

FH JOANNEUM - University of Applied Sciences

GreenWeb Prototype

Prototype-Entwicklung für Energie- und Performance-Messungen

Bereichsübergreifende Projektarbeit (BUEPR)

Verfasser:

Colin Jochum, Tom Kleinhapl, Marcel Kahr

Betreuer:

DI Georg Mittenecker

Graz, 20.12.2019

FH | JOANNEUM
University of Applied Sciences

Inhaltsverzeichnis

1	Abkürzungsverzeichnis	5
2	Einleitung	7
2.1	Zielsetzung	7
2.2	Struktur und Aufbau	7
3	Allgemeines	9
4	Lösungsansätze von Institutionen für Green Web/IT	10
5	Netzwerkstromverbrauch	12
6	Infrastruktur	14
7	Hardware	15
7.1	HP Elite Desk 800 G1 TWR	15
7.2	HP Elite Desk 800 G2 TWR	15
7.2.1	VM1 – Greta Debian	16
7.2.2	VM2 – Greta Ubuntu	16
7.2.3	VM3 – Greta Ubuntu 2	16
7.3	HP-Server G6	16
7.4	Linksys Router	16
8	Software	18
8.1	Open Broadcaster Software (OBS)	18
8.2	NGINX	18
8.2.1	NGINX für RTMP Streaming	19
8.2.2	NGINX Installation inkl. RTMP Modul	19
8.3	ffmpeg	23
8.4	Snapd	25
9	Mess-Tools	26
9.1	Intel Power Gadget	27
9.2	HP ILO	27
9.3	TP LINK HS110	28
9.3.1	TP-LINK HS110 Parameter	29
10	Software Tools	30
10.1	Apache	30
10.2	MySQL	30
10.3	JMeter	30

10.4	XAMPP.....	33
10.5	XCP-ng Center.....	33
11	Messzenario 1 - Streaming	35
11.1	Messungen Streaming Server (HP Elite Desk 8300 G2)	35
11.1.1	Resultate und Analyse	36
11.1.2	Diagramme.....	40
11.2	Messungen Streaming Server (HP ProLiant G6).....	50
11.3	Resultate und Analyse.....	51
11.4	Messungen Diagramme	56
12	Messszenario 2 – Belastungstest.....	61
12.1	Streamingserver HP Elite Desk 8300 G2	61
12.1.1	Resultate und Analyse	62
12.1.2	Messdiagramme	64
13	Probleme & Lessons Learned.....	73
13.1	OS Wahl.....	73
13.2	NGINX Konfiguration	73
13.3	JMETER	73
13.4	Zeitmanagement	73
13.5	Konfiguration & Installation von Diensten.....	73
14	Zusammenfassung / Conclusion	74
15	Projektantrag	76
16	Meilensteine	78
17	Meilensteintrendanalyse	79
18	Projektplan.....	80
19	Projektstrukturplan.....	81
20	Gantt Chart.....	82
21	Stundenerfassung Marcel Kahr.....	83
22	Stundenerfassung Colin Jochum.....	84
23	Stundenerfassung Tom Kleinhapl	85
24	Tätigkeitsbericht	86
25	Zugangsdaten.....	97
26	IP Konzept	98
27	Abbildungsverzeichnis	99

28	Messdiagrammverzeichnis	100
29	Tabellenverzeichnis.....	102

1 Abkürzungsverzeichnis

1080p	1920 x 1080 Pixel Auflösung
4k	3840 x 2160 Pixel Auflösung
AAC	Advanced Audio Coding
BIOS	Basic Input/Output System
BUEPR	Bereichsübergreifende Projektarbeit
CO ₂	Kohlenstoffdioxid
CPU	Central Processing Unit
CRF	Constant Rate Factor
CSV	Comma Separated Values
DDos	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
EPEL	Extra Packages for Enterprise Linux
FPS	Frames Per Second
Ffmpeg	Fast Forward MPEG
Full HD	Full High Definition (1080p)
GB	Gigabyte
GHz	Gigahertz
GUI	Graphical User Interface
HD	High Definition (720p)
HLS	Http Live Streaming
HP	Hewlett-Packard
ILO	Integrated Lights-Out
IP	Internet Protokoll
IT	Informationstechnik
JSON	JavaScript Object Notation
kWh	Kilowattstunden
MHz	Megahertz
OBS	Open Broadcaster Software

OS	Operating System
PC	Personal Computer
PHP	PHP: Hypertext Preprocessor
PVC	Polyvinylchlorid
RAM	Random Access Memory
RTMP	Real Time Messaging Protocol
SCP	Secure Copy Protocol
SSH	Secure Shell
URL	Uniform Resource Locator
VM	Virtuelle Maschine

2 Einleitung

In der Lehrveranstaltung „Bereichsübergreifende Projektarbeit“ kurz BUEPR, welche im 5. Semester des Studiengang Informationsmanagement absolviert werden muss, entschieden wir uns für das Thema GreenWeb Prototype. Dieses Thema wurde von Herrn DI Georg Mittenecker in Kooperation mit Herrn Gerhard Sprung, MSc, welcher das Thema GreenWeb Visualisierung betreut, erstellt. Dabei handelt es sich um die Entwicklung eines Prototyps zur Energie- und Performance-Messungen. Nach dem Kickoff-Meeting und ausführlicher Absprache im Projektteam, kamen wir zum Entschluss, Energieverbrauchsmessungen sowohl serverseitig als auch clientseitig beim Streamen von Videos umzusetzen. Dabei wird der Energieverbrauch diverser Qualitätsstandards, Verschlüsselungstechniken, als auch der Zugriff auf dynamische und statische Webseiten betrachtet. Da es im Bereich GreenWeb zwei Projektgruppen gibt, tauschen wir die gemessenen Daten mit dem anderen Team aus, damit diese etwaigen Visualisierungen umsetzen können.

2.1 Zielsetzung

Die Zielsetzung der Projektarbeit ist es eine funktionierende Netzwerkinfrastruktur aufzubauen, um Stromverbrauchsmessungen durchführen zu können. Dabei werden zwei verschiedene CPU Generationen, ein HP-Server, sowie die eigenen Notebooks der Studenten verwendet, um aussagekräftige Ergebnisse zu erzielen. Die Systeme werden zum Teil physisch auf dem OS aufgesetzt, jedoch zum überwiegenden Teil in virtualisierter Form (VMs). Weiters soll anhand diverser Tools und Schnittstellen der Energieverbrauch ermittelt, ausgelesen und untereinander verglichen werden. Die einzelnen Tools bzw. Schnittstellen werden in einem späteren Kapitel genauer betrachtet. Bei den Messungen werden natürlich diverse Einstellungen, wie unterschiedliche Qualitätsstandards (1080p, 4k), Verschlüsselungstechniken oder Codecs, berücksichtigt. Durch diese Einstellungen soll auch veranschaulicht werden, welche Unterschiede diese bewirken und ob mehr oder weniger Energie dafür benötigt wird. Weiters sollen die Messergebnisse aussagekräftig und lesbar sein, damit das Partnerteam mit den Ergebnissen anschauliche Visualisierungen generieren kann. Neben der praktischen Umsetzung und Analyse geht es auch darum, durch Recherchen auf schon bekannte bzw. in Aussicht befindliche Lösungsansätze von Institutionen wie Facebook und Co. für GreenWeb/IT zu finden und zu verweisen.

2.2 Struktur und Aufbau

Anhand der vordefinierten Meilensteine wird nach dem Projektstart und dem offiziellen Kickoff-Meeting zuerst der Aufbau der Netzwerkinfrastruktur und die Installation diverser Softwarekomponenten wie Windows Server 2019, Native Linux und andere Linux Distributionen für die virtuellen Maschinen, durchgeführt. Weiteres wird nach einem geeigneten Webserver sowie etwaige Tools zum Messen des Energieverbrauchs recherchiert. Diese Tools werden auf ihre Kompatibilität für die etwaigen in Einsatz befindlichen System getestet, um damit aussagekräftige Messungen durchführen zu können. Nach erfolgreichem Aufbau der Infrastruktur und testen diverser Tools werden erste Probetests und Messungen

durchgeführt. Weiters wird ein Plan erstellt, in dem eingetragen wird, welche Systeme wie und mit welchem Tool untereinander verglichen werden. Nach erfolgreicher Dokumentation der Messergebnisse werden wir ein finales Abschlussmeeting mit unserem Projektbetreuer ansetzen. Die Messergebnisse werden mit dem Partnerteam geteilt und eine Gesamtdokumentation des Projektes wird verfasst. Zu guter Letzt erfolgt die Projektpräsentation sowie die Abgabe des Projektes.

3 Allgemeines

Greenpeace veröffentlichte eine Studie namens „Clicking Green“ und verdeutlichte, dass wenn das Internet ein Land wäre, den sechstgrößten Energieverbrauch der Welt hätte.

Die größten Stromverbraucher können grundsätzlich in vier große Gruppe unterteilt werden:

1. PCs, Laptops, Smartphones, Tablets, Spielekonsolen
2. Server in Daten- und Rechenzentren
3. Mobilfunkstationen und Router
4. Herstellung der Hardware für die Punkte Eins, Zwei und Drei

Laut Google soll das Suchen nach „Wieviel Strom verbraucht einmal „Googlen“?“, rund 0,3 Wattstunden verbrauchen. Das bedeutet, dass mit rund 200 Suchanfragen ein Hemd gebügelt oder mit 3000 Suchanfragen ein Eimer Wasser zum Kochen gebracht werden kann.

Die digitalen Technologien verursachen rund vier Prozent der gesamten Treibhausgasemission, was sogar mehr ist, als die Luftfahrt produziert.

Rund 80% des weltweiten digitalen Datenverkehrs entsteht beim Übertragen von Videos. Dieser Datenverkehr kann in drei große Bereiche eingegliedert werden. Der größte Teil ist das Streamen von Onlinevideos. Das Übertragen von Onlinevideos (60% des Weltdatenverkehrs) machte im Jahr 2018 rund 1,05 Zettabytes auf. Das sind rund $1,127429 \times 10^9$ Terabytes, was für Menschen nahezu unvorstellbar ist. Dabei entstanden rund 306 Millionen Tonnen an CO₂, was etwa 20 Prozent der Treibhausgasemission im Bereich der digitalen Technologie ausmacht. Weltweit gesehen sind das trotzdem nur gut ein Prozent der gesamten Treibhausgasemission.

Der Bereich des Online Videos kann in vier weitere Teilbereiche unterteilt werden. 34 Prozent wird von Video on Demand, das Streamen von Filmen und Videos über Netflix und Amazon Prime, verursacht. Weitere 27 Prozent fallen auf das Streamen von pornografischen Inhalten. Der dritte Teilbereich bezeichnet sich als „Tube“ und beinhaltet unterschiedliche Videostreaming Inhalte, dominierend zu 95 Prozent von YouTube. Zu guter Letzt verursachen rund 18 Prozent des Online Video Streamings Soziale Netzwerke wie Facebook, Instagram, Snapchat oder Twitter.

Auf den Bereich „Andere Videos“ fallen rund 20 Prozent und beinhaltet alles rund um Live Fernsehen, Live Videos (Skype, „Camgirls“ etc.) oder auch Video Überwachung.

Nur rund 20 Prozent des gesamten weltweiten Datenverkehrs wird nicht für Videos verwendet. Dazu gehören Webseiten, E-Mails, Sofortnachrichten (WhatsApp), Onlinespeicher diverser Fotos und Daten und Firmennetzwerke.

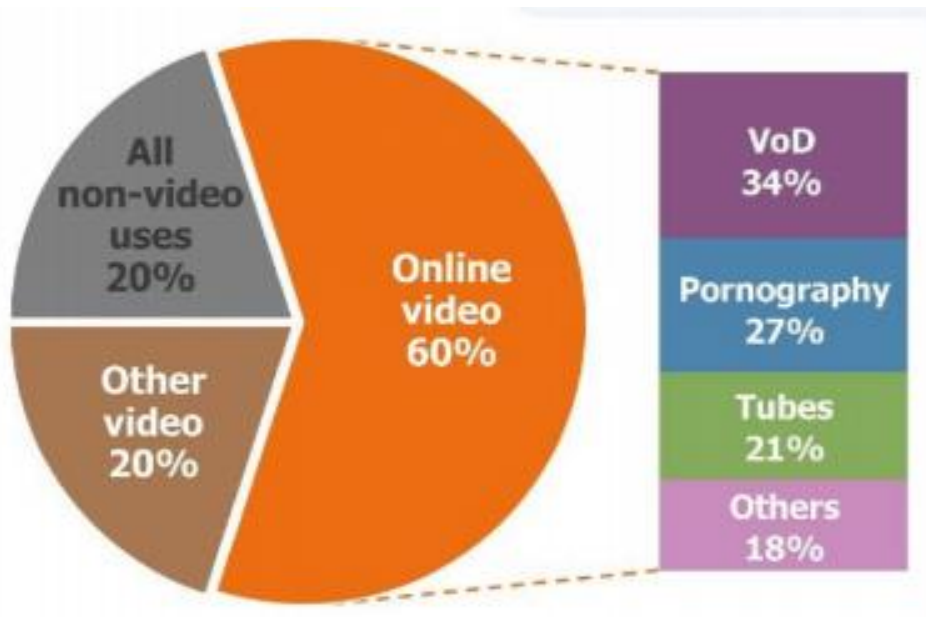


Abbildung 1: Der Globale Internet Traffic besteht zu rund 80% aus Videoverkehr. Livestreaming macht etwa 20% davon aus.

1

4 Lösungsansätze von Institutionen für Green Web/IT

Heutzutage setzen vor allem große Unternehmen wie Apple, Google, Facebook und Co. in zahlreichen Projekten auf Nachhaltigkeit und erneuerbare Energien. Dabei wird versucht der fortschreitenden Klimaerwärmung und Umweltverschmutzung entgegenzuwirken sowie den Energieverbrauch zu minimieren und Energiequellen effizienter zu nutzen.

Apple ist Spitzenreiter im Bereich Green IT und betreibt schon heute seine weltweiten Einrichtungen aus CO₂ emissionsfreien Strom. Nicht nur der Stromverbrauch der Einrichtungen steht im Mittelpunkt sondern auch der Einsatz von recycelten und nicht gesundheitsschädlichen Stoffen in ihren Produkten. Dabei wird vor allem auf Quecksilber, PVC oder Arsen vermieden.

Neben Apple ist auch Google unter den Top drei der grünsten Unternehmen. Seit 2007 nutzt Google bereits erneuerbare Energien für den Betrieb ihrer Rechenzentren. Ein interessantes Projekt von Google ist Sunroof. Es ermöglicht dem Kunden mit Hilfe von Google Maps einen persönlichen Solarplan für das eigene Dach zu erstellen.

Das drittgrünste Unternehmen ist Facebook. Das Unternehmen möchte seine Treibhausgasemission von 2017 bis 2019 um rund 75 Prozent reduzieren und ab 2020 ihre gesamten Rechenzentren mit Energie aus 100 Prozent erneuerbarer Energie betreiben.

¹ Bezug auf Theorieteil Bachelorarbeit 2 „Energieverbrauchsmessungen unterschiedlicher mobiler Endgeräte“ Kahr Marcel

Aus der unterhalb ersichtlichen Grafik sind alle fünfzehn weltweit verteilten Rechenzentren von Facebook sowie ihre erneuerbaren Energiegewinnungsquellen ersichtlich.

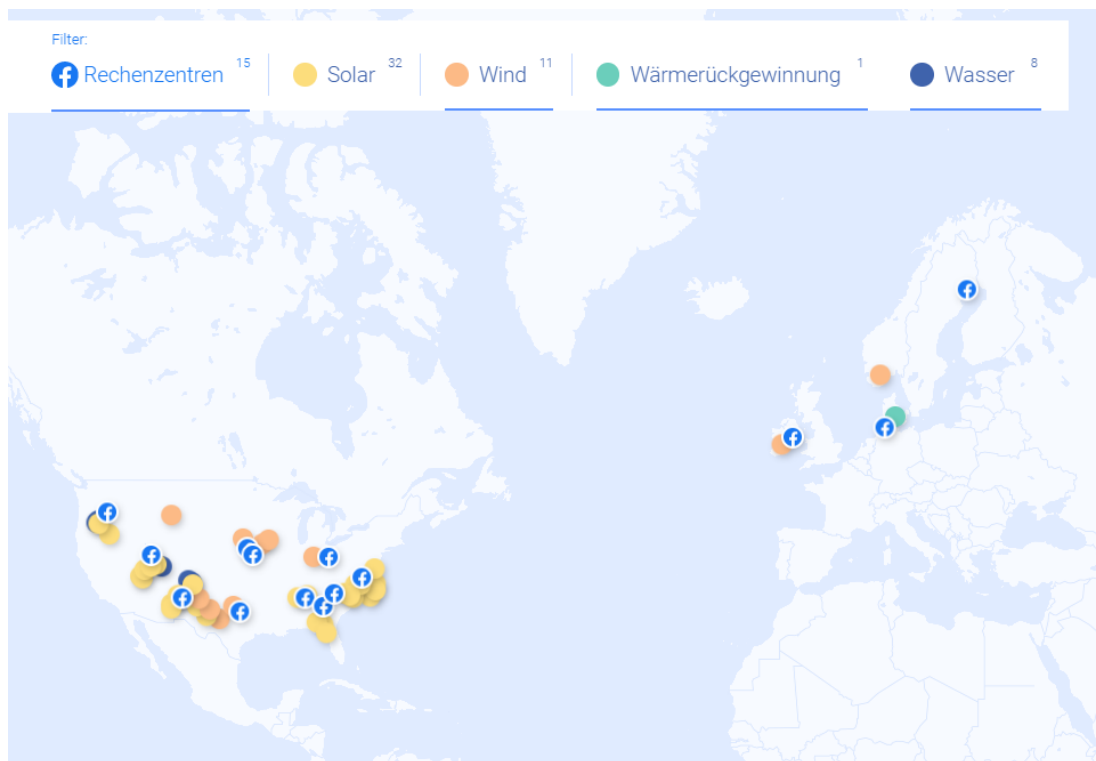


Abbildung 2: Übersichtskarte Erneuerbare Energien und Rechenzentren

Facebook nutzt beispielsweise in Dänemark für ihre Rechenzentren eine sogenannte Wärmerückgewinnung Methode. Hiermit wird die überschüssige Wärme der Server in ein lokales Fernwärmesystem weitergeleitet.

Amazon ist im Bereich Green IT noch etwas im Rückstand. Amazon hat sich bis zum Jahr 2040 das Ziel gesetzt, seine CO₂ Werte auf Null zu setzen. Hauptaugenmerk ist dabei der Transportsektor. Das Unternehmen möchte durch Alternativlösungen wie Elektrifizierung der Fahrzeuge oder die Auslieferungen in Ballungszentren mit Fahrrädern oder Prime Air Drohnen. Nicht nur im Transportsektor, sondern auch in der Cloud möchte Amazon klimaneutral arbeiten. Seit 2018 konnte die Energie für Amazon Web Service aus 50 Prozent erneuerbarer Energien gewonnen werden. Zur Minimierung des Trinkwasserverbrauchs für die Kühlung der Server in den Rechenzentren setzt Amazon auf eine sogenannte Verdunstungskühlung.²

² Bezug auf Theorieteil Bachelorarbeit 2 „Energieverbrauchsmessungen unterschiedlicher mobiler Endgeräte“ Kahr Marcel

5 Netzwerkstromverbrauch

Bei der Betrachtung einer Reihe verschiedener Studien, die versuchen den Energiebedarf des Internets abzuschätzen, unterscheiden sich die Ergebnisse GB häufig erheblich. In einer Forschungsarbeit namens „Electricity Intensity of Internet Data Transmission: Untangling the Estimates“ wurden die Ergebnisse von 14 verschiedenen Studien analysiert, die zwischen 2000 und 2015 durchgeführt wurden und den Energiebedarf des globalen Internetverkehrs als kWh/GB berechnen. Es wurde festgestellt, dass die Ergebnisse der Energieberechnungen um bis zu 5 Größenordnungen schwankten. Ein Unterschied von bis zu 2 Größenordnungen kann durch die Steigerung der Effizienz im Laufe der Zeit erklärt werden. Der Grund für diese viel größere Variation sind zum einen die unterschiedlichen Herangehensweisen an das Problem:

- Top-down: Gesamtstromverbrauch eines Netzwerks geteilt durch die Gesamtmenge der Daten, die das System durchlaufen. Es wurde festgestellt, dass dieser Ansatz den Energieverbrauch meist überschätzt.
- Bottom-up: Summe des Stromverbrauchs eines Netzwerkgeräts geteilt durch die durch das Gerät übertragenen Daten. Es wurde festgestellt, dass dieser Ansatz den Energieverbrauch meist unterschätzt.

Und auf der anderen Seite steuern die unterschiedliche Systemgrenzen, die von Studie zu Studie variieren, den großen Abweichungen bei. Um die Studien vergleichbar zu machen, konzentrierte sich der Berechnungsumfang auf das Übertragungsnetz, das aus dem IP-Kernnetz und den Zugangsnetzen besteht.

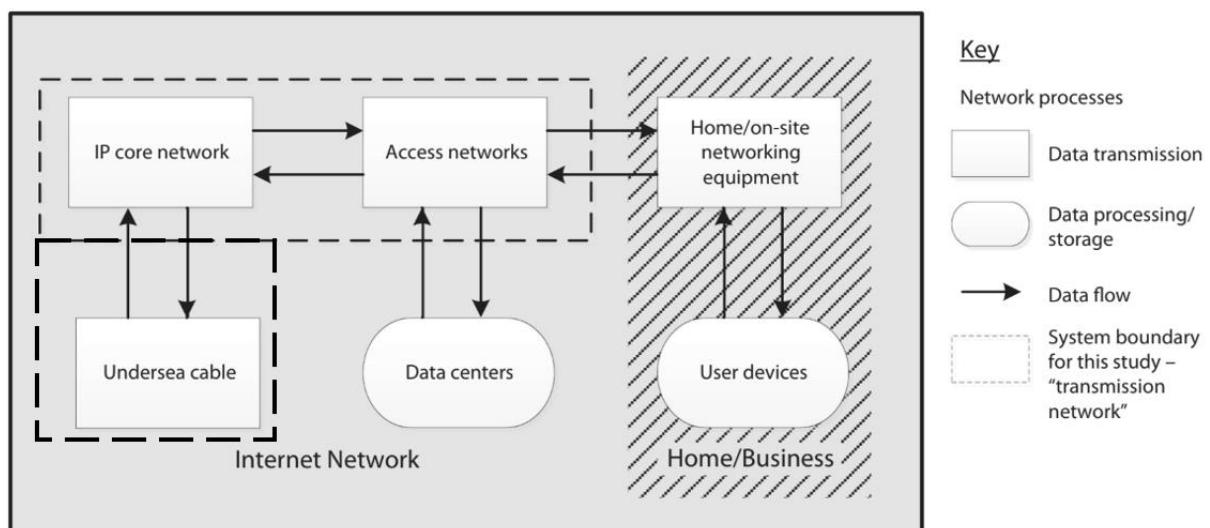


Abbildung 3: Um verschiedene Studien vergleichbar zu machen, wurde der Systemumfang auf das Transportnetzwerk beschränkt (IP core network, Access networks). Die Studie, welche für die Illustration des Netzwerkverbrauchs genutzt wird inkludiert ebenfalls den Stromverbrauch von Unterseekabeln.

Der Grund, warum Heimnetzwerke und End User Geräte nicht in den Umfang der Studie einbezogen werden ist, dass der Stromverbrauch in erster Linie vom Verbrauch abhängt. Netzwerkgeräte für Endverbraucher sind so konfiguriert, dass sie die Spitzenleistung jederzeit

bewältigen können. Die minimalen Änderungen des Stromverbrauchs, die durch unterschiedlichen Netzwerkverkehr verursacht werden, sind daher für das Endergebnis unerheblich. Es ist wichtig zu beachten, dass diese Modelle, die erstellt wurden, um die von IP-Netzwerken verwendete Energiemenge zu approximieren, in hohem Maße von der Genauigkeit der Annahmen der Eingangsvariablen abhängen. Beispielsweise sind die für eine Berechnung herangezogenen Netzwerkgeräte möglicherweise nicht repräsentativ für den Großteil der tatsächlich verwendeten Geräte.

Figure 1. Internet Energy Breakdown (kWh per GB)

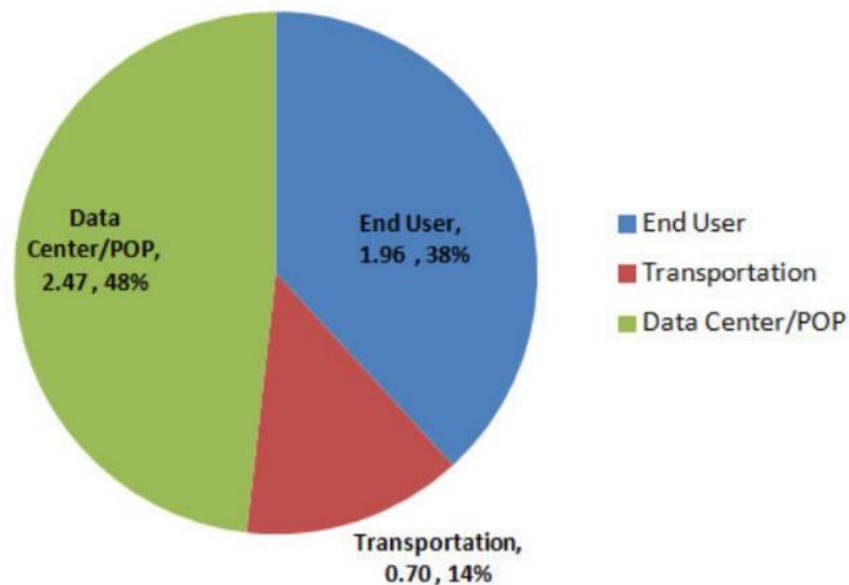


Abbildung 4: Der Stromverbrauch in kWh/GB für Datenzentren, Transportnetzwerke und End User.

Abbildung 4 stammt von einer Studie aus dem Jahr 2012, der Transportnetzwerkteil umfasst hier auch die den Verbrauch von Unterseekabeln, welcher zwischen 0,55 und 0.6 kWh/GB ausmacht. Der Grund weshalb diese Studie für die Illustration des Netzwerkverbrauchs herangezogen wird ist, dass es zwei weitere Studien aus demselben Jahr gibt die zu sehr Ähnlichen Ergebnissen kommen, was bei neueren Ergebnissen nicht der Fall ist. Der Trend im Stromverbrauch von Transportnetzwerken (ohne Unterseekabel) zeigt eindeutig, dass der Einfluss von Transportnetzwerken auf den Gesamtverbrauch immer geringer wird. Dasselbe lässt sich allerdings auch für Datenzentren und Endgeräte sagen, der relative Anteil dürfte sich daher über die Jahre nicht dramatisch verändert haben. Der meiste Stromverbrauch ist für den End User daher nicht ersichtlich. Die Messungen dieses Projekts sollen den Stromverbrauch von Datenzentren beim streamen von Videos simulieren, welcher basierend auf Abbildung 4 rund 48% des Gesamtverbrauchs ausmacht.^{3,4}

³ <https://onlinelibrary.wiley.com/doi/full/10.1111/jiec.12630>

⁴ <https://aceee.org/files/proceedings/2012/data/papers/0193-000409.pdf>

6 Infrastruktur

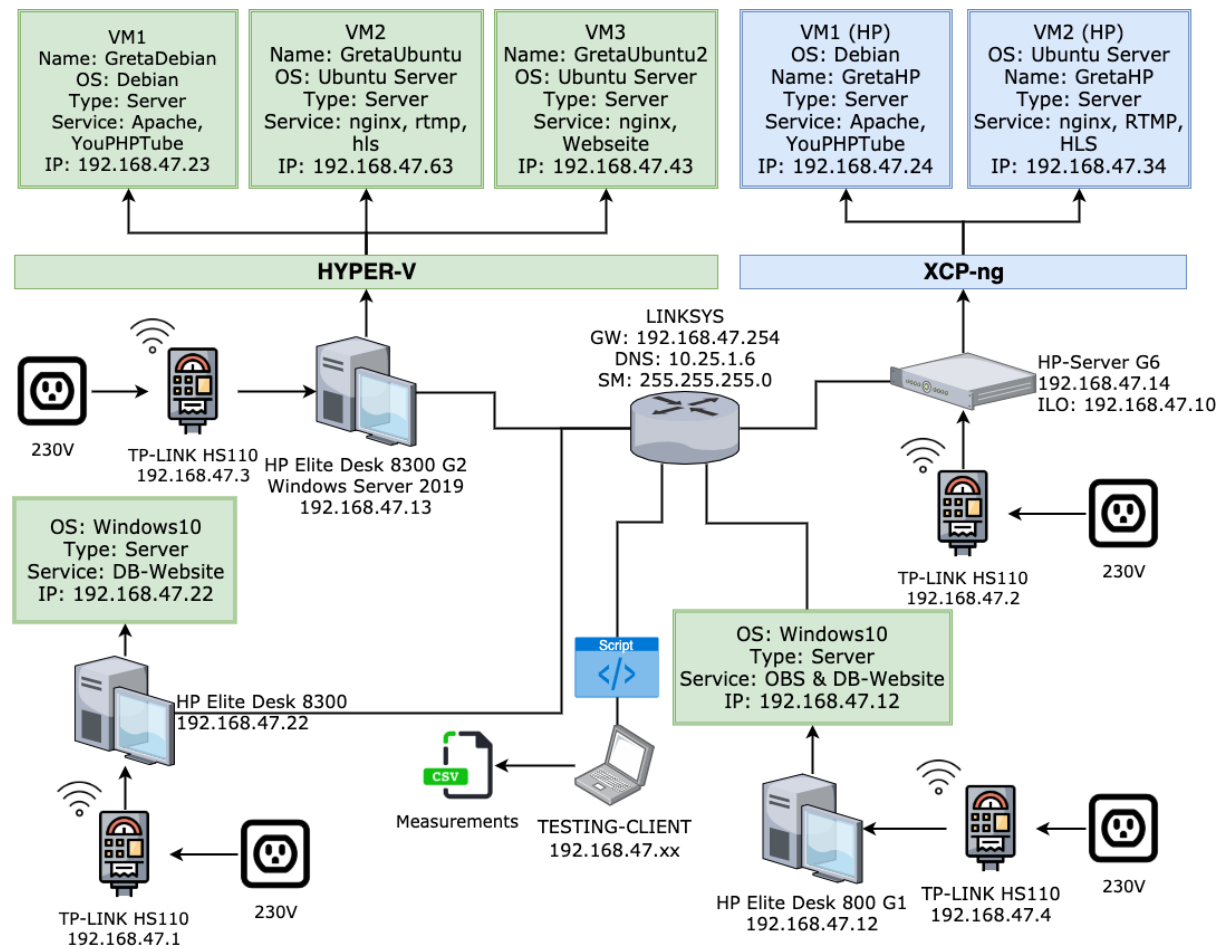


Abbildung 5: Infrastrukturumgebung

7 Hardware

Da momentan sehr viel Unruhe und Tumulte im Bereich Klimaschutz und Klimaerwärmung vorherrschen, einigten wir uns, unsere PCs und VMs nach der Klimaaktivsten Greta Thunberg zu benennen.

Da wir hauptsächlich mit den eigenen Laptops arbeiteten, nutzten wir Putty um über SSH auf die VMs zugreifen zu können. Weiters nutzten wir ein sehr mächtiges Tool namens WinSCP. Eine Windows Client, der das Secure Copy Protokoll implementiert, welches ebenfalls mit SSH Verbindungen arbeitet. Es ermöglicht es sich auf diverse Server zu verbinden und per Drag and Drop direkt Dateien von einem Server auf den andren Server zu verschieben.

7.1 HP Elite Desk 800 G1 TWR

Der Desktop-PC HP Elite Desk 800 G1 TWR ist unser Upstream Client und wird rein zum Streamen und Aufnehmen des Video Samples „Big Buck Bunny“ mit Hilfe von OBS Studio verwendet. Auf diesem PC wurde Windows 10 Enterprise installiert.

Komponente	Spezifikation
CPU	i7 4790 (3.60GHz)
Graphics	Intel HD 4600
RAM	DDR3 32GB (1600MHz)
SSD	Samsung 840 EVO (500GB)
Betriebssystem	Windows 10 Professional
BiosVer.	L01 v.02.75 (05/04/2018)

Tabelle 1: HP Elite Desk 800 G1 TWR - Komponenten und Spezifikationen

7.2 HP Elite Desk 800 G2 TWR

Der HP Elite Desk 800 G2 TWR fungiert in unserem Projekt als Server (Webserver, Streaming Server) welcher Aufgaben wie die Encodierung, Trans Codierung, Verteilung der Streams und Bereitstellung der Webseite bewältigen muss. Als Betriebssystem wurde Windows Server 2019 installiert, um darauf die Virtuellen Maschinen zu installieren.

Komponente	Spezifikation
CPU	i7 6700 (3.40 GHz)
Graphics	Intel HD 530
RAM	DDR3 32GB (2133 MHz)
SSD	Samsung 840 EVO (500GB)
Betriebssystem	Windows Server 2019
BiosVer.	N01 Ver. 02.44 (10/30/2019)

Tabelle 2: HP Elite Desk 800 G2 TWR - Komponenten und Spezifikationen

7.2.1 VM1 – Greta Debian

Auf der VM1 wurde die Linux Distribution Debian aufgesetzt. Weiters installierten wir den Apache Webserver zum Hosten der statischen und dynamischen Webseiten. Zum Starten und der MySQL Datenbank sowie des Apache Servers, wurde das XAMPP Control Panel installiert. Für etwaige Video-on-Demand Testzwecke, welche zeitlich bedingt in der BÜPA nicht umgesetzt wurden, erstellen wir einen eigenen YouTube Clone.

7.2.2 VM2 – Greta Ubuntu

Auf der zweiten virtuellen Maschine wurde die Linux Distribution Ubuntu installiert, welche unseren NGINX Webserver hostet. Dieser Webserver wurde als Streaming Server eingesetzt. Als Erweiterung wurden HLS und RTMP installiert. RTMP (Real Time Messaging Protokoll) wird zum Streamen von Audio, Video und anderen Daten verwendet. Im Projekt wurde das Protokoll eingesetzt, um das über OBS gestreamte Video auf unseren NGINX Server zu streamen. Dieser wandelte den Stream in HLS um. Beim HTTP Live Streaming Protokoll (HLS) handelt es sich um ein auf HTTP basierendes Kommunikationsprotokoll, welches für Live Streaming Zwecke eingesetzt wird.

7.2.3 VM3 – Greta Ubuntu 2

Auf der dritten VM wurde wiederum die Linux Distribution Ubuntu installiert. Mit Hilfe des Webserver NGINX konnten die statische und dynamische Webseite gehostet werden. Als Datenbank wurde wiederum MySQL eingesetzt.

7.3 HP-Server G6

Der HP-Server G6 gehört einer sehr alten Server Generation an und eignet sich deshalb nur bedingt zum Durchführen der Messungen.

Auf unserem HP-Server ProLiant DL380 G6 wurden zwei Virtuelle Maschinen installiert. Die erste virtuelle Maschine hostet das Linux OS Debian, worauf der Webserver Apache sowie der YouTube Clone installiert wurden. Die zweite virtuelle Maschine basiert auf der Linux Distribution Ubuntu worauf der Webserver NGINX, RTMP und HLS installiert wurden.

Komponente	Spezifikation
CPU	2 x Intel Xeon X5550 (2,67 GHz) (16 Cores total)
Graphics	-
RAM	24 GB
HDD	2 x HP 146GB 10K SAS
Betriebssystem	XCP-ng 8.0
BiosVer.	HP P62

Tabelle 3: HP-Server ProLiant DL380 - Komponenten und Spezifikationen

7.4 Linksys Router

Der Linksys Router wurde mit OpenWRT konfiguriert. Außerdem wurde die Vergabe von IP Adressen für externe Geräte mit Hilfe eines DHCP-Servers geregelt. Allen physischen Desktop

PCs, Server und VMs wurde eine statische IP Adresse zugeteilt. Der DHCP Server wurde benötigt, damit die TP Links konfiguriert und ins Netzwerk eingebunden werden konnten.

8 Software

8.1 Open Broadcaster Software (OBS)

Die Open Broadcaster Software (OBS) wird verwendet, um Inhalte auf dem Bildschirm des Users über das Internet in Echtzeit zu übertragen. Die Datenübertragung hierfür erfolgt über das Real Time Messaging Protokoll, welches auf der Empfängerseite von Streamingdiensten empfangen und verarbeitet. Die Zuordnung der Daten sowie die Authentifizierung erfolgt hierbei über sogenannte Streaming Schlüssel.

Als Encoder stehen in OBS x264, Intel Quick Sync Video und Nvidia NVENC zur Verfügung, welche in weiterer Folge den Stream im H.264 Format kodieren. Für die Audioübertragung kann auf die MP3 oder AAC Kodierung zurückgegriffen werden.

Die Open Source Broadcasting Software OBS stellt für uns ein geeignetes Tool dar, da es möglich ist Video Streaming Workloads in unterschiedlichen Qualitätsstufen und Kodierungen zu generieren und deren Einfluss auf den Stromverbrauch zu messen.

OBS wurde auf dem HP Elite Desk 800 G1 installiert da dieser als Streaming Client definiert ist. Der Stream wird in weiterer Folge an den HP Elite Desk 8300 G2 (VM4) geschickt auf welchem der Streaming Server NGINX (incl. RTMP Extension) installiert wurde.

8.2 NGINX

NGINX wird als Open Source Projekt betrieben und kann für eine Vielzahl von Anwendungen verwendet werden. Diese Flexibilität war einer der Gründe warum wir uns für diesen entschieden haben. Ressourcenschonen, hohe Performance, und steigende Verwendung im Business Bereich waren weitere Gründe, um NGINX anstelle von Apache zu verwenden. Zudem verfügt NGINX über eine große Community was es uns erleichtert den RTMP Streaming Server zu konfigurieren.

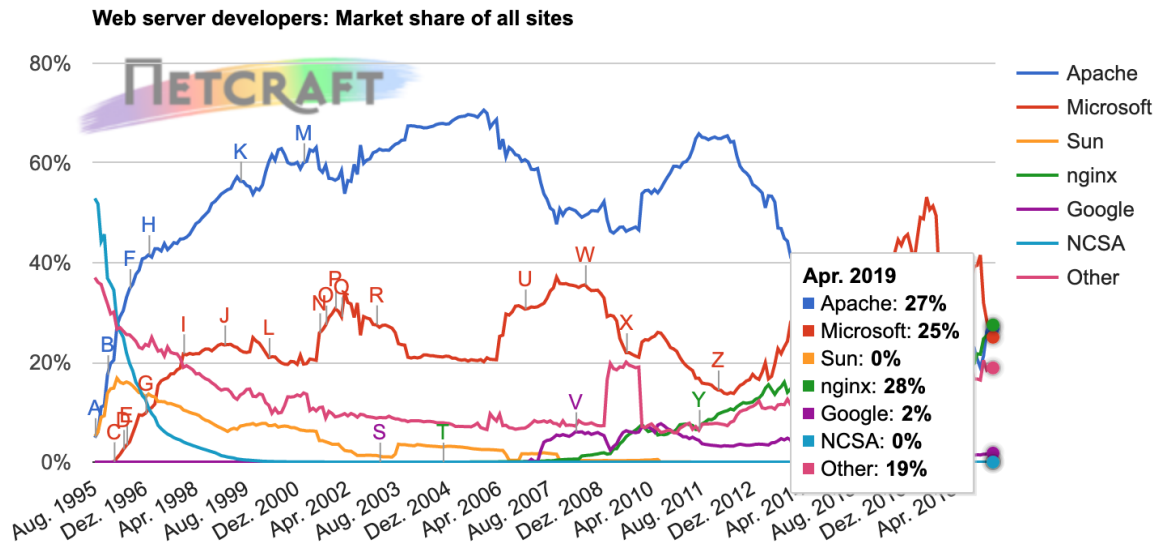


Abbildung 6: Market share of all sites ⁵

8.2.1 NGINX für RTMP Streaming

In unserem Project wurde NGINX mit dem RTMP Modul auf der VM4 des HP Elite Desk 8300 G2 installiert und fungiert hier als Streaming Server, welcher den Stream des Clients entgegennimmt und an die Zuseher (Clients) verteilt.

Auf dem NGINX RTMP Server werden zwei „use-cases“ behandelt.

