**Същност на оптимизацията на WEB сайтове**

1. Увод
2. Обща информация за сайтовете
3. Методи и техники за оптимизация - производителност, оптимизация за търсещи машини и подобряване на потребителското преживяване, Индексиране, Предотвратяване на индексирането, увеличаване на видимостта, основни направления, техники за оптимизация, оптимизация на сайт.
4. Предимствата на HTML5 и с какво те се отличават от HTML4.
5. Оптимизация на ASP.NET страница.

**Увод**

През 1984 в CERN Тим Бърнарс-Лий разглежда проблемите на организацията, свързани с представянето и обмена на информация. Физици от целия свят имат нужда да обменят данни, но не разполагат нито с общи машини, нито с общ софтуер за целта. През 1989-та Лий написва предложение за създаване на база данни, работеща чрез ръчно въвеждани линкове. През 1990-та година Тим Бърнарс-Лий и Роберт Кайо реализират първата успешна комуникация между клиент и сървър, базирана на HTTP протокола.

Повратна точка в еволюцията на мрежата е представянето на графичния браузър Мозаик през 1993-та, който позволява едновременно представяне на медия и текст - нещо необичайно до този момент. Година по-късно на първата международна конференция, посветена на глобалната мрежа от CERN обявяват, че всеки може да използва кода и протокола безвъзмездно и поставят началото на световната мрежа и Интернет. По този начин мрежата се отваря към публичния сектор и се превръща в инструмент за споделяне на информация, комуникация и търговия в реално време. Същата година е пусната и първата интернет търсачка W3Catalog, а две години по-късно стартира и Yahoo.

Скоро става ясно, че публичното представяне в интернет е необходимост, а не екстра. Заедно с развитието на технологиите нарастват и възможностите на мрежата, а съдържането се обогатява с все повече медия. Успоредно с това се наблюдава и развитие на електронните услуги, както от страна на частни доставчици, така и реализирани от правителствени и международни институции.

Изследванията показват, че в края на 2011 година съществуват над половин милиард сайтове, като близо 50% от тях са ново добавени, явно доказателство за лавинообразния растеж на мрежата. Същевременно броят на Интернет потребителите надвишава 2,1 милиарда. В края на същата година Google обявяват, че средният брой търсения на ден надвишава 4.7 милиарда. Тези данни са показател за силната конкуренция, на която са подложени участниците в уеб пазара, заради което се налага оптимизацията на Уеб сайтовете.

В книгата си “Оптимизация на уеб страници” Андрю Кинг дава следното определение - “Оптимизацията на уеб страниците цели да максимизира(обикновено във финансов аспект) възвръщаемостта на инвестицията в уеб страницата”[1]. В резюмето на “Още по-бързи уеб сайтове” се набляга на факта, че “Оптимизацията е критична за успеха на всеки уеб сайт, а приложенията достигат границата на възможностите на браузърите, чрез нарастващо количество съдържание и употреба на AJAX”[2]. Методите за оптимизация могат да се разделят основно в 3 категории - производителност, оптимизация за търсещи машини и подобряване на потребителското преживяване. Не съществува възможност за точно метрично определяне на ефективността на отделните методи поради силната зависимост между трите категории, но като резултат от практически опит могат да бъдат изведени методики за комбинирано използване на различни методи с цел постигане на оптимален резултат с минимална инвестиция.Актуалността на темата произлиза от постоянно нарастващия брой уеб страници и приложения и засилващата се конкуренция в уеб пространството.

**Обща информация за сайтовете**

Сайтовете представляват набор от свързани помежду си страници, обединени под общо име(домейн) и достъпни през Интернет или локална мрежа. Информацията в тях е достъпна през HTTP протокол или при нужда от криптиране на информацията през HTTPS. За да се разглеждат страниците, се използват програми, наречени уеб браузъри, които се грижат както за трансфера на данните, така и за тяхното представяне. От гледна точка на начина им на работа сайтовете могат да се разделят на два основни типа - статични и динамични. Статични се наричат сайтове, чиито страници са реализирани във формат, четим от уеб браузърите. Обикновено това е HTML и по-рядко XML формат. При динамичните сайтове имаме код, който се изпълнява на сървъра и генерира страници в удобен за браузъра формат. В зависимост от мястото на изпълнение на кода приложението може условно да се раздели на две основни части – бекенд(backend), това е частта от кода, която се изпълнява на сървъра и фронтенд(frontend) –код, който се изпълнява в браузъра, наричан още „клиентска част“.

При клиентската част не съществува широк избор от възможни технологии за разработка - страниците са базирани на HTML (HyperTextMarkupLanguage) езика, като стиловите особености се определят с помощта на CSS (CascadeStyleSheets). За създаване на интерактивни страници и улеснен диалог с потребителя се използват технологии като Flash (в процес на затихване) и Silverlight – алтернатива на Flash, създадена от Майкрософт, но най-широко приложение намира езикът JavaScript.

Противно на ограничение в клиентската част, изборът на езици и платформи за сървърната такава е практически неограничен. Сред най-използваните езици са PHP, C# и платформата .NET, Java и C++, но множество не толкова популярни доскоро езици като Python и Ruby намират все по-широко приложение в уеб индустрията. Интерес представлява фактът, че през последните години, благодарение на платформи като Node.JS споменатият JavaScript започва да се налага и като език за сървърно програмиране.

