele serwerów). Bufor może być **scentralizowany**, czyli umieszczony w jednej lokalizacji, lub **rozproszony**, o zawartości replikowanej lub podzielonej pomiędzy wiele węzłów. Zarządzanie buforem rozproszonym jest bardziej złożone i wymaga stosowania specjalizowanych protokołów komunikacyjnych jak np. ICP czy CARP.

Protokół ICP (Internet Cache Protocol) pozwala serwerowi proxy otrzymującemu żądanie od użytkownika na przeszukanie zawartości buforów okolicznych serwerów proxy. Jeżeli żądany dokument zostanie znaleziony, wtedy będzie on przesłany między serwerami proxy, a następnie dostarczony użytkownikowi. Rozwiązania bazujące na protokole ICP cechują się słabą skalowalnością oraz tendencją do wprowadzania redundancji, gdyż kopie tego samego dokumentu mogą być przechowywane wielokrotnie.

Protokół CARP (Cache Array Routing Protocol) umożliwia zorganizowanie wielu serwerów proxy w jeden duży bufor WWW. Gdy użytkownik wysyła żądanie dostarczenia dokumentu, wtedy przy pomocy funkcji haszowej dokonuje się wyboru serwera proxy, który będzie odpowiedzialny za realizację żądania. W ten sposób kopie jednego obiektu nigdy nie będą znajdować się w różnych buforach, a rozwiązania bazujące na CARP będą skalowalne.

3. Spójność bufora

Ponieważ źródłowe dokumenty WWW mogą ulegać modyfikacjom, dlatego istnieje ryzyko, iż zawartość bufora WWW przestanie odpowiadać stanowi faktycznemu. Zgodność wersji przechowywanych w buforze kopii obiektów WWW z ich oryginałami określa się mianem **spójności bufora**. W sytuacji idealnej użytkownik oczekuje od bufora dostarczania kopii identycznych z oryginałami. W praktyce wymaganie to bywa jednak rozluźniane poprzez sformułowanie trzech typów spójności bufora: **spójność silna**, **spójność delta** i **spójność słaba**. Spójność silna występuje wówczas, gdy kopia obiektu WWW w buforze jest binarnie identyczna z oryginałem. Spójność delta wymaga, aby kopia w buforze została odświeżona w ciągu maksymalnie *delta* sekund. Spójność słaba występuje w sytuacji, gdy kopia w buforze nie jest

wprawdzie binarnie identyczna z oryginałem, ale odpowiada mu merytorycznie (np. zmienia się tylko sposób sformatowania dokumentu).

W praktyce stosowane są trzy techniki zapewniania spójności bufora: **aktywna, deklaratywna i przez unieważnianie**. Uspójnianie aktywne polega na tym, że podczas obsługi każdego żądania użytkownika, algorytm obsługi bufora komunikuje się ze źródłowym serwerem WWW w celu porównania aktualności kopii obiektu w buforze. Druga metoda, uspójnianie deklaratywne, polega na deklarowaniu czasu ważności dokumentu przez jego twórcę. Gdy czas ten zostanie przekroczony, kopia w buforze uważana jest za niespójną i musi zostać odświeżona (być może z wykorzystaniem uspójniania aktywnego). Ostatnia z metod, uspójnianie przez unieważnianie, zakłada, że kopia obiektu WWW w buforze pozostaje aktualna dopóty, dopóki nie zostanie jawnie unieważniona przez twórcę oryginalnego dokumentu. Realizacja wymienionych technik zapewniania spójności bufora wymaga stosowania zaawansowanych własności protokołu HTTP lub też dodatkowych protokołów specjalizowanych.

