

Heidi Virta

# Sisällönhallintajärjestelmää käyttävän verkkosivun suorituskyvyn optimointi

Metropolia Ammattikorkeakoulu  
Insinööri (AMK)  
Mediatekniikan koulutusohjelma  
Insinöörityö  
15.5.2012

Tekijä Otsikko Sivumäärä Aika	Heidi Virta Sisällönhallintajärjestelmää käyttävän verkkosivun suorituskyvyn optimointi 43 sivua + 4 liitettä 15.5.2012
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen media
Ohjaajat	yliopettaja Harri Airaksinen lehtori Ilkka Kylmäniemi
<p>Insinööriyössä tutkittiin sisällönhallintajärjestelmän avulla toimivan verkkosivun suorituskykyä ja sen optimointia. Työssä keskityttiin verkkosivun optimointiin käyttäjäpuolella (client-side), mutta raportissa käydään läpi myös palvelinpuolen (server-side) optimointitoimenpiteitä. Insinööriyön tavoitteena oli selvittää sisällönhallintajärjestelmällä rakennetun sivuston ongelmakohdat latautumisenopeuden kannalta ja maksimoida sen suorituskyky ja kestävyys ruuhkatilanteessa.</p> <p>Työn tutkimuksissa perehdyttiin verkkosivun latautumisen aikaansaamaan selaimen ja palvelimen väliseen liikenteeseen ja etsittiin sitä kautta sivuston latautumisenopeuteen vaikuttavia hidasteita. Tutkimuskohteena oli WordPress-sisällönhallintajärjestelmällä rakennettu verkkosivu, joka esittelee kiinteistövälitysyrityksen myyntikohteita. Sivusto oli julkaistu jo ennen insinööriyön aloittamista.</p> <p>Insinööriyön tutkimuksissa selvitettiin myös WordPress-teemojen rakenteiden ja sivustolla esiintyvien kuvien vaikutukset verkkosivun suorituskykyyn. Suorituskykyä mitattiin neljän eri mittaustyökalun avulla, jotka pisteivät verkkosivun toiminnan tiettyjen sääntöjen perusteella. Tehdyt mittaukset osoittivat, että verkkosivu oli lähtötilanteessa suorituskyvyllään heikko, joten optimointitoimenpiteiden toteuttaminen oli erittäin tärkeää.</p> <p>Optimoinnin myötä sivuston lähdekoodia siistittiin, HTTP-pyyntöjen määrää vähennettiin, välimuisti otettiin käyttöön ja ulkoisten tiedostojen kokoa pienennettiin. Suurin osa optimoinnista toteutettiin WordPressin lisäosien avulla, mikä oli nopea optimointitapa, mutta parhaisiin tuloksiin pääseminen vaati lisäosien optimaalisten asetusten selvittämisen.</p> <p>Insinööriyön lopputuloksena syntyi sujuvasti toimiva verkkosivu, joka kestää suuremman määrän samanaikaisia käyttäjiä. Tehdyt mittaukset osoittivat, että optimointi on oleellinen osa verkkosivujen kehittämistä, eikä sitä tule laiminlyödä. Sivuston lähtötilanteen mitaustuloksia analysoitaessa WordPress-teemoissa havaittiin heikkouksia, jotka on syytä ottaa huomioon vastaisuudessa, kun sisällönhallintajärjestelmällä rakennetaan verkkosivuja. Myös työssä ilmi tulleet tiedot WordPress-teemoista ja järjestelmän lisäosista ovat erittäin hyödyllisiä tulevia projekteja ajatellen.</p>	
Avainsanat	sisällönhallintajärjestelmä, WordPress, suorituskyky, optimointi

Author Title	Heidi Virta Performance optimization of a content managed website
Number of Pages Date	43 pages + 4 appendices 15 May 2012
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructors	Harri Airaksinen, Principal Lecturer Ilkka Kylmäniemi, Lecturer
<p>This thesis examines the performance optimization of a website built on a content management system (CMS). The thesis focused on optimizing the website on the client-side, but some server-side optimization techniques are presented in the report as well. The aim of the thesis was to examine matters that negatively affect the speed of a CMS based website and to maximize the performance and durability of the site during traffic peaks.</p> <p>A real estate website built on the WordPress CMS served as the target of the thesis. The site had been published before the start of this thesis. The research in this thesis was concentrated on the traffic that occurs between a server and a browser when a website is loaded. The structure of a WordPress theme is also examined in the research, as is the impact the website images have on the performance of a site. Performance was measured with four different measurement tools, which graded the website based on certain rules. The results showed that the initial performance of the site was poor, which proved the optimization to be very important.</p> <p>During the optimization the website source code was partly rewritten, the amount of HTTP requests was reduced, caches were activated and the size of external files was minimized. Most of the optimization was carried out with WordPress plug-ins, which was a fast optimization method, but best results were reached only after finding the ideal settings to these plug-ins.</p> <p>The result of the thesis was a smoothly functioning website that endures a large number of concurrent users. The taken measurements show, that performance optimization is an important part of web development, and should not be neglected. Measurements made at the starting point of the thesis revealed weaknesses in WordPress themes that should be taken into account in the future, when working with this CMS. Also other significant information concerning WordPress themes and plug-ins was discovered during this study that will be very useful in future projects.</p>	
Keywords	content management system, WordPress, performance optimization

## Sisälllys

1	Johdanto	1
2	Sisällönhallintajärjestelmä	3
2.1	Sisällönhallintajärjestelmän toiminta	3
2.2	Soveltuvuus eri ympäristöihin	6
3	Verkkosivun optimointi	8
3.1	Optimoinnin tarkoitus	8
3.2	Optimoinnin käytäntöjä	10
3.3	Sisällönhallintajärjestelmän optimointi	17
3.4	Sisällönhallintajärjestelmän lisäosien vaikutus optimointiin	19
4	Yrityksen verkkosivun optimointiprosessi	20
4.1	Verkkosivun suorituskyky ja kuormituksen kesto	20
4.2	Mittaustulokset ennen optimointia	24
4.3	Optimoinnin tavoitteet	27
4.4	Optimointitoimenpiteet lisäosien avulla	30
4.5	Muut optimointitoimenpiteet	33
4.6	Verkkosivun analysointi optimoinnin jälkeen	34
4.7	Jatkotoimenpiteet	36
5	Yhteenveto	39
	Lähteet	41
	Liitteet	
	Liite 1. Tutkimussivuston kävijämäärä maaliskuussa 2012	
	Liite 2. Etusivun mittaustulokset GTmetrix-työkalulla ennen optimointia	
	Liite 3. Kuvien optimointi Yagoon Smush.it-työkalulla	
	Liite 4. Mittaustulosten ennen ja jälkeen -vertailu	

## 1 Johdanto

Vuonna 2012 tietoverkot kattavat miltei koko maailman ja yhteydet saavuttavat parhaimmillaan 100 gigatavun sekuntinopeuden [1]. Verkon infrastruktuuri ja tekninen toteutus muistuttavat taidetta. Erittäin helpoksi luokiteltavan internetin selailun mahdollistavat huipputeknologiat, joita pääsee hyödyntämään lähes maksutta suuren käyttäjämäärän ansiosta. Palvelinverkostosta muodostuva internet on käyttäjien lisääntyessä paisunut rönsyileväksi tietovarastoksi, joka kaikessa kaoottisuudessaan kuitenkin pysyy toimivana.

Jos haluaa jatkossakin käyttää hyvin toimivaa indeksoitua tietoverkkoa, voi aloittaa pitämällä huolta omien verkkosivujen sisällöstä, turhan datan karsimisesta sekä sujuvasta sisällön tarjoilusta käyttäjille. Tässä insinööriyössä tutkitaan sisällönhallintajärjestelmän avulla toimivan verkkosivun suorituskykyä ja optimoidaan sivusto toimimaan mahdollisimman sujuvasti. Optimoinnissa keskitytään sivuston latautumisnopeuteen ja palvelimen kuormittumiseen ja esitellään keinoja, joilla verkkosivun suorituskykyä voidaan parantaa sisällönhallintajärjestelmän suomien mahdollisuuksien puitteissa.

Työn tutkimuskohteena on espanjalaisen kiinteistövälitysyrityksen, SuVillan, verkkosivut. SuVilla S.L. on pieni perheyriutus, joka välittää asuntoja erityisesti suomalaisille Etelä-Espanjassa, Aurinkorannikolla. Insinööriyön yritykselle tuoma hyöty liittyy lähinnä asiakastyytyväisyyteen ja yrityksen nykyaikaistamiseen. Verkkosivun suorituskyvyn parantaminen tarkoittaa sivujen selaajan näkökulmasta toimintojen nopeutumista, ja mitä nopeampi sivusto, sitä tyytyväisempi käyttäjä. SuVilla on aiemminkin esitellyt myytäviä asuntojaan internetissä, mutta insinööriyön myötä tehtyjen muutosten ansiosta verkossa tapahtuva markkinointi tehostuu ja asiakkaan on helpompi lähestyä yritystä.

SuVillan sivustoa päivitetään melko usein myytävien kohteiden lisääntyessä, ja sen päivittäinen kävijämäärä on kohtalaisen suuri, noin 50–100 kävijää. Verkkosivuja ylläpidetään WordPress-sisällönhallintajärjestelmän avulla. SuVillan sivusto on vastikään vaihdettu toimimaan WordPress-alustalla, eikä optimointitoimenpiteitä ole vaihdon jäl-

keen tehty. Sivusto toimii lisäksi kolmella eri kielellä, mikä lisää verkkosivun kuormaa ja vaikuttaa siten optimoinnin haastavuuteen.

Työn raportissa luodaan katsaus sisällönhallintajärjestelmien ja erityisesti WordPressin toimintaan ja sen soveltavuuteen erilaisten verkkosivujen sisällönhallintajärjestelmäksi. Verkkosivujen optimoinnin tarkoitus ja optimointikäytännöt käydään läpi yleisellä tasolla ja selvitetään, mitä eroa palvelinpuolen ja käyttäjäpuolen optimoinnilla on. Suorituskykyyn vaikuttavat tekijät tunnistetaan työssä esiteltävien testaustyökalujen avulla, ja tehdyt optimointitoimenpiteet raportoidaan. Optimoinnilla aikaansaadut tulokset analysoidaan, ja raportin lopussa käydään läpi, minkälaisia jatkotoimenpiteitä SuVillan verkkosivulle olisi mahdollista suorittaa vielä nopeamman sivuston aikaansaamiseksi.

## 2 Sisällönhallintajärjestelmä

### 2.1 Sisällönhallintajärjestelmän toiminta

Sisällönhallintajärjestelmällä viitataan sovellukseen, jolla hallitaan verkkosivun sisältöä eli tekstejä, kuvia ja videoita [2]. Sisällönhallintajärjestelmiä on satoja, maksullisia ja ilmaisia, ja erilaisia järjestelmiä on rakennettu erityyppisiä verkkopalveluita varten. Sisällönhallintajärjestelmä sijaitsee samalla palvelimella kuin verkkosivu, ja sivuston sisältöä päästään hallinnoimaan kirjautumalla sisään järjestelmään selaimen kautta.

Suurin sisällönhallintajärjestelmästä saatava hyöty on sen helppokäyttöisyys. Sen avulla verkkosivun ylläpitäjä hallinnoi verkkosivun sisältöä selkeässä visuaalisessa käyttöliittymässä ohjelmoinnin sijaan. Verkkosivun sisällön lisääminen ja muokkaaminen on myös huomattavasti nopeampaa sisällönhallintajärjestelmän avulla verrattuna staattisen verkkosivun sisällön hallinointiin. [2, s. 5–6.]

Viime vuosien aikana avoimen lähdekoodin sisällönhallintajärjestelmät ovat yleistyneet nopeasti. WordPress on yksi näistä avoimen lähdekoodin sovelluksista, ja sen avulla voidaan julkaista ja hallita sisältöä verkossa. Avoimella lähdekoodilla tarkoitetaan ilmaiseksi ladattavissa olevaa sovellusta tai koodia, jota kuka tahansa voi kehittää ja käyttää haluamallaan tavalla [3]. WordPressin kehitys on aloitettu vuonna 2003, ja alun perin sen oli tarkoitus palvella vain henkilökohtaisten kirjoitelmien julkaisualustana. Innokkaan ja innovatiivisen kehittäjäyhteisön myötä WordPress kuitenkin kasvoi täysiveriseksi sisällönhallintajärjestelmäksi. WordPress-sovellus on ladattavissa osoitteesta [www.wordpress.org](http://www.wordpress.org), ja se on saatavilla 72 eri kielellä. [4.]

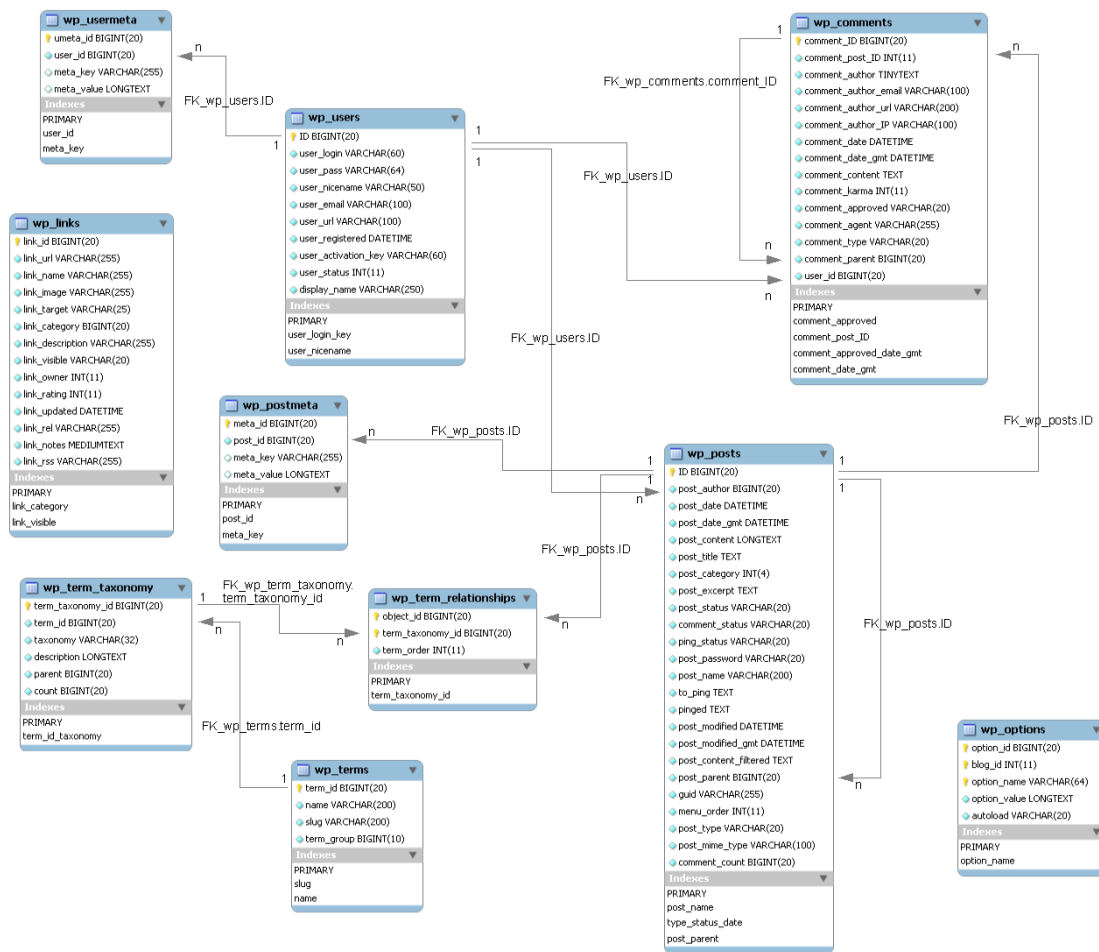
Käyttääkseen WordPressiä verkkosivuston sisällönhallintajärjestelmänä verkkosivun ylläpitäjän tulee asentaa sovellus palvelintilaansa. Asennusvaiheessa palvelimen tietokantayhteys määritellään config.php-tiedostoon, minkä jälkeen WordPressin tiedostot ladataan palvelimelle. Latauksen jälkeen selaimella siirrytään oman palvelimen osoitteeseen, missä asennus viimeistellään ohjattuna toimintona. Ohjattu toiminto luo tietokantaan tarvittavat taulut ja syöttää niihin asennuksen vaatimat tiedot. [2, s. 9–12.]

WordPress on kirjoitettu PHP-kielellä, ja sen toiminta perustuu ytimeen, tietokantaan, teemoihin ja lisäosiin. Ydin luo pohjan koko sisällönhallinnalle ja sivuston perusominaisuuksille. Teema määrittää sivuston ulkonäön sekä sivutyypit ja toiminnallisuudet, joita sivustolla esiintyy. WordPressin lisäosat ovat usein yksityisten kehittäjien tekemiä liitännäisiä, joita asentamalla sivustolle saadaan helposti uusia toimintoja ilman ohjelmointia. Lisäosan avulla voi esimerkiksi tehdä sivustosta monikielisen, kuten SuVillan tapauksessa on tehty. [5.]

WordPress-sivuston sisältö tuodaan dynaamisesti tietokannasta selaimeen, mikä on yksi sisällönhallintajärjestelmän pääpiirteistä. Dynaaminen verkkosivu eroaa staattisesta sivusta siten, ettei teksti- ja mediasisältöjä ole kirjoitettu sivun lähdekoodin sekaan, vaan tieto haetaan tietokannasta pyydettäessä ja upotetaan koodiin juuri, ennen kuin selain prosessoi sivun näytettäväksi. Dynaamisuus mahdollistaa sisältöjen tuomisen rakenteeltaan erilaisiin sivupohjiin, ja samaa sisältöä voidaan toistaa eri kohdissa. Joskus dynaamisella verkkosivulla viitataan myöskin sivuun, jonka sisältö tuodaan eri palvelimelta, kuin missä verkkosivu sijaitsee, eikä sisältöä ole tuotettu itse. [6.]

Niin sanotun *puhtaan asennuksen* jälkeen WordPressin tietokantarakenne on melko yksinkertainen, ja se sisältää 11 taulua. Puhtaalla asennuksella tarkoitetaan WordPress-asennusta, johon ei ole liitetty lisäosia tai muuta ylimääräistä. Tietokannan taulut sisältävät WordPressin komponenttien, artikkelien, sivujen, kommenttien, lisäosien, käyttäjien ja järjestelmän rakenteen sisältö- ja metatietoja. [2, s. 103–105.] Kuviossa 1 esitellään WordPress-tietokannan rakenne ja taulujen keskinäiset relaatiot (viivat), joiden avulla määritellään tallennettujen tietojen suhteet toisiinsa. Kuvioista nähdään myös, että jokaisella taulukolla on indeksejä. Indekseillä on suuri merkitys tietokannan optimoinnissa, sillä ne toimivat kuten kirjan sisällysluettelot ja nopeuttavat siten tiedon hakua kannasta [7].



