



C++ webservers als alternatief voor Apache

Naam: Redacted

Klas: IS104

Studentnummer: Redacted

Faculteit: HBO-ICT



SAMENVATTING

Apache is een populair webserver voor PHP en is het dominant op het web maar als programmeertaal is C++ populairder dus waarom wordt C++ niet gebruikt voor webservers? Onderzocht is hoe verschillende C++ webservers en frameworks zich verhouden tot Apache i.c.m. PHP. Hiervoor is een prestatie matrix opgezet die aan te passen is op wensen van andere waaruit de geprefereerde C++ webserver kan worden gehaald. In de prestatie matrix worden zowel bekwaamheden als gedragingen getoetst. Uiteindelijk bleek de CppCMS webserver zich het beste te verhouden tegen Apache i.c.m. PHP. Apache & PHP scoorde 29 punten waar CppCMS 27 punten scoorde, de webservers Wt en Tree frog framework zijn ook onderzocht deze scoorde 21 en 20 punten.

Author Keywords

C++; webserver; webapplicaties; apache; php, java.



INHOUDSOPGAVE

C++ webservers als alternatief voor Apache 1

SAMENVATTING 2

 Author Keywords 2

INHOUDSOPGAVE 3

INLEIDING 5

1. IDENTIFICEREN VAN C++ WEBSERVERS 5

 1.1. Hoe kan het aanbod van C++ webservers worden geïdentificeerd? 5

 1.2. Wat zijn de verschillen tussen webservers, webapplicaties, webframeworks en webservices? 5

 1.2.1. Webservers 5

 1.2.2. Webapplicaties 5

 1.2.3. Webframeworks 5

 1.2.4. Webservices 5

2. HET AANBOD VAN C++ WEBSERVERS 5

 2.1. Zijn in andere onderzoeken al mogelijkheden voor C++ webservers ontdekt? 5

 2.2. Wat is het aanbod van C++ webservers? 5

 2.3. Welke van C++ webservers zullen worden onderzocht? 6

 2.4. Waarom is voor deze C++ webservers gekozen? 6

3. PRESTATIES VAN C++ WEBSERVERS METEN 6

 3.1. Hoe meten we de prestaties van C++ webservers? 6

 3.2. Hoe worden prestatie gedefinieerd? 6

 3.3. Wat zijn de voorwaarde voor elke prestatie? 7

 3.3.1. Hoe snel de server reageert 7

 3.3.2. Hoe veel geheugen de server verbruikt 7

 3.3.3. Maximum aantal verbindingen per seconden 7

 3.3.4. Headers opzetten en aanpassen 7

 3.3.5. Cookies beheren en bijhouden 7

 3.3.6. SSL/TLS ondersteuning 8

 3.3.7. Compressie ondersteuning 8

 3.3.8. MVC ondersteuning 8

 3.3.9. CSRF bescherming 8

 3.3.10. XSS bescherming 8

 3.4. Wat zijn mogelijke testmethodieken? 8

4. DE PRESTATIES VAN DE C++ WEBSERVERS? 8

 4.1. Hoe presteert het Tree Frog Framework ? 10

 4.2. Hoe presteert CppCMS ? 10

 4.3. Hoe presteert Wt ? 10

5. CONCLUSIE 11



6. DISCUSSIE.....11
NOTITIES.....11
REFERENTIES.....12
BIJLAGE.....12



INLEIDING

Voor websites worden verschillende programmeertalen gebruikt, de meest populairsten[3] zijn Java, C# en PHP*. Buiten deze programmeertalen valt de taal C++[4], dit ondanks de grote populariteit van deze taal[3]. Het toepasbaar en toegankelijk maken van C++ voor webontwikkeling is vanwege de grote adoptie erg interessant, dit zou een potentie zijn voor een nieuwe vorm van webontwikkeling. Onderzocht is in hoeverre C++ gebruikt kan worden als programmeertaal voor websites. De prestaties van deze taal in verhouding tot populaire weboplossingen zoals Java en PHP. Als webserver platform voor Java en PHP is Apache[5] versie 2.4.7 gebruikt. Apache is de meest dominante webserver met 40% marktdeel in 2015[11]. Door de grote populariteit en ondersteuning is het uitstekend vergelijkingsmateriaal.

