

Hochschule für Telekommunikation Leipzig (FH)
Institut für duales Studium und Wissenstransfer

Abschlussarbeit zur Erlangung des akademischen Grades

Bachelor of Engineering

Thema: **Untersuchung der Integration des Konferenzsystems
BigBlueButton in die Infrastruktur der Hochschule für
Telekommunikation Leipzig**

Vorgelegt von: **Udo Siebert**

geboren am: **04.02.1982**
in: **Leipzig**

Erstprüfer: **Prof. Dr. Thomas Meier**

Zweitprüfer: **Dipl.-Inf. (FH) Tobias Welz**

Vorwort

Bachelorarbeit zum Thema:

Untersuchung der Integration des Konferenzsystems BigBlueButton in die Infrastruktur der Hochschule für Telekommunikation Leipzig

In dieser Arbeit werden die Möglichkeiten der Einbeziehung des Open Source Online Konferenzsystems BigBlueButton (BBB) in die E-Learning-Infrastruktur der Hochschule für Telekommunikation Leipzig (HfTL) untersucht. Im Fokus stehen dabei die technische Integration sowie eine Wirtschaftlichkeitsanalyse unter Einbeziehung kommerzieller Software.

Angestrebgt wird die Einbindung des Zuganges zum Tutorium in die bisher genutzte Lernplattform ILIAS ohne weitere vom Benutzer durchzuführende Authentifizierungsschritte. Der Teilnehmer der Onlinekonferenz benutzt ein ihm bekanntes Konzept und gelangt direkt in sein Tutorium.

Bisher müssen die Teilnehmer dafür unterschiedliche Plattformen aufsuchen und verschiedene Authentifizierungsschritte durchlaufen. Nicht alle Plattformen werden dabei von der HfTL in Eigenverantwortung administriert.

Die dazu genutzte kommerzielle Lösung Saba Centra® eClassroom bietet keine von der Hochschule nutzbaren Schnittstellen. Zusätzlich gibt es je nach Studienart unterschiedliche Zugangswege mit verschiedenen Authentifizierungssystemen und für jeden Weg mehrere Accounts pro Benutzer, deren Sicherheitsrichtlinien zusätzliche Barrieren bilden.

Ein Teil der Arbeit beschäftigt sich mit der Performance- und Skalierungsanalyse von BBB. Aus dieser erfolgt eine Abschätzung der erforderlichen Rechenleistung der Hardware für bestimmte Teilnehmermengen. Die Kosten werden vergleichbaren kommerziellen Systemen gegenübergestellt.

Danksagung

Herzlich bedanken möchte ich mich bei Herrn Professor Dr. Thomas Meier für die Unterstützung bei der Themenfindung und Strukturierung der Aufgabenstellung. Weiterhin bedanken möchte ich mich bei Herrn Marcel Potschin für seinen geduldigen Support im Rechenzentrum der Hochschule und bei Herrn Peter Tempel für die Mitwirkung beim Einsatz der Computerpools.

Mein besonderer Dank gilt Herrn Tobias Welz für seine fachlichen Hinweise und die Anregungen bei der inhaltlichen Umsetzung des Themas.

Inhaltsverzeichnis

1	Abkürzungsverzeichnis.....	5
2	Abbildungsverzeichnis.....	7
3	Einleitung.....	8
4	Aktueller Stand der Implementierungen von BigBlueButton.....	10
5	Angewandte Forschungsmethoden und Ergebnisse.....	12
6	BigBlueButton	14
6.1	Historie	14
6.2	Funktionen	14
6.3	Architektur.....	18
7	Komponenten von BigBlueButton.....	19
7.1	Nginx – Webserver.....	19
7.2	Red5 – Flash Media Server	19
7.3	FreeSWITCH – Echtzeit Audiomixer	20
8	Verwendete Protokolle und Dienste	21
8.1	RTMP – Real Time Media Protocol.....	21
8.2	Ruby – Skript Sprache	21
9	Verwendete Linux Distributionen.....	23
9.1	Ubuntu Server 10.04 LTS & 12.04 LTS	23
9.2	SUSE Linux Enterprise 11 SP2	25
10	ILIAS – Open-Source Lernplattform.....	27
11	Erweiterung des BigBlueButton Plug-In für ILIAS	29
11.1	BBB-API	29
11.2	Funktionen	31
11.3	Vorgehen	31
12	Last- und Skalierungstests	36
12.1	Vorgehen	36
12.2	Benutzte Tools.....	38
12.3	Ergebnisse.....	41
12.4	Aufgetretene Probleme	45

13 Wirtschaftlichkeitsanalyse	47
13.1 Aufstellung Kosten BigBlueButton.....	47
13.2 Vergleich mit Saba Centra® eClassroom.....	48
13.3 Vergleich mit Adobe Connect in House Hosting	48
13.4 Fazit	49
14 Literatur-/Quellenverzeichnis	50
Selbstständigkeitserklärung.....	52

1 Abkürzungsverzeichnis

AAC	Advanced Audio Coding
AMF	Action Message Format
API	Application Programming Interface (Programmierschnittstelle)
BBB	BigBlueButton
CPU	Central Processing Unit
dpkg	Debian GNU/Linux package manager
DT AG	Deutsche Telekom AG
GPL	General Public License (freie Software-Lizenz)
HfTL	Hochschule für Telekommunikation Leipzig
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ILIAS	Integriertes Lern-, Informations- und Arbeitskooperations-System
JPEG	Joint Photographic Experts Group (Dateiendung für Bilddateien)
KVM	Kernel-based Virtual Machine
LTS	Long Term Support
PC	Personal Computer
PDF	Portable Document Format (Dateiendung für Dokument-Dateien)
PHP	Hypertext Preprocessor (ursprünglich: Personal Home Page Tools)
PNG	Portable Network Graphics (Dateiendung für Grafik-Dateien)
PPA	Personal Package Archive
RBAC	Role Based Access Control

RC	Release Candidate
RPM	RPM Package Manager
RTMP	Real Time Messaging Protocol
SIP	Session Initiation Protocol
SP2	Service Pack 2
SSL	Secure Sockets Layer (Netzwerkprotokoll)
SWF	Shockwave Flash (Dateiendung für abspielbare Adobe-Flash-Animationen)
TCP	Transmission Control Protocol
VM	Virtuelle Maschine
VoIP	Voice over IP
XML	Extensible Markup Language (Dateiendung für Textdateien)

2 Abbildungsverzeichnis

Abbildung 4-1 Stresstest der BBB-Entwickler an einem 8-Kern-System [1].....	10
Abbildung 6-1 Interface des BBB Flash-Clients.....	16
Abbildung 6-2 Werkzeugleiste mit Pointer.....	17
Abbildung 6-3 Komponenten von BBB und deren Zusammenwirken[6]	18
Abbildung 10-1 ILIAS Objektliste mit BBB	28
Abbildung 11-1 BBB Objekt - Startmaske	29
Abbildung 11-2 BBB-Objekt Raum erstellt (Moderatoransicht)	33
Abbildung 11-3 BBB-Objekt Raum beendet	33
Abbildung 12-1 Notepad++ Replacement-Dialog	40
Abbildung 12-2 OriginPro 9 Importergebnis	41
Abbildung 12-3 Lasttest mit 4 CPU Kernen	42
Abbildung 12-4 Lasttest mit 12 CPU Kernen	43
Abbildung 12-5 Lasttest mit 6 CPU Kernen	44
Abbildung 12-6 Lasttest mit 6 CPU Kernen (Bandbreite)	45

3 Einleitung

Ein wichtiger Bestandteil des Lehrszenarios für duale und berufsbegleitende Studiengänge an der Hochschule für Telekommunikation Leipzig (HfTL) sind die online durchgeführten Tutorien. Als Basis wird bisher die kommerzielle Lösung Saba Centra® eClassroom genutzt, die einen Klassenraum mit Whiteboard simuliert.

Die mehrseitige Kommunikation erfolgt per Audio- und Videostream und ähnelt einem Telefonat. Zusätzlich werden vorbereitete Materialien auf dem für alle Teilnehmer sichtbaren Whiteboard präsentiert. Die Tutorium-Sitzungen dienen als Ersatz der Übungsstunden eines klassischen Studiums, zur Festigung des Lernstoffes, zum Austausch von eventuell auftretenden Fragen und zur Vermittlung neuen Wissens. Sie sind fest im Studienablauf integriert und finden zu vorher geplanten Terminen statt.

Damit Studierende an ihren Tutorien teilnehmen können, müssen sie sich bisher über verschiedene Plattformen authentifizieren, die sich zusätzlich nach Studienform unterscheiden. Die Sitzungen der dualen Studierenden sind ausschließlich über Teltec¹ und die der berufsbegleitenden Studierenden über eClassroom-2.telekom.de zu erreichen. Zusätzlich nutzen die Studenten die Lernplattform ILIAS und können so auf die Vielzahl der Lernunterlagen, welche asynchron ausgetauscht werden, zugreifen.

ILIAS wird direkt vom Rechenzentrum der HfTL administriert. Accounts für die anderen Plattformen müssen beauftragt werden. Dies bedeutet mehrfache Datenhaltung mit allen damit verbundenen Schwierigkeiten. Zudem stören die verschiedenen Plattformen den einheitlichen Onlineauftritt der Hochschule.

Saba Centra® eClassroom ist ein durch die Deutsche Telekom AG eingekauftes Produkt, welches an die HfTL kostenpflichtig weiter lizenziert wird. Die dabei eingesetzten finanziellen Mittel sollen zukünftig anderen Projekten zur Verfügung stehen. Aus diesen Gründen wurde seit dem 4. Quartal 2012 nach einer Alternative gesucht.

Als Ergebnis intensiver Gespräche und Recherchen fiel die Wahl auf das Open Source online Konferenz System BigBlueButton.

Der Funktionsumfang von BigBlueButton gleicht schon jetzt in vielen Punkten dem von Saba Centra® eClassroom. So können alle Teilnehmer mittels Audio- oder Videostream miteinander kommunizieren, Dokumente, Texte und Zeichnungen über das Whiteboard austauschen und die Anzeige ihres Bildschirmes als Videostream freigeben.

Diese Arbeit beschäftigt sich mit der Integration von BigBlueButton in die Infrastruktur der HfTL sowie der Analyse der Leistungsfähigkeit und den damit verbundenen Anforderungen an die technische Infrastruktur.

Am Anfang steht das Deployment von BigBlueButton auf einem neueren Betriebssystem auf einem Testserver im Rechenzentrum der HfTL. Anschließend wird das bestehende ILIAS

¹ www.ttp.telekom.de

Plug-In an die neue API angepasst, um damit einen einfacheren Zugang zu den Tutorien über die Lernplattform zu ermöglichen.

Das neue Konferenzsystem ist damit ein zusätzliches ILIAS Objekt und kann innerhalb bereits bekannter Benutzer Interfaces verwendet werden.

Es existiert keine weitere Plattform, und es sind auch keine zusätzlichen Authentifizierungsschritte erforderlich, um an Sitzungen teilzunehmen.

Ein Bestandteil der Arbeit ist die Leistungs- und Skalierungsanalyse von BigBlueButton. Es wird untersucht, wie hoch der Ressourcenverbrauch bei größeren Teilnehmerzahlen auf einem Server ist, wie hoch der Bandbreitenbedarf ist und wie gut die Software auf Mehrkernsystemen arbeitet. Mit Hilfe der gewonnenen Kennzahlen ist ein Kostenvergleich mit kommerziellen Anbietern möglich.

4 Aktueller Stand der Implementierungen von BigBlueButton

Es existieren wenige fundierte Informationen zur Leistungsfähigkeit des Open Source online Konferenz System BigBlueButton (BBB).

Einige dieser Informationen befinden sich auf der Homepage des Projektes (Internetauftritt des Entwickler-Teams). Sie enthalten jedoch keine Angaben zur verwendeten Hardware. Der letzte Eintrag stammt aus dem März 2011² und enthält nur eine schlichte Angabe zur gleichzeitigen Verbindung von 60 Benutzern auf einem Server.

Etwas genauer ist der Eintrag in den Google Groups BigBlueButton-dev vom 24.11.2011. Dort wurde ein dokumentierter Stresstest von BBB internal Build Version 0.8 mit maximal 80 verbundenen Clients auf einem 8-Kern Xeon System durchgeführt [1]. Zusätzlich wurden 20 Webcams erfolgreich mit 60 Clients geteilt. Nach dem Zuschalten der Desktopfreigabe stürzte das System ab und der Test endete. In der Dokumentation befindet sich auch eine Angabe zur benötigten Bandbreite von etwa 60 kbit/s je verbunden Client sowie die verwendeten Versionen der Kernkomponenten FreeSWITCH (1.0.6) und red5 (0.91).

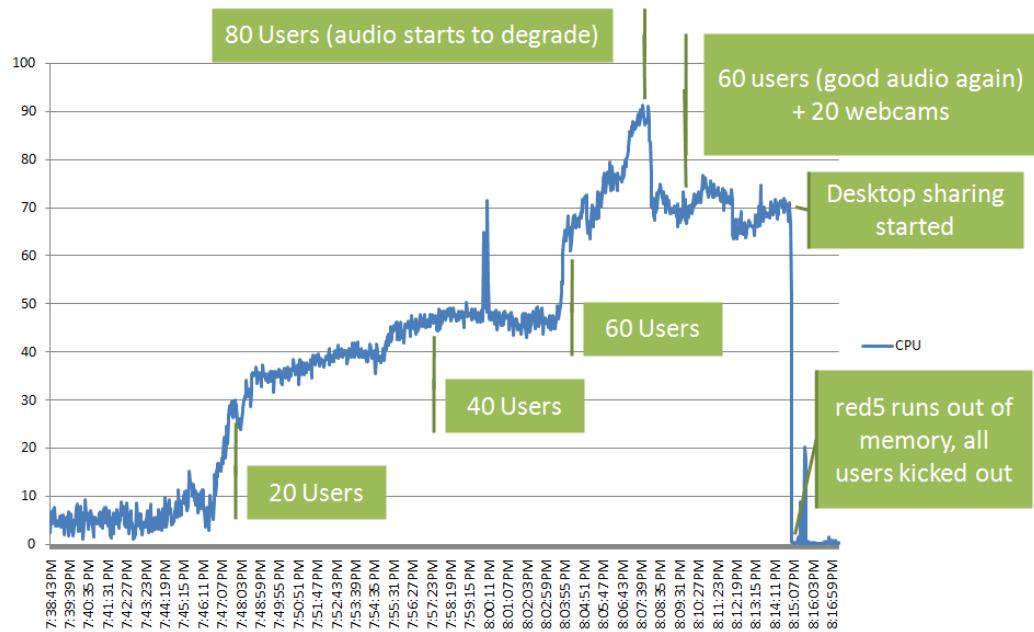


Abbildung 4-1 Stresstest der BBB-Entwickler an einem 8-Kern-System [1]

Im November 2013 erschien eine neuere Version von BBB (v 0.81) mit aktualisierten Komponenten.

Für das Deployment auf weitere Betriebssysteme als das von den Entwicklern unterstützte Ubuntu Server 10.04 LTS existieren Hinweise und knappe Anleitungen im Netz. Jedoch für

² <http://bigbluebutton.org/tag/testing/>

die aktuelle Linux-Version mit den notwendigen Paketabhängigkeiten sind nur wenige Informationen zu finden.