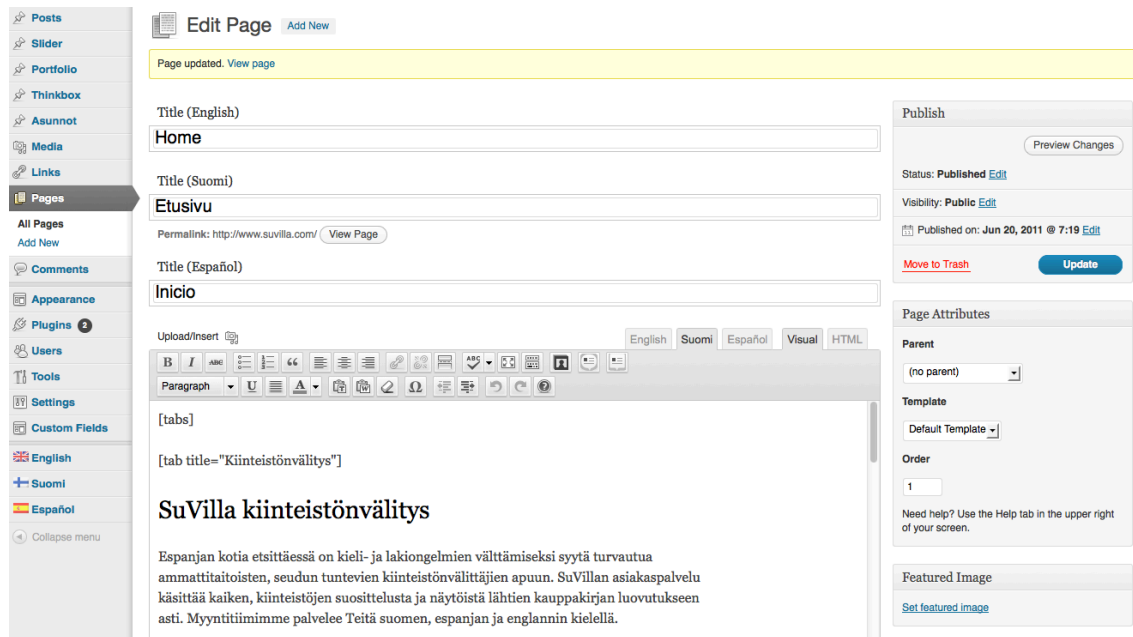


Kuvio 1. WordPressin asennuksessa luotava tietokantarakenne.

Lisäosia tai teemoja asennettaessa WordPressin tietokantaan tulee uusia tauluja ja sen rakenne saattaa monimutkaistua [2, s. 103]. Joskus huolimattomasti koodatut lisäosat luovat turhaa tai indeksoimatonta tietoa tietokannan tauluihin. Näin ollen lisäosat ovat ensimmäinen epäilyksen kohde, mikäli verkkosivun toiminta on häiriintynyt.

Dynaamisen verkkosivun paras puoli on se, että koska sisältö sijaitsee tietokannassa, erillään verkkosivun rakenteesta, sitä voidaan myös muokata erikseen. Sisällön hallintaan WordPress tarjoaa yksinkertaisen muokausnäköm, joka muistuttaa pitkälti tekstinkäsittelyohjelmaa, ja jokaiselle toiminnolle on olemassa visuaaliset painikkeet. Kuviossa 2 on havainnollistettu SuVillan etusivun sisällön muokkaamista WordPressin sisäisen muokkaimen avulla. Kuviossa nähdään selkeät tekstikentät ja painikkeet, joiden avulla sisältöä muokataan ja tallennetaan tietokantaan. Update-painiketta napsauttamalla muokattava sisältö tallentuu ja päivittyy verkkosivulle. Kuviossa voi myös huo-

mata, miten monikielisen sivuston hallinnointi on tehty yksinkertaiseksi qTranslate-lisäosan avulla, josta kerrotaan enemmän luvussa 3.4.



Kuvio 2. Yksittäisen sivun muokkausnäkyminen WordPress-järjestelmässä.

WordPressin käyttöliittymä on selkeä ja helppokäyttöinen muutenkin, kuin pelkästään sisällön muokkaamisen osalta. Kuvion 2 vasemmassa laidassa nähdään sisällönhallintajärjestelmän valikkorakenne, jossa tärkeimmät toiminnalliset kokonaisuudet on niputettu yksittäisen otsikon alle. Rauhallinen, vaalea värimaailma ja elementtien viimeistelymuotoilu ovat myös WordPressin selkeyttä edistäviä tekijöitä.

## 2.2 Soveltuvuus eri ympäristöihin

Alun alkaen bloggaajille suunnattu WordPress-julkaisutyökalu soveltuu nykyään lähes minkäläisen tahansa verkkosivun hallintaan, aina henkilökohtaisista sivustoista laajoihin verkkokauppoihin asti. Lukuisat suuryritykset, kuten Ford, Samsung ja Ebay käyttävät WordPressiä blogi-, tuki- ja uutissivuillaan. Tämä kertoo siitä, kuinka suuressa arvossa tätä ilmaissovellusta pidetään. [8.]

Juuri muuntautumiskykyä ansiosta tämän opinnäytetyön tutkimuskohteena toimivan SuVilla-kiinteistönvälityksen verkkosivusto on rakennettu WordPress-alustalle. Tarvemmin ajatellen bloggaussivuston ylläpitäminen on melko lähellä kiinteistönvälitysyri-

tyksen verkkosivujen toimintaa. Voisi ajatella, että jokainen välitettävä asunto on ajan-kohtaisaihe, joka päivitetään sivulle heti, kun uutta kerrottavaa ilmenee eli uusi kohde otetaan myyntiin. Uusimmat ajankohtaisaiheet saavat huomioita etusivulla, ja sivuston selaajat voivat hakea toiveidensa mukaisia aiheita sivuston hakukoneella. Kiinteistönvälityspuolella uusimmat kohteet näkyvät etusivulla isoina, ja sivuston hakukoneella haetaan unelmien kotia.

SuVillan verkkosivujen rakentaminen ja julkaiseminen oli kiireellinen projekti, sillä yrityksen vanha sivusto oli jäämässä käyttökelvottomaksi vanhentuneen teknologiansa vuoksi. Kiireellisen tilanteen takia uuden verkkosivun pohjaksi valittiin kiinteistönvälittäjille suunniteltu maksullinen WordPress-teema, joka ostettiin [www.themeforest.net](http://www.themeforest.net)-verkkopalvelusta. Teeman avulla sivustolle saatiin hetkessä asiallinen ja näyttävä ulkonäkö. Valmiin teeman ostaminen säästää huomattavan määrän aikaa sivuston rakennusvaiheessa ja tarjoaa valmiita toimintoja verkkosivulle, kuten tässä tapauksessa muun muassa räätälöidyn hakukoneen ja lomakkeen, jonka avulla asunto lisätään myyntiin [9].

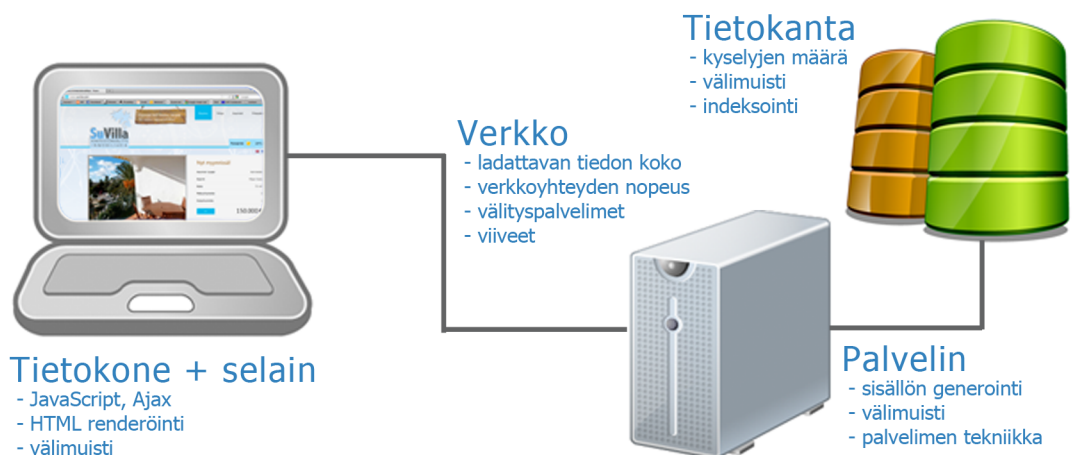
Verkkosivun optimoinnin kannalta valmis teema on haastava vaihtoehto, sillä sen rakenne täytyy tarkastaa mahdollisten ohjelmointivirheiden varalta, jotka voisivat hidastaa sivun suoriutumista. Valmiin teeman kanssa optimoinnin tulokset ovat harvoin täydellisiä, koska teema rakennetaan sopivaksi mahdollisimman laajalle käyttäjäkunnalle, jolloin osa sen ominaisuuksista jää melkein poikkeuksetta käyttämättä ja lähdekoodiin. Näin ollen suuremmille yrityksille on ensisijaisen tärkeää aloittaa rakentamalla teema itse, jotta vältetään turhalta sisällöltä ja tehottomalta koodilta.

### 3 Verkkosivun optimointi

#### 3.1 Optimoinnin tarkoitus

Käyttäjän siirtyessä internetissä verkkosivulle www-palvelin lähettää selaimelle HTML-dokumentin (HyperText Markup Language). Dokumentti sisältää tiedot käyttäjälle näytettävästä sisällöstä sekä tarvittavien ulkoisten tiedostojen osoitepolut. Itse HTML-dokumentin latausaika on vain noin 10–20 prosenttia koko verkkosivun latausajasta. Loput 80–90 prosenttia kuluu sivun sisällön vaatimien ulkoisten tiedostojen, kuten kuvien ja videoiden, lataukseen. Optimointitoimenpiteillä pyritään oleellisesti nopeuttamaan tämän sisällön latautumista, joka syö suurimman osan koko sivuston latautumisajasta. [10.]

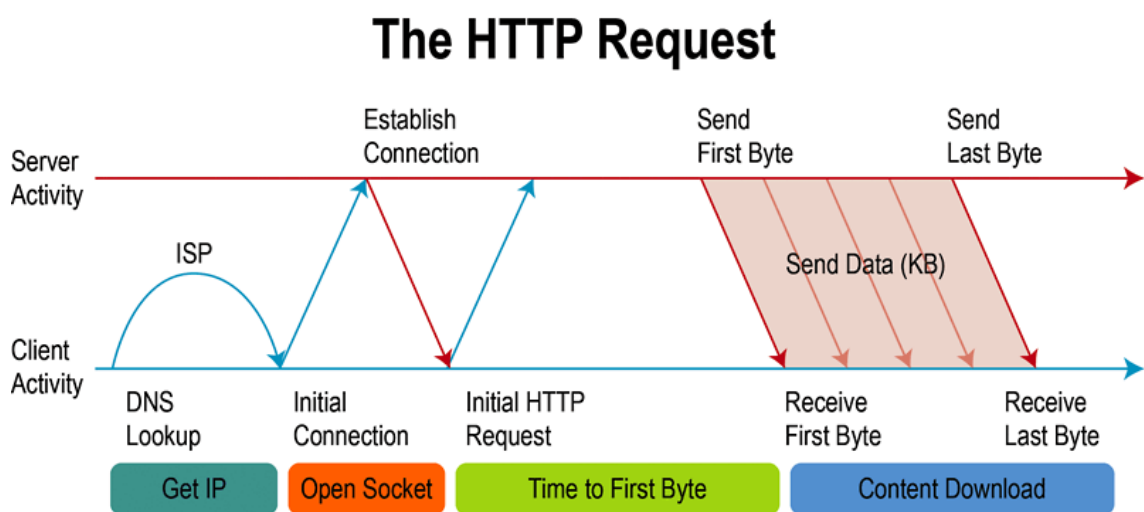
Kuviossa 3 on havainnollistettu sisällönhallintajärjestelmän avulla toimivan verkkosivun latautumiseen yleisesti tarvittavat komponentit ja latautumiseen vaikuttavat tekijät. Kaikkia kuvion komponentteja ei jokaisen sivun latauksessa tarvita, kun taas joissakin niitä tarvitaan vielä enemmän, kuin kuviossa näkyy. Kuviossa esitetystä komponenttien ja muiden tekijöiden kirjosta voidaan päätellä, ettei verkkosivun optimoinnille voi luoda selkeää kaavaa.



Kuvio 3. Verkkosivun latausreitti ja suoriutumiseen vaikuttavat komponentit.

Verkkosivun latausprosessissa selain lähettää palvelimelle useita pyyntöjä HTTP-protokollan välityksellä. Kuviossa 4 on esitetty, miten HTTP-pyyntöt hakevat tietoa

palvelimelta, tietokannoista ja verkkopalveluista, jotta käyttäjälle pystytään näyttämään toivottu sisältö [10]. Kuvion HTTP-pyyntö on ensimmäinen palvelimelle lähetetty pyyntö, ja se alkaa osoitteen selvittämisellä. Verkkosivun muut HTTP-pyyntöt suorittavat vain tiedon noutamista, mikä on havainnollistettu kuvion oikealla puoliskolla. Se, miten nopeasti palvelimelle lähetetyt pyynnot palauttavat vastauksena kokonaisen verkkosivun, riippuu välimuistin käytöstä, HTTP-pyyntöjen määrästä, teksti- ja kuvatiedostojen koosta, verkkosivun koodin oikeasta asettelusta, skriptien jäsentelystä, palvelimen ruuhkautumisesta ja käyttäjän fyysisestä sijainnista suhteessa palvelimeen [11]. Kaikkiin paitsi viimeiseen voidaan vaikuttaa optimoimalla verkkosivustoa.



Kuvio 4. HTTP-pyyntön kulku [12, s. 337].

Verkkosivun optimoinnilla tarkoitetaan siis www-sivun käytettävyyden, latausnopeuden ja kuorman kestävyuden parantamista. Optimointia voi tehdä sekä käyttäjän että palvelinpuolen näkökulmasta. Käyttäjän näkökulmasta hyvin optimoitu verkkosivu näyttää latautuvan selaimeen välittömästi. Verkkosivu, joka on optimoitu palvelinpuolelta, ei liiallisessa määrin kuormita palvelimen resursseja, ja välttää palvelimen kaatumiselta ruuhka-aikoina ja siten verkkosivun käyttökatkoksilta. [12, s. 155.] Tässä työssä keskitytään enimmäkseen sivuston nopeuden optimointiin käyttäjän kannalta katsottuna, mutta tehdään myös kevyt katsaus palvelinpuoleen eli verkkosivun kuormansietokyvyn optimointiin.

Optimoinnilla tavoitellaan mahdollisimman kevyttä ja jouhevasti toimivaa sivua, jotta käyttökokemus olisi mieluisa kaikilla selauslaitteilla. Kasvavassa suosiossa olevat mobiili-

lilaitteet tekevät optimoinnista vielä tärkeämmän osan verkkosivun kehittämistä, sillä mobiiliverkot ovat vielä melko hitaita verrattuna kotien laajakaistayhteyksiin. Mobiililaitteiden selaimet eroavat myös melkoisesti tietokoneiden selaimista, eivätkä ne näin ollen suoriudu yhtä monimutkaisista tehtävistä.

Optimointi on tärkeää myös liiketoiminnallisista syistä. Markkinoinnin parantamiseksi on tärkeää minimoida sivuston latautumisaika, jotta käyttäjän huomio ei kiinnittyisi muihin kuin verkkosivun sisältöön. Useimmat meistä ovat jo niin tottuneita verkon selailijia, että ideaalisen verkkosivun latautumisaikan tulisi tutkimusten mukaan olla noin yksi sekunti tai mieluiten vähemmän [13]. Myös Google palkitsee hyvin optimoidut verkkosivut, sillä vuodesta 2010 lähtien Google on ottanut verkkosivun latautumisnopeuden huomioon sivujen sijoituksessa hakutulosten joukossa [14].

### 3.2 Optimoinnin käytäntöjä

Optimointiprosessissa lähdetään liikkeelle sivuston nykyisen rakenteen analysoimisesta ja selvitetään, mitä sivustolta tullaan vaatimaan ja mistä voidaan tinkiä. Sivustolta vaadittava suoriutuminen tulee suhteuttaa sivuston laajuuteen ja käyttäjäkuntaan. Pienyrityksen ”käyntikorttisivu” ei todennäköisesti joudu samanlaisen paineen alle kuin suuremman yrityksen asiakaspalvelusivusto, ja silloin palvelimen ruuhkautumisesta ei tarvitse huolehtia. Ennen optimointia on siis hyvä arvioida, mikä verkkosivun latautumisaikan tulisi suurin piirtein olla ja montako samanaikaista käyttäjää sen tulisi sietää.

Verkkosivun rakenteen nykytilan analyysi on helppo tehdä verkosta löytyvillä ilmaisilla työkaluilla. Käyttäjäpuolen optimointia varten Google tarjoaa Page Speed -nimisen palvelun, joka analysoi sivuston rakennetta ja pisteittää sen tiettyjen sääntöjen perusteella [11]. Yagoon Yslow-palvelu on samankaltainen, sen säännöt tosin eroavat Googlen säännöistä hiukan [10]. Molemmat palvelut kertovat kuitenkin, missä ongelmakohtat sijaitsevat, ja suosittelevat korjaustoimenpiteitä selkokielisesti, Page Speed aavistuksen tarkemmin kuin Yslow.

Palvelimen kuormansietokykyä testaavat analysointityökalut ovat usein maksullisia, varsinkin mikäli tuloksista halutaan visuaalista raportointia. Yksi markkinoiden johtavista työkaluista on Load Impact. Load Impactin avulla testattavalle verkkosivulle voi si-

muloida suurenkin määrän virtuaalikäyttäjiä toteuttamaan valittuja tehtäviä. Ohjelma lisää verkkosivun kuormaa asteittain, jolloin ylläpitäjän on helppo nähdä, minkälaisen kuormituksen alla sivu alkaa hidastua tai jopa kaatuu.

Miellyttävä ja helppokäyttöinen käyttöliittymä ja kattava visuaalinen raportointi ovat Load Impactin vahvuuksia [15]. Ilmaisiakin testivälineitä on tarjolla, kuten Siege ([www.joedog.org/siege-home](http://www.joedog.org/siege-home)) ja Pylot ([www.pylot.org](http://www.pylot.org)). Nämäkin testaavat millä tahansa palvelimella sijaitsevan verkkosivun kestävyyttä ruuhkatilanteessa, mutta ohjelmien käyttö tapahtuu komentoikkunassa. Pylotille on saatavilla myös graafinen käyttöliittymä, mutta sen epävakaisuus tekee siitä käyttökelvottoman. Sekä Siege että Pylot luovat raportin tehdyistä testeistä, mutta nekin ovat pelkkiä tekstiversioita, joiden sisältöä tulee osata tulkita voidakseen tehdä johtopäätöksiä palvelimen suoriutumisesta.