* php is een recursive afkorting en betekent “php hypertext processor”, waar de php in de afkorting php voor: “personal home page” staat. Het geheel betekent dus “personal homepage hypertext preprocessor”.

1. IDENTIFICEREN VAN C++ WEBSERVERS

1.1. Hoe kan het aanbod van C++ webservers worden geïdentificeerd?

Om C++ webservers te herkennen moet eerst duidelijk worden gemaakt wat de kenmerken ervan zijn. Een webserver is een programma verantwoordelijk voor het afhandelen van aanvragen voor een webpagina. Tijdens het verwerken van zoon aanvraag wordt een programma uitgevoerd afhankelijk van de url en de uitvoer daarvan wordt teruggestuurd naar degene die de webpagina bezoekt. Bij C++ webservers is het van belang dat het programma dat de uitvoer naar de bezoeker verzorgd is geschreven in de programmeertaal C++.

1.2. Wat zijn de verschillen tussen webservers, webapplicaties, webframeworks en webservices?

Om duidelijk te maken wat er daadwerkelijk onderzocht wordt is het belangrijk om de interpretaties tussen webservers, webapplicaties, webservices en webframeworks in kaart te brengen. De volgende omschrijvingen maken duidelijk hoe de termen verder in dit document zullen worden geïnterpreteerd.

1.2.1. Webservers

Onder webservers wordt verstaan het programma verantwoordelijk voor het afhandelen van aanvragen voor webpagina's en het terugkoppelen van de aangevraagde informatie. Webservers koppelen de informatie van webapplicaties terug naar degene die de website bezoekt, ook wel client genoemd. Sommige van de getesten applicaties zijn zowel webserver als webapplicatie.

1.2.2. Webapplicaties

Onder webapplicatie verstaan we een uitvoerbaar programma die informatie teruggeeft na het uitvoeren, deze informatie kan worden opgevangen door een webserver en vervolgens door de webserver worden teruggekoppeld aan de gebruiker. Veel webapplicaties zullen gebruik maken van een of meerdere webframeworks.

1.2.3. Webframeworks

Webframeworks zijn programma's die interactie voor webapplicaties makkelijker maken door middel van voorgeprogrammeerde functionaliteiten.

1.2.4. Webservices

Een webservice is een dienst via een webadres beschikbaar gemaakt, hier is zowel een webserver als webapplicatie voor nodig en vaak zal er een of meerdere webframeworks bij worden gebruikt. Hierbij hoeft de webserver niet hetzelfde programma als de webapplicatie te zijn.

2. HET AANBOD VAN C++ WEBSERVERS

2.1. Zijn in andere onderzoeken al mogelijkheden voor C++ webservers ontdekt?

Verschillende onderzoeken hebben aangetoond dat C++ webservers voor “Extensible Markup Language” (XML) of “Simple Object Access Protocol” (SOAP) applicaties geheel mogelijk zijn en zelf commercieel gebruikt worden[1,2]. De onderzoeken beperken zich vaak op een webserver of framework dit laat onderzoeksmogelijkheden over voor het vergelijken van Webservers die “HyperText Markup Language” (HTML) applicaties voorzien.

2.2. Wat is het aanbod van C++ webservers?

Een geruim aanbod is voor C++ webservers beschikbaar ondanks dat het niet vaak gebruikt lijkt te worden, een van de meest toegankelijk mogelijkheden voor het opzetten van een C++ webserver is: het gebruiken van Apache i.c.m. “common gateway interface” (CGI) of FastCGI. CGI voert systeem executables uit en voert de console uitvoer terug