Die zutreffendste Information findet sich in einem Blog [2] mit einem Eintrag und beschreibt die Vorgehensweise eines Deployments auf openSUSE 11.4. In dieser Notiz wird die Installation der notwendigen Pakete aus den Quellen beschrieben. Die Installation der Kernkomponenten bezieht sich jedoch auf nicht mehr verfügbare Installationsskripte (bbb-setup). Diese Skripte sind in keiner aktuellen Version von BBB enthalten. Auch die Beschreibung der Installation aus den Quellen wurde von den BBB-Entwicklern nach Version 0.71a nicht erneuert. Sie ist im Wiki als veraltet gekennzeichnet [3].

Für das Deployment auf SUSE Linux Enterprise Server 11 SP2 gibt es keine ähnliche Beschreibung. Der Großteil der für BBB erforderlichen Pakete/Bibliotheken ist nicht über das zugehörige Paketmanagementsystem YaST2 verfügbar.

5 Angewandte Forschungsmethoden und Ergebnisse

Auf einer vom Rechenzentrum der HfTL bereitgestellten virtuellen Maschine, auf welcher SUSE Linux Enterprise Server 11 SP2 vorinstalliert ist, wird das Deployment von BigBlueButton untersucht. Dazu werden alle erforderlichen Pakete mit ihren Abhängigkeiten kompiliert und installiert. Die durch diesen Prozess entstandenen RPM-Pakete vereinfachen ein Deployment auf weiteren ähnlichen Systemen.

Der so entstandene Prototyp soll anschließend mit Hilfe von Lastsimulation durch Rechner der Computerpools auf seine Skalierbarkeit untersucht werden. Dafür sind der virtuellen Maschine unterschiedliche Ressourcen, wie verschiedene Anzahl von Rechenkernen und unterschiedliche Hauptspeichermengen, zur Verfügung zu stellen. Mit Hilfe von PCs aus den Computerpools wird eine Vielzahl von Clients simuliert. Die Audio- und Sprachqualität wird dabei überprüft.

Ziel ist es, die maximale Anzahl verbindbarer Clients bei den zur Verfügung stehenden Ressourcen zu ermitteln. Dabei dürfen keine Unterbrechungen bei der Audiomeldung oder anderweitige Beeinträchtigungen entstehen.

Für diese Arbeit wurden die Methoden des Prototyping und der Simulation eingesetzt. Prototyping oder auch Prototypenbau ermöglicht, sehr zügig erste Ergebnisse zu erzielen und so ein frühzeitiges Feedback zu erhalten. Probleme werden frühzeitig erkannt, und Anpassungen können mit geringem zeitlichem Verzug vorgenommen werden. Der entstandene experimentelle Prototyp dient dem Sammeln weiterer Erfahrungen und ist die Basis für die folgende eigentliche Entwicklung.

Unter Simulation verstehen wir in der heutigen Zeit das Nachempfinden des Verhaltens einer technischen Lösung, eines Systems oder eines Prozesses. Sie wird genutzt, wenn eine Erprobung am Original aus bestimmten Gründen nur mit erhöhtem Aufwand oder nicht kalkulierbaren Risiken möglich ist. Die Simulation ist eine experimentelle Lösungsmethode und kann das Verhalten des Originale oft in entscheidenden Punkten nachbilden. Aus den Ergebnissen einer Simulation lassen sich deshalb häufig wesentliche Erkenntnisse für das weitere Vorgehen ableiten.

Die beiden Forschungsmethoden wurden gewählt, um in der für die Arbeit zur Verfügung stehenden Zeit von einem Monat Ergebnisse zu erreichen, welche im Anschluss die für die Wirtschaftlichkeitsuntersuchungen erforderlichen Hardwarevoraussetzungen ergeben.

Ein lauffähiger BigBlueButton Prototyp wurde erfolgreich auf einem virtualisierten Testserver im Rechenzentrum der HfTL deployed.

Die erste genutzte Distribution war das von den Entwicklern vorgeschlagene Betriebssystem Ubuntu Server 10.04 LTS. Danach wurde das BBB Plug-In [4] in eine Kopie des Livesystems von ILIAS integriert und getestet. Nach einigen Konfigurationsanpassungen funktionierte der Zugang zur Tutorium-Sitzung fehlerfrei.

Anschließend wurde die Ubuntu Distribution auf einen aktuellen Stand – Ubuntu Server 12.04 LTS – aktualisiert. BBB funktionierte in der Version 0.80 fast fehlerfrei. Einzig der headless Server Dienst von Apache Open Office, welcher zum Konvertieren von Office Dokumenten genutzt wird, verursachte beim Serverstart Fehler, welche in dieser Version nicht behoben werden konnten. Ein Neustart des Dienstes nach jedem Start des Systems konnte zumindest die Funktionalität sicherstellen.

Im nächsten Schritt erfolgte ein Deployment auf einer vom Rechenzentrum gewünschten Distribution. Hierfür wurde ein neues SUSE Linux Enterprise 11 SP2 aufgesetzt und alle für BigBlueButton erforderlichen Pakete zusammengestellt, auf dem System kompiliert sowie installiert. Nach etwa vier Wochen Arbeit wurde das Projekt eingestellt, da der entstandene Aufwand sich als zu hoch für eine weitere Pflege der Installation nach Abschluss der Arbeit herausstellte.

Aus diesen Gründen wurde auf die vorherige Ubuntu-Installation zurückgegriffen und die dort installierte BBB-Version auf die im Juni 2013 veröffentlichte 0.81 beta Version aktualisiert. Diese Version bietet deutlich erweiterte API-Schnittstellen und eine Integrationsmöglichkeit der Aufzeichnungen in ILIAS.

6 BigBlueButton

6.1 Historie

Im Herbst 2007 wurde von Richard Alam, einem Masterstudenten des Technology and Innovation Management Programms, eine Version im Rahmen seines Studiums an der Carleton University Ottawa in Kanada entwickelt. Er suchte eine günstige Alternative, um mit vielen dezentral verteilten Studierenden einen virtuellen Klassenraum zu bilden und dort zu kommunizieren.

Diese von Alam entwickelte Version ersetzte daraufhin die bisher genutzte kommerzielle Lösung, Fernstudenten die Teilnahme an diesem Programm zu ermöglichen. Ein Jahr später bildete sich eine erste Entwicklergruppe mit dem Ziel, das Projekt weiter zu entwickeln und so einen kommerziellen Erfolg zu erzielen.

Das Produkt wurde seitdem von einer aktiven Community weiterentwickelt und basiert auf dem Real Time Messaging Protokoll von Adobe sowie dem Open Source Streaming Server Red5, welcher durch Reverse Engineering entwickelt wurde.

Weitere wichtige Komponenten des Systems sind der Streammixer FreeSWITCH, Adobe Flash und der Apache Tomcat Webserver. Als Entwicklungsumgebung wird bisher Ubuntu Server 10.04 (Lucid Lynx) genutzt.

Nach einem Codereview im Sommer 2009 wurde die erste Version auf Google Code veröffentlicht. Durch die Veröffentlichung wurde das Projekt weiterentwickelt und es entstanden weitere Features. Auf Desktop Sharing folgte eine API, VoIP Nutzung, ein Whiteboard und eine Aufzeichnungsfunktion.

In der letzten Betaversion vom Juni 2013 wurde zudem die Aufzeichnungsfunktion stark verbessert und bietet jetzt HTML5 Support für die Ausgabe. Am 6. September 2013 erschien die für diese Arbeit genutzte Version 0.81-RC. [5]

6.2 Funktionen

Das Open Source online Konferenz System BigBlueButton (BBB) ermöglicht den verbundenen Nutzern, in einem virtuellen Klassenraum Dokumente zu teilen und mittels Audio, Video und Text-Chat zu kommunizieren.

BBB bietet in der Version 0.81 Audio- und Videokonferenzen, deren Teilnehmeranzahl nur durch die Rechenleistung oder die Bandbreite des Servers beschränkt ist. Der Audiostream wird dabei mittels Speex Codec durch den Flash Client im Browser komprimiert und dann an den BBB-Server gesendet. Dort werden alle eingehenden Audiostreams in Echtzeit durch FreeSWITCH in einen Stream gemischt und anschließend an alle Teilnehmer per Broadcast gesendet. Die Qualität des Streams wird durch eine Konfigurationsdatei auf dem Server ge-

steuert³. In der Standardinstallation nutzt BBB das wideband Profil mit einer Abtastrate von 16 kHz. Das aufgenommene Videobild wird mittels h264 Codec komprimiert, an den Server übertragen und dort ohne weitere Verarbeitung an alle verbunden Clients des Raumes per Broadcast versendet. Der bzw. die sprechenden Teilnehmer werden mit einem grünen Rahmen im Videofenster hervorgehoben.

Die Auflösungsqualität der Videoübertragung ist im Client einstellbar und reicht von 320 x 240 px bis 1280 x 720 px. Durch die Bandbreite des Servers wird die Anzahl der Videostreams und deren Qualität maßgeblich bestimmt. Die Anzahl der verbundenen Clients ist ein weiterer Einflussfaktor. Wenn nicht genügend Übertragungskapazität zur Verfügung steht, verringert sich die Bildwiederholrate der Videostreams, um den Audiostream weiterhin unterbrechungsfrei zu übermitteln.

Im Teilnehmerfenster werden alle Teilnehmer mit ihren freigegebenen Kommunikationsmitteln – Audio und Video – angezeigt. Die Audioverbindung ist permanent aktiv, aber jeder Teilnehmer kann sich selbst stumm schalten. Zusätzlich können die Moderatoren des Raumes alle Teilnehmer stumm schalten oder alle Videostreams unterbrechen.

BBB ermöglicht die gemeinsame Betrachtung vorbereiteter Präsentationen. Diese Dokumente müssen auf den Server über den im Client integrierten Dialog hochgeladen werden.

BBB akzeptiert viele Dokumentformate wie z.B. von Microsoft Office, Libre/Openoffice und auch PDF-Dokumente sowie Bilder im JPEG- und PNG-Format. Anschließend wandelt der Server die Datei in SWF-Dokumentseiten, die vom Flashclient angezeigt werden können. Dort kann der Moderator einzelne Seiten anwählen und bis zu 400% vergrößern. Die Vergrößerung ist bei allen Teilnehmern sichtbar. Zusätzlich wird ein roter Punkt – ähnlich eines Laserpunktes – an der Spitze des Mauszeigers des Moderators für alle Teilnehmer sichtbar angezeigt, sobald dieser sich über dem Dokument befindet. Mehrere Dokumente lassen sich nur einzeln über den Importdialog auswählen.

³ opt/freeswitch/conf/autoload_configs/conference.conf.xml

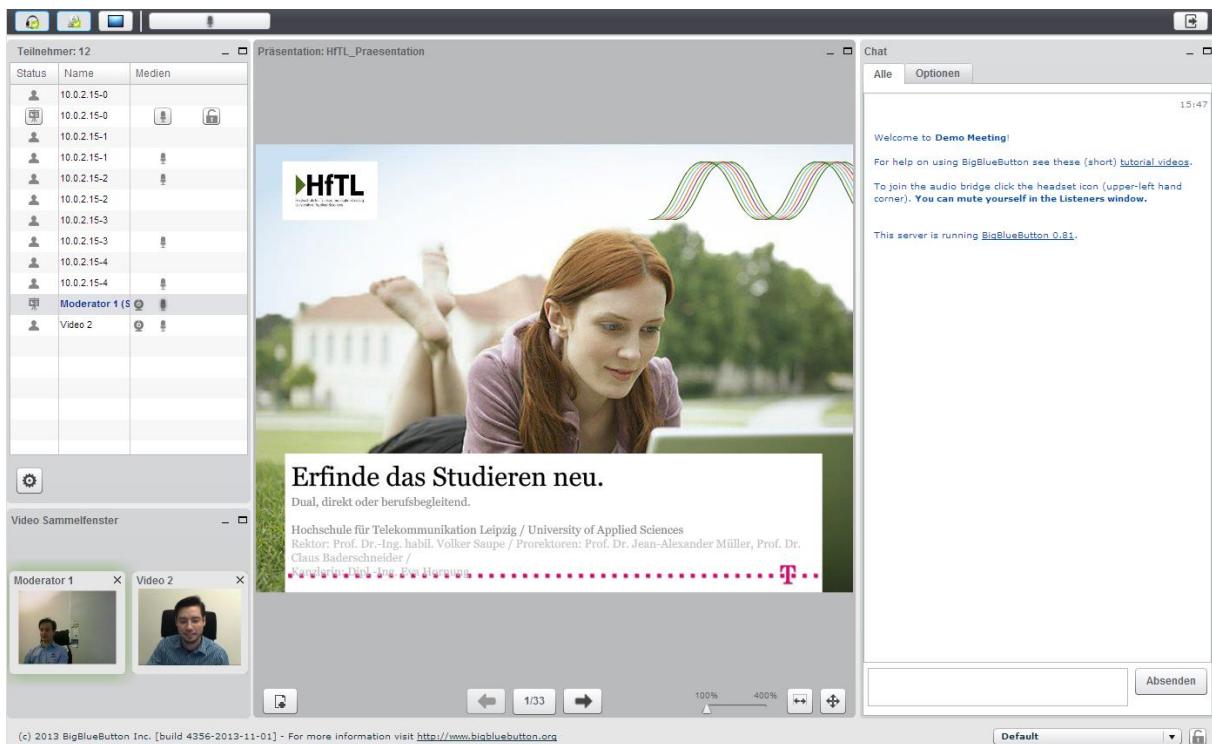


Abbildung 6-1 Interface des BBB Flash-Clients

Die Qualität der Umwandlung hängt stark von der Komplexität der Vorlage ab. Vor allem Microsoft Office Dokumente mit vielen Ebenen oder Animationen führen zu Fehldarstellungen.

Die Darstellung der im BBB-Client hochgeladenen PDF-Dokumente erfolgte im Test fehlerfrei. Dies wurde im Testzeitraum mehrfach erfolgreich überprüft.

Das Whiteboard bietet eine Funktion zum Austausch von Schrift und Zeichnungen, welches ähnlich einer Tafel funktioniert. In der aktuellen Version von BBB lässt sich ohne Import eines Dokumentes mit leeren Seiten kein leeres Whiteboard anzeigen.

Mit dem Werkzeug „Hervorhebung“ kann man es mit der passenden Hardware (z.B. Stifttablett) beschreiben wie mit Kreide auf einer Tafel. Zusätzlich gibt es fertige geometrische Formen – Rechteck, Kreis und Linie – die ähnlich einem Grafikprogramm zu bedienen sind. Damit lassen sich zum Beispiel Bereiche des Whiteboards mit Rahmen versehen. Die Farbe (13 voreingestellte Farben und ein Eingabefeld für Farbcodes im Hexadezimal Format) und die Breite (von 1 px – 30 px) des Rahmen- und des Hervorheben-Werkzeuges sind einstellbar, jedoch gibt es keine Füllfarbe. Der Bereich innerhalb der Form bleibt transparent.