Implementacja uspójniania aktywnego korzysta najczęściej z warunkowych komend protokołu HTTP: If-Modified-Since, If-Unmodified-Since, If-Match i If-None-Match, żądających pobrania dokumentu z serwera WWW tylko wtedy, jeżeli jego wersja różni się od posiadanej w buforze. Do stwierdzenia rozbieżności kopii od oryginału służy porównanie daty ostatniej modyfikacji pliku źródłowego (atrybut last-modified) lub porównanie numeru wersji dokumentu (atrybut entity-tag, tzw. ETag). Numer wersji dokumentu, zapisany jako ciąg znaków, jest zmieniany podczas każdej modyfikacji dokumentu źródłowego i gwarantuje niepowtarzalność w obrębie danego adresu URL. W praktyce metody te wzajemnie się wykluczają, a popularniejszą, chociaż nie pozbawioną wad, jest pierwsza z nich – korzystająca z daty ostatniej modyfikacji. Jej główne wady wynikają z trudności synchronizacji zegarów w systemie rozproszonym oraz, mniej istotna, z rozdzielczości opisu czasu w protokole HTTP (1 sekunda).

Uspójnianie deklaratywne wykorzystuje etykiety Expires i Cache-Control, zapisywane w nagłówkach HTTP lub w znacznikach META języka HTML. Etykieta Expires pozwala na określenie daty, po której kopia dokumentu w buforze stanie się nieaktualna, np. „Expires: Fri, 30 Oct 1998 14:19:41 GMT”. Wówczas, niezależnie od tego, czy dokument źródłowy faktycznie uległ jakiegokolwiek modyfikacji, nastąpi odświeżenie jego kopii w buforze. Druga ze wspomnianych etykiet, Cache-Control, umożliwia deklarowanie czasu życia kopii w buforze, liczonego od momentu jej załadowania, np. „Cache-Control: s-maxage=3600”. Etykieta Cache-Control pozwala też na bardziej zaawansowane sterowanie buforowaniem – jej funkcje przedstawiono w Tabeli 1.

Tabela 1. Atrybuty etykiety Cache-Control w protokole HTTP 1.1

Atrybut	Znaczenie
max-age=n [sekund]	Określa maksymalny czas ważności obiektu; podobne do Expires; określa liczbę sekund, przez jaką obiekt pozostanie ważny
s-maxage=n [sekund]	Podobny do max-age, ale stosowany wyłącznie przez bufor proxy (shared)
public	Zaznacza jako buforowalny obiekt, który normalnie nie mógłby być buforowany np. z powodu wymogu uwiarygadniania
private	Treść jest przeznaczona dla indywidualnego użytkownika i nie powinna być buforowana przez serwer proxy; może być buforowana przez przeglądarkę
no-cache	Wymaga każdorazowej weryfikacji buforowanej kopii przez serwer WWW

no-store	Zakaz buforowania zawartości obiektu
must-revalidate	Zakazuje serwerowi proxy udostępniania unieważnionych kopii obiektów w sytuacjach np. awarii połączenia z serwerem czy niskiej przepustowości sieci
proxy-revalidate	Podobne do must-revalidate, ale stosowane wyłącznie przez bufor proxy

Najnowsza z przedstawianych technik - zapewnianie spójności bufora przez unieważnianie - opiera się na założeniu, że to twórca dokumentu (statycznego lub dynamicznego) potrafi najlepiej określać moment jego dezaktualizacji. Przykładowo, kiedy pojawia się nowy rekord w tabeli bazodanowej, na której oparto treść dynamicznego dokumentu hipertekstowego, wtedy odpowiedni wyzwalacz (trigger) mógłby przesłać do bufora żądanie unieważnienia tego dokumentu, w ten sposób sprawiając, że podczas obsługi kolejnego żądania, treść dokumentu zostanie pobrana bezpośrednio z serwera WWW. Mechanizmy komunikacji z buforem WWW w celu unieważniania jego zawartości dostępne są m.in. w specyfikacji ESI Invalidation Protocol [4] lub w propozycji Web Cache Invalidation Protocol (WCIP) [6].