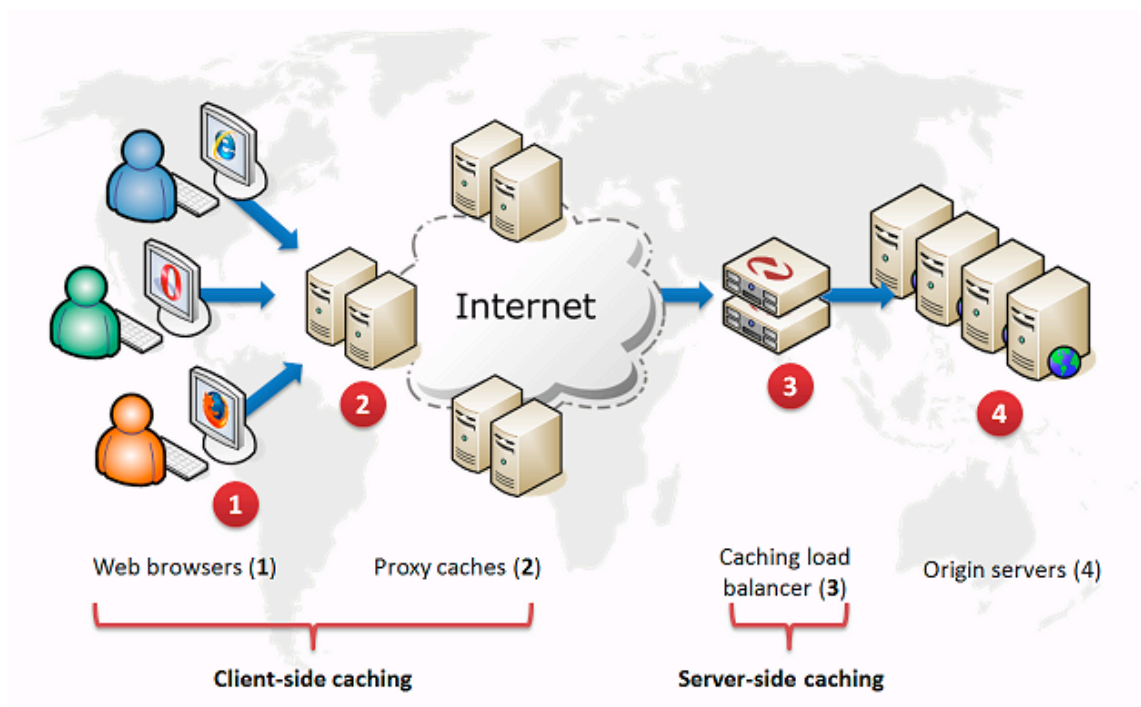
Sivuston nykytilanteen analysoimisen jälkeen verkkosivun heikot kohdat ovat selvillä ja voidaan siirtyä optimointitoimenpiteisiin. Tehtävät toimenpiteet vaihtelevat hyvinkin paljon riippuen sivuston luonteesta. SuVillan tapauksessa ennakoitiin, että olisi keskityttävä erityisesti sivuston satojen kuvien optimointiin. Seuraavissa luvuissa käydään läpi keskeisimmät asiat, jotka on hyvä huomioida verkkosivuja optimoitaessa.

### Välimuistin käyttö

Optimoinnin tärkein toimenpide on välimuistin aktivoiminen. Välimuisti toimii verkkosivun väliaikaisena tietovarastona, jonne tallennetaan kertaalleen ladattua tietoa helpommin ja nopeammin uudelleen esitettäväksi. Välimuistiin tehdään paikallinen kopio verkkosivusta tai sen osista, ja se säilytetään määritetyn voimassaoloajan puitteissa. Voimassaoloaika on tärkeä tieto, jonka tulee sisältyä HTML-dokumenttiin, jotta vältetään vanhentuneen sisällön näyttämistä käyttäjille. Tiedon voimassaoloajalle ei ole ylä- tai alarajaa, vaan se tulee valita verkkosivun sisällön mukaan. Usein päivittyvällä, dynaamisella sivulla välimuistiin tallennetaan tietoa lyhyiksi ajoiksi, muutamiksi minuuteiksi kerrallaan. Passiivisemmin päivittyvillä sivuilla tietoa säilytetään pidempään, yleensä korkeintaan muutamia päiviä. Välimuistiin säilötyn tiedon vanhentumisen jälkeen se joudutaan tarkistuttamaan verkkosivun palvelimella ja lataamaan uudestaan, jos tieto on muuttunut. [12, s. 261–270.]

Välimuisteja on kahta eri tyyppiä: käyttäjäpuolen välimuisti ja palvelinpuolen välimuisti. Välimuisti voi sijaita selaimessa, välityspalvelimella tai verkkopalvelimella. Selaimen välimuisti sijaitsee käyttäjän omalla tietokoneella, ja se on automaattisesti päälle kytketty. Sitä käytettäessä palvelimen kuorma pienenee, sillä verkkosivulle lähetettävät pyynnöt vähenevät. Välityspalvelimella sijaitseva välimuisti on useimmiten operaattorin puolesta kytketty päälle sellaisille palvelimille, joiden kautta kulkee paljon liikennettä. Palvelinpuolen välimuisti palvelee lähinnä verkkosivun ylläpitäjää, ja se tallentaa muistiinsa koko verkkosivun sisällön. [12, s. 261–270.]

Kuviossa 5 nähdään, miten välimuistit sijoittuvat muihin tietoverkon komponentteihin nähden ja miten käyttäjäpuolen (client-side) ja palvelinpuolen (server-side) välimuistit jakautuvat eri puolille internetiä. Kuviossa on myös havainnollistettu käyttäjien liikennöinti saman välityspalvelimen kautta internetiin.



Kuvio 5. Välimuistien sijainti tietoverkossa [16].

Käyttäjäpuolella välimuistiin tallentuu tietoja yksittäisen käyttäjän parhaillaan käynnissä olevasta istunosta verkkosivulla. Tässä hyödynnetään yleensä selaimen välimuistia, joka varastoi väliaikaistietoa käyttäjän tietokoneelle ja nopeuttaa siten tiedon noutamista. Toinen käyttäjäpuolen välimuisti sijaitsee julkisilla välityspalvelimilla, joita on esimerkiksi suurilla yrityksillä tai operaattoreilla. Välityspalvelin on osa tietoverkon reittiä, jota



pitkin data kulkee. Päälle kytkettynä verkkosivun osia tallentuu välityspalvelimen välimuistiin. Tämä hyödyttää paitsi yhtä sivulle palaavaa käyttäjää, myös muita, jotka saapuvat sivulle saman välityspalvelimen kautta eli samaa reittiä pitkin. [12, s. 261–270.]

Palvelimen välimuistia käytetään tallentamaan koko verkkosivun sisällöstä versio, joka näytetään kaikille käyttäjille. Tämä mahdollistaa sivun muokkaamisen palvelimen puolella ilman, että käyttäjä huomaa jatkuvia muutoksia. Muutokset tulevat näkyviin vasta, kun verkkosivun ylläpitäjä tyhjentää palvelimen välimuistin ja uusi versio sivusta tallentuu. Tietokannan välimuisti on yksi palvelinpuolen välimuisteista, ja WordPressiä käytettäessä se on tärkeässä roolissa. Tietokannan välimuistiin voidaan tallentaa sivuilla toistuvien kyselyiden tuloksia, jottei tietokantaan aina tehtäisi uutta kyselyä. [2. s. 278–279.]

#### Ulkoiset tiedostot

Usein suuri rasite verkkosivujen latautumisenopeudelle on liian monien HTTP-pyyntöjen määrä. Jokainen HTML-dokumentin ulkopuolinen tiedosto haetaan palvelimelta erillisellä HTTP-pyyntöllä, johon palvelin vastaa lähettämällä pyydetyn tiedoston palvelimelta selaimelle. Ulkoiset tiedostot voivat olla esimerkiksi skriptitiedostoja tai kuvia. Pyyntöjen määrää voidaan huomattavasti vähentää yhdistämällä tiedostoja, ja pyyntöjä voidaan nopeuttaa tekemällä haettavista tiedostoista pienempiä ja tiiviimpiä. [17.]

Selain tulkitsee verkkosivun lähdekoodin aina ylhäältä alaspäin ja näyttää käyttäjälle palvelimelta tulleen datan heti saatuaan sen. Tärkein tapa, jolla käyttäjä seuraa sivun latautumista, on ulkoasun näkyville tuleminen. On siis tärkeää ladata CSS-tyylitiedostot (Cascading Style Sheet) heti HTML-dokumentin alussa, HEAD-osiossa, jotta sivusto näyttäisi latautuvan nopeasti. CSS-tiedostojen koodin loogisuutta on myös syytä tarkkailla optimoinnin kannalta. Selain lukee CSS-syntaksia oikealta vasemmalle, jolloin avainvalitsimien, eli viimeiseksi oikealle jäävien valitsimien, tulisi olla mahdollisimman uniikkeja. Kuviossa 6 on havainnollistettu, miten saman elementin tyylimäärittelyn voi kirjoittaa eri tavoin. Hyviksi merkityillä tavoilla elementeille käytetään luokkaa tai uniikkia ID:tä. Luokka-arvoja on hyvä käyttää tapauksissa, joissa samaa tyylimäärittelyä tarvitaan usealle vakioelementille, ID taas palvelee parhaiten yksittäistapauksissa. Kummassakin tapauksessa tavoite on saada CSS-syntaksi mahdollisimman lyhyeksi ja yksiselitteiseksi. [18.]

- X `ul li a.red {color: red;}`
- ✓ `a.red {color: red;}`
- ✓ `#red {color: red;}`

Kuvio 6. CSS-tyylimäärittelyjen vertailu [18].

Huono tapa on määrittää sivustolla esiintyvien vakioelementtien, kuten listojen, tyylejä loogisilla, mutta turhan monilla alavalitsimilla. Rungas määrä alavalitsimia tarkoittaa pidempää tyylitiedoston tulkitsemisaikaa selaimelle. [18.]

JavaScript-tiedostot sisältävät selaimelle annettavia käskyjä, joiden avulla sivusto muuttuu interaktiivisemmaksi. Päinvastoin kuin tyylitiedostot, on toiminnallisuuksista vastaavat JavaScript-tiedostot suositeltavaa kirjoittaa vasta HTML-dokumentin loppuun, ennen BODY-osion päättymistä. Ajatuksena on, että käyttäjä aloittaa toiminnot verkkosivulla vasta, kun sivu on muilta osin latautunut. JavaScriptin liittäminen sivun loppuun on suotavaa myös siitä syystä, että JavaScript-tiedostoa suoritettaessa selain ei aloita uuden HTTP-pyyntöä, vaan odottaa, että skripti on käsitelty. [10.]

Sekä CSS- että JavaScript-tiedostot voidaan tiivistää, jotta optimointi olisi entistä tehokkaampaa. Tiivistämisellä tarkoitetaan turhien tekstinvälien ja rivinvaihtojen poistamista ja tiedostojen yhdistämistä mahdollisuuksien mukaan. Sisällönhallintajärjestelmät käyttävät lisäosien takia usein enemmän kuin yhtä tyylitiedostoa, mikä tarkoittaa yhtä montaa HTTP-pyyntöä. Tiedostojen yhdistämisen voi tehdä käsin, mutta helpoimmalla pääsee, jos käyttää esimerkiksi Googlen kehittämää Minify-työkalua, joka suorittaa tiivistämisen hetkessä. [19.]

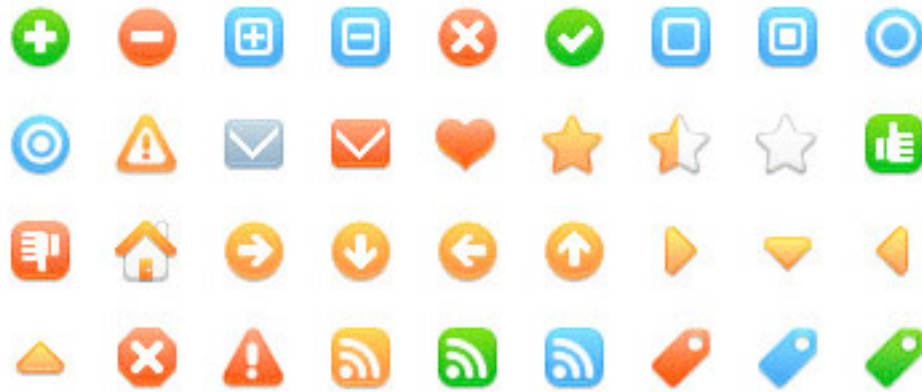
Kuvien optimointi

Verkkosivulla näytettäviä kuvia voidaan optimoida pakkaamalla niitä. Usein kuvat on tallennettu kuvankäsittelyohjelmalla, jolloin niiden mukana tallentuu paljon käyttötarkoituksen kannalta turhaa dataa. JPEG (Joint Photographic Experts Group) on yleisin tallennusmuoto kuville. Se pakkaa kuvia häviöllisellä pakkausmenetelmällä, jolla voi helposti saavuttaa 20:1-pakkauksen hävittämättä kuvasta värejä, kuitenkin muutta-

matta sitä juurikaan erinäköiseksi. [20.] Silti myös JPEG-kuvat sisältävät paljon ylimääräistä dataa, jonka olemassaolo ei tuo lisäarvoa verkkosivulla näytettävälle kuvalle. Kuvan ylimääräinen data saattaa sisältää esimerkiksi metatietoja kuvan alkuperästä ja suuremmasta värimaailmasta, kuin mitä selain pystyy näyttämään. [11.]

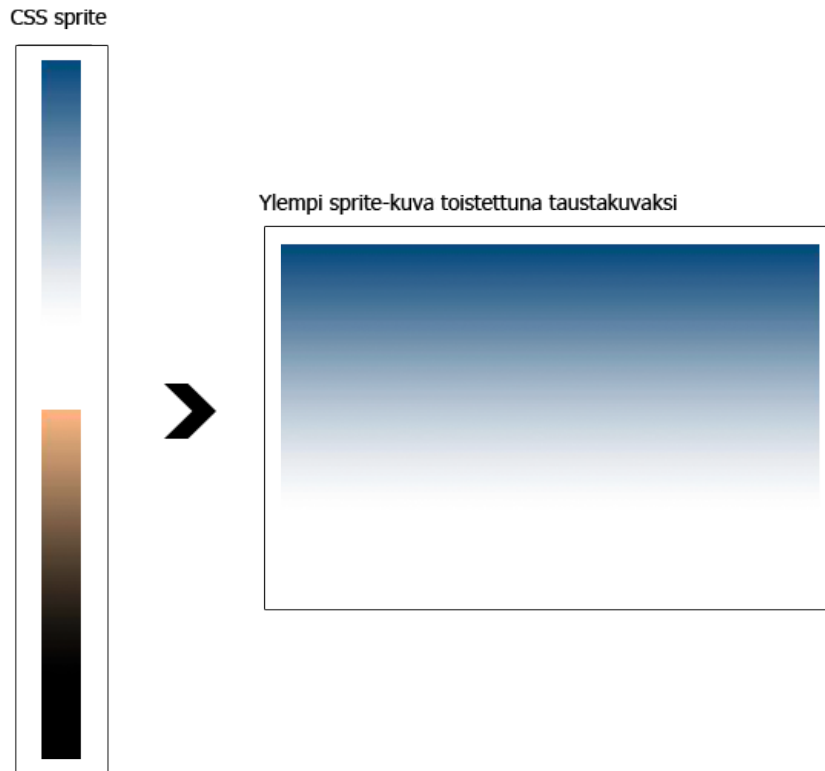
Kuvien ylimääräinen data olisi lähes mahdotonta poistaa käsin, niinpä Yahoo on kehittänyt Smush.it-palvelun, jolla kuvien turhista metatiedoista pääsee eroon. Selaimessa toimivaan Smush.it-palveluun voi syöttää kerralla koko verkkosivun kuvat, jotka palvelu käy läpi erilaisten skriptien avulla ja tarjoaa tilalle optimoidut versiot kuvista. Smush.it optimoi kuvia hävittämällä turhaa dataa kuvista ilman, että kuva kärsii laadullisesti. GIF-kuvat Smush.it-palvelu muuttaa mahdollisuuksien mukaan PNG-päätteisiksi. PNG on tilansäästöä ja kuvan laatua ajatellen paras tallennusmuoto tällä hetkellä, ja lisäksi se tukee läpinäkyviä pikseleitä. Smush.it-palvelusta saa optimointitoimenpiteen jälkeen raportin, joka kertoo, minkä verran kuvien koko on optimoinnin aikana pienentynyt. [21.]

Kuten tyylitiedostojen ja skriptien tapauksessa, myös kuvien yhdistäminen nopeuttaa sivustoa. Verkkosivun ulkoasussa käytetään useimmiten samoja kuvia monella alisivulla, ja jokainen näistä haetaan palvelimelta erillisellä HTTP-pyyntöllä. Tältä voidaan välttyä yhdistämällä useita kuvia yhdeksi. Kuviossa 7 näkyy, miten useampi verkkosivun ulkoasuun käytettävä kuva voidaan yhdistää yhdeksi niin sanotuksi sprite-kuvaksi. Tällöin kuvaa käytetään monessa eri kohdassa, mutta se noudetaan palvelimelta vain kerran. Sprite-kuvasta näytetään määrättyssä kohdassa vain haluttu osa määrittelemällä osan koordinaatit CSS-tyylitiedostossa. Näin sivustoa selaava ei koskaan saa tietää, että kuvat on yhdistetty yhdeksi suuremmaksi kuvaksi. [12, s. 206.]



Kuvio 7. Sprite-kuva [22].