С развитието на технологиите в областта на мобилните устройства те стават все по-значим фактор в процеса на изграждане и оптимизация на уеб страниците. Особеностите като понижена скорост на обмен на данни и разлика в размера и формата на дисплея, рефлектират върху методите за оптимизация. Налага се създаване на все по-компактни като размер страници и интерфейс, удобен за разглеждане както на мобилни устройства, така и на настолни или преносими компютри.

Заедно с това развитие започва разделение между браузърите, като всеки от тях представя нови възможности или притежава различно поведение при определени условия. Това затруднява създаването на стабилно работещи сайтове върху множество платформи и води до необходимост от изработване на обща спецификация характеризираща особеностите на езика.

В резултат на това, през 2004-та година, 4 години след завършването на HTML 4.01 стандарта, работната група по уеб и хипертекст технологии и приложения (WHATWG) започва работа по следващата версия на езика[33]. Същевременно Консорциума за Глобалната мрежа W3C работи по XHTML 2.0 близък до HTML език, включващ част от ограниченията заложени в XML. През 2009-та година W3C прекратяват работата по XHTML 2.0 и решават заедно с WHATWG да се фокусират върху петата версия на езика – HTML 5. Макар официалната дата за завършване на работата по спецификацията да е обявена за 2014 година, повечето от компаниите, занимаващи се с разработка на браузърни приложения отдавна са приели стандарта, а някои от тях са и активни участници в разработката и специфицирането му. За съжаление дългият процес на разработка и неяснота по отношение на някои особености на езика се отразява на браузърите. Резултатът е видим– липсваща и/или непълна поддръжка на част от функционалностите между отделните браузъри. Това от своя страна дава нов тласък на индустрията и започва разработка на механизми, основно с помощта на JavaScript, за симулация на липсващите функционалности. Тъй като не всички възможности на езика могат да бъдат симулирани, потребители с по-стари браузъри биват лишени от тях, като в масовия случай става въпрос за функционалности, свързани с дизайна и представянето, а не конкретно с работата на сайта. Макар при последните версии на основните модели браузъри поддръжката да обхваща голяма част от нововъведения в езика, проблемът с по-старите версии на приложенията все още стои пред разработчиците. В това отношение страницата <http://caniuse.com/> представлява ценен ресурс за систематизирана детайлна информация за отделните функционалности и поддръжката им в различните браузър и техните версии, както и относителния пазарен дял на съответната версия. По същия начин стои и въпросът с нововъведенията, предлагани от модулите, дефинирани в третата версия на езика за каскадни стилове – CSS3.

Основните промени в HTML5 са в две направления – създаване на документи с по-добра семантика и подобряване на работата при устройства с по-малка изчислителна мощ, като мобилни телефони и таблети. Новите семантични тагове като <section>, <article>, <header> и др. позволяват по-описателно, маркиране на документа правейки го по-лесен за разбиране от търсещи машини и софтуери за обработка и анализ, което от своя страна води до подобряване на търсенето и показване на по-релевантни и подходящи резултати за потребителите. Въведени са и няколко нови тага, свързани с употребата на мултимедийно съдържание като видео и аудио файлове, както и нов таг (<canvas>), позволяващ визуализация и модификация на графични елементи с помощта на скриптове.

**Методи и техники за оптимизация**

Методите за оптимизация могат да се разделят основно в 3 категории - производителност, оптимизация за търсещи машини и подобряване на потребителското преживяване. Не съществува възможност за точно метрично определяне на ефективността на отделните методи поради силната зависимост между трите категории, но като резултат от практически опит могат да бъдат изведени методики за комбинирано използване на различни методи с цел постигане на оптимален резултат с минимална инвестиция. Актуалността на темата произлиза от постоянно нарастващия брой уеб страници и приложения и засилващата се конкуренция в уеб пространството.

1. **Производителност**

За подобряване на призводителността на Уеб сайта трябва да се минимизира размерът му възможно най-много. Първата стъпка е максималното оптимизиране на изображенията, които изискват наистина добра оптимизация, за да се постигнат добри резултати. Един от утежняващите фактори за бързо зареждане са именно изображенията. Оптимизацията става чрез намаляване на качеството на изображението до такава степен, че да не се улови разлика в качеството с просто око. За целта Photoshop предлагат опция „Save for web...“ от където може да се намали качеството на изображението и да се запази изображението във формат Base JPEG или Progressive JPEG. На много места можете да срещнете мнението, че трябва да използвате прогресив JPEG за постигане на по-добро потребителско преживяване, но истината е, че е добре да се използват само там, където е първоначалното визуално взаимодействие на потребителя със страницата. Казано с други думи, ако там където се зарежда първоначалната видима част от страницата, има изображение – използваме прогресив JPEG, а по-надолу в страниците, изображенията са в обикновен /Base/JPEG формат. Компресията на Base JPEG изображенията, е с около 5 % по-добра, спрямо Progressive JPEG при едно и също оптимално цветово разпределение и контрастни нива. Разликата между двата формата е, че при Base JPEG изображението се зарежда постепенно отгоре надолу, а при Progressive JPEG се изобразява цялото изображение замъглено и постепенно се избистря. На Фиг. 1 е показана разликата.

