Scalable Video-on-Demand With Edge Resources

Alagu Sanjana Haribhaskaran

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2016-96
http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-96.html

May 13, 2016

Acknowledgement

Professor Kannan Ramachandran
Kangwook Lee (graduate student advisor)

Capstone Report

Scalable Video-on-Demand With Edge Resources

Alagu Sanjana Haribhaskaran

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2016

Electrical Engineering and Computer Sciences

Signal Processing and Communications

Scalable Video-on-Demand with Edge Resources

Alagu Sanjana Haribhaskaran

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: ________________ Date 5/5/16

Print Name/Department: Kannan Ramchandran/EECS

2. Faculty Committee Member #2:

Signature: ________________ Date 5/5/16

Print Name/Department: Laura Waller/EECS

Table of Contents

Chapter 1: Technical Contributions

Chapter 2: Engineering Leadership

Appendix

Chapter 1

Technical Contributions

1 Introduction:

Scalable Video-on-Demand (VoD) with edge resources is an interactive VoD system that aims to deliver high quality content at an improved speed and at a reduced load on the central server. The VoD industry is growing at an exponential rate and its current leading players like Netflix, Hulu and Amazon use centralized server architecture, which is very expensive and consumes a lot of bandwidth and storage. A solution to this problem is our scalable VoD system with hybrid peer-to-peer architecture. This system exploits the resources available on the user's edge devices, thus reducing the bandwidth usage on the central server and making the system faster and cheaper. This system provides the user with an affordable video streaming platform, thus with this product we can either compete in the current VoD market or provide VoD in regions like the rural parts of India, China and Africa which have limited connectivity and poor infrastructure.

This project also known as the CalVoD project had been initiated a few years ago at University of California, Berkeley and at present has fully developed theory and algorithms. Our graduate student advisor, Kangwook Lee, has built a basic prototype of this system under the guidance of Professor Kannan Ramachandran. Their work has been described in detail in these two papers: "*An Optimized Distributed Video-on-Demand Streaming System: Theory and Design*", "*A VoD System for Massively Scaled, Heterogeneous Environments: Design and Implementation*". In these papers, a general framework of the distributed VoD content is presented. They solved a static optimization problem for the network bandwidth and storage capacity constraints and demonstrated the robustness of the system (Ramchandran). We were required to resume the work on this project and successfully build an end product.

2 System Architecture:

The key concept of this system is to cache the popular videos, that is, store chunks of a video in user edge devices like laptops, tablets, so that the user does not have to rely on the central server and instead, get the data from these caches (Lee, 2012). This reduces Internet traffic and improves the speed and quality of videos.

In this system, we have four main components: tracker, server, caches and users.

Tracker:

The tracker acts like a database to this system, it keeps track of the caches and users running in the system. It is built on Webpy framework. The tracker also records incentive currency points accumulated by the viewer and provides real time data visualizations.

Server:

The server sends chunks of the video file to the caches and users using FTP protocol. It is built using open source FTP library. The server interacts with the tracker to know which chunk of video file has to be sent to the caches and users.

Caches:

The caches download chunks of the video file from the server, store them and send them to the users upon request from tracker and the viewer accumulates currency points for running caches.

Users:

The user downloads a video file chunk by chunk from the caches and server. The user uses the currency points to purchase a new video.

Dataflow:

The tracker has a list of videos, keeps track of the server connection and information about caches, like IP address, and content of the chunks. First, a cache registers itself in the tracker and runs the resource allocation algorithm. This algorithm updates the chunks in the cache and the upload rates to the connected users. The user is initially provided with a list of videos to choose from. When the user chooses a video, it registers itself with the tracker and connects to the server. The user receives the first video frame from the central server to start watching the video immediately. During this time, the user downloads chunks of the next frame from the connected caches and this repeats until the user finishes watching the video (Lee, 2013).

