

FH JOANNEUM Gesellschaft mbH

High Performance und Hardening
Konfiguration von Virtuellen Servern oder Root-Servern

Bachelorarbeit 2
zur Erlangung des akademischen Grades
Bachelor of Science in Engineering (BSc)
eingereicht am 02. September 2012
Fachhochschul-Studiengang Internettechnik

Betreuer: DI (FH) MSc Mathias Knoll, MSc

eingereicht von: Richard Fussenegger
Personenkennzahl: 0910418059

September 2012

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Wolfurt, 02. September 2012; *Fussenegger Richard*

Zusammenfassung

Die Erarbeitung einer Webserver-Konfiguration für kleine bis große Web-Projekte liegt im Fokus der vorliegenden Bachelorarbeit. Die resultierenden Konfigurationen sind Grundlage für Benchmarks und Hochrechnungen die als Vergleichswerte für andere Testergebnisse dienen. Der Hauptfokus der Einstellungen liegt hierbei auf Geschwindigkeit, Ressourcenverbrauch und Sicherheit.

Der erste Teil befasst sich mit verschiedenen Kriterien die eine Auswahl des richtigen Hosting-Produktes erleichtern. Die so erarbeiteten Entscheidungsmerkmale bestimmen die Hard- und Software der Testumgebung. Der darauf folgende Abschnitt widmet sich voll und ganz der Konfiguration des Servers. Nach jedem größeren Systemeingriff folgen Tests die den Geschwindigkeitszuwachs durch die Änderungen belegen. Der letzte Abschnitt befasst sich mit möglichen Erweiterungen die Geschwindigkeit und Sicherheit erhöhen können. Am Schluss folgt ein Resümee, in dem auf die abschließende Konfiguration eingegangen wird und die Testergebnisse näher behandelt werden.

Abstract

The development of a web server configuration for small to large web projects is the focus of this thesis. The resulting configurations are the basis for benchmarks and projections that serve as reference for other test results. The main focus of the settings is on speed, security and resource consumption.

The first part deals with various criteria to facilitate the selection of the right hosting product. The so developed decision characteristics determine the hardware and software of the test system. The next section is devoted to fully configure the server. After each major system intervention tests follow that prove the increase in speed by the changes. The last section deals with possible extensions that can increase speed and security. At the end a résumé follows, which recaps the final configuration and the test results are discussed in more detail.

Danksagung

Ein ganz besonderes Dankeschön geht an die EDIS GmbH in Graz. Sie waren so freundlich mir zwei Virtual Private Server (VPS) kostenlos zu Testzwecken in Österreich zur Verfügung zu stellen. Die Konfigurationen und Tests hätten ohne die Unterstützung der EDIS GmbH nicht in einer realen Umgebung stattfinden können und wären dadurch in ihrer Aussagekraft stark gemindert worden.

Bei Herrn Gerhard Kleewein der EDIS GmbH möchte ich mich im Speziellen für den netten Kontakt und die Hilfe bei der Auswahl an den am besten geeigneten Produkten bedanken.

<https://www.edis.at/>

Inhaltsverzeichnis

1. Einleitung.....	9
1.1 Motivation.....	10
1.2 Abgrenzung.....	11
1.3 Aufbau und Struktur	11
2. Serverauswahlkriterien	13
2.1 Standort.....	13
2.2 Recht und Gerichtsbarkeit	13
2.3 Hardware.....	15
2.4 Betriebssystem	17
2.5 Software	18
2.5.1 Auswahlkriterien.....	19
2.5.2 Programmiersprache	20
2.5.3 Content Management System	21
2.5.4 Datenbanksystem	22
2.5.4.1 Percona Server	22
2.5.4.2 MariaDB	23
2.5.5 Webserver	24
2.5.5.1 Apache HTTP Server.....	24
2.5.5.2 nginx	25
2.5.5.3 Weitere Alternativen.....	25
2.5.5.4 Vergleich.....	27
2.6 Testumgebung.....	29
2.7 Finaler Testaufbau	30
3. Konfiguration.....	32
3.1 Systeminbetriebnahme.....	32
3.2 OpenSSH Daemon.....	34
3.3 Paketverwaltung.....	36
3.4 nginx	39

3.4.1 Erster Benchmark	42
3.4.1.1 Testablauf.....	43
3.4.1.2 Ergebnisse – Erster Durchlauf	44
3.4.2 Systemkonfiguration tunen	46
3.4.3 Zweiter Benchmark.....	48
3.4.4 Dritter Benchmark	49
3.5 PHP und MariaDB	51
3.5.1 PHP-Konfiguration	53
3.5.2 MariaDB-Konfiguration	53
3.5.2.1 Hugepages.....	54
3.6 Drupal installieren.....	55
3.7 Finaler Benchmark.....	56
3.7.1 Siege Ergebnisse	57
3.8 Firewall	59
4. Konfigurationserweiterungen	61
4.1 CloudFlare	61
4.2 Asynchrone Frameworks	61
4.3 Hardening.....	62
5. Resümee.....	64
Anhänge	67
Anhang 1 sshd-Konfiguration.....	68
Anhang 2 nginx-Konfiguration.....	68
Anhang 3 nginx init-Skript	69
Anhang 4 nginx Cachezonen-Konfiguration	70
Anhang 5 nginx GZIP-Konfiguration.....	71
Anhang 6 nginx MIME-Typ-Konfiguration	71
Anhang 7 nginx NoCache-Konfiguration.....	72
Anhang 8 nginx Upstream-Konfiguration	73
Anhang 9 nginx FastCGI-Konfiguration	73

Anhang 10	nginx FastCGI-Drupal-Konfiguration.....	73
Anhang 11	nginx Hidden-Files-Konfiguration.....	74
Anhang 12	nginx PHP-Konfiguration	74
Anhang 13	nginx Proxy-Cache-Konfiguration.....	74
Anhang 14	nginx SSL-Konfiguration.....	74
Anhang 15	nginx Static-Files-Konfiguration	75
Anhang 16	nginx Static-Host-Konfiguration.....	75
Anhang 17	nginx Drupal-Proxy-Konfiguration.....	75
Anhang 18	nginx Drupal-Backend-Konfiguration	76
Anhang 19	PHP-Konfiguration	76
Anhang 20	PHP-FPM-Konfiguration	77
Anhang 21	APC-Konfiguration	78
Anhang 22	MariaDB-Konfiguration.....	78
Anhang 23	MariaDB init-Skript-Shellskript.....	80
Abbildungsverzeichnis.....		81
Quelltextverzeichnis		82
Literaturverzeichnis		84

1. Einleitung

Eine solide Server-Infrastruktur für ein junges Web-Unternehmen, aber auch für bereits bestehende, ist essentiell. Die Erwartungshaltung an eine Web-Applikation ist immer dieselbe. Sie soll 24 Stunden, sieben Tage die Woche zur Verfügung stehen. Doch damit nicht genug, sie soll schnell und effizient sein. Diverse Studien zeigen, dass die Geschwindigkeit einer Website für deren Monetisierung¹ und eine positive Benutzererfahrung extrem wichtig ist (vgl. Website Optimization, 2008). Eine gute Performance hat so direkten Einfluss auf den Erfolg oder Misserfolg eines Projektes (vgl. Farber, 2006). Ein weiterer Grund, weshalb Google seit geraumer Zeit sehr viel Geld und Ressourcen ausgibt um Webmasters² rund um den Globus über das Thema Geschwindigkeit und Ladezeiten aufzuklären (vgl. Brutlag, 2009) (vgl. Brutlag, 2009).

Trotzdem leiden auch heute noch viele Websites darunter langsam zu sein, oder durch teure und überdimensionierte Server eine einigermaßen akzeptable Geschwindigkeit bieten zu können. Wobei genau dies auch ein falscher Schritt sein kann (vgl. McKenzie, 2010). Ein Problem stellen hier die verwendeten Frameworks und die Programmierung der Anwendungen selbst dar. Im Rahmen von Legacy-Systemen³ ist es jedoch oftmals schwierig daran etwas zu ändern. Die oft nicht vorhandene Dokumentation (Anforderungen, Anwendungsfälle, etc.) und der eingeschulte, benötigte Funktionsumfang erschweren oft den Austausch eines solchen Altsystems. Wird ein neues System erstellt, wird darauf leider immer noch zu oft kaum Augenmerk gelegt. Noch viel fataler sieht es jedoch bei der Konfiguration der Server selbst aus. Der Großteil von Web-Entwicklern/innen wissen zwar wie sie ihre Applikation einiger-

¹ Bedeutet im Allgemeinen „Etwas zu Geld machen“ bzw. „in Geldeinheiten bewerten“. Im obigen Zusammenhang ist damit gemeint, wie viel Geld aus dem Projekt im Endeffekt herausgeholt werden kann. Die angesprochene Amazon-Studie zeigte, dass 100 ms zusätzliche Ladezeiten einen Umsatzrückgang von 1 % bedeuten.

² Als Webmaster wurden ursprünglich Webserveradministratoren bezeichnet, heute wird der Begriff synonym für Spezialisten verwendet die sich um die Planung, Gestaltung, Entwicklung, Wartung und Verwaltung von Websites im Gesamten oder auch nur in Teilen befassen.

³ Als Legacysystem (zu Deutsch „Altsystem“) werden historisch gewachsene IT-Systeme (Hard- und/oder Software) bezeichnet die oftmals schlecht bis gar nicht dokumentiert wurden und aus verschiedenen Gründen nicht erneuert wurden.

maßen leistungsfähig umzusetzen haben, jedoch nicht wie sie ihre Software auswählen haben, geschweige denn wie diese konfiguriert werden muss.

1.1 Motivation

Heutige Web-Applikationen sehen sich einer steigenden Anzahl an Herausforderungen ausgesetzt. Immer mehr Menschen besitzen Internetzugänge und das WWW sieht jeden Tag neue Angebote. Dabei stellen die Benutzer hohe Anforderungen, die jedoch oft nicht erfüllt werden können. Studien von Unternehmen wie Amazon haben gezeigt, dass die Geschwindigkeit dabei einen entscheidenden Faktor darstellt. So hat z. B. Google bereits die Ladezeiten der Websites mit in die Ranglistenalgorithmen aufgenommen. Die Sicherheit ist ein weiteres essentielles Kriterium. Benutzer und Benutzerinnen verlieren sofort ihr Vertrauen, sollten Lücken, oder gar Datenklau, innerhalb der eigenen Produktreihen publik machen. Größen wie Facebook und Google zahlen deshalb Unsummen an Personen die Probleme melden.

Der Großteil an verfügbarer Literatur in Buchform und im WWW befasst sich mit geschlossenen Themenbereichen. Es gibt kaum Autoren die sich dem Thema im Gesamten widmen und wenn dann ist der Inhalt meist veraltet und der Aufbau konservativ. Hinzu kommt, dass die Zielgruppen meist auf Hobby-Administratoren oder Einsteiger, die vielleicht einmal Administratoren werden wollen, eingeschränkt sind. Insofern soll die Konfiguration die hier präsentiert wird, die neuesten verfügbaren, am weitesten verbreiteten Technologien einbeziehen und zusätzlich den kompletten Vorgang, von der Anmietung bis zur finalen Website, abdecken. Wo angebracht wird auch angeführt welche Unternehmen im deutschsprachigen Raum professionellen Support bieten, sollte es im Betrieb einmal zu Engpässen kommen ist dies sehr wichtig für Firmen als auch Privatpersonen. Auszeiten im Betrieb bedeuten immer Einnahmenverlust und Besucherschwund.

1.2 Abgrenzung

Den Leser/innen wird nicht die Programmierung von performanten Web-Applikationen beigebracht, noch der Umgang mit Linux. Eine weitere klare Abgrenzung ist darin zu finden, dass in den Kapiteln zu Linux ausschließlich auf die für die Erreichung des Ziels – eine schnelle und sichere Server-Konfiguration für Websites – wichtigen Punkte eingegangen wird. Leser/innen müssen sich darüber im Klaren sein, dass sich Technologien sehr schnell verändern. Die im Text vorgestellte Konfiguration und die Testergebnisse sind somit nur für die verwendeten Versionen gültig und lassen sich nicht zwangsläufig auf frühere oder zukünftige Konfigurationen oder Testergebnisse umlegen.

1.3 Aufbau und Struktur

Wie aus dem Titel „High Performance und Hardening – Konfiguration von Virtuellen Servern oder Root-Servern“ hervorgeht, liegt der Fokus auf der Konfiguration eines Servers zur Auslieferung von Webseiten mit Fokus auf Privatpersonen und/oder kleine Unternehmen.

Im zweiten Kapitel wird erörtert welche Kriterien für die Auswahl eines passenden Hosting-Produktes von Relevanz sind. Welche rechtlichen Überlegungen gilt es anzustellen, worauf ist bei der Hardware zu achten und welche Software genießt den größten Verbreitungsgrad. All diese Entscheidungsfaktoren bilden zusammen die Grundlage zur Wahl des später konfigurierten Testsystems.

Das zuvor angesprochene Testsystem ist Hauptbestandteil des dritten Kapitels, das sich ganz der Konfiguration und den Benchmarks widmet. Schritt für Schritt wird dabei darauf eingegangen, wie mit dem just neu erhaltenen Server – das Testsystem – eine funktionstüchtige, performante und sichere Infrastruktur zu installieren und konfigurieren ist. Jedem größeren Systemeingriff folgt ein Benchmark bzw. Lasttest. Die sich daraus ergebenden Daten dienen als Vergleichswerte um die Effekte der einzelnen Konfigurationsschritte zu belegen.

Das vierte Kapitel befasst sich mit möglichen Erweiterungen, beschränkt sich hierbei jedoch ausschließlich auf die Vorstellung dieser. Die Programme werden kurz vorgestellt und erläutert welchen Nutzen ihre Verwendung dem Administrator / der Administratorin bringt. Anhand von weiterführenden Weblinks und Literaturtipps kann der geneigte Leser / die geneigte Leserin sich weitere Informationen beschaffen.

Am Schluss folgt ein Resümee in dem die Testergebnisse etwas kritischer betrachtet werden, welche Bedeutung sie haben und wo die Einschränkungen liegen. Die Gesamtkonfiguration wird anhand von kurzen Kommentaren nochmals rekapituliert und mögliche Engpässe identifiziert.

2. Serverauswahlkriterien

Es existieren unzählige Hosters auf der ganzen Welt. Die Wahl des richtigen Anbieters für die eigene Website ist der erste wichtige Schritt den es zu machen gilt. Dabei müssen diverse Gesichtspunkte beachtet werden. Nachfolgend werden diverse Themen aufgegriffen die bei der Wahl des Hosters und der Hardware ausschlaggebend sind.

2.1 Standort

Um eine optimale Performance für Besucher und Besucherinnen sicherzustellen sollte der Server immer so nahe an dessen Standort sein wie möglich. Dies erleichtert das Routing und verringert die Latenz. Aufgrund der verkürzten Distanz sind beim Routing weniger Zwischenstellen bis zum Ziel vorhanden, dadurch können die Pakete viel schneller zum antwortenden Server weitergeleitet werden. Die Latenz verringert sich aus dem gleichen Grund, hinzu kommt jedoch auch noch, dass die verbindenden Leitungen eine geringere Auslastung aufweisen – im Vergleich zu einem Unterseekabel von Europa in die USA oder gar nach Asien. Globale Projekte können hier auf diverse Dienste zurückgreifen um die Geschwindigkeit von überall auf der Welt sicherzustellen (mehr dazu im Kapitel 4) ohne gleich mehrere Server rund um den Globus selbst betreiben zu müssen. Lokal begrenzte Projekte profitieren aus Suchmaschinenoptimierungssicht von der Verwendung eines Servers aus dem Land auf das das Projekt abzielt (vgl. Choi, 2008). Bei globalen Projekten hingegen spielt der Serverstandort eine untergeordnete Rolle.

2.2 Recht und Gerichtsbarkeit

Besonders für Unternehmen ist dieser Punkt sehr wichtig, da die Strafen die aus einer falschen Standortwahl resultieren oft fatal sein können. Es gilt im Vorfeld abzuklären

ob das Angebot mit den allgemeinen Geschäftsbedingungen (AGB) eines ISPs⁴ vereinbar sind. Viel wichtiger jedoch ist es abzuklären, ob die eigenen Angebote im Land in dem die Server stehen, bzw. der Hostersesshaft ist, legal sind. Dabei kann es sich um etwas Triviales handeln wie z. B. Filme oder Videospiele mit expliziter Gewaltdarstellung für Erwachsene. Deutschland besitzt eine sehr strikte Zensurpolitik⁵, Österreich ist im Vergleich dazu liberal da es keinerlei Zensur besitzt. In diesem Zusammenhang ist es wichtig auf die Gerichtsbarkeit des Anbieters zu achten, diese unterscheidet sich womöglich vom physikalischen Standort der Hardware. Somit könnte es unter Umständen zu weiteren Problemen kommen. Im Idealfall sollten Standort und Gerichtsbarkeit übereinstimmen wodurch sich die Evaluierung stark vereinfacht.

Insofern Nutzerdaten oder gar Zahlungsdaten (Bankverbindungen, Kreditkarteninformationen, etc.) gespeichert werden sollen ist nicht nur eine erhöhte Sicherheit wichtig um Diebstahl dieser Daten vorzubeugen. Auch rechtlich muss geklärt sein ob diese ohne weiteres auf dem Server gespeichert sein dürfen. Eine Übermittlung von österreichischen Nutzerdaten in z. B. die Vereinigten Staaten ist nicht so einfach möglich. Die unterschiedlichen Datenschutzbestimmungen und andere Rechte in Bezug auf Personendaten im Allgemeinen sind hier problematisch. Das beginnt bereits beim Einsatz von Analysetools wie Google Analytics (vgl. Ihlenfeld, 2011), durch dessen permanente Speicherung von eigenen Nutzerdaten eine weitaus kritischere Betrachtungsweise nötig ist.

Im Zweifelsfall ist ein Rechtsbeistand mit Spezialisierung auf Internetrecht und Datenschutz zu Rate zu ziehen. Wer die Kosten scheut oder einfach nicht aufbringen kann sollte im eigenen Land bleiben, oder zumindest innerhalb der EU da die Gesetze hier oft angeglichen sind.

⁴ Die Abkürzung ISP steht für „Internet Service Provider“ (Internetdiensteanbieter). Das Wort dient als Überbegriff für Unternehmen und Firmen die Dienste und Services die im Zusammenhang mit dem Internet stehen anbieten. Das können sowohl Internetzugänge als auch Server sein.

⁵ <http://www.bundespruefstelle.de/> – Die Bundesprüfstelle für jugendgefährdende Medien (BPjM) ist die oberste Prüfstelle in Deutschland die unter dem Deckmantel des Jugendschutzes Medien aller Art zensiert und diese somit illegal in Deutschland werden.

2.3 Hardware

Die Hardwareauswahl spielt bei einem Server eine genau so große Rolle wie z. B. bei einem Gaming-PC für zu Hause. Trotzdem gibt es gewisse Merkmale die im Serverbereich wichtig sind. Server müssen zu jeder Zeit zur Verfügung stehen und werden nie ausgeschaltet. Entsprechend muss die Hardware für den Dauerbetrieb geeignet sein und um Ausfällen effizient entgegenwirken zu können, sollte jede Komponente redundant ausgelegt sein. Fällt einmal eine Komponente aus, muss sofort eine andere deren Arbeit übernehmen um den Dauerbetrieb gewährleisten zu können. Dedizierte Serverhardwarekomponenten bieten in der Regel auch längere Garantielaufzeiten als Consumer-Hardware.

Generell lässt sich vorab im Zusammenhang mit der Geschwindigkeit für die Applikation festhalten, je potenter die Hardware, desto besser. Performancelücken lassen sich oft durch stärkere Hardware leicht kaschieren. Besonders bei Legacy-Systemen die noch auf verfügbarer Hardware aufsetzen ist es in der Regel kostengünstiger die Hardware aufzustocken, statt am Programm selbst oder der verwendeten Software etwas zu ändern. Der Effekt setzt zudem auch sofort ein.

Bei der Entwicklung neuer Systeme ist es vorteilhaft gleich die schnellste und beste Hardware anzumieten. Jedoch dürfen die damit einhergehenden Kosten nicht außer Acht gelassen werden. Besonders Privatpersonen oder kleinere Web-Startups mit einer revolutionären Geschäftsidee und wenig Kapital müssen hier genau kalkulieren. Hardware kann zu jeder Zeit sehr leicht aufgestockt werden, deshalb ist es ratsam zu Beginn auf die Kosten zu achten und wenn genügend Geld vorhanden ist und stärkere Hardware benötigt wird, kann nachgerüstet werden.

Neben den bereits erwähnten höheren Anforderungen an die Hardware in Bezug auf Dauerbetrieb, Redundanz und Garantielaufzeiten sind für die endgültige Performance die typischen Komponenten ausschlaggebend. Dies umfasst den Hauptprozessor (CPU), Arbeitsspeicher (RAM) und die Festplatte (HDD). Neben diesen drei langläufig für Rechenleistung bekannten Bestandteilen spielt die Netzwerkanbindung eine weitere entscheidende Rolle. Während die verwendete Netzwerkkarte meist

nicht von großer Bedeutung ist für den Administrator / die Administratorin, ist die Netzanbindung des Rechners im Datenzentrum sehr wohl bedeutend. Diese wird von ISPs entsprechend mit der Serverhardware im Angebot angeführt.

Für optimale Leistung sollte der Server mindestens einen dedizierten 100 MB Down- und Upstream vorweisen können. Auch hier gilt: umso mehr, umso besser. Vorsicht sollte geboten sein wenn die Anbindung nicht dediziert ist. Durch die Verteilung einer 100 MB-Anbindung auf z. B. 20 oder mehr virtuelle Hosts⁶ (Virtual Private Server, bzw. abgekürzt VPS) kann die Performance zeitweise stark einbrechen. Dies macht sich für die Benutzer dann besonders beim Download größerer Dateien bemerkbar. Größere Dateien sind im Web bereits Dateien mit mehr als 1 MB. Bricht die Übertragungsrate auf z. B. 10 KB/s ein dauert der Download einer 1 MB großen Datei bereits über 13 Minuten. Ein durchschnittlicher Breitbandanschluss in Österreich bietet eine Downloadgeschwindigkeit von effektiven 523 KB/s (vgl. Mey, 2011). Insofern die gesamte Datenübertragungsrate⁷ genutzt werden kann, verringert sich die Downloadzeit auf gerade mal 16 Sekunden für dieselbe 1 MB-Datei (vgl. Anon., 2012).

Während bei Root-Servern⁸ in der Regel die Hardware mit Hersteller und der genauen Modellbezeichnung angeführt sind, fehlen diese Informationen bei VPS-Angeboten. Dies hängt mit der Virtualisierung der Hardware zusammen. Bei einem VPS wird nicht die reale Hardware angesprochen, sondern nur ein virtualisierter Teil des echten Servers. Entsprechend spielen die Merkmale der echten Serverkonfiguration nur noch eine untergeordnete Rolle. Es empfiehlt sich aber gerade hier mit dem

⁶ Ein virtueller Host ist ein Server der sich für den Endanwender wie ein physikalischer Server verhält, jedoch nur in einer virtuellen Umgebung auf einem physikalischen Rechner simuliert wird. Das Gegenstück – also ein physikalischer Server – wäre der Root-Server. Bei den sogenannten VPS (Virtual Private Server) teilen sich mehrere dieser virtualisierten Server einen physikalischen Server und somit dessen Hardware-Ressourcen.