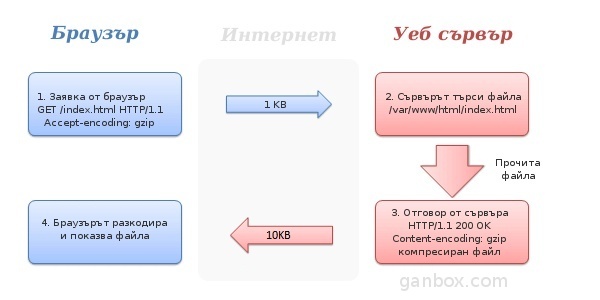


Фиг. 1

Друга нова и изключително добра практика е използване на WebP формат за изображения. Формата е разработен от Google и съвместява два вида компресия – lossy и lossless, които осигуряват 26% по-малък размер в сравнение с PNG изображенията, и между 25 и 34% по-малък размер в сравнение с JPEG формата. Форматът е Open Source базиран, а създателите му са на мнение, че може да намали с повече от 1/3 обема на графичните изображения в нета без видима промяна в качеството. Според статистиката 60-70 % от информацията, която се обменя при разглеждане на произволен сайт е в изображенията – целта на формата е да намали обема на изображението, като с това се намаля времето за трансфер, но без това намаляне на обема да влияе на качеството на изображението. Подобна оптимизация би подобрила цялостния екпириънс при браузване, особено при набиращия все по-голяма популярност мобилен интернет. За момента (към 07.2015) форматът се поддържа само от Chrome 9, а за създаването му може да се използва Photoshop плъгин.

Втората важна стъпка е оптимизирането на Джава скриптове (Javascripts). Джава скриптовете са ресурс, който сериозно може да забави възпроизвеждането на една интернет страница. Причините за това, са няколко основни. Първата от тях е, че докато интернет страницата зарежда, един или няколко джава скрипта, могат да блокират рендирането на страницата докато не се заредят. Доста често, особено при използване на множество плъгини и аддон-и, в много готови системи за управление на съдържанието се появява казусът с наличието на много (като брой) скриптове, които се съревновават един с друг за приоритет при зареждането на страницата. През цялото това време, страницата ще чака изтеглянето, разкомпресирането и изпълнението им. Проверка дали на сайта има блокиращи скриптове може да се направи чрез инструмента на Google, Pagespeed Insight, който ще каже и кои точно са файловете които блокират изобразяването. Поради тази причина – блокирането на рендеринга, има два основни метода за работа с джава скриптовете. Първият е асинхронен метод за зареждане, при който скриптовете се зареждат заедно със съдържанието на страницата, но не блокират изпълнението и, а вторият е Deferring, при който скриптовете започват да се зареждат след съдържанието на страницата. И двата метода си имат минусите и плюсовете, като – и при двата е нужно да се подходи с повишено внимание.

Трета стъпка е да се използва gzip компресия. Формулата е проста: по-малък файл = по-малък трафик = по-бързо зареждане на страницата => всички са доволни: потребителя защото отваря сайта бързо и собственика на сайта, защото плаща по-малко за трафик (ако има ограничение на трафика). Тънкият момент при тази комуникация е браузърът и сървърът да се договорят, че ще се пренася компресирано съдържание за конкретния файл. Споразумението протича в две части: – ﻿браузърът изпраща хедър ﻿“Accept-Encoding: gzip, deflate“, с което казва на сървъра, че може да приема компресирани файлове, по gzip компресия или по ﻿deflate компресия; – сървърът връща хедър „Content-Encoding: gzip“, ако съдържанието наистина е компресирано. Браузърът не може да задължи сървъра да върне компресирано съдържание, той само изпраща молба. Ако браузърът е казал, че приема компресирани файлове, но сървърът не върне „Accept-encoding“ хедър, това означава, че сървърът по някаква причина не е компресирал съдържанието. На Фиг.2 е показан разговорът между браузър и сървър, когато се ползва компресиране на данните.



Фиг.2 Разговорът между браузър и сървър, когато се ползва компресиране на данните.

1. Браузър: Здрасти domain.com, искам да получа index.html, мога да получа компресирана версия, ако има.

2. Сървър: Добре, чакай да видя дали има такъв файл и виждам, че може да приемаш компресирани файлове.

3. Сървър: OK намерих файла (response code 200 OK). Компресирам и изпращам файла.

4. Браузър: Супер! Само 10 KB е, ще го разкомпресирам и покажа на потребителя.