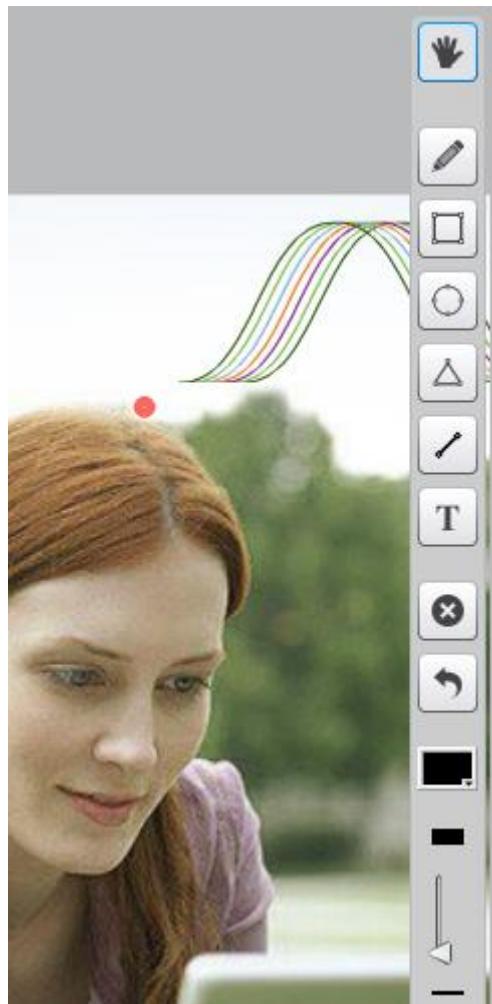


Abbildung 6-2 Werkzeugleiste mit Pointer

Mit Version 0.81 wurde ein Text-Werkzeug hinzugefügt. Um Texte auf dem Whiteboard mittels Tastatur einzufügen, muss zuerst ein rechteckiger Rahmen gezogen werden. Anschließend erscheint das Textfeld mit der Eingabemöglichkeit. Im eingeblendeten Optionsmenü lassen sich Zeichengröße und Schriftfarbe einstellen. Es existiert keine Möglichkeit, die vorgenommenen Änderungen am Whiteboard zu exportieren. Die Funktionen des Whiteboards können systembedingt nur der Moderator oder ein vom Moderator zum Präsentator ernannter Teilnehmer nutzen. Eine gleichzeitige Nutzung weiterer Teilnehmer zum Erfassen von Meinungsbildern ist nicht möglich.

Eine weitere Kommunikationsmöglichkeit ist der Chat. In diesem Fenster können alle Mitglieder Textnachrichten versenden, die von allen gelesen werden können. Zusätzlich ist es möglich, private Chaträume zwischen einzelnen Teilnehmern oder Teilnehmergruppen zu erstellen.

Mit BBB ist es außerdem möglich, den Bildschirminhalt des Moderators nach der Bildschirmfreigabe allen Teilnehmern anzuzeigen. Die Freigabe lässt sich auf einen Bereich des Monitors beschränken, lässt aber auch eine vollständige Darstellung zu. Die Anzeige der Freigabe erfolgt bei den Teilnehmern verzögert und der erste Aufbau des Bildes ähnelt einem Schachbrettmuster. Dieses Muster tritt auch bei stark wechselnden Bildinhalten, wie zum Beispiel einem Video, auf.

6.3 Architektur

BBB besteht aus vielen einzelnen Komponenten, deren Zusammenwirken die Funktionen der Software ermöglicht.

Der Anwender verbindet sich mit Hilfe des Flash Clients im Browser, der einen VoIP Client für die Audiokommunikation enthält, mit dem Webserver nginx über den Standard-http-Port 80. Nginx stellt gleichzeitig eine mögliche Tunnelverbindung für RTMP bereit, falls der Verbindungsauftbau über den Standardport 1935 fehlschlägt.

Nach dem Verbindungsauftbau leitet nginx die Informationen zur Erstellung und/oder zum Betreten der Konferenz an die Grails Anwendung bbb-web über http Port 8080 weiter. Bei einer Standardinstallation ist dies eine serverinterne Verbundung. Anschließend gelangen diese Informationen unter Nutzung von redis an die bbb-apps Anwendung, die mit Hilfe von FreeSWITCH-esl einen Raum in FreeSWITCH erstellt und den Anwender als Teilnehmer hinzufügt.

Sofern der Anwender an der Audiokonferenz teilnimmt, wird versucht eine RTMP-Verbindung über Port 1935 zu ihm zu öffnen, um den Audiostream zu übertragen. Falls dies fehlschlägt, wird der Umweg über den nginx Tunnel mit Hilfe eines RTMP Tunnels über Port 80 benutzt.

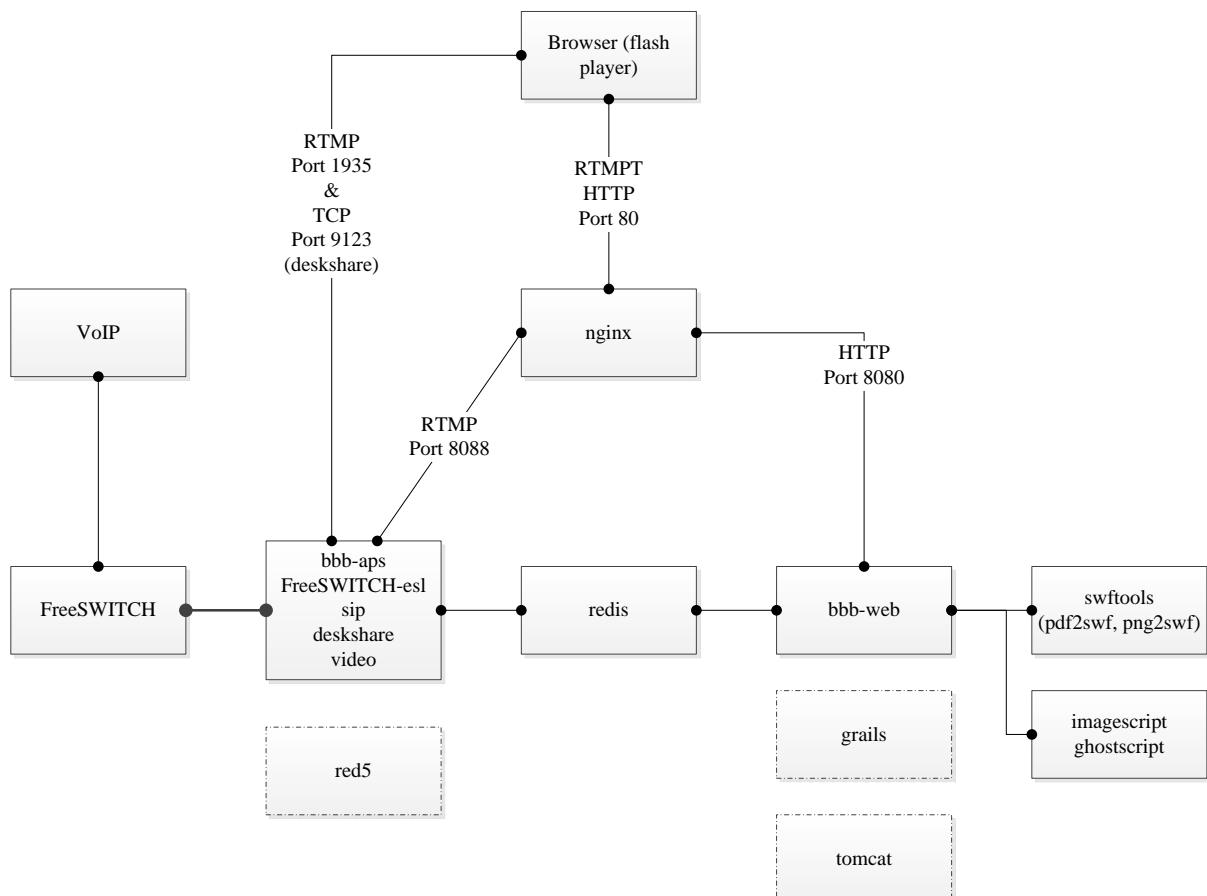


Abbildung 6-3 Komponenten von BBB und deren Zusammenwirken [6]

7 Komponenten von BigBlueButton

7.1 Nginx – Webserver

Nginx ist ein Webserver, der im Vergleich zum Marktführer Apache sehr schnell und schlank ist. Er wurde mit der Vorgabe konzipiert, selbst unter hoher Last nur wenige Ressourcen zu verbrauchen. Nginx wurde von Igor Sysoev entwickelt und wird laut Netcraft von ca. 13 % aller Websites genutzt (Stand: März 2013).

Da im Gegensatz zu Apache keine (nachladbaren) Module zum Einsatz kommen, werden benötigte Funktionen bereits beim Kompilieren von nginx angegeben. So kann man im Gegensatz zu den vorkompilierten Paketen aus den offiziellen Paketquellen oder dem PPA den Webserver noch genauer auf den geplanten Einsatzzweck abstimmen und weitere Performance-Gewinne erzielen. Bevor nginx kompiliert werden kann, werden die Pakete build-essential (Basiswerkzeuge zur Kompilierung), libpcre3-dev, libssl-dev (wird nur für SSL bzw. HTTPS benötigt) und zlib1g-dev benötigt. Je nach gewünschter Funktionalität können auch noch weitere Pakete hinzukommen.

Nginx bildet sich aus einem „Master“-Prozess und vielen „Slave“- bzw. „Client“-Prozessen. Man steuert nginx mit dem Master-Prozess, den man mit dem Befehl `nginx` anspricht. Dieser Webserver kann alle Typen von Websites und Textdateien komprimieren im Gegensatz zu dem ZLib-Modul von PHP, welches nur PHP komprimiert.

Nginx besitzt standardmäßig nur einige Module. Allerdings kann man nginx mit vielen weiteren Modulen benutzen, deren offizielle Anlaufstellen im Internet zu finden sind. Die Module kann man allerdings - nicht wie bei allen anderen Webservern - direkt benutzen, sondern muss sie mit nginx kompilieren. [7]

7.2 Red5 – Flash Media Server

Red5 ist ein freier Streaming-Server für das Real Time Messaging Protocol (RTMP). Er wird als freie Software auch im Quelltext unter den Bedingungen der Apache Public License 2.0 verbreitet und kann bei der Linux-Distribution Ubuntu direkt aus den Standard-Paketquellen installiert werden.⁴

Red5 ist in Java geschrieben. Die erste öffentliche Version (0.2) stammt vom 21. Oktober 2005. Er wurde auf Basis eines Reverse Engineering des Real Time Messaging Protocol von Adobe [8] und des Action Message Format (AMF) erstellt und steht in Konkurrenz zum Flash Media Server von Adobe Systems.

Red5 unterstützt Streaming und Aufzeichnung von Audio- und Video-Inhalten, die Veröffentlichung von Live Streams und das Konzept des Flash Remoting, womit Objekte und Methodenaufrufe zwischen Flash-Programmen (zum Beispiel Web-Anwendung und Anwendungs- server) versendet werden können und so beispielsweise Flash-Anwendungen im Hintergrund mit einem Red5-Server kommunizieren können. Audio-Inhalte können in MP3 oder AAC,

⁴ <http://packages.ubuntu.com/search?red5>

Video als Flash Video oder in H.264 gestreamt werden. Red5 unterstützt weiterhin verschlüsselte Verbindungen (RTMPS), Skripting (des Servers) mit ActionScript (AS) und Serverlastverteilung (mit Multiplexing RTMP und der Clustering-Software Terracotta). [9]

7.3 FreeSWITCH – Echtzeit Audiomixer

FreeSWITCH wird im BBB-Server als Echtzeit Audiomixer verwendet und realisiert mit Open-Source-Codecs qualitativ hochwertige Audioverbindungen. Es ermöglicht, unterschiedliche Qualitäten und Sample-Raten zu mixen. Alle eingehenden Verbindungen werden permanent in Echtzeit zusammengefasst und wieder an alle User verteilt.

Diese Plattform läuft unter den Betriebssystemen Windows, Linux und MacOS und ist lizenziert unter Mozilla Public Licence (MPL). Es bietet kompatible Protokolle zu SIP, H.323, GoogleTalk, RTMP, Skype, SRTP, ZRTP, PSTN, ISDN, SS7. Das RTMP-Protokoll wird in BBB hauptsächlich genutzt. [10]

Libfreeswitch, die Kernbibliothek von FreeSWITCH hat die Fähigkeit, in andere Produkte integriert zu werden oder als eigenständige Anwendung zu funktionieren.

Das Projekt wurde erstmals im Januar 2006 auf der O'Reilly Media's ETEL Conference angekündigt. Das erste offizielle Release erfolgte im Mai 2008.

FreeSWITCH baut auf frei verfügbaren Softwarebibliotheken auf, wie zum Beispiel Apache Portable Runtime, SQLite, PCRE, Sofia-SIP und libspeex (ab Version 1.0.3). BBB nutzt zur Audiokompression den Speex-Codec mit einer Abtastrate von 24 kHz. Die Audioqualität ist dabei eher befriedigend. Durch einen manuellen Eingriff in die Konfigurationsdatei `/opt/freeswitch/conf/autoload_configs/conference.conf.xml` lässt sich leicht eine bessere Qualität erzielen, indem der Eintrag Rate im wideband Profil von 24000 auf 48000 ändert. Damit erhöht sich jedoch die benötigte Bandbreite integrierte leicht und die Aufzeichnungsfunktion kann nicht mehr genutzt werden.

```
<profile name="wideband">
    <param name="domain" value="$$\{domain\}" />
    <param name="rate" value="24000"/>
```

(Auszug conference.conf.xml)

FreeSWITCH bietet eine API zur Steuerung der Audiofunktion und Rechte der Teilnehmer. Die Anwendungen können dabei in unterschiedlichsten Programmiersprachen geschrieben sein. Zur Anrufsteuerung wird der Event Socket genutzt. Dieser bietet ein programmiersprachenunabhängiges Interface und ist über TCP-Socket basierte Verbindungen erreichbar.

Die Event Socket Library (ESL) ist unter anderem für JavaScript, Perl, PHP und Ruby verfügbar. Die Ruby Schnittstelle wird durch die bbb-web Anwendung genutzt um den Verbindungsaufbau zu steuern.

Ab der FreeSWITCH Version 1.4 existiert ein Support für WebRTC. [11] Dieses Protokoll wird unter HTML5 als Streaming-Protokoll für Audio- und Videoverbindungen verwendet. Damit ist es möglich auf den Flash-Client zu verzichten und auf eine native Browseranwendung zurück zu greifen. Dies erleichtert den Zugang von mobilen Endgeräten.

8 Verwendete Protokolle und Dienste

8.1 RTMP – Real Time Media Protocol

Das Real Time Messaging Protocol (RTMP) ist ein Netzwerkprotokoll, mit dem Audio-, Video- und andere Daten von einem Mediaserver zu einem Flash-Player mittels Internet übertragen werden können. Dieses Protokoll wurde von Adobe Systems zu diesem Zweck entwickelt. Es wurde für Flash geschaffen, aber auch Adobe LiveCycle Data Services ES oder Openmeetings genutzt. Die Protokollspezifikation wurde im April 2009 offengelegt.

In Abhängigkeit von den Umgebungsbedingungen wird eine der drei nachstehend beschriebenen Varianten genutzt.

Variante 1 ist RTMP direkt auf Basis TCP-Port 1935. Die TCP-Variante nutzt eine persistente Verbindung, um eine Echtzeitkommunikation zu ermöglichen. Um eine gleichmäßige Wiedergabe sicherzustellen, werden größere Datenblöcke übertragen. Das Protokoll zerlegt die Daten in 128-Byte-Blöcke (64-Byte für Audio).

Die zweite Variante ist RTMPT auf Basis HTTP, um Firewalls zu überwinden. Mittels RTMPT können Tunnel aufgebaut werden, um Firewalls zu überwinden.