Protokół ESI Invalidation Protocol wymaga, aby źródło danych dla dokumentów dynamicznych przesyłało komunikat HTTP POST do oprogramowania zarządzającego buforem, powodując unieważnienie kopii dokumentów zależnych. Przykładowy komunikat unieważniający został przedstawiony poniżej:

```

1      POST /x-invalidate HTTP/1.0
2      Authorization: Basic aW52YWxpZGF0b3I6aW52YWxpZGF0b3I=
3      Content-Length: 217
4
5      <?xml version="1.0" ?>
6      <!DOCTYPE INVALIDATION SYSTEM "invalidation.dtd">
7      <INVALIDATION VERSION="WCS-1.0">
8      <OBJECT>
9      <BASICSELECTOR URI="/cache.htm" />
10     <ACTION />
11     </OBJECT>
12     </INVALIDATION>

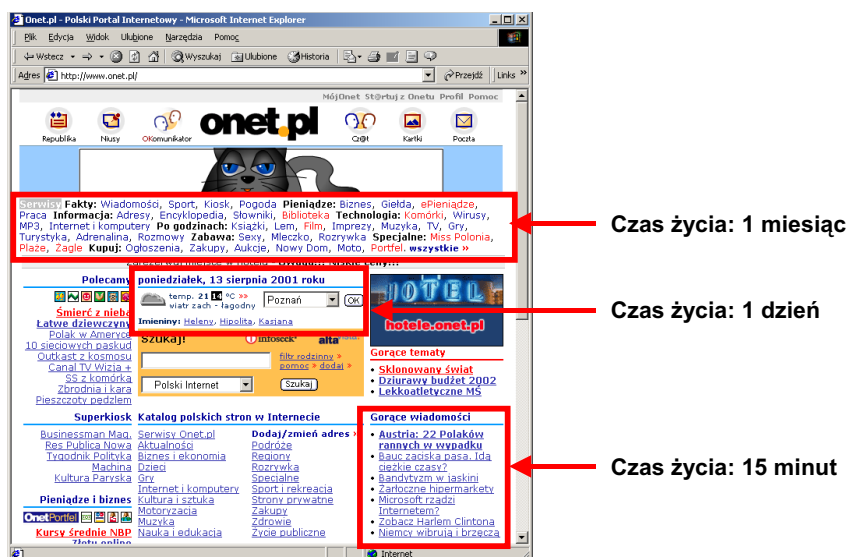
```

W wierszu 1 znajduje się nagłówek komunikatu HTTP POST (/x-invalidate jest zastępowane etykietą zależną od implementacji oprogramowania bufora). Wiersz 2 zawiera nazwę i hasło użytkownika uprawnionego do unieważniania zawartości bufora, zakodowane w formacie Base64. Wiersz 3 określa rozmiar całego komunikatu. Wiersz 4 jest separatorem. W wierszu 5 znajduje się znacznik rozpoczynający dokument XML, po którym w wierszu 6 znajduje się deklaracja typu dokumentu. Wiersze 7-12 opisują żądanie unieważnienia znajdującej się w buforze kopii dokumentu „/cache.htm”.

4. Ziarnistość buforowania i rozróżnialność wersji

Gdy kopia dokumentu w buforze WWW staje się niespójna z oryginałem, powinna zostać odświeżona. Nie zawsze jednak niezbędne jest odświeżenie całego dokumentu: często dokumenty posiadają charakter złożony i składają się z komponentów o różnym czasie życia. Istnienie możliwości odświeżania wyłącznie pojedynczego komponentu pozwala obniżyć koszt wykonania całej operacji. Jednak aby było to możliwe, twórca dokumentu musi w stosowny sposób deklarować granice takiego podziału, posługując się np. ramkami HTML lub znacznikami ESI. Przykład

złożonego dokumentu hipertekstowego zawierającego komponenty o różnych okresach zmienności (czasach życia) przedstawiono na Rysunku 1. Część dokumentu prezentująca aktualne wiadomości prasowe odświeżana będzie najczęściej (minuty), część zawierająca prognozę pogody pozostanie prawdopodobnie aktualna przez cały dzień, natomiast blok łączników do innych sekcji serwisu będzie modyfikowany dopiero podczas przebudowy systemu. To oznacza, że gdyby jednostką buforowania był cały dokument, wtedy z dużą częstotliwością z serwera WWW pobierane byłyby wszystkie jego komponenty, co w efekcie pogarszałoby wydajność bufora (wzrost opóźnienia).