⁷ Oft auch fälschlicherweise als Bandbreite bezeichnet, der Grund dafür ist im Englischen zu finden. Da das Wort Bandwidth sowohl für die Bandbreite (*analog bandwidth*) als auch für die Datenübertragungsrate (*digital bandwidth*) verwendet wird.

⁸ Eigentlich handelt es sich bei einem Root-Server um die Root-Nameserver (siehe auch <http://www.root-servers.org/>), umgangssprachlich werden damit jedoch Rechner mit Superuseranmeldung bezeichnet. Die Bezeichnung ist in mehrerlei Hinsicht somit verwirrend, da die bereits erwähnten VPS auch eine Anmeldung als Superuser ermöglichen. Der Unterschied zum VPS ist die reale, physikalisch vorhandene Hardware auf die der Administrator Zugriff hat. Im Kontrast dazu bietet der VPS lediglich virtualisierte Hardware die eigentlich nicht existiert.

Anbieter erst in Kontakt zu treten und die genauen Details der Produktpalette zu hinterfragen, insofern die Angaben nicht ausreichend detailliert sind. Sollte ein ISP hier keine detaillierten Antworten liefern können oder wollen, ist das bereits ein Grund zu einem anderen zu wechseln. Hier sollte direkt auch abgeklärt werden mit wie vielen anderen Benutzern der Server geteilt wird und welche Virtualisierungssoftware zum Einsatz kommt. Anhand dieser Daten lässt sich vorab klären ob das System den gewünschten Anforderungen entspricht und auch Unterstützung für spezielle Software – insofern benötigt – mit sich bringt.

CPU, RAM und HDD sollten zum Zeitpunkt der Anmietung aktuell sein und für den Dauereinsatz geeignet sein. Mehrere physikalische Kerne und/oder Hyper-Threading können die Leistung bestimmter Software enorm steigern. Eine Error-Correction-Code-Erweiterung (ECC) beim RAM kann hingegen zur Stabilität eines Servers im Dauerbetrieb beitragen. Zu guter Letzt sind Redundant-Array-of-Independent-Disks-Verbünde (RAID) von schnellen Serial-Advanced-Technology-Attachment- (SATA) oder Serial-Attached-Small-Computer-System-Interface-Festplatten (SAS) für eine schnelle Zugriffszeit auf Dateien sehr gut. Eine weitere Steigerung der Festplattengeschwindigkeit ist z. B. durch die Verwendung von Solid-State-Drives (SSDs) möglich, weiteres Steigerungspotential ergibt sich auch hier wieder durch RAID-Verbindung mehrerer SSDs. Es sollte im Vorfeld mit dem Anbieter abgeklärt werden welche Angebote mit welchen Laufwerken ausgestattet sind. SSDs sind mittlerweile nicht mehr zwangsläufig schneller als z. B. ein SAS-RAID-Verbund.

2.4 Betriebssystem

Die Wahl des Betriebssystems (OS vom Englischen Operating System) ist der erste wichtige Schritt hin zu einem performanten System. Software die ursprünglich für Unix entwickelt wurde bietet unter einem Microsoft Betriebssystem nur einen Bruchteil der Leistung an. Hingegen ist Microsoft spezifische Software oftmals gar nicht für Unix-Systeme verfügbar. Dies hängt mit der internen Architektur der Betriebssysteme zusammen. Während Unix auf eine Multiprozessarchitektur setzt, greift Windows auf Multithreading zurück (vgl. Microsoft, 2006). Unix kommt laut

W³Tech⁹ auf 63,6 % aller Websites zum Einsatz, 36,4 % fallen auf Microsoft Windows und nur 0,1 % greifen zu Apples Mac OS (vgl. Q-Success, 2012).

Während Debian¹⁰ für viele Jahre das bevorzugte Linux Derivat unter Webmasters war, änderte sich das im Juni 2010 plötzlich und CentOS¹¹ wurde die Nummer Eins. Seit Januar 2012 hat sich dies wieder zu Gunsten von Debian geändert (vgl. Noyes, 2012). Mit einer Nutzungsverbreitung von etwa 31 % im August 2012 bleibt Debian somit das beliebteste Betriebssystem für Websites (vgl. Q-Success, 2012).

Für die Performance ist die Auswahl des OS nicht besonders ausschlaggebend, da sich generell alle Betriebssysteme anpassen lassen. Manche bieten in der Grundkonfiguration erhöhte Sicherheit, aber auch diese lässt sich bei jedem durch entsprechende Konfiguration implementieren. In dieser Arbeit wird Debian verwendet, das aufgrund seiner Verbreitung am geeignetsten erschien.

2.5 Software

Die Wahl der richtigen Software zum Betrieb einer Website ist ausschlaggebend für die Performance, den Ressourcenverbrauch und die Sicherheit der endgültigen Applikation. Jedes Programm bietet verschiedene Vor- und Nachteile, egal ob Programmiersprache, Datenbanksystem oder Webserver. Die Auswahl der richtigen Software kann dabei auf verschiedenen Kriterien beruhen. Ausgehend von den Vorlieben oder den Kenntnissen eines Administrators / einer Administratorin oder der Angestellten, möglicherweise gibt es aber auch Vorgaben vom Kunden / der Kundin, oder andere Restriktionen. Die Auswahl für die Testumgebung fußt erneut auf den Verbreitungsdaten von W³Tech. Dadurch soll die Relevanz der Konfiguration, die im Folgenden Erarbeitet wird, steigen.

⁹ Die Daten von W³Techs (<http://w3techs.com/>) basieren auf Befragungen und Überprüfungen der 1.000.000 größten Websites laut Alexa (<http://www.alexa.com/>) und sind nicht zwingend repräsentativ. Werden jedoch allgemein in Foren und Communities meist so bestätigt.

¹⁰ <http://www.debian.org/> – Debian Website

¹¹ <https://www.centos.org/> – CentOS Website

2.5.1 Auswahlkriterien

Zuerst ist es wichtig darauf hinzuweisen, dass eine unvoreingenommene Grundeinstellung wichtig ist um objektiv bewerten zu können. Dies ist meist der Knackpunkt bei der Festlegung von Auswahlkriterien die sich im Internet finden lassen, die Autoren lassen zu oft ihre eigenen Vorlieben einfließen. Wie bereits erwähnt bietet jede Software Vor- und Nachteile, aber auch diese ändern sich von Version zu Version teilweise stark und dies sollte somit nicht außer Acht gelassen werden.

- **Programmiersprache:**

Die Auswahl der Programmiersprache sollte in erster Linie auf Effizienz beruhen. Egal ob nun ein Framework¹² oder eine fertige Software zum Einsatz kommt, es werden Anpassungen gewünscht an die eigene Anwendungsdomäne gewünscht sein. Damit diese Anpassungen effizient durchgeführt werden können, sollte eine Programmiersprache gewählt werden die von den eigenen Entwicklern beherrscht wird und die für die Anwendung Sinn macht. So wäre z. B. Assembler oder C als Programmiersprache für eine Website sehr ungeeignet aufgrund ihrer hardwarenahen Natur. Höhere Programmiersprachen wie Java, Ruby oder PHP sind hier zielführender. Die getätigte Auswahl schränkt die Auswahl an Frameworks, Content Management Systemen oder fertiger Software (Blog-, Foren-, Wiki-Software) ein. Es kann jedoch auch anders herum sein, z. B. eine Vorgabe, dass Software XY verwendet werden muss. Dies würde dann natürlich die Auswahl der Programmiersprache einschränken, oft beschränkt sich die Wahl dann ausschließlich auf die Version der Programmiersprache.

- **Datenbanksystem:**

Welches Datenbanksystem (DBS) eingesetzt werden kann, wird bereits durch die verwendete Programmiersprache eingeschränkt. Nicht jede Sprache bietet die Möglichkeit jede Datenbank anzusprechen, insofern nicht selbst eine sol-

¹² Ein Framework ist ein Grundgerüst das als Basis für eigene Software dient. Dabei stellt das Framework bestimmte Funktionen bereit oder gibt gar vor wie eine Applikation zu programmieren ist. Was genau das Framework macht und bietet ist auch abhängig vom Framework-Typ, wobei diese Unterteilung meist auch lose ist. Bekannte Beispiele für Frameworks sind z. B. .NET von Microsoft, Zend Framework für PHP, JavaServer Faces oder JUnit für Java und Ruby on Rails für Ruby.

che Anbindung programmiert werden will, ist dies als klare Restriktion zu sehen. Bei der Wahl des DBS sollten Geschwindigkeit und Ressourcenverbrauch im Vordergrund der Auswahl stehen. Auch sollte eine Einschätzung der eigenen Applikation durchgeführt werden: wird sie Lese- oder Schreibintensiv, oder gar beides. Weitere Kriterien umfassen die Art und Weise des Umgangs mit den Daten. Dabei müssen diverse Fragen an die eigene Einsatzweise gestellt werden: ist ein relationales System das Richtige, zeilen- oder spaltenorientiert, einfache Schlüssel-Wert-Struktur, spielt Konsistenz eine Rolle. Dies sind nur wenige von möglichen Fragen die es zu beantworten gilt.

- **Webserver:**

Hier muss geprüft werden ob der zu verwendende Webserver die gewünschten Funktionen bereitstellt. Die Auswahl wird anhand der Programmiersprache wieder eingeschränkt, denn nicht jeder Webserver unterstützt jede Programmiersprache. Wer z. B. auf eine Microsoft Programmiersprache setzt ist oftmals auf einen Microsoft Webserver beschränkt, ähnlich verhält es sich bei Java. Die Performance variiert sehr stark je nach verwendeter Plattform, wie bereits erwähnt im Kapitel 2.4.

2.5.2 Programmiersprache

Die Wahl der Programmiersprache ist ähnlich wie die Wahl des Betriebssystems eine Glaubensfrage. Im Rahmen dieser Arbeit wird deshalb auf die populärste Web-Programmiersprache zurückgegriffen. Dadurch gewinnen die Testresultate und der Nutzen der Konfiguration mehr an Relevanz für alle Leser und Leserinnen.

Laut den Umfragen von W³Techs ist PHP^{13,14} auch heute noch die am häufigsten eingesetzte serverseitige Programmiersprache mit über 77,8 %. Gefolgt von ASP.NET¹⁵ mit 21,4 % und Java¹⁶ mit lediglich 4 % (vgl. Q-Success, 2012). Die

¹³ <http://www.php.net/> – PHP Website

¹⁴ PHP ist ein rekursives Akronym und steht für „PHP: Hypertext Preprocessor“

¹⁵ <http://www.asp.net/> – ASP.NET Website

¹⁶ <http://www.java.com/> – Java Website

Ergebnisse von Google Trends zeigen einen anderen Trend auf, beziehen sich jedoch auf Suchanfragen und spiegeln entsprechend nicht unbedingt die tatsächliche Verwendung wieder (vgl. Google, 2012).

Professionellen PHP-Support für Unternehmen gibt es von diversen anderen Firmen rund um den Globus. Am bekanntesten ist mit Sicherheit einer der größten PHP-Sponsoren überhaupt, Zend Technologies¹⁷. Zend ist auch im deutschsprachigen Raum tätig und unterstützt namhafte Großfirmen wie z. B. die Kabeldeutschland GmbH (vgl. Zend, 2012).

2.5.3 Content Management System

Die Festlegung auf PHP als Programmiersprache erlaubt es ein passendes Framework bzw. Content Management System (CMS) für eine Web-Applikation zu wählen. Ein Framework würde jedoch bedeuten, dass lediglich ein Grundgerüst zur Verfügung steht. Aufbauend auf diesem kann in weiterer Folge eine Website erstellt werden. Für die Tests ist dies jedoch kontraproduktiv da die Entwicklung einer eigenen Anwendung die Benchmark-Resultate verfälschen und schwer vergleichbar machen würde, deshalb werden lediglich CMSs in Betracht gezogen.

Erneut bietet W³Techs genaue Statistiken zu diesem Thema. Laut deren Erhebungen wird WordPress¹⁸ von 16,7 % aller Websites als CMS eingesetzt. Joomla¹⁹ befindet sich auf Platz zwei mit 2,8 %, gefolgt von Drupal²⁰ mit 2,1 % (vgl. Q-Success, 2012) (vgl. Google, 2012). In Deutschland, Österreich und der Schweiz sieht die Verteilung anders aus, da hier besonders bei Unternehmen das deutsche CMS TYPO3²¹ oft eingesetzt wird (vgl. Google, 2012).

Das Ziel der fiktiven Testwebsite ist eine solide Grundlage für Unternehmen zu bieten. Drupal bietet hierbei die größte Flexibilität für Entwickler an, die höchste Si-

¹⁷ <http://www.zend.com/> – Zend Website

¹⁸ <http://wordpress.org/> – WordPress Website

¹⁹ <http://www.joomla.org/> – Joomla Website

²⁰ <http://drupal.org/> – Drupal Website

²¹ <http://typo3.org/> – TYPO3 Website

cherheit und die Möglichkeit auch für wirklich riesengroße Projekte zu skalieren (vgl. Uzayr, 2012).

Es gibt international diverse Unternehmen, die professionellen Drupal-Support und -Entwicklung von Websites anbieten. Am bekanntesten dürfte Aquia²² sein, wurde das Unternehmen doch vom Drupal-Initiator Dries Buytaert selbst gegründet (vgl. Acquia, 2012). Im deutschsprachigen Raum tritt besonders Amaze Labs²³ aus Zürich in der Schweiz hervor. Das Engagement des Unternehmens innerhalb der Drupal-Community ist sehr umfangreich.

2.5.4 Datenbanksystem

Die Wahl des richtigen Datenbanksystems (DBS) hat einen extrem großen Einfluss auf die finale Geschwindigkeit einer Webanwendung. Zum gegenwärtigen Zeitpunkt werden besonders NoSQL-Lösungen diskutiert (vgl. Google, 2012). Trotzdem ist für die Abbildung klassischer Daten – wie die einer Website – ein relationales Datenbanksystem (RDBS) immer noch am besten geeignet (vgl. Walker-Morgan, 2010).

In der Welt der kostenlosen DBS ist auch heute noch MySQL²⁴ das Maß aller Dinge. Über die Jahre verlor die Software zwar immer mehr an Anhängern, dies hängt jedoch stark mit der wachsenden Anzahl an Alternativen zu klassischen RDBS zusammen. Andere Vertreter wie PostgreSQL²⁵ können gleichzeitig auch keine höhere Verbreitung aufweisen (vgl. Google, 2012). Von MySQL gibt es zwei wichtige Forks, die auch bessere Performance versprechen.

2.5.4.1 Percona Server

Percona Server²⁶ wird von der Percona, Inc. seit mehreren Jahren entwickelt und setzt dabei auf die eigene XtraDB Storageengine, welche wiederum ein Fork von

²² <http://www.acquia.com/> – Acquia Website

²³ <http://www.amazeelabs.com/> – Amaze Labs Website

²⁴ <http://www.mysql.com/> – MySQL Website

²⁵ <http://www.postgresql.org/> – PostgreSQL Website

²⁶ <http://www.percona.com/software/percona-server> – Percona Server Website

InnoDB²⁷ ist. In unzähligen Benchmarks im WWW wurde bereits nachgewiesen, dass Percona Server in Kombination mit XtraDB eine bessere Performance als MySQL bietet und zugleich auch konsistenter arbeitet.

Das von Peter Zaitsev gegründete Unternehmen Percona, Inc. bietet auch professionelle Unterstützung für Unternehmen an, die MySQL, Percona Server oder andere Forks von MySQL einsetzen (vgl. Percona, 2012).

2.5.4.2 MariaDB

Als 2008/09 Oracle Sun aufkaufte gab es viele Diskussionen um die Zukunft der freien Datenbank MySQL. Vom Gründervater Michael „Monty“ Widenius hagelte es auch bereits früh harsche Kritik an der Oracle Politik (vgl. Kirsch, 2008). Im Februar 2009 verließ Widenius Oracle und gründete sein eigenes Unternehmen: Monty Program AB. Gleichzeitig mit der Gründung des Unternehmens kündigte er auch an, dass an einem neuen MySQL-Fork namens MariaDB²⁸ gearbeitet wird. Die Ziele sind klar: kompatibel zu MySQL, bessere Codebasis, erhöhte Performance und freie Software (vgl. Diedrich, 2009).

Die MariaDB-Entwickler arbeiten eng mit den Percona-Entwicklern zusammen. So gibt es auch professionellen Support für MariaDB-Installationen bei Percona zu beziehen. Des Weiteren führt diese Zusammenarbeit dazu, dass MariaDB – wie Percona Server – den InnoDB-Fork XtraDB als Standard-Storageengine verwendet. Ungleich Percona Server bietet MariaDB jedoch die Möglichkeit auf InnoDB umzusteigen. Dadurch bietet das quelloffene DBS die höchste Flexibilität aller MySQL-Versionen.

Als weiteres großes Unternehmen, das kommerziellen Support für MariaDB und andere MySQL-Derivate anbietet, ist SkySQL²⁹ zu nennen. SkySQL hat auch Nie-

²⁷ <http://www.innodb.com/> – InnoDB Website

²⁸ <http://mariadb.org/> – MariaDB Website

²⁹ <http://www.skysql.com/> – SkySQL Website

derlassungen in Deutschland und bietet deutsche Unterstützung an. Weitere Serviceanbieter können der offiziellen MariaDB Website entnommen werden³⁰.

2.5.5 Webserver

Der Begriff Webserver wird im Zusammenhang mit Websites oft in mehrfacher Bedeutung verwendet. Eigentlich handelt es sich hierbei jedoch um die Software die auf einem Server installiert wird, um die Beantwortung von Hypertext-Transfer-Protocol-Anfragen (HTTP) abzuhandeln (und diese ggf. an ein Backend zur Generierung von Inhalten weiterzuleiten).

2.5.5.1 Apache HTTP Server

Der Apache HTTP Server³¹ (kurz httpd nach der ausführbaren Datei oder historisch Apache (vgl. The Apache Software Foundation, 2012)) ist auch heute noch die bekannteste und am meisten verwendete Webserver-Software (vgl. Netcraft Ltd., 2012). Bei der Entwicklung des Apaches stand jedoch nie im Vordergrund den schnellsten Webserver zu programmieren. Anstatt eine einzige Architektur zu implementieren, bietet er gleich verschiedene Modelle zur Konfiguration an. Die Multi-Processing Modules (MPMs) erlauben es so die Webserver Modi (Event, Worker und Prefork) zu wechseln. Dies soll auf lange Sicht die Latenz und den Durchsatz konsistent und zuverlässig auf gleichem Niveau halten (vgl. Jagielski, 2011).

In unzähligen Benchmarks wurde jedoch immer wieder aufgezeigt, dass der Apache mit anderen Webservern nicht mithalten kann. Besonders die Standardkonfiguration ist verglichen mit anderer Software extrem langsam und ressourcenhungrig. Darüber sind sich jedoch auch die Apache Entwickler im Klaren und arbeiten deshalb an der Optimierung der Performance (vgl. ebd.).

Aufgrund der enormen Verbreitung des Apaches ist es auch einfach, professionellen Support für die Konfiguration zu erhalten. Am bekanntesten dürfte hier

³⁰ <http://mariadb.org/en/service-providers/> – Support Serviceanbieter für MariaDB

³¹ <http://httpd.apache.org/> – Apache HTTP Server Website

SpringSource³² sein, ein Tochterunternehmen von VMware³³, die auch Hilfe für deutschsprachige Kunden anbieten (vgl. SpringSource, 2012).

2.5.5.2 nginx

Nginx ist die am zweithäufigsten eingesetzte Webserver-Software unter Unix basierten Systemen mit einer Verbreitung von fast 12,5 % unter allen aktiven Websites³⁴ (vgl. Netcraft Ltd., 2012). Anders als der Apache mit den MPMs setzt nginx auf eine asynchrone, eventbasierte Abhandlung von Anfragen und wurde von Anfang an als hoch nebenläufige, performante und speicherschonende Software entwickelt.

Professionellen Support für nginx gibt es seit Juli 2011 durch die Nginx, Inc.³⁵. Das Unternehmen wurde von Igor Sysoev, dem Autor von nginx selbst, gegründet (vgl. Nginx, 2012).

2.5.5.3 Weitere Alternativen

2.5.5.3a lighttpd

Lighttpd³⁶ (auch *Lighty* genannt) funktioniert ähnlich wie nginx, findet jedoch weniger Verbreitung und professionellen Support gibt es kaum bis gar nicht. Trotzdem stellt Lighty eine gute Wahl bei der Umsetzung von hoch performanten Websites dar.

2.5.5.3b gatling

Gatling³⁷ ist ein extrem kleiner und laut eigener Aussage performanter Webserver, bisher jedoch noch völlig unbekannt. Es gibt jedoch vom Unternehmen Code Blau³⁸ das vom gatling Autor Felix von Leitner gegründet wurde professionellen deutschsprachigen Support für Unternehmen.

³² <http://www.springsource.com/> – SpringSource Website

³³ <http://www.vmware.com/> – VMware Website

³⁴ <http://news.netcraft.com/active-sites/> – Netcraft Definition von „active sites“

³⁵ <http://nginx.com/> – Nginx, Inc. Website

³⁶ <http://www.lighttpd.net/> – Lighttpd Website

³⁷ <http://www.fefe.de/gatling/> – gatling Website

³⁸ <http://www.codeblau.de/> – Code Blau Website

2.5.5.3c G-WAN

G-WAN³⁹ hat sich in vielen Benchmarks die im WWW publiziert wurden als der mit Abstand schnellste Webserver überhaupt hervor getan. Der Geschwindigkeitsvorteil gegenüber nginx / lighttpd liegt dabei in der Regel beim Faktor zwei. G-WAN hat jedoch ein Problem, es setzt auf C als Programmiersprache für Websites. Wie die zuvor vorgestellten Statistiken zeigen und jeder Web-Programmierer selbst weiß, ist C sehr schlecht für die Programmierung von derart abstrakter Software geeignet.

Mittlerweile unterstützt G-WAN auch Java, dies ist natürlich weitaus interessanter für das Web. Eine PHP-Unterstützung ist geplant für die Zukunft. Voraussetzung dafür ist jedoch, dass es von PHP eine threadsichere Binärdatei gibt. Dies ist bis heute nicht der Fall und dürfte wahrscheinlich noch etwas Zeit in Anspruch nehmen. Ein Benutzer hat im G-WAN-Forum im August 2011 angekündigt, an einer Erweiterung zu arbeiten, die es erlauben soll via FastCGI⁴⁰ auf PHP als Backend zugreifen zu können. Seit Oktober 2011 gab es jedoch nichts Neues zu diesem Thema (vgl. Diverse, 2011).

Professionellen Support für G-WAN gibt es vom TrustLeap Unternehmen des G-WAN-Autors Pierre Gauthier selbst in verschiedenen Preisklassen. Der Autor selbst ist leider nicht wirklich unumstritten aufgrund seiner Persönlichkeit und seinem Umgang mit Kritik (vgl. Kliś, 2011). Insofern sollte ein Einsatz von G-WAN gut durchdacht sein, da die Entwicklung jederzeit eingestellt werden könnte und der Quelltext nicht offen zur Verfügung steht.