naar de webserver. Er bestaat ook een framework om CGI van additionele functionaliteit te voorzien genaamd CGICC[IV]. Naast webservers die op CGI leunen om C++ faciliteiten te voorzien zijn er ook veel webservers die C++ ondersteuning geheel zelf hebben gefaciliteerd. C++ server pages[IV] is daar een van, ondanks de naam gaat het hier om een framework waar veel functionaliteit is toegevoegd om het ontwikkelen van webapplicaties te vereenvoudigen. Er zijn zelfs C++ webservers gebaseerd op in andere programmeertalen gemaakte frameworks zoals Node Native[IV]. Bij deze webserver is het populaire NodeJS[12] omgezet tot een C++ webserver Node Native is wel voor het laatst in 2013 bijgewerkt en er kan dus worden aangenomen dat er niet langer aan wordt gewerkt. HydraExpress is een webserver niet gericht op het “hypertext transfer protocol” (HTTP) maar op XML, SOAP en “Web Services Description Language” (WSDL). HydraExpress richt zich op “service oriented architectures” (SOA). Naast HydraExpress is er ook gSoap een SOA gerichte C++ webserver waar ook meerdere research papers over zijn geschreven, opvallend is dat al deze research papers zijn geschreven door de oprichter van het bedrijf achter gSoap. Er zijn dus veel mogelijkheden voor C++ webservers sommige beter onderhouden dan andere, veel C++ webserver bieden ook een framework aan om samen met de webserver te gebruiken. Opvallend is ook dat veel C++ webservers niet gemaakt zijn voor HTML maar XML, SOAP en WSDL. De hierboven genoemde C++ webservers zijn niet de onderzochten webservers er zijn andere C++ webservers gevonden waarvan hun waarde significant is ingeschat, deze worden in 5.3 behandeld.

2.3. Welke van C++ webservers zullen worden onderzocht?

Zoals eerder genoemd zijn er veel C++ webservers met allemaal hun eigen kwaliteiten en functionaliteit. Hiervan zijn vanuit optiek de drie meest interessante meegenomen in het onderzoek. De drie webservers zijn als volgt:

1. CppCMS[8]
2. Tree frog framework[9]
3. Wt[10]

2.4. Waarom is voor deze C++ webservers gekozen?

Voor deze webservers is gekozen omdat ze ondermeer gericht zijn op het serveren van HTTP pagina net zoals Apache. Verder bieden al deze webservers extra functionaliteit om HTTP verkeer te vereenvoudigen. Sommige van de onderzochten webservers bieden ook extra functionaliteit zoals het ondersteunen van “Model View Controller” (MVC)*.

* bij MVC maken we onderscheid tussen het deel dat de logica voorziet (controller), een deel dat het data model voorziet (model) en een deel dat de layout bepaald (view).

3. PRESTATIES VAN C++ WEBSERVERS METEN

3.1. Hoe meten we de prestaties van C++ webservers?

Voor het testen van webservers is een virtuele machine opgezet met een volledige software suite[V]. Voor de hercontroleerbaarheid van onderzoeksresultaten is een zogeheten ‘virtual appliance’ gecreerd, deze kan in veel virtuele machine software worden geïmporteerd[II]. Voor het meten van prestaties en om daar een waarde aan hechten is een prestatie matrix gemaakt. Hierin worden een serie prestatie gedefinieerd waarvan elk een aantal punten waard is. De punten zijn opgesteld vanuit optiek maar door beïnvloeden van de prestatie matrix kan op basis van ieders eigen voorkeuren de best presterende webserver worden gevonden.

3.2. Hoe worden prestatie gedefinieerd?

De prestaties zijn opgedeeld in verschillende groepen: gedrag en bekwaamheden. Bij gedrag worden er vergelijkingen gemaakt met de andere webservers. Bij bekwaamheden beoordelen we of een bepaalde functionaliteit beschikbaar is. Omdat het hier om een het meten van een webserver gaat en niet een webframework of webapplicatie wegen de gedragingen zwaarder dan meesten bekwaamheden. De uitzondering hierop is voor bekwaamheden welke alleen door een webserver kunnen worden geïmplementeerd, deze bekwaamheden zijn 3 punten waard.