- Zum einen wird der Stream welcher an den Server geschickt wird direkt an die Zuseher verteilt. Diese Verteilung erfolgt ohne jegliche Veränderung des Videos.
- Zum anderen wird der Stream mithilfe von ffmpeg eine niedrigere Qualitätsstufe gebracht. Diese Konfiguration erfolgt direkt in der NGINX Konfigurationsdatei.

8.2.2 NGINX Installation inkl. RTMP Modul

Centos:

Für die Installation von NGINX mit dem zugehörigen RTMP Modul wurde das Tutorial von Samuyi mit dem Titel: „How to Setup NGINX for hls video streaming on Centos 7“ verwendet. Da das RTMP Modul nicht standardmäßig in NGINX verfügbar ist muss das ganze Programm einschließlich gewünschter Module neu kompiliert werden.⁶

Hier werden nochmal die grundlegenden Installationsschritte zusammengefasst:

Folgende Module müssen auf CentOS vorhanden bzw. installiert sein, um einen reibungslosen Installationsprozess zu gewährleisten.

- git, wget, gcc, gcc-c++, perl, gd, gd-devel, perl-ExtUtils-Embed, geoip, geoip-devel, tar

⁵ <https://news.netcraft.com/archives/2019/04/22/april-2019-web-server-survey.html>

⁶ <https://dev.to/samuyi/how-to-setup-NGINX-for-hls-video-streaming-on-centos-7-3jb8>

CENTOS aktualisieren & Pakete Installation

```
> sudo yum update
> sudo yum install epel-release
> sudo yum install git wget gcc gcc-c++ tar gd gd-devel perl-ExtUtils-Embed geioip
geioip-devel
```

Pearl Regular Expression (PCRE) Installation

```
> wget ftp.pcre.org/pub/pcre/pcre-8.42.tar.gz
> tar -zxf pcre-8.42.tar.gz
> rm -rf pcre-8.42.tar.gz
> cd pcre-8.42
> ./configure
> make
> sudo make install
```

ZLIB (v1.2.11) compression Library Installation

```
> wget http://zlib.net/zlib-1.2.11.tar.gz
> tar -zxf zlib-1.2.11.tar.gz
> rm -rf zlib-1.2.11.tar.gz
> cd zlib-1.2.11
> ./configure
> make
> sudo make install
```

OPEN SSL (v.1.0.2q) Installation

```
> wget http://www.openssl.org/source/openssl-1.0.2q.tar.gz
> tar -zxf openssl-1.0.2q.tar.gz
> rm -rf openssl-1.0.2q.tar.gz
> cd openssl-1.0.2q
> ./config
> make
> sudo make install
```

NGINX-RTMP-MODULE Open Source Projekt von <https://github.com/arut/NGINX-rtmp-module>

```
> git clone https://github.com/arut/NGINX-rtmp-module
```

NGINX (v1.14.2) installation

```
> wget https://NGINX.org/download/NGINX-1.14.2.tar.gz
> tar zxf NGINX-1.14.2.tar.gz
> rm -rf NGINX-1.14.2.tar.gz
> cd NGINX-1.14.2
```

NGINX Konfigurationsoptionen mit den zugehörigen Modulen

```
> ./configure --add-module=../NGINX-rtmp-module \  
--sbin-path=/usr/sbin/NGINX \  
--lock-path=/var/run/NGINX.lock \  
--conf-path=/etc/NGINX/NGINX.conf \  
--pid-path=/run/NGINX.pid \  
--with-pcre=../pcre-8.42 \  
--with-zlib=../zlib-1.2.11 \  
--with-openssl=../openssl-1.0.2q \  
--error-log-path=/var/log/NGINX/error.log \  
--http-log-path=/var/log/NGINX/access.log \  
--user=NGINX \  
--group=NGINX \  
--with-http_auth_request_module \  
--with-http_degradation_module \  
--with-http_geoip_module \  
--with-http_gunzip_module \  
--with-http_gzip_static_module \  
--with-http_image_filter_module \  
--with-http_mp4_module \  
--with-http_perl_module \  
--with-http_realip_module \  
--with-http_secure_link_module \  
--with-http_slice_module \  
--with-http_ssl_module \  
--with-http_stub_status_module \  
--with-http_v2_module \  
--with-stream_ssl_module \  
--with-stream \  
--with-threads \  
--prefix=/etc/NGINX
```

NGINX kompilieren

```
> make  
> sudo make install
```

NGINX USER ERSTELLEN / LOG DIRECTORY

```
> sudo useradd --system --home /var/lib/NGINX --shell /sbin/nologin --comment  
"NGINX system user" NGINX  
> sudo mkdir /var/log/NGINX && sudo chown NGINX:NGINX /var/log/NGINX
```

NGINX service file

```
> sudo vim /lib/systemd/system/NGINX.service  
  
[Unit]  
Description=NGINX - high performance web server  
Documentation=https://NGINX.org/en/docs/
```

```

After=network.target remote-fs.target nss-lookup.target
Wants=network-online.target
[Service]
Type=forking
PIDFile=/run/NGINX.pid
ExecStartPre=/usr/bin/rm -f /run/NGINX.pid
ExecStartPre=/usr/sbin/NGINX -t -c /etc/NGINX/NGINX.conf
ExecStart=/usr/sbin/NGINX -c /etc/NGINX/NGINX.conf
ExecReload=/bin/kill -s HUP $MAINPID
KillSignal=SIGQUIT
TimeoutStopSec=5
KillMode=process
PrivateTmp=true
[Install]
WantedBy=multi-user.target

```

RELOAD AND RESTART NGINX

```

> sudo systemctl daemon-reload
> sudo systemctl start NGINX
> sudo systemctl enable NGINX

```

Video Stream Files directory

```

> sudo mkdir -p /var/www/hls/live

```

NGINX CONFIG FILE /etc/NGINX/NGINX.conf

...

Wichtig! CENTOS wurde aufgrund seiner strikten Sicherheitsrichtlinien nicht verwendet und anstelle dessen wurde ausschließlich Ubuntu Server verwendet.

Ubuntu Server:

Da es nicht möglich war die über Snap installierte Version von ffmpeg aus der NGINX Konfiguration auszuführen wurde das NGINX Setup auf eine Ubuntu Server VM verschoben. Ein großer Vorteil bei der Installation auf Ubuntu ist, dass das RTMP Modul in Aptitude, dem standardmäßigen Paketmanager auf Debian basierenden Linux Distributionen, verfügbar ist. NGINX unterstützt seit Version 1.9.11 dynamische Module. Das ermöglicht es Features in Form von Modulen in NGINX zu aktivieren, ohne es neu kompilieren zu müssen. Diese Module müssen entweder selbst kompiliert werden oder falls sie in Aptitude verfügbar sind einfach installiert werden. Das RTMP Module ist unter dem Namen libNGINX-mod-rtmp im Repository zu finden. Ist das Paket einmal installiert muss nur mehr in der NGINX Konfiguration darauf verwiesen werden:⁷

```
load_module "modules/ libNGINX-mod-rtmp";
```

⁷ <https://docs.NGINX.com/NGINX/admin-guide/dynamic-modules/dynamic-modules/>

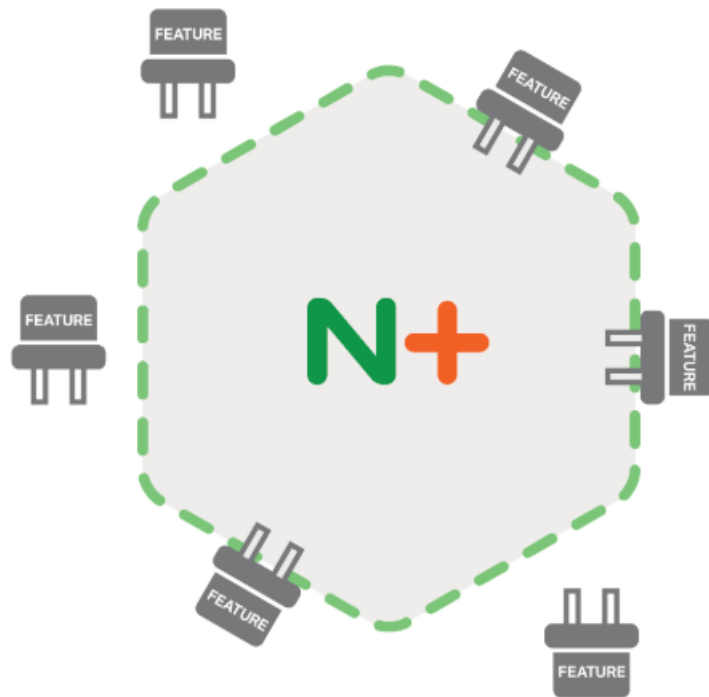


Abbildung 7: Seit Version 1.9.11 unterstützt NGINX einen modularen Aufbau.

8.3 ffmpeg

Bei ffmpeg oder auch „Fast Forward MPEG“ genannt, handelt es sich um ein vielfach genutztes Multimedia-Framework und Befehlszeilenprogramm mit zahlreichen Funktionen. Es bietet Möglichkeiten beinahe alles zu dekodieren, kodieren, transkodieren, streamen oder zu filtern. Es unterscheidet sich von anderen GUI Programmen dahingehend, da es die WYSIWYG-Methode (was Sie sehen, ist was Sie bekommen) verwenden.^{8,9}

ffmpeg liest eine beliebige Anzahl von Dateien ein und schreibt eine beliebige Anzahl von Ausgangsdateien, die angegeben werden in einen einfachen Ausgabe-URL.¹⁰

Zum Encoden wird folgender ffmpeg Befehl als Vorlage verwendet:

```
ffmpeg -i rtmp://localhost:1935/live/ -r 60 -vf scale=1920:1080 -c:v  
libx264 -preset medium -b:v 6M -minrate 6M -maxrate 6M -bufsize 12M  
-b:a 320k -f flv rtmp://localhost:1935/mobile/;
```

⁸ <https://snapcraft.io/install/ffmpeg/centos>

⁹ <https://riptutorial.com/de/ffmpeg/example/24782/was-ist-ffmpeg->

¹⁰ <https://ffmpeg.org/ffmpeg.html#Description>

Preset / Voreinstellung

Das Attribut Preset umfasst eine Sammlung von Optionen, welches ein bestimmtes Verhältnis von Kodierungsgeschwindigkeit zur Komprimierung darstellt. Im Allgemeinen heißt das, mit einer langsameren Voreinstellung erzielt man eine bessere Komprimierung der Qualität. Aus der unten angegebenen Grafik können die unterschiedlichen Kodierungsgeschwindigkeiten entnommen werden. Falls kein Preset definiert wird, verwendet ffmpeg standardmäßig die Kodierungsgeschwindigkeit „medium“.

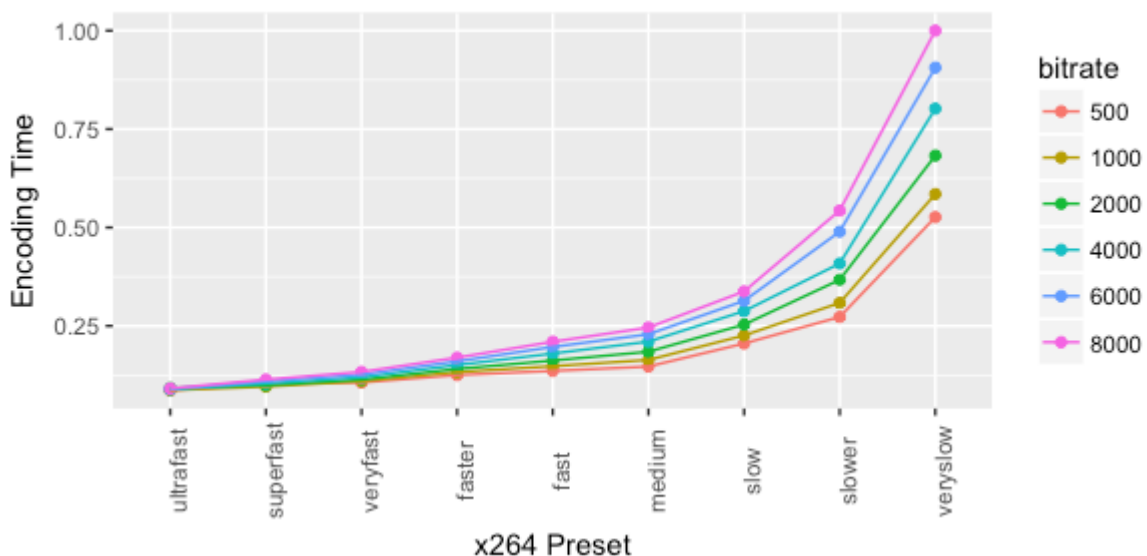


Abbildung 8: Kodierungszeitbeispiel eines 1080p-Videos

Die Kodierungszeit hängt von der Qualität des Quellmaterials, von der Ziel-Bitrate sowie von der Hardwarekonfiguration ab. Grundsätzlich gilt, je höher die Bitrate, desto mehr Zeit wird beim Kodieren benötigt.

Von mittel bis langsam erhöht sich die benötigte Zeit um ca. 40 Prozent. Wenn man stattdessen langsamer kodiert, wird etwa 100 Prozent mehr Zeit benötigt (das heißt, es dauert doppelt so lange). Im Vergleich zu mittel, sehr langsam erfordert 280 Prozent der ursprünglichen Codierungszeit, mit nur minimalen Verbesserungen gegenüber langsamer in Bezug auf Qualität.

Mit schnell spart man etwa 10 Prozent der Kodierungszeit und bei schneller rund 25 Prozent. Bei ultraschnell spart man rund 55 Prozent, jedoch mit einer sehr viel geringeren Qualität.¹¹

Parameter	Description	Example Value
-b:v	Video Bitrate	150k
-b:a	Audio Bitrate	192k
-ac	Set Number of Audio Channels	2

¹¹ <https://trac.ffmpeg.org/wiki/Encode/H.264#Listpresetsandtunes>

Parameter	Description	Example Value
-crf	Constant Rate Factor	23
-qscale	Constant Quality Scale. {1,30}	20
-prefwidth	Preferred Width	640
-prefheight	Preferred Height	480
-vcodec	Video Codec	libx264
-acodec	Audio Codec	libfaac
-preset	Provides a certain encoding speed to compression ratio	medium
-threads	Set the Thread Count	2
-b	Set Bitrate (Output, Audio, Video)	200k
-vpre	Loads vpre preset file (Legacy)	
-fpre	Loads fpre preset file (Legacy)	
-fps	Desired Output Framerate	30
-ab	Audio Bitrate	192k
-ar	Set Audio Sampling Rate (in Hz)	44100
-setpts	Change the PTS (presentation timestamp) of the input video frames	10
-scaledownonly	Prevents scaling width and height greater than original video size	True
-i	Input	rtmp://localhost:1935/live/
-r	Seconds	60
-vf	Resolution	scale=1920:1080
-c:v	Kodierung	libx264
-b:v	Bitrate	6M
-preset	Kodiergeschwindigkeit:Komprimierung	medium
-minrate	Minimum bitrate	6M
-maxrate	Maximum bitrate	6M
-bufsize	Set this twice the maxrate	12M
-f	Output Format	flv rtmp://localhost:1935/mobile/

Tabelle 4: FFmpeg Parameter¹²

8.4 Snapd

Bei Snapp Apps (snaps) handelt es sich um ein sogenanntes Paketformat, welches zusätzlich zur normalen Paketverwaltung installiert und verwendet werden kann. Dieses Paketformat war ursprünglich für das Internet of Things und für den Einsatz auf Servern und Clouddiensten entwickelt worden. Ein Vorteil von Snapd ist, dass sie bereits alle Dateien und Abhängigkeiten wie z.B. diverse Konfigurationsdateien und Bibliotheken mit sich bringen. Die Laufzeitumgebung der snaps bezeichnet man als Core. Für die Verwaltung der snaps wird der

¹² <https://entermediadb.org/knowledge/9/avconvffmpeg-parameters/>

Hintergrunddienst snapd verwendet. Es können unterschiedliche snaps im Snap Store heruntergeladen werden.

Snapd wurde benötigt, da ffmpeg für CentOS nur bis zur Version 2 supportet wird. Doch mit Hilfe von Snapd konnte wir die aktuelle 4. Version von ffmpeg installieren.

Um Snaps verwenden zu können, müssen diese zuerst aktiviert werden. Da Snap im EPEL-Repository (Extra Packages for Enterprise Linux) verfügbar ist, kann dieses mit dem nachfolgenden Befehl hinzugefügt werden.

\$ sudo yum install epel-release

Nun kann Snap mit dem Befehl **\$ sudo yum install snapd** installiert werden.

Nach der Installation muss die Systemeinheit aktiviert werden.

\$ sudo systemctl enable --now snapd.socket

Im nächsten Schritt muss ein Support mit Hilfe einer symbolischen Verknüpfung zwischen /var/lib/snapd/snap und /snap erstellt werden.

\$ sudo ln -s /var/lib/snapd/snap /snap

Nachdem snap richtig installiert wurde muss das System neu gestartet werden, damit die Pfade von snap aktualisiert werden. Nach dem Neustart kann nun mit Hilfe von snap FFmpeg installiert werden.

\$ sudo snap install ffmpeg ^{13,14}

9 Mess-Tools

Für unsere Projektarbeit wurden zwei wichtige Messwerkzeuge verwendet. Zum einen der TP-Link HS110 Smart Wall Plug welcher den Strom direkt von der Steckdose misst. Zum anderen das Intel Power Gadget welches den Stromverbrauch der CPU direkt ausliest.



Abbildung 9: Messtools

¹³ <https://wiki.ubuntuusers.de/snap/>

¹⁴ <https://snapcraft.io/install/ffmpeg/centos>

9.1 Intel Power Gadget

Beim Intel Power Gadget handelt es sich um ein softwarebasiertes Tool von Intel. Es dient zur Überwachung des Stromverbrauchs der Intel Prozessoren und unterstützt die Generationen zwei bis zehn. Dabei werden in Echtzeit mithilfe des Energiezählers im Prozessor der Stromverbrauch in Watt ermittelt. Das Power Gadget unterstützt nur Core und keine Atom Prozessoren von Intel. Betriebssysteme wie Windows, MacOS sowie auch Linux unterstützen dieses Tool ebenfalls. Diverse Plattformen wie Notebooks, Desktops und Server unterstützten das Power Gadget jedoch nur am Parent-OS und nicht in virtuellen Umgebungen.

Mit Hilfe des Start Log Buttons kann der Energieverbrauch für einen zuvor definierten Zeitraum aufgenommen werden. Dabei wird eine CSV-Datei mit den einzeln gemessenen Komponenten erstellt.

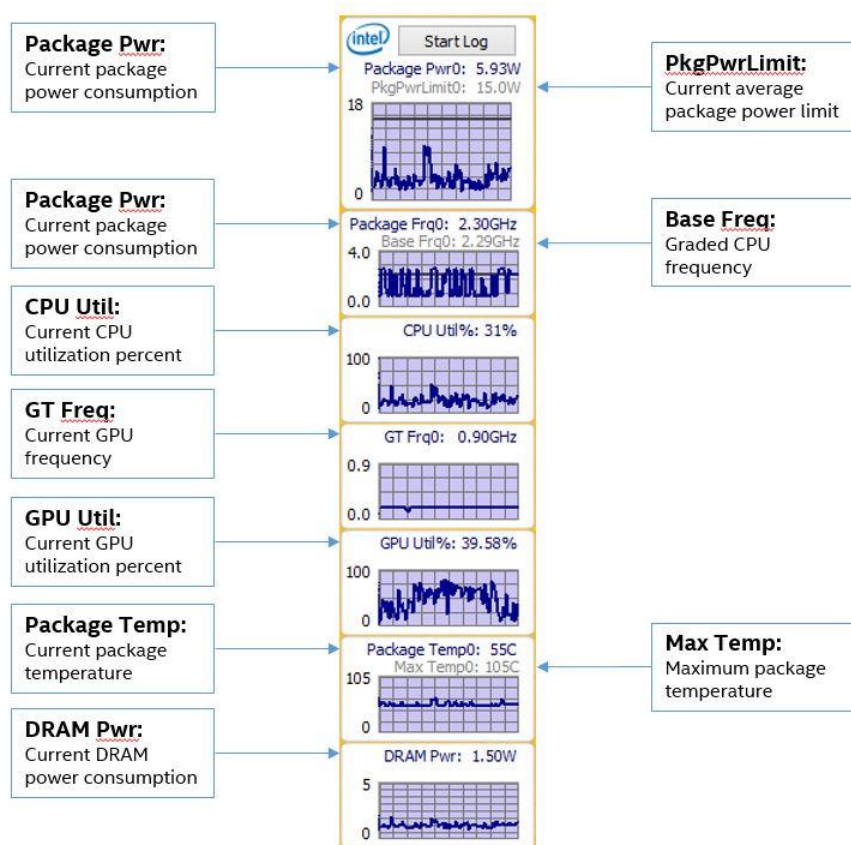


Abbildung 10: Überblick Intel Powergadget Tool ¹⁵

9.2 HP ILO

Beim HP ILO (Integrated Lights-Out) handelt es sich um ein proprietäres Management-System zur Administration von HP Servern. In diesem Fall handelt es sich um das ILO2. Die Firmware zur Konfiguration des ILOs wird beim Booten des Servers beim Drücken der F8 taste

¹⁵ <https://software.intel.com/en-us/articles/intel-power-gadget>

aufgerufen. Hier lässt sich das ILO aktivieren, User anlegen und die Schnittstelle genauer konfigurieren. Das ILO wird entweder über die eigens dafür verbaute Ethernet Port mit dedizierter NIC angesprochen oder als "shared Ethernet Port" über das Ethernet Port des Motherboards.

Ist das ILO einmal aktiviert stellt es ein Webinterface zur Verfügung. Das Interface des ILO2 wird von vielen derzeitigen Browsern nicht mehr unterstützt, wir konnten es nur mit Internet Explorer und Firefox (dank exzellenter Rückwärtskompatibilität) aufrufen. Das Webinterface bietet einiges an Funktionalität mit sich, die für unser Projekt interessanten Powermonitoring Tools befinden sich allerdings hinter einer weiteren Paywall in der Form einer Advanced License Key. Glücklicherweise können diese wiederverwendet werden, Zugriff zu den Erweiterten Features kann man daher durch eine einfache Google Suche erlangen. Im Webinterface lässt sich nun ein 24 Stunden Rückblick über den Stromverbrauch ausgeben.

Zusätzlich gibt es eine Python API, über welche der Stromverbrauch ebenfalls ausgelesen werden kann. Für den Zugriff über die API-Schnittstelle wurde ein einfaches Python-Skript geschrieben, welches den aktuellen Stromverbrauch im Sekundentakt ausgibt. Da sich der Wert im ILO nicht oft ändert und es in der Dokumentation des ILO2 nicht erläutert wird wurde das ILO nicht für Messungen herangezogen. Es ermöglichte allerdings die Genauigkeit der smarten Steckdosen zu überprüfen.

Wichtig! Die HP-ILO wurde aufgrund seiner sehr langsamen Abtastrate in diesem Projekt nicht für Messungen verwendet!

9.3 TP LINK HS110

Eine hardwareseitige Messmethode, die für unser Projekt ausgewählt wurde, ist der TP-Link HS110. Die TP-Links wurden mit Hilfe eines Smartphones und der APP Kasa Smart konfiguriert. Damit können die TP-Links auch ein- und ausgeschaltet werden und bekommen einen Überblick des Stromverbrauchs. Dieser eignet sich besonders gut für unsere Messungen da er vom Unternehmen softScheck, welches sich auf Software Schwachstellen bzw. Sicherheitslücken spezialisiert hat, analysiert wurde und durch eine Schwachstelle ein vollständiger Zugriff auf das Gerät möglich ist.

Die genaue Analyse der Sicherheitslücke wurde von softScheck genauestens auf ihrer Webseite dokumentiert.¹⁶

SoftScheck hat zudem das Python 2 Script öffentlich zugänglich gemacht, um die Daten sehr einfach und komfortabel auszulesen. Dieses Script befindet sich auf einem GitHub Repository.¹⁷

Durch diese Schwachstelle können Befehle im JSON Format an das Gerät gesendet und empfangen werden.

¹⁶ <https://www.softscheck.com/en/reverse-engineering-tp-link-hs110/>

¹⁷ <https://github.com/softScheck/tplink-smartplug>

9.3.1 TP-LINK HS110 Parameter

Parameter	Beschreibung
on	TP Link einschalten
off	TP Link ausschalten
info	Geräteinformationen
time	Systemzeit
reboot	Gerät neustarten
reset	TP Link zurücksetzen
energy	Live: Spannung – Strom – Leistung

Tabelle 5: TP-Link HS110 Parameter

JSON Antwort des TP-Link HS110 auf “energy”abfrage:

```
{'emeter':  
  {'get_realtime':  
    {'voltage_mv': 231705, 'current_ma': 55, 'power_mw': 1153, 'total_wh': 0, 'err_code': 0}}}
```

Als Grundlage für unser Script wurde das Projekt von softScheck verwendet welches in weiterer Folge auf die neueste Python v3 aktualisiert wurde. Zusätzlich wurden zusätzliche Features und Komponenten implementiert und umgesetzt welche eine live Aufzeichnung der Messung in eine CSV bzw. XLSX Datei ermöglichen. Das vollständige Script welches in diesem Projekt eingesetzt wurde, befindet sich auf GitHub. ¹⁸

¹⁸ <https://github.com/colingit93/TP-Link-HS110>

10 Software Tools

10.1 Apache

Apache wird verwendet um den Stromverbrauch hinsichtlich statischer bzw. dynamischer Webseiten zu ermitteln. Da das Projektteam auf die dynamische Webseite Apache getestet hat wurde eine Virtuelle Maschine mit Windows aufgesetzt.

Bei Apachen handelt es sich um die weltweit meistgenutzte Anwendung zum Hosten von Webinhalten. Die Applikation wird von Apache Software Foundation als freie Software zu Verfügung gestellt. Wir nutzen den Server, um die statische und dynamische Webseite des Partnerprojektteams zu hosten. Da Apache intern anscheinend Komponenten installiert hat, die zahlreiche Userzugriffe (DDoS-Attacken) blockieren, setzten wir als Alternative einen NGINX Server auf. Am NGINX Server konnten die Tests viel flexibler durchgeführt werden.

10.2 MySQL

Bei MySQL handelt es sich um eine weltweit verbreitetes relationales Datenbankverwaltungssystem. In Kombination mit Apache Webserver und der Skriptsprache PHP, wird MySQL zur Datenspeicherung für Webservices eingesetzt.

10.3 JMeter

Apache JMeter ist eine Java Open-Source-Software. Mit Hilfe von JMeter können Leistungstests von Webanwendungen sowie Analysen von zahlreichen Diensten durchgeführt werden. Weiters ist es möglich einen Lastentest von mehreren gleichzeitigen Benutzerzugriffen und diverse andere Einstellungen durchzuführen, um beispielsweise 1000 http Requests auf einen Streaming-Server oder eine Website zu testen. Weitere mögliche Requests sind FTP Requests, JUnit Requests, Java oder HTTP2 Requests.

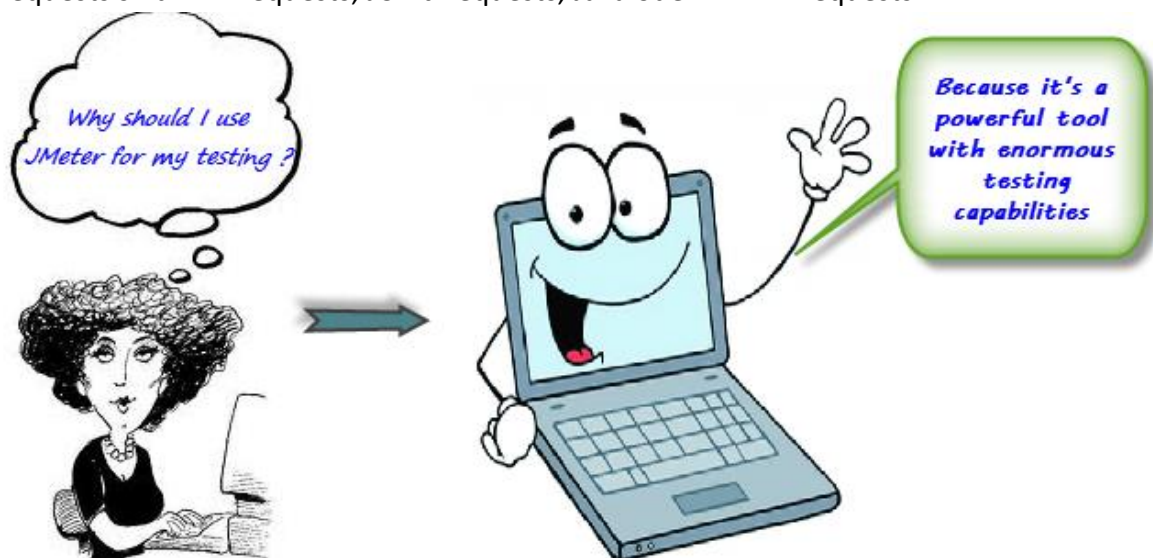


Abbildung 11: JMeter Einsatz

Zum Testen von Webapplikationen wird zunächst ein Testplan erstellt. Dieser besteht immer aus einem Threadgroup und diversen Samplern. Im Threadgroup lassen sich die Anzahl an Threads, die Ramp-up Period und die Loop Count definieren. Die Anzahl an Threads ist die Anzahl an Usern die später auf die definierte Webseite zugreifen wird. Die Ramp-up Period bestimmt die Dauer bis alle User aktiv sind. Ein Threadgroup von 30 Usern mit einer Ramp-up Period von 120 würde also alle vier Sekunden einen neuen Thread starten. Dieses Setting ist vor allem dann nützlich, wenn sehr viele Threads auf die Webseite zugreifen, die den Webserver ohne Ramp-up überlasten würden. Die Art des Zugriffes auf den Webserver wird durch den Sampler definiert. Ein einfacher GET reuest wird über einen HTTP Request Sampler gesendet. Da es sich in unserem Fall um HLS handelt, benötigen wir einen HLS Sampler. Dafür muss der Jmeter Plugins Manager installiert werden, ein JAR File welches in den lib/ext Folder kopiert wird. Im Plugins Manager lässt sich dann der von Blazemeter entwickelte HLS Sampler installieren. Diesen hängen wir an unseren Testplan und geben ihm den Pfad zum HLS Playlist File (.m3u8).

Abbildung 12: Beim HLS Sampler geben wir den Pfad zum Playlist File. Die Video dauer wurde hier auf 60 Sekunden gesetzt.

Zuletzt finalisieren wir den Testplan mit Listenern, mittels welchen wir die Resultate des Tests sehen können.

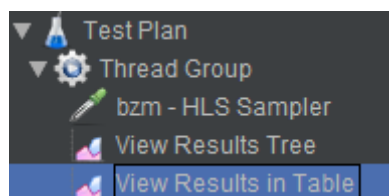


Abbildung 13: Der Fertige testplan für die Simulation von user Zugriffen auf den HLS Stream.

Beim Arbeiten mit HLS Samplern ist der die große Menge an RAM zu beachten. Die Standardeinstellung für Jmeter sind nur 512 MB RAM. Um mehr RAM für die Java Applikation zu reservieren muss im Launchscript (jmeter.bat oder jmeter.sh) die Java Heap Size vergrößert werden. Bei einer Threadgroup Größe von 500 Usern werden zirka 20GB benötigt:

```
Set HEAP= -Xms1g -Xmx20g;
```

Im Listener view Result in Table werden beim Ausführen den Testplans die empfangenen Responses aufgelistet. Bei HLS video sind die zum einen GET requests für das Playlist File (media playlist) und GET requests für die im Playlistfile verwiesenen HLS Chunks (media segment).^{19, 20, 21}

42	17:30:32.229	Thread Group 1-41	bzm - HLS Sampler - media playlist	22	✓	449
43	17:30:32.045	Thread Group 1-42	bzm - HLS Sampler - media playlist	22	✓	449
44	17:30:32.064	Thread Group 1-43	bzm - HLS Sampler - media playlist	23	✓	449
45	17:30:32.085	Thread Group 1-44	bzm - HLS Sampler - media playlist	25	✓	449
46	17:30:32.105	Thread Group 1-45	bzm - HLS Sampler - media playlist	22	✓	449
47	17:30:32.126	Thread Group 1-46	bzm - HLS Sampler - media playlist	23	✓	449
48	17:30:32.144	Thread Group 1-47	bzm - HLS Sampler - media playlist	24	✓	449
49	17:30:32.164	Thread Group 1-48	bzm - HLS Sampler - media playlist	35	✓	449
50	17:30:32.200	Thread Group 1-49	bzm - HLS Sampler - media playlist	28	✓	449
51	17:30:32.204	Thread Group 1-50	bzm - HLS Sampler - media playlist	24	✓	449
52	17:30:31.282	Thread Group 1-3	bzm - HLS Sampler - media segment	1225	✓	9414582
53	17:30:31.793	Thread Group 1-2	bzm - HLS Sampler - media segment	1065	✓	8311210
54	17:30:31.373	Thread Group 1-8	bzm - HLS Sampler - media segment	1859	✓	9414582
55	17:30:31.458	Thread Group 1-12	bzm - HLS Sampler - media segment	1855	✓	9414582
56	17:30:31.438	Thread Group 1-11	bzm - HLS Sampler - media segment	1929	✓	9414582
57	17:30:31.294	Thread Group 1-4	bzm - HLS Sampler - media segment	2130	✓	9414582
58	17:30:31.413	Thread Group 1-10	bzm - HLS Sampler - media segment	2103	✓	9414582
59	17:30:31.395	Thread Group 1-9	bzm - HLS Sampler - media segment	2261	✓	9414582

Abbildung 15: Im Listener "view result in table" werden die HTTP responses des Servers aufgelistet

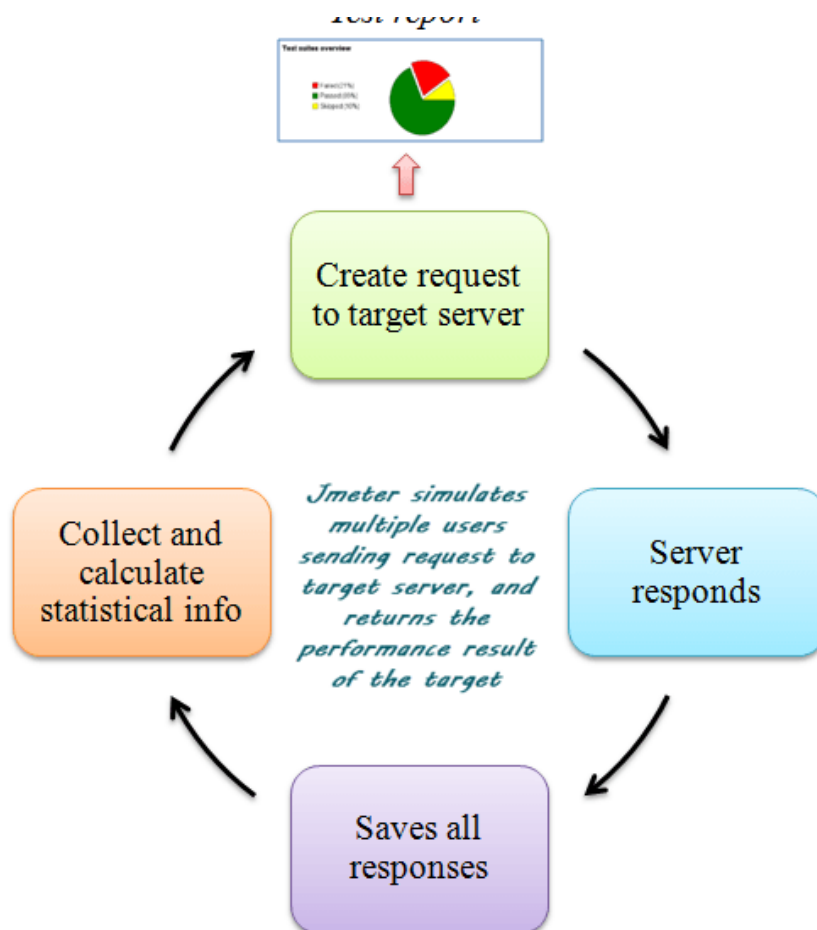


Abbildung 14: JMeter Arbeitsfluss

¹⁹ https://jmeter.apache.org/usermanual/test_plan.html

²⁰ <https://www.blazemeter.com/blog/the-new-hls-plugin-for-jmeter-the-complete-guide/>

²¹ <https://www.guru99.com/introduction-to-jmeter.html>

10.4 XAMPP

Bei XAMPP handelt es sich nicht nur um ein Softwaretool, sondern um eine Zusammenstellung mehrerer freier Softwares. Der Name des Tools setzt sich aus den Anfangsbuchstaben der Kernkomponenten zusammen. Die Kernkomponenten umfassen den Webserver Apache, die relationale Datenbanken MySQL und Maria DB und die Skriptsprachen PHP und Perl. Mit Hilfe dieses Tool können die diversen Komponenten leichter konfiguriert, gestartet oder gestoppt werden. Außerdem bietet es die Möglichkeit für die einzelnen Komponenten Log-Files zu erstellen. Das X steht als Platzhalter für diverse Betriebssysteme wie Linux, Windows und Mac OS X. Wir nutzen das Tool um mit Hilfe von Apache und MySQL die dynamische und statische Webseiten des Partnerprojektteams zu hosten. Mit Hilfe von JMeter wurden dann Benutzerzugriffe simuliert und der Stromverbrauch dabei gemessen.²²

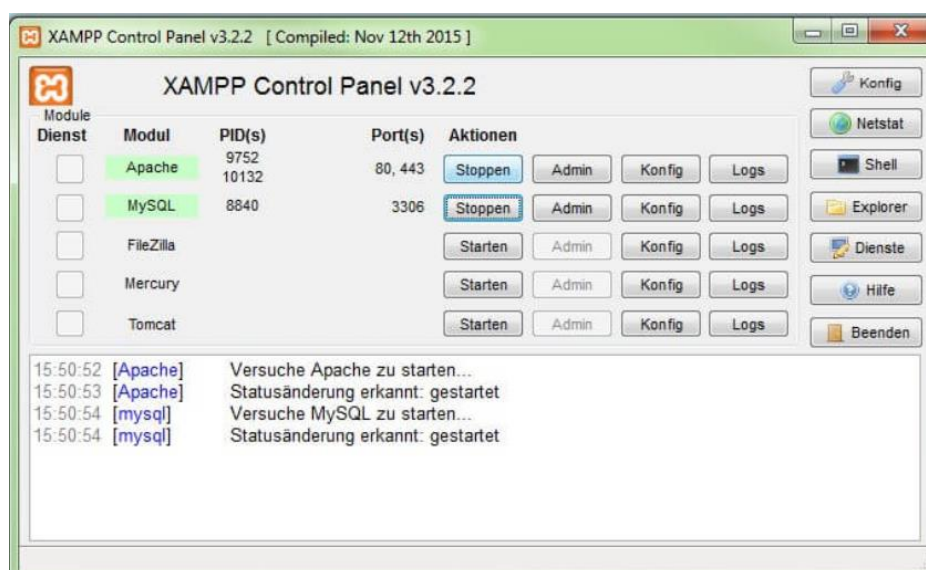


Abbildung 16: XAMPP Schalttafel

10.5 XCP-ng Center

Beim XCP-ng Center handelt es sich um eine OpenSource-Virtualisierungslösung, welche auf Basis von XEN arbeitet. Das Center bietet zahlreiche Möglichkeiten von automatischen Backups, das Erstellen und Verschieben von VMs im laufenden Betrieb und zahlreiche andere Funktionen. Wir nutzten dieses Tool als grafische Steueroberfläche für unseren HP ProLiant Server G6. Mit Hilfe von XCP-ng Center konnte der Server per IP hinzugefügt werden und basierend auf dem Server auch auf die darauf liegenden VMs. Somit war es möglich die VMs direkt über diese Oberfläche zu starten oder zu beenden und auf die Kommandozeile der einzelnen VMs zuzugreifen.²³

²² <https://www.ionos.at/digitalguide/server/tools/xampp-tutorial-so-erstellen-sie-ihren-lokalen-testserver/>

²³ <http://docs.linuxmuster.net/de/latest/getting-started/install-on-xcp-ng/>

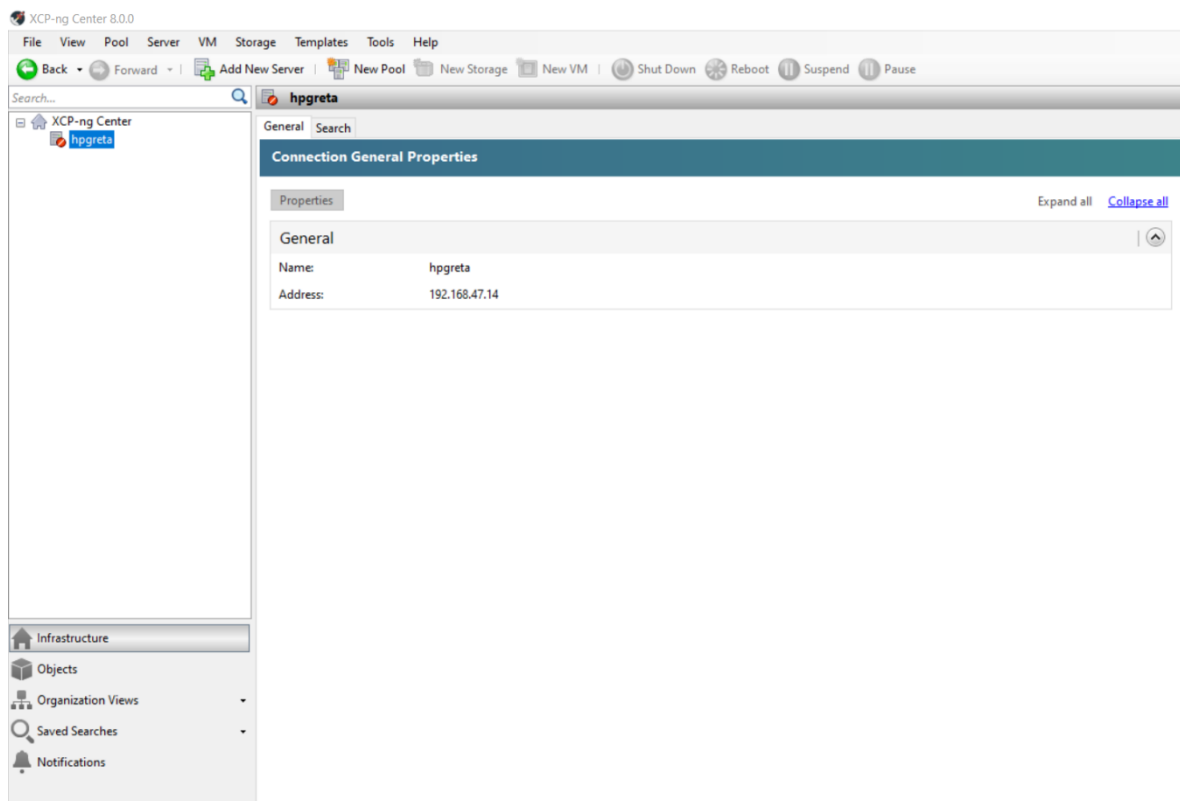


Abbildung 17: XCP-ng Center

11 Messzenario 1 - Streaming

Bei den Messungen wurden unterschiedliche Testszenarien durchgeführt. Beim ersten Szenario wurden mit ffmpeg explizit **ein** Userzugriff simuliert. Im zweiten Szenario versuchten wir mit Hilfe von JMeter Traffic zu generieren, um Multiuserzugriffe auf den Streamingserver zu simulieren. Auf das dritte Testszenario (Webseitenzugriff Messungen) wird in dieser Projektarbeit nicht näher eingegangen, möchten uns aber gerne auf das Partnerprojektteam beziehen, welches die Auswertung der Daten in ihre Arbeit übernommen hat.