1. **Оптимизация за търсещи машини**

Оптимизацията за търсещи машини протича в следните етапи

**Индексиране**

Водещите търсачки като Google, Bing и Yahoo! използват софтуер, познат като роботи (robots), който обхожда уеб страниците и ги индексира чрез специални алгоритми. Роботите вземат предвид различни фактори при обхождане на сайтовете, така че не всяка страница бива намерена. Те следят за най-различни показатели, като например разстоянието от дадена страница до главната директория на сайта. Google предлага инструменти за програмисти (Webmaster Tools), с които може да се изработи карта на сайта (XML Sitemap), за да се гарантира, че всички страници ще бъдат намерени, особено тези, които не са откриваеми посредством автоматизирания метод на търсене. Някои от търсачките предлагат платена услуга, която гарантира присъствието на изпратения сайт в търсачката. Таксата за услугата се определя на база на реализирани посещения или т.нар. система Pay per Click във връзка със зададени от рекламодателя ключови думи. Тази услуга обикновено гарантира включване на сайта в базата данни, но не и специфично класиране в резултатите от търсенията. Освен автоматичното индексиране, всеки сайт може да бъде регистриран и в някой от указателите (директориите) като Yahoo и Open Directory Project, но при това се изисква ръчно подаване на информация.

**Предотвратяване на индексирането**

За да се предотврати намирането на дадена страница, така наречените роботи на търсещите машини могат да бъдат инструктирани да не достъпват определени файлове или директории. Това може да се направи от уеб администраторите на сайта чрез подаване на инструкции към роботите, зададени предварително в специфичен файл, намиращ се в главната директория на домейна. Освен това, на самата страница могат да бъдат поставени етикети (meta tags), които изрично да изключват от базата данни на търсачките. Когато търсеща машина посещава сайт, файлът с инструкциите, намиращ се в главната директория на домейна, бива прочетен първи, като това предопределя кои от страниците не трябва да бъдат обхождани. Страници, които обикновено имат забрана за обхождане са такива, които съдържат лична информация, като потребителски профил или пазарска количка в онлайн магазин. През март 2007 г. Google предупреждава уеб администраторите, че трябва да предотвратят индексирането на резултатите от вътрешните търсения, тъй като те се считат за спам и са против правилата, поставени от Google.

**Увеличаване на видимостта**

Има много различни методи, с които да се увеличи видимостта на уеб страница в резултатите от търсене. Кръстосаните връзки (cross linking) към най-важните страници на същия сайт може да подобри нейната видимост. Писане на съдържание, което включва често търсена ключова фраза, така че да бъде свързана с широк кръг от заявки за търсене, потенциално ще увеличи трафика. Честото обновяване на информацията може да накара търсещите роботи да се връщат по-често, като така ще придават допълнителна тежест на сайта. Добавянето на подходящи заглавия, както и ключови думи в етикетите на уеб страницата, може да подобри състоятелността на сайта в базата данни на търсачките.

**Основни направления**

При изчисляване на релевантността (степен на съответствие) на запитването, машината за търсене (търсачка, Search Engine) отчита такива параметри на сайта като:

* плътност на ключовите думи (фрази): Сложните алгоритми на съвременните търсачки позволяват провеждане на семантичен анализ на текста с цел филтриране на евентуален спам - когато ключовата дума/фраза се среща прекалено често в страницата (над определен процент от целия текст);
* свързани думи: Google Търсене все повече набляга на цялостното съдържание на текста и се опитва да индексира страниците семантично, т.е. по смисъл. Силно влияние има наличието на т. нар. свързани думи в текста. Например, страница за български сувенири е добре да включва информация за България, за български традиции, известни етнографски местности и др.;
* индекс на цитиране на сайта или количество и качество на ресурсите с хипервръзка към сайта: Много от търсачките не отчитат взаимните връзки между сайтовете, затова се предпочитат хипервръзките да бъдат от тематично сродни сайтове;
* данни за потребителското поведение: Много важен критерий за качеството на един сайт е доколко потребителите са доволни от сайта. Съществуват измерители на потребителското удовлетворение, които се използват от търсачките. Най-ясно разбираемите показатели са: време прекарано на сайта, брой разгледани страници, повече от едно посещение на сайта, граматично и правописно правилен текст.

Оптимизацията на даден сайт включва работа с разнообразни вътрешни фактори — привеждане на текста на сайта в съответствие с избраните ключови думи (фрази), повишаване на качеството и количеството текстове на сайта, оптимизация на структурата, навигацията и вътрешните връзки, а също и с външни фактори — обмен на хипервръзки, регистрация в каталози и директории и други техники, целящи повишаване на броя и качеството на хипервръзките, насочени към конкретния ресурс.

**Техники за оптимизация**

SEO техниките за оптимизация може да се разделят на три основни категории. Техниките, която търсещите машини препоръчват като част от дизайна на сайта са така наречените „Бяла“ оптимизация. Съществува втори тип, които не са препоръчителни - „Сива“ оптимизация", и такива, които са в директен разрез с правилата на търсачките - „Черна“ оптимизация. По този начин търсачките се опитват да намалят влиянието на тези, които злоупотребяват и манипулират класирането, често чрез строго забранени от търсачките методи.