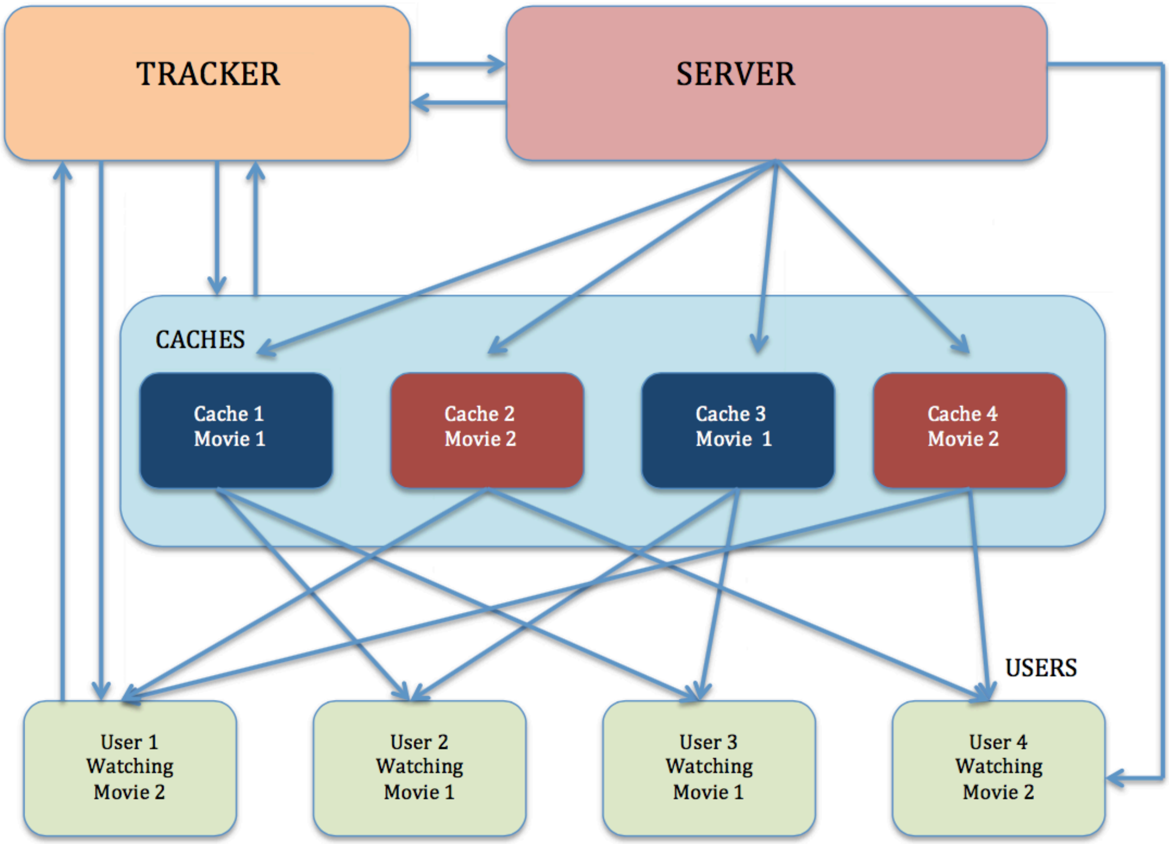

Fig.1: System Architecture

3 List of Associated tasks:

The CalVoD project has been handed over to my capstone team to develop a working scalable VoD system, that is a fully integrated application that delivers video content in a highly scalable, robust and distributed way in a period of one academic year. Kangwook Lee provided us with the source code of this system that he had worked on previously. In the previous semester, we started this project by understanding the already developed system and perusing the theory and algorithms used. First we cleaned the code and wrote a clean documentation for the existing work. We ran the system across multiple machines and then experimented with Amazon Web Services to test the scalability of the system. We worked to optimize the system and make it compatible on different environments. This semester, we resumed work by addressing and resolving the scalability problems with AWS and robustness issues we faced earlier. We implemented a detailed user interface with an integrated video player, developed a web and desktop application with username/password feature and employed an incentive mechanism for this VoD system.

3.1 Work breakdown structure:

In this chapter, I will focus on my contribution to this capstone project, starting with writing a clean documentation, running the system on multiple machines, testing the scalability of the system using Amazon Web Services (AWS), implementing a detailed user interface for this system and lastly, developing a desktop application with username/password feature, whereas Ryan Kashi's paper concentrates on choosing the appropriate video format, erasure coding, video chunking and encoding, data visualization, addressing problems with system's robustness, and implementation of incentive mechanisms for this system. Jiayuan Chen's paper discusses the system topology, writing the shell scripts and file management. He also writes about implementation of tracker's database, tracker's webpy server, FTP and pyftpdlib libraries, and also steps taken to resolve the issues faced with AWS. The three reports together give a detailed picture on the work done over the last two semesters and challenges faced in our capstone project. They also discuss the future work that can improve the performance of the system.