Die Prozessor Power (CPU) des Intel Powergadget Tools errechnet sich aus der Summe von IA Energy (Energie der CPU Kerne), der GT Energy (Energie der Prozessorgrafikkarte) und anderen Komponenten.

Bei der Mittelwertberechnung werden die ersten vier Sekunden und die letzten vier Sekunden des Streams weggelassen, da beim Starten und Beenden die größten Fluktuationen (Spikes) entstehen.

Bei Szenario eins wurden mit Hilfe des ffmpeg Befehls die Encoder Einstellungen im NGINX Konfigurationsfile für das Beispielvideo „Big Buck Bunny“ durchgeführt. Mit Hilfe von OBS Studio wurde das Video an den NGINX Server weitergeleitet. Dabei wurde der Energieverbrauch mit Hilfe des Intel Power Gadgets und des TP-Links für den HP Server ProLiant und den HP Elite Desk 8300 G2 gemessen.

11.1 Messungen Streaming Server (HP Elite Desk 8300 G2)

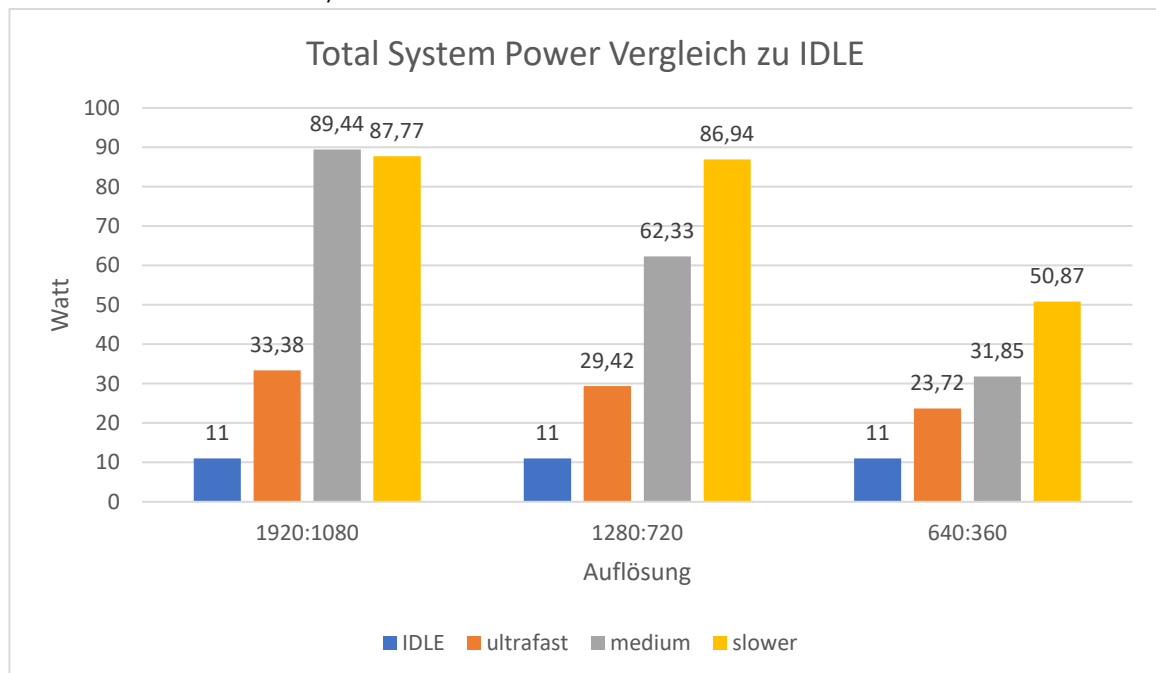
Die **Tabelle HP Elite Desk 8300 G2** zeigt die durchgeführten Messungen am HP Elite Desk 8300 G2. Dabei wurden jeweils zwei Messungen (A und B) mit der gleichen Einstellung durchgeführt und bei der Auswertung die aussagekräftigeren Ergebnisse herangezogen. Die verwendeten Auflösungen sind 1920:1080 mit einer Bitrate von 6000 Kilobits pro Sekunde, 1280:720 mit einer Bitrate von 4500 Kilobits und 640:360 mit 3000 Kilobits pro Sekunde. Als Orientierungshilfe für die zu verwendete Auflösung und Bitrate wurden die empfohlenen Einstellungen der Streaming Plattform Twitch herangezogen. Bei den diversen Auflösungen wurden jeweils drei Messungen mit unterschiedlicher Preset-Einstellung (medium, ultrafast, slower) durchgeführt. Preset wurde bereits in den vorderen Kapiteln erklärt.

Zu jeder Messung gibt es immer zwei Diagramme. Das Diagramm mit CPU POWER im Titel enthält die Auswertung der Daten mithilfe des Intel Power Gadget und zeigt ausschließlich die beanspruchte Leistung der CPU in Watt. Das zweite Diagramm mit SYS-POWER im Titel zeigt die Daten, welche mit dem TP-LINK HS110 aufgenommen wurde und somit representieren diese den Verbrauch des gesamten Computer Systems!

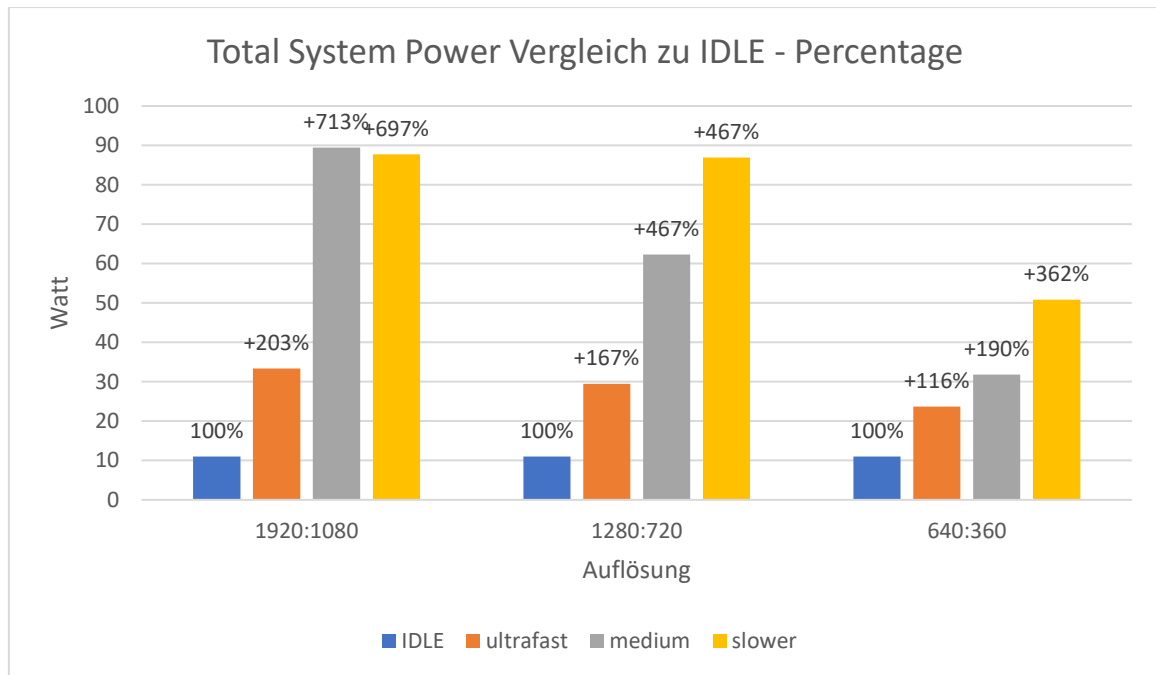
HP Elite Desk 8300 G2								
Auflösung	Bitrate	Encoder	FPS	Preset	CRF	SYS POWER average [W]	CPU POWER average [W]	Mess. Nr.
1920:1080	6000	CBR	60	medium	23	89.44 (1b)	62.05 (1b)	1
1920:1080	6000	CBR	60	ultrafast	23	33.38 (2a)	16.59 (2a)	2
1920:1080	6000	CBR	60	slower	23	87.77 (3b)	60.57 (3b)	3
1280:720	4500	CBR	60	medium	23	62.33 (4b)	40.36 (4b)	4
1280:720	4500	CBR	60	ultrafast	23	29.42 (5a)	13.91 (5a)	5
1280:720	4500	CBR	60	slower	23	86.94 (6b)	60.71 (6b)	6
640:360	3000	CBR	60	medium	23	31.85 (7b)	16.31 (7b)	7
640:360	3000	CBR	60	ultrafast	23	23.72 (8b)	9.83 (8b)	8
640:360	3000	CBR	60	slower	23	50.87 (9b)	31.92 (9b)	9

Tabelle 6: HP Elite Desk 8300 G2 – Messungen

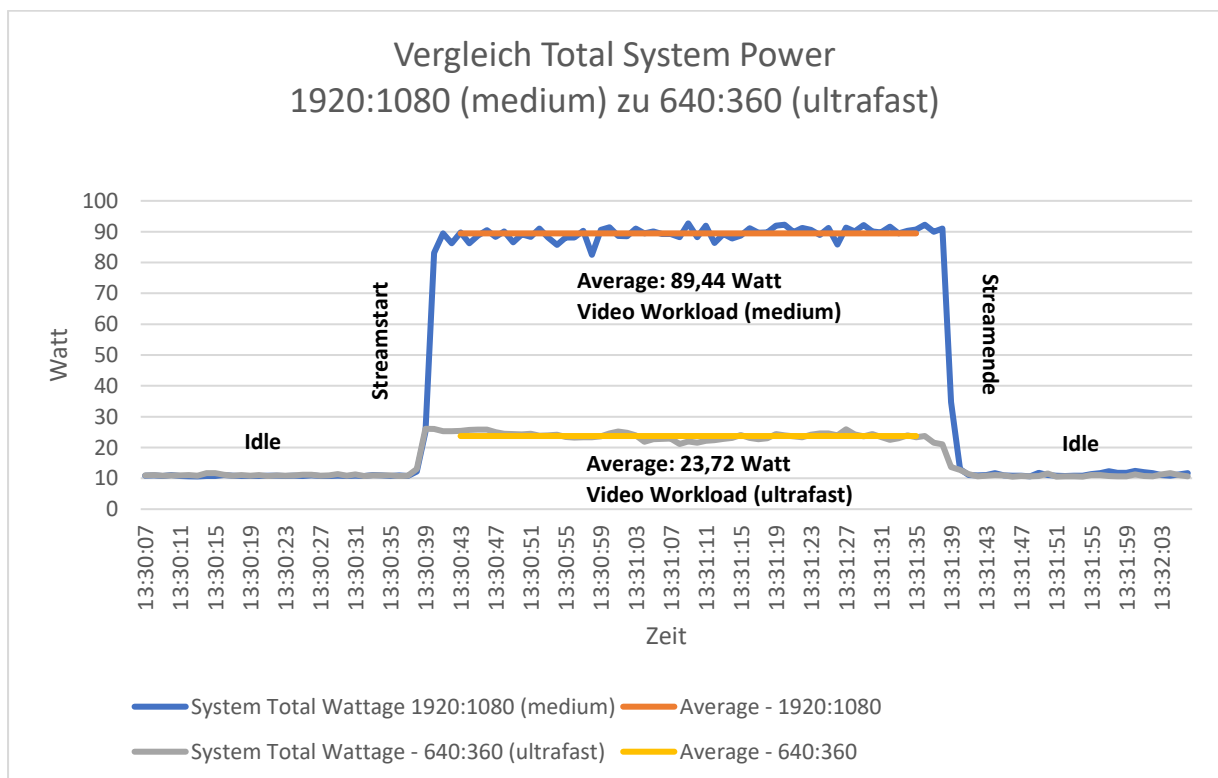
11.1.1 Resultate und Analyse



Messdiagramm 1

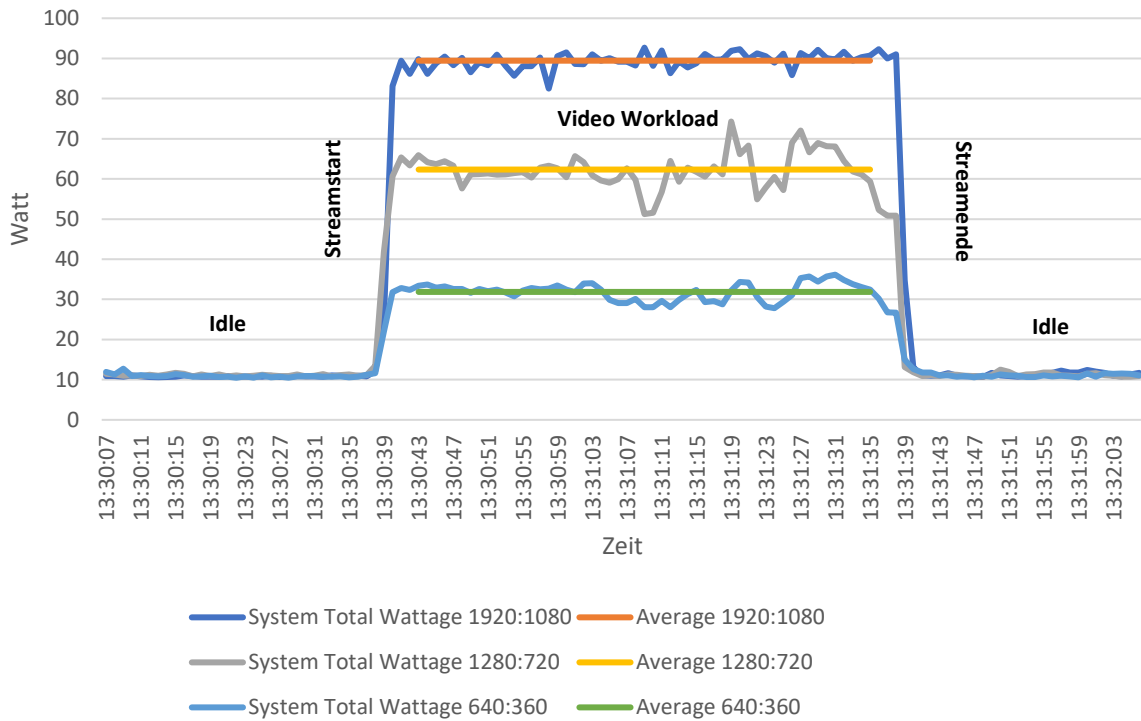


Messdiagramm 2

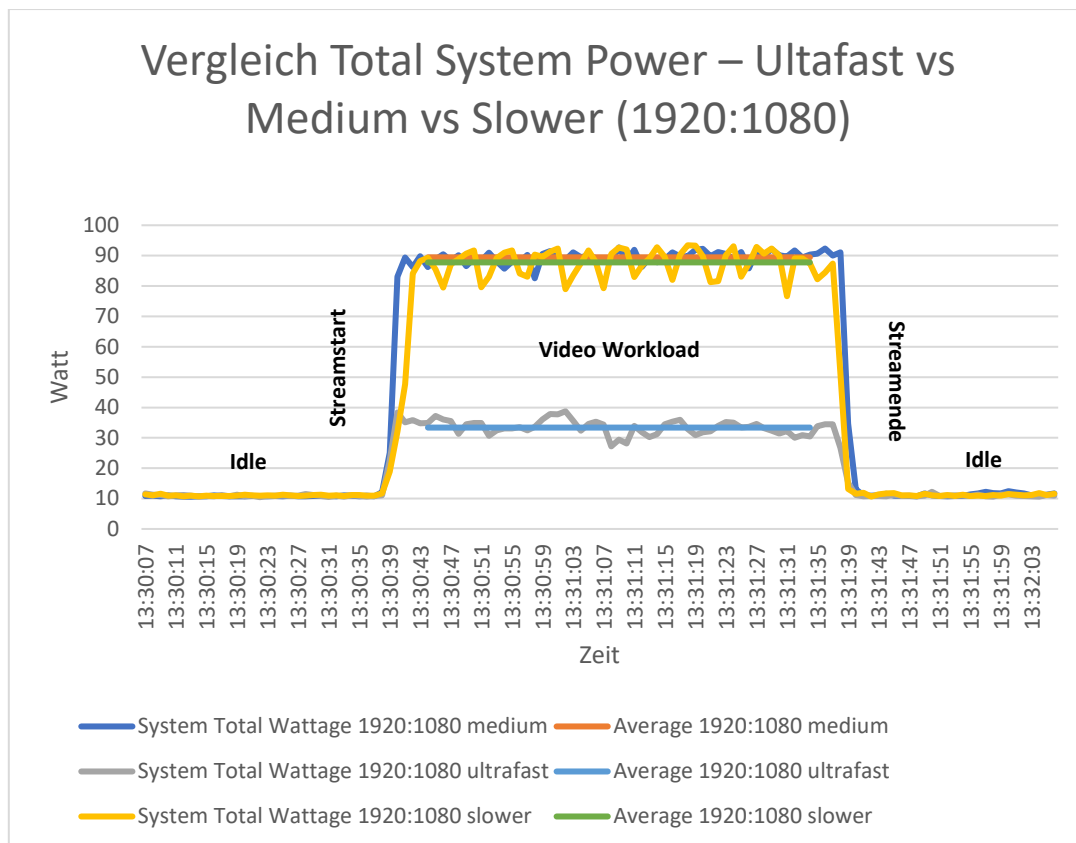


Messdiagramm 3

Vergleich Total System Power – 1920:1080 vs 1280:720 vs 640:360 (medium)



Messdiagramm 4



Messdiagramm 5

Die Balkendiagramme 1 und 2 zeigen die Auswirkung auf die Leistung bei unterschiedlicher Auflösung und Preset Einstellungen. Zur besseren Veranschaulichung wurde auch der IDLE Wert in das Diagramm eingefügt.

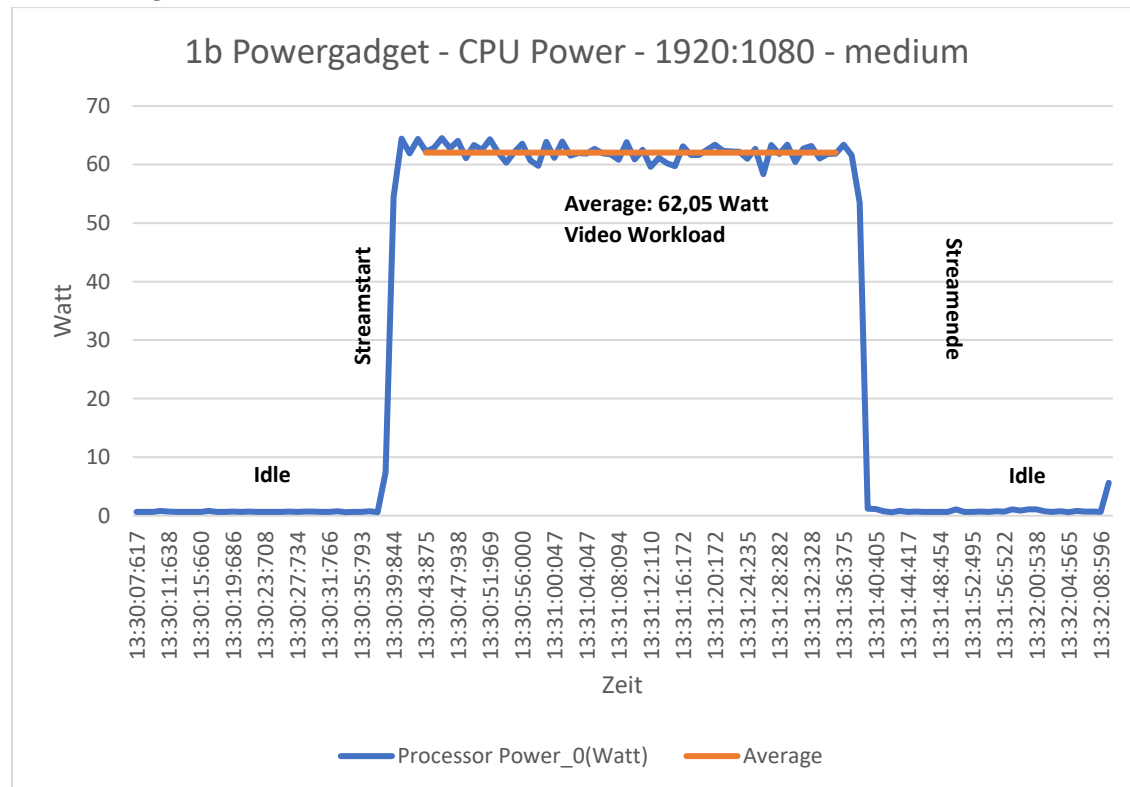
Bei einer Auflösung von 1920x1080 ist ein interessanter Effekt zu beobachten: Bei den Presets Medium und Slower entsteht kein Leistungsanstieg! Dies ist darauf zurückzuführen, dass hier eine Maximalauslastung des Systems auftritt da jede CPU festgelegte Limits bezüglich Leistung und Temperatur einhält, welche in diesem Fall erreicht wurden. Er bei einer niedrigeren Auflösung lässt sich wieder zwischen Medium und Slower unterscheiden da hier die Limits bei Medium nicht erreicht werden.

Es ist klar erkennbar, dass das Preset ultrafast keinen signifikanten Leistungsanstieg bei verschiedenen Auflösungen zur Folge hat da dieses Preset nur eine sehr geringe CPU Auslastung erzeugt jedoch auch eine schlechtere Bildqualität zufolge hat. Zudem fällt bei einer Auflösung 640:360 der Unterschied zwischen den Presets im Vergleich zu den anderen Auflösungen minimaler aus.

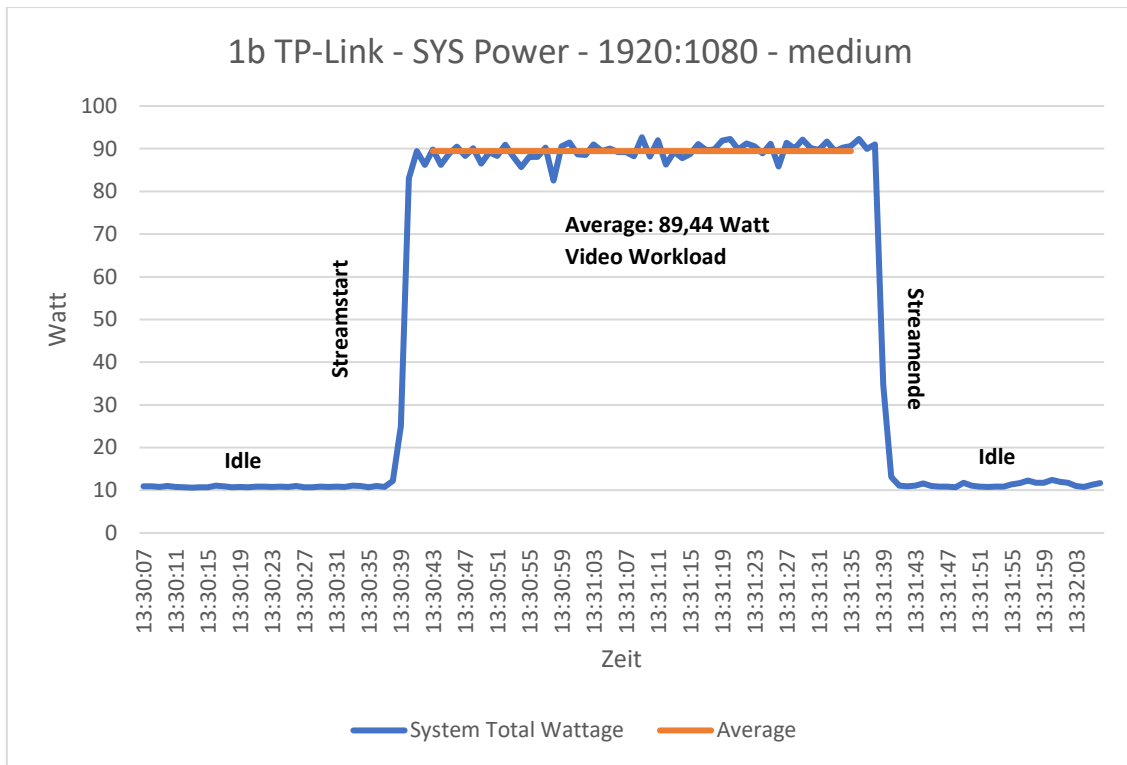
Zusätzlich ist deutlich erkennbar, dass die verschiedenen Presets einen größeren und signifikanteren Einfluss auf den Leistungsverbrauch haben als die Auflösung. Hierbei ist natürlich die Messung mit der Auflösung 1920x1080 auszuschließen da hier die CPU an die thermischen und leistungsspezifischen Limits geraten ist.

Vergleicht man die Auflösungen mit der Preset Einstellung Medium untereinander, benötigt wie erwartet die Full HD Auflösung die meiste Energie mit rund 89,44 Watt. Dahinter folgt die Auflösung 1280:720 mit 62 Watt was bereits einer Reduktion von 30% entspricht. Am wenigsten Leistung benötigt die Auflösung 640:360 da hier gerade einmal durchschnittlich 31 Watt verbraucht werden was bezüglich der Full HD Auflösung einer Reduktion von 65% entspricht.