2.5.5.3d Hiawatha Webserver

Der Hiawatha Webserver⁴¹ ist der letzte der an dieser Stelle vorgestellt wird. Die von Hugo Leisink⁴² in C entwickelte Software ist auch ein leichtgewichtiger Server, das Hauptaugenmerk liegt hier jedoch nicht auf Performance – was nicht bedeuten soll,

³⁹ <http://gwan.com/> – G-WAN Website

⁴⁰ <http://www.fastcgi.com/> – FastCGI Website mit weiterführender Dokumentation zur Technologie.

⁴¹ <http://www.hiawatha-webserver.org/> – Hiawatha Webserver Website

⁴² Hugo Leisink bietet neben dem Hiawatha Webserver auch das Hochsicherheits-PHP-Framework Banshee PHP (<http://www.banshee-php.org/>) an.

dass Hiawatha langsam ist. Leisinks Beweggrund die Entwicklung vom Hiawatha Webserver anzugehen und der Hauptfokus liegen auf Sicherheit. Seiner eigenen Aussage nach boten andere Webserver zu der Zeit als er mit der Entwicklung seiner Software begann (das war 2002) mehr oder weniger keine Sicherheit und die Funktionen die es gab waren sehr komplex zu konfigurieren. Heute liegt der Fokus immer noch auf Sicherheit, aber auch auf einer einfachen Anwendung und Konfiguration. Der Funktionsumfang ist deshalb bewusst übersichtlich gehalten und der Webserver macht was ein Webserver machen soll, Webseiten ausliefern.

2.5.5.4 Vergleich

Webserver-Vergleich						
	Apache HTTP Server	nginx	lighttpd	gatlign	G-WAN	Hiawatha Webserver
Entwicklet durch	Apache Software Foundation	NGINX, Inc.	Jan Kneschke Beitragende ⁴³	Felix von Leitner	Pierre Gauthier	Hugo Leisink
Lizenz	Apache License 2.0 ⁴⁴	BSD Variante ⁴⁵	BSD ⁴⁶	GPLv2 ⁴⁷	Freeware ⁴⁸	GPLv2
Betriebssystem	Windows Linux OS X uvm.	Windows Linux OS X uvm.	Windows Linux OS X uvm.	Linux	Windows Linux	Windows Linux OS X
Sicherheit						
Basic Access Authentication ⁴⁹	+	+	+	k. A.	k. A.	+
Digest Access Authentication	+	+ ⁵⁰	+	k. A.	k. A.	k. A.
SSL/TLS (HTTPS)	+	+	+	+	- ⁵¹	+

⁴³ <http://redmine.lighttpd.net/projects/lighttpd/wiki/DevelopersList> – Liste an beitragenden Entwicklern am lighttpd Projekt.

⁴⁴ <http://www.apache.org/licenses/> – Die Apache Lizenz

⁴⁵ <http://www.nginx.org/LICENSE> – Nginx Lizenz

⁴⁶ <http://www.lininfo.org/bsdlicense.html> – BSD Lizenzinformation vom The Linux Information Project (LINFO)

⁴⁷ <http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt> – GNU General Public License Version 2.0

⁴⁸ Freeware bedeutet, dass es sich um proprietäre Software handelt bei der der Quelltext nicht offenliegt. Die Verwendung der Software ist jedoch kostenlos. Im Fall von G-WAN ist die private und kommerzielle Verwendung erlaubt.

⁴⁹ <http://tools.ietf.org/html/rfc2617> – Im RFC 2617 sind sowohl die Basic Access Authentication als auch die Digest Access Authentication spezifiziert.

⁵⁰ <https://github.com/samizdatco/nginx-http-auth-digest> – Digest Access Authentication nur via Drittherstellermodul möglich.

SSL/TLS (SPDY) ⁵²	+ ⁵³	+ ⁵⁴	- ⁵⁵	-	-	-
Dynamische Inhalte						
CGI	+	-	+	+	-	+
FastCGI	+	+	+	+	-	+
SimpleCGI	+	+	+	+	-	+
Java Servlets	-	-	-	-	+	-
Server Side Includes (SSI)	+	+	+	-	-	-
Internet Server Application Programming Interface (ISAPI)	+ ⁵⁶	-	-	-	-	-
Sonstiges						
Virtuelle Hosts	+	+	+	+	+	+
IPv6	+	+	+	+	-	+
HTTP-Kompression	+	+	+	+	+	± ⁵⁷
Reverse Proxy	+	+	+	-	+	+
Bewertung						
Funktionsumfang	[++]	[++]	[+]	[±]	[--]	[+]
Konfigurationsaufwand	[+]	[++]	[++]	[-]	[-]	[++]
Support	[++]	[+]	[+]	[-]	[-]	[+]
Performance	[--]	[+]	[+]	[+]	[++]	[+]
[++] sehr gut [+] gut [±] zufriedenstellend [-] schlecht [--] sehr schlecht + vorhanden - nicht vorhanden k. A. keine Angabe						

⁵¹ Laut Pierre Gauthier existiert eine Version mit SSL/TLS-Unterstützung, jedoch sieht er keinen Vorteil in der Verwendung der Technik und gibt diese deshalb nicht frei (vgl. tag, et al., 2011).

⁵² SPDY ist der HTTP 1.1-Nachfolger von Google (vgl. Google, Inc., 2012)

⁵³ SPDY-Unterstützung via mod_spdy und nur für Apache HTTP Server Version 2.2 (vgl. Google, Inc., 2012)

⁵⁴ SPDY-Unterstützung ist als Quellpatch verfügbar (vgl. NGINX, Inc., 2012)

⁵⁵ SPDY-Unterstützung für lighttpd 1.x ist nicht geplant, möglicherweise in Version 2.x (vgl. waza123a & spaam, 2012).

⁵⁶ ISAPI-Unterstützung via mod_isapi und nur unter Microsoft Windows Betriebssystemen (vgl. Apache Software Foundation., 2012).

⁵⁷ Der Hiawatha Webserver kann nicht zur Laufzeit selbst komprimieren, es besteht jedoch die Möglichkeit bereits komprimierte Dateien auszuliefern insofern die Anfrage dies anfordert. In der FAQ wird empfohlen Skriptaussagen mit der verwendeten Programmiersprache zu komprimieren (vgl. Leisink, 2012).

2.6 Testumgebung

Um eine realistische Testumgebung gewährleisten zu können, kommt ein VPS des österreichischen Unternehmens EDIS GmbH aus Graz⁵⁸ zum Einsatz. Der VPS wurde vom Budget passend zu dem einer Privatperson bzw. eines Startups gewählt. Die Hardwareeckdaten sehen aus wie folgt:

KVM ADVANCED plus	
Virtualisierung:	Kernel-based Virtual Machine ⁵⁹
Hauptprozessor:	Intel basiert (2 Kerne) mit Intel VT Instruktionen
Arbeitsspeicher:	1,536 MB RAM
Festplatte:	22 GB SAS-RAID
Netzwerk:	Geteilte GB-Anbindung
Standort:	Graz, Österreich
Kosten:	19,99 € / Monat

Die EDIS GmbH war so nett und hat für die Test einen weiteren VPS zur Verfügung gestellt, die Eckdaten des zweiten Servers sehen aus wie folgt:

VRS ADVANCED	
Virtualisierung:	Linux vServer ⁶⁰
Hauptprozessor:	Intel basiert mit Intel VT Instruktionen
Arbeitsspeicher:	1,536 MB RAM
Festplatte:	44 GB SAS-RAID
Netzwerk:	Geteilte GB-Anbindung
Standort:	Graz, Österreich
Kosten:	10,99 € / Monat

⁵⁸ <https://www.edis.at/> – EDIS GmbH Website

⁵⁹ <http://www.linux-kvm.org/> – Linux KVM Website

⁶⁰ <http://linux-vserver.org/> – Linux vServer Website

Da der VRS auf Linux vServer als Virtualisierung setzt ist er für die Konfiguration zu einem High-Performance Server ungeeignet. Zu wenige Parameter des Betriebssystems lassen sich beeinflussen. In den Tests wird deshalb ausschließlich der KVM VPS eingesetzt.

2.7 Finaler Testaufbau

Abschließend ein Kurzüberblick über die finale Konfiguration. Die Auswahl erfolgte nach dem statistisch höchsten Verbreitungsgrad und in weiterer Folge nach dem Gesichtspunkt der Performance und Konfigurierbarkeit bzw. Erweiterbarkeit. So hat Drupal zwar nicht den höchsten Verbreitungsgrad, bietet jedoch aus Sicht der Skalierbarkeit und Erweiterbarkeit gegenüber der Konkurrenz einen enormen Vorteil (siehe Kapitel 2.5.3).

Finale Testumgebung	
Server:	EDIS KVM ADVANCED plus
Betriebssystem:	Debian GNU/Linux 6 „Squeeze“ x64
Programmiersprache:	PHP: Hypertext Preprocessor 5.4.5
Content Management System:	Drupal 7.15
Datenbanksystem:	MariaDB 5.5.25
Webserver:	nginx 1.3.4

Eine alternative Konfiguration die sich stärker an aktuellen Trends und stabiler Software orientiert könnte wie folgt aussehen.

Alternative Testumgebung	
Server:	Webtropy Cloud vServer Linux M ⁶¹
Betriebssystem:	Ubuntu 12.04 „Precise Pangolin“ x64
Programmiersprache:	PHP: Hypertext Preprocessor 5.3.10

⁶¹ <http://www.webtropy.com/de/vservercloud/vserver-cloud-linux-m.html> – Angebotsseite

Content Management System:	WordPress 3.4 „Green“
Datenbanksystem:	MySQL 5.5.24
Webserver:	nginx 1.1.19

Mit „stabiler Software“ ist hier gemeint, dass immer die aktuellen Pakete aus dem stabilen Betriebssystemrepositorium verwendet werden (mehr dazu im Kapitel 3.3).

3. Konfiguration

Die gesamte Konfiguration basiert auf der Annahme, dass der VPS neu vom Anbieter erhalten wurde. Was bei dem Testserver der EDIS GmbH auch der Fall ist. Alle Arbeiten werden von einem Microsoft Windows 7 basierten Computer ausgeführt.

3.1 Systeminbetriebnahme

Nach Erhalt der Zugangsdaten zum Server kann zum ersten Mal eine Verbindung zum Server hergestellt werden. Um sich auf einem Windows System via Secure Shell (SSH)⁶² auf einen entfernten Rechner zu verbinden, wird eine Software benötigt. Das ausgewählte Programm PuTTY⁶³ ist frei erhältlich und gilt als der Quasi-Standard für Telnet- und SSH-Programme unter Windows. Für die erste Verbindung zum Testserver ist es ausreichend die IP-Adresse in das dafür vorgesehene Eingabefeld zu kopieren.

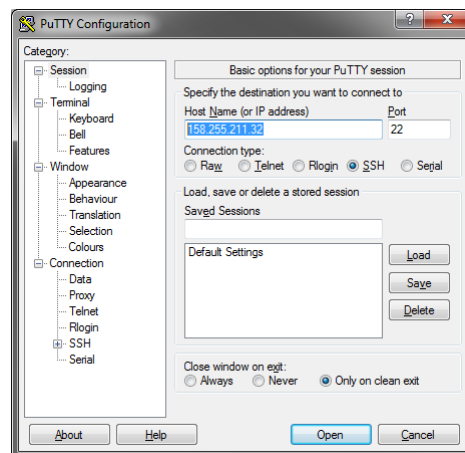


Abbildung 1 – PuTTY-Hauptfenster

Da die Verbindung zum ersten Mal aufgebaut wird, ist der Rechner mit dem eine Verbindung aufgebaut werden soll dem verbindenden System (PuTTY) noch unbekannt. PuTTY wird deshalb eine Sicherheitswarnung anzeigen, diese kann – insofern die IP-Adresse korrekt eingegeben wurde – mit einem Klick auf „Yes“ bestätigt werden. Sollte diese Meldung später nochmals erscheinen kann das bedeuten, dass sich

⁶² <http://www.ietf.org/rfc/rfc4251> – RFC 4251: The Secure Shell (SSH) Protocol Architecture

⁶³ <http://www.chiark.greenend.org.uk/~sgtatham/putty/> – PuTTY Website

unter derselben IP ein anderer Server meldet, oder dass jemand eingebrochen ist und den Schlüssel geändert hat. Der Host-RSA-Schlüssel befindet sich unter `/etc/ssh/ssh_host_rsa_key`, dieser kann auch im Vorfeld vom Anbieter verlangt werden. Dies ermöglicht es bei der ersten Verbindung den RSA2-Schlüsselfingerabdruck zu verifizieren bevor auf „Yes“ geklickt wird.

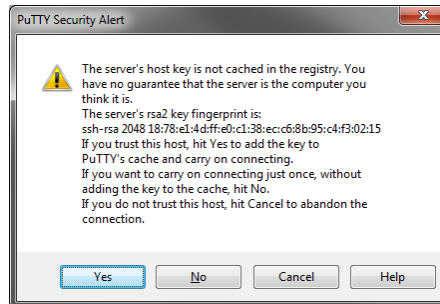


Abbildung 2 – PuTTY-Sicherheitswarnung

Nach erfolgreichem verbinden sollte die Anmeldung erscheinen. Da es sich um einen neuen VPS und eine saubere Debian-Installation handelt, existiert zu diesem Zeitpunkt ausschließlich ein Benutzer: root. Das Passwort für den root-Benutzer sollte vom Anbieter mit den Zugangsdaten zum Server übermittelt worden sein.

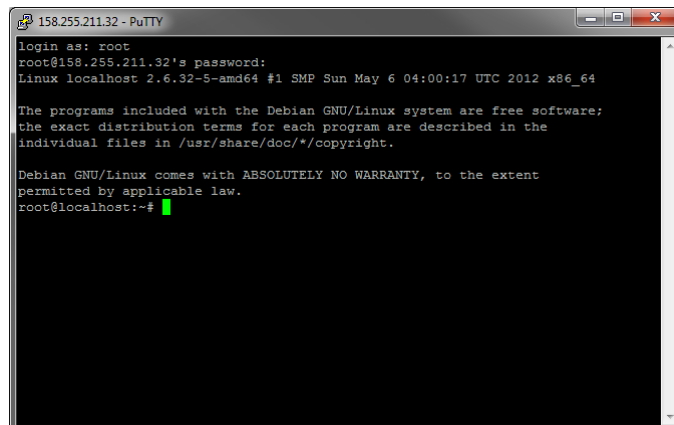


Abbildung 3 – dash^{64,65,66} nach erfolgreicher Authentifizierung

Nun können noch bestimmte Systemvariablen angepasst werden. Wurde z. B. die Domain nicht zusammen mit dem Server angemietet, oder möglicherweise ist der

⁶⁴ Die Abkürzung dash steht für „Debian Almquist shell“.

⁶⁵ <http://gondor.apana.org.au/~herbert/dash/> – dash Website

⁶⁶ <http://wiki.debian.org/Shell> – Shell Artikel im Debian Wiki

Server trotzdem falsch konfiguriert, sollte nun der Hostname und die Namensauflösung des Servers angepasst werden.

```
# hostname -v kvm.edis.bobdo.net
# echo '# /etc/hosts

127.0.0.1        localhost
158.255.211.32   kvm.edis.bobdo.net kvm localhost' > /etc/hosts
```

Quelltext 1 – Konfiguration von Hostnamen und Namensauflösung

3.2 OpenSSH Daemon

Nach erfolgreicher Verbindung müssen die ersten Sicherheitslücken beseitigt werden, die Authentifizierung selbst. Es ist nicht ratsam sich als root am System anzumelden, da zu diesem Zeitpunkt noch keine sichere Verbindung zum Server besteht. Deshalb soll die Anmeldung als root komplett deaktiviert werden und nur bestimmte Benutzer sollen die Möglichkeit haben sich am System authentifizieren zu können.

Hierzu muss die OpenSSH Daemon (sshd) Konfiguration angepasst werden. Ziel ist ein asymmetrisches Kryptosystem zur Authentifizierung anzuwenden. Diese – besser als Public-Key-Kryptosystem bekannte Methode – bietet die höchste Sicherheit, verglichen mit normalen Passwörtern (vgl. Planella, 2011). Des Weiteren sollen alle Möglichkeiten zum unsicheren verbinden via SSH deaktiviert werden und ausschließlich SSH in Version 2 erlaubt sein. Die Konfigurationsdatei für sshd befindet sich im Ordner `/etc/ssh/` mit dem Namen `sshd_conf`. Mit `cp /etc/ssh/sshd_conf /etc/ssh/sshd_conf.bak` kann eine Sicherung der originalen Datei erstellt werden bevor die neue Konfiguration, siehe Anhang 1, erstellt wird.

Bevor sshd neugestartet werden kann – um die neue Konfiguration zu übernehmen – muss noch ein Benutzerkonto angelegt werden. In der Testkonfiguration heißt dieser Benutzer einfach *richard* (der Vorname des Autors). Eine Gruppe für alle Web-Entwickler und -Entwicklerinnen wird auch angelegt, dadurch lassen sich später alle gleichzeitig ansprechen. Dies erleichtert das Rechtemanagement ungemein. Das Heimverzeichnis des neuen Benutzers wird für den Rest der Welt gesperrt. Dies ist

wichtig, damit spätere *chroot*-Umgebungen auch Sinn machen und andere Benutzer nicht auf das Heimverzeichnis des Benutzers zugreifen können.

Damit der neue Benutzer sich am System Authentifizieren kann benötigt es ein Schlüsselpaar. Dies wird mit dem neu angelegten Benutzerkonto erstellt. Bei der Erstellung des Schlüsselpaares muss lediglich den Bildschirmanweisungen gefolgt werden und eine Passphrase für den Schlüssel angegeben werden (diese kann z. B. komfortabel mit KeyPass⁶⁷ generiert werden). Abschließend werden noch die Rechte für den öffentlichen Schlüssel korrigiert (vgl. Jurzik, 2009).

```
# addgroup webdev
# adduser Richard
# usermod -a -G webdev Richard && chmod 700 /home/Richard
# su Richard
$ cd && ssh-keygen -t dsa
$ chmod 600 .ssh/id_dsa.pub
```

Quelltext 2 – Benutzer anlegen für Anmeldung

Damit eine Authentifizierung vom Windows-Rechner möglich ist, muss der private Schlüssel heruntergeladen werden. Mit PuTTYgen kann der Schlüssel dann in verschiedene Formate exportiert werden. Dazu wird PuTTYgen gestartet, der private Schlüssel geladen, die Passphrase eingegeben und dann kann die Konvertierung durchgeführt werden. Damit eine Anmeldung via PuTTY (und unter Zuhilfenahme von Pageant) möglich ist, muss der Schlüssel in das PuTTY eigene Schlüsselformat konvertiert werden.

⁶⁷ <http://keepass.info/> – KeyPass Website

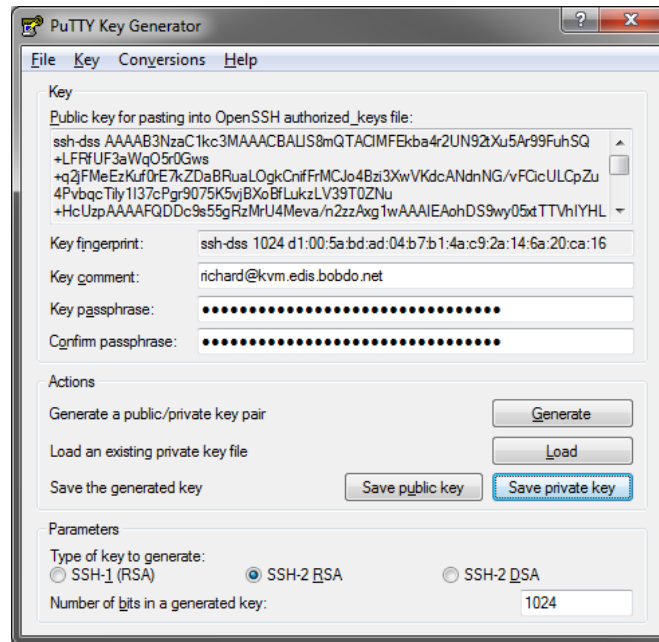


Abbildung 4 – PuTTYgen Hauptfenster

Nun kann die Konfiguration mit `sshd -t` getestet werden und durch `/etc/init.d/sshd restart` liest der Daemon die neuen Parameter ein und startet mit diesen Neu. Wichtig ist, die bestehende Session noch nicht zu beenden, falls es bei der Authentifizierung mit dem neu erstellten Benutzer zu Problemen kommt. Nur als root ist es möglich die `sshd` Konfiguration erneut abzuändern.

In PuTTY wird am besten ein Profil angelegt. IP-Adresse und Port wie in der Konfiguration. Unter *Translation* auf UTF-8 umstellen. Bei *Data* den Benutzernamen gleich vorab eingeben. Unter *SSH* auf *2 only* umstellen und Kompression aktivieren. Im Unterpunkt *Auth* von *SSH Agent Forwarding* aktivieren und den absoluten Pfad zur PuTTY-Schlüsseldatei angeben. Profil speichern und verbinden. Es sollte direkt die Frage nach der Passphrase erscheinen. Insofern die Authentifizierung mit der Passphrase geklappt hat – was sie sollte – kann die noch offene root Sitzung beendet werden.

3.3 Paketverwaltung

Nach erfolgreicher Absicherung der Authentifizierung muss die Paketverwaltung von Debian erst angepasst werden, bevor die erste Software installiert werden kann.

Die Grundkonfiguration listet ausschließlich Pakete aus dem Squeeze-Zweig (stable branch) die auch frei sind (im Sinn von freier Software⁶⁸). Debian stable Repositorien bieten jedoch fast durchgängig veraltete Programme an, wenn auch die stabilsten Versionen. Damit also auch neuere Versionen installiert werden können, muss die Liste der Repositorien die zur Installation zur Verfügung stehen überarbeitet werden.

```
# /etc/apt/sources.list

deb http://ftp.at.debian.org/debian/ squeeze main non-free contrib
deb-src http://ftp.at.debian.org/debian/ squeeze main non-free contrib
deb http://security.debian.org/ squeeze/updates main non-free contrib
deb-src http://security.debian.org/ squeeze/updates main non-free contrib
deb http://ftp.at.debian.org/debian/ squeeze-updates main non-free contrib
deb-src http://ftp.at.debian.org/debian/ squeeze-updates main non-free contrib
deb http://ftp.at.debian.org/debian/ testing main non-free contrib
deb-src http://ftp.at.debian.org/debian/ testing main non-free contrib
deb http://packages.dotdeb.org squeeze all
deb-src http://packages.dotdeb.org squeeze all
deb http://packages.dotdeb.org squeeze-php54 all
deb-src http://packages.dotdeb.org squeeze-php54 all
deb http://mirror.switch.ch/mirror/mariadb/repo/5.5/debian squeeze main
deb-src http://mirror.switch.ch/mirror/mariadb/repo/5.5/debian squeeze main
```

Quelltext 3 – Paketquellen-Konfiguration

Anschließend müssen die GNU-Privacy-Guard-Schlüssel (GnuPG oder GPG) von Dotdeb und MariaDB zum System hinzugefügt werden (vgl. Plessis, 2012) (vgl. Bartholomew, et al., 2012). Ohne diese Schlüssel kann Debian die installierten Pakete nicht auf Integrität überprüfen und verweigert deshalb auch die Installation von Software die aus diesen Repositorien stammen.

```
# wget http://www.dotdeb.org/dotdeb.gpg
# cat dotdeb.gpg | apt-key add -
# rm -f dotdeb.gpg
# apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 0xc9cb082a1bb943db
```

Quelltext 4 – Hinzufügen der GPG-Schlüssel

Um eine genaue Kontrolle darüber zu haben, von welchem Repository ein Paket installiert wird, bietet das Advanced Packaging Tool (apt) von Debian die Möglich-

⁶⁸ <http://www.gnu.org/philosophy/free-sw.de.html> – Erläuterung was unter freier Software verstanden wird.

keit eine preferences-Datei anzulegen, mit der Prioritäten festgelegt werden können (vgl. Kenyon, et al., 2012).

```
# /etc/apt/preferences

Package: *
Pin: Origin mirror.switch.ch
Pin-Priority: 900

Package: *
Pin: Origin packages.dotdeb.org
Pin-Priority: 750

Package: *
Pin: release o=Debian,a=stable
Pin-Priority: 500

Package: *
Pin: release o=Debian,a=testing
Pin-Priority: 250

Package: *
Pin: release o=Debian
Pin-Priority: -1
```

Quelltext 5 – Paketquellen-Prioritätskonfiguration

Die Konfiguration bestimmt, dass Pakete von MariaDB (*mirror.switch.ch*) immer vorgezogen werden. Insofern das Repositorium keine Software bietet, wird auf Dotdeb zurückgegriffen und erst dann auf Debian stable. Mit der Angabe der Kommandozeilenoption *-t* mit dem Wert *testing* können Programme aus dem Wheeze-Zweig (testing branch) installiert werden (z. B. *aptitude -t testing install package-name*).