* «Бяла» оптимизация: При тази оптимизация се цели резултатите, които се извеждат да се запазят по-дълго във времето, като се изключва използването на забранени методи за рекламиране и класиране на ресурси. Следват се правилата на големи търсачки като Google, Yahoo, Bing и др., и не се позволяват отклонения от тях. Важно е да се отбележи, че тези правила не са записани изрично, а по-скоро са като правилни насоки според съответната търсачка. Следването на тези насоки предполага създаване на съдържание в полза на потребителите, а не такова което цели подвеждането на търсещите машини и алгоритми с цел по-високо индексиране.
* «Сива» оптимизация: Използват се методи и техники, които не са желателни за използване при търсачките — използване на „ферми за хипервръзки“, закупуване на връзки от високо класирани ресурси.
* «Черна» оптимизация: Този вид оптимизация се опитва да подобри класирането, по начини, които включват измами. Примери за черна SEO техника са използването на скрит текст или текст с цвят, подобен на фоновия, също така и позициониране на елементи извън екрана. Друг метод е да се извеждат различни страници в зависимост от това дали се посещава от човек или от търсеща машина (т.нар. прикриване)или използването на т.нар. „doorway — страници“ и ресурси, създадени специално за роботите и паяците на търсачките. Методите за оптимизация и класиране на уеб сайт с използване на doorway са нарушение на правилата на търсачките.

Наказанието за използване на такива методи от черната оптимизация може да е чрез намаляване на класирането или цялостното му изключване от базата данни на търсачката. Прилагането на тези наказания може да станат автоматично, чрез включените в търсачките алгоритми или чрез ръчно преглеждане на сайта. Пример за това е премахването през февруари 2006 г. от Google на страниците BMW Германия и Ricoh Германия, поради използването на подвеждащи търсещата машина практики. И двете компании, обаче, бързо се извиняват и поправят инкриминираните си страници, като така възстановява позициите си в Google.

**Оптимизиране на сайт**

Препоръчва се оптимизирането да се извършва в най-големите търсачки: Google, Yandex, Bing или Yahoo!. Голяма част от останалите търсещи системи, използват информация от тях. Етапите на оптимизацията условно се делят на външни и вътрешни (спрямо оптимизирания сайт) фактори:

***Вътрешна оптимизация***

Чрез вътрешната оптимизация се цели постигането на повече посещения, посредством търсения в търсачки като Google, Yahoo! и др. по ключова дума или фраза. Към нея влизат валидният (според W3C) HTML и CSS код, добре структурираното съдържание, както и правилно поставените заглавия и подзаглавия. Създаването на карта на сайта (sitemap) е още един похват спомагащ за този тип оптимизация. Тази карта служи за отбелязване на връзките в един сайт, като така дава възможност на „паяците“, които използват търсещите машини, да се насочат към вътрешните страници на дадения сайт. Те са особено полезни при „по-дълбоки“ сайтове, съдържащи много страници и категории. По-големите търсещи машини предлагат инструменти за администратори, като Google Webmaster Tools и Google Analitycs в които може да се регистрира или добави сайт в базата данни на търсачката, като така се гарантира обхождането му. Преобразуването на URL-а (адреса на страницата) в четим или „приятелски“ формат е още една стъпка към по-добра вътрешна оптимизация. Пример за това е: www.example.com/#psj=1&q=page2, което може да изглежда по четим за човека начин като: www.example.com/Добре\_дошли. Използването на такива адреси създава допълнителни ключови думи за съдържанието на страницата, които биват индексирани от търсачките.

***Вътрешни фактори***

* Определяне на търсената аудитория и организиране на съдържанието. Ядрото на сайта се съставя като се вземат предвид следното: съдържание на сайта, сайтове-конкуренти, статистика по търсени думи (примерно Adwords на Google).
* Анализи на конкурентни страници — мястото им в търсачките, структурата и съдържанието им.
* Анализ на страниците.

***Външна оптимизация***

Външната оптимизация може да се определи като изграждане на онлайн авторитет или репутация на даден сайт. Най-често това се постига чрез поставянето на линкове (хипервръзки) към сайта в други сайтове. Чрез използването на рекламни кампании, които да поставят тези връзки на сайтове с подобно съдържание, до някаква степен се гарантират повече посещения на хора с интереси, сходни с тематиката на сайта. Поради това, че търсачките се стремят да извеждат полезни за потребителите резултати, то връзки от даден сайт, които се срещат често, биват тълкувани като полезни и съответно индексирани по-добре. Това обаче създава предпоставки за използване на забранени от търсачите методи, а именно закупуване на линкове в големи количества от източници с неизяснен произход.

***Външни фактори***

* „Правилна“ регистрация в основните търсещи машини и каталози. В „по-нисшите“ каталози и търсачки може да се използват автоматизирани услуги за оптимизация.
* Увеличаване на „индекса на цитиране“: обмен на връзки между страници със сходно съдържание. Най-популярният „индекс“ е PageRank в Google.

**Поведенчески фактори**

Поведенческите фактори представляват анализ и водене на статистика за дейността на потребителя. Това включва отчитане на продължителността на престой в страницата, отчитане на ключови думи посредством, които потребителят е стигнал до страницата и къде точно в нея е бил препратен. Възползвайки се от тази информация SEO специалистите оптимизират към по-горно класиране появата на уеб страницата в общите резултати на търсачката.

***Мета тагове и ключови думи***

Целта на мета таговете и „ключовите думи“ е да направят описанието на страницата, разбираемо от търсещите машини. За съжаление това е било използвано от така наречените „спамери“, които злоупотребяват с това. Заради тях търсещите машини практически са премахнали индексирането по ключовите думи и мета таговете, защото в резултатите при търсенето те често не дават искания от потребителя резултат. В момента от най-голямо значение е съдържанието на страницата, а не мета таговете. Повече „тежест“ има текстът в определени тагове (<h1>..<h6>, <li>, <p>, <a>). Въпреки всичко това мета таговете не са напълно изключени и все още са важна част от оптимизацията.