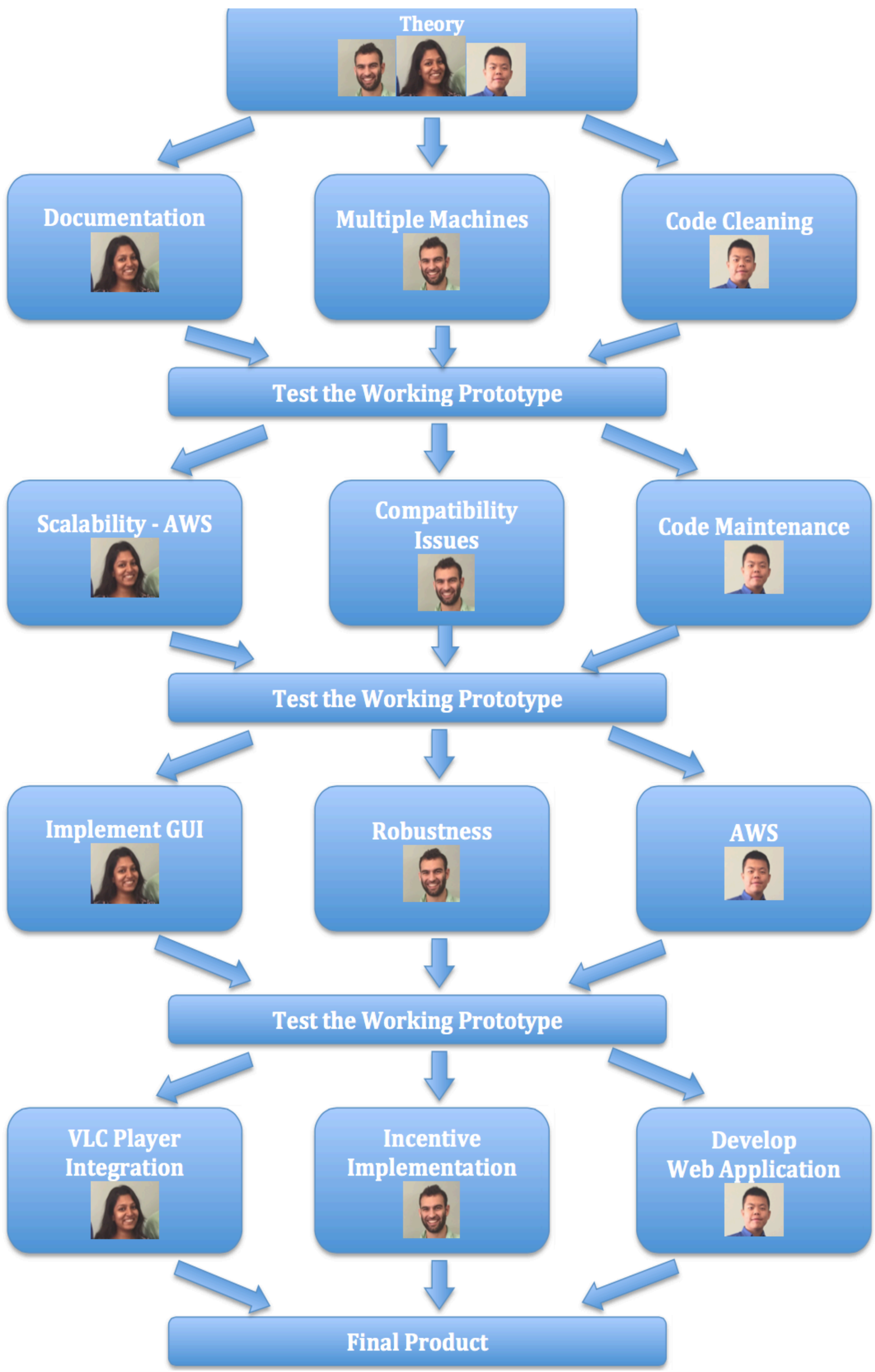

Fig. 2: Work breakdown structure

4 List of Accomplished tasks:

Over the last one year, my capstone team has put in a lot of hard work and has made remarkable progress in developing this system. In this section, I have listed my contributions in building this scalable VoD system during the course of this project.

4.1 Documentation:

The first task I had to deal with was to write a clean documentation. Since we were resuming work on an existing project, it was crucial to understand the work that has already been done. After comprehending the current work, I updated the wiki pages on Github, where all our documentation currently exists. I started the documentation by explaining the overall view of the system. Then, I provided a detailed description for the dependencies that have to be installed to run the system on the local machine. Finally, there is a clear procedure on how to run the system. These wiki pages are updated as the system is being modified. This documentation has been written in such a way that a third party will be able to run the system successfully. A comprehensible documentation is crucial for any project and also provides a clear picture on which direction to take the project.

4.2 Multiple Machines:

In the existing working prototype, we can run 1-server, 1-tracker and 1-cache, 1-user locally on my laptop, but cannot connect to other machines. We used Web.py library to build the system so that it is compatible with both the IPv4 and IPv6 addresses and a connection could be established between two or more machines. Thus, by inputting the right IP address of the machine, we could run the system on more than one machine. Now, the server and tracker can be run publically on one machine and the user and cache can be run on another machine. At this stage, we ran up to five users and ten caches. Thus, the VoD system was successfully implemented across multiple machines.

4.3 Scalability:

Until now we discussed working of the VoD system on a few local machines, but did not deal with scalability issue. We were able to successfully run the system with a maximum of five users and ten caches. However, in real-time there are thousands of users who use this system. We had to test the scalability of this system. This could be done using cloud computing, which is an on-demand service that gives access to servers, storage, and databases. It provides on-demand access to a shared pool of computing resources. AWS is one of the most popular cloud computing service providers that have their own network-connected hardware needed for these services. By general definition, AWS is a paid service for on-demand delivery of IT resources and applications through the Internet.

Since I was new to the AWS platform, I had to learn and become familiar with it before implementing the VoD system on it. AWS provides a variety of instances, among which I chose the EC2 Linux instance since it was one of the free tier instances available for duration of one year. I launched and connected to the instance using the private key provided. The Linux instance faced a few minor compatibility issues with the system.

The security groups had to be changed and it required different dependencies to be installed to run the system. I was successfully able to run 1 server, 1 tracker and 1 cache, 1 user and 1 tracker, 1 server, N caches and N users locally in that instance, but I was not successful in running the tracker and server on the EC2 instance and the user and cache on my local machine. There were several connectivity issues and at this point, I came to the conclusion that the IP of the EC2 instances cannot be accessed the same way as local machines due to the security implemented by AWS. Testing the scalability of the system is one of the most important tasks of this project. This connectivity issue faced with the AWS instances was a major drawback to this project and caused delay in the progress of the project. Jiayuan Chen took over the AWS issue, he decided to address and resolve this problem, and he discusses the final implementation of AWS in his report.

4.4 User Interface:

I started working on the next key aspect of this project, implementation of Graphic User Interface (GUI) for this VoD system. So far, we have been using the Command Line Interface (CLI) to run the system, but this is not user-friendly and is a tedious process. The CalVoD system needs a detailed and attractive GUI for the user to select a video of his choice and enter other specifications.

Good user interface design can decide the success or failure of a product. Earlier, we first ran the tracker and server followed by caches and users on the CLI and this required the user to have certain expertise with the CLI. The cache specifications like port number, IP address, cache size had to be entered by the user and after which, a list of videos is displayed to the user, he/she had to enter the video of his choice on the CLI. This starts downloading the chunks of the chosen video in the user's folder, which can be manually run on the video player. Most of these tasks have to be automated in order to create a user-friendly experience, because a viewer of the VoD system should not require any knowledge about system's design and implementation.