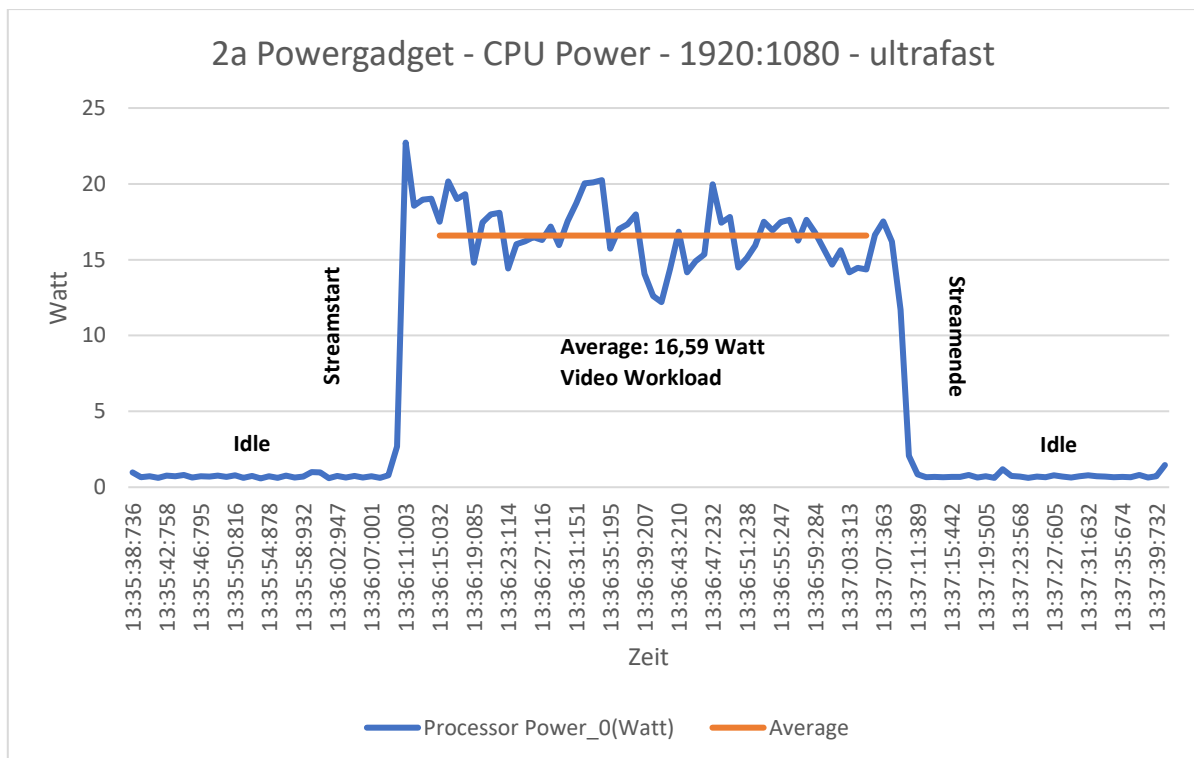
11.1.2 Diagramme



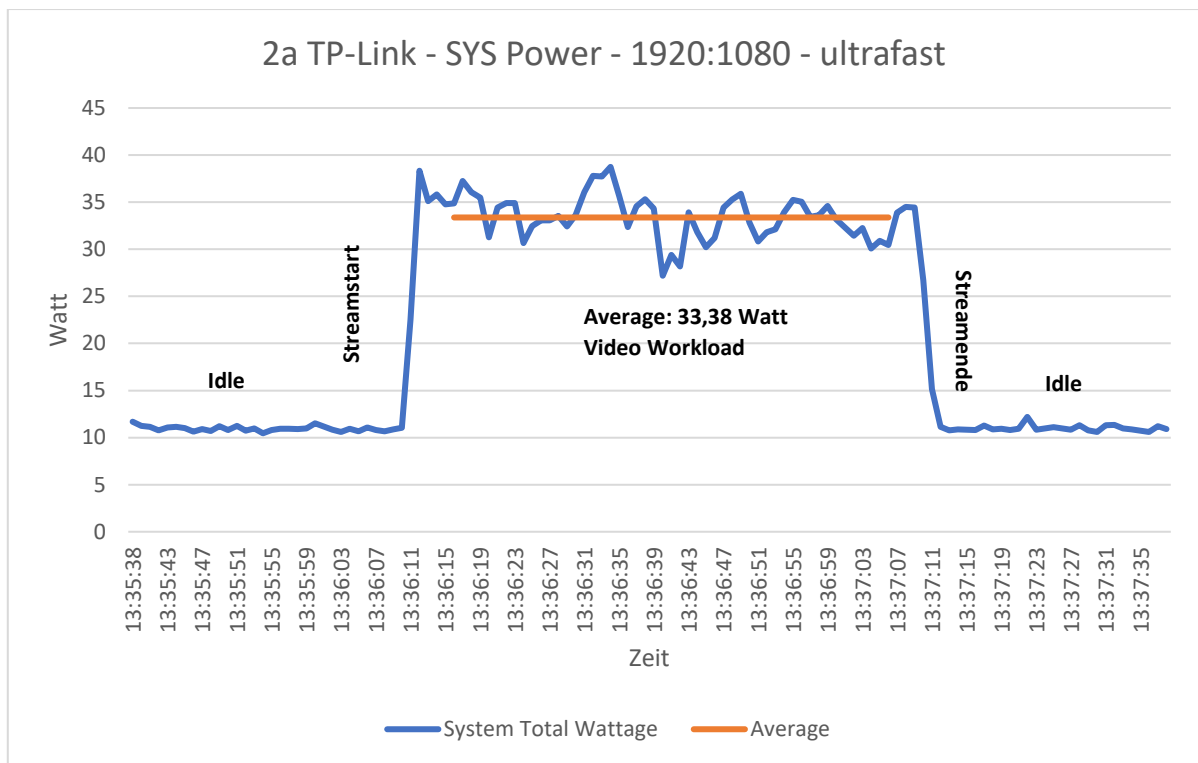
Messdiagramm 6



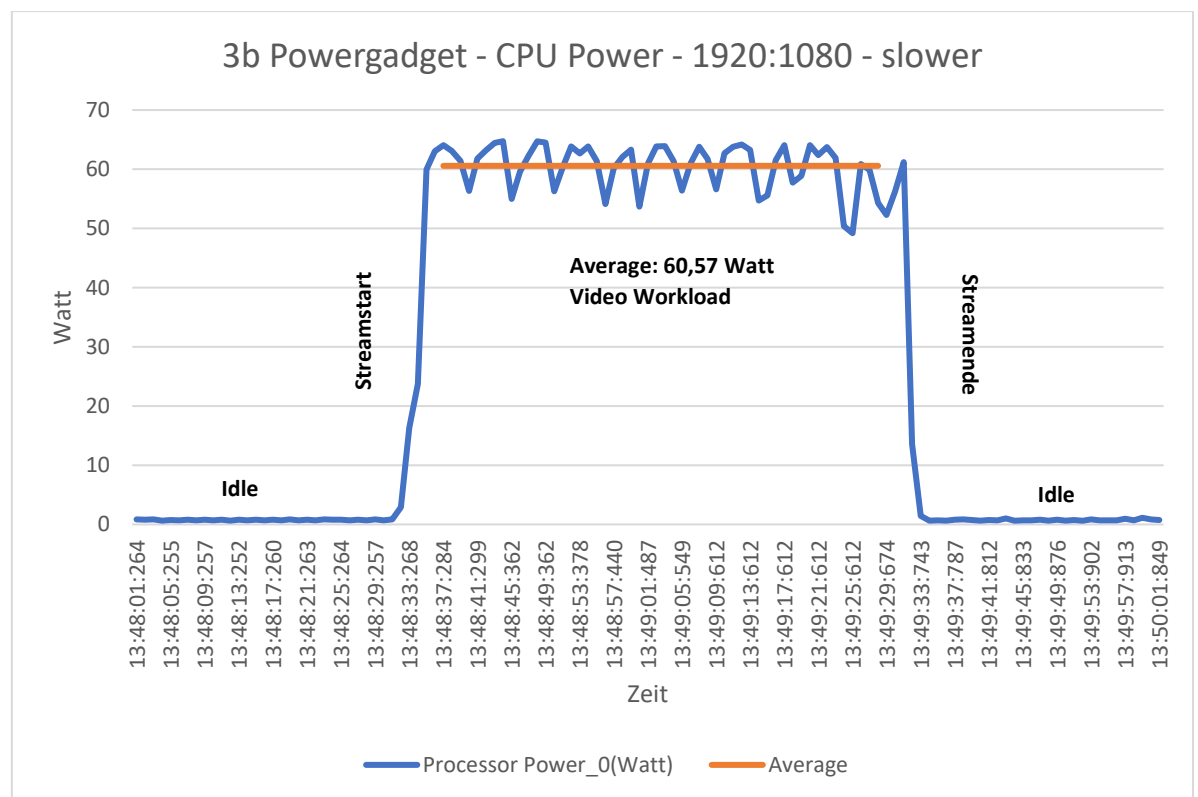
Messdiagramm 7



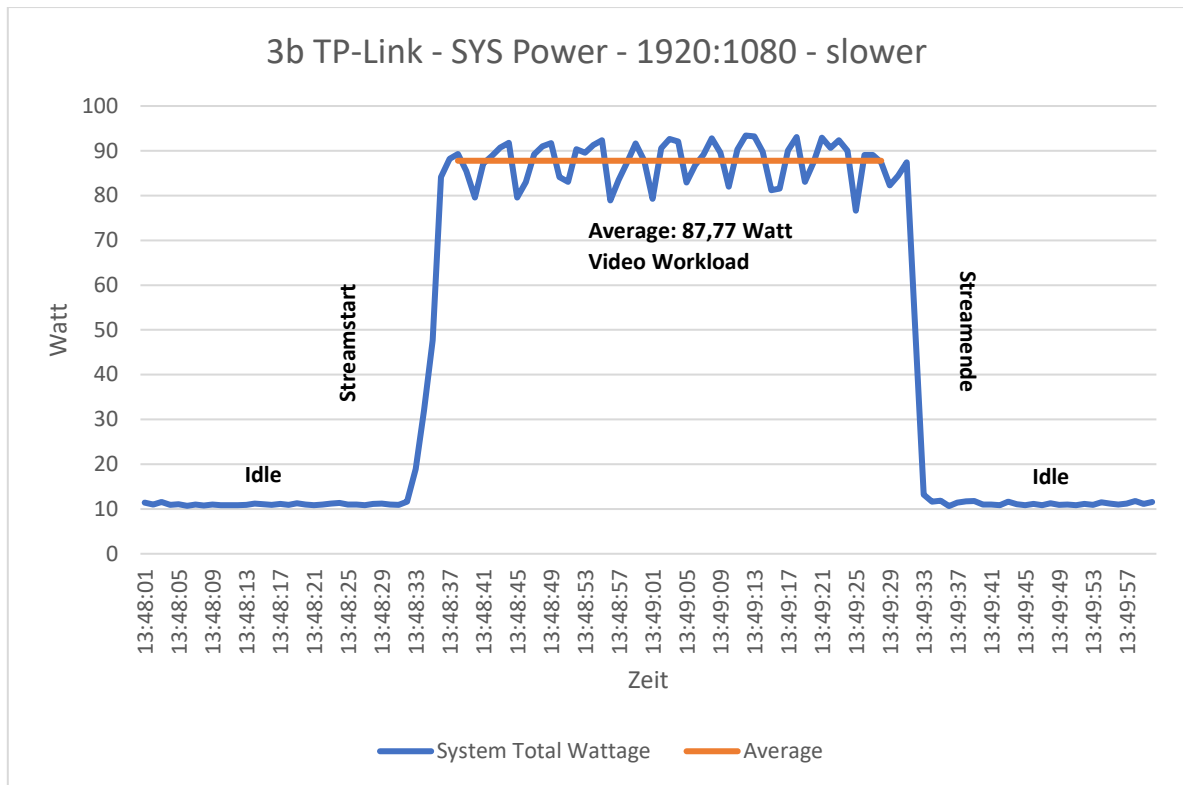
Messdiagramm 8



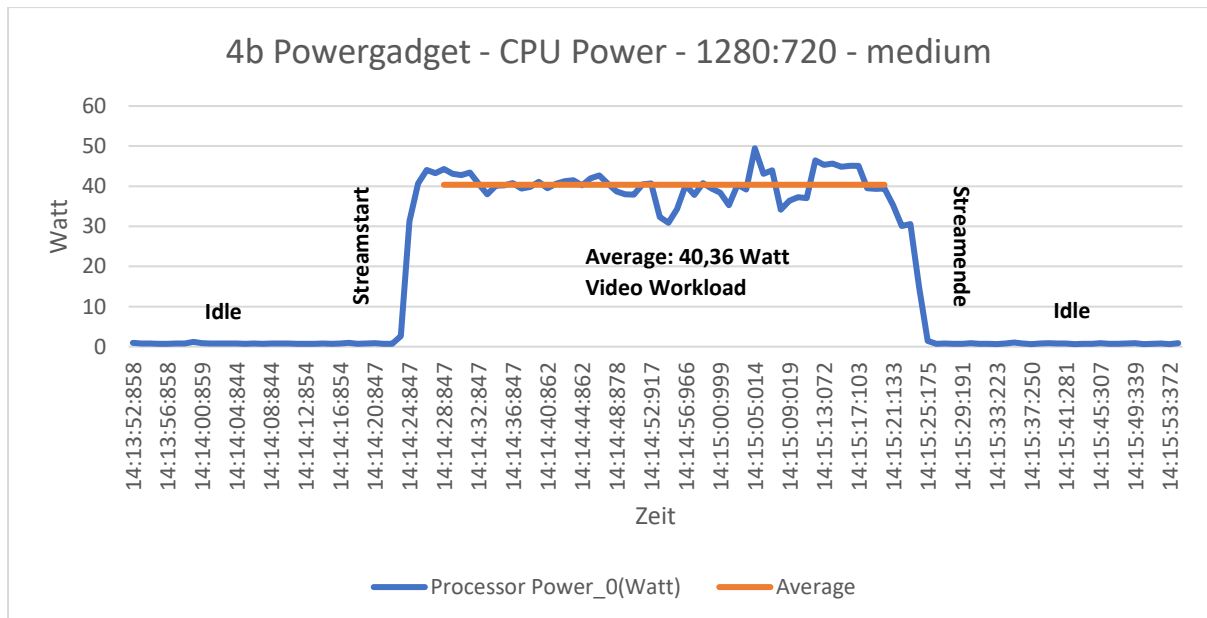
Messdiagramm 9



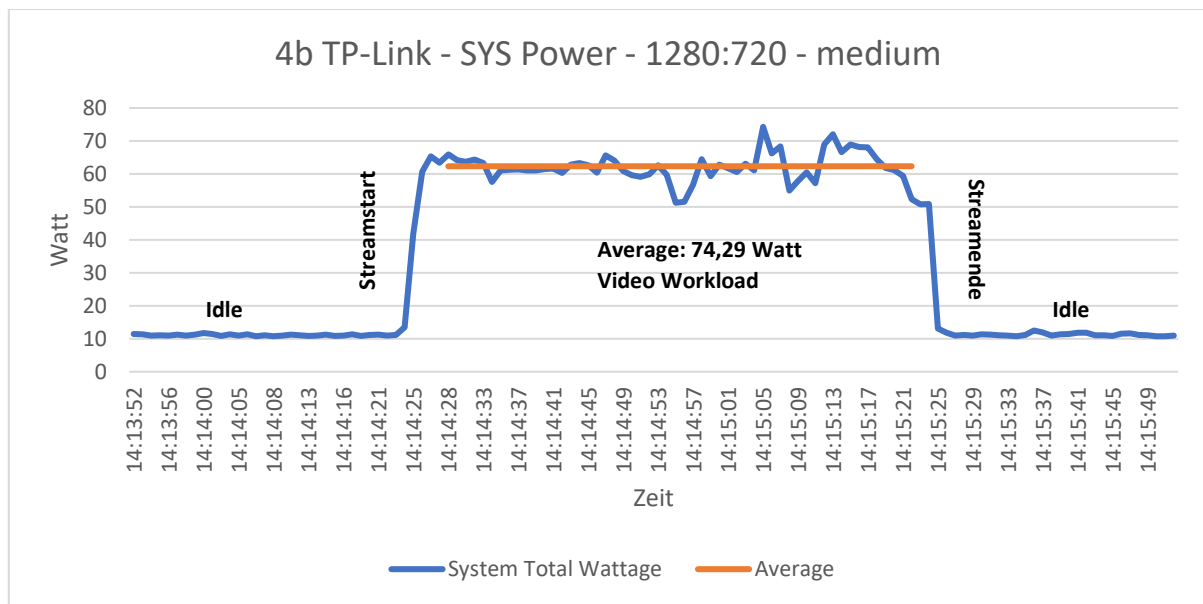
Messdiagramm 10



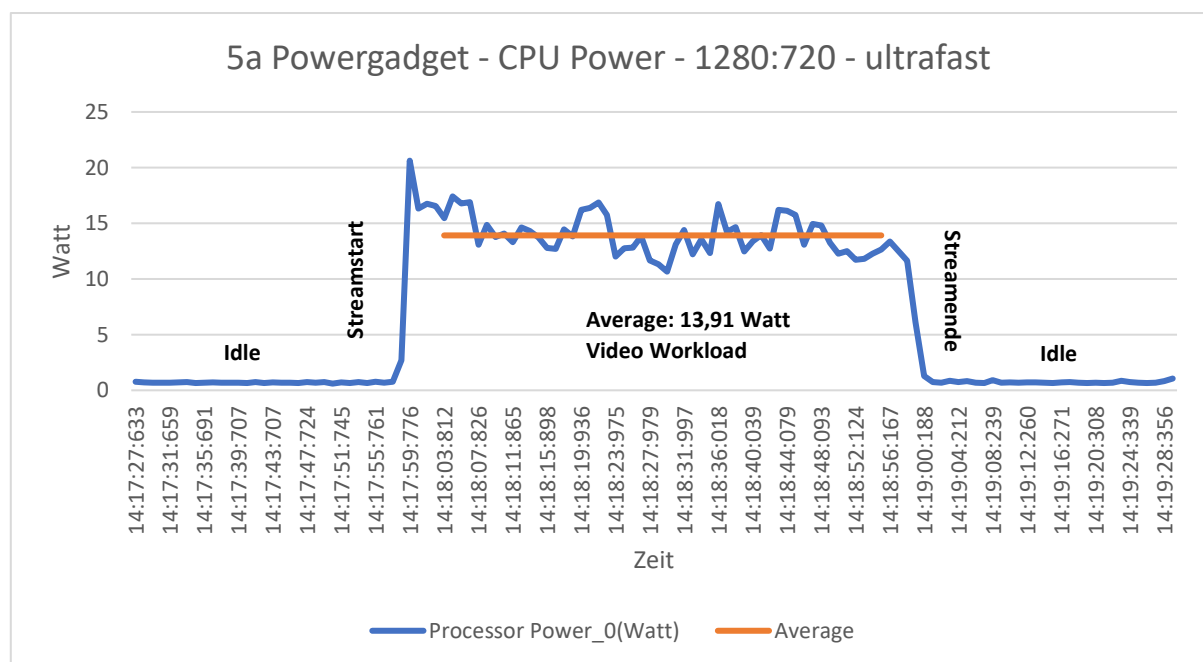
Messdiagramm 11



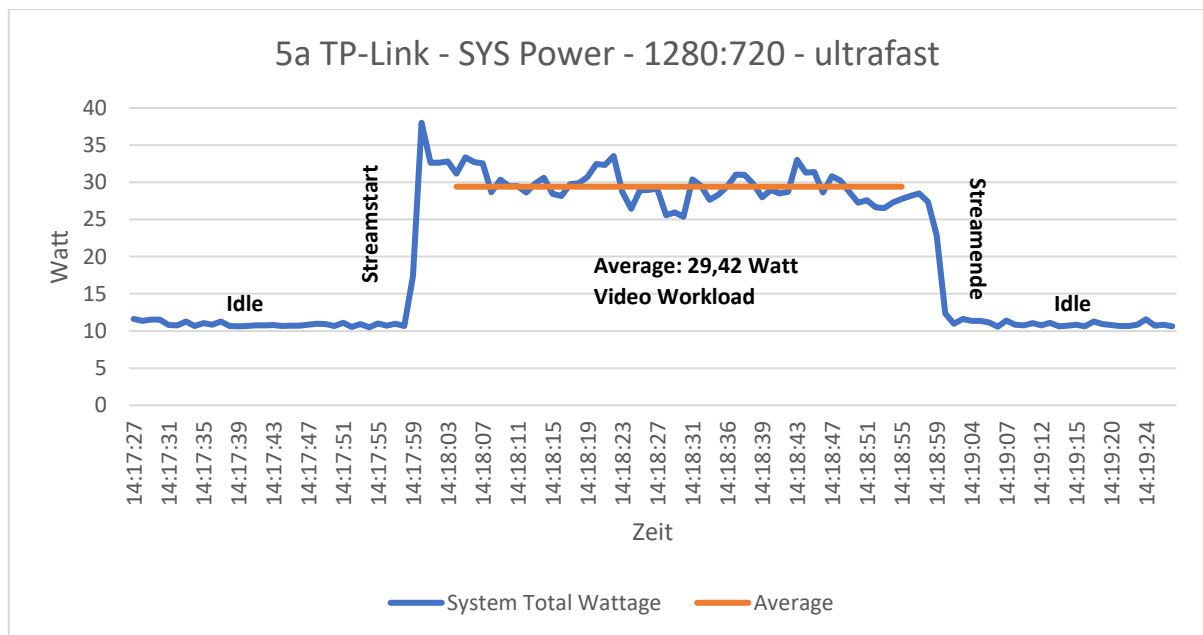
Messdiagramm 12



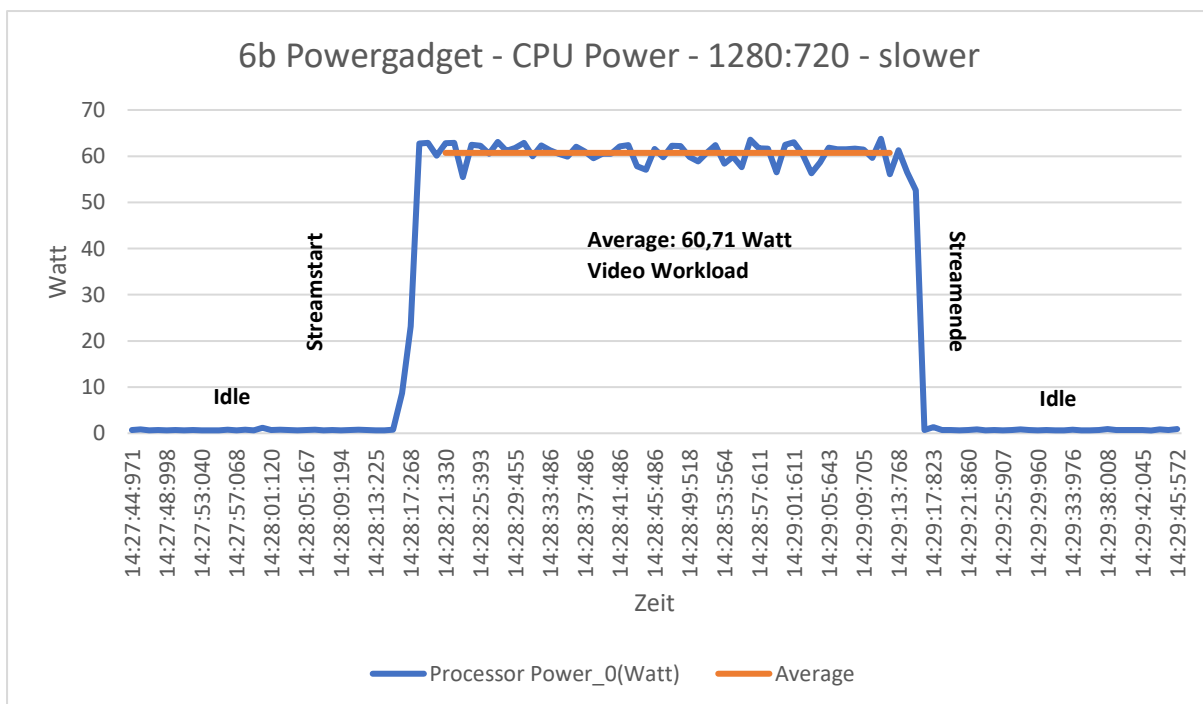
Messdiagramm 13



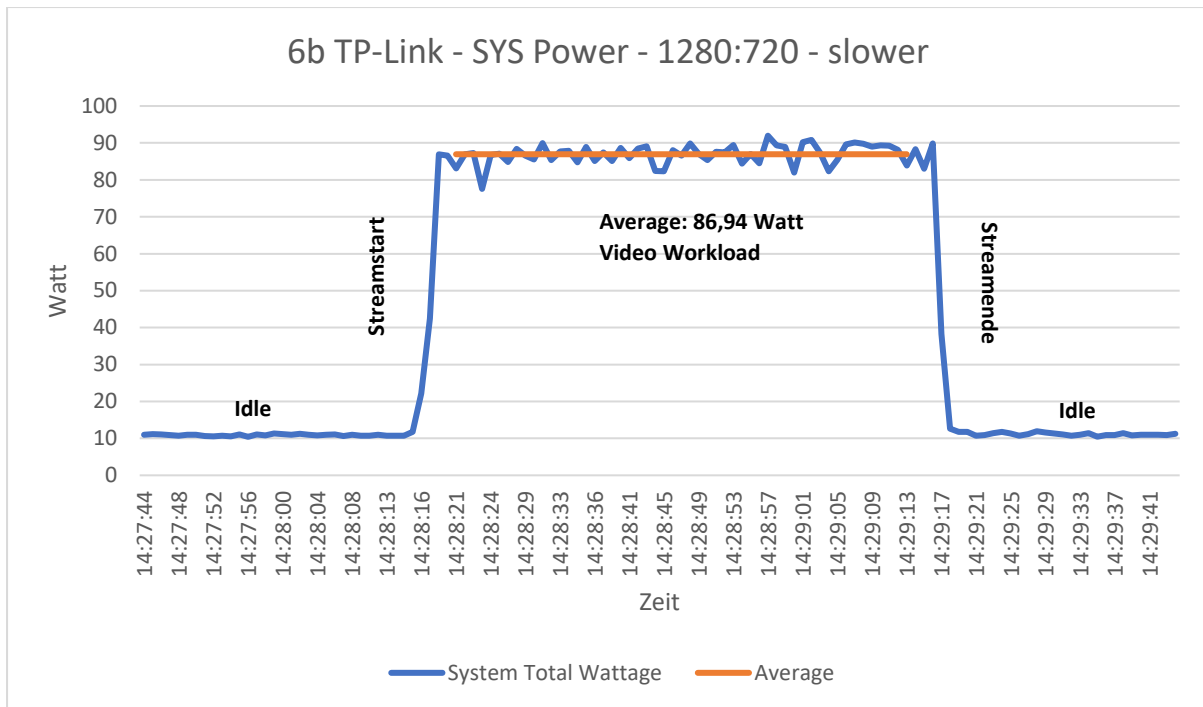
Messdiagramm 14



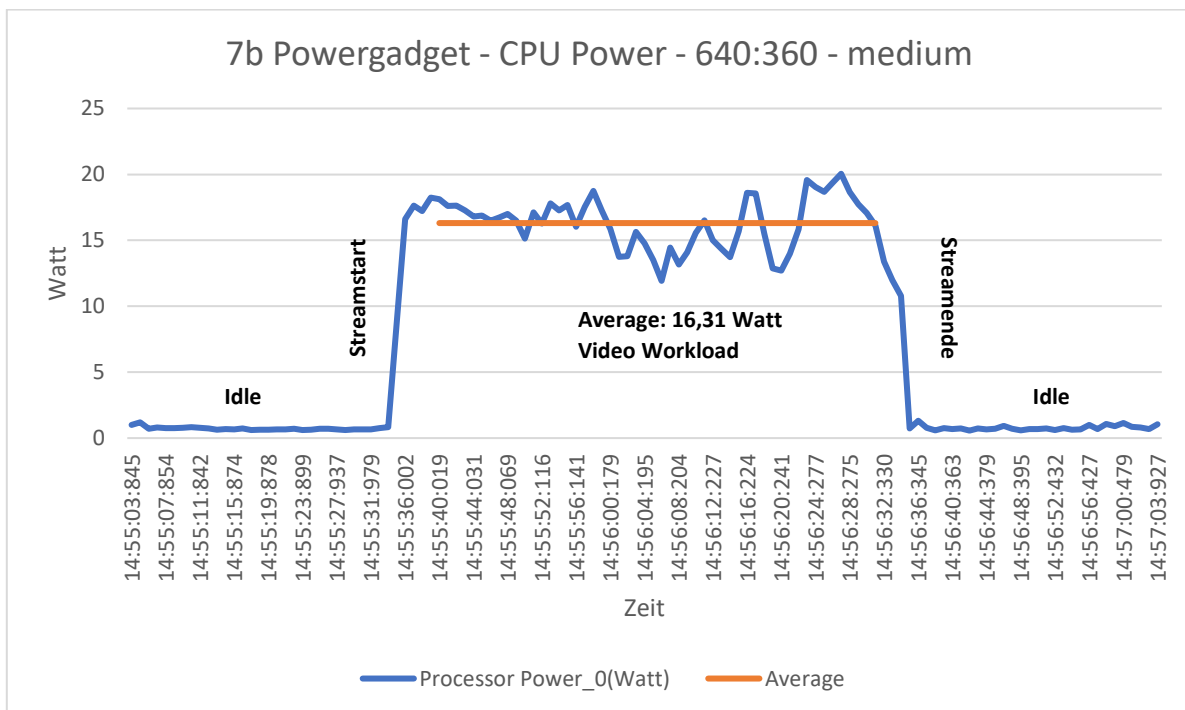
Messdiagramm 15



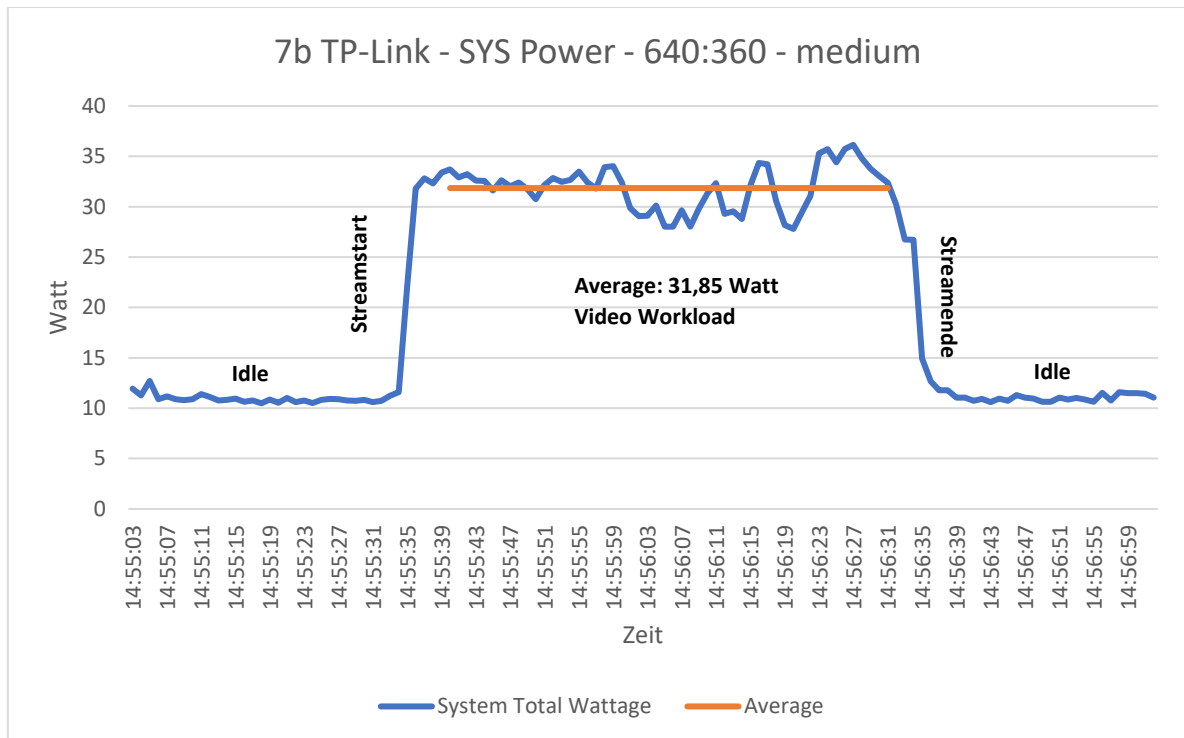
Messdiagramm 16



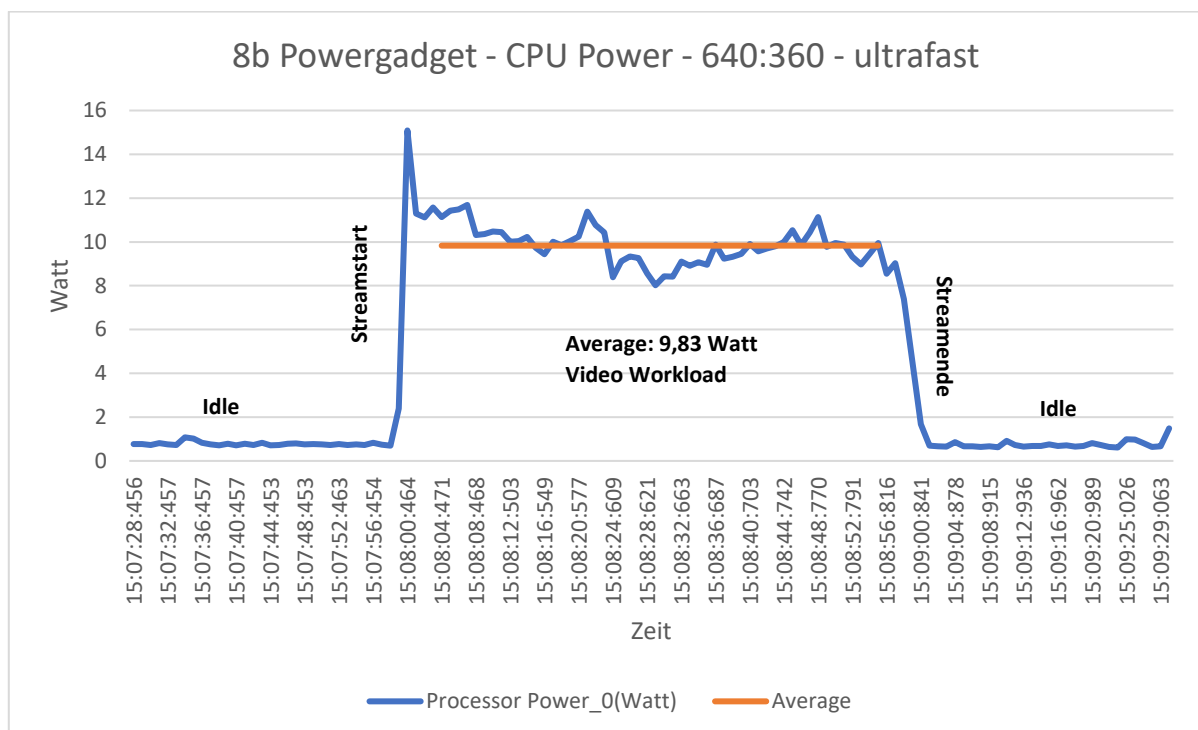
Messdiagramm 17



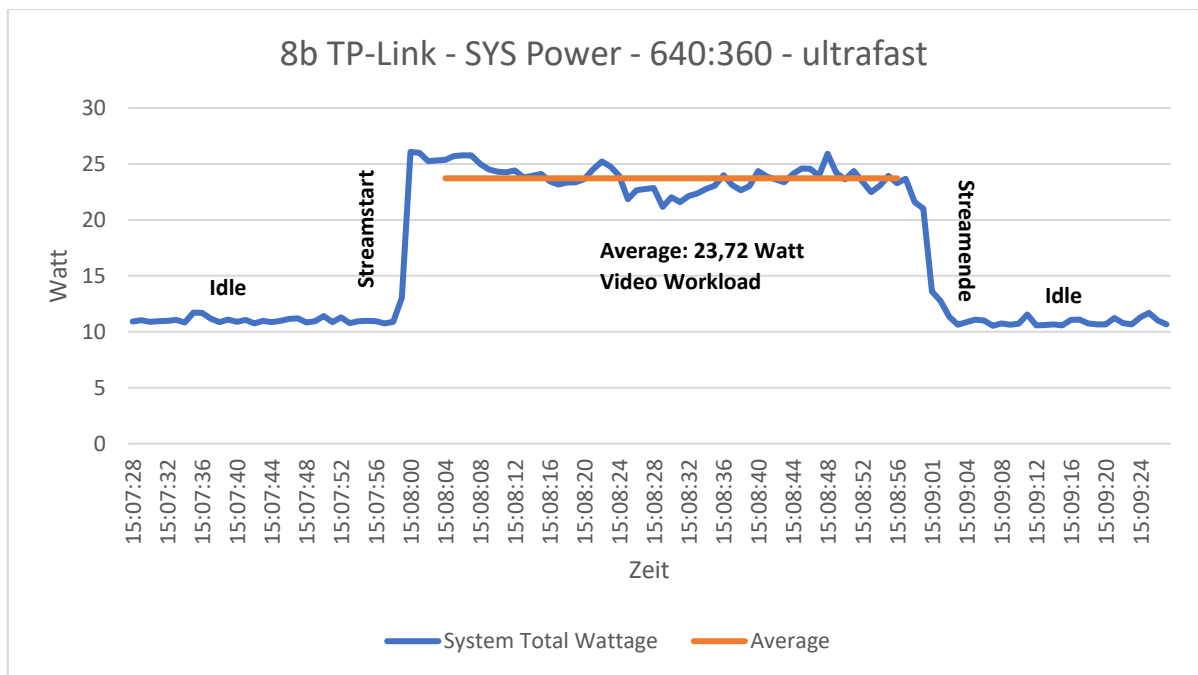
Messdiagramm 18



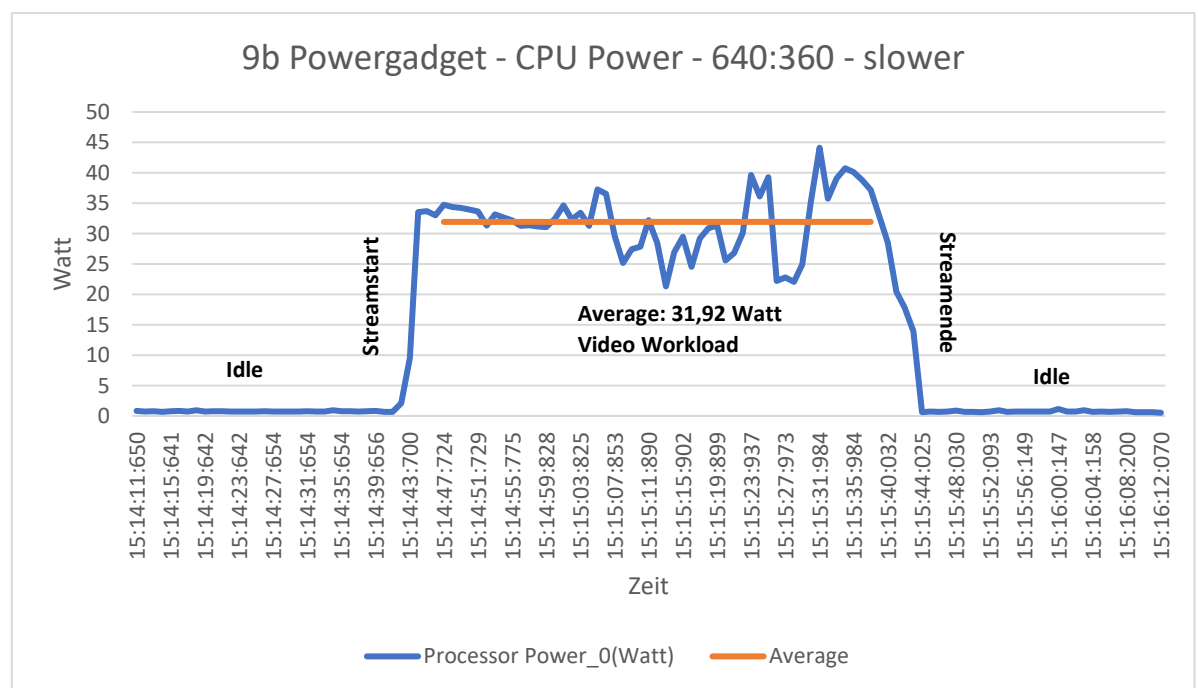
Messdiagramm 19



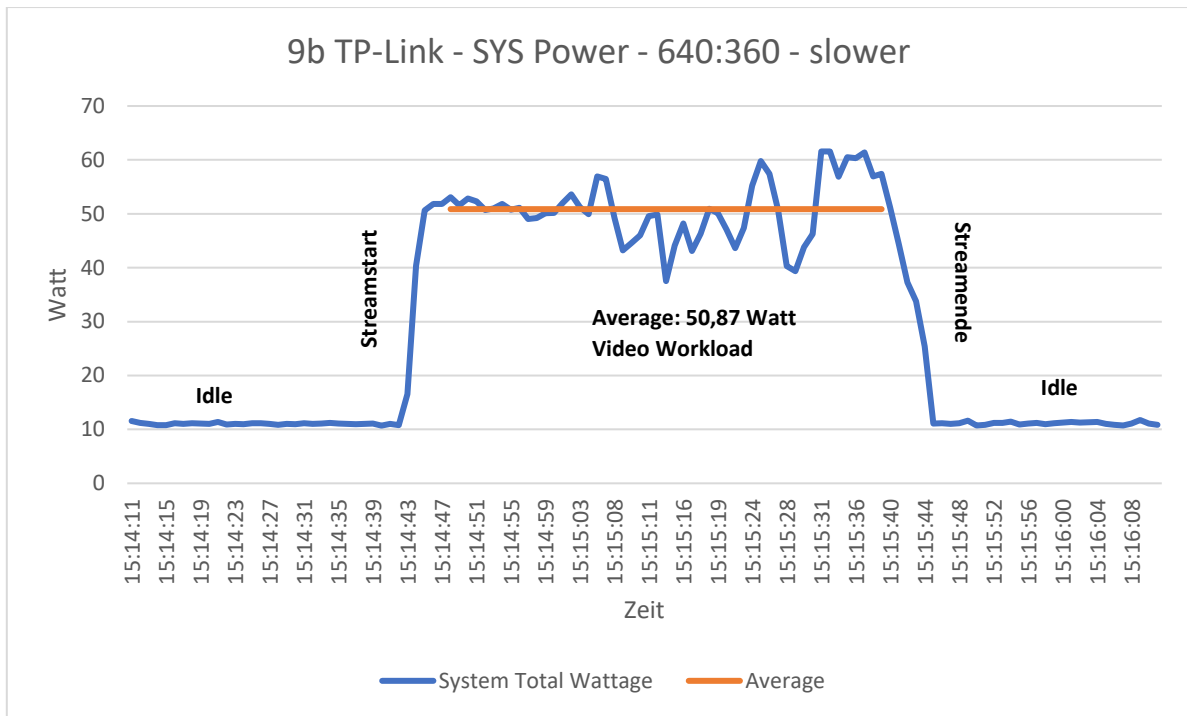
Messdiagramm 20



Messdiagramm 21



Messdiagramm 22



Messdiagramm 23

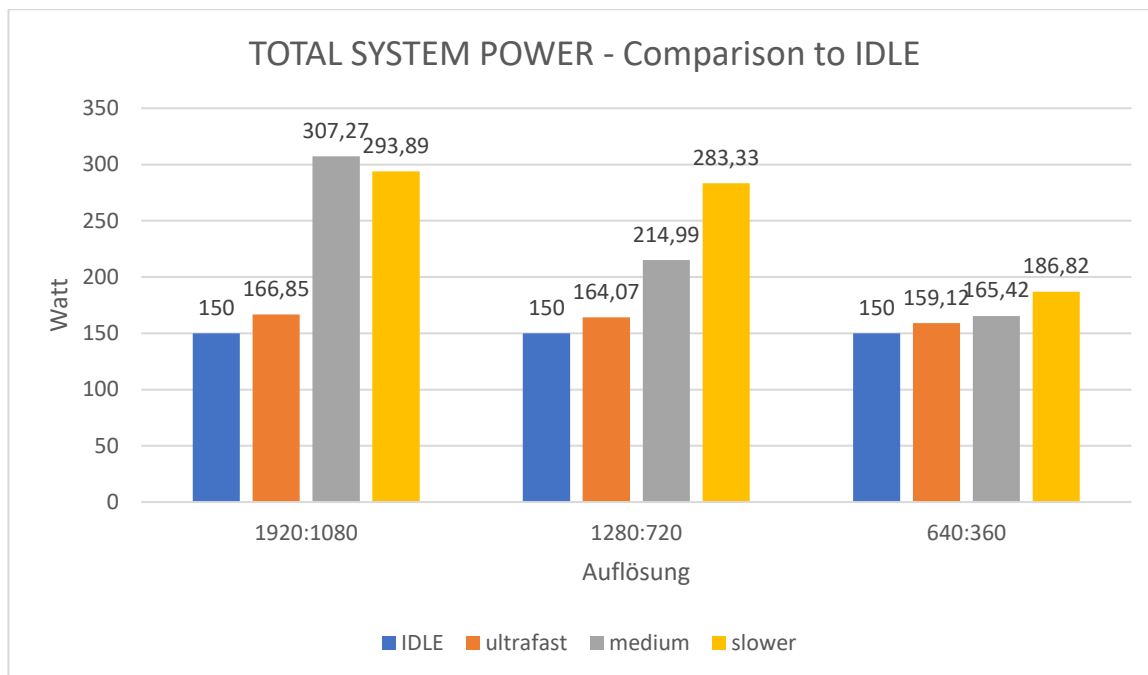
11.2 Messungen Streaming Server (HP ProLiant G6)

Die **Tabelle HP Server ProLiant G6** zeigt die durchgeführten Messungen am HP Server. Die verwendeten Auflösungen sind wie bei den Tests beim Desktop PC 1920:1080 mit einer Bitrate von 6000 Kilobits pro Sekunde, 1280:720 mit einer Bitrate von 4500 Kilobits und 640:360 mit 3000 Kilobits pro Sekunde. Für die drei unterschiedlichen Auflösungen wurden jeweils drei Messungen mit unterschiedlichen Preset-Einstellungen (medium, ultrafast, slower) durchgeführt. Da die CPU Generation HP Server ProLiant G6 das Intel Power Gadget nicht unterstützt, konnten nur die Leistungsmessung für das gesamte System durchgeführt werden!