Примери за мета тагове:

**<head>**

**<title>**До 60-70 символа**</title>** - заглавие (най-важната част)

**<meta** name="author" content="xxx@xxx.xx" **/>**

**<meta** name="owner" content="xxx@xxx.xx" **/>**

**<meta** name="area" content="Industry" **/>**

**<meta** name="rating" content="General" **/>**

**<meta** name="description" content="Описание на страницата" **/>**

**<meta** name="abstract" content="" **/>**

**<meta** name="keywords" content="Ключови думи около 250 символа" **/>**

**<meta** name="copyright" content="" **/>**

**<meta** http-equiv="audience" content="Professional" **/>**

**<meta** http-equiv="content-type" content="text/html;charset=utf-8" **/>**

**<meta** http-equiv="Content-Language" content="bg" **/>**

**<meta** name="subject" content="ххх" **/>**

**<meta** http-equiv="cache-control" content="no-cache" **/>**

**<meta** http-equiv="pragma" content="no-cache" **/>**

**<meta** name="revisit-after" content="15 days" **/>** - интервал от време през, който „роботът“ отново да сканира страницата

**<meta** name="robots" content="index, follow" **/>**

**</head>**

1. **Потребителското преживяване**

Посещавайки уеб сайт, кликайки на бутон, четейки статия или поръчвайки продукт, потребителят преминава през процеса на т.нар. Потребителско изживяване. То може да бъде както много добро, така и много лошо и безмислено. Да вземем например големите сайтове – Google, Facebook, Youtube. Там всеки е наясно за какво е сайта, как да търси, навигира и извърши целта, за която е дошъл в сайта. Лош вариант на потребителското изживяване са претъпкани сайтове, в които трудно се разбира къде е менюто или дори не може да се разбере за какво иде реч.

Има 3 основни стъпки с които може да се подобри потребителското изживяване.

1. Да се използват по-малко изображения

Факта, че все повече хора използват смартфона за бърз достъп до информация в Интернет, говори много за това, колко бързо трябва да зарежда един сайт. Ако уеб сайта съдържа много изображения, то скороста на зареждане определено ще е ниска и много хора биха избрали друг сайт, който зарежда по-бързо.

1. Да се направят важните бутони ВИДИМИ

Много е важно бутоните да се виждат добре за да насочат лесно потребителя към важните страници в сайта. Пример за това би бил бутон за Отстъпка, Ваучер или Оферта. Дори слагайки голям бутон “Контакти”, е голямо предимство, защото просто се навигира лесно и който има намерение, би се свързал с вас лесно.

1. Залепено меню с търсачка

Колкото и да се скролва надолу по страницата, менюто винаги остава видимо за потребителя и по този начин той има лесен достъп до всяка една страница, независимо къде се намирате в сайта. Освен достъп до страниците, имайки търсачка в менюто е много удачен начин да се улеснят потребителите и да се подобри потребителското преживяване в сайт.

**Предимствата на HTML5 и с какво те се отличават от HTML4**

HTML5 – нов подход към маркирането

За разлика от разработения през 90-те години HTML4 стандарт , през 2004 година, когато тепърва се разработваше новата му версия, в Интернет вече се разпространяваха аудио и видео записи като официални и неофициални. На преден план изпъкна тенденцията за поместване на анимационни филми и видеоклипове, а също така аудио блоговете с подкастове станаха особено популярни.

Но действащият стандарт не е предвиждал описание на мултимедийно съдържание чрез средствата на езика. За тази цел се е използвала функцията за поставяне на определен обект – „object„. Възможността той да се възпроизведе зависи от това, дали браузърът на клиента е съвместим с такъв контент – т.е., дали има определени плъгини, създадени от трети разработчици. Отнася се предимно за Adobe Flash, Quick Time или по-неизвестните Real Player и Silverlight. Всичко това – „надстройките“, допълненията към браузъра, които не се явяват негова съставна част и изпълняващи посредническа дейност, преобразуват дигиталното съдържание във видео и звук.

Тази система и сега е напълно ефективна, но липсата на общоприети стандарти води до това, че в различните страници се използват различни технологии за преглеждане на медийно съдържание – т.е. изискват се различни плъгини и тъй като те са отделна част от браузъра – трябва да се инсталират всеки път самостоятелно за разглеждането на определен контент.

В стандарта HTML5, който комбинира HTML, CSS и JavaScript, медийното съдържание се възпроизвежда по различен начин – явява се като неизменна част от кода на самата страница. HTML5 осигурява единен стандартен метод за описание (синтаксис), който надгражда аудио визуалния контент в сайта. За интеграцията на аудио и видео е достатъчно използването на определени тагове, описващи какъв тип контент искате да вмъкнете – «video» или «audio».

Проблемът в обратната съвместимост – първо HTML, после плъгини.

За съжаление, все още има потребители, използващи остарял браузър – напълно задоволяващ техните нужди, но неподдържащ възможностите на HTML5. Предимно Internet Explorer версия 8, която и до днес се смята за най-популярния бразуър. Последващите версии – 9 и 10, вече практически поддържат изцяло HTML5. Що се касае до Chrome, Opera и Safari, те от години имат функцията да обработват таговете «video» и «audio».