Darüber hinaus gibt es die Variante RTMPS auf Basis HTTPS für sichere Verbindungen. RTMPS erlaubt die Nutzung von SSL, um verschlüsselte Datenübertragung zu erlauben. Diese kommt aber bei BBB bisher nicht zum Einsatz. Bei BBB wird mittels Variante 1 versucht über den TCP-Port 1935 beim einem sich einwählenden Client, eine Verbindung zu erstellen. Sollte dies nicht möglich sein, wird mittels RTMPT ein Tunnel über den nginx Webserver erstellt und somit die RTMP Verbindung erstellt. Diese wird als Fallback-Methode bezeichnet.

8.2 Ruby – Skript Sprache

Die Skriptsprache Ruby wurde 1993 bis 1995 von Yukihiro „Matz“ Matsumoto in Japan entwickelt. Er kombinierte Teile seiner Lieblingssprachen (Perl, Smalltalk, Eiffel, Ada und Lisp) und formte daraus eine neue Programmiersprache, in der funktionale und imperitative Programmierung ausbalanciert sind.

In Ruby ist alles ein Objekt. Jedes Stück Information kann eigene Eigenschaften und Aktionen besitzen. In der Welt der objektorientierten Programmierung nennt man diese Eigenschaften Instanzvariablen und die Aktionen Methoden. Rubys rein objektorientierter Ansatz wird üblicherweise mit einem Stück Code demonstriert, in dem die Methode einer Zahl aufgerufen wird:

```
5.times { print "Wir *lieben* Ruby -- es ist ungeheuerlich!" }
```

In vielen Sprachen sind Zahlen und andere primitive Typen keine Objekte. Ruby folgt dem Einfluss von Smalltalk und gibt all seinen Typen Methoden und Instanzvariablen. Das erleichtert die Verwendung von Ruby, da die Regeln für Objekte überall in der Sprache gültig

sind. [7] Ruby wird in BBB vor allem zu Steuerung des Sitzungsaufbaus mittels bbb-web und der Steuerung der Umwandlung hochgeladener Dokumente durch swf-tools, imagemagick und ghostscript genutzt. Ein weiterer Vorteil

9 Verwendete Linux Distributionen

9.1 Ubuntu Server 10.04 LTS & 12.04 LTS

Ubuntu ist eine freie und kostenlose Linux-Distribution, die auf Debian basiert. Debian ist ein seit 1993 gemeinschaftlich entwickeltes freies Betriebssystem.

„Der Begriff Ubuntu kommt aus den Sprachen der afrikanischen Völker Zulu und Xhosa und steht für „Menschlichkeit“ und „Gemeinsinn“, aber auch für den Glauben an ein universelles Band des Teilens, das alles Menschliche verbindet. Alle Entwickler müssen zudem den Code of Conduct unterzeichnen, mit dem sie sich verpflichten, den Grundsatz der „Menschlichkeit“ Freundlichkeit, Respekt voreinander, Rücksicht, Teamarbeit und Ähnliches sowohl bei der Entwicklung und der Kommunikation untereinander als auch bei dem Umgang mit den Benutzern einzuhalten. [12]

Jede Version hat einen eigenen Codenamen und eine Versionsnummer, die auf das jeweiligen Veröffentlichungsjahr und den Veröffentlichungsmonat schlussfolgern lässt. Die Versionsnummer 10.04 steht für April 2010. Der Codename dieser Ubuntu-Version heißt *Lucid Lynx* (englisch für *klarer Luchs*) und lässt sich auch in verkürzter Weise (nur Lucid) bei der Suche nach kompatiblen Programmen nutzen, die nicht im Paketmanager verfügbar sind. Meist stehen die Codenamen auch in Kommentaren der Quellcodedateien. Die Version 12.04 LTS ist im April 2012 erschienen und trägt den Namen *Precise Pangolin* (englisch für *akkurates Schuppentier*). Die Abkürzung LTS steht für long-term Support (Langzeitunterstützung). Generell besitzen alle Ubuntu-Versionen einen offiziellen Supportzeitraum von 18 Monaten. In dieser Zeit werden Updates durch die Ubuntu-Entwickler über das Ubuntu-Repository bereitgestellt. Bei LTS-Versionen ist dieser Supportzeitraum auf 3 Jahre (Desktop) und auf 5 Jahre (Server) nach Erstveröffentlichung verlängert. Die längeren Zeiträume geben Einsatzsicherheit, denn ein Versionsupgrade läuft in den seltensten Fällen problemlos ab. Die größten Schwierigkeiten entstehen durch nicht mehr unterstützte Pakete und durch aktualisierte Programmabibliotheken, die mit den älteren Versionen inkompatibel sind.

Das Deployment von BBB auf ein Ubuntu-Server 10.04 gestaltet sich vergleichsweise einfach. Im BBB Wiki existiert eine Schritt für Schritt Anleitung zum Vorgehen⁵. Nach dem Einstellen der Sprache auf `en_US.UTF-8`, dem Überprüfen ob Ubuntu in der 64bit Version (`x86_64`) installiert wurde, der Installation von Ruby 1.9.2 und dem Hinzufügen der BBB Repositorys (`sources.list`) reicht ein `sudo apt-get install bigbluebutton` aus, um die Installation und damit das Deployment zu starten. Anschließend installiert der Paketmanager alle für den BBB Betrieb notwendigen Programme und Bibliotheken mit ihren Abhängigkeiten.

Der vom Rechenzentrum bereitgestellte Server kann Daten aus dem Internet nur über einen Proxy-Server erreichen. Daher mussten vor dem Installationsprozess die Konfigurationen

⁵ <https://code.google.com/p/bigbluebutton/wiki/InstallationUbuntu>

aller beteiligten Tools (`apt`, `wget`, `gem`, `svn`) angepasst werden um Fehlermeldungen zu Kommunikationsproblemen zu vermeiden.

Das Deployment auf Ubuntu-Server 12.04 verläuft größtenteils identisch zum Deployment auf Lucid Lynx. Vor der Installation von BBB müssen einige Bibliotheken installiert werden, die unter 10.04 vorinstalliert waren. Eine Beschreibung hierfür findet sich in der Google-Groups BigBlueButton-dev. [13]

```
cd /tmp
wget http://security.ubuntu.com/ubuntu/pool/main/libf/libffi/libffi5_3.0.9-
3ubuntul_amd64.deb
sudo dpkg -i libffi5_3.0.9-3ubuntul_amd64.deb
sudo apt-get install zlib1g-dev libssl-dev libreadline-dev libyaml-dev
build-essential bison checkinstall libffi5 gcc checkinstall libreadline5
libyaml-0-2

sudo apt-get install libgmp3c2
wget http://free.nchc.org.tw/ubuntu//pool/main/m/mpfr/libmpfr1ldbl_2.4.2-
3ubuntul_amd64.deb
sudo dpkg -i ./libmpfr1ldbl_2.4.2-3ubuntul_amd64.deb
sudo gem install gherkin -v '2.2.9' -- --with-cflags=-w
sudo gem install cucumber -v '0.9.2' -- --with-cflags=-w

sudo mkdir /var/www/nginx-default

sudo vi /var/www/nginx-default/50x.html

Past the following content and save the file.

<!DOCTYPE html>

<html>
<head>
<title>Error</title>
<style>
body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
}
</style>
</head>

<body>
```

```

<h1>An error occurred.</h1>

<p>Sorry, the page you are looking for is currently unavailable.<br/>

Please try again later.</p>

<p>If you are the system administrator of this resource then you should
check the <a href="http://nginx.org/r/error_log">error log</a> for de-
tails.</p>

<p><em>Faithfully yours, nginx.</em></p>
</body>
</html>

If you see this error after installation [ nginx: [emerg] bind() to
0.0.0.0:80 failed (98: Address already in use) ]

sudo /etc/init.d/apache2 stop
sudo /etc/init.d/nginx restart

```

[13]

Nach der Durchführung dieser Schritte kann die Installation von BBB gestartet werden. Zum Testzeitpunkt wurde von BBB noch OpenOffice genutzt, welches unter 12.04 nicht mehr verfügbar ist. Als Nachfolger wird von Ubuntu LibreOffice genutzt, welches mit der BBB Version 0.81 auch von den BBB-Entwicklern unterstützt wird.

9.2 SUSE Linux Enterprise 11 SP2

SUSE Linux Enterprise Server ist eine speziell auf Unternehmen ausgerichtete Linux Distribution mit langen Supportzeiträumen (bis zu 9 Jahren). Die SUSE GmbH (Gesellschaft für Software und System Entwicklung) wurde 1992 in Nürnberg gegründet. Aus Ermangelung eines guten Installationswerkzeuges für die erste SUSE Linux Distribution wurde YaST (Yet another Setup Tool) entwickelt, dass noch heute Bestandteil jeder Distribution ist. Nach einer Kooperation mit IBM wurde die Firma 2004 von Novell aufgekauft. Derzeit existieren zwei verschiedene Produkttypen. Auf der einen Seite das frei verfügbare openSUSE und auf der andern die kommerzielle Produktpalette von SUSE Linux Enterprise. Letzteres bildet in der Version 11 mit SP2 die Basis für das Deployment von BBB.

Unter SUSE Linux existiert kein fertiges Paket zur Installation von BBB, daher wurde als erster Ausgangspunkt die Paketliste der der Installationsroutine aus Ubuntu 10.04 und die zusätzlichen Pakete für 12.04 gewählt. Daraufhin wurden die einzelnen Pakete mittels YaST gesucht und sofern möglich direkt installiert. Dies war jedoch in den wenigsten Fällen erfolgreich. Zum einen unterscheiden sich die Paketnamen zwischen SUSE und Ubuntu und zum anderen waren die wenigsten Pakete im SUSE Repository verfügbar. Alleine durch die

schlichte Anzahl (über 600 Suchbegriffe) wurde dieses Vorgehen abgebrochen. Nach Rücksprache mit den Mitarbeitern des Rechenzentrums wurde daraufhin die Internetrecherche intensiviert. Ergebnis war ein Blogeintrag über das erfolgreiche Deployment von BBB v0.8 auf openSUSE 11.4. [2] Die Zahl der in der Anleitung enthaltenen Pakete war überschaubar. Für die Enterprise Version musste jedoch zu jedem Paket die Quelldatei geladen werden und diese in ein RPM Paket kompiliert werden. Das Paket konnte anschließend über YaST installiert werden. Als RPM-Paketquellen kamen software.opensuse.org/search und rpmfind.net hauptsächlich zum Einsatz. Dabei wurde jeweils das entsprechende Source-Paket geladen, da die Funktion der vorkompilierten auf Grund von fehlenden Abhängigkeiten nicht sichergestellt werden konnte. Die Vorgehensweise um vom Source-Paket zur installationsfähigen Version zu kommen ist bei allen Paketen identisch: Nachdem das Paket per `wget` oder über SCP auf den Server übertragen wurde, folgte die Extraktion der Dateien über den Konsolen Befehl `rpm -Uvh [Sourcepaketname]`. Die Paketinhalte werden daraufhin unter `/usr/src/packages` kopiert. Dort existieren anschließend vier Unterverzeichnisse: `SPECS` (mit den Specfiles), `SOURCES` (Quelldateien), `BUILD` (Kompilierungsanweisungen) sowie `RPMS` (fertige RPM-Dateien) mit den Unterverzeichnissen `noarch` und `x86_64`. Der Kompilierungsvorgang wird mittels `rpmbuild -bb [specfile]` über die Konsole gestartet. Sofern alle Abhängigkeiten des Paketes auf dem Server verfügbar sind und das Spec-File keine Fehler enthält, entsteht ein unter YaST installierbares auf das System zugeschnittenes Paket. Probleme beim Kompilierungsprozess entstanden durch die Verwendung von openSUSE Paketen mangels Alternativen. Schon kleinste Unterschiede in der Bezeichnung von Abhängigkeiten führten zu Fehlern während der Kompilierung. Oft half eine Korrektur bzw. ein auskommentieren des entsprechenden Eintrages im Spec-File oder die Aktualisierung der Fehler verursachenden Abhängigkeiten. Führte dies nicht zu einem erfolgreichen Abschluss, musste ein anderes Source-Paket verwendet werden. Dieser letzte Ausweg konnte aber zu weiteren Problemen führen, z.B. wenn das fehlerverursachende Paket Glied einer Kette von Abhängigkeiten war, und jetzt alle folgenden Pakete neu erstellt werden mussten.

Letztendlich führte die Komplexität des gesamten zeitaufwendigen Deploymentprozesses zu einem Abbruch dieses Projektes. Auch aus wirtschaftlicher Sicht ist ein derartiger Aufwand kaum zu rechtfertigen, da er nicht einmalig bleibt. Alle Pakete, die nicht im SUSE Linux Enterprise Repository vorhanden sind, müssen manuell gewartet werden. Dies ist bei einem aus dem Internet erreichbaren Server besonders wichtig um Sicherheitslücken der Software schnell schließen zu können.

10 ILIAS – Open-Source Lernplattform

Die erste Form der ILIAS Lernplattform - Integriertes Lern-, Informations- und Arbeitskooperations-System - wurde 1997-1998 im Rahmen des VIRTUS-Projektes der Universität zu Köln entwickelt.

In erster Linie handelte es sich bei dieser frühen Version um eine Datenbankapplikation, mit der die Autoren die im Rahmen des Projektes entwickelten Lerneinheiten im Web darstellen, bearbeiten und erstellen wollten. Eine Oberfläche für die Lernenden wurde integriert. Diese Oberfläche enthält elementare Funktionen wie Suche und personalisierte Bookmarks, die auch heute wichtige Funktionen der Lernplattform bilden.

Aufgrund der starken Nachfrage wurde ILIAS im Jahr 2000 unter GPL als Open-Source-Software veröffentlicht. Bis 2004 erfolgte eine grundlegende Neuentwicklung auf Basis modularer, objektorientierter XML-Technologien, die auch eine rollenbasierte Zugriffskontrolle (RBAC) ermöglichen.

Diese Basis hat bis heute Gültigkeit. Einzig die Technologien zur Darstellung der Bedienoberfläche (von tabellenbasierter zu CSS steuerbarer Ebenen Struktur) wurden den Webstandards angepasst.

Im Jahr 2009 wurde für die sichere Weiterentwicklung und Betreuung der Software der Verein ILIAS open source e-Learning e.V. durch Mitglieder der ILIAS-Community gegründet.

An der HfTL war und ist ILIAS als Lernplattform das Kommunikationsmittel für Studierende aller Studienformen.

Mit der Open-Source Lernplattform ILIAS steht der gesamten Hochschule ein leistungsfähiges Learning Management System zur Verfügung, mit dessen Hilfe internetbasierte Lehr- und Lernmaterialien erstellt und verfügbar gemacht werden. Lehrveranstaltungen werden zunehmend mit E-Learning-Elementen gestaltet.

ILIAS dient auch der Kommunikation und Zusammenarbeit unter Benutzern. Es erlaubt eine effiziente Erstellung von Kursmaterialien.

Besonders hervorzuheben ist, dass alle Anwender über einen personalisierten Arbeitsbereich verfügen. Von diesem aus beginnt der Nutzer seine Sitzung im System, kann zahlreiche Funktionen wie zum Beispiel Kalender und Terminplaner nutzen und sich insbesondere auch Informationen über den eigenen Lernfortschritt einholen.