Sprite-kuvien käyttö on mahdollista vain, jos tietyt ehdot täyttyvät. Ensinnäkin kuvaa, jota toistetaan esimerkiksi sivun taustalla sekä vaaka- että pystysuunnassa CSS-syntaksin REPEAT-käskyn avulla, ei voida sisällyttää sprite-kuvaan, muuten koko sprite-kuva toistuisi. Sprite-kuvan osia voidaan toistaa vain joko vaaka- tai pystysuuntaan, ja sekin vaatii kuvien oikean asettelun sprite-kuvassa. Keskenään eri suuntiin toistettavia kuvia ei myöskään voida liittää samaan sprite-kuvaan, eli vaakasuunnassa toistettavat kuvat tulee sijoittaa yhteen sprite-kuvaan ja pystysuuntaan toistettavat toiseen. Kuvia voi siis toistaa jompaankumpaan suuntaan, täytyy vain muistaa, että mikäli sprite-kuvan osia toistetaan vaakasuuntaisesti, tulee siihen sisältyvien kuvien sijaita allekkain, ja jos niitä toistetaan pystysuuntaisesti, niiden tulee sijaita vierekkäin. Kuviossa 8 on havainnollistettu, miten vaakasuuntaisesti toistuvat kuvat asetellaan sprite-kuvassa allekkain ja vain toista osaa kuvasta toistetaan CSS:n avulla taustakuvan luomiseksi valitulle alueelle. Sprite-kuvaa ei ole mahdollista käyttää siinäkään tapauksessa, jos kuva on jonkin elementin taustakuvana, jonka sijainti on määritetty CSS:ssä POSITION-käskyllä. Samaa käskyä käytetään sprite-kuvan näytettävän osion sijainnin määrittelyyn. [12, s. 206–208.]



Kuvio 8. Sprite-kuvien asettelu.

Edellä lueteltuja ehtoja on mahdollista kiertää CSS-lausekkeilla, mutta ne eivät ole arvostettu keino optimoinnin maailmassa. CSS-lausekkeet ovat helppo tapa tehdä verkkosivun ulkonäkö dynaamiseksi esimerkiksi vaihtuvalla taustakuvalla, mutta lausekkeiden huono puoli on, että niiden paikkansapitävyys lasketaan uudelleen useammin kuin verkkosivua päivitetään. Joskus selain käy lausekkeet läpi jopa silloin, kun käyttäjä vain liikauttaa hiirtä, joten niistä saattaa muodostua suorituskyvyn suurkuluttajia. [10.]

### 3.3 Sisällönhallintajärjestelmän optimointi

Sisällönhallintajärjestelmää käyttävän verkkosivun optimointi eroaa sikäli tavallisen verkkosivun optimoinnista, ettei muutoksia kannata tehdä sen ydintiedostoihin, jotka sijaitsevat palvelimen juurikansiossa. Tavallisten verkkosivujen tapauksessa juuri näitä tiedostoja muokataan ja kevennetään, jotta sivusto saataisiin optimoitua. Sisällönhallintajärjestelmää käytettäessä ydintiedostot määrittelevät järjestelmän ja verkkosivujen keskeisimmät toiminnot, ja on suositeltavaa tyytyä siihen, miten järjestelmän kehittäjät ovat optimoineet ne. Useimmiten kehittäjäryhmä on niin laaja ja asiantunteva, että järjestelmän ytimen toiminta täyttää jo valmiiksi lähes kaikki optimoidun lähdekoodin

kriteerit. WordPress-sisällönhallintajärjestelmän ydintä päivitetään jatkuvasti ajantasaiseksi satojen kehittäjien toimesta ja sivuston ylläpitäjälle jää vastuu verkkosivulla käytettävän teeman ja ulkoisten komponenttien optimimisesta [4].

Teemat ovat WordPress-sivuston käyttöliittymä ja graafinen ulkoasu. Ne koostuvat vähintään yhdestä CSS-tiedostosta, joka sisältää sivuston tyylimäärittelyt, ja muutamasta PHP-tiedostosta, jotka toimivat sivupohjina. Teeman PHP-tiedostot sisältävät lyhyitä mutta tärkeitä koodinpätkiä, joita käytetään eräänlaisina paikan pitäjinä. Paikan pitäjät osoittavat palvelimen PHP-suorittimelle, mihin kohtaan sivua tietokannasta haettava sisältö lisätään, ennen kuin sivu näytetään käyttäjälle selaimessa. [23, s. 25–26.]

Teemoja voidaan kehittää loputtomasti, ja tästä syystä myynnissä olevat teemat sisältävät kymmenittäin PHP- ja JavaScript-tiedostoja, jotta toimintoja olisi mahdollisimman paljon ja yksi teema sopisi mahdollisimman laajalle yleisölle. WordPressin alkuperäisen asennuksen mukana tulee kaksi maksutonta teemaa, jotka ovat WordPressin kehittäjäyhteisön tekemiä. Ilmaisen ja juuri oikeanlaisen teeman saa tekemällä sellaisen itse. Mukana tulevat teemat ovat hyviä lähtökohtia oman teeman rakentamiseen, sillä niihin ei ole upotettu mitään erikoisominaisuuksia eikä koodissa varmasti ole virheitä.

WordPressiin myytävät valmiit teemat tuovat paljon etuja verkkosivun rakentajalle, mutta optimoinnin kannalta ne voivat olla myös suuri kompastuskivi. Teemoja ei tehdä millään tietyllä reseptillä, eikä niiden tekijältä tai myyjältä edellytetä minkäänlaista todistusta pätevyystään. Näin ollen teeman rakenne saattaa olla ontuva ja esimerkiksi turhan tiedon karsiminen CSS-tiedostoista voi osoittautua hyvinkin hankalaksi.

Myös sisällönhallintajärjestelmän ylläpitopuolta on mahdollista optimoida. Yksi keino on antaa tietokannan välimuistin tallentaa myös kirjautuneen käyttäjän kyselyt, jolloin ylläpitosivujen latautuminen on sujuvampaa. Tämä voi kuitenkin aiheuttaa ongelmia muun muassa uloskirjautumisen kanssa, kuten on käynyt ilmi WordPress.org:n tukifoorumissa [24]. Toinen tapa on ladata selaimen verkkosivun ylläpitäjälle suunnattu lisäosa, joka tallentaa väliaikaistietoja, kuten kuvia, omalle tietokoneelle tehokkaammin kuin välimuisti ja pienentää siten latausaikoja. Sopivin vaihtoehto tähän olisi Googlen Gears, mutta siitä on nykyään saatavilla vain lähdekoodi, sillä suurin osa sen ominaisuuksista on liitetty Googlen Chrome-selaimen. Lisäosavaihtoehtoja Mozillan Firefoxiin on useita, ja niitä voi selata osoitteessa [addons.mozilla.org/fi/firefox](http://addons.mozilla.org/fi/firefox).

### 3.4 Sisällönhallintajärjestelmän lisäosien vaikutus optimointiin

WordPressin verkkosivuilla on 19 375 ilmaiseksi ladattavissa olevaa lisäosaa, jotka on mahdollista asentaa oman WordPress-sivuston yhteyteen [26]. Lisäosa on tavanomaisesti pienikokoinen sovellus, joka tuo sivustolle muutaman uuden ominaisuuden [25]. Lisäosia voi ladata joko WordPress-asennuksen sisällä, jolloin ne asentuvat sivustolle automaattisesti, tai vaihtoehtoisesti verkosta ja siirtää FTP-yhteyden avulla oman WordPress-sivuston Plugins-kansioon. WordPressin suosituimpia lisäosia ovat suorituskykyyn ja hakukoneoptimointiin liittyvät lisäosat. [26.]

Lisäosat olivat tärkeä osa päätöstä, joka johti WordPressin valintaan SuVillan sivuston sisällönhallintajärjestelmäksi. Oli tiedossa, että WordPress-järjestelmään on tarjolla ilmaisia lisäosia, jotka mahdollistavat sivuston kääntämisen monille eri kielille hyvin pienellä vaivalla. SuVillan sivuston kääntämisen ja jatkuvan päivittämisen tuli onnistua suomeksi, englanniksi ja espanjaksi, ja tehtävän toteuttamista varten sivustolle asennettiin helppokäyttöinen qTranslate-lisäosa, joka on saatavilla osoitteesta [www.wordpress.org/extend/plugins](http://www.wordpress.org/extend/plugins). QTranslate on tehokas lisäosa, jonka avulla sivujen sisällön kääntäminen sujuu mutkitta, mutta optimointia ajatellen se saattaisi tuoda lisäkuormitusta sivustolle, koska se joutuu kysymään palvelimelta, mikä kieli milloinkin on käytössä, ja noutaa sisältöä tietokannasta sen mukaisesti.

## 4 Yrityksen verkkosivun optimointiprosessi

### 4.1 Verkkosivun suorituskyky ja kuormituksen kesto

Ennen optimointitoimenpiteitä SuVillan verkkosivun suorituskyky mitattiin tarkkaan eri testaustyökaluilla. Useiden testaustyökalujen koekäytön jälkeen päädyttiin käyttämään Gossamer Threadsin tarjoamaa GTmetrix-testaussivustoa ([www.gtmetrix.com](http://www.gtmetrix.com)), joka testaa kohdesivun kerralla sekä Google Page Speedillä että Yagoon Yslow'lla. Raportointityyli GTmetrixin sivustolla on harvinaisen selkeä ja visuaalisesti hyvin havainnollistava. Lisäksi sivusto tallentaa aiemmin tehdyt testit ja näyttää verkkosivun optimoinnin kehityksen kuvaajana. [27.]

Toinen testaussivusto, AOL:n kehittämä Webpagetest.org, toimii vertailevana testaustyökaluna GTmetrixin rinnalla, sillä sen raportointiin sisältyi tarkempia tietoja HTTP-pyyntöjen kulusta. Webpagetest.org:n generoimassa vesiputoustyyppisessä raportissa saattoi erottaa yksittäisten kuvien ja muiden ulkopuolisten tiedostojen vaikutuksen latausnopeuteen hyvinkin tarkasti. Raportoinnin eroavaisuuksista huolimatta webpagetest.org arvioi verkkosivun suorituskykyä samoilla perusteilla kuin Page Speed. [28.]

Suorituskyvyn mittaamiseen käytetyt työkalut keskittyvät arvioimaan taulukossa 1 esitettyjen sääntöjen noudattamista verkkosivulla. Säännöt on listattu tärkeysjärjestyksessä, ja ne perustuvat Googlen ja Yagoon määrittämiin hyviin käytäntöihin, joita noudattamalla verkkosivun toiminta on sujuvaa. Taulukossa luetellut säännöt ovat suurempia kokonaisuuksia, jotka on todellisuudessa jaettu tarkentaviin alasääntöihin. Testatessaan verkkosivuaan edellä mainituilla työkaluilla ylläpitäjä saa palautetta sääntöjen noudattamisesta tai noudattamatta jättämisestä ja ehdotuksia siitä, miten mahdolliset komplikaatiot voidaan ratkaista. Tärkeimpien sääntöjen rikkomisesta menettää suorituskykyteisteissä enemmän pisteitä kuin vähemmän tärkeiden sääntöjen noudattamatta jättämisestä. [11.]



Taulukko 1. Suorituskyvyn mittaamiseen sovellettavat säännöt.

Sääntö	Selitys
Välimuistin optimoitu käyttö	Välimuistia tulisi hyödyntää mahdollisimman paljon datan noutamisen nopeuttamiseksi, mutta harkiten, jottei käyttäjälle näytetä vanhentunutta tietoa.
Palvelinpyynnön edestakaisen matkan kesto	HTTP-pyyntöjen määrä tulisi minimoida ja ulkoisia tiedostoja mahdollisuuksien mukaan yhdistää.
Pyyntöjen koon minimointi	HTTP-pyyntö tulisi mahtua yhteen datapakettiin (noin 1500 tavua), eli pyynnön mukaan lähtevien evästeiden käytön tulee olla harkittua.
Optimoitu verkkosivulla hyödynnettävä kuorma	Turha data tulisi poistaa ulkoisista tiedostoista ja latautumisjärjestys tulisi optimoida; ulkonäkö ensin, toiminnot viimeiseksi.
Selaimen renderöintiprosessin helpottaminen	Ulkonäölliset seikat tulisi määrittää tarkasti: CSS-valitsimet, kuvien koot, käytettävä kirjaimisto jne.

Testaustyökaluja valittaessa tehtiin useita kokeiluja, miten työkalut käyttäytyvät ja kuinka hyvin niiden antamiin tuloksiin voitaisiin luottaa. Tärkeä huomio, joka kokeiluissa kävi ilmi oli, että verkkosivun välimuisti tuli tyhjentää juuri ennen testausta, jotta saatiin paikkansapitäviä tuloksia. Tyhjentämättä jäänyt välimuisti häiritsi testausta tuntemattomasta syystä, jolloin testaustulokset näyttivät todellista huonommilla. Ainoa työkalu, joka teki oletuksena kaksi vertailevaa testiä, oli Webpagetest.org. Se testasi sivuston ensin puhtaalta pöydältä, minkä jälkeen se kävi läpi vielä välimuistista saadun sivuston. Näin saatiin selkeä näyttö siitä, miten välimuistin käyttö vaikuttaa sivuston uudelleenlatautumiseen.

Kokeiluissa huomattiin myös, ettei GTmetrixin ja Wepagetest.org:n antamiin sivuston latautumisaikojen tuloksiin voitu luottaa. Vaikutti siltä, että sen palvelimilla oli toisinaan ruuhkaa, joka aiheutti viivettä latautumisaikoihin, vaikka verkkosivulle ei ollut tehty muutoksia. Tämä johti siihen johtopäätökseen, että verkkosivun latautumisaika on ennen kaikkea riippuvainen käyttäjän internetyhteyden nopeudesta ja hänen palveluntar-

joajansa palvelimista ja reitityksistä tietoverkossa. Mitä useamman välityspalvelimen kautta reitti vierailtavalle verkkosivulle kulkee, sitä hitaampaa matkan teko on [29].

Latautumisaika oli kuitenkin tärkeä mitta, jonka mahdollisimman tarkka arvo tarvittaisiin kuvastamaan loppukäyttäjän selailukokemusta. Niinpä mittaustyökaluihin lisättiin suomalaissyntyisen Sami Nurmen luoma Pingdom-työkalu, joka aiemmin mainittujen mittaustyökalujen tapaan toimii selaimessa ja testaa verkkosivun suoriutumista. Pingdom on laaja kokoelma verkkosivujen toiminnan optimointiin liittyviä työkaluja, mutta alun perin yrityksessä on keskitytty sivujen pystyssä pysymisen tarkkailuun ja latautumisnopeuden mittaamiseen. [30.] Pingdomin sivuilla luvataan, että yritys käyttää erityisesti testaukselle omistettuja palvelimia [31]. Toisista työkaluista poiketen Pingdom näyttää sivuston latauksen etenemisen reaaliaikaisesti, mikä tuntui luotettavalta.

#### Kuormituksen sietokyvyn mittaaminen

Verkkosivun kuormituksen sietokykyä voi mitata vain yhdellä tavalla: tuomalla sivulle kymmeniä tai jopa satoja käyttäjiä samanaikaisesti. Kuormituksen kesto on mahdollista testata siirtymällä sivulle usealla tietokoneella samanaikaisesti, mutta helpompaa lienee hyödyntää virtuaalikäyttäjiä. Virtuaalikäyttäjät ovat käytännössä vain pyyntöjä, jotka lähetetään verkkosivun palvelimelle. Ne vastaavat tavallisia käyttäjiä, mutta toimivat ja toistavat automaattisesti määrättyjä tehtäviä sen mukaan, kuin niitä on käsketty. [32.]

SuVillan verkkosivujen kuormituksen sietokykyä oli tarkoitus testata maksullisella Load Impact -työkalulla, mutta palomuuriongelmiensa takia tämä laadukas testaustyökalu jouduttiin lopuksi hylkäämään. Testausta kokeiltiin monilla eri työkaluilla, joiden toiminnat ja saatavuus on selvitetty taulukossa 2. Taulukossa esitetyt testaustyökalut ovat ilmaisia avoimen lähdekoodin sovelluksia, sillä yhden epäonnistuneen yrityksen jälkeen testaukseen ei haluttu enää haaskata rahaa. Monet taulukossa mainitut testaustyökalut toimivat vain komentokäyttöliittymässä, mutta totuttelun jälkeen se tuntuu jopa paremmalta ratkaisulta kuin graafiseen käyttöliittymään toteutettu sovellus.

Taulukko 2. Avoimen lähdekoodin kuormituksen sietokyvyn testaustyökaluja ja niiden ominaisuudet.

Nimi	Käyttö ja saatavuus
Pyload	Melko yksinkertainen testaustyökalu graafisella käyttöliittymällä. Huonona puolena epävakaa toiminta. Saatavilla osoitteesta <a href="http://www.pyload.org/">http://www.pyload.org/</a> .
Siege	Helppokäyttöinen testityökalu, verkkosivuilla hyvä dokumentaatio ohjelman käytöstä. Saatavilla osoitteesta <a href="http://www.joedog.org/siege-home/">http://www.joedog.org/siege-home/</a> .
Apache JMeter	Selainta muistuttava näkymä, jossa tehdään testaussuunnitelma ja suoritetaan testi. Kehittynyt ohjelma, käyttö hidasta. Saatavissa osoitteesta <a href="http://jmeter.apache.org">http://jmeter.apache.org</a> .
The Grinder	Java-pohjainen testaustyökalu, joka pystyy tekemään rasiustestejä monille eri verkkopalveluille. Huono dokumentaatio. Saatavilla osoitteesta <a href="http://grinder.sourceforge.net">http://grinder.sourceforge.net</a> .
JCrawler	Java-pohjainen ohjelma, jolle voi antaa useita verkko-osoitteita läpikäytäväksi. Melko vaikeakäyttöinen ohjelma. Saatavissa osoitteesta <a href="http://jcrawler.sourceforge.net/">http://jcrawler.sourceforge.net/</a> .

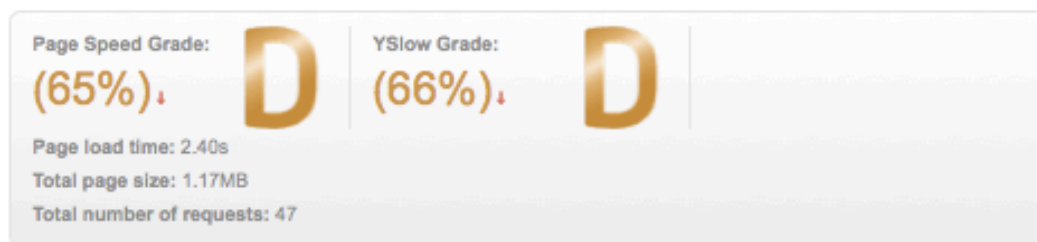
Testaustyökalujen seasta parhaaksi vaihtoehdoksi valikoitui jo aiemmin työssä mainittu Siege. Sen toiminta oli vakaan oloista, ja sen konfigurointi ja käyttö oli helppoa. Siegen asennus sujui yhdellä komennolla ja kesti muutamia sekunteja. Asennuksen jälkeen Siege asetettiin lähettämään SuVillan osoitteeseen haluttu määrä virtuaalikäyttäjiä (käskyllä `-c`) ja valittiin testille sopiva kesto (käskyllä `-t`) sekä virtuaalikäyttäjien pyyntöjen toistamiskerrat (käskyllä `-r`) [33].