Nun ist alles konfiguriert und das System kann einem ersten *update* und *upgrade* unterzogen werden. An dieser Stelle muss darauf hingewiesen werden, dass einschlägige Literatur und Onlinequellen immer die Verwendung von *apt* nahelegen. *Apt* hat jedoch einen großen Nachteil gegenüber *aptitude* – welches auch standardmäßig auf jedem Debian-System vorinstalliert wird – da es nämlich Software die ausschließlich aufgrund einer Abhängigkeit installiert wurde nicht als automatisch installiert markiert. Bei der Deinstallation eines Programmes bleiben somit alle automatisch installierten Zusatzprogramme im System vorhanden. Da bekanntlich jede

installierte Software ein Sicherheitsrisiko darstellt (vgl. Amberg, 2007) und es unmöglich ist als Administrator oder Administratorin einen Überblick über installierte Abhängigkeiten zu behalten, ist es schwer empfehlenswert statt apt auf aptitude als Paketmanager zu setzen. Aptitude führt die gewünschten Markierungen durch und bei Deinstallation werden auch alle Zusatzpakete, die nicht mehr benötigt werden, automatisch deinstalliert.

Mit *aptitude update* und *aptitude upgrade* wird das System auf den letzten Stand gebracht und die Konfiguration kann fortgesetzt werden.

3.4 nginx

Um eine optimale Performance zu erreichen und nginx genau an die eigenen Bedürfnisse anzupassen ist es empfehlenswert den Webserver selbst zu kompilieren. Damit nginx selbst kompiliert werden kann werden natürlich Compiler, andere essentielle Programme und Git benötigt, mit *aptitude install build-essential git* kann alles in einem Rutsch installiert werden. Der korrekte Ort zum Installieren von eigener Software unter Debian ist */usr/local/*. Die neueste nginx Version – Entwicklerversion wohlgemerkt – zum Zeitpunkt der Verfassung ist 1.3.4.

```
# cd /usr/local/src
# wget http://nginx.org/download/nginx-1.3.4.tar.gz
# tar -zxvf nginx-1.3.4.tar.gz
# rm -f nginx-1.3.4.tar.gz
```

Quelltext 6 –nginx herunterladen und extrahieren

Um Reguläre Ausdrücke sehr schnell und effizient verarbeiten zu können wird die neueste „Perl Compatible Regular Expressions“-Version (PCRE)⁶⁹ benötigt. Seit Version 8.20 bietet PCRE dank dem PCRE Performance Project⁷⁰ eine Just-in-time-Kompilierung (JIT) an, was durch die Verwendung von sljit erreicht wird. Der Autor von sljit und PCRE-sljit stellt ein paar Benchmarks zur Verfügung die klar zeigen, wie PCRE durch die Verwendung von PCRE-sljit beschleunigt werden kann (vgl.

⁶⁹ <http://www.pcre.org/> – PCRE Website

⁷⁰ <http://sljit.sourceforge.net/pcre.html> – PCRE Performance Project Website

dark100, 2011). Des Weiteren wird die neueste OpenSSL-Version⁷¹ benötigt, nicht nur um HTTPS- und SPDY-Server umsetzen zu können, OpenSSL wird auch zur Generierung von MD5- und SHA1-Hashes verwendet werden. Die letzte benötigte Bibliothek ist zlib⁷² um Ressourcen zur Laufzeit komprimieren und dekomprimieren zu können.

```
# cd ../../lib
# wget ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.31.tar.gz
# tar -zxvf pcre-8.31.tar.gz
# rm -f pcre-8.31.tar.gz
# wget http://www.openssl.org/source/openssl-1.0.1c.tar.gz
# tar -zxvf pcre-1.0.1c.tar.gz
# rm -f pcre-1.0.1c.tar.gz
# git clone git://github.com/madler/zlib.git
```

Quelltext 7 – Benötigte Bibliotheken herunterladen

Um den Fortschritt von Dateiuploads anzeigen zu können – wie es von Drupal unterstützt wird – muss das Drittanbieter-Modul NginxUploadProgressModule inkludiert werden. Das Modul implementiert dabei eine Überwachung von RFC 1867⁷³ konformen POST-Uploads während diese von nginx an den Upstream übermittelt werden und kann den Fortschritt auf verschiedene Arten (konfigurierbar) an eine Applikation zurückgeben.

```
# cd ../../include
# git clone git://github.com/masterzen/nginx-upload-progress-module.git
```

Quelltext 8 – Klonen des Master-Zweiges des NginxUploadProgressModules

Nach Anlegen eines dedizierten nginx-Benutzers und hinzufügen von diesem zur webdev-Gruppe kann nginx konfiguriert, kompiliert und installiert werden. Nach erfolgreicher Installation wird natürlich aufgeräumt.

```
# adduser --no-create-home --disabled-password --disabled-login nginx
# usermod -a -G webdev nginx
# ./configure \
  --prefix=/usr/local \
  --sbin-path=/usr/local/sbin \
  --conf-path=/etc/nginx/nginx.conf \
```

⁷¹ <http://www.openssl.org/> – OpenSSL Website

⁷² <http://zlib.net/> – zlib Website

⁷³ <http://www.ietf.org/rfc/rfc1867.txt> – RFC 1867 Form-based File Upload in HTML


```
--pid-path=/var/run/nginx.pid \  
--lock-path=/var/lock/nginx.lock \  
--error-log-path=/var/log/nginx/error.log \  
--http-log-path=/var/log/nginx/access.log \  
--user=nginx \  
--group=webdev \  
--http-client-body-temp-path=/dev/shm/nginx/body \  
--http-fastcgi-temp-path=/dev/shm/nginx/fastcgi/temp \  
--http-proxy-temp-path=/dev/shm/nginx/proxy/temp \  
--with-cpu-opt=amd64 \  
--with-debug \  
--with-file-aio \  
--with-http_gzip_static_module \  
--with-http_realip_module \  
--with-http_ssl_module \  
--with-md5=/usr/local/lib/openssl-1.0.1c \  
--with-md5-asm \  
--with-openssl=/usr/local/lib/openssl-1.0.1c \  
--with-sha1=/usr/local/lib/openssl-1.0.1c \  
--with-sha1-asm \  
--with-pcre=/usr/local/lib/pcre-8.31 \  
--with-pcre-jit \  
--with-zlib=/usr/local/lib/zlib \  
--without-http_browser_module \  
--without-http_geo_module \  
--without-http_limit_req_module \  
--without-http_limit_conn_module \  
--without-http_memcached_module \  
--without-http_referer_module \  
--without-http_scgi_module \  
--without-http_split_clients_module \  
--without-http_ssi_module \  
--without-http_upstream_ip_hash_module \  
--without-http_userid_module \  
--without-http_uwsgi_module \  
--add-module=/usr/local/include/nginx-upload-progress-module  
# make  
# make install  
# rm -rf /usr/local/src/* /usr/local/lib/* /usr/local/include/*
```

Quelltext 9 – Benutzer anlegen und nginx installieren

Nginx steht nun unter `/usr/local/sbin/nginx` zur Verfügung, die Konfigurationsdateien wurden nach `/etc/nginx/` kopiert – wie angegeben bei der Konfiguration. Die Standardkonfiguration wird jedoch gelöscht und stattdessen eine eigene angepasste Konfiguration eingespielt. Dazu werden erst alle benötigten Ordner erstellt und die korrekten Rechte vergeben, dann kann die Konfiguration die vom Autor in Form eines

GitHub Repositoriums zur Verfügung gestellt wird heruntergeladen und eingespielt werden (vgl. Fussenegger, 2012). Die gesamte Konfiguration ist auch am Ende angehängt.

```
# mkdir -p /dev/shm/nginx/body\  
/dev/shm/nginx/proxy/temp\  
/dev/shm/nginx/fastcgi/temp\  
/var/www/drupal-backup\  
/var/www/passwd\  
/var/www/static  
# chown -R php-fpm:webdev /var/www  
# chown nginx:nginx /var/www/passwd  
# chmod -R 770 /var/www  
# chmod -R g+s /var/www  
# rm -rf /etc/nginx  
# git clone git://github.com/Fleshgrinder/nginx.git /etc/nginx  
# cp /etc/nginx/nginx.sh /etc/init.d/nginx  
# chmod 755 /etc/init.d/nginx  
# update-rc.d nginx defaults  
# cp /etc/nginx/sites/examples/static.conf /etc/nginx/sites/  
# /etc/init.d/nginx reload
```

Quelltext 10 – nginx Installation abschließen

3.4.1 Erster Benchmark

Nachdem nun der Webserver installiert und der erste Host konfiguriert ist können erste Benchmarks durchgeführt werden. Für den ersten Durchlauf wird eine Indexdatei mit genau 100 Byte erstellt.

```
# touch /var/www/static/index.html  
# truncate -s 100 /var/www/static/index.html
```

Quelltext 11 – Erstellung einer statischen 100-Byte-Datei

Nun kann nginx mit `/etc/init.d/nginx start` gestartet werden. Beim Besuch der Seite <http://kvm.edis.bobdo.net/> (die Domain wurde vom Autor extra angelegt und zeigt via DNS auf den Edis KVM Server) erscheint nichts, was korrekt ist, da die Datei leer ist.

3.4.1.1 Testablauf

Um vergleichbare Ergebnisse zu erhalten werden alle Benchmarks nach dem gleichen Schema durchgeführt. Es wird die aktuellste Version 2.70 vom Stresstesttool *siege*⁷⁴ aus dem Testzweig von Debian installiert. Zum Einsatz kommen insgesamt drei Testserver. Der VRS von Edis, der KVM von Edis (localhost) und ein VPS des Autors (Standort Rumänien, GB-Uplink). Jede Belagerung (*siege*) wird für eine Minute mit 500 konkurrierenden Anfragen angesetzt. Die Clients zwischenspeichern keine Daten und fordern komprimierte (GZIP) Inhalte an. Im Heimverzeichnis des Benutzers der / der Benutzerin die *siege* ausführt wird eine Konfigurationsdatei mit dem Namen *.siegerc* mit dem folgenden Inhalt angelegt (vgl. Fulmer, 2010).

```
# ~/.siegerc

logfile          = $(HOME)/siege.log
verbose          = true
csv              = true
logging          = true
protocol         = HTTP/1.1
chunked          = true
cache            = false
accept-encoding  = gzip
benchmark        = true
concurrent       = 500
connection       = close
delay            = 1
internet         = false
show-logfile     = true
time             = 1M
zero-data-ok     = false
```

Quelltext 12 – siege-Konfiguration

Mit *siege http[s]://domain.tld/[url]* wird ein Test gestartet. Es ist auch möglich mehrere Seiten mit *siege* zufällig aufrufen zu lassen, durch die Eingabe von *siege -f datei-mit-urls.txt*. Damit es zu keinen Problemen mit zu vielen geöffneten Sockets kommt, sollte die Begrenzung für Dateihandles angehoben werden. Dies kann zur Laufzeit durch *ulimit -Hn 262144* und *ulimit -Sn 131072* erreicht werden, wobei *Hn* für das sogenannte Hardlimit (nicht überschreitbar) und *Sn* für das Softlimit steht.

⁷⁴ <http://www.joedog.org/siege-home/> – Siege Website

Diese Änderung gilt nur für die aktive Sitzung und wird bei der nächsten Anmeldung wieder auf Standardwerte zurückgesetzt.

3.4.1.2 Ergebnisse – Erster Durchlauf

Transactions:	154942 hits
Availability:	99.99 %
Elapsed Time:	59.64 secs
Data transferred:	14.63 MB
Response time:	0.18 secs
Transaction rate:	2597.95 trans/sec
Throughput:	0.25 MB/sec
Concurrency:	476.50
Successful transactions:	154942
Failed transactions:	20
Longest transaction:	9.47
Shortest transaction:	0.00

Quelltext 13 – Ergebnis Edis VRS erster Durchlauf

Beinahe 2.600 Seitenaufrufe, hochgerechnet wären das ca. 216.000.000 Seitenaufrufe pro Tag. Jedoch gab es 20 fehlgeschlagene Transaktionen, 1.728.000 Besucher erhalten somit einen Fehler im Verlauf des Tages. Bei diesem Ergebnis darf nicht vergessen werden, dass sich beide Server im gleichen Datencenter befinden. Entsprechend ist der Kommunikationsweg sehr kurz und die Server können über ein extrem schnelles LAN miteinander kommunizieren.

Transactions:	111246 hits
Availability:	100.00 %
Elapsed Time:	59.98 secs
Data transferred:	10.50 MB
Response time:	0.26 secs
Transaction rate:	1854.72 trans/sec
Throughput:	0.18 MB/sec
Concurrency:	489.91
Successful transactions:	111246
Failed transactions:	5
Longest transaction:	9.79
Shortest transaction:	0.09

Quelltext 14 – Ergebnis bobdo VPS RO erster Durchlauf

Obwohl der VPS in Rumänien steht und das Routing entsprechend komplexer ist, steht das Ergebnis dem VRS nicht wirklich hinten nach. 155.520.000 Seitenaufrufe pro Tag und „nur“ 432.000 fehlgeschlagene Übertragungen.

```
Transactions:          406689 hits
Availability:          100.00 %
Elapsed Time:          59.49 secs
Data transferred:      38.40 MB
Response time:         0.07 secs
Transaction rate:      6836.26 trans/sec
Throughput:            0.65 MB/sec
Concurrency:           494.12
Successful transactions: 406689
Failed transactions:    0
Longest transaction:    9.37
Shortest transaction:   0.00
```

Quelltext 15 – Ergebnis localhost erster Durchlauf

Einen Benchmark vom localhost auszuführen ist wichtig, da bei entfernten Anfragen immer die Leitung selbst mitgemessen wird. Dies ist lokal natürlich nicht der Fall, dafür muss der Server die Anfragen und die Antworten generieren. Der Server kann fast 7.000 Anfragen pro Sekunde generieren und auch beantworten, nicht ein Fehler ist aufgetreten.

```
Transactions:          344338 hits
Availability:          99.97 %
Elapsed Time:          59.80 secs
Data transferred:      32.51 MB
Response time:         0.16 secs
Transaction rate:      5758.16 trans/sec
Throughput:            0.54 MB/sec
Concurrency:           927.76
Successful transactions: 344338
Failed transactions:    100
Longest transaction:    11.49
Shortest transaction:   0.00
```

Quelltext 16 – Ergebnis localhost 1.000 konkurrierende Verbindungen

Wird die Anzahl an konkurrierenden Verbindungen verdoppelt verschlechtert sich das Ergebnis zusehends. Siehe meldet Socket-Fehler, es stehen nicht genügend Ressourcen zur Verfügung um diese Menge an Anfragen zu generieren und gleichzeitig auch noch zu beantworten.

Dieser Test konnte vom Edis VRS nicht durchgeführt werden. Die Virtualisierung erlaubt es nicht das *ulimit* anzuheben und siege stürzt somit kurzzeitig nach dem Start ab, mit der Meldung, dass die Tabellen für Dateihandles voll sind.

Der bobdo VPS RO ist nicht in der Lage genügend Verbindungen zu generieren und bricht den siege nach etwas mehr als 30 Sekunden ab. Die Resultate sind somit irrelevant.

3.4.2 Systemkonfiguration tunen

Die siege Benchmarks haben direkt gezeigt, dass der Server noch nicht alle eingehenden Anfragen beantworten kann. Mit bis zu 20 Fehlern, bzw. Zeitüberschreitungen pro Sekunde, muss gerechnet werden. Der Flaschenhals ist hierbei weniger der Webserver *nginx* als vielmehr die gesetzten Transmission-Control-Protocol-Parameter (TCP). Die Standardkonfiguration von Linux Kernel (Betriebssystemkerne) ab Version 2.6.+ sind bereits sehr gut, für wirklich schnelle Netzwerke können trotzdem bestimmte Optimierungen vorgenommen werden um einen höheren Durchsatz zu generieren (vgl. ESnet, 2012). Bevor die Systemkonfiguration verändert wird, sollte ein Backup der Standardwerte erstellt werden. Mit *sysctl -a > ~/sysctl_defaults.conf* kann dies erreicht werden (vgl. Fussenegger & Graziano, 2012).

```
# ~/sysctl_tuning_1.conf

net.core.rmem_max           = 67108864
net.core.wmem_max           = 67108864
net.ipv4.tcp_rmem            = 4096 87380 33554432
net.core.wmem_max           = 4096 65536 33554432
net.core.netdev_max_backlog = 250000
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_congestion_control = vegas
net.ipv4.tcp_window_scaling  = 1
net.ipv4.tcp_timestamps     = 1
net.ipv4.tcp_sack            = 1
```

Quelltext 17 – Systemkonfiguration TCP-Tuning

Die neuen Systemkonfigurationsparameter von Quelltext 17 (vgl. ESnet, 2012) können mit `sysctl -p ~/sysctl_tuning_1.conf` eingespielt werden und sind sofort aktiv. Für die Stausteuering (congestion control) von TCP gibt es diverse Algorithmen. Die Standardeinstellung der meisten Betriebssysteme ist TCP Reno, das jedoch auch extrem veraltet ist. Im Linux Kernel 2.6.13–2.6.18 wurde auf TCP BIC (Binary Increase Congestion) gesetzt (vgl. LKDDb, 2012) und seit 2.6.19+ kommt CUBIC TCP zum Einsatz (vgl. LKDDb, 2012). Microsoft verwendet seit Windows Vista respektive Windows Server 2008 Compound TCP (vgl. Sridharan, et al., 2005), es existieren auch Versionen für Linux (vgl. Castellani, 2006), Windows XP (x64) und Windows Server 2003 via Hotfixes (vgl. Microsoft, 2011). Compound TCP funktioniert dabei wie eine Mischung aus TCP Reno und TCP Vegas (vgl. Castellani, 2006).

Welche Algorithmen auf dem System zur Verfügung stehen kann durch Eingabe von `sysctl net.ipv4.tcp_available_congestion_control` herausgefunden werden. Sollte der gewünschte Algorithmus nicht zur Verfügung stehen, ist er möglicherweise als ladbares Kernelmodul verfügbar und kann durch `/sbin/modprobe tcp_$(name)` geladen werden (vgl. ESnet, 2012).

```
# sysctl net.ipv4.tcp_available_congestion_control
net.ipv4.tcp_available_congestion_control = cubic reno
# /sbin/modprobe tcp_vegas
# sysctl net.ipv4.tcp_available_congestion_control
net.ipv4.tcp_available_congestion_control = vegas cubic reno
```

Quelltext 18 –Aktivierung TCP-Stausteueralgorithmus Vegas

In einem Vergleich verschiedener TCP-Stausteuering von 2008 kam TCP Vegas als klarer Sieger hervor (vgl. Jamal & Sultan, 2008). Insofern Vegas zur Verfügung steht, ist es gegenüber den anderen Algorithmen zu favorisieren.

Um einen optimalen Durchsatz zu erreichen, sollte noch die Standardübertragungswarteschlange der Netzwerkkarte an die Geschwindigkeit des Datencenters bzw. des Uplinks des Servers angepasst werden. Der Standardwert in den meisten Linux-Distributionen liegt bei 1.000, dies ist ein optimierter Wert für eine Gbit/s-Anbindung. Für 100 Mbit/s müsste dieser Wert entsprechend auf 100 reduziert werden und für 10 Gbit/s auf 10.000 angehoben werden (vgl. ESnet, 2012).

```
# ifconfig eth0 txqueuelen 1000
# echo '#!/bin/sh -e

# Set the interface queue length for a GBs connection.
/sbin/ifconfig eth0 txqueuelen 1000

exit 0' > /etc/rc.local
```

Quelltext 19 – Anpassung der Übertragungwarteschlange für ein GBit/s

3.4.3 Zweiter Benchmark

Nach Anpassung der Systemkonfiguration folgt ein weiterer Benchmark um den Leistungszuwachs zu bestätigen oder zu widerlegen.

```
Transactions:          650912 hits
Availability:          100.00 %
Elapsed Time:          59.47 secs
Data transferred:      61.46 MB
Response time:         0.04 secs
Transaction rate:      10945.22 trans/sec
Throughput:            1.03 MB/sec
Concurrency:           491.47
Successful transactions: 650912
Failed transactions:    0
Longest transaction:   11.06
Shortest transaction:  0.00
```

Quelltext 20 – Ergebnis Edis VRS zweiter Durchlauf

Die Ergebnisse nach der Sysctl-Anpassung vom VRS-Stresstest sprechen für sich: 950.400.000 Seitenaufrufe pro Tag.

```
Transactions:          203142 hits
Availability:          100.00 %
Elapsed Time:          59.79 secs
Data transferred:      19.37 MB
Response time:         0.15 secs
Transaction rate:      3397.59 trans/sec
Throughput:            0.32 MB/sec
Concurrency:           498.61
Successful transactions: 203142
Failed transactions:    0
Longest transaction:   9.16
Shortest transaction:  0.08
```

Quelltext 21 – Ergebnis bobdo VPS RO zweiter Durchlauf

Nahezu eine Verdoppelung, am Schluss bleiben 293.760.000 Aufrufe pro Tag ohne einen einzigen Fehler.

```
Transactions:          391580 hits
Availability:          100.00 %
Elapsed Time:          59.20 secs
Data transferred:      36.97 MB
Response time:         0.07 secs
Transaction rate:      6614.53 trans/sec
Throughput:            0.62 MB/sec
Concurrency:           490.39
Successful transactions: 391580
Failed transactions:    0
Longest transaction:    9.13
Shortest transaction:   0.00
```

Quelltext 22 – Ergebnis localhost zweiter Durchlauf

Dieses Ergebnis ist mitunter das Interessanteste von allen. Werden die Werte mit dem des ersten lokalen Durchgangs verglichen, so haben sich alle verschlechtert. Was auf den ersten Blick schlecht oder verwirrend aussieht, ist – werden die anderen Benchmarks hinzugezogen – eine perfekte Bestätigung für die zuvor durchgeführten TCP-Stack-Optimierungen. Siehe kann selbst nicht mehr so viele Verbindungen generieren wie es gerne würde, da nginx mehr Betriebssystemressourcen erhält um sich um die Antworten der Anfragen zu kümmern. Dies wird besonders beim Ergebnis vom Edis VRS sichtbar. Als weiterer Test wären 1.000 konkurrierende Benutzer vom VRS zum KVM sehr interessant, dieser lässt sich jedoch – wie bereits erwähnt – aufgrund der Virtualisierungseinschränkungen nicht durchführen.