Rys. 1. Dokument hipertekstowy zawierający komponenty o różnym czasie życia

W związku z tym, że znaczny procent wszystkich dokumentów udostępnianych w sieci Internet stanowią dokumenty dynamiczne, techniki buforowania WWW muszą pokonywać dodatkowy problem: problem rozróżnialności dokumentów. Problem rozróżnialności wynika z tego, że w przypadku dokumentów dynamicznych, adres URL nie zawsze jednoznacznie identyfikuje treść dokumentu. Oprócz adresu URL, zawartość dokumentu uzależniona może być od takich czynników jak: parametry wywołania przekazywane przez użytkownika (dotyczy metody POST), zmienne Cookies przesyłane przez przeglądarkę WWW, zmienne środowiskowe CGI, itp. Przykładowo, w zależności od konfiguracji przeglądarki WWW, użytkownicy mogą pod tym samym adresem URL otrzymywać dokumenty WWW w różnych wersjach językowych (np. polskiej lub angielskiej). Uwzględnianie takich czynników przez algorytm buforowania WWW pozwala zapewnić poprawne działanie bufora dla dokumentów dynamicznych.

Z podobnym problemem stykamy się w przypadku parametryzowanych dokumentów dynamicznych, w których wartość przekazanego parametru służy wyłącznie umieszczenia jej w treści zwracanego dokumentu. Przykład takiego dokumentu został przedstawiony na Rysunku 2, gdzie nazwisko użytkownika przesyłane w zmiennej Cookie jest wprost zamieszczane w dokumencie i stanowi jedyną różnicę pomiędzy dokumentami dostarczonymi różnym użytkownikom. O ile umieszczanie w buforze tylu kopii tego dokumentu, ilu istnieje różnych nazwisk użytkowników wydaje się bezcelowe, o tyle praktycznym byłoby, aby algorytm buforowania WWW potrafił samodzielnie podstawić podaną zmienną Cookie do buforowanego szablonu dokumentu i w ten sposób operować na pojedynczej kopii. Zastosowanie takiego mechanizmu wymaga udziału projektantów, którzy muszą przygotować i odpowiednio oznakować punkty podstawienia wartości parametrycznych (np. korzystając z ESI).



Rys. 2. Przykład prostego dokumentu personalizowanego

ESI (Edge Side Includes) [3] jest prostym językiem znaczników, służącym do opisu fragmentów dokumentów hipertekstowych, które mogą być buforowane niezależnie, a następnie kompletowane na poziomie bufora WWW. Istnieje również propozycja standardu JESI (Java Edge Side Includes) [5], służącego do strukturalizacji dokumentów dynamicznych generowanych przez aplikacje JSP (Java Server Pages). Poniżej przedstawiono przykład prostego dokumentu hipertekstowego, wykorzystującego znaczniki ESI w celu poprawy skuteczności buforowania.

```

1      <HTML>
2      <BODY>
3      <H2>Witamy na naszym przykładowym portalu</H2>
4      <!--ESI
5      <H3>Prognoza pogody:</H3>
6      <ESI:INCLUDE SRC="/weather.jsp?sessID=$(QUERY_STRING{sessID})" />
7      <H3>Najnowsze wiadomości:</H3>
8      <ESI:INCLUDE SRC="/news.jsp?sessID=$(QUERY_STRING{sessID})" />
9      <ESI:TRY>
10     <ESI:ATTEMPT>
11     <H3>Informacje giełdowe:</H3>
12     <ESI:INCLUDE SRC="/stocks.jsp?sessID=$(QUERY_STRING{sessID})" />
13     </ESI:ATTEMPT>
14     <ESI:EXCEPT>
15     Informacje giełdowe są chwilowo niedostępne. Przepraszamy.
16     </ESI:EXCEPT>
17     </ESI:TRY>
18     <ESI:CHOOSE>
19     <ESI:WHEN TEST="$(QUERY_STRING{type}) == 'Sport'">
20     Informacje sportowe:
21     <ESI:INCLUDE SRC="/sport.jsp?sessID=$(QUERY_STRING{sessID})" />
22     </ESI:WHEN>
23     </ESI:CHOOSE>
24     -->
25     </BODY>
26     </HTML>

```

W wierszu 6 dołączany jest dokument o adresie URL "weather.jsp". Jest to dokument dynamiczny, generowany przez aplikację JSP, pobierającą jeden parametr wywołania: sessID.