Palvelimen suorituskykyyn vaikuttaa myös usealle käyttäjälle jaettu palvelintila. SuVillan verkkosivu sijaitsee kaupallisessa, vuokratussa palvelintilassa, joka todellisuudessa on vain yksi kansio yhdellä isolla palvelimella, jonka tilaa on vuokrattu sadoille muille käyttäjille. Näin ollen palvelimen resurssit jakaantuvat ja palvelimella sijaitsevien muiden verkkosivujen ruuhkauput vaikuttavat myös SuVillan sivustoon. Jaetun palvelintilan toimivuuteen ei valitettavasti voi itse vaikuttaa muutoin, kuin siirtämällä oma sivusto sille omistetuille palvelimelle. [34.]

## 4.2 Mittaustulokset ennen optimointia

Tutkimuksessa testattaviksi sivuiksi valittiin SuVillan verkkosivun etusivu, myytäviä asuntoja listaava sivu ja yksittäisen myyntikohteen sivu. Nämä sivut ovat verkkosivuston eniten selatut sivut, kuten liitteen 1 Google Analytics -kävijäraportista voi huomata. SuVillan sivuston toiminta oli lähtökohtaisesti kohtalaisen nopeaa, kun käytössä oli laajakaistayhteys. Sivuston toimintanopeutta haluttiin kuitenkin kiihdyttää esimerkiksi mobiilikäyttäjää varten. SuVillan www-palvelin oli aina toiminut moitteetta, mutta koska yrityksen toiminta laajentuu lähivuosina Kanariansaarille, oli tärkeää testata myös palvelimen kuorman kesto, jotta voitiin taata sujuvasti toimiva sivusto suuremmallekin käyttäjämäärälle.

Suorituskykymittausten tulokset GTmetrixin testissä vastasivat odotuksia. Otos etusivun testausraportista kuviossa 9 todistaa, että sivun optimointi on keskitason alapuolella, kouluarvosanalla mitattuna D, kun asteikko on A–F. Sivuston koko, 1,17 MB, on turhan paljon jopa laajakaistayhteydellä ladattavaksi. Tuollainen koko vastaa puolikasta musiikkikappaleen kokoa, eikä sellaisen latautuminen suju muutamassa sekunnissa läheskään kaikilla nykypäivän yhteyksillä. Myös etusivun HTTP-pyyntöjen määrä on korkeahko, 47. Kuviosta 9 nähdään, että suurimpana puutteena testissä todetaan välimuistin käytön olevan pois päältä sivustolla. Tämä tarkoittaa sitä, ettei WordPress-sovellus käytä minkäänlaista välimuistia toistuvan tiedon tallentamiseen palvelimella tai tietokannassa. Myös kuvien optimointi on arvosteltu ala-arvoiseksi, sillä niitä ei ollut pakattu lainkaan pienempään muotoon, kuin mitä ne ovat alun perin olleet. Huonoimman mahdollisen kouluarvosanan saavat myös JavaScript- ja CSS-tiedostot sekä GZIP-pakkausmenetelmän käyttämättä jättäminen palvelimen tiedostoille. GTmetrix-työkalulla suoritettujen lähtötilanteen mittauksen laaja tulosraportti on liitteessä 1.



### Breakdown

RECOMMENDATION	GRADE	TYPE	PRIORITY
Leverage browser caching	F (11)	Server	High
Defer parsing of JavaScript	F (35)	JS	High
Combine images using CSS sprites	F (37)	Images	Medium
Optimize Images	F (40)	Images	High
Enable gzip compression	F (44)	Server	High
Minify CSS	F (49)	CSS	High
Minify HTML	D (53)	Content	High
Specify image dimensions	C (70)	Images	High
Use efficient CSS selectors	F (0)	CSS	Low
Minify JavaScript	C (78)	JS	High
Optimize the order of styles and scripts	B (85)	CSS/JS	High
Remove unused CSS	C (74.3)	CSS	Low
Remove query strings from static resources	A (92)	Content	High

Kuvio 9. Ote www.suvilla.com-sivuston testituloksista lähtötilanteessa.

Sivuston muiden testauskohteiden, asuntojen listaussivun ja yksittäisen kohteen sivun, lähtötilannemittaukset vastasivat etusivun tilannetta. Huomioitavaa on, että asuntojen listaussivun HTTP-pyyntöjen määrä oli 119, ja yksittäisen kohteen sivullakin niitä lähetettiin 71. Muutoin suurimmat puutokset optimoinnin kannalta olivat näilläkin sivuilla välimuistin puute ja kuvien pakkaamatta jättäminen.

Pingdomin antamat mittaustulokset ennen optimointia mukailivat GTmetrixin tuloksia. Myös sivuston latausaika oli tällä kertaa samaa luokkaa. Pingdomin tulokset kaikilta testattavilta sivuilta on yhdistetty yhdeksi kuvaksi kuviossa 10. Kuvan arvoja vertailtaessa voidaan todeta, että HTTP-pyyntöjen määrä vaikuttaa latausnopeuteen oleellisesti. Sekä Pingdomin että GTmetrixin tulokset paljastavat, että erityisesti asuntojen listaussivun latausaikaan on kiinnitettävä huomiota sen huomattavan korkean HTTP-pyyntöjen määrän takia, mikä venyttää sivun latautumisaikaa 4,4 sekuntiin. Syy niin

moniin pyyntöihin tosin selittyy siten, että asuntoja listattaessa jokainen kohde sisältää kuvan ja se haetaan näytille HTTP-pyyntöllä. Tähän ei ole suoranaista korjaustoimenpidettä, mutta välimuistin aktivoiminen takaa, ettei niin monia pyyntöjä suoriteta toistuvasti.



www.suvilla.com		www.suvilla.com/asunnot		www.suvilla.com/property/las-gaviotas-calahonda/	
Requests	47	Requests	119	Requests	71
Page load time	2.38 sec	Page load time	4,40 sec	Page load time	2.48 sec
Page size	1.7 MB	Page size	1.70MB	Page size	1.29MB
Performance grade	85/100	Performance grade	62/100	Performance grade	80/100

Kuvio 10. Pingdomin mittaustulokset ennen optimointia.

Palvelimen kuormituksen sietokykyä testattiin Siegellä lähetettyjen virtuaalikäyttäjien avulla. Kuvio 11 on otos Siegellä tehdystä rasiustestistä, jossa SuVillan sivustoa ruuhkautettiin 30 yhtäaikaishella virtuaalikäyttäjällä ja 10 toistuvalla pyynnöllä käyttäjää kohti. Kuviossa näkyvän raportin arvoista tärkeitä ovat *Availability*, eli tavoitettavuus, joka kertoo sivun tavoitettavuuden testin aikana, *Response time*, eli vasteaika, josta näkee, kuinka nopeasti simuloitujen käyttäjien pyyntöihin on vastattu, *Transaction rate*, eli tapahtumankäsittely, joka ilmaisee sekunnin aikana käsitellyt pyynnöt, ja *Concurrency*, eli samanaikaisuus, josta ilmenevät sekunnin aikana käsitellyt pyynnöt suhteessa testiin kuluneeseen aikaan, ja tässä pienempi luku on parempi. [33.]

```

Heksun-MacBook:siege Heksu$ siege -r10 -c30 www.suvilla.com
** SIEGE 2.72
** Preparing 30 concurrent users for battle.
The server is now under siege..      done.

Transactions:          164 hits
Availability:          54.67 %
Elapsed time:          22.88 secs
Data transferred:     4.90 MB
Response time:         2.65 secs
Transaction rate:      7.17 trans/sec
Throughput:            0.21 MB/sec
Concurrency:           18.96
Successful transactions: 164
Failed transactions:   136
Longest transaction:   3.85
Shortest transaction:  0.34

```

Kuvio 11. Siege-työkalulla tehty ruuhkauttamistesti.

Kuvion 11 rasiustestauksen tuloksista näkee, ettei SuVillan sivuston kestävyttä ole optimoitu. Rasiustestauksen aikana palvelinta kuormitettiin porrastetusti ensin 10:llä, sitten 30:llä ja lopuksi 50 käyttäjällä samanaikaisesti. Kaikki edellä luetellut tärkeät arvot ovat kuvan testissä anteeksiantamattoman huonoja, ja erityisen huolestuttavaa on, että vain 30 käyttäjää voi aiheuttaa sivustolla käyttökatkoksen, kuten tavoitettavuuden arvosta käy ilmi. 50 yhtäaikaisen käyttäjän aikaansaama kuorma sai tavoitettavuuden arvon romahtamaan reiluun 20 prosenttiin, mistä voidaan päätellä, että sivusto luultavasti kaatuu kokonaan jo 100 samanaikaisen käyttäjän aiheuttamasta kuormasta. Kuvassa myös näkyvä melko hidaskasteaika ja korkea samanaikaisuusarvo kertovat liian raskaasta sivustosta, eli sivuston optimoiminen pienempään kokoon on ensisijaisen tärkeää.

#### 4.3 Optimoinnin tavoitteet

SuVillan verkkosivun lähtötilanteen mittaustuloksille tarvittiin vertailuarvoja, jotta voitiin luoda tavoitteita optimoinnille. Hyviä vertailuarvoja tarjosi Googlen toukokuussa 2010 julkaisema tutkimus verkkosivujen sisällöstä. Tutkimuksessa oli käyty läpi miljoonia verkkosivuja, mukaan lukien sivujen käyttämät ulkoiset resurssit, kuten esimerkiksi kuvat, videot ja tyyli tiedostot. Tutkimuksen raportissa tulokset oli jaettu suosittujen

sivujen tuloksiin ja kaikkiin tuloksiin. SuVillan verkkosivujen vertailuarvoiksi valittiin kaikki tulokset, sillä sivusto voidaan luokitella vähemmän suosittujen kategoriaan. [35.]

Taulukossa 3 nähdään Googlen tekemän tutkimuksen tulokset, jotka ovat tämän työn kannalta keskeisimpiä arvoja, ja myös SuVillan etusivun vastaavat arvot. Googlen tutkimuksessa ei selvitetty sivustojen latautumisenopeutta todennäköisesti juuri sen takia, että se on riippuvainen niin monesta tekijästä. Verrattaessa tutkimuksen tuloksia SuVillan arvoihin sivuston optimointi asettuu kaiken kaikkiaan jonkin verran keskitasoa huomomalle sijalle, kuten aiemmin GTmetrixin antamasta kouluarvosanasta pääteltiin. SuVillan verkkosivun koko on huomattavan suuri, kuten myös sivulla käytettävien kuvien yhteenlaskettu koko. Ulkopuolisten resurssien käyttö, eli HTTP-pyyntöjen määrä, on SuVillan etusivulla samaa luokkaa kuin tutkituilla sivuilla keskimäärin.

Taulukko 3. Googlen tutkimustulosten vertailu SuVillan vastaaviin arvoihin.

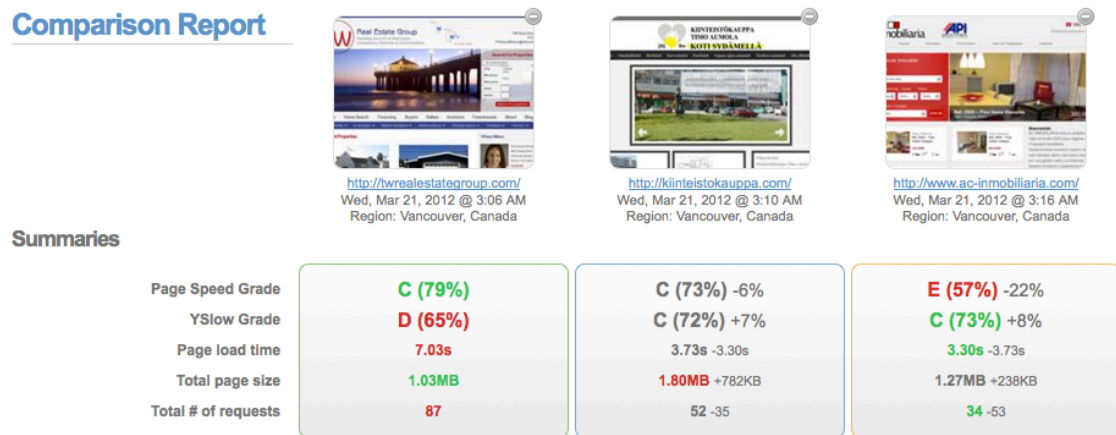
<b>Mittauskohde</b>	<b>Tutkittujen sivujen ka</b>	<b>www.suvilla.com</b>
Sivun koko	320 kilotavua	1,17 megatavua
HTTP-pyynnöt	43,91 kpl	47 kpl
Kuvien koko sivulla	205,99 kilotavua	983 kilotavua

Vertailuarvoja haluttiin myös yrityksen oman alan kilpailijoilta. SuVillan kilpakumppaneita ovat luonnollisesti muut Espanjassa toimivat suomalaiset kiinteistövälitysyrietykset, mutta yksikään niistä ei käyttänyt WordPressiä tai muutenkaan sisällönhallintajärjestelmää verkkosivuillaan, ja siksi suorituskyvyn vertailun tulokset olisivat olleet ristiriitaisia. Näin ollen katsottiin parhaaksi etsiä maailmanlaajuisesti kiinteistövälitysyrietyksiä, joiden verkkosivut oli rakennettu WordPress-alustalle.

Löydetyistä vaihtoehdoista poimittiin kolme vertailukelpoista verkkosivua, joiden ominaisuudet vastasivat SuVillan sivuston ominaisuuksia. Vertailukohteiksi valittiin kolmen pienyrityksen, TW Real Estate Groupin, Kiinteistökauppa Timo Aumolan ja AC Inmobiliarian verkkosivut. Kuviossa 12 nähdään, miten GTmetrix on arvostellut vertailtavat sivustot. Vertailusta ilmenee, ettei pienyritysten verkkosivujen optimointiin läheskään aina panosteta. Vertailukohteiden saamat kouluarvosanat sivustojen suorituskyvystä



ovat C:stä E:hen, eivätkä sivujen koot ole lähelläkään taulukossa 3 esitettyä keskiarvoa. Kiinteistökaupan ja AC Inmobiliarian sivuilla HTTP-pyyntöjen määrä on kuitenkin pidetty kurissa. Jälkimmäinen on pärjännyt jopa vähemmillä pyynnöillä kuin SuVilla. Kaiken kaikkiaan SuVillan etusivu oli melko lailla samalla tasolla vertailukohteiden kanssa.



Kuvio 12. Samankaltaisten sivustojen suorituskyvyn vertailu SuVillan arvoihin.

Lähtötilanteen analysoinnin jälkeen verkkosivun optimoinnille asetettiin tiukat, mutta realistiset tavoitteet. Tavoitteet olivat seuraavat:

1. sivujen latautuminen alle 1,5 sekunnissa
2. sivujen koon puolittaminen
3. HTTP-pyyntöjen vähentäminen kolmanneksella
4. kuvien pakkaaminen keskimäärin 40 prosenttia pienemmiksi
5. palvelimen kestävyyden takaaminen 100 yhtäaikaiselle käyttäjälle
6. YSlow ja Page Speed pistemäärä vähintään 90/100.

#### 4.4 Optimointitoimenpiteet lisäosien avulla

SuVillan WordPress-sivuston optimointi vaati keskittymistä erityisesti kolmeen laajaan kokonaisuuteen: välimuistin aktivoimiseen, tiedostojen pakkaamiseen ja yhdistämiseen, ja kuvien optimointiin. Lisäksi käytössä olevan qTranslate-lisäosan takia oli tärkeää löytää keino, jolla varmistettaisiin sivuston tietokannan ja sille lähetettävien kyselyiden optimaalinen toiminta. Koska sivusto toimii sisällönhallintajärjestelmän avulla, oli itsestään selvää, että optimointikin tehtäisiin WordPressin ehdoilla ja sen ominaisuuksia hyödyntäen.

WordPressin palvelimille ilmestyy päivittäin uusia lisäosia, joista toiset tuovat verkkosivuille pieniä, toiset suurempia lisätoiminnallisuuksia [26]. Lisäosien joukossa on runsaasti hyödyllisiä vaihtoehtoja, jotka parantavat sisällönhallintajärjestelmän ja verkkosivun toimivuutta. SuVillan sivuston optimointi toteutettaisiin siis pitkälti WordPressin lisäosien avulla, sillä siihen käyttötarkoitukseen oli tarjolla lukuisia vaihtoehtoja.

Ennen lisäosien valintaa tutkittiin eri vaihtoehtoja ja niiden saamia arvosteluja WordPressin lisäosasivustolla. WordPressin omalta tukisivustolta löytyi myös hyödyllistä tietoa siitä, miten verkkosivun suorituskykyä parannetaan lisäosien avulla. [24.] Käyttökelpoisten lisäosien suosituksia etsittiin luonnollisesti myös hakukoneiden avulla, ja kolmen lisäosan nimet tuntuivat toistuvan monissa kirjoituksissa: W3 Total Cache (W3TC), WP Smush.it ja WP-DBManager.

Nämä kolme lisäosaa todettiin sopiviksi SuVillan verkkosivun optimointiprosessiin. Yhtenä perusteluna valinnoille oli, että niiden vaikutukset eivät kumoa toisiaan millään tavalla. W3TC huolehtii välimuistien toiminnasta, JavaScript- ja CSS-tiedostojen tiivistämisestä ja tiedostojen pakkaamisesta. Smush.it taas optimoi kuvat, ja DBManager huoltaa ja optimoi tietokantaa. Tärkeää on myös huomioida, että näiden lisäosien kehitystyö on ollut aktiivista ja jatkuvaa jo useamman vuoden ajan [36; 37; 38]. Lisäksi W3TC:n ja Smush.it:n taustateknologiat ovat luotettavia, sillä ne ovat suurten yritysten luomia [19; 21].

## W3 Total Cache -lisäosa

W3 Total Cachen nimessä sana "total" on oikeutettu, sillä se on lisäosa, joka hoitaa kaikki verkkosivun tärkeimmät optimointitoimenpiteet muutamalla napin painalluksella. W3TC:n asennuksen jälkeen olemassa olevaa parempi lisäosan dokumentointi olisi ollut tarpeen, sillä sen käyttöönotto oli jokseenkin haastavaa. Mitään ei toki tarvinnut käsin kirjoittaa, mutta oikeiden valintojen tekeminen välimuistien ja tiedostojen pakkaamisen asetuksissa vaati perusteellista perehtymistä. Lisäosan todellinen loistokkuus tuli esiin sitten, kun oma tietotaito sen käytöstä oli kunnossa.