3.4.4 Dritter Benchmark

Um herauszufinden wie viel Einfluss die GZIP-Kompression zur Laufzeit auf die Benchmarks hat (vgl. Souders & Koechley, 2007), soll im dritten Durchgang das NginxGzipStaticModule zum Einsatz kommen. Dazu wird die zuvor erstellte Indexdatei mit `gzip -9 index.html` komprimiert. Die Option `-9` sorgt hierbei dafür, dass die maximale Kompression zum Einsatz kommt und die Datei schrumpft von genau 100 Byte auf 35. Damit das Modul die Datei auch finden kann müssen die folgenden Benchmarks mit der URL <http://kvm.edis.bobdo.net/index.html> ausgeführt werden.

```
Transactions:          693138 hits
Availability:          100.00 %
Elapsed Time:          59.44 secs
Data transferred:      23.14 MB
Response time:         0.04 secs
Transaction rate:      11661.14 trans/sec
Throughput:            0.39 MB/sec
Concurrency:           489.21
Successful transactions: 693138
Failed transactions:    5
Longest transaction:    9.24
Shortest transaction:   0.00
```

Quelltext 23 – Ergebnis Edis VRS dritter Durchlauf

Eine klare Steigerung, viel mehr war aufgrund der minimalen Optimierung nicht zu erwarten. Was jedoch zu erwarten war: die maximale CPU-Auslastung am KVM Server sank von 98 % auf 58 % (die Arbeitsspeicherauslastung von nginx stieg nie über 1 % und blieb meist sogar unter 0,5 %), womit mehr Rechenleistung für andere Prozesse vorhanden ist.

```
Transactions:          208587 hits
Availability:          100.00 %
Elapsed Time:          59.59 secs
Data transferred:      6.96 MB
Response time:         0.14 secs
Transaction rate:      3500.37 trans/sec
Throughput:            0.12 MB/sec
Concurrency:           484.75
Successful transactions: 208587
Failed transactions:    1
Longest transaction:    9.15
Shortest transaction:   0.09
```

Quelltext 24 – Ergebnis bobdo VPS RO dritter Durchlauf

```
Transactions:          399833 hits
Availability:          100.00 %
Elapsed Time:          59.95 secs
Data transferred:      13.35 MB
Response time:         0.07 secs
Transaction rate:      6669.44 trans/sec
Throughput:            0.22 MB/sec
Concurrency:           488.05
Successful transactions: 399833
Failed transactions:    0
```

```
Longest transaction:      9.40
Shortest transaction:     0.00
```

Quelltext 25 – Ergebnis localhost dritter Durchlauf

Auch am bobdo VPS und localhost gibt es eine kleine Steigerung. Die CPU-Rechenleistung die der localhost-Server beim Antworten einspart kann er in siege investieren um mehr Anfragen zu generieren.

3.5 PHP und MariaDB

Die Benchmarks mit einer 100 Byte großen statischen HTML-Datei sind interessant um die Kapazitäten des Servers auszuloten, bieten jedoch keinen Ausblick auf eine reale Anwendung. Websites werden dynamisch zur Laufzeit aus Daten in einem Datenbanksystem generiert. Um also realistische Benchmarks erstellen zu können, muss eine derartige Umgebung umgesetzt werden. Die Installation von PHP und MariaDB gestaltet sich hierbei dank Dotdeb und MariaDB Repositorien sehr einfach.

```
# adduser --no-create-home --disabled-password --disabled-login php-fpm
# usermod -a -G webdev php-fpm
# adduser --no-create-home --disabled-password --disabled-login mysql
# usermod -a -G webdev mysql
# aptitude install\
  php5-fpm php5-cli php5-gd php5-apc php-pear php5-mysqld mariadb-server-5.5
```

Quelltext 26 – Installation von PHP und MariaDB

- **php5-fpm**

Der PHP FastCGI Process Manager⁷⁵ ist eine alternative Implementierung von PHP FastCGI mit zusätzlichen Funktionen.

- **php5-cli**

Das PHP Command Line Interface⁷⁶ ermöglicht es PHP-Skripte von der Kommandozeile auszuführen. Dies wird benötigt um administrative Drupal-Aufgaben mit Drush⁷⁷ erledigen zu können. Drush ist dabei nicht nur sehr praktisch in der Anwendung für Systemadministratoren und -

⁷⁵ <http://php-fpm.org/> – PHP-FPM Website

⁷⁶ <http://www.php-cli.com/> – PHP-CLI Website

⁷⁷ <http://drupal.org/project/drush/> – Drush Projektwebseite

administratorinnen die die Kommandozeile gewohnt sind und es erhöht auch die Sicherheit, da keine PHP-Skripts die administrative Aufgaben erfüllen über das Web aufrufbar sein müssen.

- **php5-gd**

Die GD Graphics Library⁷⁸ wird benötigt um Bildmanipulationen (vergrößern, verkleinern, zuschneiden, ...) durchführen zu können. Alternativ oder zusätzlich kann auch php5-imagick installiert werden. Das würde ImageMagick⁷⁹ installieren, das einen größeren Funktionsumfang bietet und teilweise auch bessere Performance.

- **php5-apc**

Der Alternative PHP Cache⁸⁰ ist ein Opcodecache⁸¹ für PHP. Da PHP eine dynamisch interpretierte Programmiersprache ist, muss jedes Skript zur Laufzeit interpretiert werden. Dieser Vorgang kostet natürlich Zeit und Rechenleistung. Mit APC können die bereits kompilierten Skripts, also der Opcode bzw. Maschinencode für die aktuell verwendete Architektur, abgefangen und zwischengespeichert werden (vgl. Henderson, 2006).

- **php-pear**

Das PHP Extension and Application Repository (PEAR)⁸² bietet viele nützliche Erweiterungen die später einfach installiert werden können, ähnlich wie apt-get bzw. aptitude. In diesem Testaufbau wird es benötigt um Drush zu installieren.

- **php5-mysqld**

MySQL native driver for PHP⁸³ ist ein Drop-in-Ersatz für die Standard MySQL-Bibliothek von PHP. Mysqlnd ist direkt in PHPs C-Infrastruktur integriert und stark optimiert und bietet deshalb eine erhöhte Performance.

⁷⁸ <http://www.boutell.com/gd/> – GD Graphics Library Website

⁷⁹ <http://www.imagemagick.org/> – ImageMagick Website

⁸⁰ <http://pecl.php.net/package/APC> – PHP-APC Projektwebseite

⁸¹ Opcode wird auch als Operation Code bezeichnet.

⁸² <http://pear.php.net/> – PEAR Website

⁸³ <http://dev.mysql.com/downloads/connector/php-mysqlnd/> – mysqlnd Webseite

3.5.1 PHP-Konfiguration

PHP ist bekannt dafür, von Haus aus mit diversen Sicherheitslücken ausgestattet zu sein. Diesem Umstand wurde von den PHP-Autoren in den letzten Jahren zwar etwas entgegengewirkt, jedoch ist spätestens seit Stefan Essers Abgang vom PHP-Security-Team bekannt, dass die PHP-Autoren keinen sehr großen Wert auf Sicherheit legen (vgl. Bachfeld, 2006). Eine korrekte Konfiguration von PHP ist deshalb unabdingbar. Die verwendete Konfiguration (siehe Anhang 19 und Anhang 20) bildet einen Kompromiss zwischen Sicherheit und Performance, wobei klar die Geschwindigkeit im Vordergrund steht. Durch eine korrekte Programmierung können eigentlich alle PHP Sicherheitslücken vermieden werden. Da Drupal zum Einsatz kommt, sollten Sicherheitslücken entsprechend gering sein.

Um die Performance von PHP weiter zu steigern wird APC eingesetzt. Der Grundgedanke bei der Konfiguration (siehe Anhang 21) ist hierbei einfach. Alle Daten zur Darstellung werden aus der Datenbank gelesen, die PHP-Dateien selbst ändern sich kaum bis nie. APC soll deshalb den Opcode von allen Skripten für immer im Cache behalten. Werden PHP-Dateien verändert, muss der gesamte Prozess neugestartet werden damit der Cache geleert wird. Dies wirkt Fragmentierung und Speicherlecks entgegen, es erübrigt auch die Verwendung eines Cache-Frontends.

3.5.2 MariaDB-Konfiguration

Alle anderen PHP-Erweiterungen müssen nicht konfiguriert werden und es kann direkt zur Konfiguration der Datenbank übergegangen werden. Optimierung von DB-Systemen wird oft als eine Wissenschaft für sich angesehen. Tatsächlich gibt es selten bei Software so viele Variablen an denen geschraubt und gedreht werden kann. Durch die Verwendung von MariaDB als DBS erhöht sich der Umfang gegenüber MySQL weiter. Es kommt eine neue Storage Engine zum Einsatz – Perconas XtraDB als Drop-in-Ersatz für InnoDB – und neue Funktionen die MySQL nicht bietet. Eine generell gültige high-performance Konfiguration für eine Datenbank zu verfassen ist fast unmöglich. Die verfasste Konfiguration (siehe Anhang 22) kann jedoch als soli-

der Startpunkt angesehen werden (vgl. Monty Program AB, 2012) (vgl. Oracle, 2012).

3.5.2.1 Hugepages

Für eine erhöhte Performance von InnoDB werden Hugepages (riesige Speicherseiten) konfiguriert. Dies erlaubt es dem DBS große Datensätze komplett im Speicher abzulegen, anstatt diese in 4 K chunks (Einheiten) aufzuteilen. Mit `cat /proc/meminfo | grep -i huge` kann überprüft werden ob Hugepages aktiv sind.

```
# cat /proc/meminfo | grep -i huge
HugePages_Total:      0
HugePages_Free:       0
HugePages_Rsvd:       0
HugePages_Surp:       0
Hugepagesize:         2048 kB
```

Quelltext 27 – Überprüfung ob Hugepages aktiviert sind

Insofern Hugepagesize einen Wert aufweist ist die Unterstützung vom Betriebssystem gegeben. Sind alle anderen Werte 0, müssen diese erst noch konfiguriert werden. Die Ausgabe von 2048 kB wie in Quelltext 27 ist die typische Größe unter Linux – dieser Wert ist jedoch nicht Betriebssystem sondern CPU-Architektur abhängig (vgl. Piat, et al., 2011). Um nun Hugepages zu aktivieren muss erneut die Systemkonfiguration angepasst werden. Die Konfiguration aus Quelltext 17 wird entsprechend mit der folgenden aus Quelltext 28 erweitert (vgl. ebd.) (vgl. Fussenegger, 2012) (vgl. Vivek, 2009) (vgl. Oracle, 2012).

```
# ~/sysctl_tuning_2.conf

vm.nr_hugepages      = 384
vm.hugetlb_shm_group = 1004
kernel.shmmax        = 805306368
kernel.shmall        = 805306368
```

Quelltext 28 – Systemkonfiguration zur Aktivierung von Hugepages

Die gid (group identifier) der mysqld-Gruppe kann einfach mit `id mysqld` herausgefunden werden. Die neue Systemkonfiguration kann mit dem Kommando `sysctl -p ~/sysctl_tuning_2.conf` eingelesen und übernommen werden. Zuletzt muss sichergestellt werden, dass der mysqld-Benutzer auch den Speicher für sich reservieren kann.

Dazu wird ein neues init-Skript erstellt, welches das Limit für den Prozess beim Start anhebt (siehe Anhang 23). Ein Neustart des DBS mit `/etc/init.d/mariadb restart` aktiviert Hugepages. Um sicherzugehen kann die Logdatei überprüft werden und nochmals die Ausgabe von `meminfo` überprüft werden. Ist diese ähnlich wie in Quelltext 29 ist alles korrekt konfiguriert.

```
# cat /proc/meminfo | grep -i huge
HugePages_Total:      384
HugePages_Free:       251
HugePages_Rsvd:       115
HugePages_Surp:        0
Hugepagesize:         2048 kB
```

Quelltext 29 – Hugepages nach korrekter Konfiguration

3.6 Drupal installieren

Nachdem nun nginx, PHP und MariaDB installiert und konfiguriert sind, kann Drupal installiert werden. Dank Drush⁸⁴ geht das sehr einfach und schnell (vgl. Lawrence, et al., 2012) (vgl. Diverse, 2012).

```
# mysqladmin -u root -p create drupal7
# mysql -u root -p
MariaDB: GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX,
        ALTER, LOCK TABLES, CREATE TEMPORARY TABLES ON
        `drupal7`.* TO 'drupal7'@'localhost' IDENTIFIED BY 'drupal7';
MariaDB: FLUSH PRIVILEGES;
MariaDB: exit
# pear channel-discover pear.drush.org
# pear install drush/drush
# cd /var/www && su php-fpm
$ drush qd\
  Drupal\
  Devel\
  --account-mail='richard@fussenegger.info'\
  --account-name='richard'\
  --account-pass='drupal7'\
  --db-su='drupal7'\
  --db-su-pw='drupal7'\
  --db-url='mysql://drupal7:drupal7@localhost/drupal7'\
  --server='cgi'\
  --site-mail='noreply@bobdo.net\'
```

⁸⁴ <http://www.drush.org/> – Drush Website

```
--site-name='Edis KVM'\n--version-control='backup'\n--backup-dir='/var/www/drupal-backup'\n$ cd drupal/ && my drupal/{,.*} . && rmdir drupal\n$ exit
```

Quelltext 30 – Installation von Drupal 7 mit Drush

Nun muss noch der virtuelle Host samt Reverse Proxy konfiguriert werden. Die Konfiguration hierfür liegt im GitHub Repository, das geklont wurde, bereits vor (vgl. Fussenegger, 2012) (vgl. Almeida, 2012), und muss nur leicht angepasst werden (siehe Anhang 17 und Anhang 18).

```
# cd /etc/nginx/sites/examples\n# cp backend_servers.conf drupal_proxy.conf ..\n# rm -f ../static.conf\n# /etc/init.d/nginx reload\n# cd /var/www/drupal\n# drush -y en devel devel_generate
```

Quelltext 31 – Virtuellen Host für Drupal und Devel Modul aktivieren

Als einziges Modul wurde Devel⁸⁵ installiert, welches nun auch durch die Eingabe von `drush -y en devel devel_generate` bereits aktiviert ist. Dieses Zusatzmodul für Drupal erlaubt es, in einem Schnellverfahren n Seiten mit n Kommentaren zu erstellen. Für die finalen Benchmarks werden 150 Seiten und Kommentare im Verwaltungsbereich erstellt.

3.7 Finaler Benchmark

Nachdem nun eine dynamische Seite erstellt wurde, samt Inhalten, Reverse Proxy mit Cache und FastCGI Cache, kann der finale Testlauf beginnen. Um reale Bedingungen zu schaffen wird eine siege URL-Liste erstellt, wodurch nicht nur die Startseite abgerufen wird, sondern Inhalte nach dem Zufallsprinzip.

```
var $ = jQuery, link = null, output = "";\njQuery("link,script,a,img").each(function () {\n  link = $(this).attr("href") || $(this).attr("src");\n  if (link != null && link != "" && link.charAt(0) != "#") {\n
```

⁸⁵ <http://drupal.org/project/devel> – Devel Projektseite


```
if (link.charAt(0) === "/") {
    link = window.location.protocol + "://" + window.location.hostname + link;
}
if (link.match("http://kvm")) {
    output += link + "\n";
}
}
});
jQuery("body").html("<pre style='width:100%;height:100%'>" + output + "</pre>");
```

Quelltext 32 – Einfaches jQuery-Skript zum Sammeln der URLs

Mit dem einfachen jQuery-Skript aus Quelltext 32 das über die Firebug-Konsole⁸⁶ oder die Chrome Konsole⁸⁷ ausgeführt die URLs der aktuell betrachteten Webseite sammelt, kann die URL-Liste für sie erstellt werden. Nach entfernen aller Duplikate bleiben 347 mehr oder minder eindeutige Links übrig.

3.7.1 Siege Ergebnisse

Transactions:	285860 hits
Availability:	100.00 %
Elapsed Time:	59.41 secs
Data transferred:	3609.42 MB
Response time:	0.10 secs
Transaction rate:	4811.65 trans/sec
Throughput:	60.75 MB/sec
Concurrency:	483.87
Successful transactions:	283931
Failed transactions:	0
Longest transaction:	15.03
Shortest transaction:	0.00

Quelltext 33 – Ergebnis Edis VRS PHP-Durchlauf

414.720.000 Anfragen pro Tag, in diesem Fall nicht nur Seitenaufrufe da auch Ressourcen wie CSS, JavaScript und Bilder angefordert wurden. Jedoch ist der VRS im gleichen Datencenter.

Transactions:	70637 hits
Availability:	99.75 %
Elapsed Time:	60.01 secs

⁸⁶ <http://getfirebug.com/> – Firebug Website

⁸⁷ <https://developers.google.com/chrome-developer-tools/docs/console> – Chrome Konsolen Webseite

```
Data transferred:      1413.56 MB
Response time:         0.42 secs
Transaction rate:      1177.09 trans/sec
Throughput:            23.56 MB/sec
Concurrency:           497.02
Successful transactions: 69974
Failed transactions:    178
Longest transaction:    14.24
Shortest transaction:   0.10
```

Quelltext 34 – Ergebnis bobdo VPS RO PHP-Durchlauf

101.692.800 ist zwar nur noch ein Viertel, kann sich aber immer noch sehen lassen. 178 Fehlschläge ist dabei das größere Problem, es darf jedoch nicht vergessen werden, dass dies durch äußere Faktoren auch zustande kommen kann und nicht unbedingt bedeutet, dass der Server nicht fähig gewesen wäre die Anfrage zu Beantworten. Auf einen localhost Test wird an dieser Stelle verzichtet, da die eigenen Kapazitäten bereits in den ersten drei Benchmarks zur Genüge ausgelotet wurden.

Wird davon ausgegangen, dass jede einzelne Seite sich aus zehn verschiedenen Ressourcen zusammenstellt (z. B. das HTML-Dokument selbst, CSS, JavaScript, Bilder, Favicon, ...) würde das bedeuten, dass der jetzige Aufbau mindestens 10 Millionen Seitenaufrufe pro Tag ohne Probleme verarbeiten kann. Wohlgermerkt kam es am Server selbst zu keinen Engpässen bezüglich Hardwareauslastung und es wären noch weitaus mehr Anfragen möglich. Leider sind die verwendeten Server nicht in der Lage so viele Anfragen überhaupt erst zu generieren. Ein Versuch des Autors mit vier Servern, mit jeweils 500 konkurrierenden Benutzern, den VPS zu bombardieren schlug fehl. Höchst wahrscheinlich aufgrund von Firewalls die diese immense Menge an Anfragen einfach unter der Annahme eines Denial of Service (DoS) Angriffes blockierten.

Beim finalen Testdurchlauf handelte es sich ausschließlich um GET-Anfragen. Die Datenbank kam hierbei eigentlich kaum zum Einsatz, da die verschiedenen, implementierten Caching-Mechanismen dafür sorgen, dass die Inhalte mehrfach zwischengespeichert werden. Insofern eine Website auch sehr viele Schreibvorgänge durchführen muss – wie z. B. Facebook mit den „Gefällt Mir“-Schaltflächen (vgl.

Sawall, 2012) – müsste die Datenbank genau beobachtet werden und weiter angepasst werden. So kam es z. B. beim automatischen Erstellen der Artikel und Kommentare mit Devel zu verschiedenen PHP- und MySQL-Fehlern weil das Modul noch nicht für PHP 5.4 angepasst wurde. Dies wäre bei reiner Verwendung von Drupal nicht der Fall gewesen, da das CMS bereits 100 % kompatibel zur neuesten PHP-Veröffentlichung. Bei kleinen bis mittleren Websites dürfte das Datenaufkommen jedoch weit geringer sein als wie bei Facebook, Google oder Yahoo! und deshalb wird in diesem Benchmark – dieser Arbeit im Gesamten – auf die reine Auslieferung von Inhalten eingegangen.

3.8 Firewall

Offensichtlich ist der Edis KVM Server nicht ungeschützt im Internet, trotzdem sind im Moment noch alle Ports, ein- und ausgehend, geöffnet und erlauben es einem Angreifer eigentlich alles zu machen was er oder sie will. Um diesem Umstand entgegenzuwirken muss eine Firewall entsprechend konfiguriert werden.

Die Administration einer Firewall unter Linux wird oft als komplex empfunden. Das quasi Standardprogramm zur Verwaltung des Linux 2.4 bis 2.6 Paketfilters ist iptables⁸⁸. Tatsächlich ist iptables extrem umfangreich und mächtig. Es existieren endlos viele andere Programme die den Umgang mit der Firewall vereinfachen sollen. Ubuntu verwendet standardmäßig ufw^{89,90} (Uncomplicated Firewall), das Programm agiert dabei als Frontend für iptables und erleichtert die Konfiguration ungemein. Die Software steht auch in den Debian Repositorien zur Installation bereit⁹¹. Mit *aptitude install ufw* kann es installiert werden.

Die Standardeinstellung von ufw blockiert alle eingehenden Verbindungen. Bevor die neuen Regeln aktiviert werden können muss sichergestellt werden, dass der sshd Port Anfragen via TCP erlaubt, ansonsten können sich Benutzer und Benutzerinnen

⁸⁸ <http://www.netfilter.org/projects/iptables/index.html> – iptables Website

⁸⁹ <https://wiki.ubuntu.com/UncomplicatedFirewall> – UFW Website

⁹⁰ <https://launchpad.net/ufw> – UFW Quelltext

⁹¹ <http://packages.debian.org/squeeze/ufw> – UFW Debian Paketwebsite

nicht mehr am System authentifizieren. Es ist bei `ufw` möglich anhand des Service-
namens Regeln zu erstellen. Da die eingesetzte `sshd` Konfiguration jedoch einen Port
verwendet der eigentlich nicht für SSH vorgesehen ist muss eine eigene Regel mit
`ufw limit 21976` erstellt werden. Die Option `limit` sorgt hierbei automatisch dafür, IP-
Adressen die zu viele Verbindungsversuche über diesen Port initiieren, für eine ge-
wisse Zeit zu blocken, dadurch werden effektiv Brute-Force-Attacken abgewendet
(vgl. K-Rich & Costales, 2008) (vgl. Kemp, 2005) (die jedoch aufgrund des verwen-
deten Public-Key-Kryptosystems sowieso keine Chance haben, siehe Kapitel 3.2).
Mit `ufw enable` wird die Firewall aktiviert und mit `ufw status verbose` können die
aktuell angewendeten Regeln eingesehen werden.

Für einen Server, der ausschließlich einen Webserver nach außen zur Verfügung
stellt, müssen natürlich die entsprechenden Ports freigegeben werden. In der vorlie-
genden Konfiguration ist dies der Port 80, mit dem Kommando `ufw allow 80` kann
dieser geöffnet werden. Soll auch HTTPS/SPDY zur Verfügung gestellt werden kann
der Port mit `ufw allow 443` geöffnet werden.