Natürlich kann E-Learning die klassische Präsenzlehre nicht ersetzen, aber vielmehr an geeigneten Stellen ergänzen und dadurch die Qualität der Hochschullehre fördern. Es ist ein effizientes Lehrangebot, welches ein zeit- und ortsunabhängiges Lernen ermöglicht und den Lebensumständen der Studierenden entspricht. Komplexe Sachverhalte können über multimediale Präsentationsverfahren besser veranschaulicht werden.

Abbildung 10-1 ILIAS Objektliste mit BBB

ILIAS ist an der HfTL im eLearning seit 2006 im Einsatz. Aktuell wird die Version 4.3.6 mit einem angepassten Custom-Style eingesetzt. Abbildung 10-1 zeigt das Benutzerinterface eines Administrators mit der Liste möglicher zu erstellender Module. Dieser Installation wurde schon das BigBlueButton Plug-In hinzugefügt und erscheint daher als letzter Punkt der Liste. Die Abbildung verdeutlicht ein Erfolgskonzept von ILIAS. Durch seine Modularität ist es gut erweiterbar und somit den Einsatzbedürfnissen anpassbar. Für die Plug-In Entwicklung gibt es einen eigenen Bereich mit Datensammlung, Forum, Bug-Tracker und Development Guide.⁶

⁶ http://www.ilias.de/docu/goto.php?target=cat_1442&client_id=docu

11 Erweiterung des BigBlueButton Plug-In für ILIAS

Eine Websuche mit google.de mit dem Such-Term „bigbluebutton plugin ilias“ brachte eine frühe Version eines BBB Plug-Ins auf der ersten Ergebnisseite. [4] Die letzte Version wurde im Mai 2011 veröffentlicht und sollte laut enthaltener plugin.php bis zur ILIAS Version 4.4.0 funktionieren.

Eine Integration nach der enthaltenen Anleitung (README.txt) führte zu keinem Erfolg. Weitere Recherchen in den ILIAS Foren führten zu einer aktualisierten Version, die auch mit der installierten ILIAS Version 4.3.x funktioniert. [14] Im Oktober 2013 hat der Autor der Aktualisierung einen Eintrag in die Plugin-Datensammlung auf der ILIAS Homepage vorgenommen.⁷

Nach der Installation, Aktivierung und Einrichtung steht ein neues Objekt mit dem Namen BigBlueButton Virtual Classroom am Ende der Erstellungsliste in ILIAS zur Verfügung. Damit ist jeder ILIAS Benutzer mit entsprechenden Rechten zur Erstellung von BBB Objekten in der Lage, einen Klassenraum anzulegen. Als einstellbare Optionen stehen zu besseren Unterscheidung ein Titel-, Beschreibungs- und Willkommenstextfeld sowie ein Online Schalter zur Verfügung. Nach dem Hinzufügen kann der Raum auf dem BBB Server mit einem Klick in der ILIAS-Oberfläche eröffnet und betreten werden.



Abbildung 11-1 BBB Objekt - Startmaske

11.1 BBB-API

Für die Kommunikation mit dem BBB-Server verwendet das ILIAS BBB Plug-In die vom Server bereitgestellte API.⁸ Per http GET wird ein API call an den Server abgesetzt welcher darauf, außer bei einem erfolgreichen join, mit einer Antwort im XML Format reagiert. Einen API call wird bei der Standardinstallation an `http://[serverurl]/bigbluebutton/api/` gesendet.

⁷http://www.ilias.de/docu/ilias.php?ref_id=3342&record_id=73&cmd=renderRecord&cmdClass=ildatacollectionrecordviewgui&cmdNode=bn:36:34&baseClass=ilRepositoryGUI

⁸<https://code.google.com/p/bigbluebutton/wiki/API>

Mögliche Aufrufe sind u.a.:

`create, join, end, isMeetingRunning, getRecordings und publishRecordings.`

An den Aufruf werden weitere Parameter mittels ? angehängt und per & verbunden.

Beispiel:

```
http://[serverurl/-ip]
/bigbluebutton/api/end?[Parameter]=[Wert]&checksum=[checksum]
```

Allen Aufrufen gemein ist der letzte Parameter `checksum`, welcher zugleich das einzige Sicherheitsfeature von BBB gegen ungewollte Nutzung des Dienstes ist. Ohne eine gültige `checksum` wird jeder API Call mit einem Fehler beantwortet. Somit gelingt zum Beispiel ein Erstellen eines neuen Raumes auf dem Server nicht, solange das `Shared Secret`, ein 32 stelliger String im Hexadezimalformat, dem Aufrufer unbekannt ist. Die `checksum` ist eine Hashsumme – gebildet mittels SHA1 Algorithmus und ausgegeben in Hexadezimaldarstellung – über den gesamten API call bis zum letzten Parameter ohne ? und `&checksum=[checksum]` und dem Anhängen des `Shared Secrets` des BBB Servers.

Dies lässt sich auf einer Linux Bash und installiertem `sha1sum` Paket nachstellen (angenommenes `Shared Secret`: `abcdef1234567890abcdef1234567890`):

```
echo -n "endmeetingID=room2&password=mpabcdef1234567890abcdef1234567890" | sha1sum
```

Ausgabe: `0777ed6d1cebcd312a8caeb225ea854f830cd17 -`

Die URL für den API call end (Sitzung beenden) lautet:

```
http://[serverurl]/bigbluebutton/api/end?meetingID=room2&password=mp&checksum=0777ed6d1cebcd312a8caeb225ea854f830cd17
```

Sofern der Aufruf erfolgreich war, sendet der Server folgende XML Antwort:

```
<response>
<returncode>SUCCESS</returncode>
<messageKey>sentEndMeetingRequest</messageKey>
<message>
A request to end the meeting was sent. Please wait a few seconds, and then
use the getMeetingInfo or isMeetingRunning API calls to verify that it was
ended.
</message>
</response>
```

Am Eintrag `returncode` ist erkennbar, ob der Aufruf wie im Beispiel erfolgreich war oder fehlgeschlagen ist. `Message Key` ist eine kurze Erläuterung zum abgesendeten API call und der `Message Tag` enthält eine Erläuterung zum weiteren Vorgehen oder eine Beschreibung des Fehlers.

Die unterschiedlichen API calls benötigen jeweils verschiedene Übergabeparameter. Der im Beispiel verwendete end call benötigt eine `meetingID` (gewöhnlich der Raumname) und das `password` (Passwort) der Moderatoren. Beide Werte sind als URL encoded⁹ Strings zu übergeben. Durch den Aufruf von end wird vom Red5 Server jedem zu dieser `meetingID` verbundenen Clienten ein logout Event gesendet und somit die Sitzung beendet. Eine vollständige Beschreibung der API findet sich im BBB wiki. [15]

11.2 Funktionen

Die im Internet verfügbare Variante des Plug-In erlaubt es nur Teilnehmern mit Änderungsrechten am ILIAS Objekt (weiterhin Moderatoren genannt), nach der Erstellung dessen einen BBB-Raum zu öffnen und ihn anschließend zu betreten. Teilnehmer ohne diese Rechte (weiterhin Teilnehmer genannt) müssen warten, bis der Raum geöffnet wurde, um ihn zu betreten. Diese Funktionalität würde bei Verzögerungen bei der Raumöffnung in der Folge zu weiteren Verzögerungen führen.

Die BBB-API erlaubt auch ein Erstellen des Raumes ohne sofortigen Moderatorenbeitritt, so dass die Teilnehmer bei Verspätungen des Moderators im bereits geöffneten Raum warten können. Weiterhin sollte die Bedienung des Plug-In vereinfacht werden, so dass keine zusätzlichen Seitenaufrufe notwendig sind, damit die Funktionen „Sitzung Beitreten“ und „Sitzung beenden“ jederzeit verfügbar sind. Das Plug-In nutzt zur Darstellung der aktuell verfügbaren Funktionen html-Seitenteile sowie in diesen verschiedene div-Ebenen. Für den Raumersteller wird ein Popup-Fenster geöffnet und dieser kann ohne weiteres Zutun den Raum betreten.

Zusätzlich sollte die Aufzeichnungsfunktion von BBB verfügbar gemacht und vorangegangene Mitschnitte im Raum angezeigt werden.

11.3 Vorgehen

Zunächst wird das bestehende Plug-In analysiert und dessen Funktionen untersucht. Im Stammverzeichnis befinden sich drei Dateien, zwei als Textdokumente und ein php-Dokument.

Die Textdokumente enthalten die Installationsanleitung (README.txt) und das Änderungslog (Changelog.txt). Der Inhalt der plugin.php ist selbsterklärend. In ihr enthalten ist die Plug-ID für ILIAS, ein Feld für die Version, eine Beschränkung zur unterstützten ILIAS Version sowie ein Feld mit Namen der bisherigen Entwickler.

Weiterhin befinden sich fünf Ordner im Stammverzeichnis: unter classes befinden sich die php-Funktionen, js enthält ein jQery-Javascript, lang die unterschiedlichen Übersetzungen der vom Plug-In verwendeten Texte, sql die Erweiterungen der ILIAS MySQL Datenbank und Templates die verwendeten html-Seitenteile.

Bei der Analyse der html-Seitenteile fällt auf, dass die Entwickler unterschiedliche div-Ebenen zur Anzeige der Zustände „Virtuellen Raum starten“ (kein BBB-Raum erstellt) und

⁹ <http://tools.ietf.org/html/rfc3986>

„Virtuellem Raum beitreten“ nutzen und deren Sichtbarkeit über eine Javascript Funktion steuern. Das Javascript wird einmalig beim Aufruf der Seite ausgeführt, und bekommt die Zustandsinformation, ob ein Raum bereits eröffnet wurde aus der php-Variablen `{isMeetingRunning}`.

Mittels einfachen Vergleichs wird entweder die Ebene mit dem Link zur Raumöffnung oder die mit dem zum Raum beitreten sichtbar gemacht und die jeweils andere versteckt. Ein Vergleich der html-Teile `tpl.BigBlueButtonClient.html` (Teilnehmerversion) und `tpl.BigBlueButtonModeratorClient.html` (Moderatorenversion) zeigt die für die Raumöffnung fehlenden Teile. Darunter fällt die div-Ebene mit dem Link zur Eröffnung des Raumes mit dem zugehörigen Formular, welches zur Übermittlung der URL für den API call genutzt wird.

Nach dem Übernehmen der fehlenden Teile in die `tpl.BigBlueButtonClient.html` führte ein Aufruf der Seite zu keinem Erfolg. Eine Untersuchung mit dem Webseitenanalysewerkzeug des Browsers zeigte, dass die php-Variablen nicht mit Werten ersetzt wurden. Dies deutet auf fehlende Berechtigungen in einer php Funktion.

Eine Volltextsuche der Variable `{FORMACTION2}` führte zu der Funktion `showContent` in `class.ilObjBigBlueButtonGUI.php`.

Die Variable wird nur mit Werten belegt, wenn das Attribut `isModerator` für die `if`-Funktion mit dem Wert `true` belegt ist.

Hier bieten sich 2 Lösungsmöglichkeiten an, entweder das `isModerator` Attribut für alle Teilnehmer setzen oder die `else` Anweisung um die Variablenvergabe zu erweitern. Die erste Möglichkeit ist ein schlechter Ansatz, da er das Rechtemanagement und somit die Sicherheit des Plug-Ins beeinträchtigt.

```
//testedit by Udo: Start des Klassenraums ohne Moderatorrechte
$my_tpl->setVariable("CMD_START_CLASS","cmd[startClass]");
$my_tpl->setVariable("START_CLASS",$this->txt('start_bbb_class'));
$my_tpl->setVariable("FORMACTION2",$this->ctrl->getFormAction($this));
$my_tpl->setVariable("noClassRunning", $this->txt("no_class_running"));
$my_tpl->setVariable("startClass", $this->txt("start_class"));
(Auszug class.ilObjBigBlueButtonGUI.php: eingefügte Zeilen)
```

Dies allein führte nicht zum gewünschten Erfolg, da das Kommando `cmd[startClass]` auch an das Rechtemanagement gebunden ist.

Eine Änderung in der betreffenden Funktion – `startClass` auch mit Leserechten – führte zum gewünschten Erfolg.

```
function performCommand($cmd)
{
    switch ($cmd) {
        case "editProperties": //list all commands that need write permission here
        case "updateProperties":
        case "endClass":
        //case "startClass":
            $this->checkPermission("write");
```

```

        $this->$cmd();
        break;

        case "startClass":      //edit by Udo: member should open rooms
too
        case "showContent":     //list all commands that need read per-
mission here
            $this->checkPermission("read");
            $this->$cmd();
            break;
    }
}

(Auszug class.ilObjBigBlueButtonGUI.php: eingefügte Zeilen kommentiert)

```

Schließlich ist es auch Teilnehmern möglich, den BBB-Raum zu öffnen. Weitere Änderungen erleichtern die Bedienung des Objektes für Moderatoren.

Es wurde ein neuer virtueller Raum erstellt. Klicken Sie hier um den virtuellen Raum beizutreten.
Klicken Sie hier um den virtuellen Raum zu beenden.
(Beachten Sie, dass alle Teilnehmer getrennt werden wenn Sie sich noch in den Raum befinden).

Abbildung 11-2 BBB-Objekt Raum erstellt (Moderatoransicht)

Die ursprüngliche Version sah nach dem Ausführen einer Aktion wie zum Beispiel „Virtuellem Raum beitreten“ oder „Virtuellen Raum beenden“ einen Seitenrefresh vor, um zur nächsten Aktion zu gelangen. So muss der Moderator die Webseite des BBB-Objektes neu laden, um einen geöffneten Raum zu beenden. Sofern er aber das neue BBB-Flash-Client Fenster schließt, gelangt er nicht ohne weiteren Hinweis zu diesem Punkt.

Daher ist es nötig gewesen, die Aktionen „Virtuellen Raum beenden“ und anschließend „Virtuellen Raum starten“ zusätzlich zur Aktion „Virtuellem Raum beitreten“ zur Verfügung zu stellen.

Der Raum wurde beendet.
Klicken Sie hier, um einen virtuellen Raum zu starten.

Abbildung 11-3 BBB-Objekt Raum beendet

Zusätzlich wurde noch das Laden der jQuery Bibliothek im Plug-In unterbunden, da diese durch ILIAS selbst zur Verfügung steht.

```
{VAL_1}

<p>

<script type="text/javascript">
    ilAddOnLoad(
        function()
        {
            var isMeetingRunning={isMeetingRunning};
            if(isMeetingRunning){
                $('#openClassLink').show();
                $('#classNotStarted').hide();
            }else{
                $('#openClassLink').hide();
                $('#classNotStarted').show();
            }
        });
</script>

<a id="openClassLink" href="{bbbURL}" target="_blank">
{clickToOpenClass}
</a>

<div id="classNotStarted">
{classNotStartedText}
</div>
```

(tpl.BigBlueButtonClient.html Ausgangsversion)

Die Ausgangsversion enthielt für die Teilnehmer lediglich einen Hinweis, dass bisher noch kein Raum gestartet wurde und keine Möglichkeit selbst einen Raum zu öffnen.

```

{VAL_1}

<p>

<script type="text/javascript">
    il.Util.addOnLoad(
        function()
        {
            $('#startClassForm').hide();

            var isMeetingRunning={isMeetingRunning};
            if(isMeetingRunning)
                $('#openClassLink').show();
                $('#startClassDiv').hide();
                window.open("{bbbURL}");
            }else{
                $('#openClassLink').hide();
                $('#startClassDiv').show();
            }

            $('#startClassLink').click(function() {
                $('#startClassFormInput').click();
            });
        });
</script>

<a id="openClassLink" href="{bbbURL}" target="_blank">{clickToOpenClass}</a>
<div id="startClassDiv">{noClassRunning} <a id="startClassLink"
href="#">{startClass}</a></div>

<!-- Commented by Udo, members should open rooms too

<div id="classNotStarted">
{classNotStartedText}
</div> -->

<form id="startClassForm" action="{FORMACTION2}" method="post">
    <input id="startClassFormInput" class="submit" type="submit"
name="{CMD_START_CLASS}" value="{START_CLASS}" >
</form>

```

(tpl.BigBlueButtonClient.html nach Anpassungen)

Nach den Änderungen sind die Funktionen zum Raum öffnen hinzugekommen. In der Javascript Funktion `il.Util.addOnLoad` wurden die Einträge `#classNotStarted` durch `#startClassDiv` ersetzt und eine Funktion zum Abschicken der Formularfelder hinzugefügt. Im unteren Bereich wurde die nicht mehr genutzte Ebene auskommentiert und das Formular für den API-call `create` hinzugefügt. Durch diese Änderungen können Teilnehmer ohne Moderatorenrechte BBB-Räume öffnen und diese anschließend betreten.