To design a good GUI, I used the TKinter Library in python. In the initial basic GUI I created, a window appears when we run the caches, where we can enter the cache specifications. Then, another window appears when we run the user, which has a drop down menu that consists of the list of videos available, and we can choose the video of choice. After successful implementation of basic GUI, we worked on implementing the username/password feature for the viewer. The end VoD product has be user-friendly with an elegant user interface design that balances both aesthetics and interactivity.

4.5 VLC Player Integration:

A crucial step in GUI implementation is to integrate a video player with this system, so that when the user chooses a video, the video player automatically starts playing the video. I worked on integrating the VLC player, which has an open source code and can be embedded with the CalVoD system. The python bindings provided by VLC player was an added advantage of using this video player, since rest of the project are written in python. However, the library versions released by VLC were not stable and updated frequently, making the older versions obsolete. Also the VLC player had compatibility issues with the Mac operating system, due to the time constraints of the project, I addressed this issue by switching to the Ubuntu operating system instead of using another video payer. Moreover, the Mac operating system can be used after the VLC developers fix the compatibility issue between Mac and VLC library. In the current system, the user can choose a video of his/her choice and the video starts playing automatically with the integrated VLC player.

4.6 Desktop Application:

Now, we have successfully developed a working VoD system, where the server and tracker are run on the AWS and users and caches are run on the local machine. I had to integrate the user interface with the rest of the VoD system on AWS, to develop a complete desktop application. When the user opens the desktop application, a *Home* page appears, where the user inputs his/her credentials that is the username and password. A HTTP request is sent to the tracker on AWS for verification of the login credentials. If the username and password match, then the user is taken to the next *Cache Settings* page, where he/she is asked how much of his/her edge device's resources is he/her willing to contribute to run the caches and given three options "Minimum", "Half", "Maximum". Depending on his/her choice, a corresponding multiplier that is 0.3 for "Minimum", 0.5 for "Half" and 1 for "Maximum" is applied to the storage and bandwidth to run the caches on his/her edge device. Then, the user is taken to *Movies* page, where a list of movies available is displayed to him/her.

This movie list is obtained by sending a request to the tracker, the list of movie names is given to a movie poster API, to return the corresponding movie posters to be displayed to the user and once the user chooses a movie to watch, the movie starts to download frame by frame on the user's edge device. After the first frame is downloaded, the video starts playing on the VLC player. Screenshots of the pages of the desktop application can be found in the appendix.

5 Future Work:

The VoD system that we have built has a lot of scope for improvements in future for better performance. Due to time constraints, we could not fully implement all our ideas and a list of suggested improvements is as follows:

- The user interface of the current system is built using the Tkinter library. A lot of design choices cannot be implemented using this library. We can explore other python libraries that can be used to develop a better user interface.
- Also, we switched to Ubuntu from Mac OS because the VLC library was not compatible with the Mac OS. This issue has to be resolved so that the VoD system is compatible with all the platforms.
- We have not dealt with any of the security issues so far, but in order to make it a complete product the security issues have to be addressed. At present, the viewer can access the video files downloaded to his/her edge device and copy the video files outside the application. This will interfere with the copyright issues of the video. Therefore, the video files downloaded to his/her edge device should be encrypted so that the user cannot access it.
- A lot of work has to be done to integrate all aspects of the project and release a final package that can be downloaded by the user and installed in his/her edge

device. The current system requires downloading the code and installing the dependencies required to run the system manually.

- Also, in the current system due to limited AWS resources, the server and tracker are run on a single AWS EC2 instance. To scale up the system, a better infrastructure is required that has more power, storage and bandwidth.
- At present, if a viewer wants to purchase a video, he/she has to use his incentive currency points that he/she accumulated by running caches. The viewer should be given the option to purchase a video using money.

6 Conclusion:

My capstone team has made commendable progress in developing this system over the course of this project. We started working on this project by putting in our full effort from the beginning of the academic year and overcame several problems by taking the right decisions. A detailed plan was executed for each deliverable and there was a clear division of work among the team members. The team saw drastic improvements throughout the year and reached all the milestones initially planned, even though small modifications were made to the plan from time to time.

Working on this capstone project has improved my programming skills and deepened my understanding of the VoD system's architecture. Also, it has given me a valuable experience of working with this team. This system has appropriate documentation so that a third party can easily resume work in improving this VoD and successfully deliver a commercially viable video on demand system. This report along with Ryan's and Jiayuan's report summarizes the accomplishments and challenges faced by the team working on this capstone project.

Works Cited

Lee, Kangwook, Hao Zhang, Ziyu Shao, Minghua Chen, Abhay Parekh, and Kannan Ramchandran. "An Optimized Distributed Video-on-demand Streaming System: Theory and Design." *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (2012): n. pag. Web.

Lee, Kangwook, Lisa Yan, Abhay Parekh, and Kannan Ramchandran. "A VoD System for Massively Scaled, Heterogeneous Environments: Design and Implementation." *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems* (2013): n. pag. Web.

Ramchandran, Kannan. "Scalable Content Delivery." *Scalable Content Delivery*. EECS Berkeley, n.d. Web. 13 Mar. 2016. <https://www.eecs.berkeley.edu/~kannanr/project_cdn.html>.

Chapter 2
Engineering Leadership
(Team written)

1 Introduction

Among all engineering leadership topics and tools we learnt over the last year, we chose the three most relevant modules to our system. This paper will analyze the current industry of video content delivery, market research that can validate the commercial success of our product, and the management of the software intellectual property.