Aus einem unerfindlichen Grund erlaubt `ufw` unter Debian keine Antworten auf An-
fragen die vom Server aus initiiert wurden. Dies führt z. B. dazu, dass selbst DNS
und `aptitude` nicht mehr funktionieren. Den Grund dafür weshalb `ufw` `iptables` falsch
konfiguriert, konnte der Autor nicht herausfinden. Mit `iptables -A INPUT -m state --
state ESTABLISHED,RELATED -j ACCEPT` kann diese Einschränkung jedoch auf-
gehoben werden (vgl. Fussenegger & nojak, 2012).

4. Konfigurationserweiterungen

Die in dieser Arbeit vorgestellte Konfiguration beschränkt sich auf das wesentliche und zeigt, wie selbst mit einem kleinen VPS eine hoch performante Web-Applikation umgesetzt werden kann. Leider ist es im Rahmen einer solchen Arbeit nicht möglich genauer in die Details einzugehen, da dies schlichtweg den Rahmen sprengen würde. Mit einzelnen, hier vorgestellten Themengebieten ließen sich ganze Bücher füllen. Deshalb werden in diesem Kapitel mögliche Erweiterungen kurz zusammengefasst, mit Weblinks zu weiteren Informationen. Keine der vorgestellten Erweiterungen ist jedoch notwendig um den Webserver zu betreiben.

4.1 CloudFlare

CloudFlare⁹² wurde 2009 von Matthew Prince, Lee Holloway und Michelle Zatlyn ins Leben gerufen und gewann noch im selben Jahr die renommierte Harvard Business School Business Plan Competition (vgl. Harvard University, 2012). Dabei dient der Dienst als Drop-in für den eigenen Domain Name Service (DNS). Alle Anfragen werden zuerst über das Cloud-System geleitet und gelangen erst dann zum eigenen Server. Dabei dient CloudFlare als Content Delivery Network (CDN), Firewall, Analysetool und vieles mehr. Besonders für Projekte die eine globale Ausrichtung anstreben, sich jedoch nicht die Serverinfrastruktur leisten können oder wollen, ist das Angebot sehr interessant. Der Grunddienst ist dabei sogar kostenlos und die Bezahlung wird erst bei enormen Besucherzahlen, oder bei Verwendung von z. B. SSL/TLS nötig.

4.2 Asynchrone Frameworks

Beim Thema „High Performance“ dürften an dieser Stelle die neuen asynchronen Frameworks (Webserver) nicht unerwähnt bleiben. Der Hype wurde erst unlängst durch node.js⁹³ losgetreten, es gibt jedoch weitere interessante und teilweise sogar

⁹² <https://www.cloudflare.com/> – CloudFlare Website

⁹³ <http://nodejs.org/> – node.js Website

schnellere Frameworks (vgl. Rusu, 2012). Hierbei handelt es sich noch um sehr junge Technologien, jedoch sprechen die Ergebnisse für sich und dürften in den nächsten Jahren einige interessante Projekte hervorbringen. Neben dem bereits erwähnten node.js sind besonders vert.x⁹⁴, Twisted⁹⁵ und Luvit⁹⁶ zu nennen.

4.3 Hardening

Die Verwendung von Security-Enhanced Linux (SELinux)^{97,98}, Application Armor (AppArmor)⁹⁹ oder grsecurity¹⁰⁰ zur Absicherung des Betriebssystems sind sehr empfehlenswert (vgl. Gite, 2009). Des Weiteren sollte bei der Installation von Linux bereits auf eine korrekte Partitionierung geachtet werden (vgl. Gite, 2008). Insofern Internet Protocol Version 6 (IPv6) nicht verwendet wird, sollte es komplett deaktiviert werden. Es kann sonst zu leicht passieren, dass Sicherheitslücken in diesem neuen Protokoll übersehen werden. Bis dato existieren noch keine Tools die automatisch beim Absichern von IPv6 helfen können (vgl. Gite, 2010). Logdateien sollten mit logwatch¹⁰¹ oder logcheck¹⁰² im Auge behalten werden. Beide Tools filtern „normale“ Logeinträge heraus und mailen eine Zusammenfassung an den Systemadministrator. Somit können ungewöhnliche Vorgänge schnell identifiziert werden. Ein Network Intrusion Detection System (NIDS) sollte eingesetzt werden, um böses Verhalten wie DoS-Attacken, Port-Scans oder gar Einbruchsversuche in das System durch Überwachung des Netzwerkverkehrs frühzeitig zu erkennen. Zum einen gibt es hier Advanced Intrusion Detection Environment (AIDE)¹⁰³, das die Integrität von Verzeichnissen und Dateien überwacht und zum anderen Snort¹⁰⁴, das Paketlogging und Echtzeitverkehrsanalyse von IP-Netzwerken vornimmt. Auch

⁹⁴ <http://vertx.io/> – vert.x Website

⁹⁵ <http://twistedmatrix.com/> – Twisted Website

⁹⁶ <http://luvit.io/> – Luvit Website

⁹⁷ <http://www.nsa.gov/research/selinux/> – Security-Enhanced Linux Website

⁹⁸ <http://wiki.debian.org/SELinux> – SELinux Debian Wiki Artikel

⁹⁹ <http://wiki.apparmor.net/> – AppArmor Wiki

¹⁰⁰ <http://grsecurity.net/> – grsecurity Website

¹⁰¹ <http://sourceforge.net/projects/logwatch/> – Logwatch Website

¹⁰² <http://logcheck.org/> – Logcheck Website

¹⁰³ <http://aide.sourceforge.net/> – AIDE Website

¹⁰⁴ <http://www.snort.org/> – Snort Website

Fail2ban¹⁰⁵ ist ein gerne eingesetztes Tool um böartiges Verhalten festzustellen und die betroffenen IPs automatisch auszusperrern.

Weitere einfache Mittel zur Absicherung des eigenen Servers umfassen grundlegende Dinge wie korrekte Datei- und Verzeichnisberechtigungen, Erstellung von regelmäßigen Sicherungskopien (im Idealfall geregelt anhand einer Backup-Policy) und das Abonnieren der Sicherheitsmailinglisten der eingesetzten Linux Distribution¹⁰⁶.

Hardening ist ein komplexes und weitreichendes Themengebiet. Jeder Systemadministrator und -administratorin muss selbst einschätzen wie wichtig Sicherheit für das eigene System ist. Gute Bücher für einen Einstieg sind „Linux Security Cookbook“ (ISBN 0596003919), „Linux Server Security“ (ISBN 0596006705) von O'Reilly Media und „Hardening Linux“ (ISBN 1590594444) von Apress. Einen guten Einstieg in das Thema gibt es Online auch von Wreski Dave unter dem Titel „Linux Security HOWTO“ unter http://tldp.org/HOWTO/html_single/Security-HOWTO/. Eine weitere Onlinequelle für Informationen ist nixCraft (<http://www.cyberciti.biz/>), dabei dreht es sich hierbei allgemein um Linux und nicht ausschließlich um Sicherheit.

¹⁰⁵ <http://www.fail2ban.org/> – Fail2ban Website

¹⁰⁶ <http://www.debian.org/security/> – Debian Security Website

5. Resümee

Die hier vorgestellte Konfiguration bietet eine solide Grundlage für Websites, aber ist bei weitem nicht perfekt. Besonders die Datenbankkonfiguration muss immer angepasst werden. Es gibt Tools die dem Administrator / der Administratorin dabei helfen, wie z. B. der bekannte MySQLTuner¹⁰⁷, der anhand von Logdateien der Datenbank mögliche Engpässe identifiziert. Das Skript gibt dabei auch gleich Vorschläge für Änderungen an der Konfiguration aus.

Die Testergebnisse von siege, die hier gesammelt wurden, sind alle mit Vorsicht zu genießen und bilden lediglich eine Grundlage zum Vergleich. Wie bereits zuvor erwähnt werden viele Dinge – die beim täglichen Surfen von Benutzern automatisch aufgrund der verwendeten Browser aktiviert sind – wie Keep-Alive oder der integrierte Browsercache nicht verwendet. Der subjektive Eindruck beim Durchklicken durch die finale Drupal-Website war extrem gut, die Seiten luden mehr oder weniger sofort und ohne Verzögerung, alle Bilder und andere Ressourcen waren einfach sofort da. Dies lässt sich jedoch nicht wissenschaftlich dokumentieren (außer anhand eines Tests mit Benutzern, was jedoch nicht das Ziel dieser Arbeit war).

Weitere limitierende Faktoren wie die Verbindungen, das Routing, die siege-Server selbst wurden im Text auch bereits angesprochen. Eventuelle Limits des KVM-Servers wurden nie erreicht, weil die siege-Server selbst nicht stark genug waren um überhaupt erst genügend Anfragen zu produzieren. Bzw. schlug der Test mit etlichen Verbindungen von verschiedenen Servern fehl weil eine Firewall von Edis dies offensichtlich blockierte. Wer also die realen Grenzen ausloten möchte, bräuchte einen anderen Server der viel mehr Ressourcen besitzt.

In der vorliegenden Konfiguration wurde nginx als Webserver verwendet. Der Grund dafür wurde bereits erklärt: light-weight server und hoher Verbreitungsgrad. Im Anschluss habe ich noch siege-Tests mit anderen Webservern durchgeführt. Es kamen

¹⁰⁷ <http://mysqltuner.com/> – MySQLTuner Website

jeweils die aktuellen Pakete aus den Debian Repositorien zum Einsatz, außer bei nginx, hier wurde von Dotdeb das nginx-full¹⁰⁸ Paket verwendet. G-WAN und Hiawatha Webserver sind nicht mit dabei, beide bieten keine fertigen Pakete und hätten selbst kompiliert werden müssen. Sinn dieses Vergleiches ist jedoch eine Standardinstallation mit der hier erarbeiteten angepassten Konfiguration zu vergleichen.

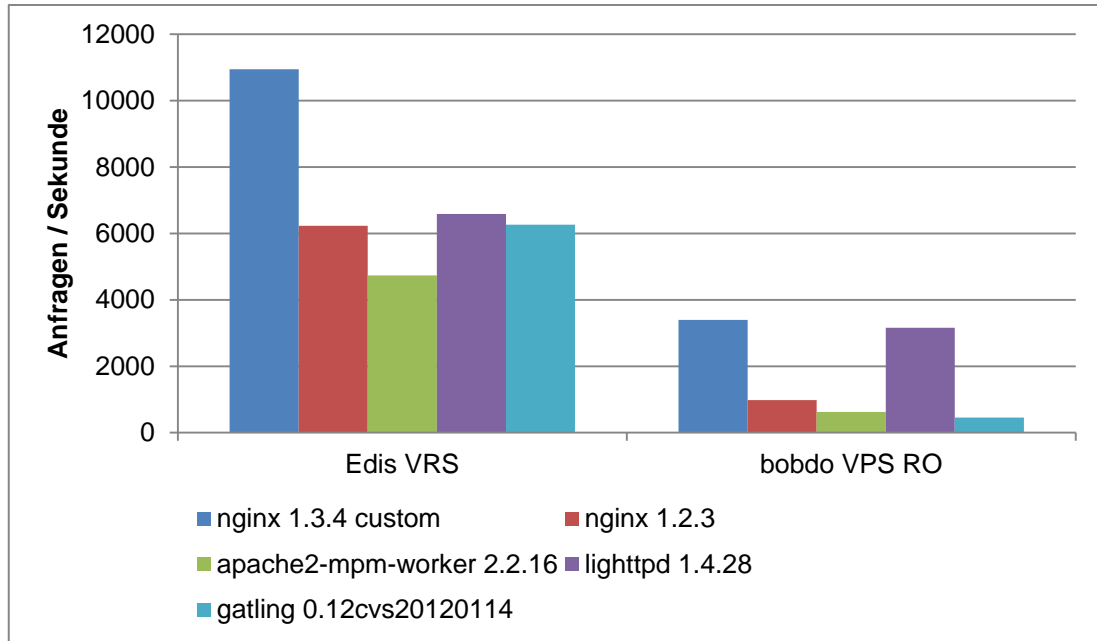


Abbildung 5 – Vergleichstest mehrerer Webserver

Wieder kam die 100 Byte große Indexdatei zum Einsatz, das finale Systemkonfigurationstuning war angewendet und dieselbe siege-Methode kam durchweg zum Einsatz. Weit abgeschlagen ist hier der Apache, dabei wurde extra die für High Performance angedachte MPM-Worker-Version verwendet. Zudem war der Ressourcenverbrauch beim Apache um ein vielfaches höher als bei allen anderen Testkandidaten. Sehr gut mithalten kann Lighty, besonders bei den Anfragen vom bobdo VPS RO. Warum hier der nginx 1.2.3 so weit abgeschlagen ist lässt sich fast nur durch Probleme in der Leitung, beim Routing oder aufgrund von Problemen im bobdo VPS RO selbst erklären. Gatling war in der Konfiguration furchtbar, hier ist definitiv noch viel Arbeit notwendig. Der Autor sagt jedoch selbst über seinen kleinen Webserver aus, dass es noch eine Beispielimplementierung ist.

¹⁰⁸ <http://www.dotdeb.org/2012/08/07/nginx-1-2-3/> – Verwendete nginx Version

Die endgültige Konfiguration kann jedoch noch auf ganz andere Weise problematisch werden. Das extrem aggressiv konfigurierte Caching vom Proxy und dem Fast-CGI-Cache kann unter Umständen dazu führen, dass Benutzer Seiten zu sehen bekommen, die eigentlich bereits abgelaufen sind. Aus Erfahrung weiß ich, dass dies Verwirrung stiften kann. Sollte das Caching also zu aggressiv für die eigene Website sein, müssen die Werte entsprechend angepasst werden.

Ein weiterer wichtiger Punkt ist die Verwendung von PHP 5.4. Diese Version der Programmiersprache ist zum gegenwärtigen Zeitpunkt noch sehr neu. Sehr viele Programme wurden noch nicht angepasst und so kam es auch im Verlauf der Arbeit zu Problemen mit dem Drupal Devel Modul. Wird PHP 5.3 verwendet muss die Konfiguration jedoch angepasst werden, da Versionen vor 5.4 noch sehr viele unsichere Optionen besitzen die es abzusichern gilt. Suhosin¹⁰⁹ wird im High Performance Kontext jedoch nicht empfohlen, da es einen unnötigen Overhead hinzufügt und viele Inkompatibilitäten einführt. Auch ist die Entwicklung 2012 etwas ins Stocken geraten.

Abschließend möchte ich noch darauf hinweisen, dass ich im Rahmen dieser Arbeit trotz des enormen Umfangs auf sehr viele interessante Dinge verzichten musste. So konnte z. B. das Thema SSL-Optimierung, oder SPDY gar nicht erst angesprochen werden. Entsprechend wäre es möglich weitere Arbeiten, basierend auf Dieser, zu Erarbeiten.

¹⁰⁹ <http://www.hardened-php.net/suhosin/> – Suhosin Website

Anhänge

Sammlung aller Konfigurationsdateien die im Verlauf der Tests verwendet wurden. Da die Lesbarkeit aufgrund der fehlenden Ordnerstruktur und ohne Syntaxhervorhebung stark leidet ist alles auch online zu finden im GitHub-Konto des Autors unter der folgenden Adresse. Die Konfigurationen bei GitHub besitzen zusätzlich noch README-Dateien die die Verwendung erklären und Versionen mit ausführlichen Kommentaren. Die Kommentare wurden hier weggelassen, da die Wortanzahl sonst viel zu stark angestiegen wäre. Die im Anhang zu findenden Konfigurationsdateien sind zu 100 %, ohne Verwendung des GitHub-Repositoriums, ausreichend die Konfiguration 1:1 zu Reproduzieren. Jedoch sind die erweiterten Konfigurationsdateien mit Kommentaren – die eben nur bei GitHub vorliegen – eigentlich Teil der Arbeit, da nur so ersichtlich wird wie viel Zeit in das Verfassen eben dieser Dateien geflossen ist und die Kommentare essentiell für das Verständnis der einzelnen Optionen sind.

<https://github.com/Fleshgrinder>

Anhang 1 sshd-Konfiguration

```
001 # /etc/ssh/sshd_config
002
003 # @see man sshd
004 # @see Pro OpenSSH [ISBN 1590594762]
005 # @see SSH, The Secure Shell [ISBN 0596000111]
006
007 AcceptEnv
008 AddressFamily inet
009 AllowAgentForwarding no
010 AllowTcpForwarding no
011 AllowUsers richard
012 AuthorizedKeysFile .ssh/id_dsa.pub
013 ChallengeResponseAuthentication no
014 Ciphers aes256-ctr,arcfour256,3des-cbc,
015 blowfish-cbc,aes256-cbc
016 ClientAliveCountMax 5
017 ClientAliveInterval 60
018 Compression delayed
019 DebianBanner no
020 GatewayPorts no
021 GSSAPIAuthentication no
022 HostbasedAuthentication no
023 HostKey /etc/ssh/ssh_host_rsa_key
024 HostKey /etc/ssh/ssh_host_dsa_key
025 ListenAddress 158.255.211.32:21976
026 LoginGraceTime 120
027 LogLevel INFO
028 MACs hmac-ripemd160,hmac-sha1
029 MaxAuthTries 4
030 MaxSessions 4
031 MaxStartups 2
032 PasswordAuthentication no
033 PermitBlacklistedKeys no
034 PermitRootLogin no
035 PermitTunnel no
036 PermitUserEnvironment no
037 PrintLastLog no
038 PrintMotd yes
039 Protocol 2
040 PubkeyAuthentication yes
041 StrictModes yes
042 Subsystem sftp /usr/lib/openssh/sftp-server
043 SyslogFacility AUTH
044 TCPKeepAlive yes
045 UseDNS no
046 UseLogin no
047 UsePAM no
048 UsePrivilegeSeparation yes
049 X11Forwarding no
```

Anhang 2 nginx-Konfiguration

```
001 # /etc/nginx/nginx.conf
002
003 pcre_jit on;
004 ssl_engine dynamic;
005 worker_processes 4;
006 worker_rlimit_nofile 3072;
007 events {
008     multi_accept on;
009     worker_connections 768;
010     use epoll;
011 }
012 http {
013     add_header X-Frame-Options DENY;
014     charset utf-8;
```

```
015 client_body_temp_path      /dev/shm/nginx/body 1 2 3;
016 client_body_timeout        65;
017 client_header_timeout       65;
018 client_max_body_size        10m;
019 default_type                 application/octet-stream;
020 index                       index.html index.php /index.php;
021 keepalive_timeout            10;
022 reset_timedout_connection    on;
023 root                         /var/www;
024 send_timeout                  65;
025 sendfile                     on;
026 server_names_hash_bucket_size 64;
027 ssl_session_cache             shared:SSL:1000m;
028 ssl_session_timeout          5m;
029 tcp_nodelay                   off;
030 tcp_nopush                    on;
031 include                      conf/*.conf;
032 include                      sites/*.conf;
033 }
```

Anhang 3 nginx init-Skript

```
001 #!/bin/sh
002
003 ### BEGIN INIT INFO
004 # Provides:      nginx
005 # Required-Start: $local_fs $remote_fs $network $syslog $named
006 # Required-Stop:  $local_fs $remote_fs $network $syslog $named
007 # Default-Start:  2 3 4 5
008 # Default-Stop:   0 1 6
009 # Short-Description: starts the nginx web server
010 # Description:     starts nginx using start-stop-daemon
011 ### END INIT INFO
012
013 PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
014 DAEMON=/usr/local/sbin/nginx
015 NAME=nginx
016 DESC=nginx
017 BODY_PATH=/dev/shm/nginx/body
018 FASTCGI_PATH=/dev/shm/nginx/fastcgi/temp
019 PROXY_PATH=/dev/shm/nginx/proxy/temp
020
021 # Include nginx defaults if available
022 if [ -f /etc/default/nginx ]; then
023     . /etc/default/nginx
024 fi
025
026 test -x $DAEMON || exit 0
027
028 set -e
029
030 . /lib/lsb/init-functions
031
032 test nginx_config() {
033     # Check if our directories exist
034     if [ ! -d "$BODY_PATH" ]; then
035         mkdir -p $BODY_PATH
036     fi
037     if [ ! -d "$FASTCGI_PATH" ]; then
038         mkdir -p $FASTCGI_PATH
039     fi
040     if [ ! -d "$PROXY_PATH" ]; then
041         mkdir -p $PROXY_PATH
042     fi
043     if $DAEMON -t $DAEMON_OPTS >/dev/null 2>&1; then
044         return 0
045     else
046         $DAEMON -t $DAEMON_OPTS
047         return $?
048     fi
049 }
```

```
049 }
050
051 case "$1" in
052     start)
053         echo -n "Starting $DESC: "
054         test_nginx_config
055         # Check if the ULIMIT is set in /etc/default/nginx
056         if [ -n "$ULIMIT" ]; then
057             # Set the ulimits
058             ulimit $ULIMIT
059         fi
060         start-stop-daemon --start --quiet --pidfile /var/run/$NAME.pid \
061             --exec $DAEMON -- $DAEMON_OPTS || true
062         echo "$NAME."
063         ;;
064
065     stop)
066         echo -n "Stopping $DESC: "
067         start-stop-daemon --stop --quiet --pidfile /var/run/$NAME.pid \
068             --exec $DAEMON || true
069         echo "$NAME."
070         ;;
071
072     restart|force-reload)
073         echo -n "Restarting $DESC: "
074         start-stop-daemon --stop --quiet --pidfile \
075             /var/run/$NAME.pid --exec $DAEMON || true
076         sleep 1
077         test_nginx_config
078         # Check if the ULIMIT is set in /etc/default/nginx
079         if [ -n "$ULIMIT" ]; then
080             # Set the ulimits
081             ulimit $ULIMIT
082         fi
083         start-stop-daemon --start --quiet --pidfile \
084             /var/run/$NAME.pid --exec $DAEMON -- $DAEMON_OPTS || true
085         echo "$NAME."
086         ;;
087
088     reload)
089         echo -n "Reloading $DESC configuration: "
090         test_nginx_config
091         start-stop-daemon --stop --signal HUP --quiet --pidfile
092             /var/run/$NAME.pid \
093             --exec $DAEMON || true
094         echo "$NAME."
095         ;;
096
097     configtest|testconfig)
098         echo -n "Testing $DESC configuration: "
099         if test_nginx_config; then
100             echo "$NAME."
101         else
102             exit $?
103         fi
104         ;;
105
106     status)
107         status_of_proc -p /var/run/$NAME.pid "$DAEMON" nginx && exit 0 || exit $?
108         ;;
109
110     *)
111         echo "Usage: $NAME {start|stop|restart|reload|force-
112             reload|status|configtest}" >&2
113         exit 1
114         ;;
115 esac
116
117 exit 0
```

Anhang 4 nginx Cachezonen-Konfiguration

```
001 # /etc/nginx/conf/cache_zones.conf
002
003 fastcgi_cache_path    /dev/shm/nginx/fastcgi
004                       levels=1:2
005                       keys_zone=fastcgi:10m
006                       max_size=1000m
007                       inactive=2h
008                       loader_threshold=2048000000
009                       loader_sleep=1
010                       loader_files=204800;
011 fastcgi_temp_path     /dev/shm/nginx/fastcgi/temp 1 2;
012
013 proxy_cache_path      /dev/shm/nginx/proxy
014                       levels=1:2
015                       keys_zone=proxy:10m
016                       max_size=1000m
017                       inactive=2h
018                       loader_threshold=2048000000
019                       loader_sleep=1
020                       loader_files=204800;
021 proxy_temp_path       /dev/shm/nginx/proxy/temp 1 2;
```