12 Last- und Skalierungstests

12.1 Vorgehen

Die folgenden Untersuchungen von BBB hinsichtlich des Verhaltens des Servers unter größerer Last sowie des Skalierungsverhalten mit unterschiedlichen Prozessorkernressourcen dient der Vorbereitung auf den produktiven Einsatz und der Prognose erforderlicher Ressourcen-Voraussetzungen für bestimmte Teilnehmermengen. Im Fokus stehen dabei die Rechenlast und deren Verteilung auf die CPU, aber auch die Bandbreite, die Hauptspeichernutzung und die Stabilität des Servers.

Die Autoren von BBB stellten eine Anleitung zur Durchführung von Lasttests in ihrem Wiki zur Verfügung¹⁰. Darunter befindet sich ein Bash-Skript, das einen Log-In Vorgang auf einem BBB-Server mit Hilfe der zusätzlich zu installierenden API-Demos mit mehreren Teilnehmern auf einem PC simuliert. Es wurde für Ubuntu Desktop 10 entwickelt, funktioniert auch unter Debian 7, sofern ein aktueller Flash-Player installiert ist.

Durch das Skript wird eine Firefox-Browser-Instanz gestartet. Anschließend werden dieser Instanz durch eine while-Schleife neue Seitenaufrufe übergeben. Die Anzahl der Aufrufe bzw. die Anzahl an virtuellen Teilnehmern und die Adresse des BBB-Servers lassen sich durch Übergabeparameter steuern.

```
./bbb-test -h 212.184.75.57 -n 5
```

(Beispielaufruf des Skriptes mit 5 Teilnehmern und fester BBB-Server-IP)

Mittels Parameter `-h` wird die Serveradresse (vorzugsweise als IP) und mittels `-n` die Anzahl der zu simulierenden Teilnehmer übergeben.

Theoretisch lässt sich hierdurch mit einem System eine unbegrenzte Anzahl an Teilnehmern simulieren. Allerdings lief das benutzte Flash-Plug-In der Version 11.2.202.297 nur bis zu sieben simulierten Teilnehmern stabil.

Bei einer größeren Anzahl von Teilnehmern kam es ständig zu einem Speicherfehler unabhängig von der Hauptspeichergröße des Systems, in deren Folge alle bestehenden Sitzungen zum BBB-Server unterbrochen wurden.

Erste Testversuche wurden mit Hilfe virtueller Maschinen durchgeführt, um die Funktionen der Skripte und das Verhalten des Servers bei den Log-In Vorgängen zu beobachten. Hierfür wurde unter dem Hypervisor VirtualBox von Oracle ein Ubuntu Desktop 12.04 installiert und das Image dieses auf den Tutorien Rechnern des Instituts für duales Studium und Wissenstransfer kopiert.

Diese Vorgehensweise wurde gewählt, um den Aufwand der Beteiligten – Rechenzentrum und Rechnerpoolverwaltung – für die Haupttests gering zu halten.

¹⁰ <https://code.google.com/p/bigbluebutton/wiki/StressTesting>

Die Rechner erlaubten den gleichzeitigen Start von bis zu 4 virtuellen Maschinen mit jeweils 7 simulierten Sessions (gesamt 70 Teilnehmer, aufgeteilt auf 10 VM mit jeweils 7 Sessions), bevor der Hauptspeicher vollständig aufgebraucht war und das Hostsystem (Windows 7 Professional SP1 64 Bit) die VMs beendete.

Für diesen ersten Lasttest standen dem virtualisierten BBB-Server 2 Rechenkerne und 2 GB Hauptspeicher zur Verfügung. Bei diesem Test wurden 2 zusätzliche Rechner benutzt, um einen Moderator und einen weiteren Teilnehmer zum Abhören des Audiosignals zu simulieren.

Zuerst wurde für die Simulation eine Sprachaufnahme mit einer Länge von 20 Sekunden in einer Endlosschleife verwendet. Der Moderator-Rechner benutzte die Stereomixfunktion der integrierten Realtek-Soundkarte als Mikrofoneingang für den BBB-Flash-Client und konnte somit einen stetigen Audiostream liefern. Am zweiten Rechner wurde der Stream abgehört und die Qualität bewertet. Als Kriterien wurden Verständlichkeit und Unterbrechungsfreiheit festgelegt.

Die kurze Sprachaufnahme erwies sich als nicht praktikabel, da sich der Inhalt zu oft wiederholte und sich die Erkennung der Verständlichkeit für die Probannten als schwierig erwies. Aus diesem Grund wurde ein frei verfügbares Hörbuch als Alternative gesucht.

Die Wahl fiel auf eine Neuvertonung der zweiten Folge der drei ???® - Der Superpapagei [16] in der hohen Qualitätsstufe einer als 128 kbit/s MPEG1 Layer 3 komprimierten Audiodatei [17]. Dieses Hörbuch zeichnete sich dadurch aus, dass 19 verschiedene Sprecher mit unterschiedlichen Stimmen und Tonlagen mitwirkten. Auch die Spieldauer von 61 Minuten trug dazu bei, dass es während der Tests zu wenigen Wiederholungen kam.

Mit Hilfe der virtuellen Maschinen konnte der BBB-Server an seine Lastgrenze gebracht werden. Der Audiostream wurde mit Unterbrechungen unterschiedlicher Länge übertragen. Die CPU-Last wurde während des Tests mit dem Linux Tool top überwacht. Auffallend war dabei, dass trotz durchschnittlicher 10% Leerlaufleistung der Rechenkerne die load average Anzeigen im 1 und 5 minütigen Mittel auf Werte über 5 stiegen. Das war ein deutliches Anzeichen für Verzögerungen bei der Echtzeitverarbeitung der Audiostreams.

Die Hauptspeicherbelegung bewegte sich während der Tests an der Grenze des maximal Verfügbaren, jedoch wurde kein Teil in den Swap Bereich ausgelagert. Nachdem die Last durch Reduktion von Clients verringert wurde, so dass keine Unterbrechungen mehr wahrnehmbar waren, wurden das Videosharing Modul und die Desktopfreigabe getestet. Beides führte zu keiner weiteren Last auf dem BBB-Server, auch die Hauptspeicherbelegung änderte sich nicht.

Dieser Test des Audio- und Videostreams wurde über einen Zeitraum von 7 Stunden durchgeführt, um etwaige Speicherprobleme festzustellen. Am Ende wurden etwa 260 kB Swap Speicher durch das System genutzt und beide Streams wurden weiterhin fehlerfrei übertragen.

Somit kann davon ausgegangen werden, dass der BBB-Server auch über längere Zeiträume bei hoher Last ausreichend stabil funktioniert.

12.2 Benutzte Tools

Zur Gewinnung der Daten und späteren Analyse wurden unterschiedliche Konzepte und verschiedene Programme genutzt. Angefangen über das von den BBB-Entwicklern präferierte Jconsole und JVisualVM¹¹, über Nagios3 mit Nagiosgrapher hin zu den Kommandozeilen Programmen sar, bwm-ng, free und netstat. Am Anfang stand das Ziel, direkt auswertbare Grafiken vom eingesetzten Tool zu erhalten. Dieses wurde im Laufe der Arbeit aufgegeben. Die verwendeten Programme konnten nicht die gewünschten Daten zur CPU-Auslastung, zur Hauptspeichernutzung in Verbindung mit der Anzahl verbundenen Clients sowie die benutzte Netzwerkbandbreite übersichtlich darstellen. Die Daten der java-basierten Tools Jconsole und JVisualVM enthielten ausschließlich Informationen über die von Red5 genutzten Ressourcen. Damit fehlte die Hauptrechenlast, verursacht durch FreeSWITCH, in den Darstellungen. Die somit gewonnenen Daten waren nicht aussagekräftig hinsichtlich der tatsächlichen Belastung des Servers.

Nach einer Internetrecherche über empfehlenswerte Analysetools für Webserver unter Ubuntu Server 12 fiel die Entscheidung auf das Monitoring Framework Nagios3. Dessen Deployment gestaltete sich schwierig, obwohl fertige Pakete im Ubuntu Repository vorhanden waren. Die Installation erforderte eine mit BBB inkompatible Version der libjpeg-dev Bibliothek (BBB nutzt 6.2 und Nagios erforderte min 8.0). Eine über Debian GNU/Linux package manager (dpkg) erzwungene Aktualisierung führte zu einem instabilen System, welches nur durch ein Einspielen des vorher angefertigten Backups zu beheben war. Anschließend wurde vom ersten Ziel Abstand genommen und eine einfache und effiziente Methode zur Aufzeichnung der gewünschten Daten gesucht. Eine weitere Recherche unter Einbeziehung der Meinung von beteiligten Kollegen führte zur Verwendung von bwm-ng (Bandbreitenmonitor mit csv Ausgabe), sar (Ausgabe der CPU Auslastung mit Zeitstempel), free (Hauptspeicherbelegung) und einer Kombination aus netstat (Netzwerkverbindungsmonitor) und wc (Zeilenzähler).

```
#logging.sh v1.2
#!/bin/bash

NOW=$(date +"%F"_"%H"-"%M"-"%S")
LOGFILE="log-$NOW.log"
sar 1 > "$LOGFILE".cpu &
bwm-ng -o csv -F "$LOGFILE".net &
./cpustat 5020 4859 5143 > "$LOGFILE".cpustat &

echo "$LOGFILE"

while true
do
    date +%-d/%m/%y\ %H:%M:%S >> "$LOGFILE"
    free | grep -v + >> "$LOGFILE"
    netstat -a | grep ESTABLISHED | grep 1935 > temp_connections
    date +%-d/%m/%y\ %H:%M:%S >> "$LOGFILE".con
    wc -l temp_connections >> "$LOGFILE".con
```

¹¹ <https://code.google.com/p/bigbluebutton/wiki/SettingUpBigBlueButtonWithJconsole>

```

    sleep 1
done

(Inhalt logging.sh v1.2)
```

Die Zeile mit dem Aufruf des cpustat Skriptes wurde als Ergebnis des zweiten Lasttests hinzugefügt, um die Auslastung der einzelnen Kerne für zu übergebende Prozess-IDs zu erfassen.

```

#!/bin/bash

pids=()
while [ $# != 0 ]; do
    pids=("${pids[@]}" "$1")
    shift
done

if [ -z "${pids[0]}" ]; then
    echo "Usage: $0 <pid1> [pid2] ..."
    exit 1
fi

for pid in "${pids[@]}"; do
    if [ ! -e /proc/$pid ]; then
        echo "Error: pid $pid doesn't exist"
        exit 1
    fi
done

while [ true ]; do
    date +%d/%m/%y\ %H:%M:%S
    for pid in "${pids[@]}"; do
        ps -p $pid -L -o pid,tid,psr,pcpu,comm=
    done
    sleep 1
done

(Inhalt cpustat)
```

Das Skript erfasst alle Kindprozesse zu den übergebenen Prozess-IDs und schreibt deren CPU-Auslastung pro Kern in eine Logdatei.

Für die grafische Auswertung wurde OriginPro 9 von Origin Lab¹² genutzt. Das Programm für wissenschaftliche Analysen bietet einen programmierbaren Importdialog für Textdateien sowie komplexe Darstellungsoptionen der numerischen Daten. Für die Analyse wurden die

¹² <http://www.originlab.de/>

Logdaten der CPU-Last "\$LOGFILE".cpu sowie die Anzahl der Verbindungen "\$LOGFILE".con genutzt. In letzterem wurde vor dem Import der Zeilenumbruch zwischen Zeitstempel und der Verbindungsanzahl mittels Ersetzen im Notepad++ über den regulären Ausdruck [0-9]\K\n durch ein Leerzeichen ersetzt.

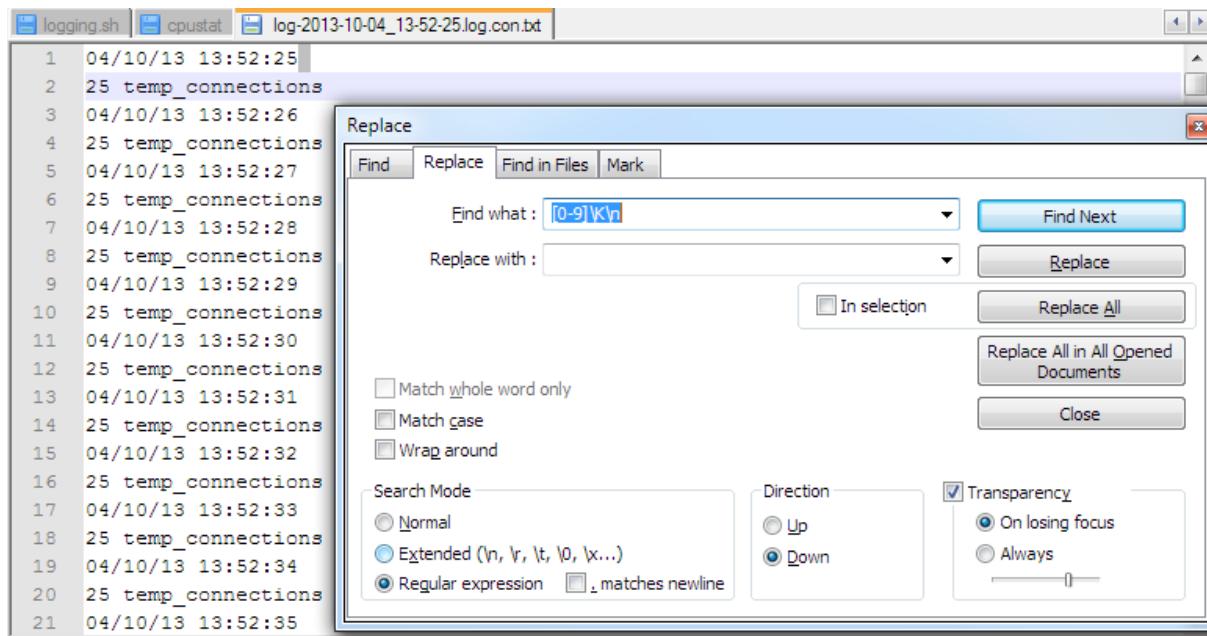


Abbildung 12-1 Notepad++ Replacement-Dialog

Das Bild zeigt den Dialog im Notepad++ und im Hintergrund die per regulären Ausdruck gefundene Stelle. Mittels Replace All werden alle ungewünschten Zeilenumbrüche entfernt. Im Anschluss lassen sich beide Logdateien über den Importassistenten von OriginPro einfach importieren. Das Programm erkennt zuverlässig die tabellenartige Struktur sowie Kommentare am Anfang der Datei. Die getätigten Importeinstellungen zu den Datentypen (zum Beispiel Zeit und Datum, englisches Dezimalzeichen) lassen sich als Profil speichern und bei weiteren Importvorgängen als Vorlage nutzen. Somit ist es möglich, eine größere Anzahl an identisch aufgebauten Logdateien zu importieren.