2 Industrial Analysis

As of 2014, broadband internet is nearly ubiquitous around the world with the "global average connection speed exceeding the 4 Mbps broadband threshold" (Young 2014) according to a report generated from the Akamai Intelligent Platform. With the increasing availability of high speed internet and internet connected devices, the delivery of entertainment video directly over the internet has become a convenient solution for consumers. Video streaming services have increasingly been accepted by consumers as the alternative to multichannel video programming distributors (MVPD) such as satellite and cable television systems and physical media rental services.

Our platform is backed behind a novel heuristic that reduces bandwidth costs on a centralized server to potentially nothing, which gives our system an edge over current video streaming services such as Netflix, Amazon Video, and Hulu. Despite this, it is important to observe and analyze the current competitors of this industry to see what could be done for us to successfully enter this line of business.

In the realm of entertainment video delivery over the Internet, Netflix dominates and controls the market. Netflix members are able to enjoy streaming media over their televisions, mobile devices, and computers. Members can play and pause content at their leisure without commercials, commitments, or restrictions. Netflix, Inc. is a membership service, but members have unlimited access to a pool of available content during their time of membership. As of July 2011, Netflix, Inc. has increased its emphasis on the removal of DVD based entertainment, focusing more on its robust video streaming platform,

An advantage Netflix, Inc. has over the market for entertainment video delivery is its emphasis on making its service compatible with many devices while maintaining streaming functionality. The company holds partnerships with "satellite and telecommunications operators to make their service available through the television set-up boxes of these service providers," (Netflix 10-K 2014).

Netflix, Inc. "strives for consumers to choose (Netflix) in their moment of free time," (Netflix 10-K 2014). Netflix, Inc. refers to this part of their company's objective as the "winning moments of truth" (Netflix 10-K 2014) and places emphasis on their "recommendation and merchandising technology to predict and recommend titles for members to enjoy," (Netflix 10-K 2014). In addition, Netflix has begun rolling out its own Content Delivery Network (CDN) called Open Connect that will reduce cost and improve the delivery of the service's content by reducing network overhead (Netflix 2014). Because Netflix is a large percentage of the traffic ISPs deliver to their end-users, Netflix, Inc. has offered a solution to peer directly with Netflix. CDN, however, will only provide a reduction in latency for users and has no effect on the total amount of data sent from Netflix servers to its customers. Thus, despite their claims, our platform has the capability of providing customers with the same video at a greatly reduced price.

A concern and risk factor for Netflix, Inc. (Netflix 10-K 2014) is the retention and attraction of members to the service. Consumers need to seek value with the service which can be primarily found in the speedy delivery of content as well as "compelling content choices, as well quality experience for selecting and viewing TV shows and movies," (Netflix 10-K 2014).

3 Market Research

Over the last two decades, there has been an exponential increase in the interest shown in watching. This indicates that the VoD system has a huge market. The current leading competitors in the VoD market are Netflix, Hulu, and Amazon Prime Video. The revenue generated by these companies indicate consumer's inclination towards the video streaming service is increasing and this technology is slowly replacing traditional satellite and cable television systems.

The core idea of success behind our VoD system is based on our conclusion from market research that by caching the few popular videos that are on high demand we can significantly reduce the bandwidth of the system, while delivering content to users at an improved speed. According to a Video statistics and Market research by Tubular, a video intelligence software, only 5% of videos uploaded to YouTube drive most of the views (Marshall 2015b). Out of 1.1 billion videos on YouTube, only 58.6 million have more than 10,000 views and these videos alone have generated 7.4 trillion views (Marshall 2015b).

Anyone who owns an internet connected device is a potential customer to our product. The Interactive Advertising Bureau, a nonprofit organization, states that there has been an increase of 35% in views collected by people watching videos on their smart devices in the past year and also one third of the people surveyed said that they watch at least one video lasting five minutes every day (Marshall 2015a). Smartphones with bigger screen, improved display, memory, and battery life indicate that the technological advancements in edge devices go hand in hand with these social trends.

Cisco predicts that by 2018, 84% of the consumer internet traffic will come from video (Zccone 2014). Consumers are streaming more video content than ever before, and Internet Service Providers are not able to maintain sufficient bandwidth required to enable constant high quality video streaming. Being well aware of this problem, Verizon and Comcast have made special agreements with the heaviest traffic producing consumers that, for an additional cost, the traffic from these customers will be handled at a higher priority (Grill 2015).

The technological innovations like ‘capture camera’ and ‘retina display TV with UHD 4k videos’ worsen the problem, as the backend is not able to keep up with the user demands. Reducing video bitrates is not a solution, as it will only lead to poor viewing experience and lose market value (Grill 2015). Our system comes up with a feasible solution to this problem. If we cache the popular videos on edge devices, the user would not have to get the data from the central server, thus reducing internet traffic and improving the performance in terms of speed and quality.

This project can be made into an open source project contributing to technological evolution. An ideal marketing strategy for this system would be to develop a final product with a good user interface, and advertise it as a commercial product. Our system, can provide high quality video at a cheaper price, making it easier to sell our product and replace existing VoD companies. Our market research indicates potential success of our product and the ability to revolutionize the VoD market.

4 Intellectual Property (IP)

The core technology of this project is the VoD architecture, including the concepts of using caches as edge resources, algorithms ruling the data flow, and methods of chunking videos into parts. Several related papers have been published (Lee 2012) (Lee 2013), and a working prototype has been built on top of them. We believe that this system architecture is patentable, and holding such patent could bring numerous advantages to the product in business.