Anhang 5 nginx GZIP-Konfiguration

```
001 # /etc/nginx/conf/gzip.conf
002
003 gzip                  on;
004 gzip_buffers          32 4k;
005 gzip_comp_level       2;
006 gzip_disable          "msie6";
007 gzip_http_version     1.1;
008 gzip_min_length       1100;
009 gzip_proxied          any;
010 gzip_static           on;
011 gzip_types
012   #text/html is always compressed by HttpGzipModule
013   text/css
014   text/plain
015   application/javascript
016   application/json
017   application/rss+xml
018   application/xml
019   application/vnd.ms-fontobject
020   font/truetype
021   font/opentype
022   image/x-icon
023   image/svg+xml;
024 gzip_vary             on;
```

Anhang 6 nginx MIME-Typ-Konfiguration

```
001 # /etc/nginx/conf/mime.types
002
003 types_hash_max_size   2048;
004
005 types {
006   application/atom+xml      atom;
007   application/java-archive  jar war ear;
008   application/javascript    js;
009   application/json          json;
010   application/msword        doc;
011   application/pdf           pdf;
012   application/postscript    ps eps ai;
013   application/rtf           rtf;
014   application/vnd.ms-excel  xls;
015   application/vnd.ms-fontobject eot;
```

```
016 application/vnd.ms-powerpoint ppt;
017 application/vnd.wap.wmlc wmlc;
018 application/x-7z-compressed 7z;
019 application/x-bittorrent torrent;
020 application/x-cocoa cco;
021 application/x-font-ttf ttf ttc;
022 application/x-httpd-php-source phps;
023 application/x-java-archive-diff jardiff;
024 application/x-java-jnlp-file jnlp;
025 application/x-makeself run;
026 application/x-perl pl pm;
027 application/x-pilot prc pdb;
028 application/x-rar-compressed rar;
029 application/x-redhat-package-manager rpm;
030 application/x-sea sea;
031 application/x-shockwave-flash swf;
032 application/x-stuffit sit;
033 application/x-tcl tcl tk;
034 application/x-x509-ca-cert der pem crt;
035 application/x-xpinstall xpi;
036 application/xhtml+xml xhtml;
037 application/xml xml;
038 application/zip zip;
039 audio/midi mid midi kar;
040 audio/mpeg mp3;
041 audio/ogg oga ogg;
042 audio/x-m4a m4a;
043 audio/x-realaudio ra;
044 audio/x-wav wav;
045 font/opentype otf;
046 font/woff woff;
047 image/gif gif;
048 image/jpeg jpeg jpg;
049 image/png png;
050 image/svg+xml svg svgz;
051 image/tiff tif tiff;
052 image/vnd.wap.wbmp wbmp;
053 image/webp webp;
054 image/x-icon ico;
055 image/x-ms-bmp bmp;
056 text/cache-manifest manifest appcache;
057 text/css css;
058 text/html html htm shtml;
059 text/mathml mml;
060 text/plain txt md;
061 text/vnd.sun.j2me.app-descriptor jad;
062 text/vnd.wap.wml wml;
063 text/x-component htc;
064 text/xml rss;
065 video/3gpp 3gpp 3gp;
066 video/mp4 m4v mp4;
067 video/mpeg mpeg mpg;
068 video/ogg ogv;
069 video/quicktime mov;
070 video/webm webm;
071 video/x-flv flv;
072 video/x-mng mng;
073 video/x-ms-asf asx asf;
074 video/x-ms-wmv wmv;
075 video/x-msvideo avi;
076 }
```

Anhang 7 nginx NoCache-Konfiguration

```
001 # /etc/nginx/conf/nocache.conf
002
003 map $http_cookie $nocache {
004     default 0;
005     ~SESS 1;
006 }
```


Anhang 8 nginx Upstream-Konfiguration

```
001 # /etc/nginx/conf/upstreams.conf
002
003 upstream php {
004     server unix:/dev/shm/php-fpm-1.sock max_fails=2 fail_timeout=30s;
005     server unix:/dev/shm/php-fpm-2.sock max_fails=2 fail_timeout=30s;
006     server unix:/dev/shm/php-fpm-2.sock max_fails=2 fail_timeout=30s backup;
007 }
008
009 upstream nginx {
010     server unix:/dev/shm/nginx/1.sock max_fails=2 fail_timeout=30s;
011     server unix:/dev/shm/nginx/2.sock max_fails=2 fail_timeout=30s;
012     server unix:/dev/shm/nginx/2.sock max_fails=2 fail_timeout=30s backup;
013 }
```

Anhang 9 nginx FastCGI-Konfiguration

```
001 # /etc/nginx/sites/conf/fastcgi_pass.conf
002
003 add_header                X-FastCGI-Cache $upstream_cache_status;
004 expires                   epoch;
005 fastcgi_cache             fastcgi;
006 fastcgi_cache_bypass     $nocache;
007 fastcgi_cache_key         "$request_method@$scheme://$server_name:$server_port$uri$args";
008 fastcgi_cache_lock        on;
009 fastcgi_cache_lock_timeout 2000;
010 fastcgi_cache_use_stale   error timeout invalid_header updating
011                          http_500;
012 fastcgi_cache_valid       200 302 1m;
013 fastcgi_cache_valid       301 1440m;
014 fastcgi_cache_valid       any 5s;
015 fastcgi_ignore_headers   Cache-Control Expires;
016 fastcgi_max_temp_file_size 1m;
017 fastcgi_no_cache         $nocache;
018 fastcgi_index             index.php;
019 fastcgi_intercept_errors  on;
020 fastcgi_keep_conn        on;
021 fastcgi_next_upstream     error timeout invalid_header http_500;
022 fastcgi_read_timeout     600;
023
024 fastcgi_param CONTENT_LENGTH $content_length;
025 fastcgi_param CONTENT_TYPE $content_type;
026 fastcgi_param DOCUMENT_ROOT $document_root;
027 fastcgi_param DOCUMENT_URI $document_uri;
028 fastcgi_param GATEWAY_INTERFACE CGI/1.1;
029 fastcgi_param HTTPS $https if_not_empty;
030 fastcgi_param QUERY_STRING $query_string;
031 fastcgi_param REDIRECT_STATUS 200;
032 fastcgi_param REMOTE_ADDR $remote_addr;
033 fastcgi_param REMOTE_PORT $remote_port;
034 fastcgi_param REMOTE_USER $remote_user;
035 fastcgi_param REQUEST_METHOD $request_method;
036 fastcgi_param REQUEST_URI $request_uri;
037 fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
038 fastcgi_param SCRIPT_NAME $fastcgi_script_name;
039 fastcgi_param SERVER_ADDR $server_addr;
040 fastcgi_param SERVER_NAME $server_name;
041 fastcgi_param SERVER_PORT $server_port;
042 fastcgi_param SERVER_PROTOCOL $server_protocol;
043 fastcgi_param SERVER_SOFTWARE nginx/$nginx_version;
```

Anhang 10 nginx FastCGI-Drupal-Konfiguration

```
001 # /etc/nginx/sites/conf/fastcgi_pass_drupal.conf
002
003 include sites/conf/fastcgi_pass.conf;
004
005 fastcgi_param QUERY_STRING      "q=$uri&$args";
006 fastcgi_param SCRIPT_NAME       /index.php;
007 fastcgi_param SCRIPT_FILENAME   $document_root/index.php;
```

Anhang 11 nginx Hidden-Files-Konfiguration

```
001 # /etc/nginx/sites/conf/hidden_files.conf
002
003 location ~ /\. {
004     return 404;
005 }
```

Anhang 12 nginx PHP-Konfiguration

```
001 # /etc/nginx/sites/conf/php.conf
002
003 location / {
004     location ~* ^.+\.php$ {
005         include sites/conf/fastcgi_pass.conf;
006         fastcgi_pass php;
007     }
008     try_files $uri @php;
009 }
010
011 location @php {
012     include sites/conf/fastcgi_pass_drupal.conf;
013     fastcgi_pass php;
014 }
```

Anhang 13 nginx Proxy-Cache-Konfiguration

```
001 # /etc/nginx/sites/conf/proxy_cache.conf
002
003 add_header          X-Proxy-Cache $upstream_cache_status;
004 expires             epoch;
005 proxy_cache         proxy;
006 proxy_cache_bypass  $nocache;
007 proxy_cache_key      "$request_method@$scheme://$server_name:$server_port$uri$args";
008 proxy_cache_lock     on;
009 proxy_cache_lock_timeout 2000;
010 proxy_cache_use_stale error timeout invalid_header updating http_500;
011 proxy_cache_valid    200 302 1m;
012 proxy_cache_valid    301 1440m;
013 proxy_cache_valid    any 5s;
014 proxy_http_version   1.1;
015 proxy_ignore_headers Cache-Control Expires;
016 proxy_max_temp_file_size 1m;
017 proxy_no_cache       $nocache;
018 proxy_redirect       off;
019 proxy_set_header     Host $host;
020 proxy_set_header     X-Forwarded-For $proxy_add_x_forwarded_for;
021 proxy_set_header     X-Real-IP $remote_addr;
022 proxy_redirect       off;
```

Anhang 14 nginx SSL-Konfiguration

```
001 # /etc/nginx/sites/conf/ssl.conf
002
003 add_header          Strict-Transport-Security "max-age=262974383";
004 keepalive_timeout   70;
005 ssl_ciphers          RC4:HIGH:!aNULL:!MD5;
006 ssl_prefer_server_ciphers on;
007 ssl_protocols        SSLv3 TLSv1 TLSv1.1 TLSv1.2;
```

Anhang 15 nginx Static-Files-Konfiguration

```
001 # /etc/nginx/sites/conf/static_files.conf
002
003 location ~*
004     \.(?:bz2|css|eot|exe|gif|gz|ico|jpe?g|js|otf|png|rar|svgz?|tar|tgz|ttf|txt|woff
005     f|xml|zip)$ {
006     add_header    Cache-Control "public, must-revalidate, proxy-revalidate";
007     add_header    Pragma public;
008     expires       max;
009     try_files      $uri =404;
010 }
```

Anhang 16 nginx Static-Host-Konfiguration

```
001 # /etc/nginx/sites/static.conf
002
003 server {
004     listen      80;
005     server_name kvm.edis.bobdo.net;
006     root        /var/www/static;
007     location / {
008         include sites/conf/static_files.conf;
009     }
010 }
```

Anhang 17 nginx Drupal-Proxy-Konfiguration

```
001 # /etc/nginx/sites/drupal_proxy.conf
002
003 server {
004     listen      80;
005     root        /var/www/drupal;
006     server_name kvm.edis.bobdo.net;
007     index       index.php;
008     include     sites/conf/proxy_cache.conf;
009     location / {
010         location ~* /files/styles/ {
011             add_header Cache-Control "public, must-revalidate, proxy-revalidate";
012             add_header Pragma public;
013             expires     max;
014             try_files    $uri @proxy;
015         }
016         include sites/conf/hidden_files.conf;
017         include sites/conf/static_files.conf;
018         location ^~ /sites/default/files/private/ {
019             internal;
020         }
021         location ~* /(crossdomain|robots)\.(txt|xml)$ {
022             try_files $uri =404;
023         }
024         location ~*
025             ^.+\.(\?:.*sql|config|engine|inc|info|install|log|make|module|php|po|profile|sh
026             |theme|txt)$ {
027             return 404;
028         }
029     }
030 }
```

```
026     }
027     location ~* /(rss|sitemap)\.xml$ {
028         try_files $uri @proxy-no-args;
029     }
030     try_files $uri @proxy;
031 }
032 location @proxy {
033     proxy_pass http://nginx$uri&$args;
034 }
035 location @proxy-no-args {
036     proxy_pass http://nginx$uri;
037 }
038 }
```

Anhang 18 nginx Drupal-Backend-Konfiguration

```
001 # /etc/nginx/sites/examples/backend_servers.conf
002
003 server {
004     root    /var/www/drupal;
005     listen  unix:/dev/shm/nginx/1.sock;
006     include sites/conf/php.conf;
007 }
008
009 server {
010     root    /var/www/drupal;
011     listen  unix:/dev/shm/nginx/2.sock;
012     include sites/conf/php.conf;
013 }
014
015 server {
016     root    /var/www/drupal;
017     listen  unix:/dev/shm/nginx/3.sock;
018     include sites/conf/php.conf;
019 }
```

Anhang 19 PHP-Konfiguration

```
001 ; /etc/php5/fpm/php.ini
002
003 ; @see http://www.php.net/manual/en/ini.list.php
004
005 [PHP]
006 short_open_tag           = 0
007 disable_functions        = system, exec, passthru,
008                           shell_exec, popen, dl, phpinfo
009 disable_classes          = com, dotnet, Reflection
010 expose_php               = 0
011 memory_limit             = -1
012 realpath_cache_size      = 128K
013 realpath_cache_ttl       = 86400
014 variables_order           = GPCS
015 request_order            = GP
016 default_charset          = utf-8
017 upload_tmp_dir           = /etc/php5/tmp
018 error_reporting          = E_ALL | E_STRICT
019 display_errors           = 0
020 log_errors               = 1
021 html_errors              = 0
022 error_log                = /var/log/php5/error.log
023 enable_dl                = 0
024 max_execution_time       = 60
025 allow_url_fopen          = 0
026 date.default_latitude    = 48.208889
027 date.default_longitude   = 16.3725
028 date.timezone            = Europe/Vienna
```

```
029 session.save_path          = /etc/php5/sessions
030 session.cookie_lifetime     = 604800
031 session.cookie_httponly     = 1
032 session.entropy_file        = /dev/urandom
033 session.entropy_length      = 128
034 session.hash_function        = sha512
035 session.hash_bits_per_character = 6
```

Anhang 20 PHP-FPM-Konfiguration

```
001 ; /etc/php5/fpm/php-fpm.conf
002
003 ; @see http://php.net/manual/en/install.fpm.configuration.php
004
005 [global]
006 pid                      = /var/run/php5-fpm.pid
007 error_log                 = /var/log/php5/error.log
008 emergency_restart_threshold = 2
009 emergency_restart_interval = 5s
010 process_control_timeout   = 2s
011 events.mechanism         = epoll
012
013 [php-fpm-1]
014 listen                    = /dev/shm/php-fpm-1.sock
015 listen.backlog            = 32000
016 listen.owner              = nginx
017 listen.group              = root
018 listen.mode               = 0600
019 user                      = php5
020 group                     = webdev
021 pm                        = dynamic
022 pm.max_children           = 9
023 pm.start_servers          = 3
024 pm.min_spare_servers      = 2
025 pm.max_spare_servers      = 4
026 pm.max_requests           = 100
027 request_terminate_timeout = 60
028 request_slowlog_timeout   = 30
029 slowlog                   = /var/log/php5/slow.log
030 rlimit_files               = 308506
031 rlimit_core                = unlimited
032 security.limit_extensions = .php
033 chdir                     = /var/www
034 catch_workers_output      = 1
035
036 [php-fpm-2]
037 listen                    = /dev/shm/php-fpm-2.sock
038 listen.backlog            = 250000
039 listen.owner              = nginx
040 listen.group              = root
041 listen.mode               = 0600
042 user                      = php5
043 group                     = webdev
044 pm                        = dynamic
045 pm.max_children           = 9
046 pm.start_servers          = 3
047 pm.min_spare_servers      = 2
048 pm.max_spare_servers      = 4
049 pm.max_requests           = 100
050 request_terminate_timeout = 60
051 request_slowlog_timeout   = 30
052 slowlog                   = /var/log/php5/slow.log
053 rlimit_files               = 308506
054 rlimit_core                = unlimited
055 security.limit_extensions = .php
056 chdir                     = /var/www
057 catch_workers_output      = 1
058
059 [php-fpm-3]
060 listen                    = /dev/shm/php-fpm-3.sock
```

```
061 listen.backlog          = 250000
062 listen.owner            = nginx
063 listen.group            = root
064 listen.mode              = 0600
065 user                    = php5
066 group                   = webdev
067 pm                      = dynamic
068 pm.max_children          = 9
069 pm.start_servers         = 3
070 pm.min_spare_servers    = 2
071 pm.max_spare_servers     = 4
072 pm.max_requests         = 100
073 request_terminate_timeout = 60
074 request_slowlog_timeout  = 30
075 slowlog                  = /var/log/php5/slow.log
076 rlimit_files             = 308506
077 rlimit_core              = unlimited
078 security.limit_extensions = .php
079 chdir                    = /var/www
080 catch_workers_output     = 1
```

Anhang 21 APC-Konfiguration

```
001 ; /etc/php5/conf.d/apc.ini
002
003 ; @see http://xerosphere.net/optimizing-apc-for-drupal
004 ; @see http://www.php.net/manual/en/apc.configuration.php
005
006 extension                  = apc.so
007 apc.enabled                = 1
008 apc.shm_segments           = 1
009 apc.shm_size                = 128M
010 apc.optimization            = 0
011 apc.num_files_hint          = 1024
012 apc.user_entries_hint      = 2048
013 apc.ttl                    = 0
014 apc.user_ttl                = 0
015 apc.gc_ttl                  = 0
016 apc.cache_by_default       = 1
017 apc.filters                 =
018 apc.mmap_file_mask          = /dev/zero
019 apc.slam_defense            = 0
020 apc.file_update_protection = 0
021 apc.enable_cli              = 0
022 apc.max_file_size           = 10M
023 apc.use_request_time        = 1
024 apc.stat                    = 0
025 apc.write_lock              = 1
026 apc.report_autofilter       = 1
027 apc.include_once_override   = 0
028 apc.localcache              = 1
029 apc.localcache.size         = 512
030 apc.coredump_unmap          = 0
031 apc.stat_ctime              = 0
032 apc.preload_path            =
033 apc.file_md5                 = 0
034 apc.canonicalize            = 1
035 apc.lazy_functions           = 0
036 apc.lazy_classes            = 0
037 apc.rfc1867                 = 0
```

Anhang 22 MariaDB-Konfiguration

```
001 # /etc/mysql/my.conf
002
003 [client]
```

```
004 default_character_set      = utf8
005 socket                      = /var/run/mysqld/mysqld.sock
006
007 [mysqld_safe]
008 log_error                    = /var/log/mysql/error.log
009 nice                          = 0
010 skip_syslog                  = true
011 socket                       = /var/run/mysqld/mysqld.sock
012
013 [mysqld]
014 back_log                     = 32000
015 basedir                      = /usr
016 bulk_insert_buffer_size      = 16M
017 character_set_server         = utf8
018 collation_server             = utf8_general_ci
019 concurrent_insert            = 2
020 connect_timeout              = 5
021 datadir                      = /var/lib/mysql
022 default_storage_engine       = InnoDB
023 enable_feedback              = true
024 expire_logs_days             = 10
025 innodb_buffer_pool_size      = 256M
026 innodb_file_per_table        = true
027 innodb_flush_method          = O_DIRECT
028 innodb_flush_neighbor_pages  = cont
029 innodb_io_capacity           = 400
030 innodb_log_buffer_size       = 8M
031 innodb_open_files            = 400
032 join_buffer_size             = 4M
033 key_buffer_size              = 128M
034 large_pages                  = true
035 lc_messages                  = en_US
036 lc_messages_dir              = /usr/share/mysql
037 log_bin                      = /var/log/mysql/mariadb-bin
038 log_bin_index                 = /var/log/mysql/mariadb-bin.index
039 log_slow_rate_limit          = 1000
040 log_slow_verbosity           = query_plan
041 log_warnings                  = 2
042 long_query_time               = 10
043 max_allowed_packet           = 16M
044 max_binlog_size               = 100M
045 max_connections              = 100
046 max_heap_table_size          = 32M
047 myisam_recover_options       = BACKUP
048 myisam_sort_buffer_size      = 512M
049 pid_file                     = /var/run/mysqld/mysqld.pid
050 query_cache_limit             = 128K
051 query_cache_size              = 64M
052 query_cache_type              = DEMAND
053 read_buffer_size              = 2M
054 read_rnd_buffer_size         = 1M
055 skip_name_resolve            = true
056 skip_networking               = true
057 slow_query_log_file           = /var/log/mysql/slow.log
058 socket                        = /var/run/mysqld/mysqld.sock
059 sort_buffer_size              = 4M
060 table_cache                   = 400
061 thread_cache_size             = 128
062 tmp_table_size                = 32M
063 tmpdir                       = /tmp
064 wait_timeout                  = 600
065 user                          = mysqld
066
067 [mysqldump]
068 quick                         = true
069 quote_names                   = true
070 max_allowed_packet           = 16M
071
072 [isamchk]
073 key_buffer                    = 16M
074
075 !includedir /etc/mysql/conf.d/
```

Anhang 23 MariaDB init-Skript-Shellskript

```
001 #!/bin/bash
002
003 echo 'Creating new MariaDB init script at /etc/init.d/mariadb\n'
004 echo '#!/bin/bash
005 #
006 ### BEGIN INIT INFO
007 # Provides:          MariaDB
008 # Required-Start:    $remote_fs $syslog
009 # Required-Stop:     $remote_fs $syslog
010 # Should-Start:      $network $named $time
011 # Should-Stop:       $network $named $time
012 # Default-Start:     2 3 4 5
013 # Default-Stop:      0 1 6
014 # Short-Description: Start and stop the MariaDB database server daemon
015 # Description:       Controls the main MariaDB database server daemon
016 #                   "mysqld"
017 #                   and its wrapper script "mysqld_safe".
018 ### END INIT INFO
019
020 ulimit -l unlimited
021 sh /etc/init.d/mysql "${1:-'\''\''}'" > /etc/init.d/mariadb
022 chmod 770 /etc/init.d/mariadb
023 update-rc.d -f mysql remove
024 update-rc.d mariadb defaults
025 echo 'New init script created and added to system startup. DONE!\n'
```


Abbildungsverzeichnis

Abbildung 1 – PuTTY-Hauptfenster	32
Abbildung 2 – PuTTY-Sicherheitswarnung	33
Abbildung 3 – dash" nach erfolgreicher Authentifizierung	33
Abbildung 4 – PuTTYgen Hauptfenster	36
Abbildung 5 – Vergleichstest mehrerer Webserver	65