И все пак, за осигуряването на обратната съвместимост в HTML5 е нужна поддръжка на използваните плъгини (тага «object»), но по схемата «първо HTML, а след това и плъгин». Това означава, че браузърът първо трябва да осъществи визуализацията на страницата чрез използването на таговете «video» и «audio» и само при евентуална невъзможност да се вмъкне код за плъгин.

За осигуряването на правилното функциониране на подобен подход може да се използва една от многото достъпни библиотеки на JavaScript, част от които предоставят изцяло интегриран програмен интерфейс (API) за управлението както на HTML контент, така и на вмъкнатите обекти. Сред тях е и програмния видеоплейър Video.js, предоставящ пълна съвместимост и с HTML5, и със старите браузъри, в които се вкарват плъгини.

Единен подход спрямо десктопни и мобилни платформи

Поддръжката на новия стандарт HTML5 се осигурява практически и на всички съвременни мобилни устройства – в това число и на iOS, Android и Windows Phone – което само по себе си е и другото му предимство. За уеб мастърите вече не е необходимо да създават няколко версии за мултимедийно съдържание, за да предложат пълноценно възпроизвеждане и на компютри, и на мобилни устройства. Известен е фактът, че iOS е напълно несъвместима с Flash, а ОС Android – само частично. HTML5 обаче преодолява тези проблеми.

Освен това, много от модерните мобилни устройства са с апаратно декодиране на видеата – в широко разпространения формат Н.264, което за пореден път улеснява уеб програмистите, като ги освобождава от задачата да надграждат видеа в няколко различни кодеци. Собствениците на тези устройства се радват на по-малка степен на натоварване и на по-дълга продължителност на автономната работа.

Поддръжка на различните видео и аудио формати

Въпреки че HTML5 позволява поместването на мултимедийни файлове чрез стандартни и прости тагове на уеб страницата, тези файлове могат да бъдат закодирани в различни формати и с различни кодери. До този момент просто няма определен стандартен формат (на контейнера) и кодер, които да работят с еднакъв успех и на браузър, и на мобилно устройство.

Най-известните контейнери за интернет видеа са AVI, MP4, FLV и WebM, а основните онлайн видео кодеци са само 3 – H.264, Ogg Theora и VP8. При това, на всеки кодер най-добре съответства определен формат на контейнера: H.264 — MP4, Theora — OGG, VP8 — WebV. Кодек H.264 свива видеото с определени загуби, за да получи файл с високо качество, но с малък размер. Стесняването може да е с различно качество и размери с цел на по-слабите мобилни устройства да се възпроизвежда „лек” файл, а на десктопите – широкополюсен достъп и по-високо качество.

Въпреки, че H.264 е комерсиален кодек, той е доста по-унверсален. Неговата надстроена поддръжка е налична в Internet Explorer, Chrome, Firefox, Safari, iOS и Android, като отсъства само в Opera, където подобно видео може да се вмъкне само с външен плъгин. Кодек Ogg Theora, по начало е създаден за системата на Linux, като също първо е бил комерсиален, но на този етап вече се лицензира безплатно. Внедрената поддръжка е налична в браузърите Chrome, Firefox и Opera, като в Internet Explorer и Safari е достъпна само с плъгини.

Кодек VP8 е ориентиран към мобилните устройства, разпространява се в Google с условията на Open Source и осигурява качеството на H.264 при по-малко сложно декодиране. Внедрената поддръжка тук е при Chrome, Firefox и Opera, а също така и в мобилните устройства с Android. По този начин, за възпроизвеждането на видео контент във всички налични системи е достатъчно да се постави файла в контейнерите Mp4 със стандартно кодиране H.264 и във формат WebM с кодек VP8. И тъй като в повечето сегашни десктопни браузъри поддръжка на H.264 липсва само при Opera, подобни съчетания ще са най-често използвани като универсални.

**Оптимизация на ASP.NET страница**

**Кеширане**

В началото, когато уеб сайтовете са представлявали само HTML файлове, с повече текст и малко изображения, скоростта на зареждане на уеб страниците все още не е била основен фактор за доброто потребителско изживяване. Когато обаче уеб сайтовете започват да предлагат динамично генерирана информация, скоростта на зареждане започва пряко да влияе на доброто интерактивно изживяване на уеб потребителите. Освен HTML файлове уеб сървърите се натоварват да работят със скриптови езици, които пък работят в близко сътрудничество с бази от данни. Размерът и сложността на уеб информацията се увеличава, което довежда до забавяне в доставянето на уеб съдържанието до уеб потребителите. За да се зареди само една страница от динамичен сайт, се извършва интензивна работа от страна на сървъра. Стартират се скриптове, които обработват данни, изпълняват се SQL заявки към базата данни и различни други процеси, само за да се генерира информацията, която уеб потребителят очаква.

**Кеширане на информация в Уеб (Web cache)**

Думата "кеш" идва от "cache" на английски, което преведено буквално означава "складирам, скривам". А Web cache означава складиране на уеб информация.