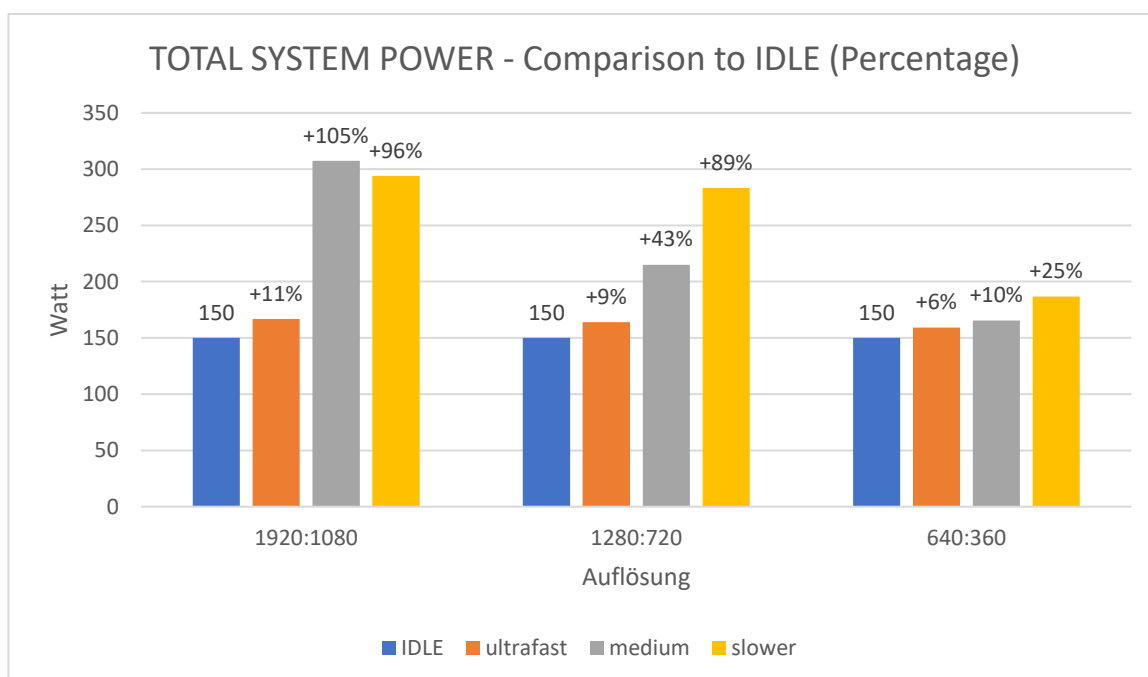
HP Server ProLiant G6							
Auflösung	Bitrate	Encoder	FPS	Preset	CRF	SYS POWER average [W]	Nr.
1920:1080	6000	CBR	60	medium	23	307.27 (1b)	1
1920:1080	6000	CBR	60	ultrafast	23	166.85 (2b)	2
1920:1080	6000	CBR	60	slower	23	293.89 (3b)	3
1280:720	4500	CBR	60	medium	23	214.99 (4b)	4
1280:720	4500	CBR	60	ultrafast	23	164.07 (5b)	5
1280:720	4500	CBR	60	slower	23	283.33 (6b)	6
640:360	3000	CBR	60	medium	23	165.42 (7b)	7
640:360	3000	CBR	60	ultrafast	23	159.12 (8a)	8
640:360	3000	CBR	60	slower	23	186.82 (9b)	9

Tabelle 7: HP Server ProLiant G6

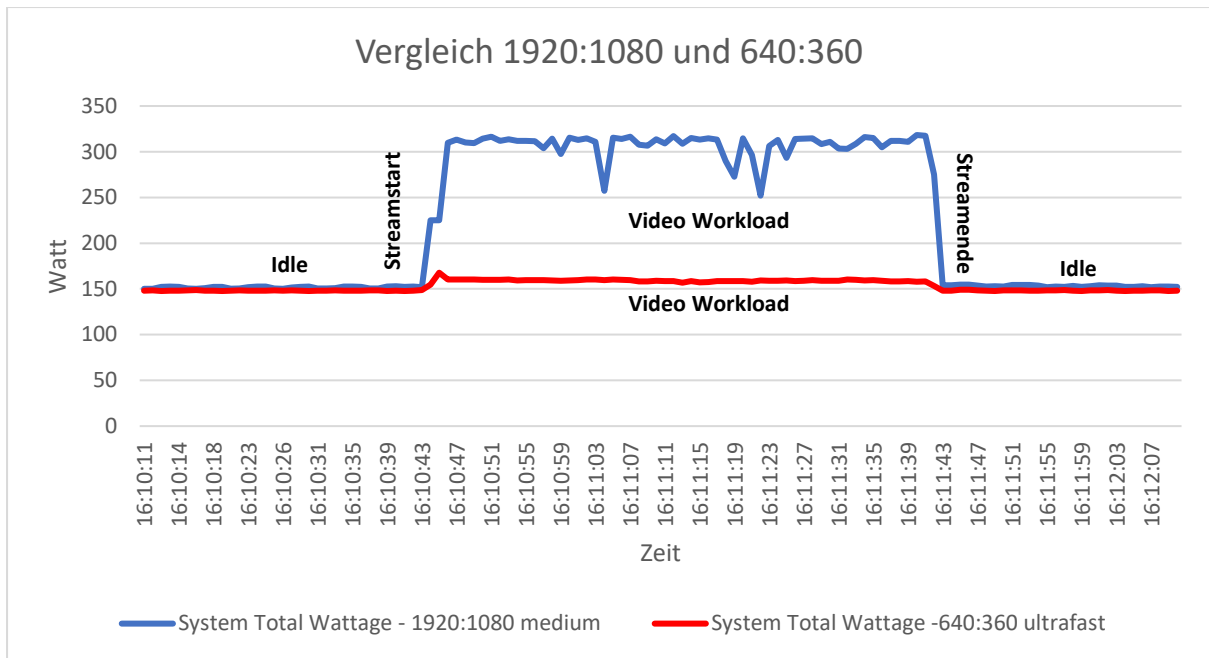
11.3 Resultate und Analyse



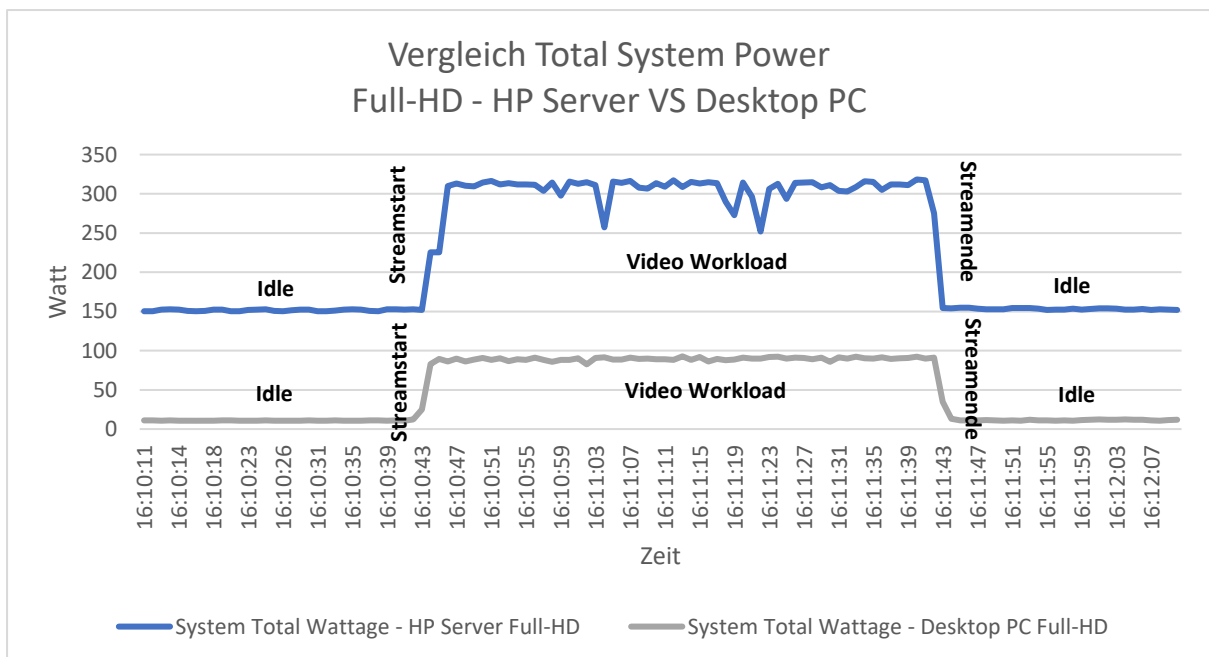
Messdiagramm 24



Messdiagramm 25

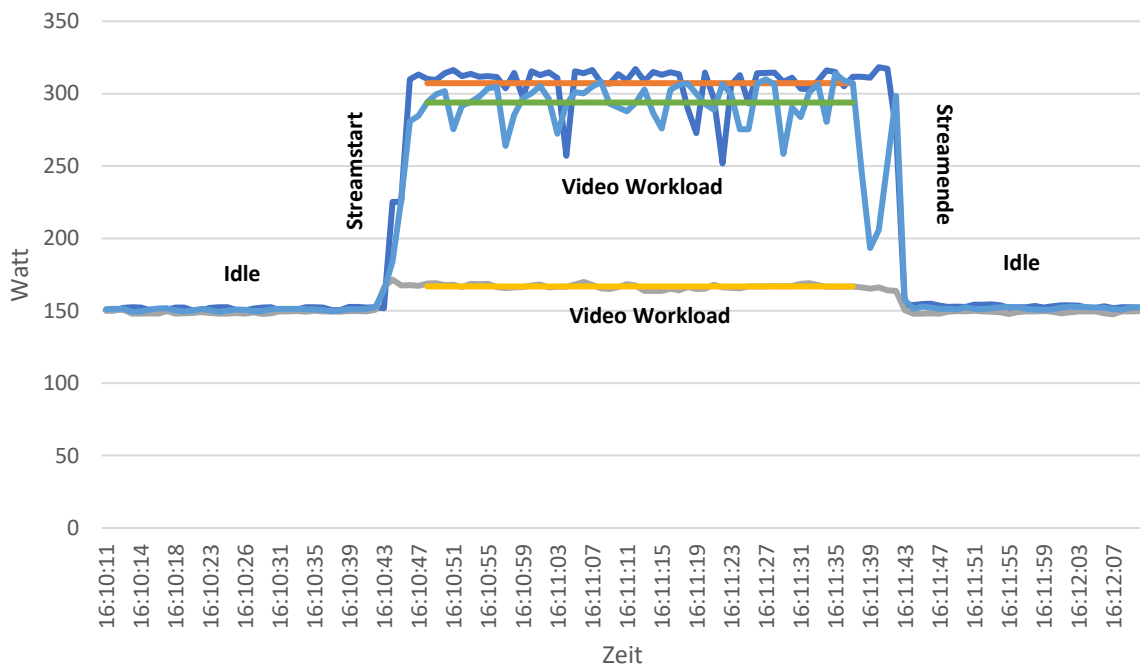


Messdiagramm 26

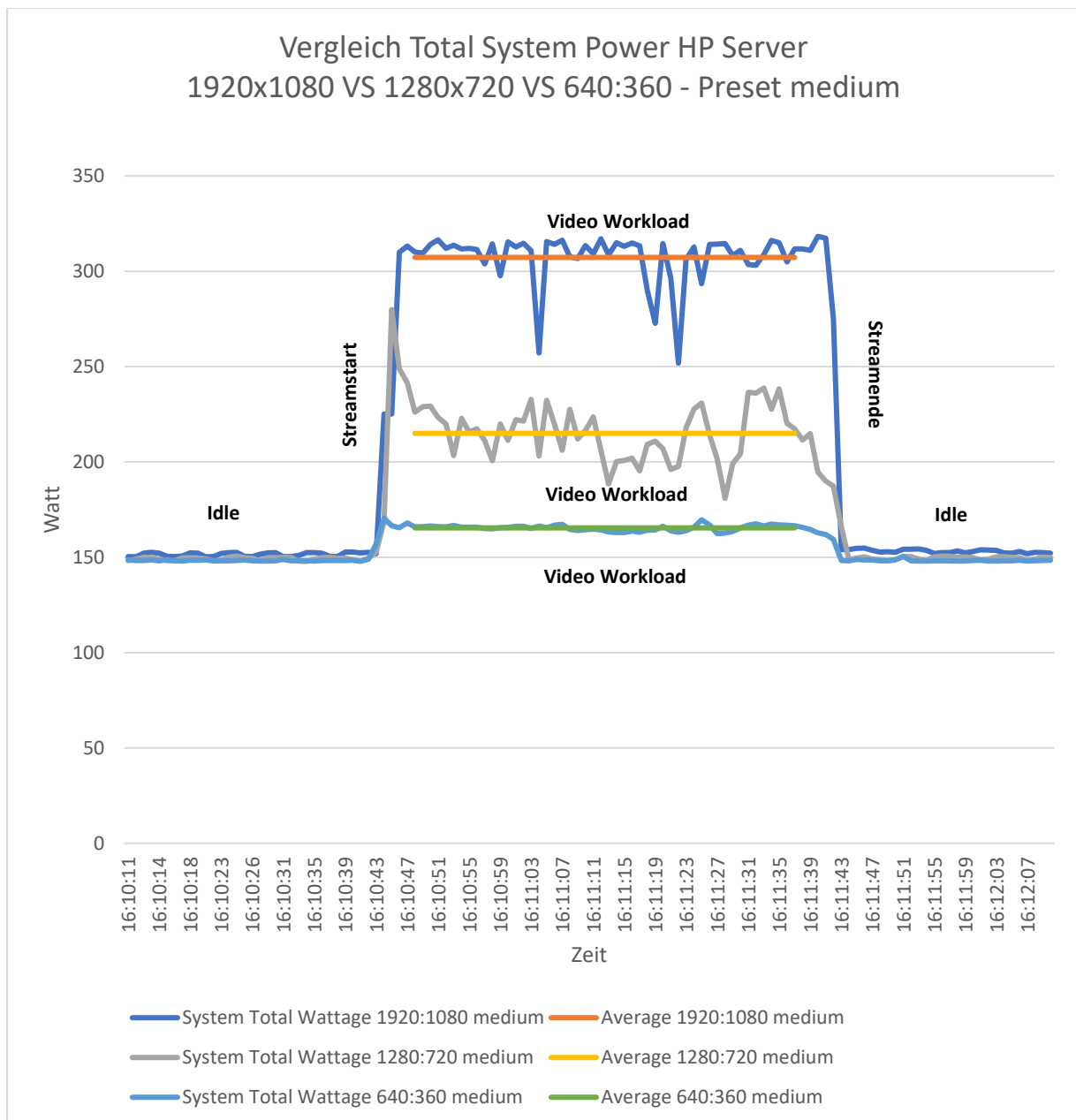


Messdiagramm 27

Vergleich Total System Power HP Server 1920x1080 ultrafast - medium -slower



Messdiagramm 28



Messdiagramm 29

Die Messungen wurden auf dem HP Server G6 durchgeführt welcher schon ein etwas älteres Modell darstellt und nur bedingt aussagekräftige Resultate liefert. Im Vergleich zu unserem Desktop-Server ist direkt ersichtlich, dass die Auslastung im IDLE mit durchschnittlich 150 Watt signifikant höher ist. Dies ist auf die ältere Hardware zurückzuführen und darauf, dass in diesem System zwei CPU's verbaut sind. Diese Differenz im Energieverbrauch der beiden Systeme wird im Messdiagramm 27 dargestellt und veranschaulicht.

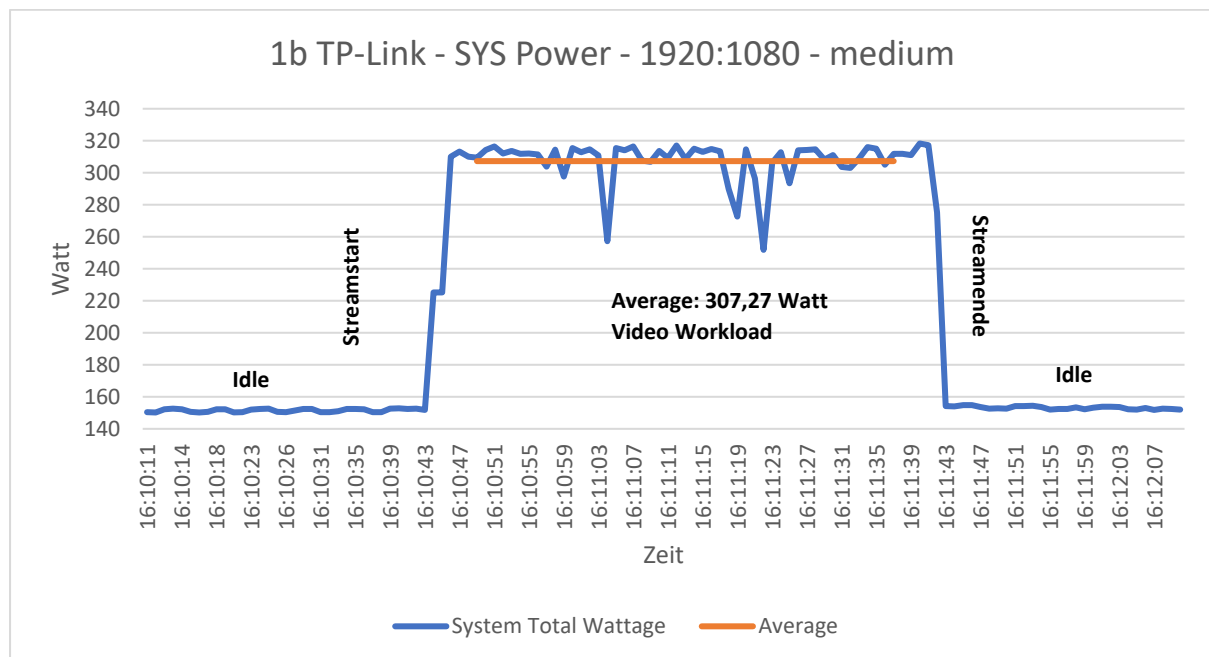
Aus dem Messdiagramm 25 ist erkennbar, dass der Leistungsanstieg im Vergleich zur IDLE Last sehr viel geringer ausfällt als bei unserem Desktop-Server. Im Messdiagramm 25 ist wie schon bei dem Desktop-Server HP Elite Desk 8300 G2 eine CPU Limitierung bei einer auflösung von

1920x1080 bei medium und slower Presets erkennbar! Wie bereits bei dem HP Elite Desk 8300 G2 ist dies auf eine Limitierung der CPU zurückzuführen. Dies kann im schlimmsten Fall dazu führen, dass Frames ausgelassen werden und für den Zuschauer der Stream nicht mehr flüssig dargestellt wird.

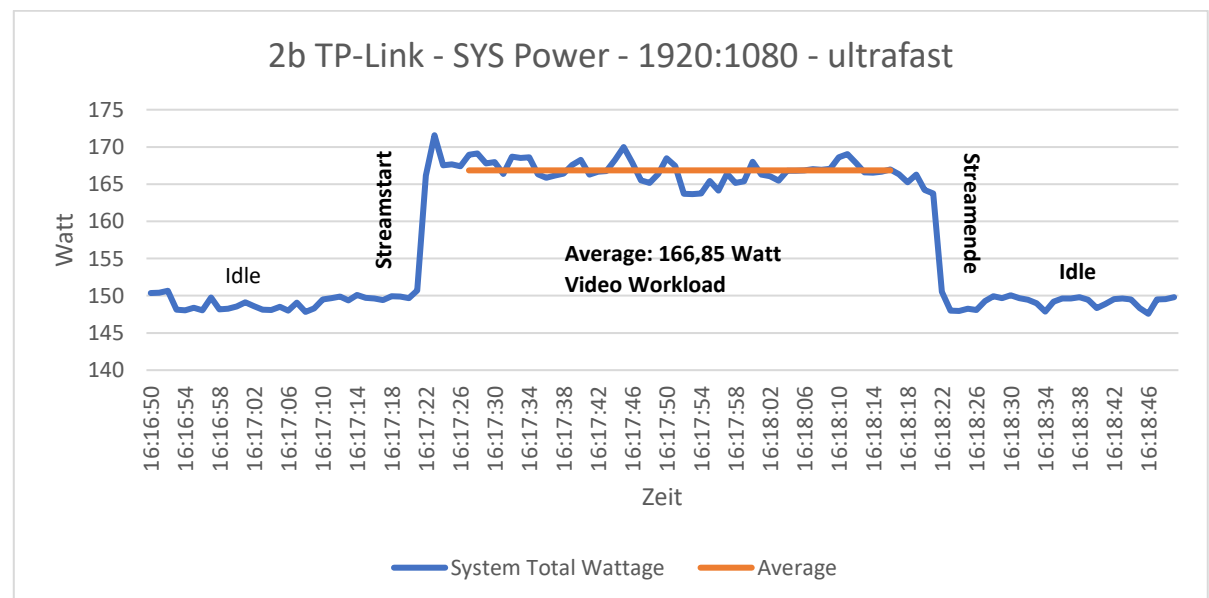
Bei den Messungen verbrauchte die Full-HD Einstellung mit dem Preset „medium“ mit rund 307,27 Watt am meisten Energie beim Streamen. Am wenigsten Energie verbrauchte das Streamen mit der Einstellung 640:360 und dem Preset „ultrafast“ mit nur 159,12 Watt. In dieser Einstellung benötigt der Server beim Streamen nur rund 9,12 Watt mehr Energie als er im Idlezustand benötigt.

Aus Messdiagramm 29 ist deutlich erkennbar, dass eine Reduktion der Auflösung von 1920x1080 auf 1280x720 mit der Preset Einstellung bereits eine deutliche Leistungsreduktion von 30% mit sich bringt! Eine weitere Reduktion der Auflösung auf 640x360 bringt eine weitere Leistungsreduktion von 23% mit sich.

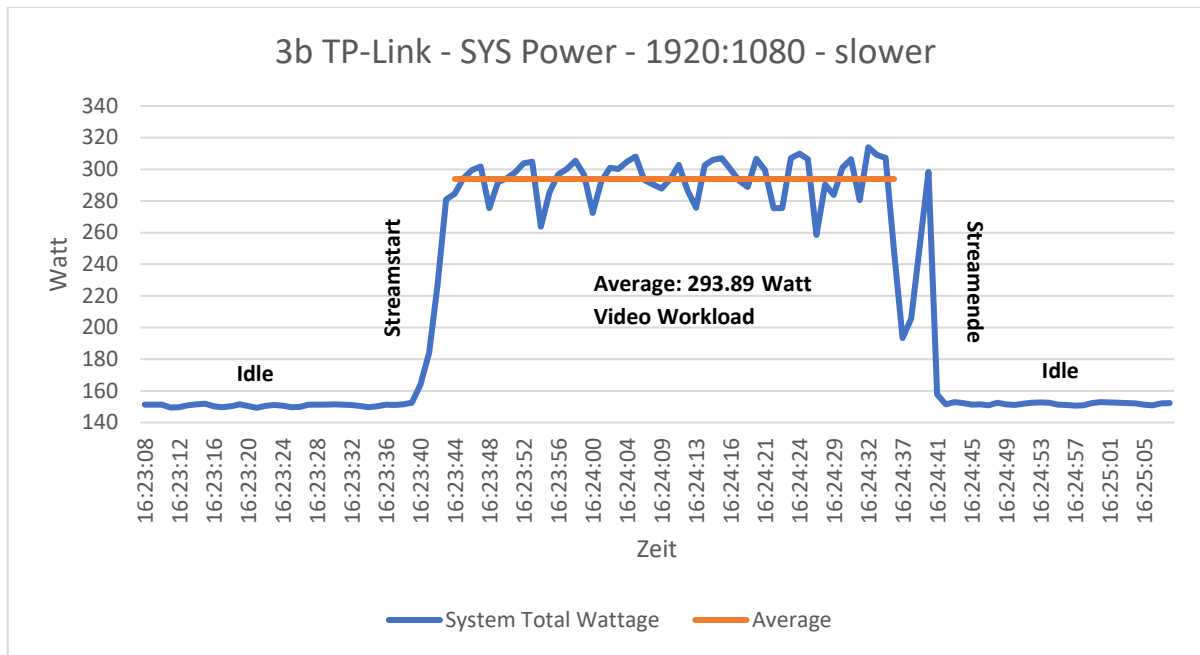
11.4 Messungen Diagramme



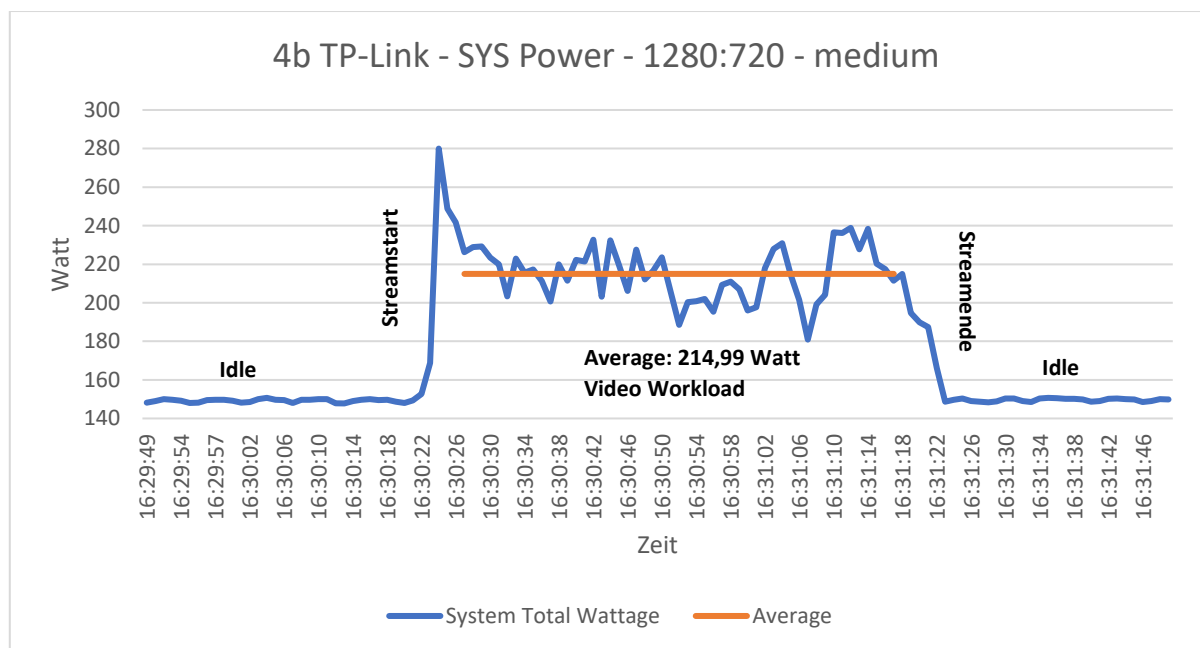
Messdiagramm 30



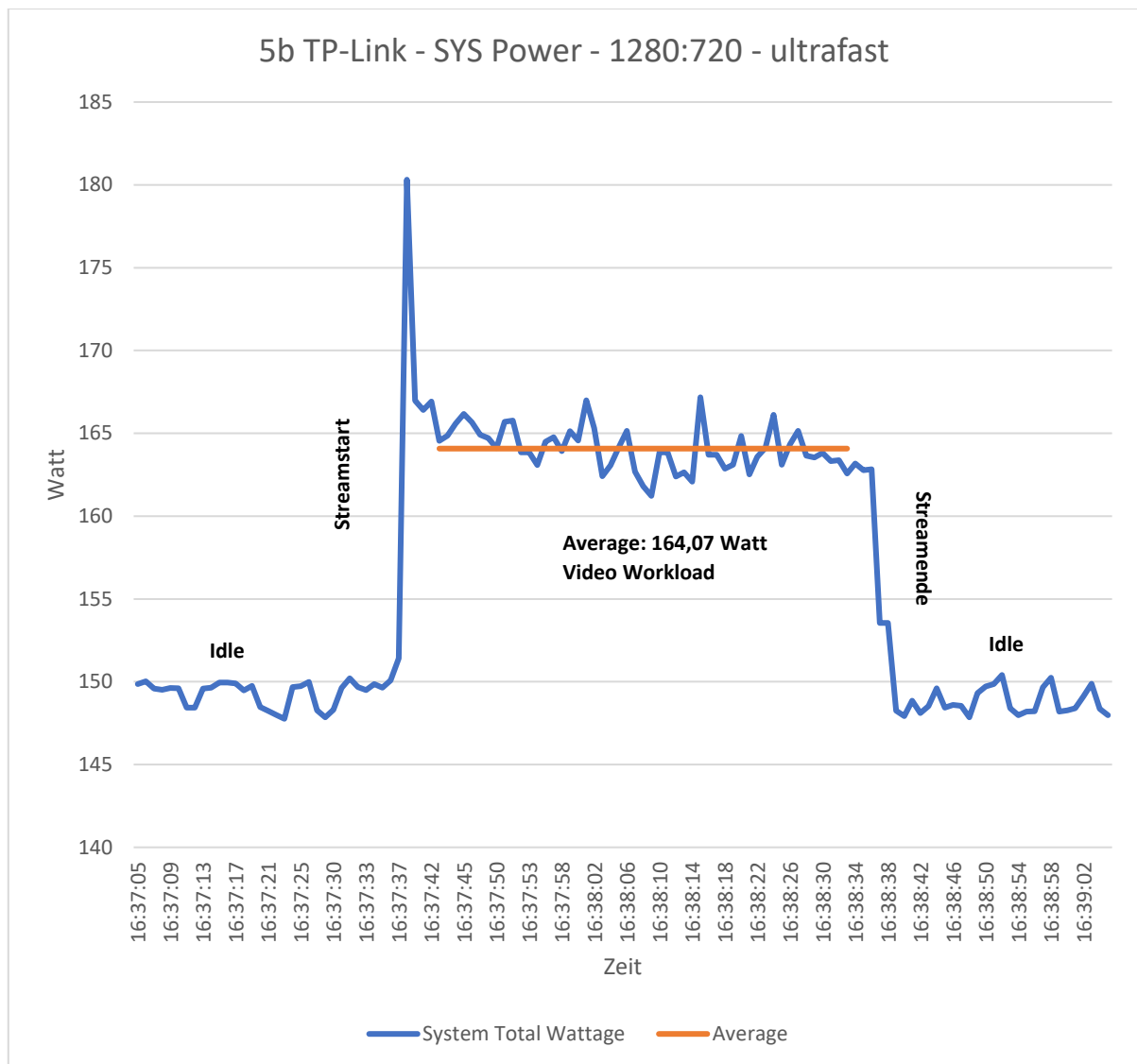
Messdiagramm 31



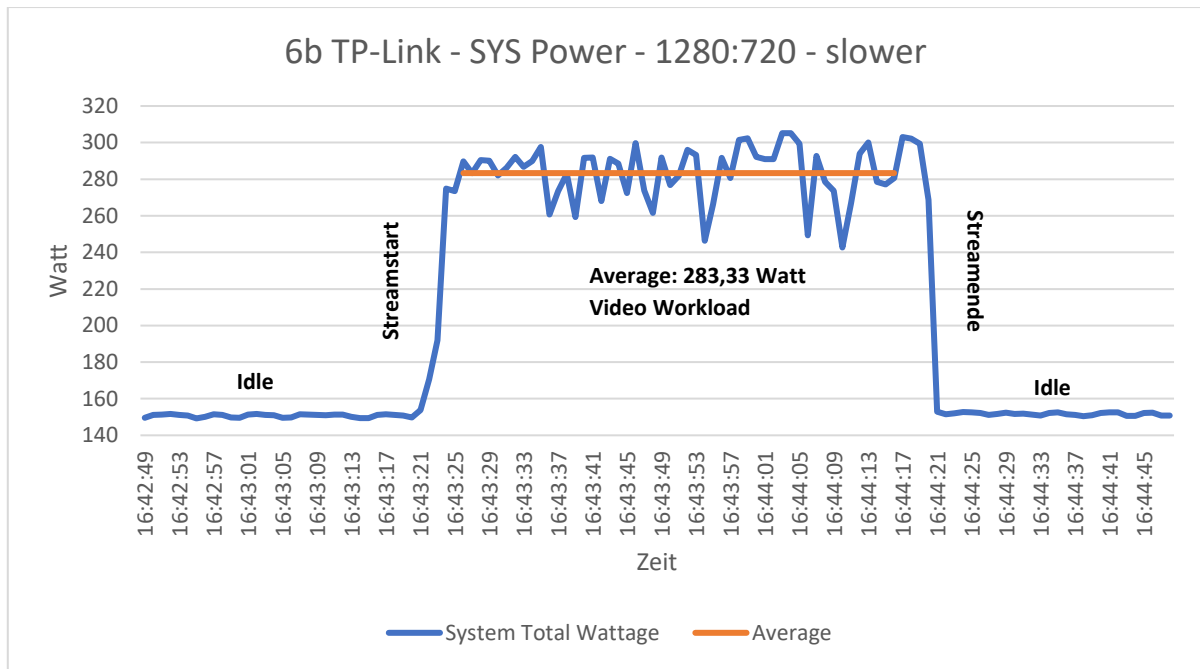
Messdiagramm 32



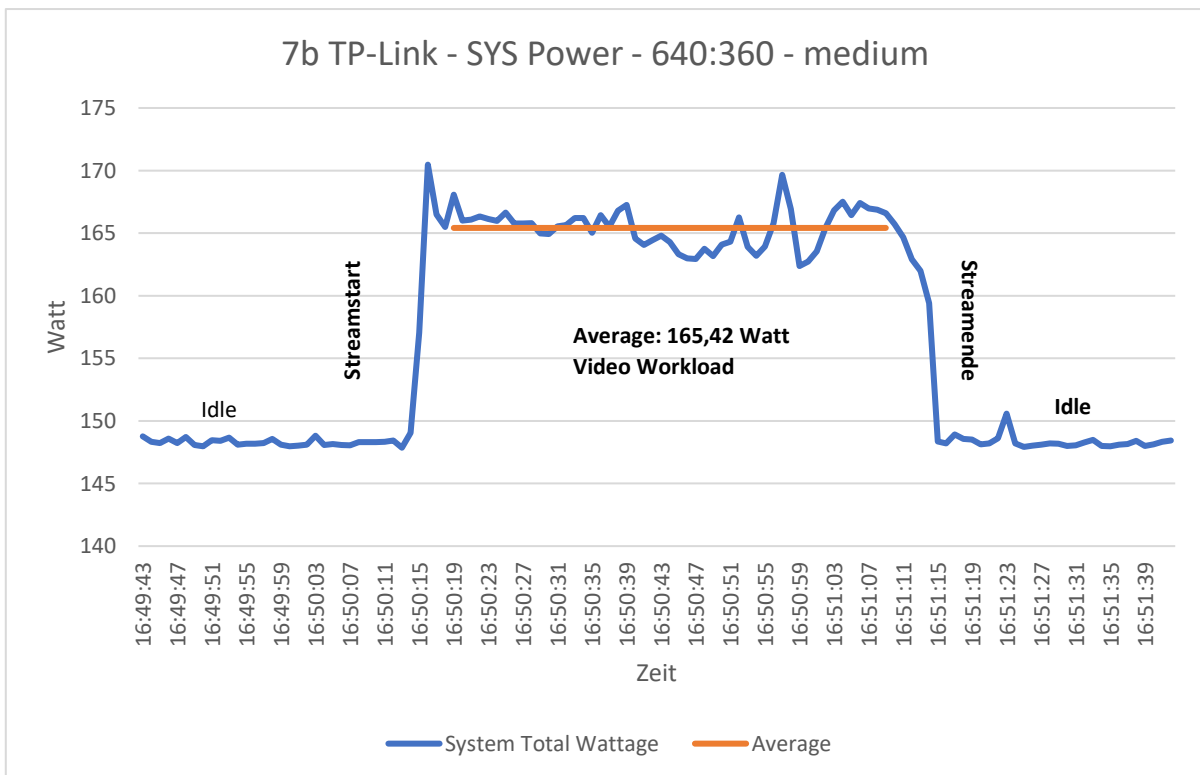
Messdiagramm 33



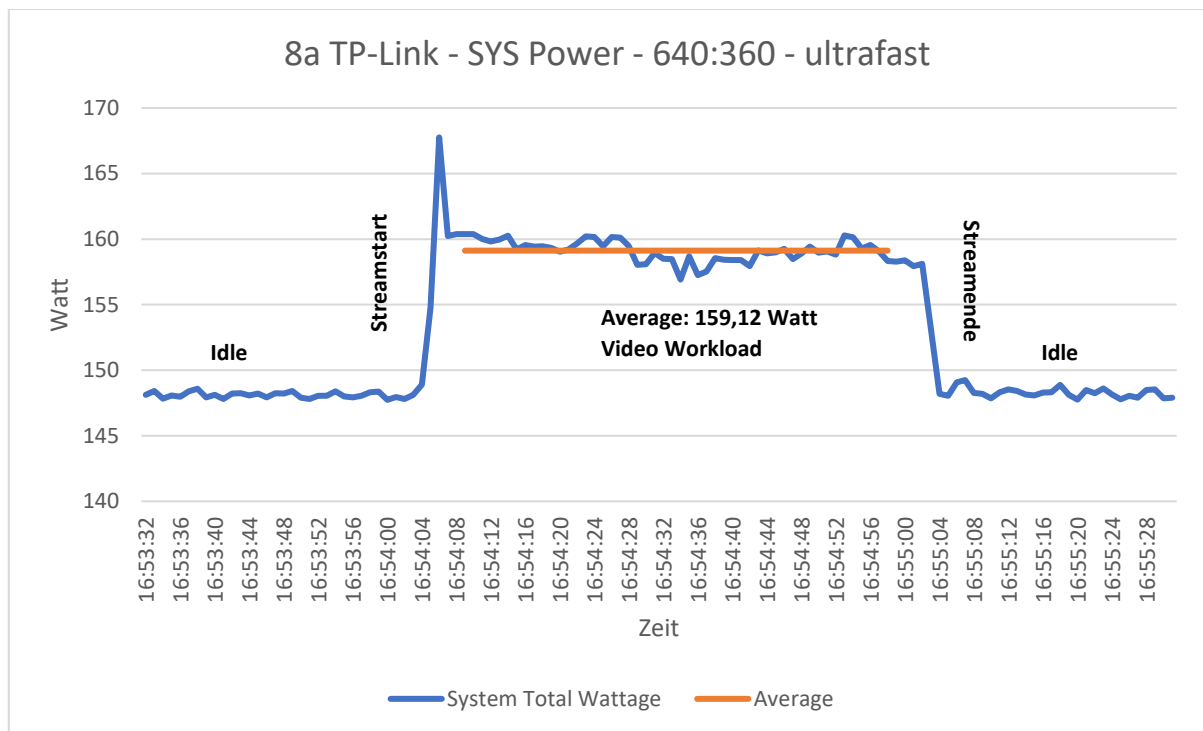
Messdiagramm 34



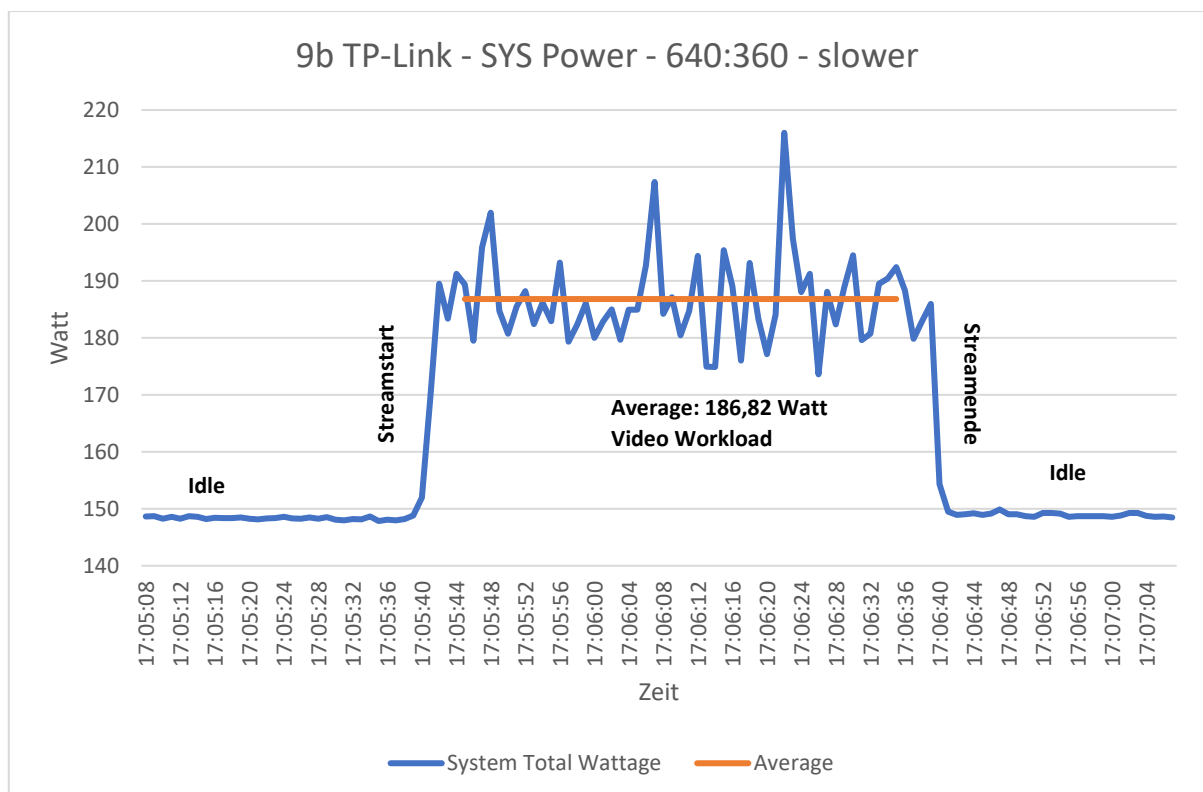
Messdiagramm 35



Messdiagramm 36



Messdiagramm 37



Messdiagramm 38

12 Messszenario 2 – Belastungstest

Bei Szenario zwei wurden wie schon bei Szenario eins mit Hilfe des ffmpeg Befehls die Encoder Einstellungen in der NGINX Konfiguration definiert. Zum Streamen wurde wiederum OBS Studio und zum Messen die beiden Tools Intel Power Gadget und TP-Link verwendet. Außerdem nutzen wir JMeter um die Multiuserzugriffe zu generieren. Dabei wurden 50, 100 und 500 Userzugriffe auf den HP Elite Desk 8300 G2 und den HP Server ProLiant simuliert. Die Messdauer musste auch angepasst werden, da zusätzlich zur Idlezeit und zur Streamlaufzeit auch die Benutzerzugriffe simuliert wurden. Somit wurde die Messungszeit von zwei Minuten bei Szenario eins auf rund zwei Minuten und vierzig Sekunden angepasst.

Idlezeit vor Streambeginn	15 Sekunden
Streamen bis zur Benutzersimulation	45 Sekunden
Benutzerzugriff	30 Sekunden
Streamen nach Benutzerzugriff	45 Sekunden
Idlezeit nach Stream	25 Sekunden

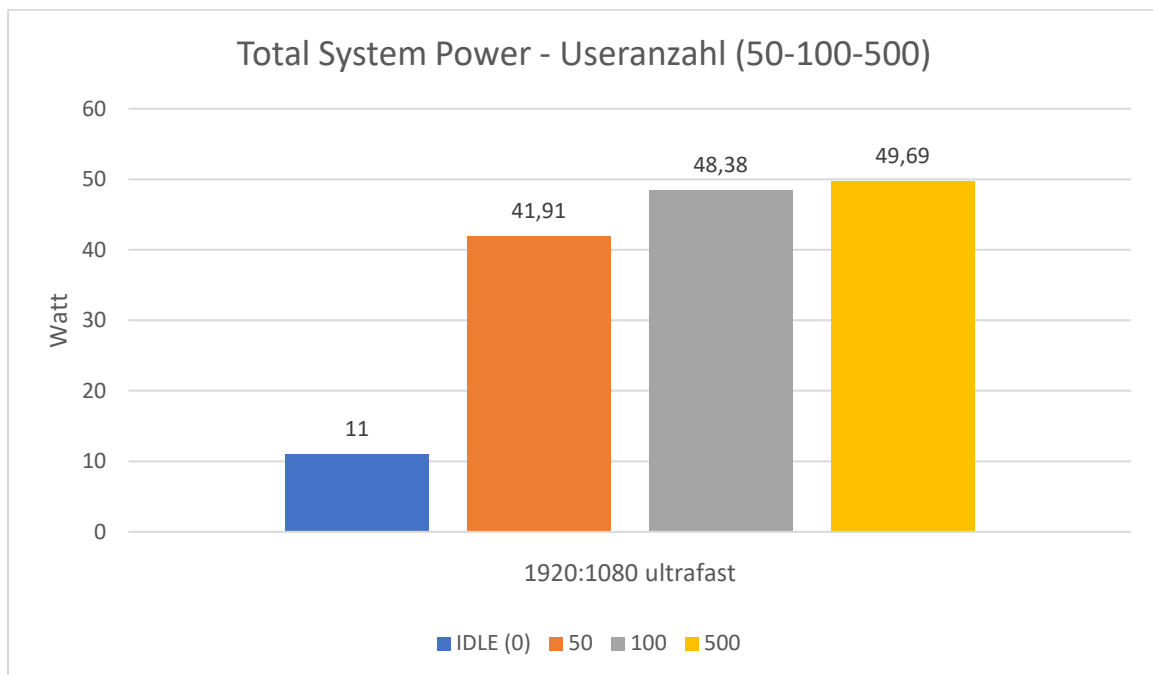
12.1 Streamingserver HP Elite Desk 8300 G2

Die Tabelle JMeter - HP Elite Desk 8300 G2 zeigt die durchgeführten Messungen am Desktop PC G2. Bei den Messungen wurden nur die Auflösung 1920:1080 mit einer Bitrate von 6000 Kilobits pro Sekunde und der Preseteinstellung „ultrafast“ betrachtet.