Quelltextverzeichnis

Quelltext 1 – Konfiguration von Hostnamen und Namensauflösung	34
Quelltext 2 – Benutzer anlegen für Anmeldung	35
Quelltext 3 – Paketquellen-Konfiguration	37
Quelltext 4 – Hinzufügen der GPG-Schlüssel	37
Quelltext 5 – Paketquellen-Prioritätskonfiguration	38
Quelltext 6 –nginx herunterladen und extrahieren.....	39
Quelltext 7 – Benötigte Bibliotheken herunterladen	40
Quelltext 8 – Klonen des Master-Zweiges des NginxUploadProgressModules.....	40
Quelltext 9 –Benutzer anlegen und nginx installieren.....	41
Quelltext 10 – nginx Installation abschließen.....	42
Quelltext 11 – Erstellung einer statischen 100-Byte-Datei.....	42
Quelltext 12 – siege-Konfiguration	43
Quelltext 13 – Ergebnis Edis VRS erster Durchlauf	44
Quelltext 14 – Ergebnis bobdo VPS RO erster Durchlauf	44
Quelltext 15 – Ergebnis localhost erster Durchlauf.....	45
Quelltext 16 – Ergebnis localhost 1.000 konkurrierende Verbindungen.....	45
Quelltext 17 – Systemkonfiguration TCP-Tuning	46
Quelltext 18 –Aktivierung TCP-Stausteueralgorithmus Vegas	47
Quelltext 19 – Anpassung der Übertragungswarteschlange für ein GBit/s	48
Quelltext 20 – Ergebnis Edis VRS zweiter Durchlauf.....	48
Quelltext 21 – Ergebnis bobdo VPS RO zweiter Durchlauf.....	48
Quelltext 22 – Ergebnis localhost zweiter Durchlauf	49
Quelltext 23 – Ergebnis Edis VRS dritter Durchlauf.....	50
Quelltext 24 – Ergebnis bobdo VPS RO dritter Durchlauf.....	50
Quelltext 25 – Ergebnis localhost dritter Durchlauf	51
Quelltext 26 – Installation von PHP und MariaDB	51
Quelltext 27 – Überprüfung ob Hugepages aktiviert sind	54
Quelltext 28 – Systemkonfiguration zur Aktivierung von Hugepages	54
Quelltext 29 – Hugepages nach korrekter Konfiguration	55
Quelltext 30 – Installation von Drupal 7 mit Drush	56

Quelltext 31 – Virtuellen Host für Drupal und Devel Modul aktivieren.....	56
Quelltext 32 – Einfaches jQuery-Skript zum Sammeln der URLs	57
Quelltext 33 – Ergebnis Edis VRS PHP-Durchlauf.....	57
Quelltext 34 – Ergebnis bobdo VPS RO PHP-Durchlauf.....	58

Literaturverzeichnis

- (1) Acquia, *Leadership*. Acquia, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.acquia.com/about-us/leadership>
- (2) António P. P. Almeida: *perusio / drupal-with-nginx*. In: *github*. GitHub Inc., 16. Mai 2012, Abgerufen am 23. August 2012 (Englisch).
Weblink: <https://github.com/perusio/drupal-with-nginx>
- (3) Ilia Alshanetsky, Rasmus Lerdorf: *Guide to PHP Security*. 1. Auflage. Marco Tabini & Associates, Inc., Toronto 2005. ISBN/ISSN: 9780973862102.
- (4) Eric Amberg: *Linux-Server mit Debian GNU/Linux*. 1. Auflage. Redline GmbH, Heidelberg 2007. ISBN/ISSN: 978-38266-1587-0,
- (5) Anon., *Download time*. Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.download-time.com/>
- (6) Apache Software Foundation., *Apache Module mod_isapi*. In: *Apache HTTP Server Documentation Version 2.2 Modules*. Apache Software Foundation., Abgerufen am 27. August 2012 (Englisch).
Weblink: http://httpd.apache.org/docs/2.2/mod/mod_isapi.html
- (7) Daniel Bachfeld: *Sicherheitsspezialist verlässt resigniert PHP-Security-Team*. In: *heise Security*. Heise Zeitschriften Verlag, 13. Dezember 2006, Abgerufen am 22. August 2012.
Weblink: <http://heise.de/-126467>
- (8) Daniel J. Barrett: Richard E. Silverman, G. R. Byrnes: *SSH, The Secure Shell: The Definitive Guide*. 2. Auflage. O'Reilly Media, Sebastopol 2005. ISBN/ISSN: 978-0-596-00895-6.
- (9) Daniel Bartholomew: Jim Galley, Krishna Prajapati: *Installing MariaDB .deb Files*. In: *AskMonty KnowledgeBase*. Monty Program AB, 22. Juli 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://kb.askmonty.org/en/installing-mariadb-deb-files/>
- (10) Jake Brutlag: *Speed Matters*. In: *Research Blog The latest news on Google Research*. Google Inc., 24. Juni 2009, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://googleresearch.blogspot.co.at/2009/06/speed-matters.html>
- (11) Jake Brutlag: *Speed Matters for Google Web Search*. Google Inc., 22. Juni 2009, Abgerufen am 22. August 2012 (PDF, Englisch).
Weblink: http://services.google.com/fh/files/blogs/google_delayexp.pdf
- (12) Angelo P. Castellani: *TCP Compound*. In: *LWN.net*. Eklektix Inc., 25. Mai 2006, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://lwn.net/Articles/185074/>
- (13) Gordon Choi: *SEO Geo Targeting*. In: *Gordon Choi's Blog*. 8. Juli 2008, Abgerufen am 22. August 2012 (Englisch).

- Weblink: <http://www.gordonchoi.com/geo-targeting-seo-strategies-for-a-single-country-20080708>
- (14) dark100: *Performance comparison of regular expression engines*. In: *sljit*. 16. Oktober 2011, Abgerufen am 22. August 2012 (Englisch).
Weblink: http://sljit.sourceforge.net/regex_perf.html
- (15) Oliver Diedrich: *MySQL-Fork vom MySQL-Gründer*. In: *heise online*. Heise Zeitschriften Verlag, 27. Februar 2009, Abgerufen am 22. August 2012.
Weblink: <http://heise.de/-201888>
- (16) Diverse: *FastCGI support for G-WAN (work in progress)*. In: *G-WAN Forum*. .
Weblink: <http://forum.gwan.com/index.php?p=/discussion/446/fastcgi-support-for-g-wan-work-in-progress/>
- (17) Diverse: *Core drush commands*. In: *Drush.org*. .
Weblink: <http://www.drush.org/help/5#core-quick-drupal>
- (18) ESnet, *Test/Measurement Host Tuning*. In: *Fasterdata Knowledge Base*. Lawrence Berkeley National Laboratory, 8. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://fasterdata.es.net/host-tuning/linux/>
- (19) Dan Farber: *Google's Marissa Mayer: Speed wins*. In: *ZDNet*. CBS Interactive, 9. November 2006, Abgerufen am 22. August 2012 (Englisch): „For Google, or any other Web site, speed means people viewing more pages in same visit and spending more time overall online...and more money.“
Weblink: <http://www.zdnet.com/blog/btl/googles-marissa-mayer-speed-wins/3925>
- (20) Jeffrey Fulmer: *Siege configuration file*. In: *Nursit Le Blog*. .
Weblink: <http://blog.nursit.net/IMG/txt/siegerc.txt>
- (21) Richard Fussenegger: *Fleshgrinder / apt*. In: *github*. GitHub Inc., 22. August 2012, Abgerufen am 22. August 2012 (Englisch, Konfigurationsdateien mit ausführlichen Kommentaren).
Weblink: <https://github.com/Fleshgrinder/apt>
- (22) Richard Fussenegger: *Fleshgrinder / mysql*. In: *github*. GitHub Inc., 22. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <https://github.com/Fleshgrinder/mysql>
- (23) Richard Fussenegger: *Fleshgrinder / nginx*. In: *github*. GitHub Inc., 22. August 2012, Abgerufen am 22. August 2012 (Englisch, Komplette nginx Konfiguration für das Testsystem).
Weblink: <https://github.com/Fleshgrinder/nginx>
- (24) Richard Fussenegger: *Fleshgrinder / nginx*. In: *github*. GitHub Inc., 22. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <https://github.com/Fleshgrinder/nginx>
- (25) Richard Fussenegger: *Fleshgrinder / php5*. In: *github*. GitHub Inc., 22. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <https://github.com/Fleshgrinder/php5>

- (26) Richard Fussenegger: *Fleshgrinder / ssh*. In: *github*. GitHub Inc., 22. August 2012, Abgerufen am 22. August 2012 (Englisch, Konfigurationsdateien mit ausführlichen Kommentaren).
Weblink: <https://github.com/Fleshgrinder/ssh>
- (27) Richard Fussenegger: *Fleshgrinder / sysctl.d*. In: *github*. GitHub Inc., 22. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <https://github.com/Fleshgrinder/sysctl.d>
- (28) Richard Fussenegger, Michael Graziano: *How to reset sysctl to it's OS defaults?*. In: *Server Fault*. .
Weblink: <http://serverfault.com/questions/417944>
- (29) Richard Fussenegger, nojak: *ufw blocking apt and dns*. In: *serverfault*. Stack Exchange Inc., 12. August 2012, Abgerufen am 22. August 2012 (Englisch): „I'd look for something like this, since it just interfaces with iptables: iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT – nojak Aug 18 at 1:28“
Weblink: <http://serverfault.com/questions/416727/ufw-blocking-apt-and-dns>
- (30) Vivek Gite: *The importance of Linux partitions*. In: *nixCraft*. nixCraft, 8. Dezember 2008, Abgerufen am 23. August 2012 (Englisch).
Weblink: <http://www.cyberciti.biz/tips/the-importance-of-linux-partitions.html>
- (31) Vivek Gite: *Linux Kernel Security (SELinux vs AppArmor vs Grsecurity)*. In: *nixCraft*. nixCraft, 18. Juni 2009, Abgerufen am 23. August 2012 (Englisch).
Weblink: <http://www.cyberciti.biz/tips/selinux-vs-apparmor-vs-grsecurity.html>
- (32) Vivek Gite: *Linux: How To Disable The IPv6 Protocol*. In: *nixCraft*. nixCraft, 10. Januar 2010, Abgerufen am 23. August 2012 (Englisch).
Weblink: <http://www.cyberciti.biz/tips/linux-how-to-disable-the-ipv6-protocol.html>
- (33) Google, Inc., *Apache SPDY module*. In: *Google Code*. Google, Inc., Abgerufen am 27. August 2012 (Englisch).
Weblink: <http://code.google.com/p/mod-spdy/>
- (34) Google, Inc., *SPDY Protocol*. In: *The Chromium Project*. Google, Inc., Abgerufen am 27. August 2012 (Englisch).
Weblink: <http://www.chromium.org/spdy/spdy-protocol>
- (35) Google, *Google Trends: Java, PHP*. In: *Google Trends*. Goggle Inc., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.google.com/trends/?q=java,+php>
- (36) Google, *Google Trends: MySQL, PostgreSQL*. In: *Google Trends*. Google Inc., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.google.com/trends/?q=mysql,+postgresql>
- (37) Google, *Google Trends: NoSQL*. In: *Google Trends*. Google Inc., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.google.com/trends/?q=nosql>

- (38) Google, *Google Trends: Wordpress, Joomla, Drupal*. In: *Google Trends*. Google Inc., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.google.com/trends/?q=wordpress,+joomla,+drupal>
- (39) Google, *Google Trends: WordPress, Joomla, Drupal, TYPO3*. In: *Google Trends*. Google Inc., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.google.com/trends/?q=wordpress,+joomla,+drupal,+typo3&geo=de>
- (40) Harvard University, *Previous Winners*. In: *Harvard Business School*. President & Fellows of Harvard College, Abgerufen am 23. August 2012 (Englisch).
Weblink: <http://www.hbs.edu/entrepreneurship/bplan/prevwinners.html#yr-2009>
- (41) Cal Henderson: *Building Scalable Web Sites*. 1. Auflage. O'Reilly Media Inc., Sebastopol 2006. ISBN/ISSN: 978-0-59-610235-7,
- (42) Jens Ihlenfeld: *Warum Google Analytics in Deutschland illegal ist*. In: *Golem.de*. Computec Media AG, 26. Januar 2011, Abgerufen am 22. August 2012.
Weblink: <http://www.golem.de/1101/80990.html>
- (43) Jim Jagielski: *Apache httpd v2.4: Hello Cloud: Buy you a drink?*. In: *ApacheCon North America (ACNA) 2011*. .
Weblink: http://people.apache.org/~jim/presos/ACNA11/Apache_httpd_cloud.pdf
- (44) Jim Jagielski: *Apache httpd v2.4: Hello Cloud: Buy you a drink?*. In: *ApacheCon North America (ACNA) 2011*. .
Weblink: http://people.apache.org/~jim/presos/ACNA11/Apache_httpd_cloud.pdf
- (45) Habibullah Jamal, Kiran Sultan: *Performance Analysis of TCP Congestion Control Algorithms*. In: *International Journal Of Computers And Communications*. .
Weblink: <http://w.naun.org/multimedia/UPress/cc/cc-27.pdf>
- (46) Heike Jurzik: *Debian GNU/Linux: Das umfassende Handbuch*. 3. Auflage. Galileo Computing, Bonn 2009. ISBN/ISSN: 978-3-8362-1386-8,
- (47) Steve Kemp: *Using iptables to rate-limit incoming connections*. In: *Debian Administration*. 17. Juli 2005, Abgerufen am 23. August 2012 (Englisch).
Weblink: <http://www.debian-administration.org/articles/187>
- (48) Kenyon, R. et al., *AptPreferences*. In: *Debian Wiki*. 24. März 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://wiki.debian.org/AptPreferences>
- (49) Christian Kirsch: *MySQL-Gründer warnt vor aktueller Version*. In: *heise online*. Heise Zeitschriften Verlag, 29. November 2008, Abgerufen am 22. August 2012.
Weblink: <http://heise.de/-218739>
- (50) Krzysztof Kliś: *Human factor*. In: *Web λ.0 - Functional programming for the Web*. .
Weblink: <http://weblambdazero.blogspot.co.at/2011/09/human-factor.html>
- (51) K-Rich, Marcos A. Costales: *What does "Limit" mean?*. In: *Launchpad Answers*. Canonical Ltd., 3. Oktober 2008, Abgerufen am 23. August 2012 (Englisch).
Weblink: <https://answers.launchpad.net/gui-ufw/+question/47059/>

- (52) Christopher Kunz, Stefan Esser: *PHP-Sicherheit*. 3. Auflage. dpunkt.verlag GmbH, Heidelberg 2008. ISBN/ISSN: 9783898645355.
- (53) Geoff Lawrence: Johann Cervantes: authenticitech Bojana Sporer: *Step 2: Create the database*. In: *Drupal Community Documentation*. 4. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://drupal.org/documentation/install/create-database>
- (54) Hugo Leisink: *Frequently Asked Questions*. In: *Hiawatha Webserver*. Abgerufen am 27. August 2012 (Englisch).
Weblink: <http://www.hiawatha-webserver.org/faq>
- (55) LKDDb, *CONFIG_TCP_CONG_BIC: Binary Increase Congestion (BIC) control*. In: *Linux Kernel Driver DataBase*. Abgerufen am 22. August 2012 (Englisch).
Weblink: http://cateee.net/lkddb/web-lkddb/TCP_CONG_BIC.html
- (56) LKDDb, *CONFIG_TCP_CONG_CUBIC: CUBIC TCP*. In: *Linux Kernel Driver DataBase*. Abgerufen am 22. August 2012 (Englisch).
Weblink: http://cateee.net/lkddb/web-lkddb/TCP_CONG_CUBIC.html
- (57) Cameron McKenzie: *The Performance Paradox of the JVM: Why More Hardware Means More Failures*. In: *TheServerSide.com*. TechTarget, 28. Juni 2010, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.theserverside.com/feature/Tackling-the-Garbage-Collection-Problem-of-the-Java-Virtual-Machine>
- (58) Stefan Mey: *Bandbreite: Österreich bloß Durchschnitt*. In: *Wirtschafts Blatt*. WirtschaftsBlatt Medien GmbH, 22. September 2011, Abgerufen am 22. August 2012.
Weblink: <http://www.wirtschaftsblatt.at/home/489368/index.do>
- (59) Microsoft, *Chapter 1: Functional Comparison of UNIX and Windows*. In: *Interoperability and Migration*. Microsoft, 31. Mai 2006, Abgerufen am 1. September 2012 (Englisch).
Weblink: <http://technet.microsoft.com/en-us/library/bb496993.aspx>
- (60) Microsoft, *Description of a hotfix that adds Compound TCP (CTCP) support to computers that are running Windows Server 2003 or a 64-bit version of Windows XP*. In: *Microsoft Support*. Microsoft Corp., 8. Oktober 2011, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://support.microsoft.com/kb/949316>
- (61) Monty Program AB, *~maria-captains/maria/5.5*. In: *Launchpad*. Monty Program AB, 22. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://bazaar.launchpad.net/~maria-captains/maria/5.5/files>
- (62) Netcraft Ltd., *August 2012 Web Server Survey*. In: *Netcraft*. .
Weblink: <http://news.netcraft.com/archives/2012/08/02/august-2012-web-server-survey.html>
- (63) Netcraft, *August 2012 Web Server Survey*. In: *Netcraft*. Netcraft Ltd., 2. August 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://news.netcraft.com/archives/2012/08/02/august-2012-web-server-survey.html>

- (64) NGINX, Inc., *README.txt*. In: *nginx.org*. NGINX, Inc., Abgerufen am 27. August 2012 (TXT, Englisch).
Weblink: <http://nginx.org/patches/spdy/README.txt>
- (65) Nginx, *Company*. In: *Nginx*. .
Weblink: <http://nginx.com/company.html>
- (66) Katherine Noyes: *Debian Linux Named Most Popular Distro for Web Servers*. In: *PCWorld*. IDG Consumer & SMB, 11. Januar 2012, Abgerufen am 22. August 2012 (Englisch).
Weblink:
http://www.pcworld.com/businesscenter/article/247845/debian_linux_named_most_popular_distro_for_web_servers.html
- (67) Oracle, *5.1. The MySQL Server*. In: *MySQL 5.5 Manual*. Oracle Corp., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://dev.mysql.com/doc/refman/5.5/en/mysqld-server.html>
- (68) Oracle, *8.11.4.2. Enabling Large Page Support*. In: *MySQL Documentation*. Oracle Corp., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://dev.mysql.com/doc/refman/5.5/en/large-page-support.html>
- (69) Percona, *Support for MySQL*. In: *Percona*. Percona Inc., Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.percona.com/mysql-support>
- (70) Franklin Piat: Geoff Simmons, Ingo Müller: *Hugepages*. In: *Debian Wiki*. 21. Juni 2011, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://wiki.debian.org/Hugepages>
- (71) David Planella: *SSH/OpenSSH/Keys*. In: *Ubuntu Documentation*. 20. März 2011, Abgerufen am 22. August 2012 (Englisch): „My computer - a perfectly ordinary desktop PC - had over 4,000 attempts to guess my password and almost 2,500 break-in attempts in the last week alone. How many thousand random guesses do you think it will take before an attacker stumbles across your pass“
Weblink: <https://help.ubuntu.com/community/SSH/OpenSSH/Keys>
- (72) Guillaume Plessis: *Instructions*. In: *Dotdeb: The repository for Debian-based LAMP servers*. Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.dotdeb.org/instructions/>
- (73) Q-Success, *Usage of content management systems for websites*. In: *W³Techs*. Q-Success DI Gelbmann GmbH (Co. Ltd.), Abgerufen am 22. August 2012 (Englisch).
Weblink: http://w3techs.com/technologies/overview/content_management/all
- (74) Q-Success, *Usage of operating systems for websites*. In: *W³Techs*. Q-Success DI Gelbmann GmbH (Co. Ltd.), Abgerufen am 22. August 2012 (Englisch).
Weblink: http://w3techs.com/technologies/overview/operating_system/all

- (75) Q-Success, *Usage of server-side programming languages for websites*. In: *W³Techs*. Q-Success DI Gelbmann GmbH (Co. Ltd.), Abgerufen am 22. August 2012 (Englisch).
Weblink: http://w3techs.com/technologies/overview/programming_language/all
- (76) Q-Success, *Usage statistics and market share of Linux for websites*. In: *W³Techs*. Q-Success DI Gelbmann GmbH (Co. Ltd.), Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://w3techs.com/technologies/details/os-linux/all/all>
- (77) Stefan „. Rusu: *Async frameworks “Hello World” showdown*. In: *printf(" SaltwaterC ")*; 12. Mai 2012, Abgerufen am 23. August 2012 (Englisch).
Weblink: <http://www.saltwaterc.eu/async-frameworks-hello-world-showdown.html>
- (78) Achim Sawall: *Facebook verarbeitet täglich 500 TByte neue Daten*. In: *News*. Golem.de, 23. August 2012, Abgerufen am 23. August 2012: „Jeden Tag [...] 2,7 Milliarden Mal der Like-Button an- oder ausgeschaltet.“
Weblink: <http://www.golem.de/1208/94046.html>
- (79) Steve Souders, Nate Koechley: *High Performance Web Sites: 14 Steps to Faster-Loading Web Sites*. 1. Auflage. O'Reilly Media Inc., Sebastopol 2007. ISBN/ISSN: 978-0-596-52930-7,
- (80) SpringSource, *Support for Apache HTTPD*. In: *SpringSource*. .
Weblink: <http://www.springsource.com/support/apachesupport/httpd>
- (81) Murari Sridharan: Kun Tan: Jingmin Song, Qian Zhang: *A Compound TCP Approach for High-speed and Long Distance Networks*. In: *Microsoft Research*. Microsoft Corp., 24. Juni 2005, Abgerufen am 22. August 2012 (PDF, Englisch).
Weblink: <ftp://ftp.research.microsoft.com/pub/tr/TR-2005-86.pdf>
- (82) Michael Stahnke: *Pro OpenSSH*. 1. Auflage. Apress, Berkeley 2006. ISBN/ISSN: 1-59059-476-2,
- (83) tag, et al., *HTTPS setup*. In: *G-WAN Forum*. Abgerufen am 27. August 2012 (Englisch).
Weblink: <http://forum.gwan.com/index.php?p=/discussion/209/https-setup/p1>
- (84) The Apache Software Foundation, *FAQ*. In: *Httpd Wiki*. The Apache Software Foundation, Abgerufen am 27. August 2012 (Englisch).
Weblink: http://wiki.apache.org/httpd/FAQ#Why_the_name_22Apache.22.3F
- (85) Sufyan bin Uzayr: *Winner Takes All: WordPress vs Drupal vs Joomla!*. In: *LINUX For You*. 9. Mai 2012, Abgerufen am 22. August 2012 (Englisch): „If security is a concern, Drupal or even Joomla! would be preferred. Along similar lines, if your website is of gigantic proportions (say, over 15,000 pages), Drupal can easily offer the robustness that you need.“
Weblink: <http://www.linuxforu.com/2012/05/winner-takes-all-wordpress-vs-drupal-vs-joomla/>
- (86) Gite Vivek: *Linux HugeTLBfs: Improve MySQL Database Application Performance*. In: *nixCraft*. 20. Mai 2009, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.cyberciti.biz/tips/linux-hugetlbfs-and-mysql-performance.html>
- (87) Dj Walker-Morgan: *NoSQL im Überblick*. In: *heise open*. Heise Zeitschriften Verlag, 1. Juni 2010, Abgerufen am 22. August 2012.
Weblink: <http://heise.de/-1012483>

- (88) Tobias Wassermann: *Sichere Webanwendungen mit PHP*. 1. Auflage. REDLINE GmbH, Heidelberg 2007. ISBN/ISSN: 9783826617546.
- (89) waza123a, spaam: *SPDY Support*. In: *Lighttpd Development Forum*. 18. Juni 2012, Abgerufen am 27. August 2012 (Englisch): „Lighttpd 1.x will not support SPDY.“
Weblink: <http://redmine.lighttpd.net/boards/3/topics/5201>
- (90) Website Optimization, *The Psychology of Web Performance*. Website Optimization LLC, 30. Mai 2008, Abgerufen am 22. August 2012 (Englisch).
Weblink: <http://www.websiteoptimization.com/speed/tweak/psychology-web-performance/>
- (91) Zend, *Zend Kunden*. In: *Zend*. Zend Technologies Inc., Abgerufen am 22. August 2012.
Weblink: <http://www.zend.com/de/company/customers/>