First, a patent enables us to provide the stable and fast VoD experience to customers in an exclusive way. It will maintains the differentiated value from the technology we developed. Second, protection of the core technology allows the ongoing product development to focus on the product itself, without worrying about similar services being released by competitors. Last but not least, a patent can add value to our project and thus make it more likely that another company will buy our technology. Typically, filing a utility patent in U.S. will cost around $20,000 (Quinn 2016). Nevertheless, this is a small investment compared to the future return, as current market competitors are generating approximately 1 billion dollars of revenue (MarketsAndMarkets 2016).

On the other hand, there are difficulties in protecting this IP as a patent. This market has become extremely competitive in recent years. Companies such as Rovi, AT&T, and Microsoft hold large patent portfolios. Moreover, software patents are hard to define. As a result, improperly defined IPs can lead to lawsuits if filed without prior research. For example, in 2013, the author of the patent "Video-on-demand systems (Nomura 2007)" sued Youtube and Amazon’s CloudFront services of infringing his entire patent (Randles 2016). Also, the product’s bandwidth and resource-saving advantages might be invalidated by network technology advancement, as the network resources will not be an issue any more.

Alternative solutions to protect our IP include transforming it into a trade secret, sell the project to existing VoD vendors, or make it an open source project. Our project can be a valuable addition to the existing VoD platforms as an upgrade to their original services.

5 Conclusion

Analysis of the industry landscape, market research, and IP have helped us to understand the potential of the product, as well as a clearer idea of how to commercialize it. Tools learnt from the engineering leadership courses have played a crucial role our analysis will be utilized in future stages of development.

Works Cited

Gill, Dror. "As The Internet Grows Bigger, What Do We Do About Video?" *ReelSEO*. Reelseo & Tubular Labs, Inc., 15 Jan. 2015. Web. 05 Mar. 2016. <http://www.reelseo.com/internet-bigger-video/>.

Lee, Kangwook., Hao Zhang, Ziyu Shao, Minghua Chen, Abhay Parekh, and Kannan Ramchandran. "An Optimized Distributed Video-on-demand Streaming System: Theory and Design." *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*(2012): n. pag. Web.

Lee, Kangwook., Yan Lisa, Parekh Abhay, and Kannan Ramchandran. "A VoD System for Massively Scaled, Heterogeneous Environments: Design and Implementation." *2013 IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems* (2013): n. pag. Web.

MarketAndMarket "Video on Demand (VOD) Market worth $61.40 Billion by 2019." *Video on Demand (VOD) Market worth $61.40 Billion by 2019*. Marketsandmarkets, Feb. 2015. Web. 05 Mar. 2016.

Marshall, Carla. "We're All Watching More Video on Mobile Devices [Study]." *ReelSEO*. Reelseo & Tubular Labs, Inc., 10 June 2015a. Web. 05 Mar. 2016. <http://www.reelseo.com/increase-mobile-video-consumption/>.

Marshall, Carla. "5% of Videos on YouTube Drive 95% of the Views." *ReelSEO*. Reelseo & Tubular Labs, Inc., 04 Sept. 2015b. Web. 05 Mar. 2016. <http://www.reelseo.com/5-percent-youtube-videos-drive-95-percent-views/>.

Netflix Inc. 2014 10-K Report. Netflix Inc. 03 Feb. 2014. Web. 06 Mar. 2016

Netflix. *OpenConnect-Deployment-Guide.pdf*. N.p.: Netflix, 2014.*Oc.nflxvideo.net*. Netflix, 2014. Web. 06 Mar. 2016. <http://oc.nflxvideo.net/docs/OpenConnect-Deployment-Guide.pdf>

Nomura, Tetsuya. Video-on-demand System. Tetsuya Nomura, Tommy Sun, assignee. Patent 7254622. 7 Aug. 2007. Print.

Randles, Jonathan. "YouTube, Amazon Escape Video On Demand Patent Suits - Law360." *YouTube, Amazon Escape Video On Demand Patent Suits - Law360*. Law360, 13 Sept. 2013. Web. 05 Mar. 2016. <http://www.law360.com/articles/472533/youtube-amazon-escape-video-on-demand-patent-suits>

Quinn, Gene. "The Cost of Obtaining a Patent in the US - IPWatchdog.com | Patents & Patent Law." *IPWatchdogcom Patents Patent Law The Cost of Obtaining a Patent in the US Comments*. IPWatchdog, 04 Apr. 2015. Web. 05 Mar. 2016. <http://www.ipwatchdog.com/2015/04/04/the-cost-of-obtaining-a-patent-in-the-us/>

Young, Jeff. "Akamai Releases Second Quarter 2014 'State of the Internet' Report."

Akamai. Akamai, 30 Sept. 2014. Web. 06 Mar. 2016.

Zaccone, Emanuela. "Why Video Consumption and Instant Messaging Rule Social Trends." *Medium*. N.p., 29 Jan. 2015. Web. 05 Mar. 2016. <https://medium.com/@zatomas/why-video-consumption-and-instant-messaging-rule-social-trends-5d86e51e4355#.rfvko98rx>.