Kuten muutkin WordPressin lisäosat, myös W3TC tekee ylläpitäjän puolesta vaikeahkoja ja ylläpitotoimenpiteitä automaattisesti, ilman että tarvitsee koskea lähdekoodiin. W3TC:n avulla välimuistin aktivoiminen oli huomattavasti helpompaa kuin asianmukaisten käskyjen kirjoittaminen palvelimen juurikansiossa sijaitsevaan .htaccess-tiedostoon, jonka kautta tehtävän voisi manuaalisesti toteuttaa. W3TC:n asetukset on jaettu välilehtiin optimoitavissa olevien osa-alueiden mukaan: sivun välimuisti, tietokannan välimuisti, objektien välimuisti, selaimen välimuisti, Minify, CDN ja käyttäjäryhmät. Kaikkien välimuistien asetusten pääasia oli niiden päälle kytkeminen. Niiden asetuksista sai myös lisättyä dokumentteihin ja ulkoisiin tiedostoihin voimassaoloajat, jotka pystyi määrittämään itse, mikäli oletusarvot eivät kelvanneet. Minify-välilehdessä päästiin asettamaan JavaScript- ja CSS-tiedostojen kompressointi päälle, ja lisäksi käskettiin W3TC:n poistaa näiden tiedostojen sisäiset kommentit. CDN:n eli sisällönjakeluverkon asetukset jätettiin rauhaan, sillä tekniikkaa ei käytetä SuVillan sivuilla. Käyttäjäryhmien asetuksiin ei myöskään vielä koskettu, mutta niihin toimintoihin perehdytään todennäköisesti lähitulevaisuudessa, sillä sitä kautta on mahdollista ohjata esimerkiksi mobiilikäyttäjät eri sivulle kuin tavallisen tietokoneen käyttäjät.

Pelkän W3TC:n aktivoiminen SuVillan sivustolle sai GTmetrixin testissä arvosanan nousemaan huimasti. Tässä välitestissä testattiin vain etusivua, jonka Page Speed -arvosana nousi ilman lisäosan asetusten hienosäätöä suoraan A:han ja Yslow puolestaan antoi arvosanan B. Tällainen parannus sivuston optimoinnissa on jo suuri saavutus, mutta tiedossa oli, että todelliset pullonkaulat sijaitsivat asuntosivuilla niiden sisältämien kuvien takia.

### WP Smush.it -lisäosa

Luvussa 3.2 on kerrottu Yagoon Smush.it-palvelusta, joka poistaa turhaa dataa kuvista ja näin ollen pienentää niiden kokoa. Kävi ilmi, että palvelusta on tehty WordPressiin myös lisäosa, joka otettiin saman tien käyttöön SuVillan sivuilla. Asennuksen jälkeen Smush.it-lisäosan valikosta näki, kuinka monta kuvaa palvelimella on, ja aika-arvion siitä, kuinka pitkään kuvien tiedostokoon pienentäminen kestää. Lisäosan avulla kaikki SuVillan sivuston sadat asuntokuvat saatiin optimoitua yhdellä kertaa massatoimintona ilman virheilmoituksia. Lisäksi lisäosa optimoi tulevaisuudessa jokaisen sivustolle lisätävän kuvan automaattisesti ilman erillistä käskyä sen latautuessa palvelimelle.

Taulukko liitteessä 2 osoittaa, kuinka paljon tilaa SuVillan palvelimella säästettiin poistamalla kuvien turha data. Liitteen taulukossa on esitetty vain sivuston ulkoasuun käytettävien kuvien Shmush.it-pakkauksen tulos, ja ne ovat jo luonnostaan kooltaan melko pieniä eivätkä sisällä runsain määrin värejä. Taulukkoa tarkastelemalla voidaan todeta, että kuvien koko pienenee keskimäärin 30–40 prosenttia, mikä on huomattava palvelintilan säästö verkkosivulla, joka sisältää runsaasti kuvia. Lopputuloksena SuVillan sivuston kuvien optimoinnin arvostelu GTmetrixin Page Speed -testissä parani 32:sta täyteen 100 pisteeseen.

### WP-DBManager -lisäosa

Koko WordPress-järjestelmän toiminta perustuu dynaamisuuteen, eli sisällön noutamiseen tietokannasta, mikä luonnollisesti kuormittaa sekä tietokantaa että palvelinta, joka generoi sivustokokonaisuuden selaimessa esitettävään kuntoon. SuVillan tapauksessa lisärasitetta tietokannalle aiheuttaa qTranslate-lisäosa kielikyselyineen. Monikielisen sivuston voi toteuttaa monella tapaa, esimerkiksi .mo- (Machine Object) ja .po-tiedostoilla (Portable Object), joihin käännetty sisältö lisätään ja joista käännökset haetaan tarvittaessa [39]. SuVillan sivuilla dynaamisen sisällön käännöksistä huolehtii qTranslate ilman erillistä ohjelmointia, mutta staattisten objektien, kuten otsikoiden, käännökset on tehty suoraan lähdekoodiin qTranslaten qtransGetLanguage()-funktion avulla. Funktion tarkoitus on tarkistaa, millä kielellä sivustoa sillä hetkellä selataan, ja näyttää sisältöä sen mukaan. Kyseessä ei ole yhtä tehokas tapa kuin edellä mainitut erilliset käännöstiedostot, mutta sivuja tehtäessä tapa oli kaikista nopein toteuttaa.

WP-DBManager asennettiin SuVillan sivustolle, jotta tietokannan toiminta ja rakenne pysyisivät optimaalisina. Sen tehtävä on pitää tietokanta puhtaana turhalta väliaikaisisällöltä, tehdä varmuuskopioita ja huolehtia tietokannan optimoidusta rakenteesta. Kaikki edellä mainitut tehtävät on lisäosan ansiosta mahdollista ajastaa, eikä niistä sen jälkeen tarvitse huolehtia itse. Suurin etu lisäosassa on se, etteivät julkaistavien sivujen väliaikaiset luonnokset jää pitkiksi ajoiksi tietokantaan viemään tilaa.

DBManagerin asentamisesta ei ollut optimoinnille suoraa hyötyä, jos seurataan suorituskyvyn mittareita testeissä. DBManager on enemmänkin palvelinpuolen optimointityökalu, joka varmistelee sisällönhallintajärjestelmän toimivuuden jatkumisen ja pitää sivuston datasta huolta. Täten palvelintilassa lojuu vähemmän roskaa ja sinne pystytään vastaanottamaan enemmän yhtäaikaista kävijöitä pienemmällä sivuston kaatumisen riskillä.

#### 4.5 Muut optimointitoimenpiteet

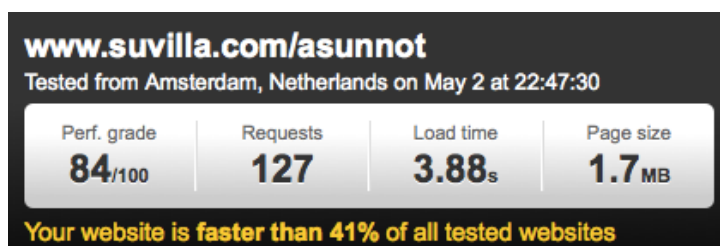
Lisäosilla toteutettujen optimointitoimenpiteiden lisäksi SuVillan verkkosivuja paranneltiin jonkin verran myös käsin. Kuten raportin alussa todettiin, teema, joka määrittelee sivujen ulkoasun, oli valmis paketti, joka ostettiin sivuston käyttöön. Teema oli onneksi hyvin rakennettu, ja vaikutti siltä, että sitä tehtäessä oli ajateltu myös teeman tulevien ostajien verkkosivujen suorituskykyä. Teemassa oli kuitenkin muutamia ominaisuuksia, joita SuVillan verkkosivu ei tarvinnut eikä tule koskaan tarvitsemaan, kuten animoidut kuvaesitykset sivujen sivupalkkiin. Teeman ylimääräiset ominaisuudet sisälsivät monessa tapauksessa elementtejä, jotka tarvitsivat kymmeniä rivejä tyylimääräittelyjä CSS-tiedostossa tai vaativat JavaScript-tiedostojen suorittamista. Näin ollen ylimääräiset ominaisuudet poistettiin teemasta kokonaan ja saatiin sivuston koko vielä hiukan pienemmäksi.

Teemassa tehtyjen muutosten jälkeenkin sivuston tyylitiedostoon jäi epäkohtia ja jonkin verran turhaa sisältöä, mutta silmämääräisesti analysoimalla koko CSS-syntaksin puhdistaminen olisi vienyt kymmeniä työtunteja. Tästä syystä päätettiin, ettei tyylitiedostoa aleta käydä läpi rivi riviltä, vaan siirryttäisiin sen sijaan käyttämään itse tehtyä WordPress-teemaa tulevaisuudessa, kun hetki olisi yritykselle sopiva.

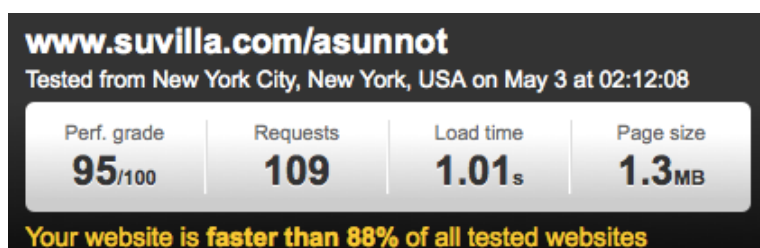
#### 4.6 Verkkosivun analysointi optimoinnin jälkeen

Optimointitoimenpiteiden jälkeen verkkosivu oli valmis uudelle testauskierrokselle. SuVillan sivustolla oli nyt otettu käyttöön eri tasojen välimuistit, ja skriptit, tyylitiedostot ja kuvat oli pakattu huomattavasti pienempään tilaan, tietokanta oli optimoitu, CSS-koodia muokattu ja HTTP-pyyntöjä vähennetty. Sivuston optimoidun version suorituskyky testattiin jälleen GTmetrix-, Pingdom-, Webpagetest.org- ja Siege-testaustyökaluilla.

Verkkosivun suorituskyvyn suhteen toiveet olivat korkealla, ja kuten kuvioista 13 ja 14 näkee, latausajoissa on päästy paljon alaspäin. Kuvien Pingdom-työkalun mittaustulokset ovat asuntojen listaussivuilta, missä optimointi on onnistunut ilmeisen hyvin siitäkin huolimatta, että sivun HTTP-pyyntöjä ei voitu karsia toivotuissa määrin pakollisten kuvien noutamisen takia. Kuvien koot saatiin kuitenkin minimiin, minkä ansiosta tavoitteet koko sivun koon pienenemisestä toteutuivat, ja se lyhentää latausaikoja jo paljon.



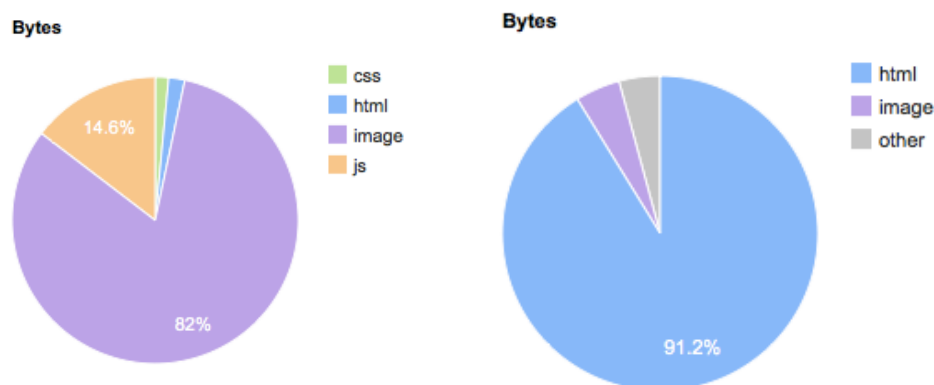
Kuvio 13. SuVillan asuntojen listaussivun suorituskyvyn mittaustulokset ennen optimointia.



Kuvio 14. SuVillan asuntojen listaussivun mittaustulokset optimoinnin jälkeen.

Samalle asuntojen listaussivulle tehty testi Webpagetest.org:n työkalulla osoittaa, että käyttäjän ladatessa sivua toista kertaa HTTP-pyyntöjä lähetetään ainoastaan kolme, sillä suurin osa sisällöstä pystytään tarjoamaan suoraan välimuistista heti näkyville. Webpagetest.org:n mittaustuloksista kuviossa 15 voi myös huomata, miten sivun kuorma jakaantuu ensimmäisellä ja toisella latauksella. Ensimmäisellä kerralla HTML:n

osuus on alle 2 prosenttia koko sivun koosta ja kuviin panostetaan 82 prosenttia resursseista, kun taas toisella kerralla tilanne kääntyy pääläelleen ja kuvien osuus on enää alle 5 prosenttia sivun koosta.



Kuvio 15. Kohdelistaussivun resurssien jakautuminen ensimmäisellä ja toisella latauskerralla.

Etusivun ja yksittäisen myyntikohteen sivun mittaustulokset olivat myös kiitettäviä optimoinnin jälkeen. Etusivulla HTTP-pyyntöjen määrä oli onnistuttu vähentämään puoleen, ja sielläkin latausnopeus pysytteli alle 1,5 sekunnissa Pingdomin mukaan. Myös yksittäisen asunnon sivu kärsi liiallisista HTTP-pyynnöistä kuvien takia, mutta kuten listaussivullakin, sivua oli onnistuttu optimoimaan muuten niin kattavasti, että tulokset näyttivät lopuksi hyvältä. Kohdelistaussivu ja yksittäisen myyntikohteen sivu jäivät Yslow, ja Page Speed -pisteissä hieman alle 90:n, eikä HTTP-pyyntöjä saatu asetettujen tavoitteiden tasolle, mutta muuten tavoitteet toteutuivat kaikin puolin. Liitteissä 4, 5 ja 6 on esitetty kaikkien sivujen mittaustulokset ja pisteitykset optimoinnin jälkeen sekä vertailuarvot lähtötilanteesta.

Seuraavaksi oli vuorossa palvelimen kuormansietokyvyn testaus, joka suoritettaisiin jälleen Siege-työkalulla Load Impactin yhä jatkuvien ongelmien vuoksi. Ennen optimointia ensimmäiset rasiustestaukset päättyivät surkeasti melkein palvelimen kaatumiseen, mutta sivuston optimointi oli onneksi saanut aikaan muutoksen tähänkin asiaan. Rasiustestaus aloitettiin tälläkin kertaa 10 virtuaalikäyttäjällä, mikä ei aiheuttanut palvelimelle mitään rasituksen merkkejä. Sitten kuormitusta kokeiltiin 30 käyttäjällä, eikä verkkosivun toiminta tuntunut muuttuvan juuri lainkaan. Lopuksi sivustoa testattiin 100 samanaikaisella käyttäjällä, jolloin joidenkin pyyntöjen käsittely kesti aavistuksen pidempään, mutta vasteajan keskiarvo pysyi silti puolen sekunnin tienoilla, mikä on

erinomainen tulos optimoimattomaan sivustoon verrattuna. Sivusto olisi varmasti kestänyt suuremman ruuhkahuipun, mutta testausta ei voitu jatkaa pidempään, jottei normaali liikenne sivuilla häiriintyisi. Liitteessä 7 nähdään Siegen raportit mainituista testeistä ja vertailuarvoina tulokset optimoimattoman sivun testeistä.

#### 4.7 Jatkotoimenpiteet

SuVillan sivuston tila on optimoinnin jälkeen erittäin hyvä, mutta aina löytyy parannettavaa. Tehdyssä optimoinnissa keskityttiin enimmäkseen lisäosien avulla toteutettaviin parannuksiin ja kevennettiin lähdekoodia niiden avulla. Koodin rakenteeseen ei niinkään puututtu sen tähden, että se on äärimmäisen aikaa vievää, kun käytössä on toisen henkilön kirjoittama koodi. Jos kuitenkin haluttaisiin tehdä viimeistelevää hienosäätöä, lähdekoodista olisi hyvä poistaa turhat dynaamiset osoitteet, joita WordPress-sivut ovat poikkeuksetta täynnä. Poistamalla nämä tietokantaa kuormittavat GET-käskyt ja korvaamalla ne staattisilla osoitteilla kuten esimerkkikoodissa kuviossa 16 on näytetty, säästetään jonkin verran palvelimen resursseja. [40.]

```

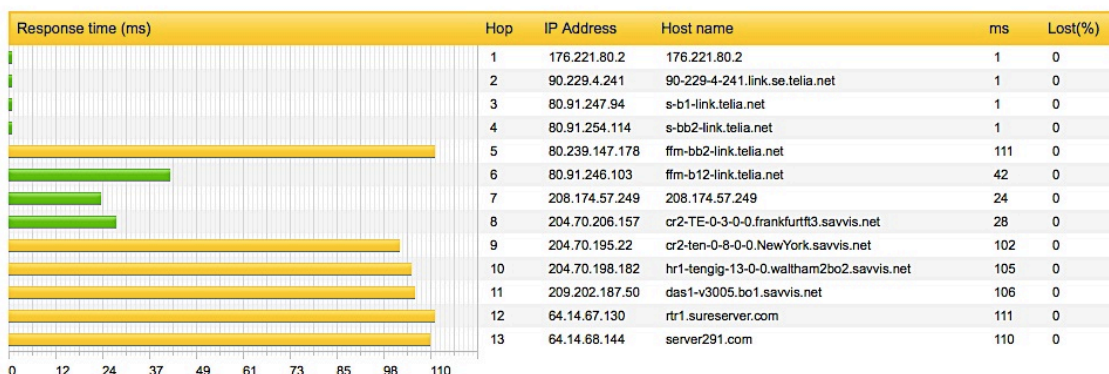
1 echo '';
2
3 echo '';

```

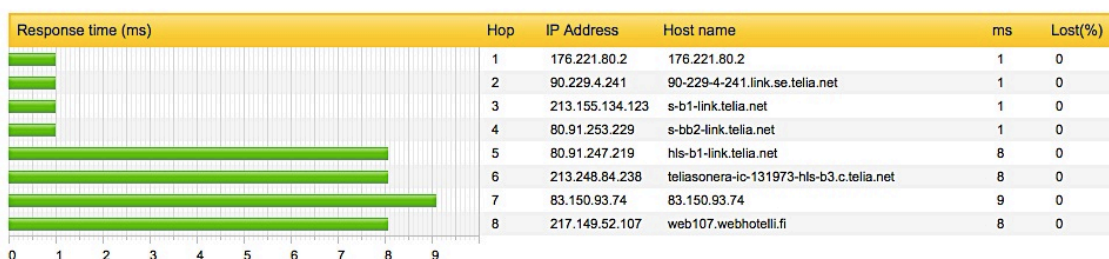
Kuvio 16. Dynaamisen osoitteen korvaaminen staattisella osoitteella.