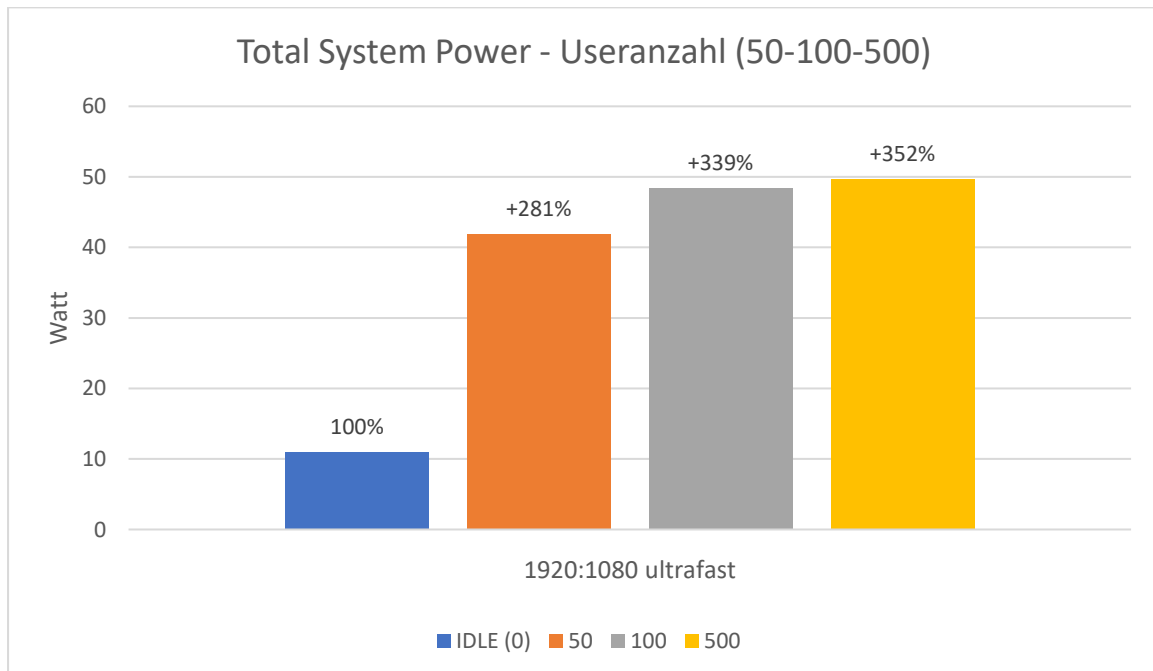
JMeter Belastungstest – HP Elite Desk 8300 G2								
Auflösung	Bitrate	Encoding	FPS	Einstell.	CRF	SYS POWER average [W]	CPU POWER average [W]	User
1920:1080	6000	CBR	60	ultrafast	23	41.91	22.60	50
1920:1080	6000	CBR	60	ultrafast	23	48.38	27.24	100
1920:1080	6000	CBR	60	ultrafast	23	49.69	27.31	500

Tabelle 8: Belastungstest - HP Elite Desk 8300 G2

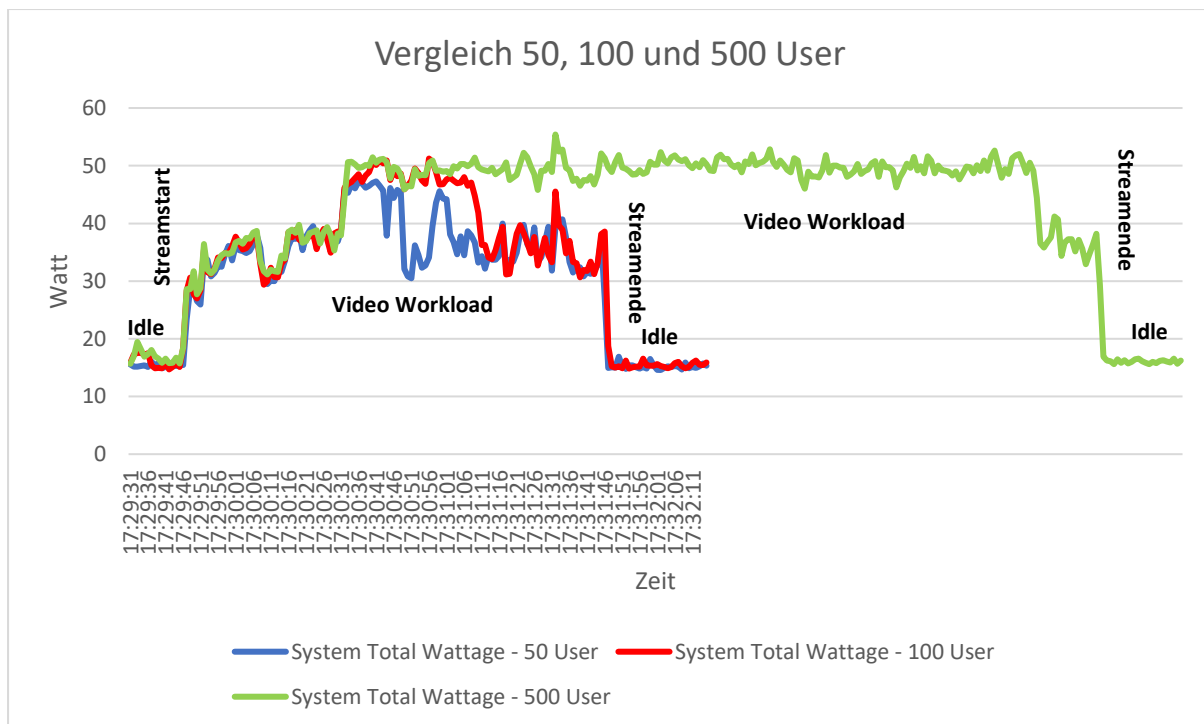
12.1.1 Resultate und Analyse



Messdiagramm 39



Messdiagramm 40



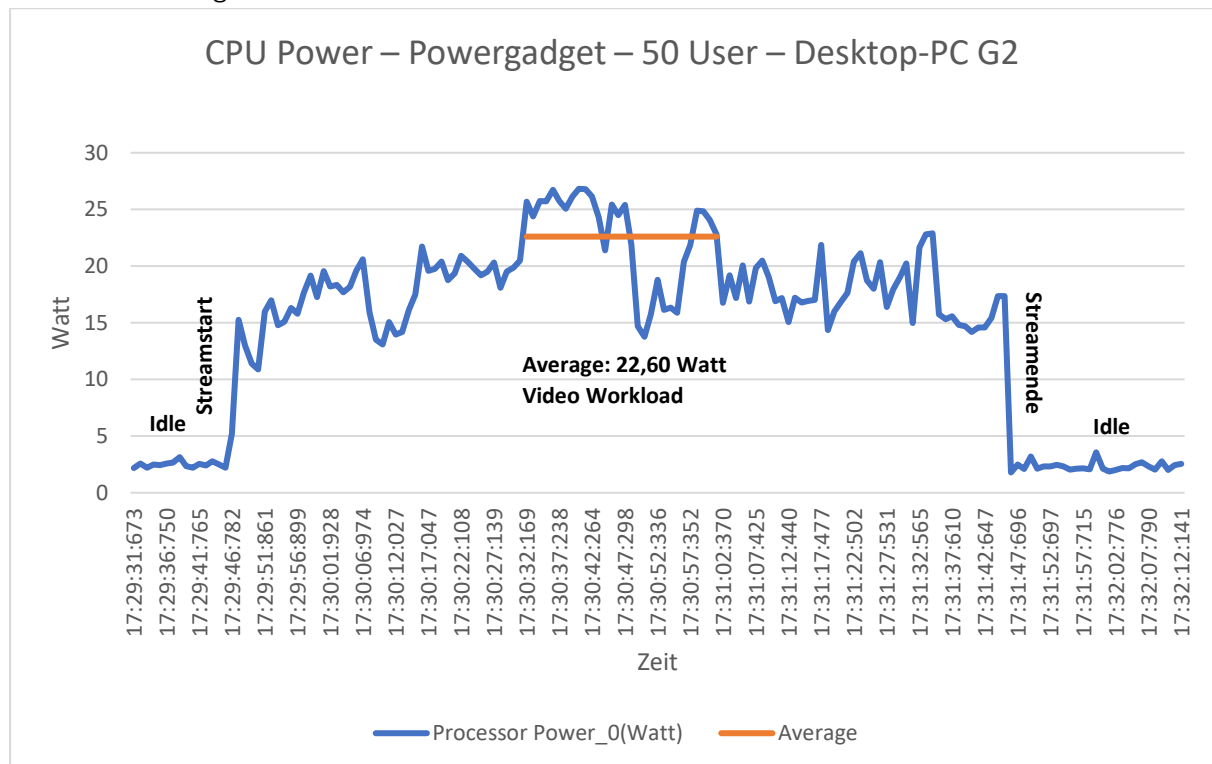
Messdiagramm 41

Wie im Messdiagramm 41 ersichtlich, sind Idelzeit und Streamingzeit bei allen drei Linien sehr ähnlich und es ergeben sich Abweichungen nur während der Benutzerzugriffe.

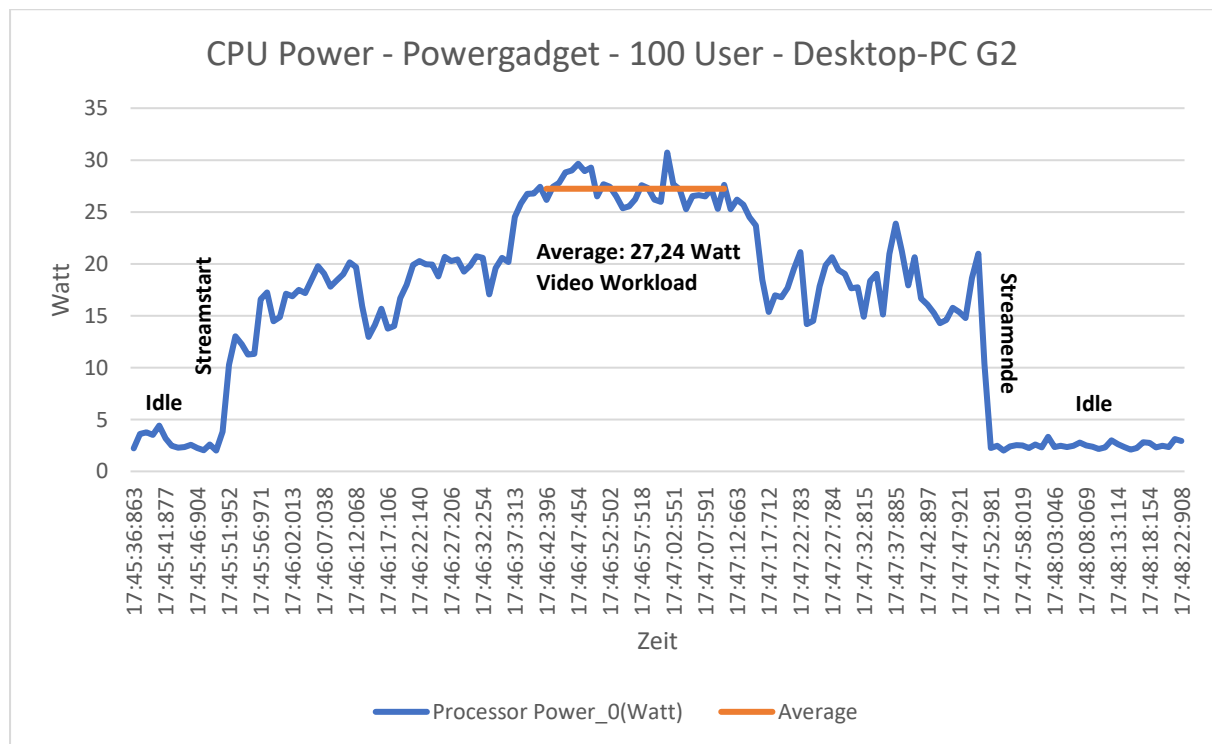
Das Ergebnis der Messung zeigt, dass der größte Unterschied zwischen 50 und 100 Usern erkennbar ist. Bei 50 Benutzern verbrauchte das Desktop-Server System rund 41,91 Watt Gesamtenergie und bei 100 Benutzerzugriffen gerademal 48,38 Watt an Gesamtenergie. Bei den 500 Usern stieg der Energiebedarf nicht signifikant an. Dies ist auf eine Limitierung des Gigabitanschlusses bzw. der Gigabit fähigen LAN Verbindung zurückzuführen. Durch diese Limitierung ist auch klar erkennbar, dass unser Stresstest wesentlich länger dauerte. Anstelle von 45 Sekunden dauerte die Messung nun ganze drei Minuten und zehn Sekunden. Deshalb musste die gesamte Simulationszeit von zwei Minuten und vierzig Sekunden auf fünf Minuten angepasst werden.

Im Vergleich zur Auslastung im IDLE Betrieb ist ein signifikanter Anstieg der Leistung von 281% bei 50 Usern erkennbar!

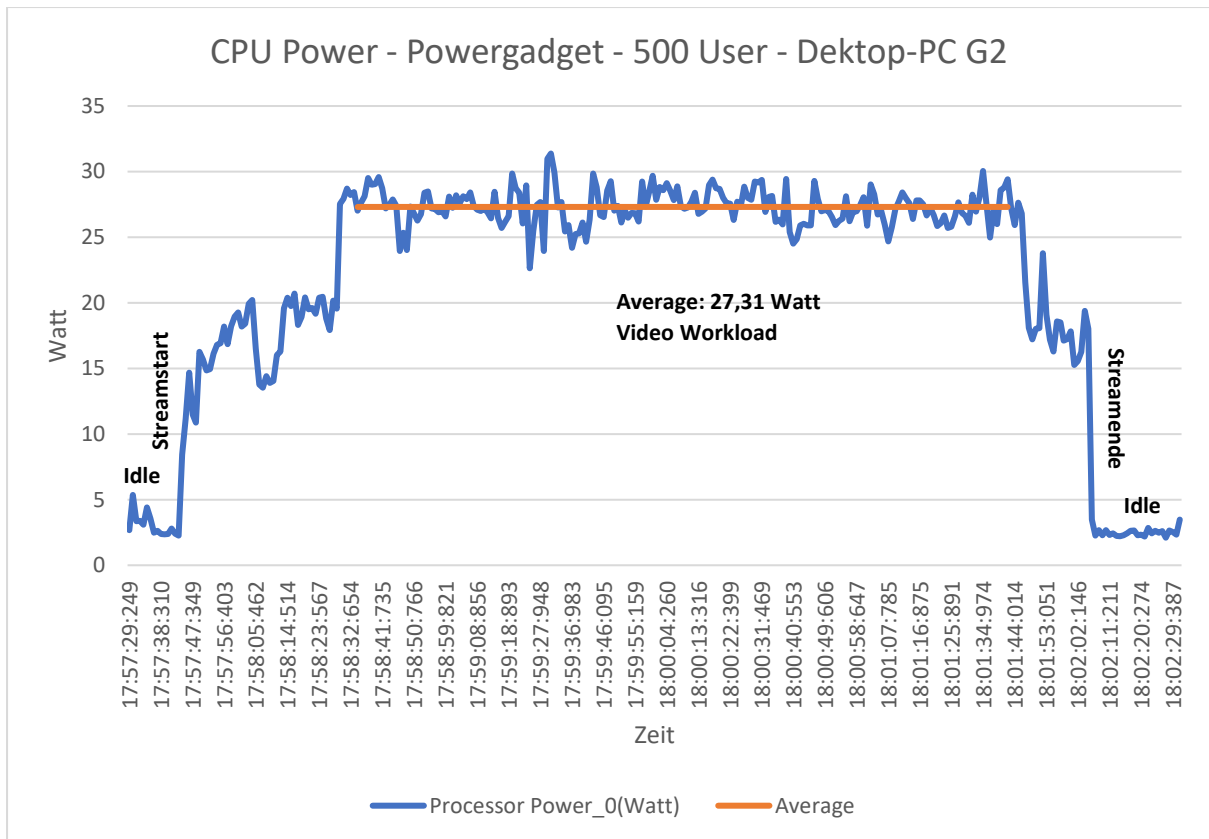
12.1.2 Messdiagramme



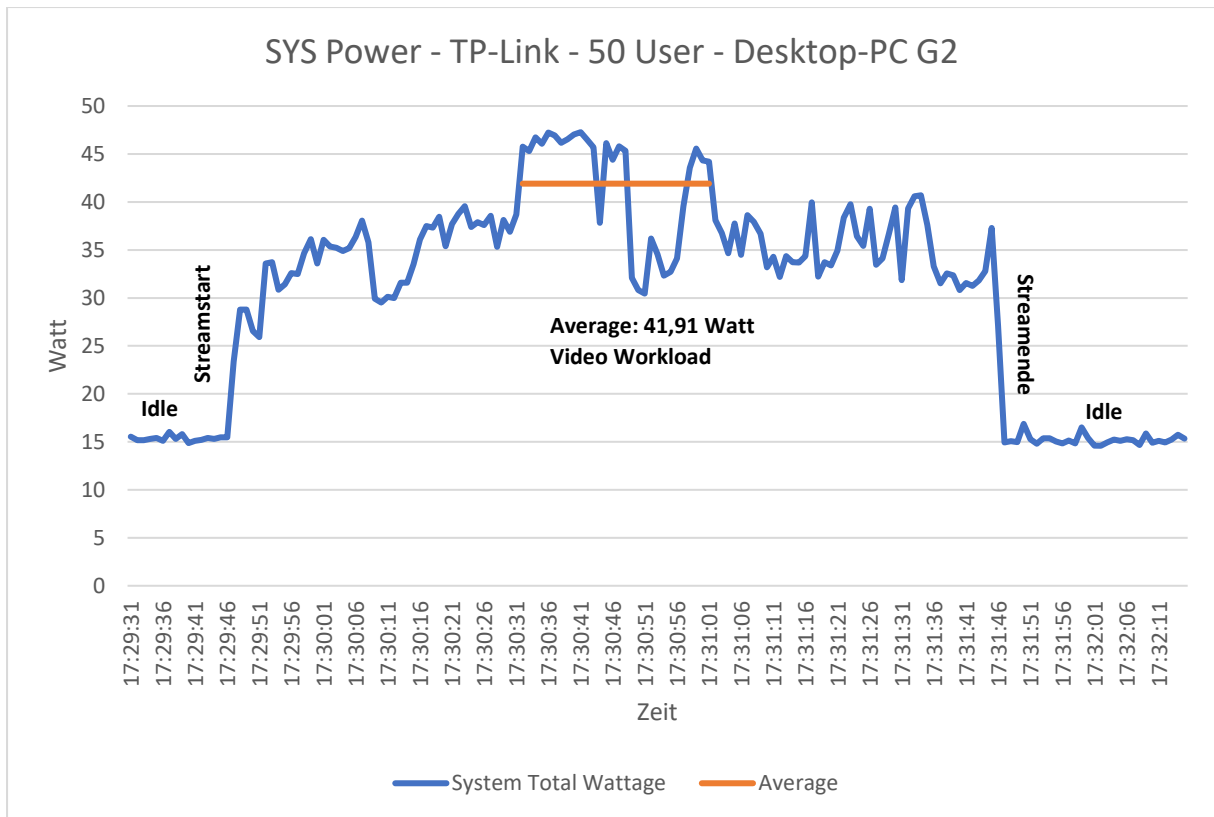
Messdiagramm 42



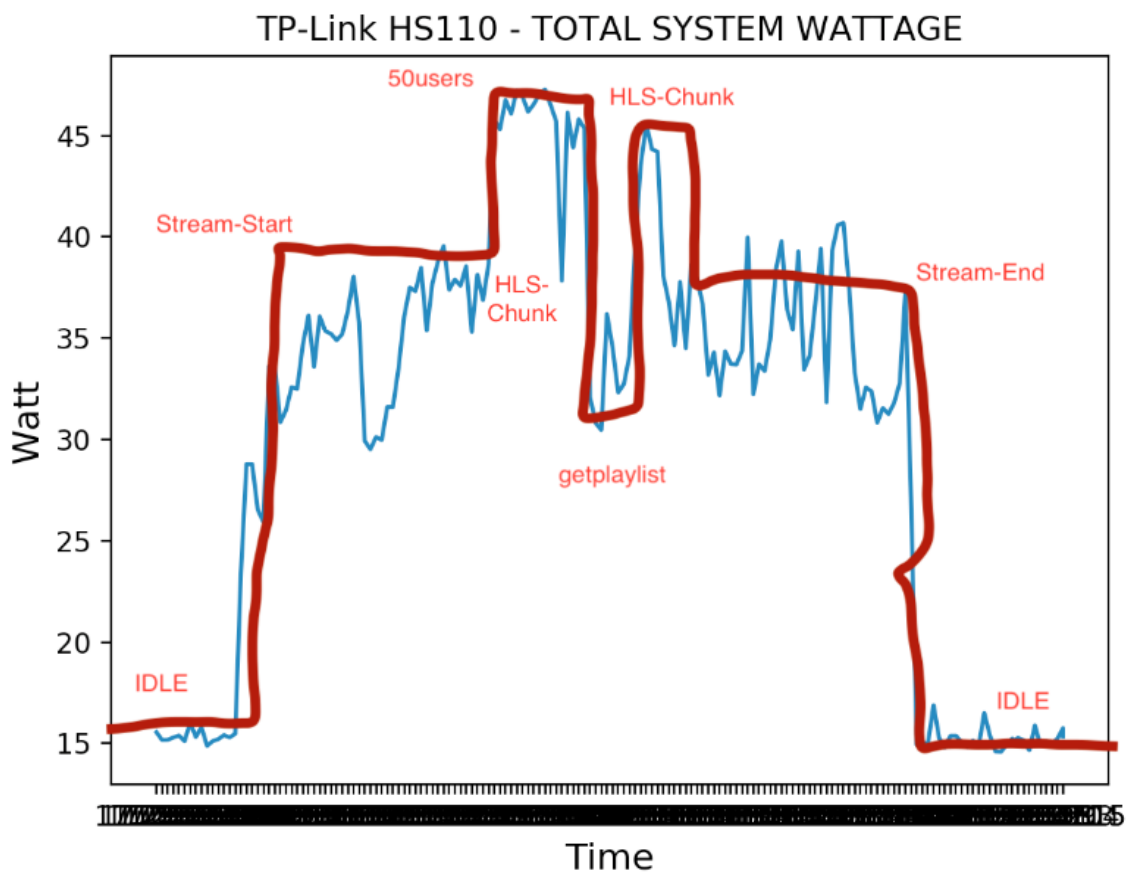
Messdiagramm 43



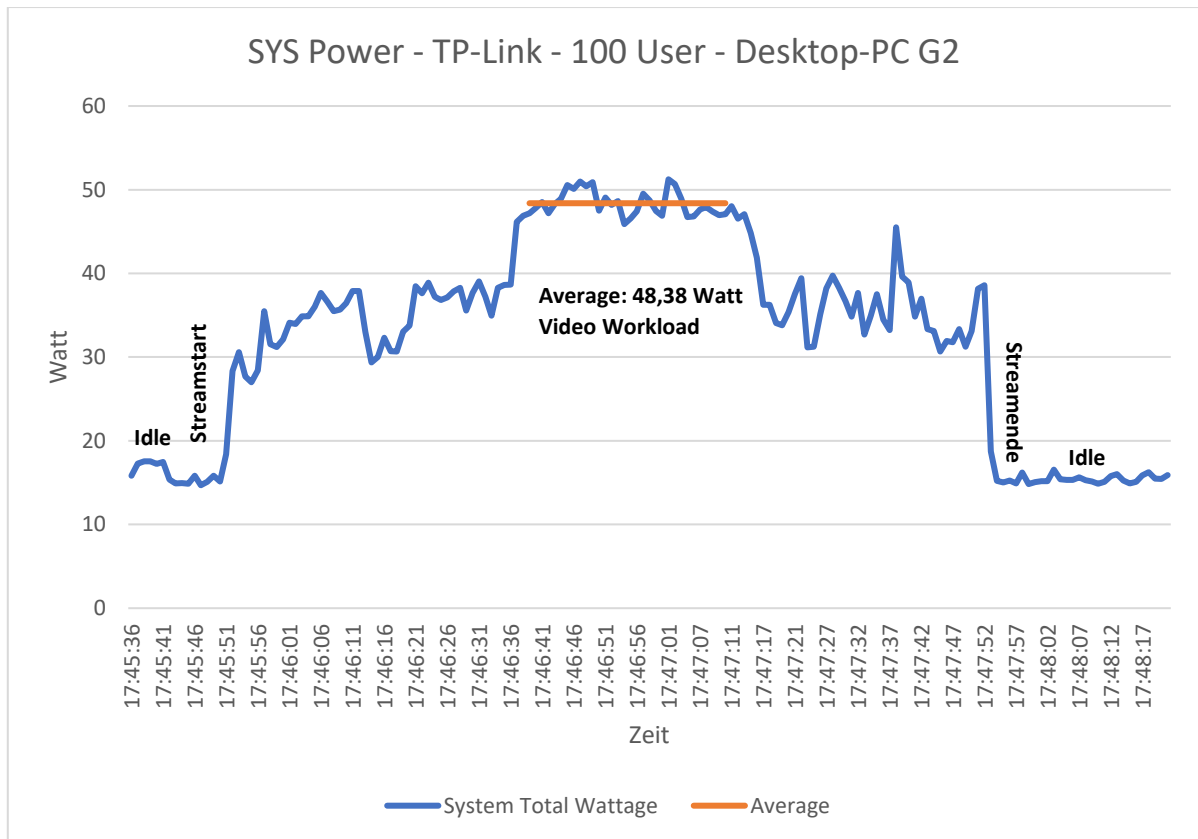
Messdiagramm 44



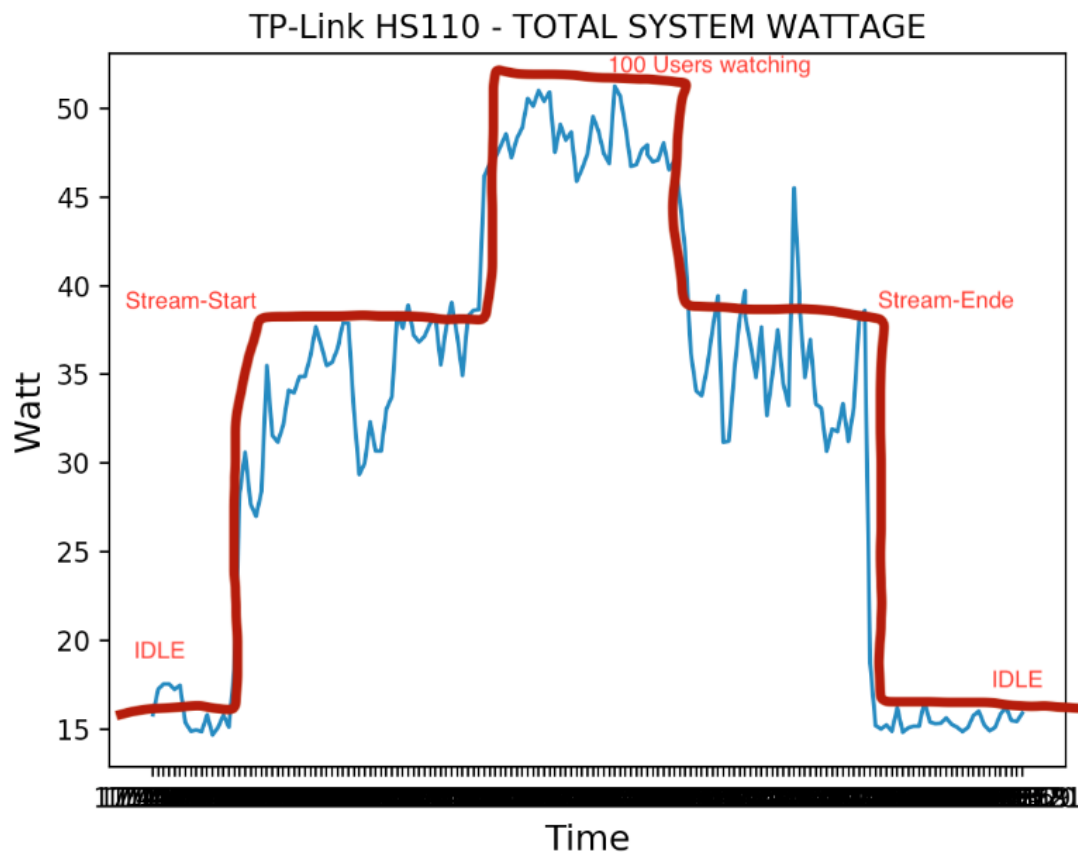
Messdiagramm 45



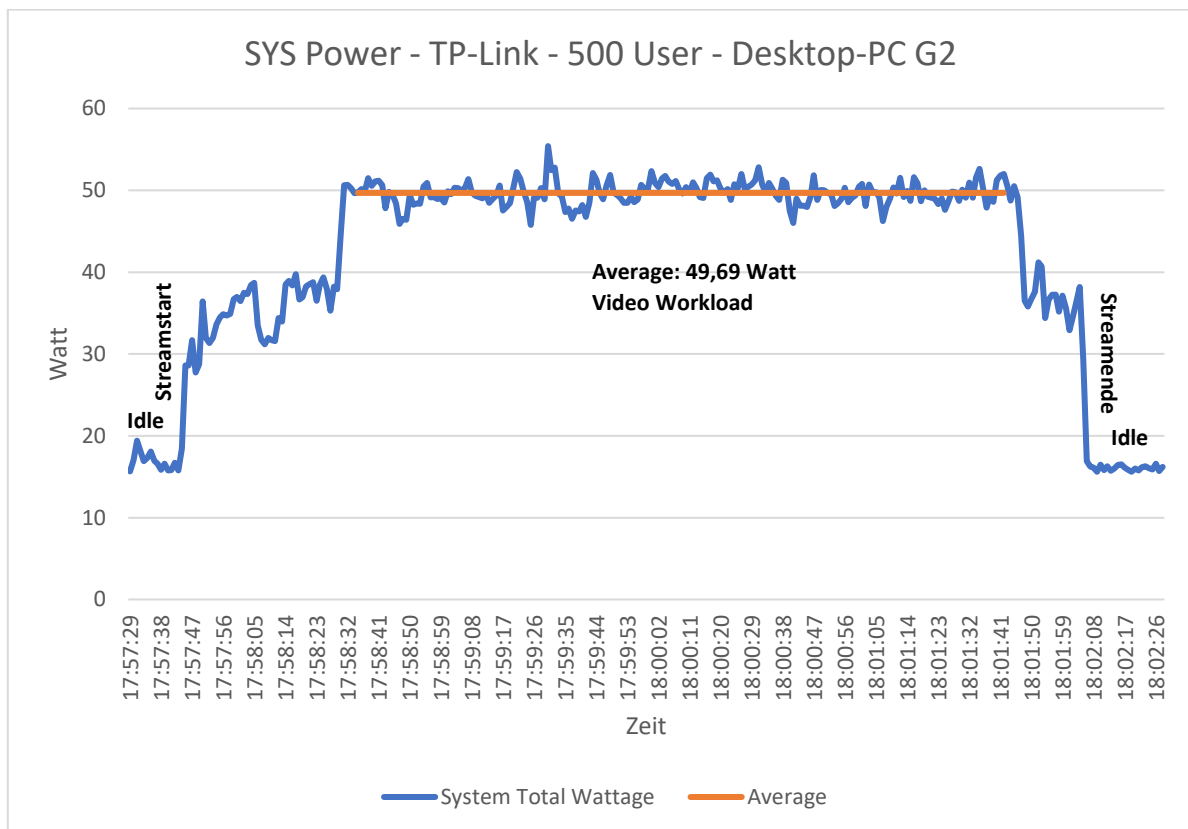
Messdiagramm 46



Messdiagramm 47



Messdiagramm 48

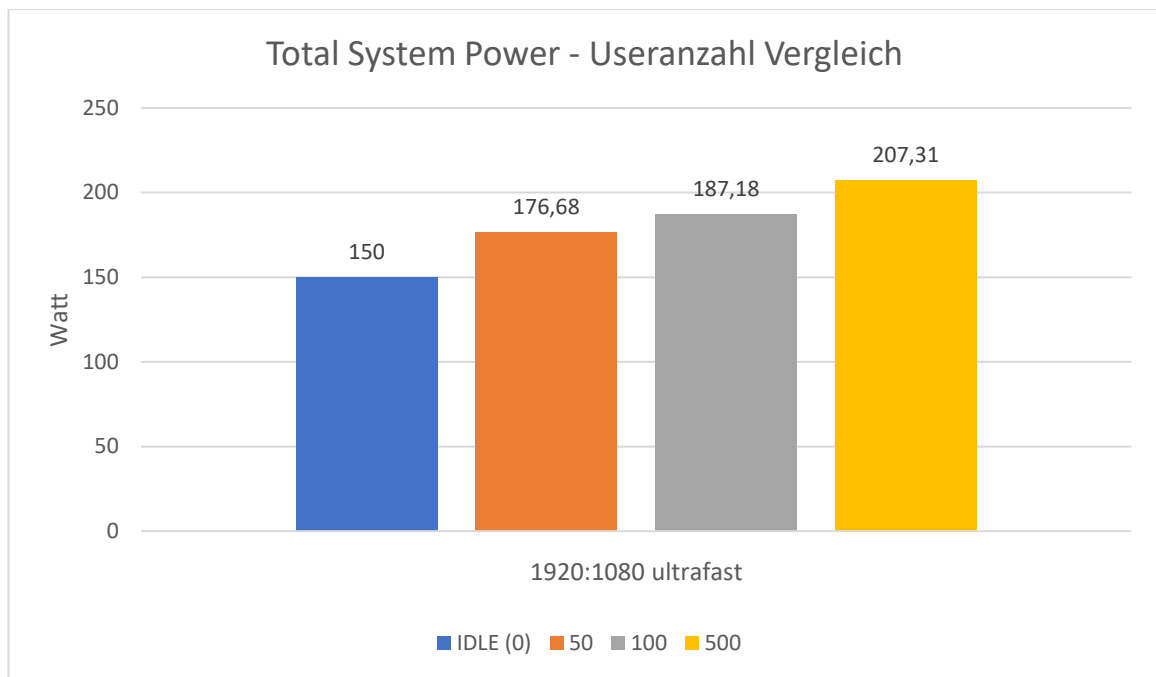


Messdiagramm 49

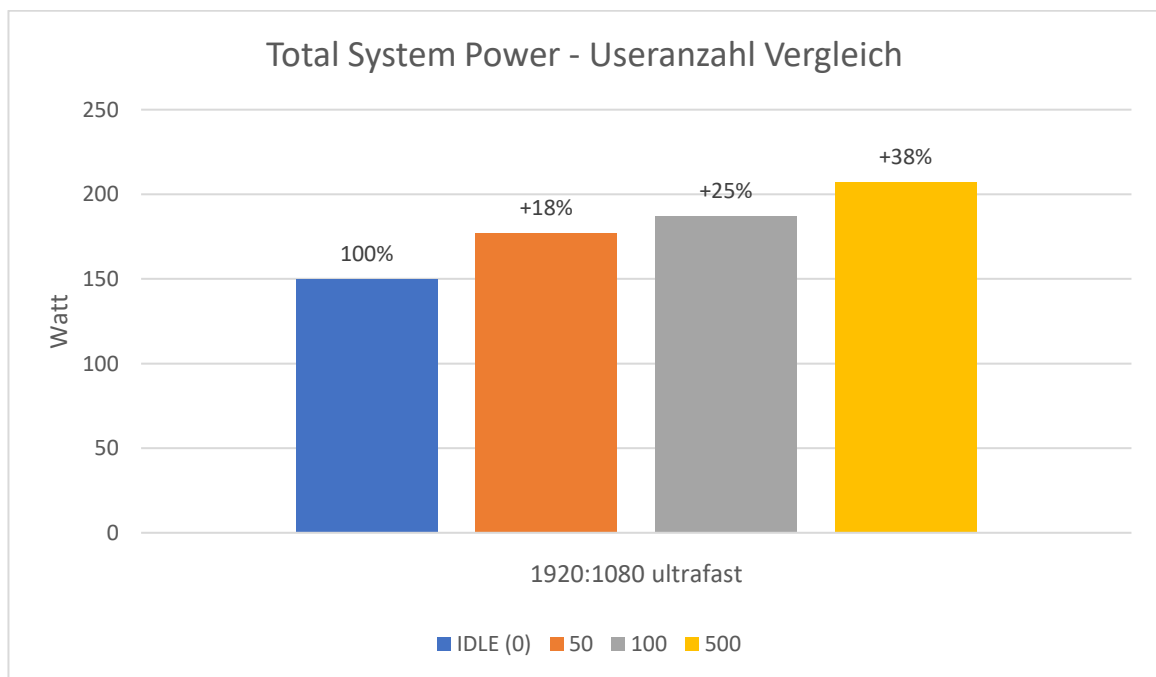
Die Tabelle JMeter – HP Server ProLiant G6 zeigt die durchgeführten Messungen am HP Server. Bei den Messungen wurden wie beim Desktop PC die Auflösung 1920:1080 mit einer Bitrate von 6000 Kilobits pro Sekunde und der Preseteinstellung „ultrafast“ betrachtet. Wie auch beim vorherigen Beispiel wurden 50, 100 und 500 Benutzerzugriffe simuliert. Da für die Übertragung die Gigabitleitung nicht ausreichte, verlängerte sich auch hier die Übertragungszeit für die Simulation von 500 Benutzerzugriffen.

JMeter – HP Server ProLiant G6							
Auflösung	Bitrate	Encoding	FPS	Preset	CRF	SYS-POWER average [W]	User
1920:1080	6000	CBR	60	ultrafast	23	176.68	50
1920:1080	6000	CBR	60	ultrafast	23	187.18	100
1920:1080	6000	CBR	60	ultrafast	23	207.31	500

Tabelle 9: JMeter - HP Server ProLiant G6



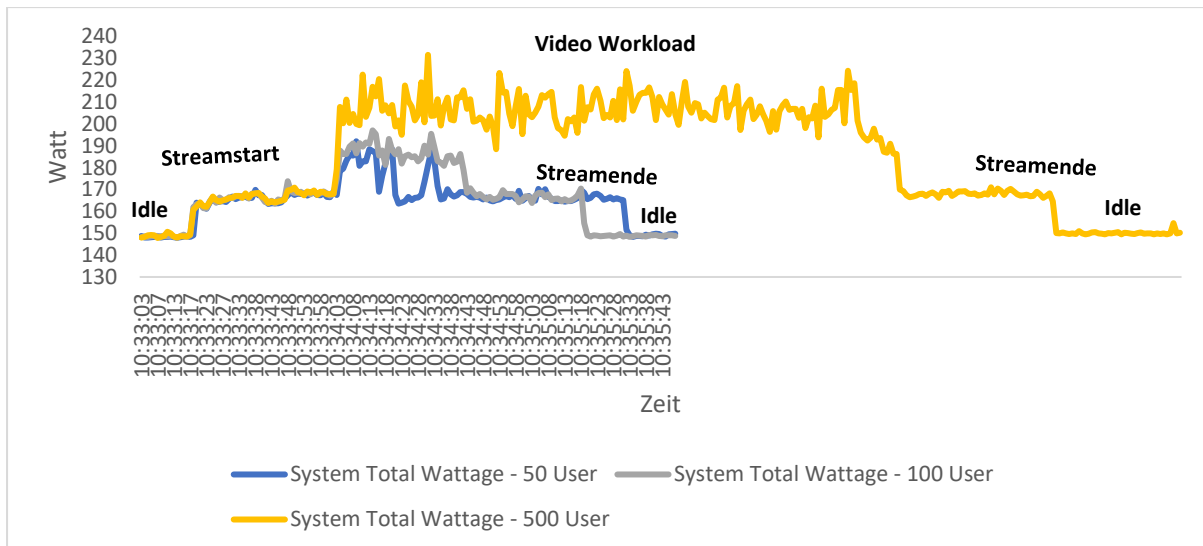
Messdiagramm 50



Messdiagramm 51

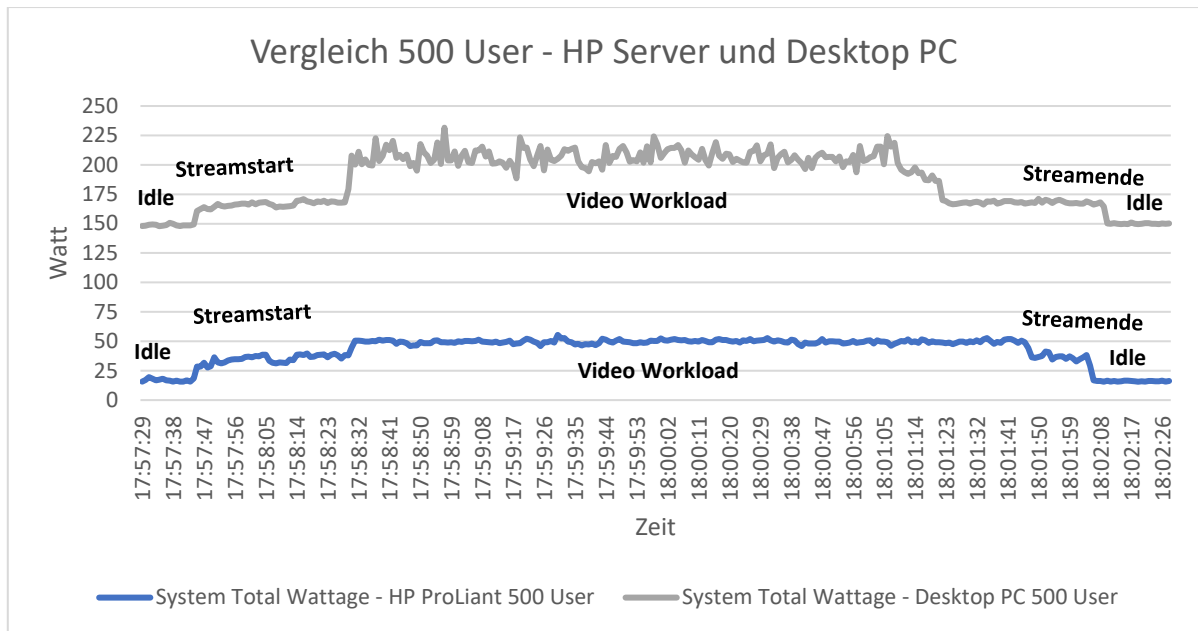
Die Messdiagramme 51 und 50 zeigen sehr schön den Anstieg der Leistung bezüglich der Zuseher. Es ist klar erkennbar, dass dieser Anstieg wesentlich höher ausfällt als bei dem Desktop System. Dies ist darauf zurückzuführen, dass in diesem System zwei CPU's verbaut sind und diese einer wesentlich älteren Generation entsprechen.