Prestatie matrix		
punten	categorie	prestatie
{5,3,2}	gedrag	Hoe snel de server reageert
{5,3,2}	gedrag	Hoe veel geheugen de server verbruikt
{5,3,2}	gedrag	Maximum aantal verbindingen per seconden
1	bekwaamheden	Headers opzetten en aanpassen
1	bekwaamheden	Cookies beheren en bijhouden
3	bekwaamheden	SSL/TLS ondersteuning
3	bekwaamheden	Compressie ondersteuning
1	bekwaamheden	MVC ondersteuning
1	bekwaamheden	XSS bescherming
1	bekwaamheden	CSRF bescherming
2	bekwaamheden	Operating system ondersteuning: linux
2	bekwaamheden	Operating system ondersteuning: unix
1	bekwaamheden	Operating system ondersteuning: os x
1	bekwaamheden	Operating system ondersteuning: windows

3.3. Wat zijn de voorwaarde voor elke prestatie?

3.3.1. Hoe snel de server reageert

De reactiesnelheid wordt gemeten door de tijd die verstrijkt tussen het moment dat een webpagina wordt aangevraagd en door de webserver wordt teruggegeven. Reactiesnelheid wordt gebruik als data voor hoe snel de webserver reageert en wordt gemeten in milliseconden. Bij het nagaan van het maximum aantal verbindingen per seconden zal de gemiddelde reactietijd worden genomen om te bepalen hoe snel de webserver reageert.

3.3.2. Hoe veel geheugen de server verbruikt

Bij het meten van geheugen wordt het totale geheugen in megabytes van het process van de webserver gemeten. Als de webserver meerdere processen opstart wordt het totaal van de meerdere processen gemeten. Gedeelde processen worden niet gemeten, bijvoorbeeld: Wanneer de webserver openssl gebruikt maar i.p.v. openssl als subprocess te starten er een “stand-alone” openssl process wordt aangesproken. De piek van het geheugen gebruik tijdens het meten van het maximum aantal verbindingen per seconden zal worden genomen als de hoeveelheid geheugen die de server verbruikt.

3.3.3. Maximum aantal verbindingen per seconden

Het aantal verbindingen per seconden wordt gemeten d.m.v. “the low orbit ion cannon”(LOIC)[13]. Dit programma is instaat verbindingen met webservers velen malen per seconden aan te gaan en bij te houden hoe vaak de webserver instaat is antwoord te geven. Het maximum aantal verbindingen is het aantal verbindingen per seconden zonder dat 1 verbinding geen resultaat levert.

3.3.4. Headers opzetten en aanpassen

Wanneer de webserver instaat is de http headers aan te passen zoals gedefinieerd in RFC2616[14], de webserver dient d.m.v. een framework hier zelf functionaliteit voor te hebben ingebouwd en niet afhankelijk te zijn van programma's van derden. Deze headers dienen voor het inlichten van de programma's van verschillende mogelijkheden en diensten.

3.3.5. Cookies beheren en bijhouden

Het bijhouden van cookies is het process waarin headers van en naar de webserver zo worden verstuurd dat we voor de gebruiker van de webserver de identiteit kunne vaststellen. Hiervoor moet de webserver instaat zijn headers op te zetten en aan te passen maar ook een zogeheten sessie bij kunnen houden op basis van deze headers. Deze functionaliteit is vaak onderdeel van een Framework.

3.3.6. SSL/TLS ondersteuning

“Secure Socket Layer” (SSL) of “Transport Layer Security” (TLS) ondersteuning wordt toegekend wanneer minimal versie 1.1 van TLS of versie 3 van SSL wordt ondersteunt door de webserver. De webserver moet zelf deze ondersteuning bieden dat wil zeggen dat hij niet gebruik mag maken van andere programma’s voor het serveren van webpagina’s zoals bijvoorbeeld bij het gebruik van CGI of FastCGI. Wel mogen er voorzieningen van derde partijen worden gebruikt om de encryptie die gedurende TLS en SSL sessies plaatsvinden te ondersteunen.

3.3.7. Compressie ondersteuning

In de HTTP standaard zijn methoden gespecificeerd om bestanden en data die wordt verstuurd te comprimeren voor het verzenden. Dit gebeurt door het meesturen van bepaalde headers en het daadwerkelijk comprimeren van het bestand. De webserver voldoet aan compressive ondersteunen al seen van de comprimatie methode zoals gedefinieerd in de HTTP standaard wordt ondersteunt.

3.3.8. MVC ondersteuning

Wanneer er vanuit de webserver d.m.v. een framework de mogelijkheid wordt geboden om een class te extenden vanuit het framework die een controller, een model en een view representeren. De classe moeten vanuit het traditionele MVC model data uitwissel. Een controller laadt modellen welke data ophalen en versturen die vervolgens door de controller aan views kan worden meegegeven.