Въпреки че информацията в уеб страниците вече се генерира динамично, тя не се променя всяка секунда. Това означава, че една и съща информация се предоставя на всички потребители, но за всеки един от тях ще се генерира наново. Този процес на динамично генериране на една и съща информация се извършва постоянно, при всяко достъпване. Това довежда до излишно използване на ресурси като време, процесорен ресурс, памет и други. Заради тази ситуация на разхищение на ресурси и време, което пряко засяга времето на зареждане на уеб сайта, се създават технологии за складиране на вече веднъж генерираната информация. След като информацията, генерирана по динамичен начин, е записана (кеширана), тя представлява статично съдържание. То се предоставя директно на потребителите без генериране, изчакване и забавяне. Кеширането на информация за следваща употреба е ефективен метод за ускоряване на зареждането на динамичен сайт. Когато се приложи и компресиране на тази записана информация, свалянето й от уеб потребителя се ускорява още повече.

*Кеширане и компресиране в Уеб:*

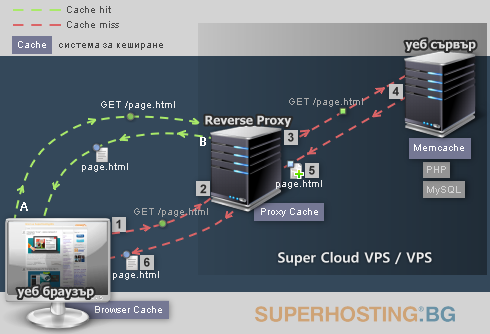
* намалява времето на изчакване;

• намалява мрежовия трафик; намалява количеството информация предавано по мрежата;

* намалява процесорното потребление и натоварване на уеб сървъра;
* помага за по-бърз достъп и по-бързо зареждане на уеб ресурсите.

**Системи за кеширане и ускоряване в Уеб**

Системите за ускоряване в Уеб (още наричани web accelerators) ползват кеширане на информацията, като част от цялостната им функция. Освен кеширане, тези системи за ускоряване може да прилагат компресиране, филтриране, предварително изтегляне на ресурси и други. Уеб информация може да се кешира локално при уеб потребителя - Browser Cache, преди сървъра-източник на уеб съдържанието - Reverse Proxy/Web Accelerator и при самия сървър-източник на ниво приложение. На Фиг. 3 е показано как работи кеша.



Фиг.3 Как работи кеширането.

**Cache miss --- (търсената информация не е налична в кеша)**

1. Уеб браузърът изпраща заявка до сървъра, отговарящ за търсения домейн, за уеб ресурса page.html, заявката се прихваща от прокси сървъра;
2. Прокси сървърът проверява в кеша за търсената страница и не я намира. Този пропуск може да се нарече **Cache miss**.
3. Прокси сървърът изпраща запитване за ресурса до сървъра-източник.
4. Сървърът-източник връща на проксито търсената уеб страница.
5. Прокси сървърът получава, копира уеб ресурса и го изпраща към уеб клиента.
6. Уеб браузърът получава търсения уеб ресурс.

**Cache hit --- (търсената информация е налична в кеша)**

1. Уеб браузърът изпраща заявка за уеб ресурса page.html до сървъра, отговарящ за търсения домейн и заявката е прихваната от прокси сървъра (тъй като той се намира пред сървъра-източник).
2. Прокси сървърът проверява в кеша за търсената страница и я намира - **Cache hit**. Директно предоставя уеб ресурса към уеб клиента, като този път не се обръща към сървъра-източник.

**Какво е Browser Cache?**

Уеб браузърът на потребителя може да кешира обекти от уеб сайтовете като изображения, скриптове, страници и други. Когато се достъпи дадена уеб страница, уеб браузърът първо ще провери дали обектите на тази страница вече не съществуват в кеша. Ако обектите са записани и все още са актуални браузърът ще ги зареди от кеша, вместо да се свързва уеб сървъра. Зареждането на данни от кеша на уеб браузъра е много по-бързо, отколкото изтеглянето им от уеб сървъра. Понякога, след извършване на промени по даден уеб сайт, за да се види актуалната информация в сайта, може да е необходимо да се изчисти кешираната информация в уеб браузъра.

**Какво е Reverse Proxy?**

Reverse Proxy (обратно прокси) е тип прокси сървър (посредник между уеб клиент и уеб сървър) и една от функциите, които може да изпълнява е кеширане. Reverse Proxy може да се постави пред уеб сървъра, като потребителите се свързват с прокси сървъра, вместо директно с уеб сървъра.

Някои от благоприятните функции, които Reverse Proxy може да изпълнява са:

* намаляване на натоварването на уеб сървъра, чрез ефективно кеширане на статично и динамично съдържание (web accelerator);
* намаляване на времето за достъп до уеб сайта;
* разпределяне на натоварването, в случай че зад прокси сървърът има повече от един уеб сървър;
* защита от общи уеб базирани атаки;
* компресиране на съдържанието, за по-бързо зареждане на уеб сайта;

Недостатък е, че Reverse Proxy похватът за кеширане на информация работи само при трафик към сайта, генериран от анонимни потребители. Когато сайтът се достъпи от потребител, който се е логнал в системата, кеширане на информацията не се прилага.