Appendix

Figure i. Home Page

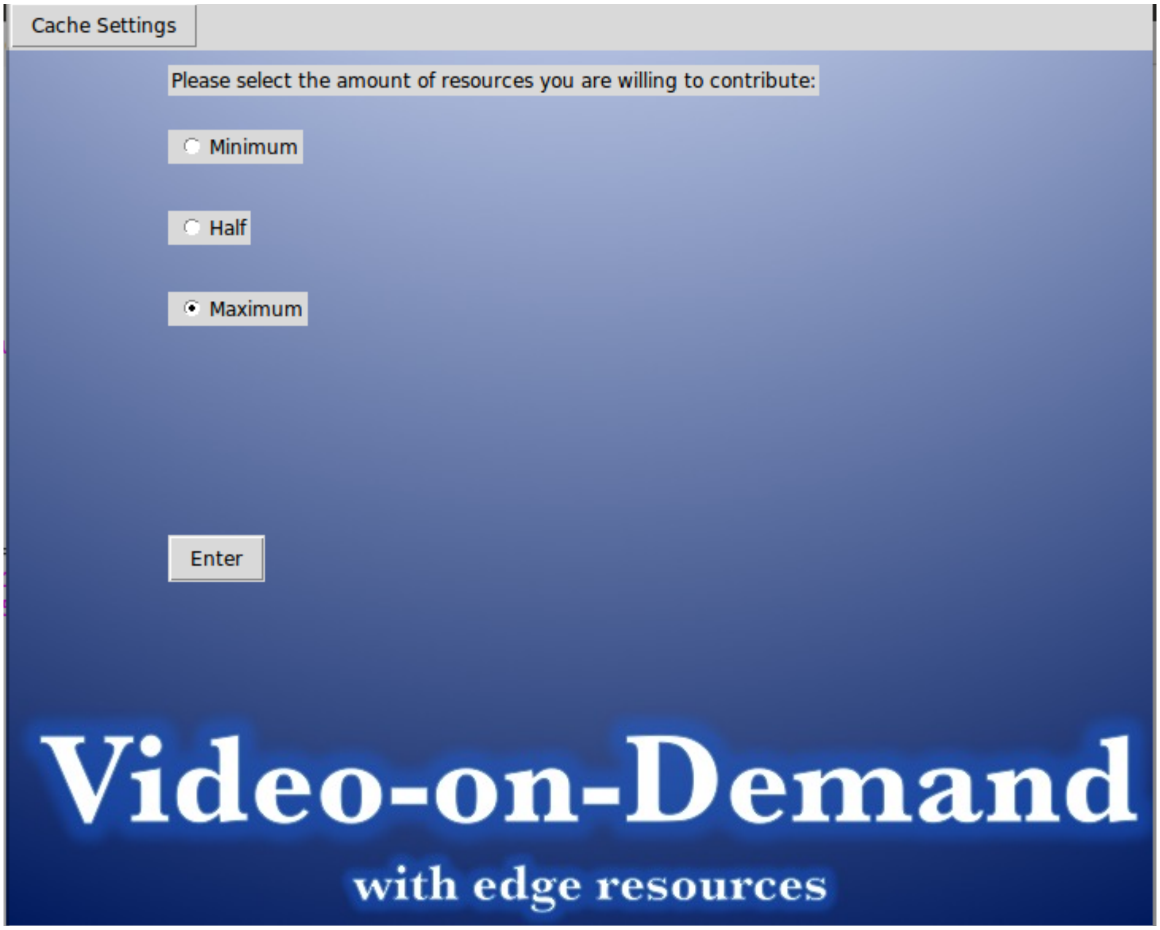

Figure ii. Cache Settings Page

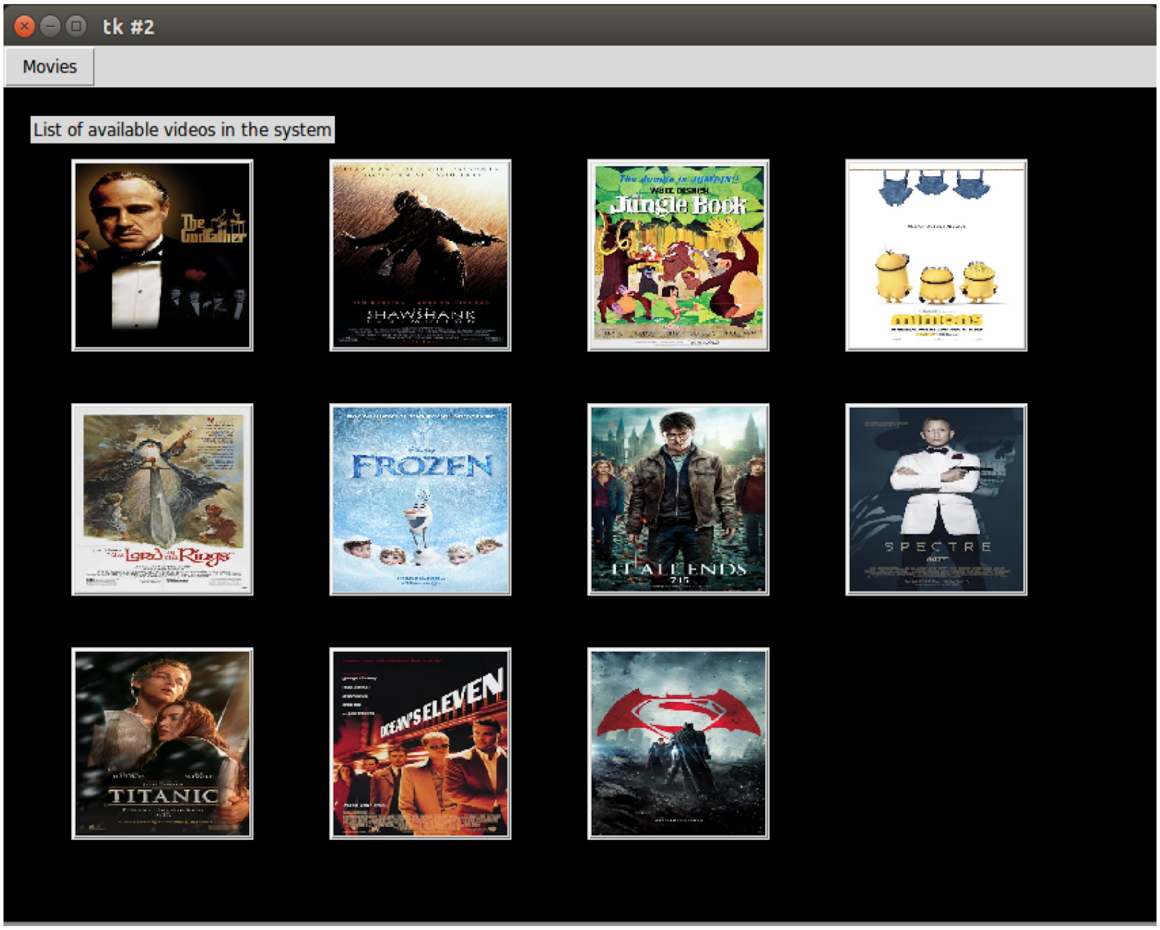

Figure iii. Movies Page