3.3.9. CSRF bescherming

“Cross-site request forgery” (CSRF) lijkt op “cross-site scripting” (XSS) maar is net een iets andere vorm van web-aanval. In plaats van dat er d.m.v. POST en GET parameters* malafide data aan de geladen webpagina wordt toegevoegd, worden bij CSRF aanvallen de cookie gegevens van gebruikers misbruikt om de aanvaller zich voor te laten doen als de gebruiker. De webserver beschikt over CSRF bescherming als de cookie parameters kunnen worden voorzien van tokens die elke keer dat de webpagina laadt worden aangepast en iedere keer worden gecontroleerd.

* POST en GET parameters zijn data attributen die mee kunnen worden verstuurd zoals gebruikersnaam en wachtwoord.

3.3.10. XSS bescherming

XSS berscherming is het filteren van POST en GET parameters doormiddel van functionaliteit voorzien door methode of classen hoogstwaarschijnlijk uit een Framework. De webserver voldoet aan XSS bescherming wanneer er methode geïmplementeerd zijn die specifiek bedoeld zijn voor het filteren van XSS aanvallen.

3.4. Wat zijn mogelijken testmethodieken?

Voor het testen zal elke C++ webserver worden geïnstalleerd en de basis functionaliteit worden opgezet. Daarna zal elke webserver een aantal taken worden gegeven op basis van de functionaliteit van de C++ webservers gezamenlijk. Dat wil zeggen dat de webserver een taak krijgen die bij elke webserver gelijk is en is gebaseerd op de functionaliteiten waar ze alle 3 over beschikken. Hierdoor is er niet een webserver in het voordeel door het niet beschikken over functionaliteit en het dus niet hoeven uitvoeren van de taak. Het gehele testplatform wordt geconfigureerd in een virtuele machine met daarop Ubuntu 15.10 en een totaal software pakket. Dit software pakket kan worden opgezocht in de notities.

4. DE PRESTATIES VAN DE C++ WEBSERVERS?

Op basis van eerder omschreven test methodieken zijn de verschillende C++ webservers getest. De resultaten zijn in een tabel overzichtelijk gemaakt. Ook zijn de prestaties van de Apache webserver i.c.m. PHP 5.6 gemeten en meegenomen in dit overzicht. Het eerste overzicht zijn de ruwe gegevens voor het meten van gedrag.



prestatie	CppCMS	Wt	Tree Frog Framework	PHP 5.6 + Apache 2.4.7
Hoe snel de server reageert	0.11 seconden	0.29 seconden	N.V.T	0.19 seconden
Hoe veel geheugen de server verbruikt	9 mb	50 mb	4.5 mb	5 mb
Maximum aantal verbindingen per seconden	472	160	1	995

Later wordt dieper ingegaan op de resultaten en hoe deze behaald zijn voor elke webserver. Voor nu zijn op basis van de bovenstaande gegevens de volgende punten toebedeeld, hierbij is het belangrijk dat we voor het meten van de nummers 1, 2 en 3 bij gedragingen de Apache webserver i.c.m. PHP niet mee tellen in de rang orde maar wel punten toekennen voor de verhouding waarin deze webserver presteert.

prestatie	CppCMS	Wt	Tree Frog Framework	PHP 5.6 + Apache 2.4.7
Hoe snel de server reageert	5	3	2	5
Hoe veel geheugen de server verbruikt	3	2	5	5
Maximum aantal verbindingen per seconden	5	3	2	5
Headers opzetten en aanpassen	1	1	1	1
Cookies beheren en bijhouden	1	1	1	1
SSL/TLS ondersteuning	0	3	0	3
Compressie ondersteuning	3	0	0	3
MVC ondersteuning	1	0	1	0
XSS bescherming	1	1	1	0
CSRF bescherming	1	1	1	0
Operating system ondersteuning: linux	2	2	2	2
Operating system ondersteuning: unix	2	2	2	2
Operating system ondersteuning: os x	1	1	1	1
Operating system ondersteuning: windows	1	1	1	1
Totaal	27	21	20	29

Uit de resultaten blijkt dat CppCMS de beste resultaten geeft daarop volgt wt en tot slot het TreeFrogFramework. Het beste uit de resultaten kwamen Apache en PHP wat naar mijn mening met de gekozen test opstelling voor de hand legt meer informatie hierover in discussie.

4.1. Hoe presteert het Tree Frog Framework ?

Van alle 3 de C++ webservers heft het tree frog framework de laagste score uit de prestatie matrix behaald. Veel van het punt verlies was te wijten aan een probleem dat ondervonden werd tijdens het testen. Wanneer er meer dan 1 verbinding tegelijkertijd met de webserver werkt gemaakt ontstond er een intern probleem in het framework en weigerde de C++ webserver verder afhandeling. Hierdoor kon er ook gemiddelde afhandel tijd voor een enkele webpagina worden gemaakt . De webserver scoorde hierdoor zowel laag op “Maximum aantal verbindingen per seconden” als op “Hoe snel de server reageert”. Deze technische problemen zouden het webserver in de praktijk ook moeilijk te gebruiken maken doordat een webserver vaak meerdere afhandelingen per seconden moet verrichten.

4.2. Hoe presteert CppCMS ?

CppCMS kwam het beste uit de test en scoorde zowel hoog in gedrag als bekwaamheden. Vooral opvallend was de compleetheid van CppCMS als webserver en als framework de enige bekwaamheid die de webserver kant liet vallen was het ondersteunen van SSL/TLS voor het encoderen van internet verbindingen. Ondanks deze complexiteit en volledigheid gingen de prestaties in verhouding tot andere webservers niet merkbaar achteruit.

4.3. Hoe presteert Wt ?

Opvallend was dat Wt als enige SSL/TLS ondersteunde en allerlei functionaliteit had voor het opzetten van webpagina's van formulieren tot chat applicaties. Hieruit valt af te leiden dat hun framework zich niet alleen op de C++ kant richt maar ook op andere onderdelen van webtalen zoals Javascript, Html en CSS. Deze vallen echter buiten het onderzoek en er zijn hier dus geen punten voor toegekent. Wt gebruikten meer geheugen dan de andere webservers dit valt te verklaren door de grote hoeveelheid extra functionaliteit met betrekking tot de eerder



genoemde webtalen. Wt heeft geen punten toegekent gekregen voor het ondersteunen van MVC toch biedt Wt functionaliteit voor het gebruik van modellen en controllers helaas ontbreekt het ondersteunen van views hierom zijn geen punten toegekent voor MVC ondersteuning.

5. CONCLUSIE

Er zijn meerdere C++ webserver beschikbaar die zich kunnen verhouden met Apache i.c.m. PHP. Om deze verhoudingen te meten is een prestatie matrix met verschillende gedragingen en bekwaamheden opgezet, de webservers zijn hieraan onderworpen en daaruit blijkt dat de CppCMS webserver het beste presteert. De prestatie matrix kan eenvoudig worden aangepast om op basis van wat belangrijk bevonden wordt een ander resultaat te krijgen. CppCMS heeft 27 punten behaald waar Apache er 29 heeft ze presteren in verhouding bijna gelijk maar Apache presteert volgens de prestatie matrix net iets beter. Het tree frog framework presteerde het slechts dit was grotendeels te danken aan een probleem rondom het serveren van webpagina's. De C++ webserver maken het mogelijk om C++ te gebruiken voor het web en met deze webservers zou C++ zich goed kunnen verhouden als webtaal.

6. DISCUSSIE

Voor het testen van de webservers zijn zogeheten test scripts opgezet dit is de code die is uitgevoerd om de gedragingen van de webservers te testen, deze was in dit onderzoek eenvoudig interessant zou zijn hoe de C++ webservers zich verhouden met Apache ic.m. PHP wanneer complexere berekingen moeten worden gemaakt. Ook zou het interessant zijn om de C++ webservers alleen als framework te gebruiken en de webserver kant over te laten aan Apache iets waar alle 3 de getesten webserver functionaliteit voor hebben. Tot slot zou het interessant zijn om gelijkwaardige scripts in zowel PHP als C++ te schrijven en bij Apache het verschil in prestaties tussen de twee te testen aangezien Apache C++ kan uitvoeren via zowel CGI als FastCGI, Ook zouden de prestaties van CGI en FastCGI nog kunnen worden getest. Het gebruik C++ voor webservers lijkt in een infantiele staat terwijl er goede frameworks voor het gebruik van C++ voor het web beschikbaar zijn.