Toinen optimointitoimenpide, joka SuVillan sivuille voisi tehdä, olisi siirtää sivut nykyiseltä suosittuun amerikkalaiseen palveluntarjoajan palvelimelta Euroopassa sijaitsevalle palvelimelle. Suurin osa sivuilla kävijöistä sijaitsee nimittäin fyysisesti joko Suomessa tai Espanjassa, joten siirtämällä sivusto eurooppalaiselle palvelimelle se saataisiin lähemmäksi käyttäjiä. Fyysisesti lyhyempi reitti sivuille pienentää latausaikoja usein merkittävästi varsinkin käytettäessä hitaampia yhteyksiä. Kuvioissa 17 ja 18 nähdään, miten pelkkään palvelimen löytämiseen käytettävä aika on reilusti pidempi, kun matka on maantieteellisesti pidempi. Kuvissa esitetään tietoverkon läpi kuljettu reitti askel askeleelta SuVillan palvelimelle Yhdysvaltoihin ja erään suomalaisen yrityksen palvelimelle Helsinkiin samasta lähtöpisteestä. Yhdysvaltoihin tehtävä matka kestää jopa yli kymmenen kertaa pidempään kuin vertailureitti Helsinkiin.



Traceroute: [suvilla.com](http://suvilla.com)

Kuvio 17. Reitti Vantaalta SuVillan palvelimelle Yhdysvaltoihin.

Traceroute: [encodedwebdesign.net](http://encodedwebdesign.net)

Kuvio 18. Reitti Vantaalta Encodedin palvelimelle Helsinkiin.

Verkkosivun suorituskykyyn voi vaikuttaa myös ottamalla käyttöön sisällönjakeluverkon. Sisällönjakeluverkon tai CDN:n (Content Delivery Network) käyttöönotto on optimointitoimenpiteistä arvostetuin ja kallein. Luvussa 3.1 todettiin, ettei käyttäjän etäisyyteen www-palvelimelta voi vaikuttaa, mikä ei täysin pidä paikkaansa, mikäli asiaan ollaan valmiita käyttämään suuri määrä rahaa. CDN:n avulla verkkosivun staattinen sisältö jaetaan monelle palvelimelle ympäri maailman ja sivuston sisältö tarjotaan käyttäjälle aina häntä lähimpänä olevalta palvelimelta. CDN:n käyttö tehostaa sisällön jakelua ja näin ollen verkkosivun latautumisenopeutta huomattavasti, mutta käytännössä sen käyttö kannattaa vain suuryrityksille, joiden sivuilla vierailaan tuhansia kertoja päivässä eri puolilta maailmaa. [29.]

Viimeinen jatkotoimenpide, joka parantaisi verkkosivun suorituskykyä vielä aavistuksen, olisi kuvien ja muiden staattisten objektien tarjoilu osoitteesta, joka ei luo evästeitä. Evästeiden huono puoli on niiden turha esiintyminen HTTP-pyyntöissä, jotka lähetetään evästeen luovan HTTP-pyyntön, kuten kuvapyynnön, jälkeen. Erillinen palvelintila

staattiselle sisällölle on kuitenkin tarpeeton, mikäli sivuston HTTP-pyyntö pysyvät vähäisinä. [41.]

Viimeisimpinä mainitut optimointikeinot vaativat rahallista panostusta ja suuria muutoksia verkkosivun jakeluun, eikä pienyrityksen sivuston optimointiin ole järkevää panostaa näin paljon. Tärkeintä on pitää verkkosivut ajan tasalla ja huolehtia siitä, että optimoinnin perusasiat ovat kunnossa.

## 5 Yhteenveto

Suorituskykyisen verkkosivun aikaansaaminen sisällönhallintajärjestelmän, kuten WordPressin, avulla ei ole itsestäänselvyys, vaan vaatii tarkkaa työtä. Parhaan tuloksen saa rakentamalla oman teeman kaiken pohjaksi, jolloin lähdekoodiin ei pääse syntymään ylimääräisiä koodirivejä. Samalla oppii tuntemaan järjestelmän läpikotaisin. Mikäli oman teeman rakentaminen tuntuu vaikealta, kunnollista valmista teemaa ja oikeita lisäosia käyttämällä voi päästä melko helpolla hyviin tuloksiin – ainakin sen jälkeen, kun lisäosat ja teema on saatu toimimaan tehokkaimmalla mahdollisella tavalla. Suorituskyvyltään huipputasoinen verkkosivu saadaan kuitenkin vain rakentamalla se alusta asti itse ja perehtymällä optimointitoimenpiteisiin kunnolla.

Sisällönhallintajärjestelmällä toimivan verkkosivun optimointi eroaa oleellisesti tavallisen verkkosivun optimoinnista, mutta se ei välttämättä ole sitä haasteellisempaa, jos tuntee työskentely-ympäristönsä. Olen itse työskennellyt WordPress-sivustojen parissa pari vuotta, ja järjestelmän tunteminen oli suuri apu tätä työtä tehdessä. Tietokannan ja tiedostojärjestelmän rakenteen tunteminen auttoi monessa kohtaa etsimään virheitä oikeista paikoista.

WordPress-sivuston optimoinnissa tulee muistaa ne perussäännöt, jotka ovat voimassa mitä tahansa sivua optimoitaessa: välimuistien käyttäminen, kuvien ja muiden ulkoisten tiedostojen pakkaaminen sekä koodin oikean rakenteen ylläpitäminen. Mikäli optimointia tehdään isommalla budjetilla, sisällönjakeluverkko ja oma palvelin ovat lisäksi ehdottoman hyviä investointeja suorituskyvyn tehostamiseksi.

Verkkosivun optimointi on mielenkiintoinen aihealue, joka kehittyy päivä päivältä, ja optimoinnin aikana tehtävät toimenpiteet muuttuvat. Vaikeinta optimoinnissa on päättää, mihin vetää rajan, jonka jälkeen ei enää yritä tehdä muutoksia sivuille. Optimointiprosessin pääasia on kuitenkin saada aikaan paineenkin alla toimiva sivusto, joka on mahdollisimman vähän riippuvainen käyttöympäristöstään.

Tämän optimointityön aikana SuVillan sivuista tuli noin 50 prosenttia kevyemmät ja niiden latautumisenopeus parani huomattavasti. Alun perin muutamasta kymmenestä samanaikaisesta käyttäjästä lähes kaatunut palvelin on nyt vakaa alusta, joka kestää

hyvin 100 samanaikaista vierailijaa. Insinöörityö ja siihen liittyvät tutkimukset opettivat minulle paljon uutta WordPressistä ja verkkosivujen toiminnasta, ja pidän työn lopputulosta onnistuneena. Verkkosivua ylläpitävä kiinteistönvälitysyritys SuVilla on ilmaissut tyytyväisyytensä työn tuloksiin, ja myös yrityksen asiakkaat ovat olleet mielissään sivuston sujuvasta toiminnasta.

Tähän loppuun todettakoon, että tällaisenkin työn toteuttamiselle voi löytää ekologiset perustelut. Tosiasia on, että jo pelkkä verkon selailu kuormittaa luontoa. Sekä tietokoneet että palvelimet kuluttavat sähköä ja vaativat säilytystiloja ja viilennystä. Optimoimaton verkkosivu käyttää turhan paljon näitä resursseja, jolloin monet samalla palvelimelle sijaitsevat optimoimattomat sivustot kuluttavat niitä jo suuremman määrän, ja sitä kautta energiaa menee hukkaan. Niinkin pieni asia kuin verkkosivun optimointi voi vähentää luonnon kuormittumista, ja hyvin optimoiduille verkkosivuille voisikin antaa ympäristömerkin.

## Lähteet

- 1 IEEE Standard 802.3ba-2010. 2011. Verkkodokumentti. IEEE Standards Association. <<http://standards.ieee.org/findstds/standard/802.3ba-2010.html>>. Luettu 22.3.2012.
- 2 Stern, Hall, Damstra, David & Williams, Brad. 2010. Professional Wordpress, Design and Development. United States of America: Wiley Publishing.
- 3 Open Standards Requirement for Software. Verkkodokumentti. Open Source Initiative. <<http://www.opensource.org/osr>>. Luettu 23.3.2012.
- 4 About Wordpress. Verkkodokumentti. Wordpress Community. <<http://wordpress.org/about/>>. Luettu 23.3.2012.
- 5 Wordpress Features. Verkkodokumentti. Wordpress Community. <[http://codex.wordpress.org/WordPress\\_Features](http://codex.wordpress.org/WordPress_Features)>. Luettu 23.3.2012.
- 6 What is Dynamic Websites. Verkkodokumentti. Rose India Technologies. <<http://www.roseindia.net/services/webdesigning/corporatewebsitedesign/What-is-dynamic-websites.shtml>>. Luettu 23.3.2012.
- 7 Introduction to Database Indexes. 2006. Verkkodokumentti. Interspire. <<http://www.interspire.com/content/2006/02/15/introduction-to-database-indexes>>. 15.2.2006. Luettu 28.3.2012.
- 8 Notable Users. Verkkodokumentti. Wordpress Community. <<http://en.wordpress.com/notable-users/>>. Luettu 7.4.2012.
- 9 Larson, Richard. Why Use Premium WordPress Themes (And Why Not). Verkkodokumentti. Slodive. <<http://slodive.com/web-development/premium-wordpress-themes-and-not/>>. Luettu 22.3.2012.
- 10 Best Practices for Speeding Up Your Web Site. Verkkodokumentti. Yahoo! Developer Network. <<http://developer.yahoo.com/performance/rules.html>>. Luettu 28.3.2012.
- 11 Web Performance Best Practices. Verkkodokumentti. Google Developers. <[http://code.google.com/speed/page-speed/docs/rules\\_intro.html](http://code.google.com/speed/page-speed/docs/rules_intro.html)>. Luettu 28.3.2012.
- 12 King, Andrew B. 2008. Website Optimization. United States of America: O'Reilly Media.
- 13 Nielsen, Jacob. 2010. Website Response Times. Verkkodokumentti. <<http://www.useit.com/alertbox/response-times.html>>. 21.6.2010. Luettu 25.3.2012.
- 14 Another Step to Reward High-Quality Sites. 2012. Verkkodokumentti. Google. <<http://googlewebmastercentral.blogspot.com/2012/04/another-step-to-reward-high-quality.html>>. 24.4.2012. Luettu 28.4.2012.

- 15 Load Impact Features. Verkkodokumentti. Load Impact. <<http://loadimpact.com/features>> Luettu 28.3.2012.
- 16 Cache. 2010. Verkkodokumentti. Muhammad Abbie Hamzah. <<http://muhammadabbie.students-blog.undip.ac.id/>>. 20.9.2010. Luettu 7.4.2012.
- 17 Optimize Caching. Verkkodokumentti. Google Developers. <[http://code.google.com/speed/page-speed/docs/rules\\_intro.html](http://code.google.com/speed/page-speed/docs/rules_intro.html)>. Luettu 28.3.2012.
- 18 Writing Efficient CSS. Verkkodokumentti. Mozilla Developer Network. <[https://developer.mozilla.org/en/Writing\\_Efficient\\_CSS](https://developer.mozilla.org/en/Writing_Efficient_CSS)>. Luettu 7.4.2012.
- 19 Minify. Verkkodokumentti. Google. <<http://code.google.com/p/minify/>>. Luettu 7.4.2012.
- 20 What is the JPEG Graphics File Format? 2011. Verkkodokumentti. Indiana University. <<http://kb.iu.edu/data/adqe.html>>. 6.8.2011. Luettu 20.4.2012.
- 21 Smush.it – Frequently Asked Questions. Verkkodokumentti. Yahoo! Developer Network. <<http://developer.yahoo.com/yslow/smushit/faq.html>>. Luettu 20.4.2012.
- 22 Lupetti, Antonio. 2008. 10 Beautiful Icons Set for Web Developers and Designers. Verkkodokumentti. Woorkup.com. <[http://woork.blogspot.com/2008/09/10-beautiful-icons-set-for-web\\_18.html](http://woork.blogspot.com/2008/09/10-beautiful-icons-set-for-web_18.html)>. 18.9.2008. Luettu 20.4.2012.
- 23 Hedengren, Thord Daniel. 2011. Smashing WordPress Themes. United Kingdom: Wiley Publishing.
- 24 Plugins and Hacks. Verkkodokumentti. Wordpress.org. <<http://wordpress.org/support/forum/plugins-and-hacks>>. Luettu 25.4.2012.
- 25 Colaiacovo, Kathy. 2010. What is a WordPress Plugin? Verkkodokumentti. My Social Media VA. <<http://www.mysocialmediava.com/wordpress-websites/wordpress-plugi>>. 26.3.2010. Luettu 25.4.2012.
- 26 Plugin Directory. Verkkodokumentti. Wordpress Community. <<http://wordpress.org/extend/plugins>>. Luettu 25.4.2012.
- 27 GTmetrix. Verkkodokumentti. Gossamer Threads. <<http://gtmetrix.com>>. Luettu 28.3.2012.
- 28 Test a Website's Performance. Verkkodokumentti. WebPagetest. <<http://www.webpagetest.org/>>. Luettu 28.3.2012.
- 29 Cloudflare CDN. Verkkodokumentti. Cloudflare. <<https://www.cloudflare.com/features-cdn>>. Luettu 25.4.2012.
- 30 Allen. 2007. Interview with Pingdom CEO Sami Nurmi. Verkkodokumentti. Center Networks. <<http://www.centernetworks.com/interview-with-pingdom-ceo-sami-nurmi>>. 7.11.2007. Luettu 25.4.2012.

- 31 Pingdom Tools. Verkkodokumentti. Pingdom. <<http://tools.pingdom.com/fpt/>>. Luettu 25.4.2012.
- 32 Load Testing Web Applications. Verkkodokumentti. Microsoft Corporation. <<http://msdn.microsoft.com/en-us/library/bb924372.aspx>>. Luettu 25.4.2012.
- 33 Fulmer, Jeff. 2012. Siege Manual. Verkkodokumentti. Joe Dog Software. <<http://www.joedog.org/siege-manual/>>. 30.1.2012. Luettu 7.4.2012.
- 34 What is Shared Hosting? Verkkodokumentti. BoogieJack.com. <<http://www.boogiejack.com/hosting/what-is-shared-hosting.html>>. Luettu 27.4.2012.
- 35 Ramachandran, Sreeram. 2010. Web Metrics: Size and Number of Resources. Verkkodokumentti. Google Developers. <<http://code.google.com/intl/fi/speed/articles/web-metrics.html>>. Luettu 28.4.2012.
- 36 WP Smush.it. Verkkodokumentti. Wordpress.org. <<http://wordpress.org/extend/plugins/wp-smushit/>>. Luettu 28.4.2012.
- 37 W3 Total Cache. Verkkodokumentti. Wordpress.org. <<http://wordpress.org/extend/plugins/w3-total-cache/>>. Luettu 28.4.2012.
- 38 WP-DBManager. Verkkodokumentti. Wordpress.org. <<http://wordpress.org/extend/plugins/wp-dbmanager/>>. Luettu 28.4.2012.
- 39 Translating Wordpress. Verkkodokumentti. Wordpress Codex. <[http://codex.wordpress.org/Translating\\_WordPress](http://codex.wordpress.org/Translating_WordPress)>. Luettu 27.4.2012.
- 40 Starr, Jeff. 2009. Optimize WordPress Performance with the Wp-config.php File. Verkkodokumentti. Digging into Wordpress. <<http://digwp.com/2009/07/optimize-wordpress-performance-with-the-wp-config-php-file/>> 8.7.2009. Luettu 2.5.2012.
- 41 Flaherty, Rob. 2010. Serving Static Content from a Cookieless Domain. Verkkodokumentti. <[www.ravelrumba.com/blog/static-cookieless-domain](http://www.ravelrumba.com/blog/static-cookieless-domain)>. 23.2.2010. Luettu 2.5.2012.

## Tutkimussivuston kävijämäärä maaliskuussa 2012

Google Analytics

<http://www.suvilla.com> - <http://www.suvilla.com> [OLETUS]

### Kävijöiden esittely

5.3.2012 - 31.3.2012

100,00 % kokonaismäärästä käynnit

Yleiskatsaus



1 411 henkilöä kävi tässä sivustossa

Käynnit: 1 935

Yksilöidyt kävijät: 1 411

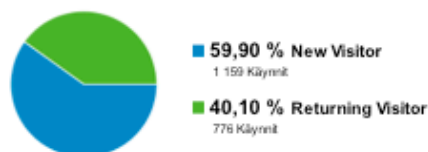
sivun katselua: 12 560

sivua/käynti: 6,49

Käynnin keskim. kesto: 00:05:16

poistui välittömästi: 29,20 %

uusia käyntejä: 59,90 %



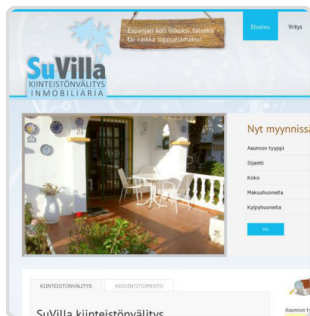


## Etusivun mittaustulokset GTmetrix-työkalulla ennen optimointia



The web should be fast.