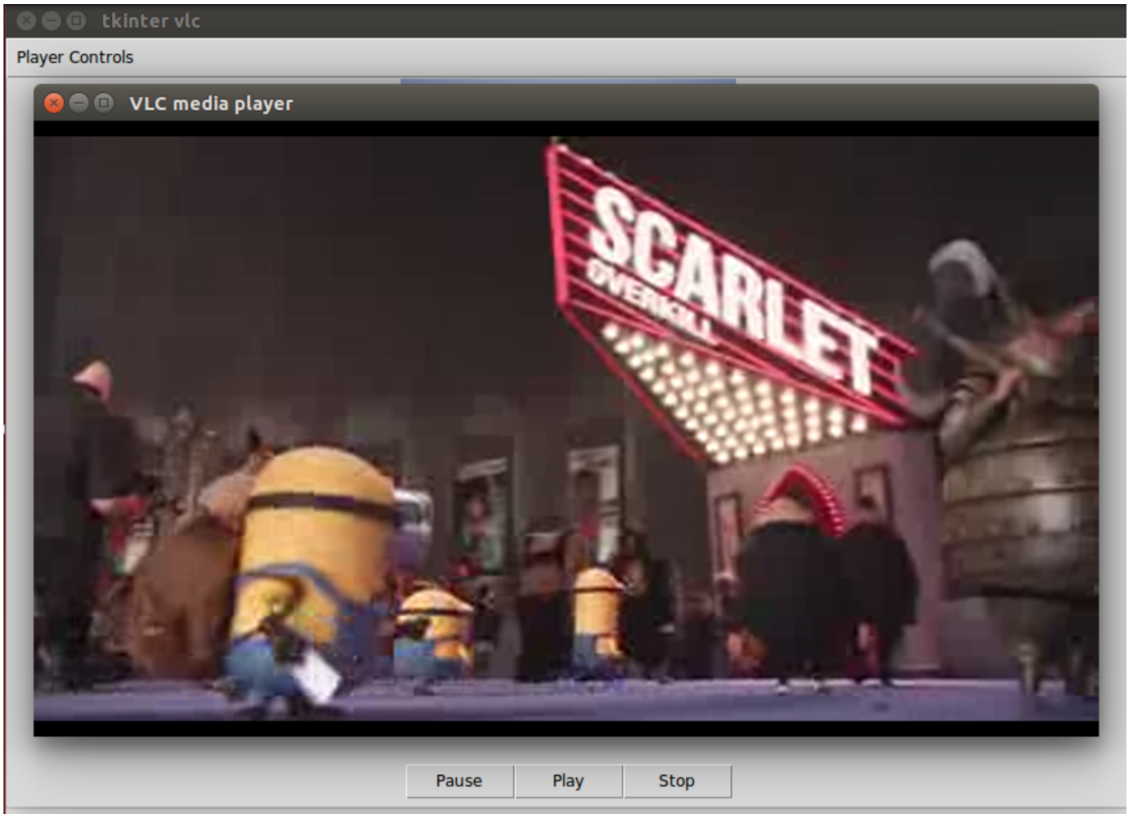

Figure iv. VLC media player playing a movie

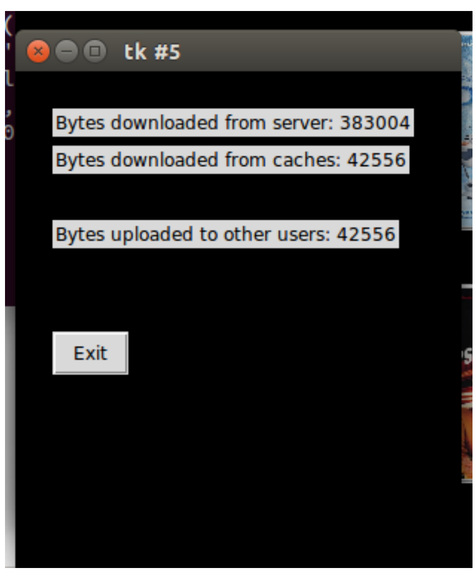

Figure v. User data display