Linux 3.2.0-53-generic (testing12)		10/04/2013		_x86_64_		(6 CPU)	
<hr/>							
01:52:25 PM	CPU	%user	%nice	%system	%iowait	%steal	%idle
01:52:26 PM	all	0.17	0.00	1.20	0.00	0.00	98.62
01:52:27 PM	all	0.34	0.00	1.03	0.00	0.00	98.62

(Auszug CPU-Log Datei)

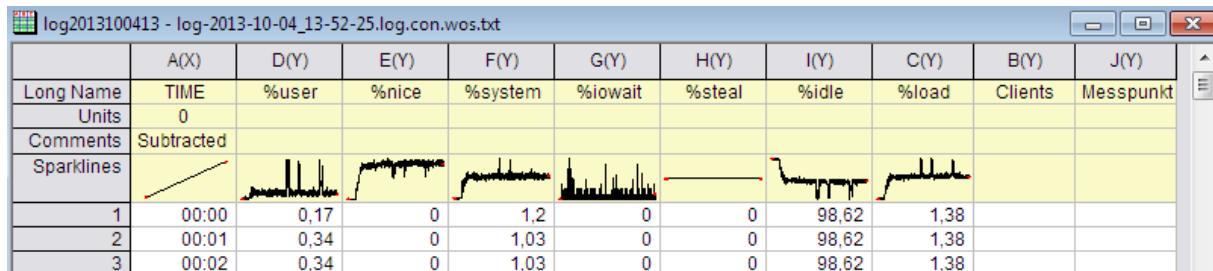


Abbildung 12-2 OriginPro 9 Importergebnis

Im Anschluss werden die für die Darstellung benötigten Spalten markiert und daraus ein Diagramm erstellt. Als Methode zur Lastdarstellung wurde ein summiertes Flächendiagramm aus den Spalten `%user`, `%nice` und `%system` erstellt. Die verbunden Clients werden in einem weiteren Darstellungslayer des gleichen Diagramms mit eigener Y-Achse und Skalierung als Liniendiagramm mit ausgewählten Messpunkten dargestellt. Alle vorgenommenen Einstellungen lassen sich als Vorlage für weitere Diagramme speichern und somit den Erstellungsprozess vereinfachen.

12.3 Ergebnisse

Der zweite Teil der Lasttests fand in Abstimmung mit dem Rechenzentrum und dem Support der Computerpools statt. Einem ersten Funktionstest mit 4 Rechenkernen zur Prüfung der Logging-, der Startskripte der Pools und zur Abschätzung der benötigten Rechenleistung folgte der Haupttest mit 12 Rechenkernen und 4 GB Hauptspeicher. Es handelte sich um einen virtualisierten Ubuntu-Server, dessen Ressourcen durch einen KVM-Hypervisor gesteuert werden. Die Hauptprozessoren sind zwei Xeon E7-2860 aus der Westmere-EX Baureihe mit jeweils 10 realen CPU-Kernen und einer Taktfrequenz von 2.26 GHz¹³. Der Ubuntu Server musste während der gesamten Testperiode ausschließlich die Rechenlast von BBB bewältigen, alle weiteren Aufgaben übernahm der Hypervisor, der auf bis 6 zusätzliche Kerne der CPUs zurückgreifen konnte. In einer realen Workumgebung ist die zu erwartende Rechenlast des Servers etwas höher, da zusätzliche Rechenarbeit für Managementaufgaben anfällt.

¹³ <http://ark.intel.com/products/53570>

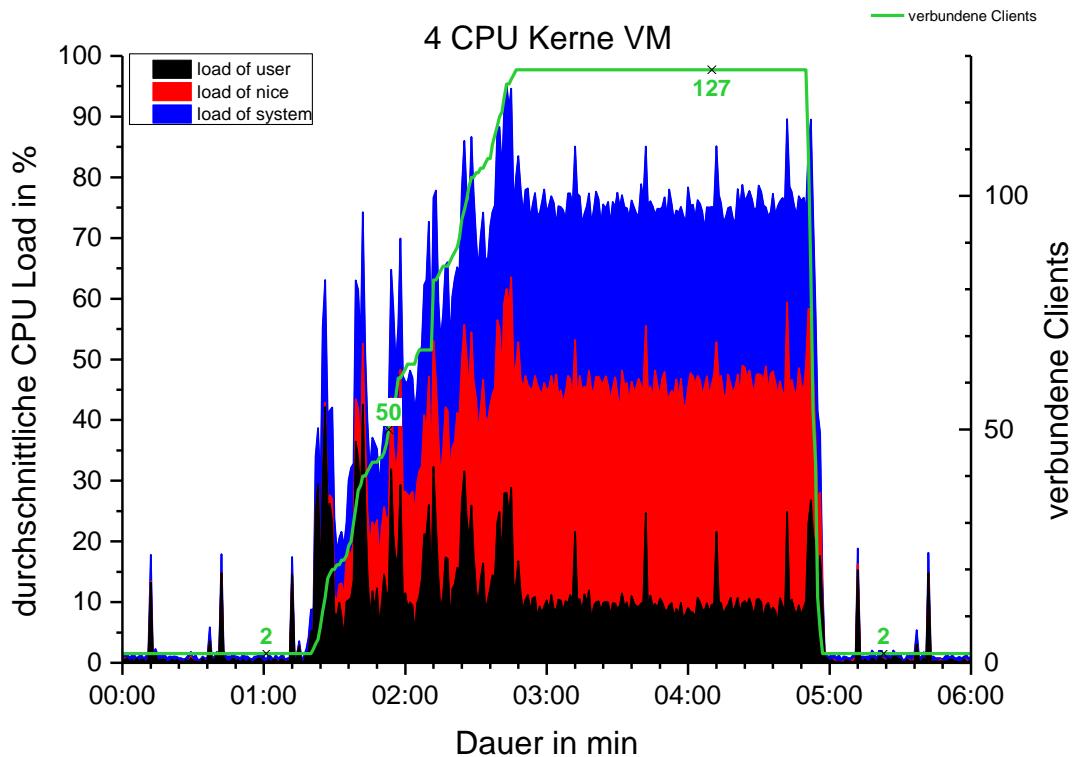


Abbildung 12-3 Lasttest mit 4 CPU Kernen

Die Abbildung 12-3 zeigt den ersten Lasttest mit maximal 127 gleichzeitig verbunden Clients. Auffällig sind die auftretenden Lastspitzen während des Verbindungsauflaufs der Clients. Hauptverantwortliche hierfür sind der Webserver nginx sowie tomcat als Application Server für die grails Anwendung bbb-web, um den Raumbeitritt zu steuern. Der Audiostreams war in diesem Test nicht mehr unterbrechungsfrei obwohl das System noch nicht zu 100% ausgelastet war. Unter der Rubrik load of nice ist einzig der FreeSWITCH Prozess zu finden, da dieser mit einem nice Wert von 19 von BBB gestartet wird. Alle weiteren BBB-Prozesse laufen ohne diese Einschränkung. Load of System wird in erster Linie durch den Red5 Java Prozess erzeugt. Unter Load of User sind alle weiteren BBB Komponenten (redis, tomcat, nginx, pdf2swf, etc.) zusammengefasst. Die auftretenden Lastspitzen nach dem Loginvorgang (ab etwa 02:00) stammen vom Ruby Hauptprozess.

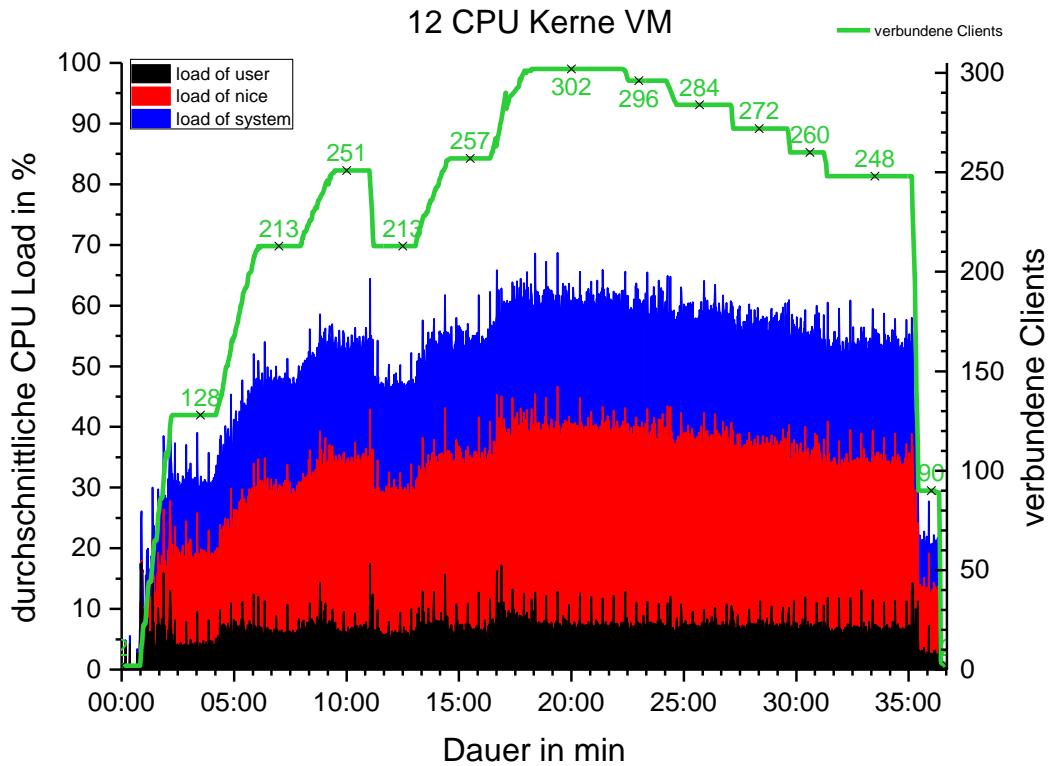


Abbildung 12-4 Lasttest mit 12 CPU Kernen

Die Abbildung 12-4 stellt den 36min andauernden Stresstest mit der maximal erreichten Anzahl an simulierten Clients aus den Computerpools dar. Jeder PC aus den Pools sollte 5 Clients simulieren. Tatsächlich variierte die Clientanzahl leicht, so dass 4 - 6 Clients gestartet wurden. Das vorher festgelegte Ziel von mindestens 300 Clients wurde erreicht. Trotz ausreichend vorhandener ungenutzter Rechenleistung war der Audiostream zu diesem Zeitpunkt (ab 18:00) nicht unterbrechungsfrei. Nach kurzer Analyse mit top zeigte sich die Ursache. Während fast alle CPU Kerne über mehr als 20% idle Zeit verfügten, war Kern 0 zu 100% ausgelastet. Der betreffende BBB-Prozess konnte nicht ermittelt werden, aber nach schrittweiser Absenkung der Teilnehmerzahl auf 248 verschwanden die Unterbrechungen im Audiostream vollständig.

Die Schlussfolgerung daraus lautet, dass BBB eine Komponente enthält, die nicht auf mehrere Kerne aufgeteilt werden kann. Für die Ermittlung der benötigten Hardware muss auch die Rechenleistung per Kern betrachtet werden und nicht ausschließlich die Gesamtrechenleistung des Servers. Ein weiterer Test mit halbierter Kern- und Clientanzahl vermittelte einen Ausblick auf möglichen load balancing, d.h. Aufteilung von Rechenressourcen.

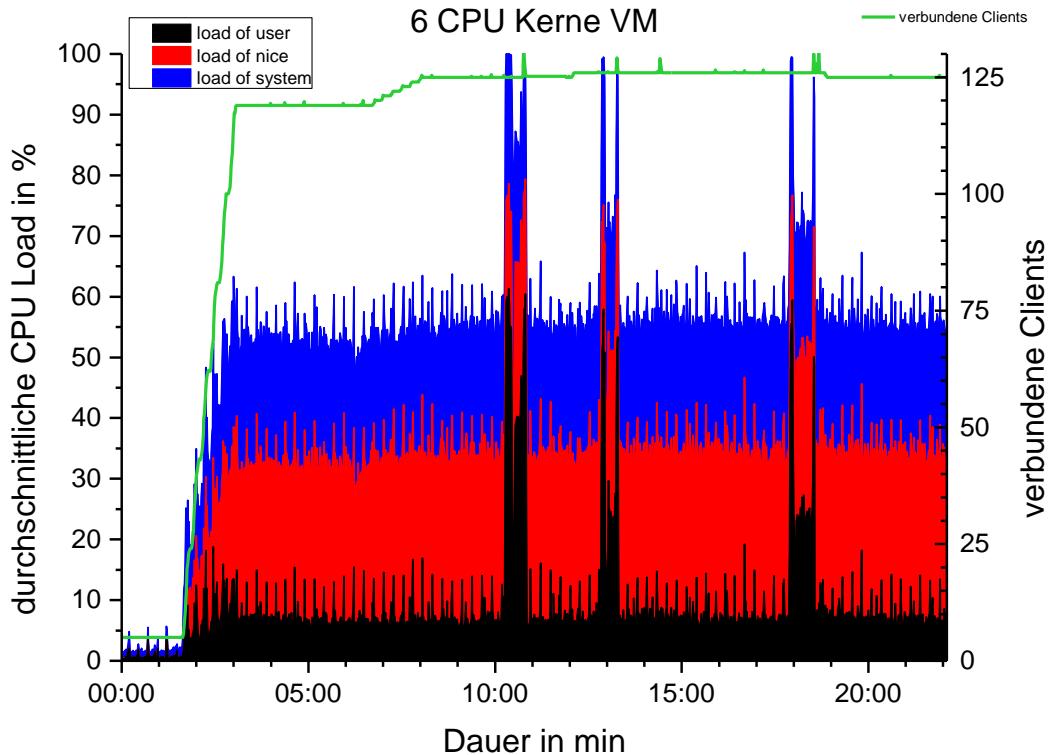


Abbildung 12-5 Lasttest mit 6 CPU Kernen

Abbildung 12-5 zeigt den durchgeföhrten Test mit halbierter Last und Ressourcen. Die Gesamtrechenlast ist vergleichbar mit dem vorherigen Test, jedoch war der Audiostream mit den 127 verbundenen Clients unterbrechungsfrei. CPU Kern 0 hatte etwas mehr als 10% idle Rechenzeit und damit genügend Ressourcen bei kurzen Laständerungen. Die Grafik zeigt 3 zusätzliche Ausschläge der Rechenlast im User Bereich (10:30, 13:00, 18:00). Hier wurden Dokumente in das Whiteboard geladen. Diese musste der Server in das SWF-Format umwandeln und dann an die Clients ausliefern. Dieser Vorgang erzeugte eine hohe Rechenlast, und der Audiostream wurde unterbrochen.

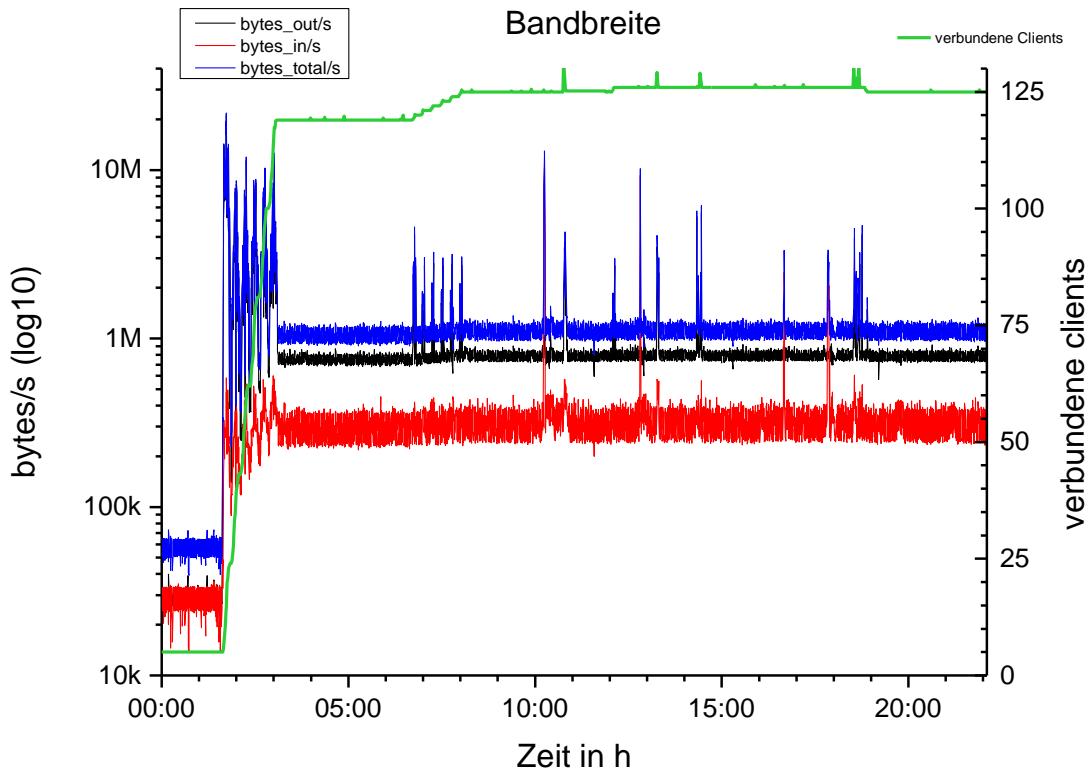


Abbildung 12-6 Lasttest mit 6 CPU Kernen (Bandbreite)

Das Diagramm der Abbildung 12-6 stellt die benötigte Bandbreite der 127 verbunden Clients dar. Auffallend ist der etwa dreifache Unterschied zwischen eingehender und ausgehender Datenmenge. Bei den Lasttests wurde immer nur ein Moderator simuliert, also nur ein eingehender Audiostream pro Raum. Dennoch war für alle Tests das Verhalten identisch.

BBB überträgt auch dann Audiodaten per RTMP, wenn der Client keinen Input liefert. Die Ausschläge der logarithmisch eingeteilten bytes/s Achse zeigen die erwähnten Dokumentuploads sowie den Login-Vorgang der Clients.

Sie entstehen durch die Hochrechnung von bwm-ng bei großen Datenpaketen. Pro Client fallen im Durchschnitt 7,6kB/s Daten an (1,9kB/s in, 5,7kB/s out).

12.4 Aufgetretene Probleme

Während der Lasttests traten verschiedene Probleme auf. Diese betreffen alle beteiligten Komponenten.

Ein Beispiel ist der Absturz des BBB Flash-Plug-Ins bei mehr als 7 simulierten Clients (Browsers) aufgrund eines Speicherfehlers. Dieser konnte im Rahmen nicht weiter untersucht werden. Ein weiteres Problem ist die benötigte Rechenlast des Clients bei Verwendung der höchstmöglichen Auflösung des Videostreams. Hierbei kann es auch zu Unterbrechungen des Audiostreams kommen, da der Client nicht beide Stream in Echtzeit zur Verfügung stellen

kann. Das BBB-Flash Plug-In kann auch bei mehreren Videostreams abstürzen, wenn nicht genügend Ressourcen (meist Hauptspeicher) zur Verfügung stehen, obwohl dieser nur kleine Videofenster empfängt und keine weiteren Daten überträgt.

Auch beim Lasttest kam es zu Problemen. Die Computer in den Pools starten trotz identischer Hardware und gleichem Betriebssystem mit unterschiedlicher Anzahlen von Clients (4 – 6). Der Grund konnte nicht näher bestimmt werden. Daher waren die Tests nicht mit mehrfachen Durchläufen reproduzierbar und bleiben Einzelmessungen.

Ein Problem liegt im BBB-Server selbst. Bei einem weiteren Testdurchlauf wurde überprüft, ob Unterschiede in der Serverlast bestehen, wenn sich die Clients nicht alle in einem gemeinsamen Raum befinden. Dafür wurden die 127 Clients in 5 verschiedene Räume mit jeweils einem Moderator aufgeteilt und anschließend die Auslastung des Servers mittels top betrachtet. Dabei existiert kein messbarer Unterschied. Beim anschließenden Test zur Bereitstellung eines Dokumentes wurde aber nicht nur im Raum, indem der Upload stattfand, sondern in allen Räumen auf dem BBB-Server der Audiostream für die Dauer der Datenumwandlung unterbrochen. Für den Produktivbetrieb ist ein derartiger Zustand ungenügend. Das betreffende Modul sollte auf einen dedizierten Server ausgelagert werden. Ein ähnliches Verhalten zeigt der Server beim Start eines Videostreams.

13 Wirtschaftlichkeitsanalyse

Die Wirtschaftlichkeitsanalyse wurde auf drei am Markt verfügbare und vergleichbare Produkte beschränkt. Dabei wurden BigBlueButton, Saba Centra® eClassroom und Adobe Connect in House Hosting betrachtet.

Diese drei Produkte sind in ihrem Leistungsumfang weitgehend vergleichbar. Sie besitzen ein Whiteboard und der Dokumentenaustausch über dieses ist mit gleichzeitiger Bearbeitungsfunktion möglich. Weiterhin verfügen sie über Audioübertragungsfunktionen ähnlich einer Telefonkonferenz. Bei Bedarf können Videostreams genutzt werden.

13.1 Aufstellung Kosten BigBlueButton

Die Kosten für BBB gliedern sich in zwei Bereiche. Das sind zum einen die Kosten für die Hardware und zum anderen die Personalaufwendungen. Die Software ist lizenzkostenfrei nutzbar.

Entsprechend einem kürzlich erhaltenen Angebot der Firma Dell Deutschland GmbH, Bereich Geschäftskunden, ergeben sich nachstehend aufgeführte Serverkosten:

Power Edge R620 Rack-Server	
mit 2x 12 Kernprozessoren (2x Xeon E5-2697 v2 mit 2,7 GHz)	
(entspricht einer Rechenleistung von 600 gleichzeitigen Usern	
bei Aufteilung in 4 VM mit jeweils 6 Kernen)	12.000,00 Euro/Server

Für die Personalkosten wurde folgende Kalkulation erstellt:

Inbetriebnahmekosten je Server	
1 Mitarbeiter a 102 Arbeitsstunden (3 Wochen)	
mit einem Stundensatz von 125,00 Euro	12.750,00 Euro

Instandhaltungskosten	
2 Arbeitsstunden je Woche a 125 €	
52 Wochen pro Jahr	13.000,00 Euro

Gesamtkosten im ersten Jahr: 37.750,00 Euro

In den Folgejahren werden nur die Kosten für Instandhaltung und Wartung wirksam.

13.2 Vergleich mit Saba Centra® eClassroom

Saba Centra® eClassroom ist ein durch die Deutsche Telekom AG zentral eingekauftes Produkt, welches an die HfTL kostenpflichtig weiter lizenziert wird.

Die Hochschule nutzt dabei das Concurrent-User-Lizenzmodell. Bei diesem Lizenzmodell wird die maximale Anzahl der User festgelegt, die gleichzeitig auf die Anwendung zugreifen können.

Die Lizenzkosten betragen 1072,00 Euro pro Lizenz und Jahr.

bei gleichzeitig 600 Usern und 1072,00 Euro/Lizenz	643.200,00 Euro
--	-----------------

13.3 Vergleich mit Adobe Connect in House Hosting

Adobe Connect Enterprise Server

Serverkosten (ca. 75% der Kosten von BBB)	9.000,00 Euro
---	---------------

einmalige Lizenzkosten der Adobe Software	5.500,00 Euro
---	---------------

einmalige Concurrent-User-Kosten von 188,00 Euro/User	
---	--

bei 600 Usern	112.800,00 Euro
---------------	-----------------

Jährliche Supportkosten	8.600,00 Euro
-------------------------	---------------

einmalige Installationskosten	4.000,00 Euro
-------------------------------	---------------

Gesamtkosten im ersten Jahr:	139.900,00 Euro
------------------------------	-----------------

In den Folgejahren werden nur die Kosten für Support wirksam.

13.4 Fazit

Beim Vergleich der Gesamtkosten im ersten Jahr ist ersichtlich, dass der BBB-Server einen deutlichen Vorteil gegenüber den betrachteten Mitbewerbern hat. Dies resultiert vor allem aus der lizenzkostenfreien Nutzung von BBB. Die gegenwärtig genutzte Variante verursacht die höchsten jährlichen Kosten.

Da Bandbreite der Internetanbindung für die Hochschule (HfTL) beschränkt ist, muss nach gegenwärtigem Kenntnisstand davon ausgegangen werden, dass die gleichzeitige Nutzung durch 600 User bei in-House Hosting (BBB und Adobe) nicht gewährleistet ist, da ein Dienst über 50% der Gesamtbandbreite nur durch die Audiostreams benötigt.

Vor einem Produktwechsel ist daher eine Investition zum Ausbau der Hochschulanbindung notwendig. Als Alternative könnte auch ein extern betriebener Server in Betracht gezogen werden.

14 Literatur-/Quellenverzeichnis

- [1] F. Dixon: Google Groups BigBlueButton-dev: Audio stress Test: Results. 24 03 2011. Internet: URL <https://groups.google.com/forum/?fromgroups=&hl=de#!topic/bigbluebutton-dev/PHzpgVqYAMM> (Stand 24.06.2013)
- [2] Terence: BigBlueButton Installation auf OpenSuse 11.4. 19 06 2012. Internet: URL <http://bbbsuse.blogspot.de/> (Stand 16.08.2013)
- [3] F. Dixon: InstallingBigBlueButton. BigBlueButton, 17 06 2012. Internet: URL <https://code.google.com/p/bigbluebutton/wiki/InstallingBigBlueButton> (Stand 14.08.2013)
- [4] andres: BigBlueButton Ilias Plugin Alpha. 09 05 2011. Internet: URL <http://sourceforge.net/projects/bbb-ilias-plug/> (Stand 12.06.2013)
- [5] BigBlueButton Inc. History of BigBlueButton. BigBlueButton Inc, Internet: URL <http://bigbluebutton.org/history/> (Stand 14.11.2013)
- [6] F. Dixon, BigBlueButton Inc: ArchitectureOverview. 07 11 2013. Internet: URL http://bigbluebutton.googlecode.com/svn/trunk/bbb-images/arch_Diagrams/architecture_diagram_08.png (Stand 31.12.2013)
- [7] Ruby-Community.Ruby About. Ruby Community, Internet: URL <https://www.ruby-lang.org/de/about/> (Stand 23.01.2014)
- [8] osflash.RTMP. 24 02 2013. Internet: URL <http://osflash.org/documentation/rtmp> (Stand 07.06.2013)
- [9] NiTenIchiRyu: Red5. wikipedia, 15 01 2014. Internet: URL <http://de.wikipedia.org/wiki/Red5> (Stand 18.01.2014)
- [10] Thwrz: Liste von VoIP-Software. wikipedia, 16 01 2014. Internet: URL http://de.wikipedia.org/wiki/Liste_von_VoIP-Software (Stand 16.01.2014)
- [11] NY Solutions: FreeSWITCH Wiki. FreeSWITCH, 13 12 2013. Internet: URL https://wiki.freeswitch.org/wiki/Main_Page (Stand 01.01.2014)
- [12] Mrlabs: Ubuntu. wikipedia, 04 01 2014. Internet: URL <http://de.wikipedia.org/wiki/Ubuntu> (Stand 04.01.2014)

- [13] a. waheed: Installation of Bigbluebutton on Ubuntu 12.04LTS(successful). Google Groups BigBlueButton-dev, 24 04 2013. Internet: URL <https://groups.google.com/forum/?fromgroups=&hl=de#!topic/bigbluebutton-dev/cIGUQBLiSoU> (Stand 18.01.2014)
- [14] sebastianz1980: BigBlueButton ILIAS 4.3 ILIAS Plugin for BigBlueButton. 21 02 2013. Internet: URL <http://sourceforge.net/projects/bigbluebuttonil/> (Stand 21.06.2013)
- [15] BigBlueButton Inc.API. BigBlueButton Inc, 23 03 2013. Internet: URL <https://code.google.com/p/bigbluebutton/wiki/API> (Stand 13.09.2013)
- [16] T. Walter: Die drei ??? Neuvertonung. Neuvertonung, Internet: URL <http://fragezeichen.neuvertonung.de/downloads.php#2> (Stand 26.08.2013)
- [17] Neuvertonung.Die drei ??? Neuvertonung - 2 - Die drei ???® und der Super-Papagei. 08 09 2006. Internet: URL <http://fragezeichen.neuvertonung.de/hoerspiele/2DerSuperpapagei.mp3> (Stand 30.08.2013)
- [18] H. Krahn: Virtueller Campus '99: heute Experiment - morgen Alltag? Band 9 von Medien in der Wissenschaft. Waxman Verlag, 2000.
- [19] Stephan13360: nginx > wiki. ubuntuusers, 24 12 2013. Internet: URL <http://wiki.ubuntuusers.de/nginx#Installation> (Stand 05.01.2014)
- [20] wikipedia.FreeSWITCH. 31 10 2013. Internet: URL <http://en.wikipedia.org/wiki/FreeSWITCH> (Stand 12.01.2014)
- [21] F. Dixon: Stress Testing. BigBlueButton, 03 11 2011. Internet: URL <https://code.google.com/p/bigbluebutton/wiki/StressTesting> (Stand 26.06.2013)
- [22] Adobe Systems Incorporated. Real-Time Messaging Protocol (RTMP) specification. Internet: URL <http://www.adobe.com/devnet/rtmp.html> (Stand 15.09.2013)
- [23] R. Alam: LinkedIn Profil Richard Alam. LinkedIn Corporation, Internet: URL <http://ca.linkedin.com/pub/richard-alam/5/b74/a91> (Stand 30.12.2013)

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir an der Hochschule für Telekommunikation Leipzig (FH) eingereichte Abschlussarbeit zum Thema

Untersuchung der Integration des Konferenzsystems BigBlueButton in die Infrastruktur der Hochschule für Telekommunikation Leipzig

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Leipzig, 31. Januar 2014

Udo Siebert