Das Ergebnis der Messungen aus dem Messdiagramm 52 veranschaulicht, dass wie schon zuvor beim Desktop PC der Energieverbrauch bei der Simulation von 500 Usern am höchsten ist. Der Energieverbrauch bei 500 Benutzern beträgt rund 207,31 Watt und bei 50 Benutzern nur 176,68 Watt. Das entspricht einer Differenz von 30,63 Watt.



Messdiagramm 52

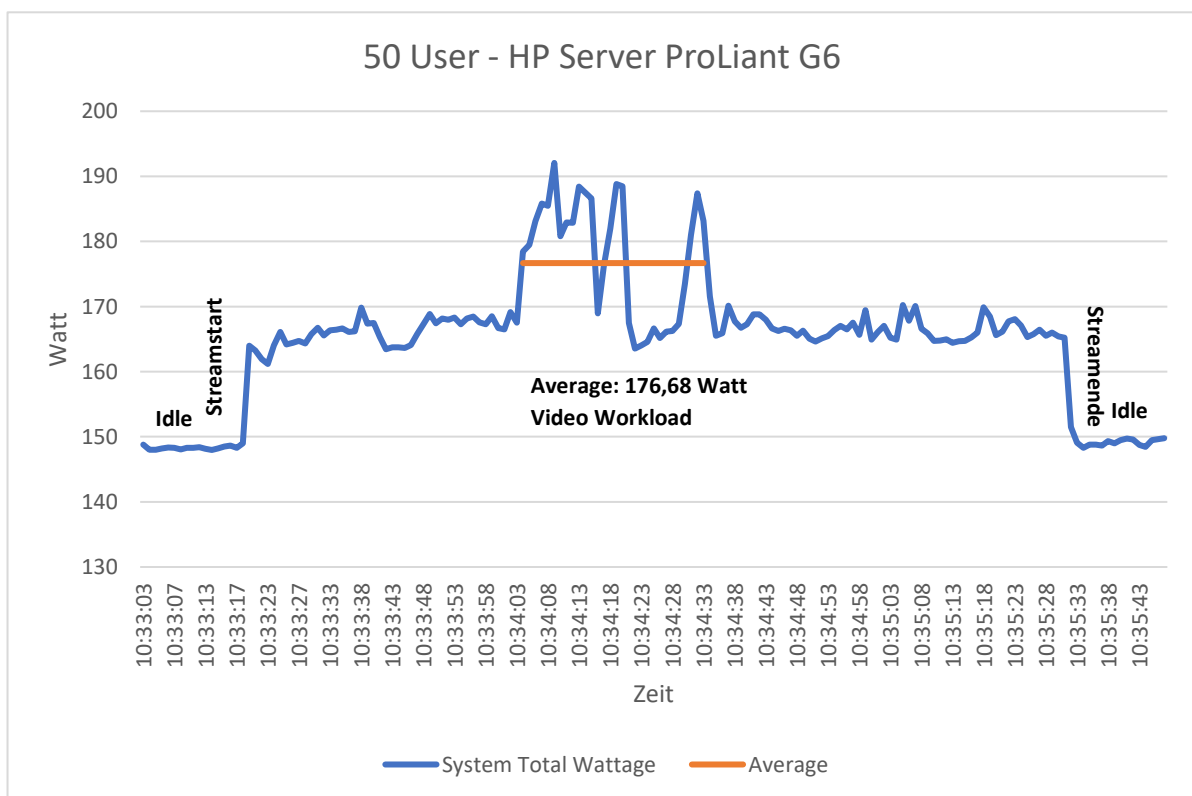
Messdiagramm 52 zeigt schön, dass der Kurvenverlauf vor dem Beginn der Benutzerzugriffe nahezu identisch ist, was auch so sein sollte. Auch nach dem Beenden der Benutzerzugriffe sind sich die Linien sehr ähnlich und liegen zwischen 165 und 170 Watt. Auch die Idlezeit ist bei allen drei Linien identisch.



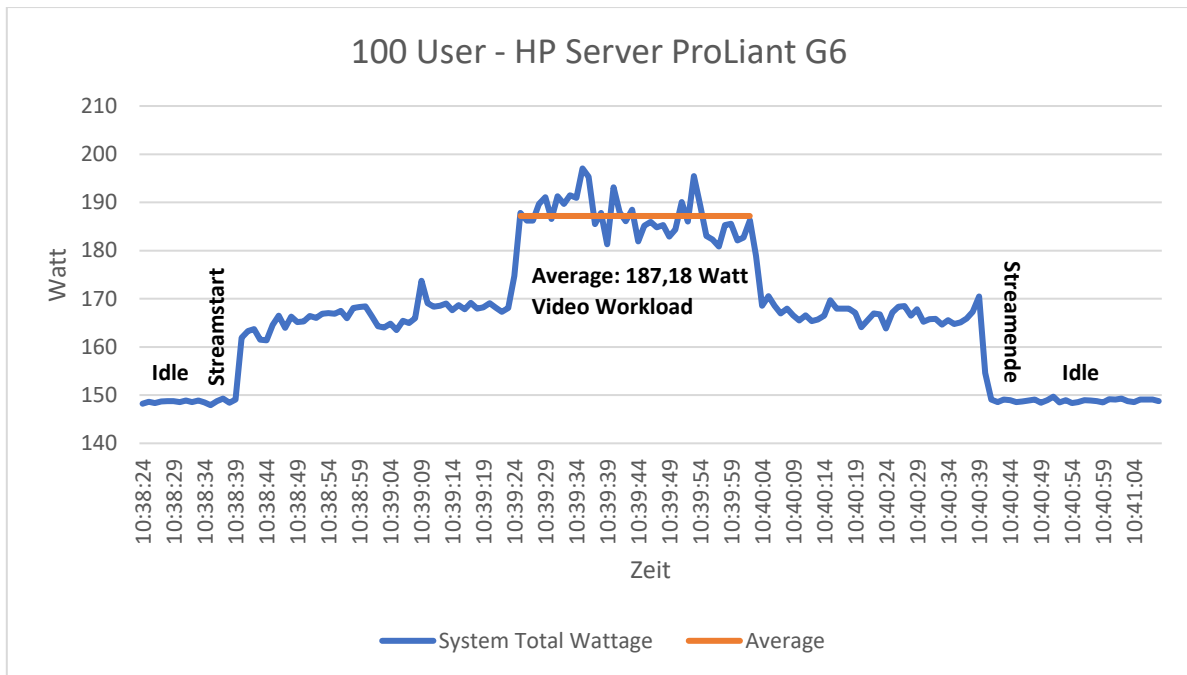
Messdiagramm 53

Vergleicht man das Ergebnis von 500 Usern des HP Servers mit dem Desktop PC, benötigt dieser gerademal 49,69 Watt. Dabei ergibt sich eine Differenz von 157,82 Watt, was ein bisschen mehr als die Idle Leistung des HP Servers ist.

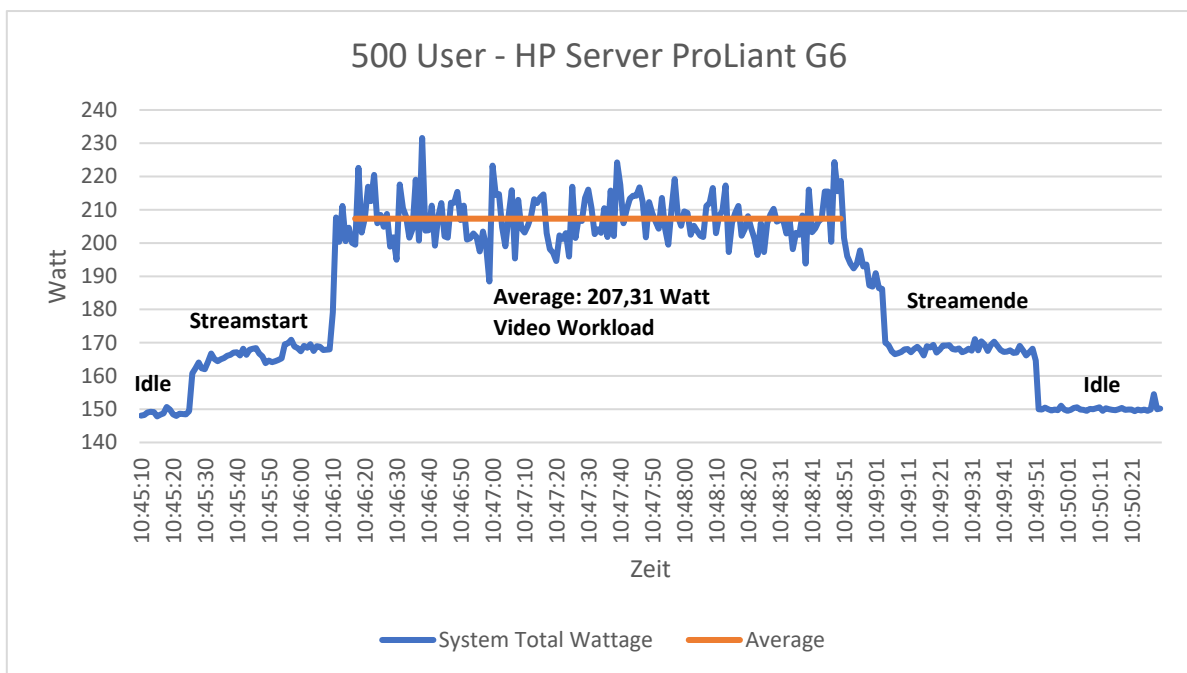
Messdiagramm 54 gibt einen Überblick der einzelnen Messungen.



Messdiagramm 54



Messdiagramm 55



Messdiagramm 56

13 Probleme & Lessons Learned

13.1 OS Wahl

Durch dieses Projekt haben wir gelernt, dass es sehr wichtig ist sich bereits im Vorhinein für ein geeignetes Betriebssystem zu entscheiden, da dies einen maßgeblichen Einfluss auf die Umsetzung hat. So hat das Betriebssystem CENTOS, welches auch durchaus in der Praxis eingesetzt wird, aufgrund seiner strikten Security Policies sehr große Probleme bereitet und wir sahen uns gezwungen auf Ubuntu Server umzusteigen. Diese Erkenntnis und der darauffolgende Wechsel des Betriebssystems erforderte einen zusätzlichen Zeitaufwand, welcher durch Planung und Recherche vermeidbar gewesen wäre.

13.2 NGINX Konfiguration

Die NGINX Konfiguration stellte uns vor Problemen, da diese auf CentOS als auch auf dem Ubuntu Server nicht sofort so funktioniert wie wir es geplant hatten. Nach ausgiebiger Recherche und Einarbeitung in die Materie gelang es uns jedoch einen streaming fähigen NGINX Server zu konfigurieren.

13.3 JMETER

Das eingesetzte Software Tool JMETER welches zur Lastgenerierung eingesetzt wird, stellte sich als sehr umständlich heraus und benötigte sehr viel Einarbeitungszeit. Es war nicht sofort klar wie das Tool intern arbeitet und somit war es zu Beginn nur sehr schwer möglich nachvollziehbare Ergebnisse zu generieren. Dies stellte uns vor eine große Herausforderung, welche jedoch nach umfangreicher Recherche und zahlreichen Selbstversuchen bewältigt werden konnte.

13.4 Zeitmanagement

Zeitmanagement war einer der wichtigsten Faktoren in diesem Projekt da sich das Projektteam neben der Projektarbeit auch mit zahlreichen anderen Lehrveranstaltungen beschäftigen musste. Im Großen und Ganzen wurde das Zeitmanagement sehr gut umgesetzt, jedoch hätte man beim Infrastrukturaufbau rund doppelt soviel Zeit einplanen können.

13.5 Konfiguration & Installation von Diensten

Die Installation und Konfiguration von Services wie NGINX, Datenbank, ffmpeg haben sehr viel Zeit und Geduld beansprucht. Einige Konfigurationen speziell im Bereich Videostreaming sind sehr aufwendig und erfordern sehr viel „Know-How“. Durch umfangreiche Literatur mussten die wichtigsten Informationen gefiltert und umgesetzt werden.

14 Zusammenfassung / Conclusion

Diese Projektarbeit untersucht wie sich Streaming Einstellungen auf den Stromverbrauch von Servern auswirken. Eine Umweltbelastung die für den Endverbraucher meist nicht ersichtlich ist. Es wurden zahlreiche Messungen durchgeführt und eine signifikante Anzahl an auswertbaren Daten generiert.

Im Vergleich zwischen den eingesetzten Streaming Servern ist deutlich aufgefallen, dass der ältere Server (HP ProLiant G6), wie erwartet, mit 150 Watt deutlich mehr Leistung im IDLE verbraucht als das Desktop System, welches sich mit nur 11 Watt begnügt. Bezogen auf den Energieverbrauch im IDLE stellten wir fest, dass der Server HP ProLiant G6 beim Streaming einen wesentlich geringeren Leistungsanstieg (maximal +105%) als das Desktop System, welches bis zu +700% mehr Leistung benötigte, erfuhr. Dies ist darauf zurückzuführen, da der HP Server ProLiant eine ältere Prozessorgeneration als das Desktop System implementiert hat.s

Die Datenauswertung führte zur Erkenntnis, dass sowohl die Auflösung als auch die Streaming Preset Einstellung einen signifikanten Einfluss auf den Leistungsverbrauch des Systems hat. So kann durch Reduktion der Auflösung von 1920x1080 auf 1280x720 eine Leistungseinsparung von beinahe 40% erreicht werden. Noch drastischer zeigen sich die Möglichkeiten der Leistungseinsparung bei den Presets ultrafast, medium und slower, welche in vielen Fällen eine Reduktion der Leistung um bis zu 63% bewirken können! Dies ist besonders interessant, da zwischen den Einstellungen slower und medium für das Projektteam kein erkennbarer Unterschied hinsichtlich der Bildqualität erkennbar war. Die Einstellung slower führte sogar zu einer schlechteren Erfahrung für den Zuseher, da hier oft bei der Transcodierung auf dem Server Frames verloren gehen und dies im Stream erkennbar wurde. Diese CPU seitige Limitierung trat auf dem HP G6 Server als auch auf dem moderneren HP Elite Desk 8300 auf. Selbst eine Preset Einstellung von ultrafast führte zu keiner maßgeblichen Reduktion der Streaming Qualität, zumindest nicht in dem Ausmaß wie wir uns dies erwartet hätten. Dies ist allerdings stark abhängig vom Anwendungsfall, würde der Stream einen Dynamischeren Hintergrund haben, würde sich eine schnellere Encodierung stärker bemerkbar machen. Eine Reduktion der Auflösung führte direkt zu einem negativen Effekt, da dies auf einem FULL-HD Monitor zu einem deutlich unschärferen Bild führte und somit weniger Details erkennbar waren.

Bezogen auf den Leistungsverbrauch lässt sich die Empfehlung abgeben, dass es sinnvoll ist in einer Auflösung von 1920x1080 zu streamen da dies sonst zu einer negativen Erfahrung für die Zuseher führen könnte. Um Energie zu sparen sollte das Preset unbedingt auf medium oder schneller gesetzt werden, da dies die Qualität nicht merklich beeinflusst, jedoch einen sehr hohen Einfluss auf den Stromverbrauch hat!

Zusätzlich wurde in unseren Messungen der Einfluss auf den Energieverbrauch bezüglich der Anzahl von Zusehern gemessen und ermittelt. Aufgrund von Bandbreitenlimitierungen war nur eine Messung bis maximal 500 Zusehern Sinnvoll. Ein Anstieg der Zuseher von 50 Zusehern auf 100 Zuseher bei einer Auflösung von 1920x1080 (ultrafast), führte auf dem HP Elite Desk 8300 G2 zu einem Anstieg um sechs Watt. Eine weitere Erhöhung auf 500 Zuseher

resultierte in einem Verbrauchsanstieg eines Watts. Grund für den geringen Watt Unterschied zwischen 100 und 500 Zusehern, ist, dass die maximale Anzahl an gleichzeitigen Zugriffen nicht wesentlich höher war da die Gigabit Ethernet Schnittstelle zu 100% ausgelastet war. Der Zugriff mit 500 Usern dauerte daher wesentlich länger, was auch zu Verzögerungen beim Enduser führen würde. Bei dem HP ProLiant G6 Server war ein noch deutlicher Anstieg des Stromverbrauchs erkennbar da dieser von 50 auf 100 Zuseher bereits um 10 Watt mehr benötigte. Ein Anstieg auf 500 Zuseher führte hier zu einem noch deutlicheren Anstieg um 20 Watt! Der deutlichere Unterschied zwischen 100 und 500 Zusehern kann hier durch die schnellere Abarbeitung der Requests dank der leistungstärkeren Hardware erklärt werden. Der viel höhere Stromverbrauch der Server Hardware ist darauf zurückzuführen, dass der HP Server mit einer deutlich ältere CPU Generation als der Elite Desk 8300 G2 ausgestattet ist und zusätzlich über zwei CPU's verfügt, welche in Summe mehr Leistung verbrauchen.

Bezüglich des Energieverbrauchs bei mehreren Zusehern lässt sich die Empfehlung abgeben, eine möglichst aktuelle Prozessorgeneration einzusetzen, welche diese Last bewältigen kann und mit einer hohen Energieeffizienz verarbeiten kann.

15 Projektantrag

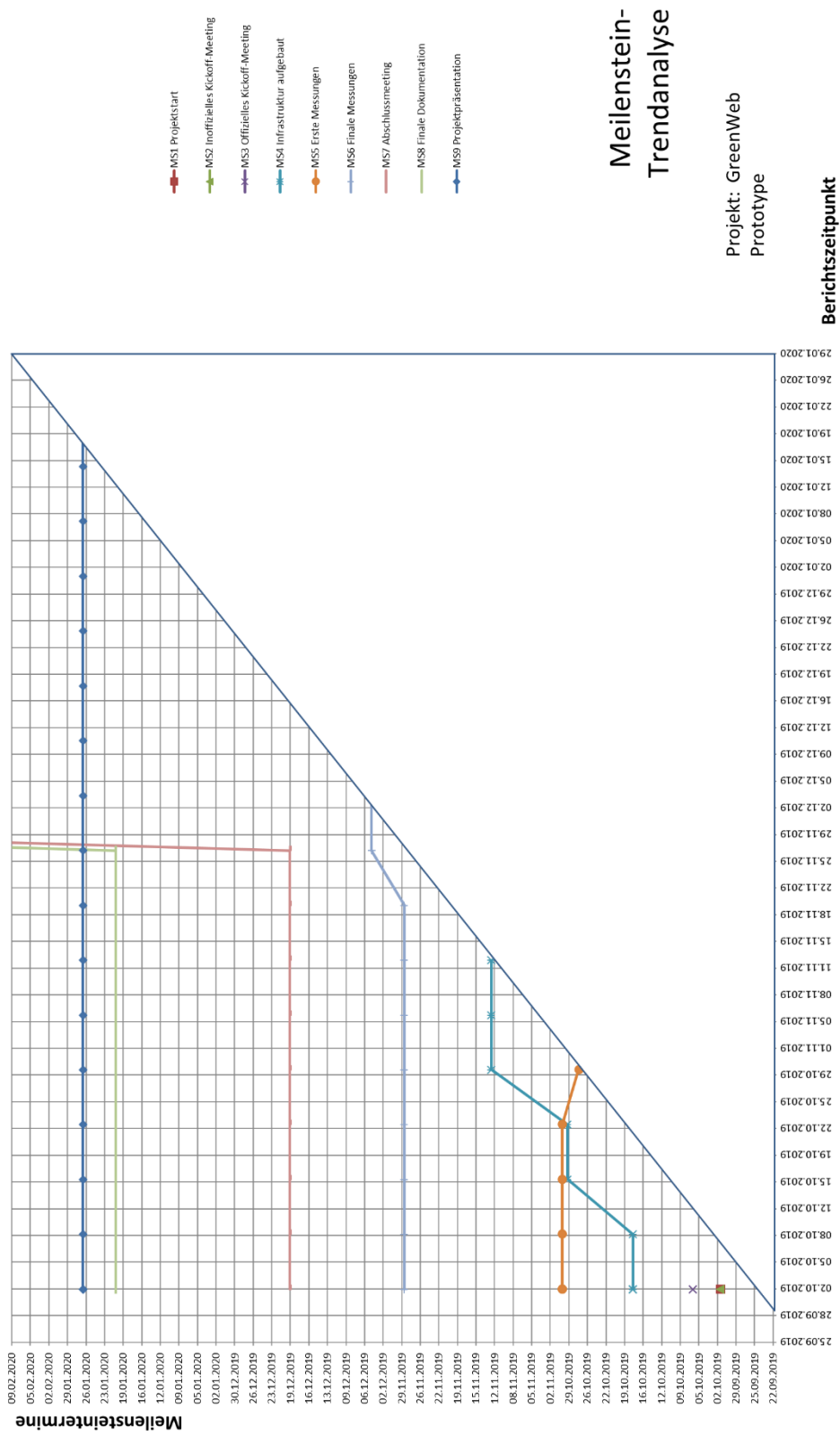
Projektname:	GreenWeb Prototype Prototype-Entwicklung für Energie- und Performance-Messungen														
Projektnummer:	6														
Projektantrag															
Vorhaben / Projektziele:	<p>Mit diesem Projekt sollen folgende Ziele erreicht werden:</p> <ol style="list-style-type: none"> 1. Recherche und Verweis auf etwaige bekannte Lösungsansätze für Green Web / IT 2. Aufbau einer Infrastruktur zur Messung und Datenerhebung bezüglich des Vergleichs des Energieverbrauchs in folgenden Anwendungsbereichen: <ol style="list-style-type: none"> a. Video-Streaming Dienst (1080p, 4k, h265, h264) b. Energieverbrauch von Verschlüsselungstechniken c. Dynamische & Statische Webseiten d. Server / Client Energiemessungen 3. Kontaktaufnahme & Recherche bezüglich GreenWeb / GreenIT incl. externe Quellen z.B. Rechenzentrum 4. Datenerhebung damit diese vom anderen Team visuell aufbereitet werden können. 														
Meilensteine	<ol style="list-style-type: none"> 1. Projektstart 2. Inoffizielles Kickoff-Meeting 3. Offizielles Kickoff-Meeting 4. Infrastruktur aufgebaut 5. Erste Messergebnisse 6. Finale Messergebnisse 7. Abschlussmeeting 8. Finale Dokumentation 9. Projektpräsentation 10. Projektabgabe 11. Projektende 														
Fertigstellungstermine	<table> <tr> <td>10.10.19</td><td>Projektantrag</td></tr> <tr> <td>10.10.19</td><td>Meilensteine definiert</td></tr> <tr> <td>25.10.19</td><td>Infrastruktur aufbauen & einrichten</td></tr> <tr> <td>22.11.19</td><td>Messungen durchführen</td></tr> <tr> <td>15.12.19</td><td>Ergebnisse analysieren</td></tr> <tr> <td>22.01.20</td><td>Projektdokumentation</td></tr> <tr> <td>27.01.20</td><td>Projektpräsentation / Abgabe</td></tr> </table>	10.10.19	Projektantrag	10.10.19	Meilensteine definiert	25.10.19	Infrastruktur aufbauen & einrichten	22.11.19	Messungen durchführen	15.12.19	Ergebnisse analysieren	22.01.20	Projektdokumentation	27.01.20	Projektpräsentation / Abgabe
10.10.19	Projektantrag														
10.10.19	Meilensteine definiert														
25.10.19	Infrastruktur aufbauen & einrichten														
22.11.19	Messungen durchführen														
15.12.19	Ergebnisse analysieren														
22.01.20	Projektdokumentation														
27.01.20	Projektpräsentation / Abgabe														
Kosten	<p>Grundsätzlich entsteht kein Kostenaufwand</p> <p>Mögliche Hardwarekosten: Energiemessgerät (z.B. Smart Meter)</p> <p>Lizenzkosten für Windows Server etc. werden von der FH-JOANNEUM gedeckt.</p> <p>Andere Kosten: Zeit, Tickets für öffentliche Verkehrsmittel</p>														

Personenaufwand	Kostenlos
Sachmittel	<p>Server: HP ProLiant DL380 G6</p> <p>Geräte: HP EliteDesk 800 G1, HP HP Elite Desk 8300 G2, Smartphone, Tablet, Smart Meter, Messgeräte zur Energiemessung, Peripheriegeräte, Netzwerkanbindung</p> <p>Software: Windows Server 2019, Hyper-V, Linux Distro, Open Source Software</p>
Risiko	Zeitmanagement, Überarbeitung, Inhaltliche Differenzen, keine aussagekräftigen Ergebnisse, Projektzielverfehlung
Projektorganisation	
Auftraggeber	FH Joanneum – DI Georg Mittenecker
Projektleiter	Colin Jochum
Projektmitglieder	Colin Jochum, Tom Kleinhapl, Marcel Kahr
Projektausschuss	DI Werner Fritz, DI Georg Mittenecker, Colin Jochum, Tom Kleinhapl, Marcel Kahr





























16 Meilensteine

02.10.19	Projektstart
02.10.19	Inoffizielles Kickoff-Meeting
07.10.19	Offizielles Kickoff-Meeting
09.10.19	Meilensteindefinition
10.10.19	Projektantrag
18.10.19	Infrastruktur aufgebaut
08.11.19	Erste Messergebnisse
22.11.19	Finale Messergebnisse
20.12.19	Abschlussmeeting
21.01.20	Finale Dokumentation
27.01.20	Projektpräsentation
27.01.20	Projektabgabe
27.01.20	Projektende

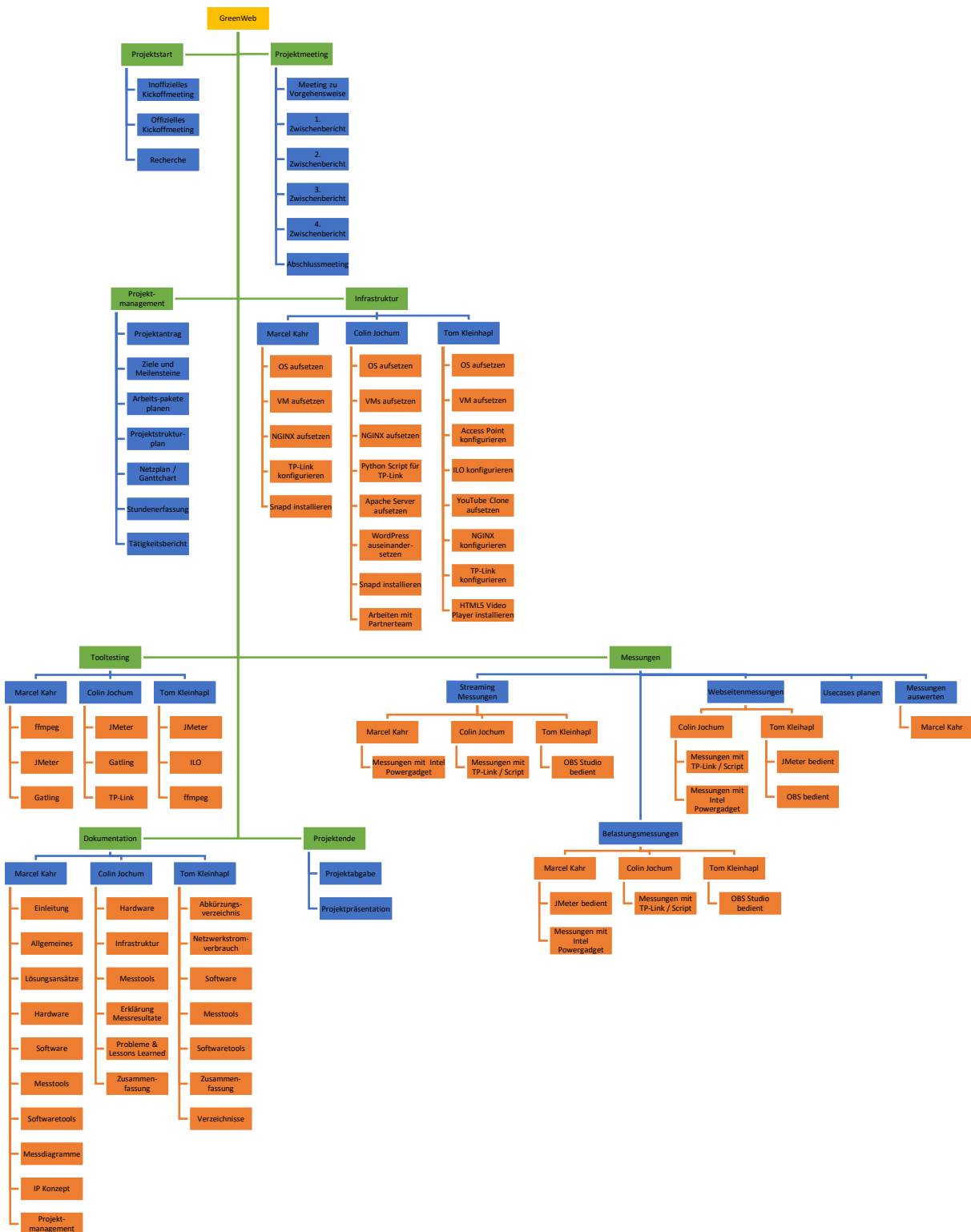
17 Meilensteintrendanalyse



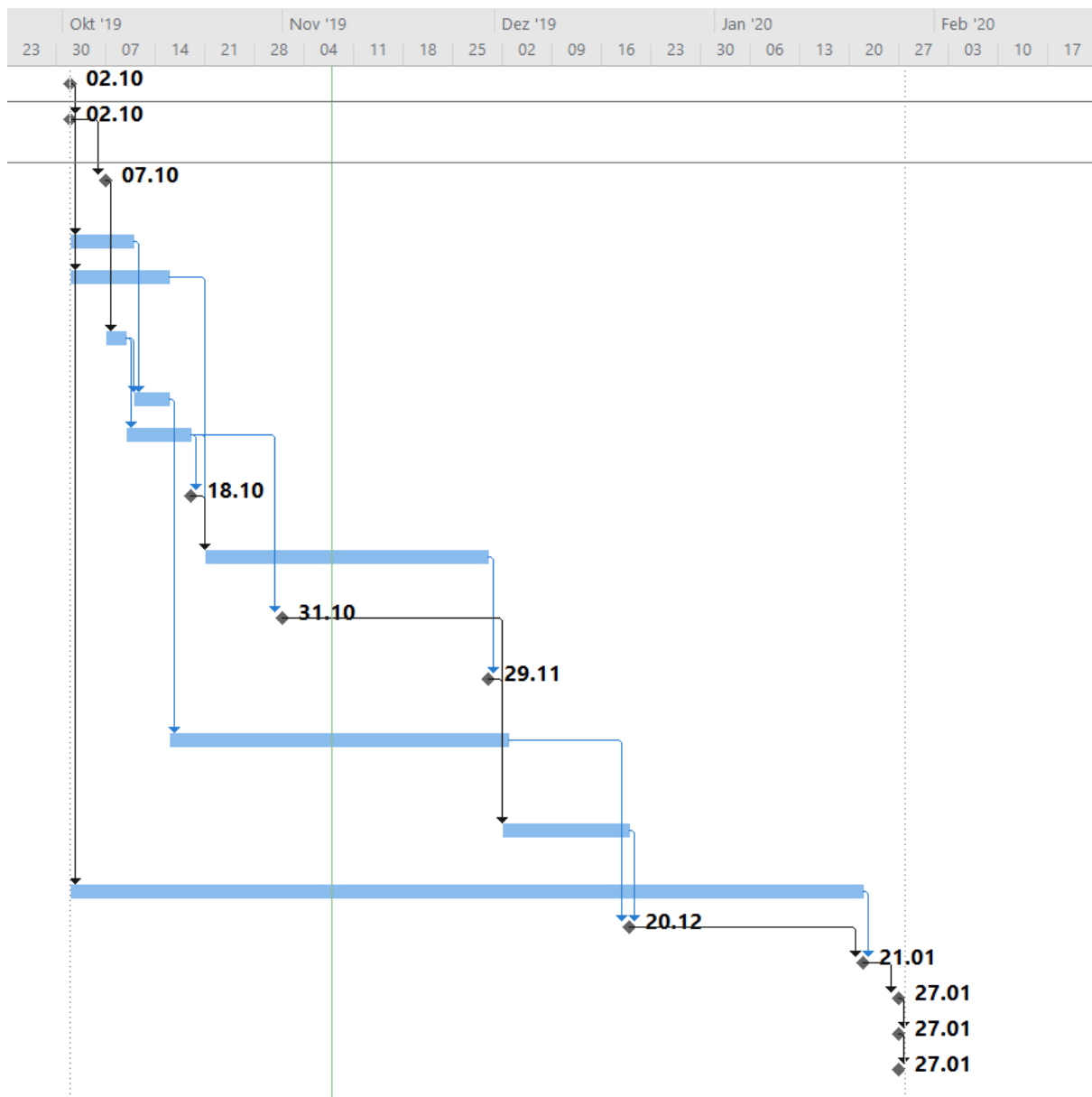
18 Projektplan

		Vorgan ▾	Vorgangsname ▾	Dauer ▾	Anfang ▾	Ende ▾	Vorgänger
1			Projektstart	0 Tage	Mit 02.10.19	Mit 02.10.19	
2			Inoffizielles Kickoff-Meeting	0 Tage	Mit 02.10.19	Mit 02.10.19	1
3			Offizielles Kickoff-Meeting	0 Tage	Mon 07.10.19	Mon 07.10.19	2
4			Projektantrag	7 Tage	Mit 02.10.19	Don 10.10.19	2
5			Recherche / Ideenfindung	10 Tage	Mit 02.10.19	Die 15.10.19	1
6			Zielvereinbarungen / Meilensteine	3 Tage	Mon 07.10.19	Mit 09.10.19	3
7			Arbeitspakete planen	3 Tage	Fre 11.10.19	Die 15.10.19	6;4
8			Infrastruktur aufbauen	7 Tage	Don 10.10.19	Fre 18.10.19	6
9			Infrastruktur aufgebaut	0 Tage	Fre 18.10.19	Fre 18.10.19	8
10			Messungen durchführen	30 Tage	Mon 21.10.19	Fre 29.11.19	8;9;5
11			Erste Messergebnisse	0 Tage	Don 31.10.19	Don 31.10.19	8
12			Finale Messergebnisse	0 Tage	Fre 29.11.19	Fre 29.11.19	10
13			Sonstige Informationen sammeln	34 Tage	Mit 16.10.19	Mon 02.12.19	7
14			Ergebnisse analysieren	14 Tage	Mon 02.12.19	Don 19.12.19	11;12
15			Projektdokumentation	80 Tage	Mit 02.10.19	Die 21.01.20	2
16			Ablussmeeting	0 Tage	Fre 20.12.19	Fre 20.12.19	14;13
17			Finale Dokumentation	0 Tage	Die 21.01.20	Die 21.01.20	15;16
18			Projektpräsentation	0 Tage	Mon 27.01.20	Mon 27.01.20	17
19			Projektabgabe	0 Tage	Mon 27.01.20	Mon 27.01.20	18
20			Projektende	0 Tage	Mon 27.01.20	Mon 27.01.20	19