Wartość tego parametru jest automatycznie podstawiana przez algorytm zarządzający buforem WWW, na podstawie otrzymanego parametru wywołania o takiej samej nazwie. Dołączany dokument jest niezależnie buforowany, w związku z czym może posiadać inne wymagania w zakresie spójności. W wierszach 9-17 w podobny sposób dołączany jest dokument "stocks.jsp", lecz tym razem przewidziano możliwość wystąpienia problemów połączeniowych - jeżeli dokument ten będzie nieosiągalny, po prostu nie zostanie włączony do odpowiedzi. Wiersze 18-23 ilustrują zastosowanie instrukcji warunkowej, pozwalającej opisywać sposób personalizacji dokumentów. Najważniejsze znaczniki ESI zostały opisane w Tabeli 2.

Tabela. 2. Wybrane znaczniki ESI

Znacznik	Opis
<esi:include>	Zamieszcza oddzielnie buforowany fragment
<esi:choose>	Instrukcja warunkowa, np. na podstawie cookies lub zmiennych środowiskowych (CGI)
<esi:try>	Wykonuje operacje zastępczą, jeżeli operacja główna nie może zostać pomyślnie wykonana
<esi:vars>	Umożliwia podstawianie zmiennych środowiskowych
<esi:remove>	Określa treść, jaka zostanie przekazana do przeglądarki WWW, jeżeli nie odbędzie się przetwarzanie ESI
<!--esi ... -->	Treść przetwarzana przez ESI, ale ukryta przed przeglądarką WWW
<esi:inline>	Zamieszcza oddzielnie buforowany fragment, którego treść znajduje się w bieżącym dokumencie
<esi:comment>	Komentarz

5. Podsumowanie

W artykule omówiliśmy podstawowe problemy buforowania dokumentów hipertekstowych oraz przedstawiliśmy techniki ich rozwiązywania. Zwróciliśmy uwagę na to, że w celu uzyskania wysokiej skuteczności buforowania, niezbędne jest jego uwzględnianie już na etapie projektowania systemu internetowego. Zastosowanie etykiet HTTP, znaczników ESI oraz protokołów unieważniania pozwala twórcom systemów internetowych na strukturalne przeniesienie obciążenia serwerów WWW na serwery buforowe proxy, dzięki czemu wydajność systemu może zostać podniesiona stosunkowo niskim kosztem.

Przykładem zaawansowanego technicznie dedykowanego serwera proxy jest Oracle9iAS Web Cache [1]. Jest to samodzielny serwer, instalowany pomiędzy aplikacyjnymi serwerami WWW a użytkownikami, pozwalający na buforowanie pełnych dokumentów statycznych i dynamicznych, buforowanie fragmentów dokumentów opisanych w języku ESI oraz programowe unieważnianie kopii obiektów w buforze. Ponadto, Oracle9iAS Web Cache oferuje dynamiczne równoważenie obciążenia źródłowych serwerów WWW, wykrywanie i maskowanie awarii oraz automatyczną kompresję przesyłanych obiektów.

Bibliografia

1. „Oracle9iAS Web Cache”, Technical White Paper, June 2001, Oracle Corporation

2. „Hypertext Transfer Protocol -- HTTP/1.1”, RFC 2616, Fielding, et al.
3. „ESI Language Specification 1.0”, Tzmelzon, Wehl, Jacobs, Akamai Technologies, Oracle Corporation
4. „ESI Invalidation Protocol 1.0”, Tzmelzon, Wehl, Jacobs, Akamai Technologies, Oracle Corporation
5. „JESI Tag Library 1.0 Specification: Tags for Edge-Side Includes in JSP”, Basu et al.
6. „WCIP: Web Cache Invalidation Protocol”, Internet Draft, Cao, Dahlin