NOTITIES

- I. Veel C++ web frameworks leggen de focus niet op standaard html pagina's maar op XML, SOAP en WSDL.
- II. Virtual Appliance is onder voorbehoud beschikbaar gemaakt op de volgende webpagina <https://dantalion.nl/cpp-webservers-als-alternatief-voor-apache>
- III. Virtual Appliance username & password:
 - a. cpptesting
 - b. testingcpp1
- IV. Naast de drie onderzochten frameworks zijn er nog andere alternatieven gevonden zoals:
 - a. [CGICC](#)
 - b. [C++ Server Pages](#)
 - c. [Node Native](#)
 - d. [HydraExpress](#)
 - e. [gSoap](#)
- V. Definitie van 'virtuele machine software suite'
 - a. Ubuntu 15.10
 - b. Apache 2.4.12
 - c. PHP 5.6.11
 - d. Git 2.5.0
 - e. Cmake 3.2.2
 - f. Qmake 4.8.6
 - g. QT 5.4.2
 - h. Netbeans IDE 8.1 ALL bundle
 - i. Apache tomcat 8.0.27
 - j. Glassfish server open source edition 4.1.1
 - k. Java jdk 1.8.73
 - l. Libboost all 1.58 (development version)
 - m. Libprce 3 (development version)
 - n. Libqt 4 (development version)
 - o. Libqt sql mysql 4 (development version)



- p. Libqt sql psql 4 (development version)
- q. Zlibg 1.2.8 (development version)
- r. Libcrypt 11 (development version)
- s. Libicu (development version)
- t. CppCMS 1.0.5
- u. Wt 3.3.5
- v. TreeFrog Framework 1.10.0

REFERENTIES

1. Engelen, R. A. (2002). *The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks*. Geraadpleegd van <http://www.cs.fsu.edu/~engelen/ccgrid.pdf>
2. Ji, X., Han, J., & Zhao, Y. (2013). *A Code Generation Toolkit for C++ Web Services Development*. Geraadpleegd van <http://sci-hub.io/10.1109/ISDEA.2012.11>
3. Wikimedia. (z.j.). C++. Geraadpleegd van <https://en.wikipedia.org/wiki/C%2B%2B>
4. IEEE. (z.j.). top tien programmeertalen van 2015. Geraadpleegd van <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>
5. APACHE SOFTWARE FOUNDATION. (z.j.). About HTTP server project. Geraadpleegd op 23 maart, 2016, van http://httpd.apache.org/ABOUT_APACHE.html
6. Wikimedia. (z.j.). Software Framework. Geraadpleegd op 23 maart, 2016, van https://en.wikipedia.org/wiki/Software_framework
7. Tutorialspoint. (z.j.). C++ Web Programming. Geraadpleegd op 23 maart, 2016, van http://www.tutorialspoint.com/cplusplus/cpp_web_programming.htm
8. CPPCMS. (z.j.). C++ webserver. Geraadpleegd op 23 maart, 2016, van <http://cppcms.com/wikip/en/page/main>
9. Tree frog framework. (z.j.). C++ webserver. Geraadpleegd op 23 maart, 2016, van <http://www.treefrogframework.org/>
10. Wt. (z.j.). C++ web webserver. Geraadpleegd op 23 maart, 2016, van <http://www.webtoolkit.eu/wt>
11. NETCRAFT. (z.j.). Webserver survey Januari 2015. Geraadpleegd van <http://news.netcraft.com/archives/2015/01/15/january-2015-web-server-survey.html>
12. NODE. (z.j.). NodeJS informatie. Geraadpleegd op 23 maart, 2016, van <https://nodejs.org/en/>
13. LOIC. (z.j.). LOIC project site. Geraadpleegd op 29 maart, 2016, van <https://github.com/NewEraCracker/LOIC>
14. RFC. (1999). HTTP 1.1. Geraadpleegd van <https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>

BIJLAGE

1. Beoordeling onderzoeksrapport door Redacted.
2. Beoordeling onderzoeksrapport door Redacted.