19 Projektstrukturplan

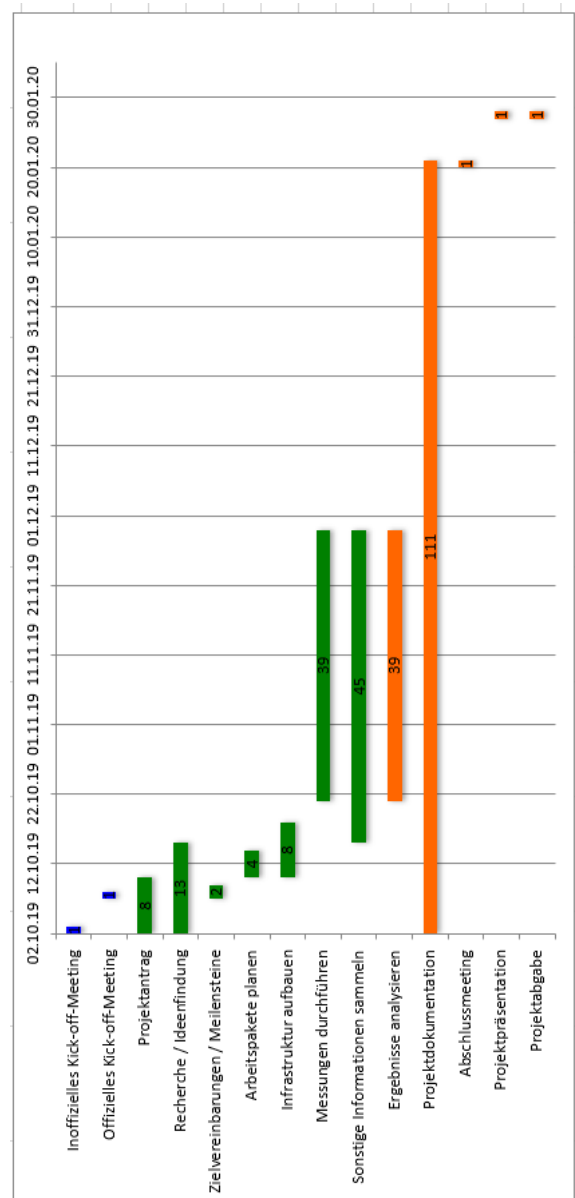


20 Gantt Chart



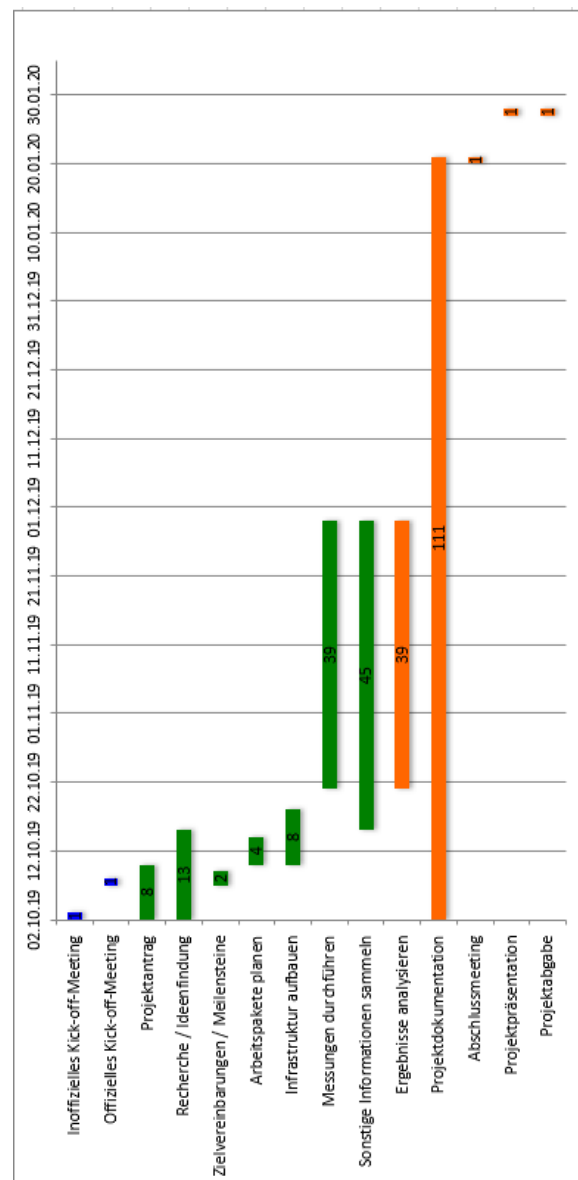
21 Stundenerfassung Marcel Kahr

Code	Aufgaben	Verantwortlich	Start	Soll-Ende	Ist-Ende	Tage	Stunden geplant	Stunden benötigt	Priorität	Bemerkung	Status
1	Inoffizielles Kick-off-Meeting	Colin Jochum	02.10.19	02.10.19	02.10.19	1	1.25	1.25	Hoch		Abgeschlossen
2	Offizielles Kick-off-Meeting	Colin Jochum	07.10.19	07.10.19	07.10.19	1	1	1	Hoch		Abgeschlossen
3	Projektantrag	Colin Jochum	02.10.19	10.10.19	09.10.19	8	2.75	2.5	Hoch		Abgeschlossen
4	Recherche / Ideenfindung	Tom Kleinhapl	02.10.19	15.10.19	18.10.19	13	10	8	Hoch		Abgeschlossen
5	Zielvereinbarungen / Meilensteine	Marcel Kahr	07.10.19	09.10.19	09.10.19	2	1	1	Hoch		Abgeschlossen
6	Arbeitspakete planen	Marcel Kahr	10.10.19	14.10.19	14.10.19	4	1	1	Hoch		Abgeschlossen
7	Infrastruktur aufbauen	Colin Jochum	10.10.19	18.10.19	17.10.19	8	30	60.35	Hoch		Abgeschlossen
8	Messungen durchführen	Tom Kleinhapl	21.10.19	29.11.19	05.12.19	39	20	22.75	Hoch		Abgeschlossen
9	Sonstige Informationen sammeln	Tom Kleinhapl	15.10.19	29.11.19	19.12.19	45	8	0	Hoch	Floss in Doku mitein	Nicht begonnen
10	Ergebnisse analysieren	Marcel Kahr	21.10.19	29.11.19	19.12.19	39	5	5	Hoch		Abgeschlossen
11	Projektdokumentation	Marcel Kahr	02.10.19	21.01.20	20.12.19	111	18	38	Hoch		Abgeschlossen
12	Abschlussmeeting	Marcel Kahr	20.01.20	20.01.20	06.12.19	1	1	0.75	Hoch		Abgeschlossen
13	Projektpräsentation	Tom Kleinhapl	27.01.20	27.01.20		1	0.5	0	Hoch		Nicht begonnen
14	Projektabgabe	Colin Jochum	27.01.20	27.01.20		1	0.5	0	Hoch		Nicht begonnen
Gesamt							100	141.6			



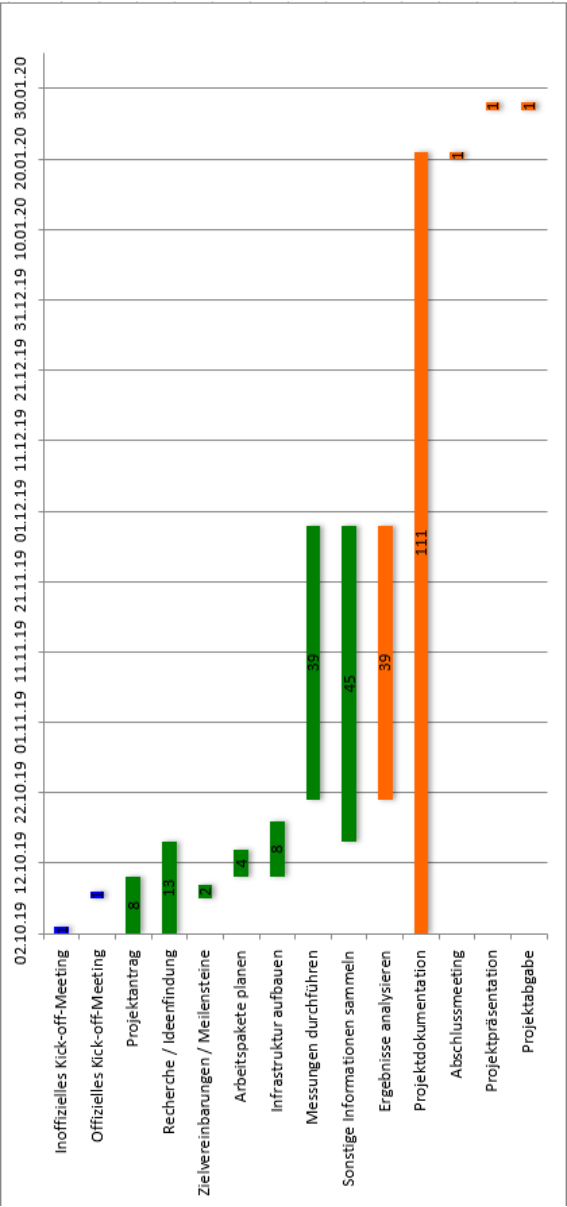
22 Stundenerfassung Colin Jochum

Code	Aufgaben	Verantwortlich	Start	Soll-Ende	Ist-Ende	Tage	Stunden geplant	Stunden benötigt	Priorität	Bemerkung	Status
1	Inoffizielles Kick-off-Meeting	Colin Jochum	02.10.19	02.10.19	02.10.19	1	1.25	1.25	Hoch		Abgeschlossen
2	Offizielles Kick-off-Meeting	Colin Jochum	07.10.19	07.10.19	07.10.19	1	1	1	Hoch		Abgeschlossen
3	Projektantrag	Colin Jochum	02.10.19	10.10.19	09.10.19	8	2.75	2.5	Hoch		Abgeschlossen
4	Recherche / Ideenfindung	Tom Kleinhlapl	02.10.19	15.10.19	18.10.19	13	10	8	Hoch		Abgeschlossen
5	Zielvereinbarungen / Meilensteine	Marcel Kahr	07.10.19	09.10.19	09.10.19	2	1	1	Hoch		Abgeschlossen
6	Arbeitspakete planen	Marcel Kahr	10.10.19	14.10.19	14.10.19	4	1	1	Hoch		Abgeschlossen
7	Infrastruktur aufbauen	Colin Jochum	10.10.19	18.10.19	17.10.19	8	30	50	Hoch	bis zum 13. Treffen	Abgeschlossen
8	Messungen durchführen	Tom Kleinhlapl	21.10.19	29.11.19	05.12.19	39	20	30	Hoch		Abgeschlossen
9	Sonstige Informationen sammeln	Tom Kleinhlapl	15.10.19	29.11.19	19.12.19	45	8	5	Hoch		Abgeschlossen
10	Ergebnisse analysieren	Marcel Kahr	21.10.19	29.11.19	19.12.19	39	5	5	Hoch		Abgeschlossen
11	Projektdokumentation	Marcel Kahr	02.10.19	21.01.20	20.12.19	111	18	18	Hoch		Abgeschlossen
12	Abschlussmeeting	Marcel Kahr	20.01.20	20.01.20	06.12.19	1	1	0.5	Hoch		Abgeschlossen
13	Projektpräsentation	Tom Kleinhlapl	27.01.20	27.01.20		1	0.5		Hoch		Nicht begonnen
14	Projektabgabe	Colin Jochum	27.01.20	27.01.20		1	0.5		Hoch		Nicht begonnen
Gesamt			117				100	123.25			



23 Stundenerfassung Tom Kleinhapl

Code	Aufgaben	Verantwortlich	Start	Soll-Ende	Ist-Ende	Tage	Stunden geplant	Stunden benötigt	Priorität	Bemerkung	Status
1	Inoffizielles Kick-off-Meeting	Colin Jochum	02.10.19	02.10.19	02.10.19	1	1.25	1.25	Hoch		Abgeschlossen
2	Offizielles Kick-off-Meeting	Colin Jochum	07.10.19	07.10.19	07.10.19	1	1	1	Hoch		Abgeschlossen
3	Projektantrag	Colin Jochum	02.10.19	10.10.19	09.10.19	8	2.75	2	Hoch		Abgeschlossen
4		Tom Kleinhapl	02.10.19	15.10.19	18.10.19	13	10	8	Hoch		Abgeschlossen
5	Zielvereinbarungen / Meilensteine	Marcel Kahr	07.10.19	09.10.19	09.10.19	2	1	1	Hoch		Abgeschlossen
6	Arbeitspakete planen	Marcel Kahr	10.10.19	14.10.19	14.10.19	4	1	1	Hoch		Abgeschlossen
7	Infrastruktur aufbauen	Colin Jochum	10.10.19	18.10.19	17.10.19	8	35	63	Hoch		Abgeschlossen
8	Messungen durchführen	Tom Kleinhapl	21.10.19	29.11.19	05.12.19	39	20	23	Hoch		Abgeschlossen
9	Sonstige Informationen sammeln	Tom Kleinhapl	15.10.19	29.11.19	19.12.19	45	8	6	Hoch	Floss in Doku mitein	Nicht begonnen
10	Ergebnisse analysieren	Marcel Kahr	21.10.19	29.11.19	19.12.19	39	5	5	Hoch		Abgeschlossen
11	Projektdokumentation	Marcel Kahr	02.10.19	21.01.20	20.12.19	111	13	16	Hoch		Abgeschlossen
12	Abschlussmeeting	Marcel Kahr	20.01.20	20.01.20	06.12.19	1	1	1	Hoch		Abgeschlossen
13	Projektpräsentation	Tom Kleinhapl	27.01.20	27.01.20		1	0.5	0.5	Hoch		Nicht begonnen
14	Projektabgabe	Colin Jochum	27.01.20	27.01.20		1	0.5	0	Hoch		Nicht begonnen
Gesamt							100	128.75			



24 Tätigkeitsbericht

	1. Treffen Aufbau Infrastruktur am 09.10.2019 (11 Uhr - 18:30 Uhr)
Colin	<ul style="list-style-type: none"> • Meilensteine überarbeitet und festgelegt • Projektantrag finalisiert • PowerPoint Präsentation für PMGL → Projektcheck 1 erstellt <ul style="list-style-type: none"> ○ Projektantrag, Grundparameter, Problemfeldanalyse • 1. Aufsetzversuch von Windows Server 2019 auf Serverhardware Version 2009 → dies verursachte einige Probleme • Windows 10 Desktop-Client aufgesetzt • Windows Server 2019 auf Desktop-PC aufgesetzt <ul style="list-style-type: none"> ○ Hyper-V installiert ○ Einen virtuellen Debian Server aufgesetzt ○ Zwei virtuelle Windows Pro Versionen aufgesetzt • Testen auf Kompatibilität diverser Energie Messungstools <ul style="list-style-type: none"> ○ Intel Power Gadget und Joulemeter nur am Parent möglich, nicht in der VM • Recherche diverser Toolkomponenten → Smartmeter (TP-Link)
Tom	
Marcel	
	2. Treffen Aufbau Infrastruktur am 10.10.2019 (12:45 Uhr - 17 Uhr)
Colin	<ul style="list-style-type: none"> • Versuchter Aufbau einer Internetverbindung mit Hilfe von WLAN-Adaptern zum Server von Herrn Geister <ul style="list-style-type: none"> ○ Verbindung konnte nicht hergestellt werden, da kein passender Treiber im Internet gefunden wurde • Persönliche Kontaktaufnahme mit Herrn Geister bezüglich Internetzugriff <ul style="list-style-type: none"> ○ Neuinstallation der auf Linux basierenden VM am Desktop-Client, aufgrund von Konfigurationsproblemen beim GUI am Vortag -> GUI LXQt (Leightweighted) ○ Netzwerkeinstellungen im GUI -> Statische IPs ○ Erstellung VM in Azure Cloud ○ Installation und Testen der Tools "PowerTop" und "PowerState" (gleiches Problem bei Azure VM) ○ Da wir keine Verbindung zum "RAPL" herstellen konnten, (kann in VMs nicht genutzt werden, nur am Parent), können mit dieser Schnittstelle konnten keine Daten zur Energiemessung ausgelesen werden ○ Nach Recherchen und Tests konnte herausgefunden werden, dass es mit Hilfe des Tools „Speed“ etwaige Änderungen des Energieverbrauchs beim Intel Power Gadget am Parent gab.
Tom	
Marcel	

	<ul style="list-style-type: none"> • "Hacken" des von der FH zur Verfügung gestellten Servers mit Hilfe von Recherchen aus dem Web, da das Root-Passwort gesetzt war und wir keinen Zugriff auf den Server hatten • Erstellung einer VM am HP Server • Mit Hilfe der "XCP-ng Center" App, konnte zum Server eine SSH Verbindung aufgebaut werden, sowie diverse VMs erstellt werden • Nach einigen Schwierigkeiten beim Einlesen des ISO-Files (DEBIAN), welches wir aus dem Internet herunterladen konnten, wurde erfolgreich die Installation einer virtuellen Linux-Maschine am Server sowie etwaige Netzwerkkonfigurationen durchführen werden
	3. Treffen Aufbau Infrastruktur am 14.10.2019 (10:10 Uhr - 12:20 Uhr)
Colin	<ul style="list-style-type: none"> • Es wurde eine native Linux-Partition als zweite Partition am Desktop-Client G2 aufgesetzt.
Marcel	<ul style="list-style-type: none"> • Weiters wurde ein NGINX-Webserver auf VM Debian am Windows Server aufgesetzt • Die Installation und Konfiguration der BSD Streaming Software wurde auf der VM durchgeführt • 1. Streamingversuch über Server via RTMP-Protokoll, welcher jedoch nicht funktionierte <ul style="list-style-type: none"> ○
Tom	<ul style="list-style-type: none"> • Inbetriebnahme und Konfiguration eines Access Points mit OpenWrt <ul style="list-style-type: none"> ○ Die Konfiguration konnte aufgrund von Komplikationen nicht vollständig umgesetzt werden
	4. Treffen Aufbau Infrastruktur am 17.10.2019 (11 - 17 Uhr)
Colin	<ul style="list-style-type: none"> • Beseitigung von Installationsproblemen mit NGINX • Installation des Mediaplayers Kodi (Media Server) auf VM Debian <ul style="list-style-type: none"> ○ Konfiguration der Kodi-Software im Web-Interface, um auf den Media Player via http zugreifen zu können ○ Download diverser Video Clips in unterschiedlichen Qualitäten (4k, UHD, 1080p) und Zugriff auf diese Clips ○ Es stellte sich heraus, dass das Streamen von Kodi nur über den Internet Explorer möglich ist • Durchführung erster Leistungsmessungen mit privaten Notebooks durch Streaming der Videos über Kodi <ul style="list-style-type: none"> ○ Das Ergebnis zeigte, dass eine kurzfristige CPU-Auslastung von rund 100% erreicht wurde
Marcel	
Tom	<ul style="list-style-type: none"> • Erneute Konfiguration des Access Points, um via Wi-Fi auf den Media-Server zugreifen zu können

	<ul style="list-style-type: none"> ○ Dabei wurden wiederum erste Leistungs- und Energieverbrauchstests durchgeführt • ILO-Schnittstelle wurde konfiguriert und getestet, ob man darauf zugreifen kann <ul style="list-style-type: none"> ○ Der Zugriff ist nur mit Hilfe älterer Explorer wie dem Internet Explorer möglich. Außerdem ist sehr interessant, dass Energiemessungen nur im Intervall von 24 Stunden aufgezeichnet werden können
	5. Treffen Aufbau Infrastruktur am 23.10.2019 (11 Uhr – 18:30 Uhr)
Colin	<ul style="list-style-type: none"> • Recherchen zu geeignete APIs, um Daten vom Intel Power Gadget auslesen zu können
Marcel	<ul style="list-style-type: none"> • Erste Ergebnisse konnten via CSV-File in Excel eingelesen und aufgelistet werden • Recherche bezüglich HSL via RMTP und NGINX • Nach der Recherche nach geeigneten Anleitungen, versuchten wir durch Änderungen des NGINX.conf-Files "HLS" freizuschalten. Da sich herausstellte, dass nach der Manipulation des config-Files sich NGINX nicht mehr starten lässt, mussten wir den Fehler debuggen • Das Debugging zeigte, dass der Fehler beim HLS liegt • Letztendlich setzten wir eine neue VM basierend auf CentOS (vom ftp-Server der TU-Graz) auf und befolgten eine „tolle“ Schritt für Schritt Anleitung, von der kompletten Neuinstallation von NGINX bis zum Aufsetzen des Streaming-Servers. • Da wir nicht mit alten Versionen arbeiten wollten, suchten wir natürlich die neuesten Releases und installierten diese. Auch neue Directories legten wir an, da wir das OS so lightweighted wie möglich aufgesetzt haben. • Dieses lightweighted OS sowie die neuen Versionen wurden uns zum Verhängnis, da diese nicht mit der Anleitung (Konfiguration) übereinstimmten. Hinzu kam, dass die Anleitung teilweise fehlerhaft war und Fehler wie (zu wenige geschwungene Klammern usw.) beinhaltete. • Nach langer Fehlersuche konnten wir die Installation von NGINX mit HLS abschließen und zum Streamen beginnen. • Beim Versuch einen Clip mit Hilfe von ffmpeg zu streamen, erhielten wir wiederum einen Fehler nämlich "Server error: NetStream.Record.Failed" und beließen es für den Tag.
Tom	<ul style="list-style-type: none"> • YouPHPTube setup auf debian VMs
	6. Treffen Aufbau Infrastruktur am 24.10.2019 (12:30 Uhr – 19:30 Uhr)

Colin	<ul style="list-style-type: none"> Nach dreistündiger Recherche und Ausprobieren von etwaigen ffmpeg-Befehlen, konnten wir zufällig beim Betrachten der Fehlermeldung, beim Ausführen des ffmpeg-Befehls herausfinden, dass man im Config-File von NGINX den Parameter "Record" auf off stellen muss. Danach konnte der ffmpeg-Befehl problemlos ausgeführt werden. Es wurden einige Probemessungen mit unterschiedlichen Parametern von ffmpeg durchgeführt, wie z.B. (unterschiedliche Kodierungen (libx264 usw.). Da standardmäßig "copy" verwendet wird, war die CPU des PCs bei minimaler Leistung. Erst nach einfügen von "libx264" erzielte die CPU max. Auslastung (100% bei allen Cores). Danach konnten wir durch diverse Einstellungen mithilfe von Parametern die CPU-Auslastung drosseln. Als nächsten Schritt versuchten wir den Stream über HLS (http) zu erreichen. Nach einigen Problemen und Recherchen konnten wir die richtige Anleitung finden. Dabei mussten wir den ffmpeg-Befehl direkt auf unserer CentOS-VM ausführen, da das MP4-File in HLS Segmente gespeichert wird. Aus diesen Segmenten wird eine Playlist generiert, auf welche per http://172.17.32.53/live/playlist.m3u8 zugegriffen werden kann. In der Config von NGINX wurde nämlich festgelegt, dass man auf den Ordner Live für HLS zugreifen kann. Da beim Streamen der Playlist, die Playlist automatisch von NGINX gelöscht wird, mussten wir in der NGINX Config "hls_cleanup = off" einstellen, damit man unendlich streamen kann
Marcel	
Tom	
	7. Treffen Aufbau Infrastruktur am 29.10.2019 (13:45 Uhr – 17 Uhr)
Colin	<ul style="list-style-type: none"> Projektcheck für PMTGL vorbereitet Versuch ffmpeg Befehl für HLS in die NGINX Konfigurationsdatei (NGINX.conf) zu implementieren Erneute Überarbeitung Nginx config
Tom	
Marcel	
	8. Treffen Aufbau Infrastruktur am 30.10.2019 (10:10 Uhr – 17 Uhr)
	<ul style="list-style-type: none"> 30 Minuten Meeting um 14 Uhr mit Sandra Schadenbauer und Georg Mittenecker → Vorführung des bereits umgesetzten und was geplant ist <ul style="list-style-type: none"> Wichtige Punkte des Meetings: Echtzeitauslesung und Übertragung der Daten, sowie verschlüsselter Stream

Colin	<ul style="list-style-type: none"> • Versuch mit der API Daten vom Intel Power Gadget in ein eigens File zu speichern, welches das Partnerteam zum Visualisieren benötigt
Tom	<ul style="list-style-type: none"> • Versuch NGINX auch auf dem HP Server zu installieren, da die Installation erfolgreich am CentOS umgesetzt wurde <ul style="list-style-type: none"> ○ Verursachte Konfigurationsprobleme, welche erst nach stundenlanger Recherche gelöst werden konnte → Firewall Portfreischaltung ○ Die Konfigurationsdatei von NGINX wurde angepasst und verbessert • Projektdokumentation erstellt sowie die wichtigsten Überschriften ergänzt • Erstellung einer PowerPoint für den 2. Projektcheck in PMTGL
Marcel	
	9. Treffen am 05.11.2019 (15 Uhr – 17:15 Uhr)
Colin	<ul style="list-style-type: none"> • Erster Versuch unsere Wi-Fi Plugs zu konfigurieren <ul style="list-style-type: none"> ○ Konnte leider nicht umgesetzt werden, da für die Konfiguration ein DHCP benötigt wird, welchen wir jedoch nicht am Access Point eingestellt hatten → Zur Genehmigung musste mit Herrn Geister Kontakt aufgenommen werden. • Durchführung von Recherchen für den HP Server Xenon, bezugnehmend auf etwaige Energie Messungstools <ul style="list-style-type: none"> ○ Bis auf die ILO-Schnittstelle wurde kein vergleichbares Tool gefunden • Recherche etwaiger ffmpeg-Befehle zur Codierung • Recherche diverser Stresstesttools um http Request Traffic zu generieren
Tom	
Marcel	
	10. Treffen am 06.11.2019 (10 Uhr – 14 Uhr)
	<ul style="list-style-type: none"> • Meeting mit anderem Projektteam <ul style="list-style-type: none"> ○ Absprache auf zukünftige Vorhaben z.B. Echtzeitmessungen ○ Projektteam Visualisierung stellt uns ihre statische und dynamische Website zur Verfügung, um Tests damit durchzuführen
Colin	<ul style="list-style-type: none"> • Aufsetzen eines Apache-Servers für die Website des Partnerteams • Zugriff mit Hilfe von Python und über eine API auf den Wi-Fi Plug

Tom	<ul style="list-style-type: none"> • Testen und arbeiten mit dem Stresstesttool JMeter und herausfinden, wie wir am besten Loadtests durchführen können
Marcel	<ul style="list-style-type: none"> • Recherche für etwaige andere Stresstesttools • Inbetriebnahme und Konfiguration der Wi-Fi Plugs mit der App Kasa sowie eine Neuvergabe von IP Adressen, da zur Konfiguration der Plugs ein eigener DHCP-Server benötigt wurde. Dabei musste uns ein internes Netz von Herrn Geister zur Verfügung gestellt werden, damit unser eigener DHCP nicht ins FH Netz routet. • Neu Konfiguration der Statischen IPs im 192.168.47.0/24 Netzwerk • Anpassung der IPs in diversen Konfigurations Files
	11. Treffen am 08.11.2019 (11:15 Uhr – 15:45 Uhr)
Colin	<ul style="list-style-type: none"> • TP-Link → Python3 Script inkl. Plotfunktion • Livescatterplot • Datenspeicherung in Listen und Datenspeicherung in CSV
Tom	<ul style="list-style-type: none"> • YouTube Clone gefixt, da durch die Neuvergabe der IP's die Konfiguration beschädigt wurde • Einige Probetests mit dem Jmeter Tool durchgeführt
Marcel	<ul style="list-style-type: none"> • Beginn der Dokumentation wie Zielsetzung, Einleitung, Struktur und Aufbau • Beginn der Definition von Use Cases für Messungen
	12. Treffen am 12.11.2019 (10 Uhr – 17:15 Uhr)
Colin Marcel	<ul style="list-style-type: none"> • Versuch auf den CentOS-VMs, zuerst auf der G2 und dann auf dem HP Server die neueste Version von ffmpeg zu installieren, da mit der neueren Version mehr Optionen möglich sind. • Da wir zuvor nur herausfanden, dass ffmpeg nur bis zur Version 2 für CentOS supported wird, konnten wir nur Version 2 installieren. Doch nach einigen Recherchen kamen wir drauf, dass doch die Version 4 mit Hilfe eines Plug-Ins namens "snapd" installiert werden kann. • Dabei mussten wir zuerst die alte ffmpeg Version von CentOS löschen, um die neue mit Hilfe von Snapd installieren zu können. Im nächsten Schritt versuchten wir, ob mit der neuen Version NGINX noch ausgeführt werden konnte. Das Ergebnis zeigte, dass es keine Probleme beim Ausführen und Starten von NGINX gab. • Einziger Fehler war, dass der Stream, welcher mit ffmpeg im Browser als Mobile Version zurückgegeben werden hätte sollen, dass index.m3u8 File im Mobile Ordner nicht fand.

	<ul style="list-style-type: none"> • Nach über zweistündiger Fehlersuche, Recherche und Tests wie z.B. das Einfügen eines NGINX Users (user root) und das verwenden eines push-Befehls für die Subapplications (Moblie) in der NGINX Config, funktionierte aus irgendeinem unerklärlichen Grund das Aufrufen des Streams plötzlich. • Somit konnten wir die vorgenommenen Einstellungen und Konfiguration auch für die HP Greta CentOS verwenden, was auch in kürzester Zeit umgesetzt wurde. • Weiters versuchten wir mit dem JMeter-Tool Traffic zu generieren, was sich schwieriger als gedacht herausstellte. Mit Hilfe einer Anleitung von Blazemeter konnten einige Testversuche durchgeführt werden, jedoch passten unsere Ergebnisse nicht mit den Ergebnissen nicht überein. Somit ließen wir diesen Punkt offen. • Zu guter Schluss fertigten wir einige Usecases an, was wir eigentlich Messen möchten wie z.B. unterschiedliche ffmpeg Einstellungen (mit Codierungen) und z.B. ohne Codierungen usw.
Tom	<ul style="list-style-type: none"> • Neukonfiguration der ILO Schnittstelle • Erstellung eines Python Skripts, welches über die API Daten im im JSON Format ausliest (wie bei der TP-Link HS110) • Probe Streams
	13. Treffen am 13.11.2019 (12:15 Uhr – 17:15 Uhr)
Colin	<ul style="list-style-type: none"> • Aufsetzen des Apache Servers für die WordPress Website vom Partnerteam • Dabei wurde die VM Ubuntu gelöscht und eine VM CentOS aufgesetzt, da es zu Problemen kam
Tom	<ul style="list-style-type: none"> • Installation des HTML5 Video Player und entfernen des Flash Players (Flowplayer). Dabei mussten einige neue Konfigurationen durchgeführt werden • FFmpeg (snap) Troubleshooting, da es sich nicht aus der nginx.conf ausführen lässt
Marcel	<ul style="list-style-type: none"> • Recherche und Tests mit JMeter durchgeführt • Ein Record Script wurde in JMeter erstellt, um den Zugriff auf dynamische Webseiten zu simulieren • Weiters wurde ein weiteres Script mit dem HLS-Plugin erstellt, damit man auf Streams Traffic generieren kann
	14. Treffen am 18.11.2019 (10:05 Uhr – 13:00 Uhr)

Colin	<ul style="list-style-type: none"> • Versuch WordPress Seiten auf den Apache Server zu importieren
Tom	<ul style="list-style-type: none"> • Versuch ffmpeg Befehl in der NGINX Config zu fixen Dieser Befehl funktionierte nicht, deshalb musste diese immer manuell ausgeführt werden. • Recherche zu Stromverbrauch beim Übertragen von Daten über das Netzwerk
Marcel	<ul style="list-style-type: none"> • 3. Zwischenbericht schreiben und das Tool JMeter ausprobiert
15. Treffen am 19.11.2019 (10 Uhr – 16 Uhr)	
Colin	<ul style="list-style-type: none"> • Arbeiten an Apache Server -> Wordpresseiten aufgesetzt • Meeting mit Partnerprojektteam bezüglich einigen Fragen zu den Webseiten
Tom	<ul style="list-style-type: none"> • Schreiben an theoretischen Teil -> Wieviel Energie der Transport von Daten über das Internet verbraucht • Änderung der nginx.conf, nur mehr ein Stream wird als HLS versendet
Marcel	<ul style="list-style-type: none"> • Auseinandersetzung mit dem JMeter Tool und Durchführung von Messtests
	<ul style="list-style-type: none"> • Messungen durchgeführt Beim ffmpeg Befehl traten Probleme bei den Bitraten auf, somit musste die Puffergröße auf 12M gestellt werden. Nur somit konnten Bitraten von rund 6000 erreicht werden • Anpassung des Pythonscripts, da bei den Messungen die Messergebnisse um Hundertstelsekunden verzögert waren als beim Intel Power Gadget
16. Treffen am 20.11.2019 (09:20 Uhr – 12:00 Uhr)	
Colin	<ul style="list-style-type: none"> • Arbeiten an Wordpresseiten für Apache Server
Tom	<ul style="list-style-type: none"> • Versuch SELinux zu deaktivieren, um den ffmpeg Befehl in der NGINX Config ausführen zu können -> dies konnte leider nicht umgesetzt werden • Versuch QuicSync für ffmpeg einzubinden

Marcel	<ul style="list-style-type: none"> • Testen und Recherche zum Tool JMeter
	17. Treffen am 22.11.2019 (09:00 Uhr – 15:15 Uhr)
Colin	<ul style="list-style-type: none"> • Problem mit Apache Server gefixt • Mit Tool Gatling Testversuche für Loadtests gemacht
Tom	<ul style="list-style-type: none"> • JMeter Tool zum ordentlichen Einsatz gebracht
Marcel	<ul style="list-style-type: none"> • Mit Tool Gatling Testversuche für Loadtests gemacht
	18. Treffen am 25.11.2019 (13:00 Uhr – 15:35 Uhr)
Colin	<ul style="list-style-type: none"> • Auseinandersetzung mit ffmpeg
Tom	
Marcel	
	19. Treffen am 26.11.2019 (11:30 Uhr – 17:30 Uhr)
Colin Tom	<ul style="list-style-type: none"> • Gemeinsam daran gearbeitet, die Webseiten für das zweite Projektteam am NGINX Server bereitzustellen, da beim Apache Server einige Probleme entstanden
Marcel	<ul style="list-style-type: none"> • Schreiben an Dokumentation wie Infrastrukturkomponenten erklärt sowie ffmpeg usw.
	20. Treffen am 27.11.2019 (12:10 Uhr – 17:10 Uhr)
Colin	<ul style="list-style-type: none"> • Messungen mit TP-Link durchgeführt, einmal am HP Elite Desk 8300 G2 und am HP ProLiant Server G6
Tom	<ul style="list-style-type: none"> • Bei den Messungen OBS Studio bedient und den Stream gestartet • Aufsetzen einer Ubuntu VM am HP ProLiant Server und anpassen der NGINX Config • Recherche HLS encryption
Marcel	<ul style="list-style-type: none"> • Schreiben an Dokumentation (Preset erklärt) sowie den ffmpeg Befehl in der NGINX Config für die unterschiedlichen Messungen angepasst • Bedienen des IntelPowerGadgets bei den Messungen • Analyse der Messungen

	21. Treffen am 28.11.2019 (09:15 Uhr – 13:25 Uhr)
Colin Tom Marcel	<ul style="list-style-type: none"> • Messungen der dynamischen Webseite durchgeführt
	22. Treffen am 03.12.2019 (09:15 Uhr – 18:00 Uhr)
Colin	<ul style="list-style-type: none"> • Neugestaltung eines Infrastrukturplans in Draw.io • Recherche zu Secure HLS • Messungen mit JMeter durchgeführt
Tom	<ul style="list-style-type: none"> • Recherche zu Secure HLS und JMeter • Messungen vorbereitet und HLS Sampler in JMeter konfiguriert • Messungen mit JMeter durchgeführt
Marcel	<ul style="list-style-type: none"> • Schreiben an Doku • Messergebnisse analysiert und dokumentiert • Messungen mit JMeter durchgeführt
	23. Treffen am 04.12.2019 (10:00 Uhr – 12:00 Uhr)
Colin Tom Marcel	<ul style="list-style-type: none"> • Messungen durchgeführt
	24. Treffen am 06.12.2019 (10:15 Uhr – 12:00 Uhr)
Colin Tom	<ul style="list-style-type: none"> • Erneute Messungen mit dem Partnerprojektteam durchgeführt
Marcel	<ul style="list-style-type: none"> • Schreiben an Dokumentation
	25. Treffen am 09.12.2019 (14:00 Uhr – 14:45 Uhr)
Colin	<ul style="list-style-type: none"> • Meeting mit Herrn Mittenecker bezüglich Fertigstellung der Bereichsübergreifenden Projektarbeit

Tom	<ul style="list-style-type: none">• Es wurde beschlossen, wie die Enddokumentation aussehen soll bzw. wie wir unsere Messergebnisse bewerten und analysieren sollen.
Marcel	<ul style="list-style-type: none">• Vorläufigen Abgabetermin der Dokumentation den 20.12.2019 definiert
	26. Schreiben an Doku am 12.12.2019 (19:00 Uhr – 21:00 Uhr)
Marcel	<ul style="list-style-type: none">• Schreiben an Dokumentation (Meilensteintrendanalyse, Stundenerfassung, Tätigkeitsbereich)
	27. Schreiben an Doku am 14.12.2019
Tom	<ul style="list-style-type: none">• Schreiben an der Dokumentation (Jmeter, nginx, ILO)
	28. Schreiben an Doku am 19.12 (11.00 Uhr – 19:00)
Marcel	<ul style="list-style-type: none">• Finalisierung der Dokumentation
Colin	
Tom	
	29. Schreiben an Doku am 20.12.2019 (17 – 21 Uhr)
Marcel	<ul style="list-style-type: none">• Finalisierung der Dokumentation

25 Zugangsdaten

Server - Host	
HP Server – G6	
hpgreta	Pa55w.rd
Server – VM	
hpgretadebian	Password1
hpgretaubuntu	Password1
ILO2 172.17.32.10	
Desktop Client – HP Elite Desk 8300 G2	
Partition – Windows Server 2019	
Administrator	Pa55w.rd
VM1 – GretaDebian - Server	Password1
VM2 – GretaUbuntu - Server	Password1
VM3 – GretaUbuntu2 - Server	Password1
Desktop Client – HP Elite Desk 800 G1	
Greta	Password1
Desktop Client – HP Elite Desk 8300	
Greta	Password1
Access Point	
OpenWrt	apple123
Youtube Clon	
Maria DB	Password1

26 IP Konzept

Host	IPs
HPGreta – Server	192.168.47.14
HPGretaDebian - VM	192.168.47.24
HPGretaUbuntu - VM2	192.168.47.34
G2-HostGreta (HP Elite Desk 8300)	192.168.47.13
G2-VMDebian	192.168.47.23
G2-VMUbuntu	192.168.47.63
G2-VMUbuntu2	192.168.47.43
G1-Client (HP Elite Desk 800)	192.168.47.12
Client (HP Elite Desk 8300)	192.168.47.22
ILO Schnittstelle	192.168.47.10
Test Clients	192.168.47.XX
TP-Link HS110 – 1	192.168.47.1
TP-Link HS110 – 2	192.168.47.2
TP-Link HS110 – 3	192.168.47.3
TP-Link HS110 – 4	192.168.47.4
DNS-Server	10.25.1.6
LINKSYS Router - Gateway	192.168.47.254
Subnetzmaske	255.255.255.0

27 Abbildungsverzeichnis

Abbildung 1: Der Globale Internet Traffic besteht zu rund 80% aus Videoverkehr. Livestreaming macht etwa 20% davon aus.	10
Abbildung 2: Übersichtskarte Erneuerbare Energien und Rechenzentren	11
Abbildung 3: Um verschiedene Studien vergleichbar zu machen, wurde der Systemumfang auf das Transportnetzwerk beschränkt (IP core network, Access networks). Die Studie, welche für die Illustration des Netzwerkverbrauchs genutzt wird inkludiert ebenfalls den Stromverbrauch von Unterseekabeln.....	12
Abbildung 4: Der Stromverbrauch in kWh/GB für Datenzentren, Transportnetzwerke und End User.	13
Abbildung 5: Infrastrukturumgebung	14
Abbildung 6: Market share of all sites	19
Abbildung 7: Seit Version 1.9.11 unterstützt NGINX einen modularen Aufbau.	23
Abbildung 8: Kodierungszeitbeispiel eines 1080p-Videos.....	24
Abbildung 9: Messtools	26
Abbildung 10: Überblick Intel Powergadget Tool	27
Abbildung 11: JMeter Einsatz	30
Abbildung 12: Beim HLS Sampler geben wir den Pfad zum Playlist File. Die Video dauer wurde hier auf 60 Sekunden gesetzt.	31
Abbildung 13: Der Fertige testplan für die Simulation von user Zugriffen auf den HLS Stream.	31
Abbildung 14: JMeter Arbeitsfluss.....	32
Abbildung 15: Im Listener "view result in table" werden die HTTP responses des Servers aufgelistet	32
Abbildung 16: XAMPP Schalttafel	33
Abbildung 17: XCP-ng Center.....	34

28 Messdiagrammverzeichnis

Messdiagramm 1.....	36
Messdiagramm 2.....	37
Messdiagramm 3.....	37
Messdiagramm 4.....	38
Messdiagramm 5.....	39
Messdiagramm 6.....	40
Messdiagramm 7.....	41
Messdiagramm 8.....	41
Messdiagramm 9.....	42
Messdiagramm 10.....	42
Messdiagramm 11.....	43
Messdiagramm 12.....	43
Messdiagramm 13.....	44
Messdiagramm 14.....	44
Messdiagramm 15.....	45
Messdiagramm 16.....	45
Messdiagramm 17.....	46
Messdiagramm 18.....	46
Messdiagramm 19.....	47
Messdiagramm 20.....	47
Messdiagramm 21.....	48
Messdiagramm 22.....	48
Messdiagramm 23.....	49
Messdiagramm 24.....	51
Messdiagramm 25.....	51
Messdiagramm 26.....	52
Messdiagramm 27.....	52
Messdiagramm 28.....	53
Messdiagramm 29.....	54
Messdiagramm 30.....	56
Messdiagramm 31.....	56
Messdiagramm 32.....	57
Messdiagramm 33.....	57
Messdiagramm 34.....	58
Messdiagramm 35.....	59
Messdiagramm 36.....	59
Messdiagramm 37.....	60
Messdiagramm 38.....	60
Messdiagramm 39.....	62
Messdiagramm 40.....	62
Messdiagramm 41.....	63
Messdiagramm 42.....	64

Messdiagramm 43.....	64
Messdiagramm 44.....	65
Messdiagramm 45.....	66
Messdiagramm 46.....	66
Messdiagramm 47.....	67
Messdiagramm 48.....	67
Messdiagramm 49.....	68
Messdiagramm 50.....	69
Messdiagramm 51.....	69
Messdiagramm 52.....	70
Messdiagramm 53.....	71
Messdiagramm 54.....	71
Messdiagramm 55.....	72
Messdiagramm 56.....	72

29 Tabellenverzeichnis

Tabelle 1: HP Elite Desk 800 G1 TWR - Komponenten und Spezifikationen	15
Tabelle 2: HP Elite Desk 800 G2 TWR - Komponenten und Spezifikationen	15
Tabelle 3: HP-Server ProLiant DL380 - Komponenten und Spezifikationen	16
Tabelle 4: FFmpeg Parameter	25
Tabelle 5: TP-Link HS110 Parameter.....	29
Tabelle 6: HP Elite Desk 8300 G2 – Messungen.....	36
Tabelle 7: HP Server ProLiant G6	50
Tabelle 8: Belastungstest - HP Elite Desk 8300 G2	61
Tabelle 9: JMeter - HP Server ProLiant G6.....	68