### Executive Summary



### Performance Report for: <http://suvilla.com/>

Report generated: Monday, February 27, 2012, 3:56 AM -0800  
Test Server Region: Vancouver, Canada  
Using: Firefox 9.0.1, Page Speed 1.12.0.1, YSlow 3.0.4

Page Speed Grade:

(62%) ↓ Avg: 73%



YSlow Grade:

(66%) ↓ Avg: 75%



Page load time: 2.60s | Total page size: 1.17MB | Total number of requests: 48

#### Priority Issues (Top 5)

Leverage browser caching		F (11)	↓ Avg Score: 43%	Server	High
Defer parsing of JavaScript		F (28)	↓ Avg Score: 66%	JS	High
Combine images using CSS sprites		F (28)	↓ Avg Score: 56%	Images	Medium
Optimize images		F (32)	↓ Avg Score: 74%	Images	High
Enable gzip compression		F (36)	↓ Avg Score: 72%	Server	High

#### How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, **Google has announced that they are using page speed in their ranking algorithm.**

#### About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

#### About the Developer



GTmetrix is developed by the good folks at **Gossamer Threads**, a Vancouver-based company with over 16 years experience in web technology. [www.gossamer-threads.com](http://www.gossamer-threads.com)

#### What do these grades mean?

GTmetrix analyzes your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The **grades you see represent** how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

#### What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:

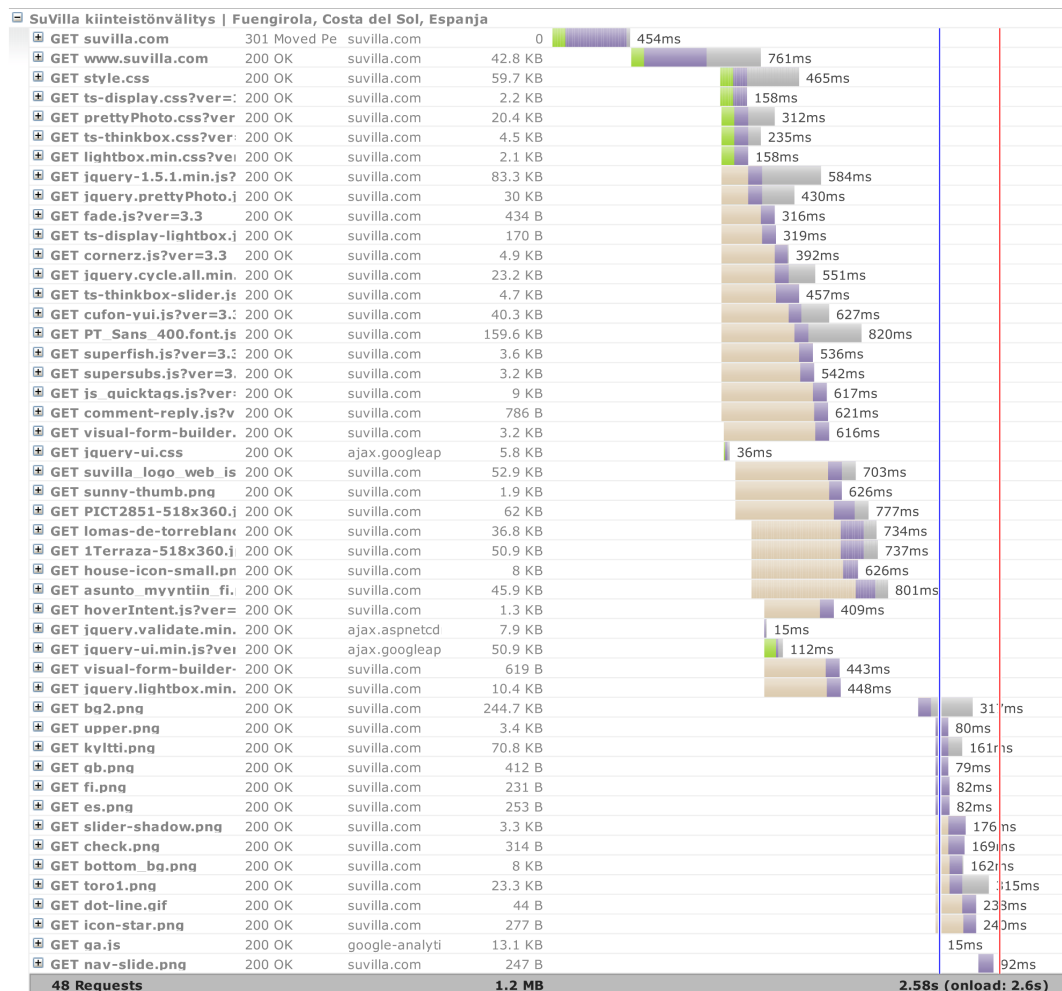
- ▶ **Executive:** Overall score information and Priority Issues
- ▶ **History:** Graphed history of past performance
- ▶ **Waterfall:** Graph of your site's loading timeline
- ▶ **Technical:** In-depth Page Speed & YSlow information

These will provide you with a snapshot of your performance.



## Waterfall

The waterfall graph displays the loading behaviour of your site in Firefox. It can be used to discover simple issues such as 404's or more complex issues such as external resources blocking page rendering.





## Page Speed Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Leverage browser caching	F (11)	↓ Avg Score: 43%	Server	High
Defer parsing of JavaScript	F (28)	↓ Avg Score: 66%	JS	High
Combine images using CSS sprites	F (28)	↓ Avg Score: 56%	Images	Medium
Optimize images	F (32)	↓ Avg Score: 74%	Images	High
Enable gzip compression	F (36)	↓ Avg Score: 72%	Server	High
Minify CSS	F (49)	↓ Avg Score: 76%	CSS	High
Minify HTML	D (63)	↓ Avg Score: 92%	Content	High
Specify image dimensions	C (70)	↑ Avg Score: 51%	Images	High
Use efficient CSS selectors	F (0)	↓ Avg Score: 26%	CSS	Low
Minify JavaScript	B (82)	↓ Avg Score: 87%	JS	High
Optimize the order of styles and scripts	B (85)	↓ Avg Score: 94%	CSS/JS	High
Remove unused CSS	C (74.3)	↑ Avg Score: 65%	CSS	Low
Remove query strings from static resources	A (92)	↓ Avg Score: 93%	Content	High
Specify a Vary: Accept-Encoding header	A (92)	↑ Avg Score: 86%	Server	High
Serve scaled images	A (99)	↑ Avg Score: 91%	Images	High
Avoid bad requests	A (100)	↓ Avg Score: 96%	Content	High
Enable Keep-Alive	A (100)	↑ Avg Score: 94%	Server	High
Inline small CSS	A (100)	↑ Avg Score: 93%	CSS	High
Inline small JavaScript	A (100)	↓ Avg Score: 97%	JS	High
Make landing page redirects cacheable	A (100)	↓ Avg Score: 99%	Server	High
Minimize redirects	A (100)	↑ Avg Score: 93%	Content	High
Minimize request size	A (100)	↓ Avg Score: 100%	Content	High
Put CSS in the document head	A (100)	↓ Avg Score: 100%	CSS	High
Serve resources from a consistent URL	A (100)	↓ Avg Score: 96%	Content	High
Specify a cache validator	A (100)	↓ Avg Score: 96%	Server	High
Specify a character set early	A (100)	↓ Avg Score: 97%	Content	High
Avoid CSS @import	A (100)	↓ Avg Score: 98%	CSS	Medium
Prefer asynchronous resources	A (100)	↓ Avg Score: 96%	JS	Medium



## YSlow Recommendations

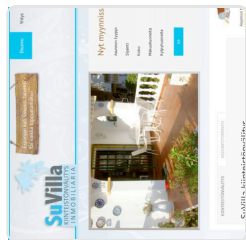
RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Add Expires headers	F (0)	↓ Avg Score: 17%	Server	High
Make fewer HTTP requests	F (0)	↓ Avg Score: 44%	Content	High
Compress components with gzip	F (0)	↓ Avg Score: 57%	Server	High
Use a Content Delivery Network (CDN)	F (0)	⬆ Avg Score: 5%	Server	Medium
Use cookie-free domains	F (0)	↓ Avg Score: 48%	Cookie	Low
Put JavaScript at bottom	F (30)	↓ Avg Score: 77%	JS	Medium
Configure entity tags (ETags)	F (0)	↓ Avg Score: 40%	Server	Low
Minify JavaScript and CSS	A (90)	↑ Avg Score: 82%	CSS/JS	Medium
Do not scale images in HTML	A (95)	↑ Avg Score: 86%	Images	Low
Make favicon small and cacheable	A (95)	⬆ Avg Score: 94%	Images	Low
Avoid empty src or href	A (100)	⬆ Avg Score: 98%	Content	High
Avoid URL redirects	A (100)	↑ Avg Score: 92%	Content	Medium
Make AJAX cacheable	A (100)	⬆ Avg Score: 99%	JS	Medium
Put CSS at the top	A (100)	⬆ Avg Score: 98%	CSS	Medium
Remove duplicate JavaScript and CSS	A (100)	⬆ Avg Score: 100%	CSS/JS	Medium
Avoid AlphasImageLoader filter	A (100)	⬆ Avg Score: 95%	CSS	Medium
Avoid HTTP 404 (Not Found) error	A (100)	⬆ Avg Score: 97%	Content	Medium
Reduce the number of DOM elements	A (100)	↑ Avg Score: 93%	Content	Low
Use GET for AJAX requests	A (100)	⬆ Avg Score: 100%	JS	Low
Avoid CSS expressions	A (100)	⬆ Avg Score: 95%	CSS	Low
Reduce DNS lookups	A (100)	↑ Avg Score: 78%	Content	Low
Reduce cookie size	A (100)	⬆ Avg Score: 100%	Cookie	Low
Make JavaScript and CSS external	(n/a)		CSS/JS	Medium

## Kuvien optimointi Yahoos Smush.it-työkälulla

Smushed Images			
wc_suihku.jpg	1.54 KB	14.52 KB	90.43%
upper_157.png	292 bytes	2.77 KB	90.68%
upper.png	598 bytes	2.82 KB	82.85%
uima-allas.jpg	1.71 KB	14.65 KB	89.57%
toro.png	7.69 KB	5.00 KB	39.41%
takka.jpg	1.66 KB	12.16 KB	88.01%
suvilla_copyright.jpg	22.74 KB	35.34 KB	60.84%
star.png	267 bytes	82 bytes	23.50%
slider-shadow2.png	415 bytes	111 bytes	21.10%
slider-shadow.png	526 bytes	2.77 KB	84.34%
scroll-right.png	160 bytes	85 bytes	34.69%
scroll-left.png	156 bytes	66 bytes	29.73%
sauna.jpg	1.42 KB	11.64 KB	89.14%
quote.png	396 bytes	123 bytes	23.70%
plus.png	496 bytes	17 bytes	3.31%
parkkipaikka.jpg	1.81 KB	11.86 KB	86.78%
opacity2.png	71 bytes	44 bytes	38.26%
opacity.png	70 bytes	39 bytes	35.78%
nophoto620x360.jpg	11.47 KB	253 bytes	2.11%
nophoto518x360.jpg	9.56 KB	205 bytes	2.05%
nav-slide.png	214 bytes	33 bytes	13.36%
makuuhuoneita.jpg	1.14 KB	14.34 KB	92.62%
m2.jpg	1.53 KB	11.58 KB	88.33%
luelisaa.png	1.64 KB	1.00 KB	37.92%
logo.png	1.62 KB	572 bytes	25.59%
link.gif			No savings
kyltti.png	57.72 KB	13.12 KB	18.52%
ilmastointi.jpg	1.77 KB	11.82 KB	87.00%
icon2.png	141 bytes	84 bytes	37.33%
icon1.png	191 bytes	89 bytes	31.79%
icon1.gif			No savings
icon-star.png	201 bytes	76 bytes	27.44%
hr.gif			No savings
hover-opacity.png	1.50 KB	696 bytes	31.25%
hover-opacity-small.png	2.11 KB	105 bytes	4.64%
house-icon.png	14.82 KB	2.65 KB	15.18%
house-icon-small.png	5.30 KB	2.65 KB	33.33%
hissi.jpg	1.33 KB	11.99 KB	90.05%
h2_trigger_a.png	325 bytes	171 bytes	34.48%
footerbg.jpg	18.74 KB	24.05 KB	56.20%
dropcaps.png	799 bytes	367 bytes	31.48%
dot.png	126 bytes	48 bytes	27.59%
dot-line.gif			No savings
circle.png	238 bytes	51 bytes	17.65%
check.png	249 bytes	65 bytes	20.70%
bottom_bg.png	3.52 KB	4.53 KB	56.24%
bg2.png	178.07 KB	66.68 KB	27.24%
bg-shadow.gif.png	369 bytes	1.41 KB	79.59%
bg-copyright.gif			No savings
bg-container-top.gif			No savings
autotalli.jpg	1.66 KB	14.75 KB	89.86%
asunto_myyntiin.png	25.02 KB	6.09 KB	19.57%
arrow.png	132 bytes	73 bytes	35.61%

# Etusivun mittaustulosten ennen ja jälkeen -vertailu

**GTmetrix** The web should be fast. **Executive Summary**



## Performance Report for: <http://suvilla.com/>

Report generated: Monday, February 27, 2012, 3:56 AM -0800  
 Test Server Region: Vancouver, Canada  
 Using: Firefox 9.0.1, Page Speed 1.12.0.1, YSlow 3.0.4

Page Speed Grade: **(62%)** ↓ Avg: 73%  
 YSlow Grade: **(66%)** ↓ Avg: 75%

Page load time: 2.60s | Total page size: 1.17MB | Total number of requests: 48

### Priority Issues (Top 5)

Leverage browser caching	<b>F (11)</b>	↓ Avg Score: 43%	Server	High
Defer parsing of JavaScript	<b>F (28)</b>	↓ Avg Score: 66%	JS	High
Combine images using CSS sprites	<b>F (28)</b>	↓ Avg Score: 56%	Images	Medium
Optimize images	<b>F (32)</b>	↓ Avg Score: 74%	Images	High
Enable gzip compression	<b>F (36)</b>	↓ Avg Score: 72%	Server	High

### How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, **Google has announced that they are using page speed in their ranking algorithm.**

### About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

### What do these grades mean?

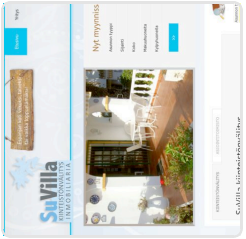
GTmetrix analyzes your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The grades you see represent how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

### What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:

**GTmetrix** The web should be fast. **Executive Summary**



## Performance Report for: <http://www.suvilla.com/>

Report generated: Monday, May 14, 2012, 8:34 AM -0700  
 Test Server Region: Vancouver, Canada  
 Using: Firefox 10.0.2, Page Speed 1.12.0.1, YSlow 3.0.4

Page Speed Grade: **(91%)** ↑ Avg: 74%  
 YSlow Grade: **(90%)** ↑ Avg: 75%

Page load time: 1.50s | Total page size: 720KB | Total number of requests: 26

### Priority Issues (Top 5)

Defer parsing of JavaScript	<b>F (24)</b>	↓ Avg Score: 67%	JS	High
Combine images using CSS sprites	<b>F (25)</b>	↓ Avg Score: 58%	Images	Medium
Use efficient CSS selectors	<b>F (0)</b>	↓ Avg Score: 24%	CSS	Low
Remove query strings from static resources	<b>B (47)</b>	↓ Avg Score: 93%	Content	High
Optimize the order of styles and scripts	<b>B (55)</b>	↓ Avg Score: 95%	CSS/JS	High

### How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, **Google has announced that they are using page speed in their ranking algorithm.**

### About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

### What do these grades mean?

GTmetrix analyzes your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The grades you see represent how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

### What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:

# Kohdelistaussivun mittaustulosten ennen ja jälkeen -vertailu



## Performance Report for: http://www.suvilla.com/asumot/

Report generated: Monday, February 27, 2012, 4:02 AM -0800  
Test Server Region: Vancouver, Canada  
Using: Firefox 9.0.1, Page Speed 1.12.0.1, YSlow 3.0.4

Page Speed Grade: **(66%)** Avg: 73%  
YSlow Grade: **(65%)** Avg: 75%

Page load time: 4.45s | Total page size: 1.70MB | Total number of requests: 119

### Priority Issues (Top 5)

Specify image dimensions	F (0)	↓ Avg Score: 51%	Images	High
Leverage browser caching	F (3)	↓ Avg Score: 43%	Server	High
Defer parsing of JavaScript	F (28)	↓ Avg Score: 68%	JS	High
Enable gzip compression	F (34)	↓ Avg Score: 72%	Server	High
Minify CSS	F (49)	↓ Avg Score: 76%	CSS	High

### How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, Google has announced that they are using page speed in their ranking algorithm.

### About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

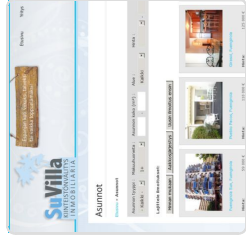
### What do these grades mean?

GTmetrix analyzes your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The grades you see represent how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

### What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:



## Performance Report for: http://www.suvilla.com/asumot

Report generated: Wednesday, May 2, 2012, 4:11 PM -0700  
Test Server Region: Vancouver, Canada  
Using: Firefox 10.0.2, Page Speed 1.12.0.1, YSlow 3.0.4

Page Speed Grade: **(95%)** Avg: 74%  
YSlow Grade: **(86%)** Avg: 75%

Page load time: 2.55s | Total page size: 1.20MB | Total number of requests: 107

### Priority Issues (Top 5)

Specify image dimensions	F (0)	↓ Avg Score: 51%	Images	High
Defer parsing of JavaScript	F (26)	↓ Avg Score: 67%	JS	High
Use efficient CSS selectors	F (0)	↓ Avg Score: 25%	CSS	Low
Combine images using CSS sprites	B (81)	↑ Avg Score: 57%	Images	Medium
Optimize the order of styles and scripts	B (85)	↓ Avg Score: 95%	CSS/JS	High

### How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, Google has announced that they are using page speed in their ranking algorithm.

### About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

### What do these grades mean?

GTmetrix analyzes your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The grades you see represent how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

### What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:

# Myyntikohdesivun mittaustulosten ennen ja jälkeen -vertailu



**Performance Report for:**  
<http://www.suvilla.com/property/las-gaviotas-calahonda/>

Report generated: Monday, February 27, 2012, 4:05 AM -0800  
Test Server Region: Vancouver, Canada  
Using: Firefox 9.0.1, Page Speed 1.12.0.1, YSlow 3.0.4

Page Speed Grade: **(66%)** ↑ Avg: 73%

YSlow Grade: **(65%)** ↓ Avg: 75%

Page load time: 2.39s | Total page size: 1.25MB | Total number of requests: 71

## Priority Issues (Top 5)

Specify image dimensions	F (0)	↓ Avg Score: 51%	Images	High
Optimize images	F (7)	↓ Avg Score: 74%	Images	High
Defer parsing of JavaScript	F (10)	↓ Avg Score: 66%	JS	High
Leverage browser caching	F (17)	↓ Avg Score: 43%	Server	High
Combine images using CSS sprites	F (15)	↓ Avg Score: 56%	Images	Medium

## How does this affect me ?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, Google has announced that they are using page speed in their ranking algorithm.

## About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

## What do these grades mean?

GTmetrix analyzes your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The grades you see represent how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

## What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:



**Performance Report for:**  
<http://www.suvilla.com/las-gaviotas-calahonda>

Report generated: Monday, May 14, 2012, 8:36 AM -0700  
Test Server Region: Vancouver, Canada  
Using: Firefox 10.0.2, Page Speed 1.12.0.1, YSlow 3.0.4

Page Speed Grade: **(93%)** ↑ Avg: 74%

YSlow Grade: **(84%)** ↑ Avg: 75%

Page load time: 3.43s | Total page size: 650KB | Total number of requests: 44

## Priority Issues (Top 5)

Specify image dimensions	F (0)	↓ Avg Score: 51%	Images	High
Combine images using CSS sprites	F (0)	↓ Avg Score: 58%	Images	Medium
Use efficient CSS selectors	F (0)	↓ Avg Score: 24%	CSS	Low
Remove unused CSS	C (70.4)	↑ Avg Score: 65%	CSS	Low
Remove query strings from static resources	A (8)	↑ Avg Score: 93%	Content	High

## How does this affect me ?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, Google has announced that they are using page speed in their ranking algorithm.

## About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

## What do these grades mean?

GTmetrix analyzes your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The grades you see represent how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

## What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings: