BRAC UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

14-12-2016

"Distributed Video Management System using the concept of Spanner"

Azima Islam Fariha (ID – 13301046)

BRAC University

SUPERVISOR: Moin Mostakim

CO-SUPERVISOR: Suraiya Tairin

Distributed Video Management System using the concept of Spanner

By

AZIMA ISLAM FARIHA

A report
submitted in the partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science & Engineering of
BRAC University

December 2016

BRAC UNIVERSITY

FINAL READING APPROVAL

Thesis Title: Distributed Video Management System using the concept of Spanner

Date of Final Presentation: 14th December, 2016

The final reading approval of the thesis is granted by Moin Mostakim, Supervisor.

Supervisor **Signature of Author**

______________________ ________________________

Moin Motakim Azima Islam Fariha

Lecturer,

Department of Computer Science & Engineering,

BRAC University

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my supervisor Moin Mostakim to be continuously supportive throughout the entire thesis. He was always available whenever I needed any help. He has not only guided me but also encouraged and helped me to come up with my own creative and innovative ideas.

Secondly, I would like to thank my co-supervisor Suraiya Tairin to always inspire me.

Last but not the least, I would like to acknowledge my friends and family members for their continuous support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THESIS OUTLINE

The thesis consists of eight chapters and each chapter has a number of sections and -subsections. The complete paper is outlined below for the ease of the reader.

Chapter One describes the main purpose, motivation and objectives of the paper. It has three sections.
They are *Background, Motivation & Objectives*.

Chapter Two describes the current problem with which I am dealing with in the paper and the solution I want to propose. It has two sections.
They are *Problem Description & Proposed Solution*.

Chapter Three gives basic information about Video Management System and the works that is previously being done. It has two sections.
They are *Video Management System & Related Works.*

Chapter Four describes briefly about the data management system which I want to use in my thesis with justifications. It has three sections.
They are *Spanner, System Architecture & Why Spanner is Better*.

Chapter Five provides a study that compares Spanner with other video management systems. After comparing, the final proposed solution is being briefly described in this chapter. It has three main sections. The first section has three sub-sections.

Chapter Six describes the proposed system with necessary diagrams and flow chart. It has two sections.
They are *Proposed System & System Flowchart*.

Chapter Seven describes the output of the proposed system and also its graphical representation. It has two sections.
They are *Results & Graph*.

Chapter Eight includes the author's concluding notes for this thesis and contains two sections.
They are *Limitations & Future Works*.

ABSTRACT

In today's world, we are highly dependent on modern technology. Everyday millions of people are watching videos online for recreation, learning and working purpose. With every passing day, this number of viewers are increasing enormously. Thus, the importance of video management has increased to a great extent. The traditional way of managing videos was centralized and time consuming. Nowadays, different platform offers different storage systems to increase the efficiency of video management. In order to achieve this greater efficiency, we need to develop an effective video management system which is a challenge. Taking this importance and challenge into account, I proposed the *Distributed Video Management System using the Concept of Spanner*, that will increase the throughput of video management. My main objective of this thesis is to use the concept of Spanner in video management which will outperform in terms of distance and time.

Chapter 1

In recent times, watching and sharing videos are very popular all around the world. Especially, the rapid development of mobile phones and applications has increased the number of internet users. People are spending more time watching videos online and exploring new dimensions of modern technology. Even the uneducated poor people of remote areas are getting education through watching videos free of cost. As a result, people are getting increasingly accustomed to consume and produce videos globally.

1.1 Background:

Videos are a major part of internet. TV shows, movies, broadcast of news and sports event, online games, online education, teleconferencing use internet video to serve users worldwide. For example, YouTube, a world-famous social video-sharing website, is now the third most visited website according Alexa ranking2, having consumed about 10% of all Internet bandwidth in the first quarter of 2010 [1]. Another report shows that Internet video traffic is reported to be the largest Internet traffic generator and will consume 62% of the total Internet traffic by the end of 2015[2]. As social media networks, such as Facebook, Twitter, Instagram, Snapchat, LinkedIn etc. are gaining much popularity all around the world, people are uploading their videos of daily activities every other minute. According to a report published in 2015, Facebook announced that, they are generating 8 billion video views per day, Snapchat announced that, they are doing 6 billion views per day. [3]

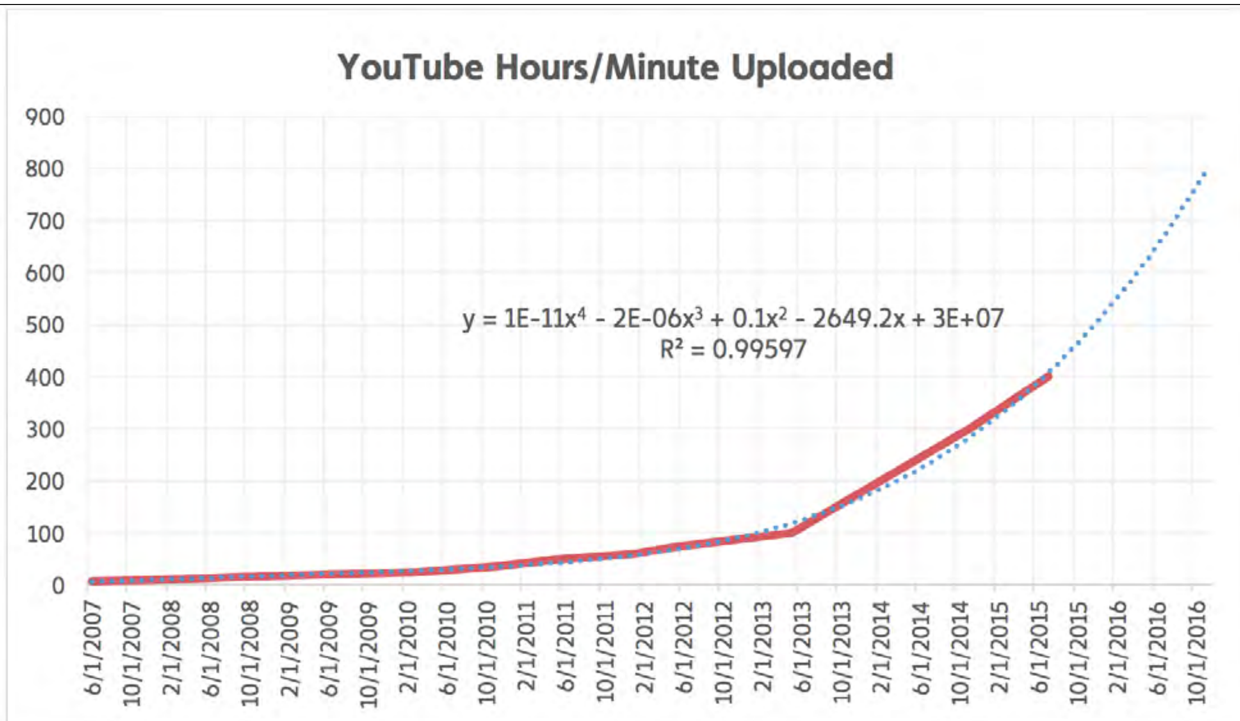

Figure 1.1.1: YouTube Video Explosion

On the other hand, YouTube announced in 2012 that they got 4 billion views per day and 60 hours of video are uploaded every minute, or one hour of video is uploaded every second [4].

So, it is clearly evident that, videos are very popular in today's world and managing this huge number of videos, in a short period of time, is an important issue. That is why, I tried to implement a new concept and provide an effective way of managing videos.

1.2 Motivation:

The background situation clearly shows that, this is high time that, we should be concerned about managing videos globally in a distributed manner to avoid centralization. I am focusing on distributed system because in the physical world in which computing systems reside, components often fail [5]. So, if we do not take this fact into account, it may result in the failure of combined systems.

Nowadays, Spanner, Google's globally distributed database, has gained much popularity among the people dealing with management information system. This paper has motivated me to use the concept of Spanner in video management system to make a contribution in handling this large number of videos.

1.3 Objectives:

As I have already discussed that, managing videos in internet is a burning problem in today's world, I have come up with an idea of using the concept of Google Spanner. When I first read the paper, I found it difficult to grasp the idea of it. But later on, I understood how amazing the idea is and how beneficial it can be.

Spanner is the successor to Goggle's Megastore system [6]. It is Google's globally distributed database, which replicates data across every data center. Its main advantage is that, it can do geographically distributed write operations.

After doing some research on Google Spanner, I came up with the idea of using the concept of it in managing the enormous number of videos in internet. Since Google Spanner is not open sourced yet, my main objective of this thesis is to create a prototype using the concept Spanner, so that I can show its efficiency in video management system.

Chapter 2

2.1 Problem Description:

Spanner is Google's latest invention. Google came up with this idea for several reasons. The main objective was to handle the huge number of data in cloud and overcome the limitations of two-main database management system, MySQL and NoSQL. For several years, researchers are working to come up with an improved solution and Spanner is a result of it.
YouTube is arguably the second largest search engine on the Web. It is the third most visited site on the Web, according to Alexa and SimilarWeb [7]. It has grown incredibly fast to over 100 million video view per day [8]. In order to deal with this enormous amount of data YouTube uses MySQL. Other social media such as, Facebook uses both MySQL and NoSQL; Snapchat uses Google Cloud Datastore which uses NoSQL [9].

MySQL:

- Rich set of features
- Relational Algebra
- Difficult to scale to the massive amount of reads and writes.

NoSQL:

- Highly Scalable
- Non- relational database
- Limited API [10]

Here, both database management system has pros and cons to certain situations. On the other hand, the inconsistency of transactions in Bigtable led to frequent complaints from users which compelled Google engineers to come up with a better solution.

2.2 Proposed Solution:

Taking the mentioned problems in "Problem Description" into account and after doing some research, I realized that Google Spanner would be the best solution for video management system. Google spanner is a NewSQL database management system, which has partial qualities of both MySQL and NoSQL [11].

Key Features:

- Schematized and semi-relational data model
- Ensures global consistency
- Globally distributed at massive scale
- Global replication of data

There are more interesting properties of Google Spanner which would be discussed later on this paper.

Chapter 3

3.1 Video management system:

With the explosion of video resources getting larger and larger, several difficulties have emerged in their management work. The architecture and process of managing this huge video data is called video management system. Due to complexities of big video data management, such as massive processing of large amount of video data, it is challenging to effectively and efficiently store and process these video data in a user-friendly way [12]. Thus, arises the importance of the study and research of video management system.

3.2 Related Works:

In order to manage this big video data in an efficient and effective way researchers have done much hard working developing many video management systems.

- Myoungjin Kim et al. proposed a Hadoop-based distributed video transcoding system in a cloud computing environment, which can transcode various video format to MPEG-4 [13]. They design and implement the platform efficiently using MapReduce framework. Comparing with their work, besides transcoding, a user-friendly interface has been developed to operate HDFS easily. Moreover, details on how to use Hadoop efficiently by tuning different parameters are also presented.
- In [14], the authors proposed a Hadoop-based storage architecture for massive MP3 files. They used classification algorithm in pre-processing module to merge small files into sequence files. They confirm that the introduced efficient indexing mechanism is a good solution to the problem of small files.

- Lin et al. [15] proposed a prototype of cloud based video recording system. The system can provide scalable video recording, data backup and feature detection. They also used HDFS to store video data.
- Liu et al. [16] present a framework for video playing and video storage based on Hadoop. Their framework provides high availability services, which could support concurrent access and playing streaming media in mobile terminals.
- In [17], the authors presented a video monitoring system that can meet the users' demands of searching video, uploading video, downloading video and transcoding video. They also used FFmpeg to transcode video. In survey work, they focused on the usability of the video management platform, plus the experiences on integration of J2EE, Flex, and Hadoop.

Chapter 4

4.1 Spanner:

Spanner is Google's scalable, multiversion, globally distributed and synchronously replicated database. It is the first system to distribute data at global scale and support externally consistent distributed transactions. One of its main functionality is, it automatically reshards data across data centers and it automatically migrates data across machines and data centers geographically to balance load and in case of failures [18]. It not only stores data across multiple data centers but also in millions and trillions of rows. Spanner is designed in such a way that it can reduce latency while retrieving data by dictating where specific data is stored.

It has also introduced a whole new concept that Google calls True time API. The True time API consists of two main elements atomic clock and GPS which helps to expose clock uncertainty.

4.2 System Architecture:

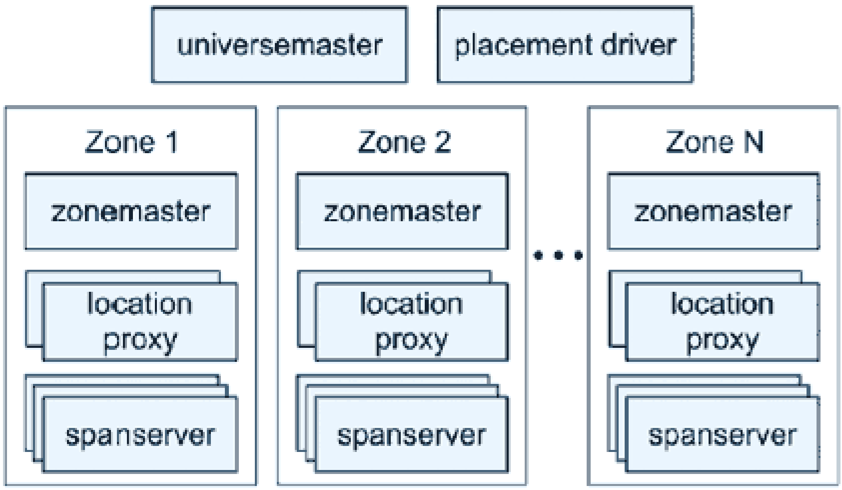

Figure 4.2.1: Spanner Server Organization

- Here universe means the deployment of Spanner.
- A zone is the unit of administrative deployment. It has 1 zonemaster and 100 to 1000s of spanservers[19].
- A Zonemaster is used to assign data to spanserver and spanserver delivers data to clients. In order to replicate data a Paxos state machine is implemented on top of each tablet. Tablet is a data structure that consists of 100 to 1000 of instances handled by each spanserver.
- Each spanserver operates a lock table to ensure concurrency control and each transaction manager is responsible to support distributed transactions.

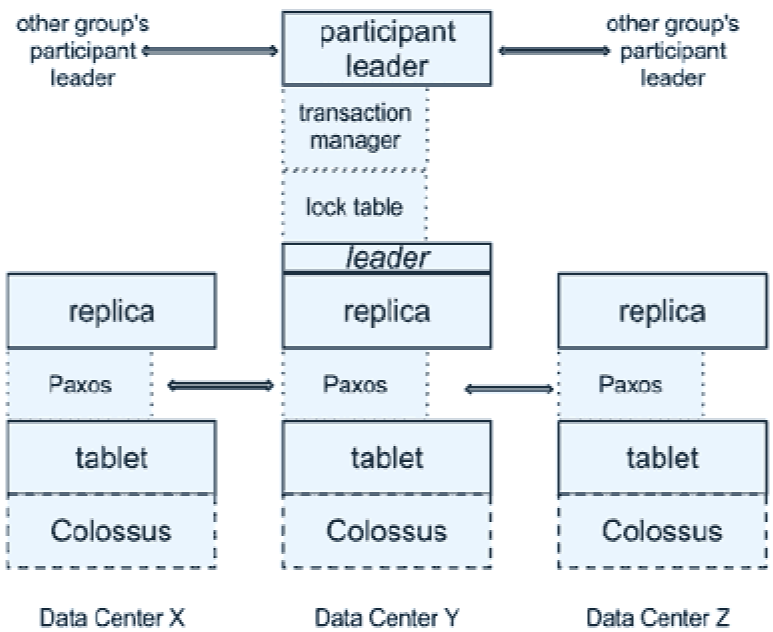

Figure 4.2.2: Spanner Software Stack

- Each replica has a leader.
- In most of the cases a transaction requires only one group of Paxos. In that case, both lock table and Paxos provide transactionality with the help of transaction manager. Sometimes more than one Paxos group may be needed. At that time, those group leaders coordinate among themselves to perform 2-phase commit. A coordinator is chosen from the groups and its leader is denoted as the coordinate leader.

4.3 Why Spanner is better:

It is better because it can overcome some of the shortcomings of Bigtable which is very important in video management system.

- It can replicate data across data centers globally maintaining right consistency which is essential for video management system.
- Applications having complex schemas can be easily dealt by Google Spanner with full transactions.
- One of the main difference between Spanner and Bigtable is that, Spanner does not use the distributed file system, GFS, rather it replicates data by using key space over Paxos. This approach has similarity with Spinnaker which is better than the distributed file system and as a result we can say that, Spanner is better than previous video management systems because of its strong grip in performance, consistency and availability trade-offs [20].
- Spanner is a NewSQL, semi relational database management system that supports ACID transactions. The leader replica of the Paxos cohort has the ability to keep control over the lock table for concurrency control. Every Spanner server has a transaction manager that can implement this lock table.
- Another main feature is TrueTime API that lets you order events which happened on different nodes without communication happening between the nodes to provide the ordering [21]. The most interesting thing is that, Google Spanner has great bounds on clock uncertainty and thus provides high availability. They engineered the TrueTime algorithms around GPS clocks and atomic clocks hooked up to time-servers in each datacenter [22].
- With Bigtable things were a lot easier and simpler but Google has built Spanner in such a way that it can scale in large class of applications which has increased its complexity and reduced limitations of concurrency control and database recovery to a great extent.

Chapter 5

5.1 A Comparison Study:

At present, there are many types of cloud data storage models that are used in video management system. In this section, I want to show a comparison between Spanner and some famous data storage models that are widely used in popular websites.

5.1.1 Spanner vs. Megastore:

- ✓ **High Performance:** Like Spanner Megastore also provides semi-relational data model and similar schema language but as it is layered on top of Bigtable it demonstrates high communication costs. That is why Google Spanner has higher performance than Megastore.
- ✓ **Long-lived Leaders:** Unlike Megastore, Spanner support long live leaders. Since Megastore does not support long-lived leaders, conflict occurs among the replicated data even if they do not conflict logically.
- ✓ **External Consistency:** As throughput collapses in Megastore due to conflict in different replicas, it does not provide external consistency. On the other hand, Spanner provides external consistency.

5.1.2 Spanner vs. DynamoDB:

- ✓ **Global Replication:** DynamoBD is a NoSQL database management system. It can replicate data across data centers as a storage service [23]. But it has a limitation. It can replicate data only within a region whereas Spanner can replicate data globally and synchronously.

5.1.3 Spanner vs. MapReduce:

- ✓ **Concurrency Control:** MapReduce is a framework through which we can handle Bigdata. MapReduce is very popular and widely used framework but Google Spanner is better because of its integrating multiple layer. As a result, it has better concurrency control and also reduces cost of commit wait with replication.

5.2 Summary of Comparison:

Test Case	**Spanner**	**Megastore**	**DynamoDB**	**MapReduce**
High Performance	✓	✗		
Long-lived Leaders	✓	✗		
External Consistency	✓	✗		
Global Replication	✓		✗	
Concurrency Control	✓			✗

Table 5.2.1: Comparison between Spanner & existing systems

From the above analysis, it is clearly shown that Google Spanner can be a better mode of distributed video management system rather than existing systems.

5.3 Final Proposed Solution:

Considering the comparison between Google Spanner and existing systems, I came up with the solution that Spanner would be a better approach for distributed video management system. I chose this system as a solution because it has some unique features which are not available in existing systems and even if they exist, it has some limitations which confines the performance of video management system. The reasons I chose to implement the concept of Google Spanner in distributed video management system are global replication of data, clock uncertainty, external consistency and higher performance.

Using the concept of Spanner, I have developed my own system with certain modifications which will be described in Chapter 6 with necessary diagrams and flowcharts.

Chapter 6

6.1 Proposed System:

My proposed system is a modified version of Google Spanner. I have done this modification for the betterment of video management system. While researching, I have come across several problems in video management system. That is why I included some new features using the concept of Google Spanner that would make my proposed system more compliant.
The steps of my proposed system are given below:

1. Suppose Client A under the region of spanserver Bangladesh is requesting for a video S. Spanserver is always selected region wise. The replication of video S is available in all the data centers of the geographically distributed spanservers. This is a feature of Google Spanner.
2. In my proposed system, I have included distance-metric among the data centers to get the video from the shortest distance available from Client A.
3. From the figure, we can see that the shortest distance holder is data center X. So, Client A will get the video from data center X.
4. But if data center X is busy or unavailable, Client A will get the video from the second shortest distance from the stack which is data center Z.
5. Before taking the video from data center Z, a threshold check will be executed.

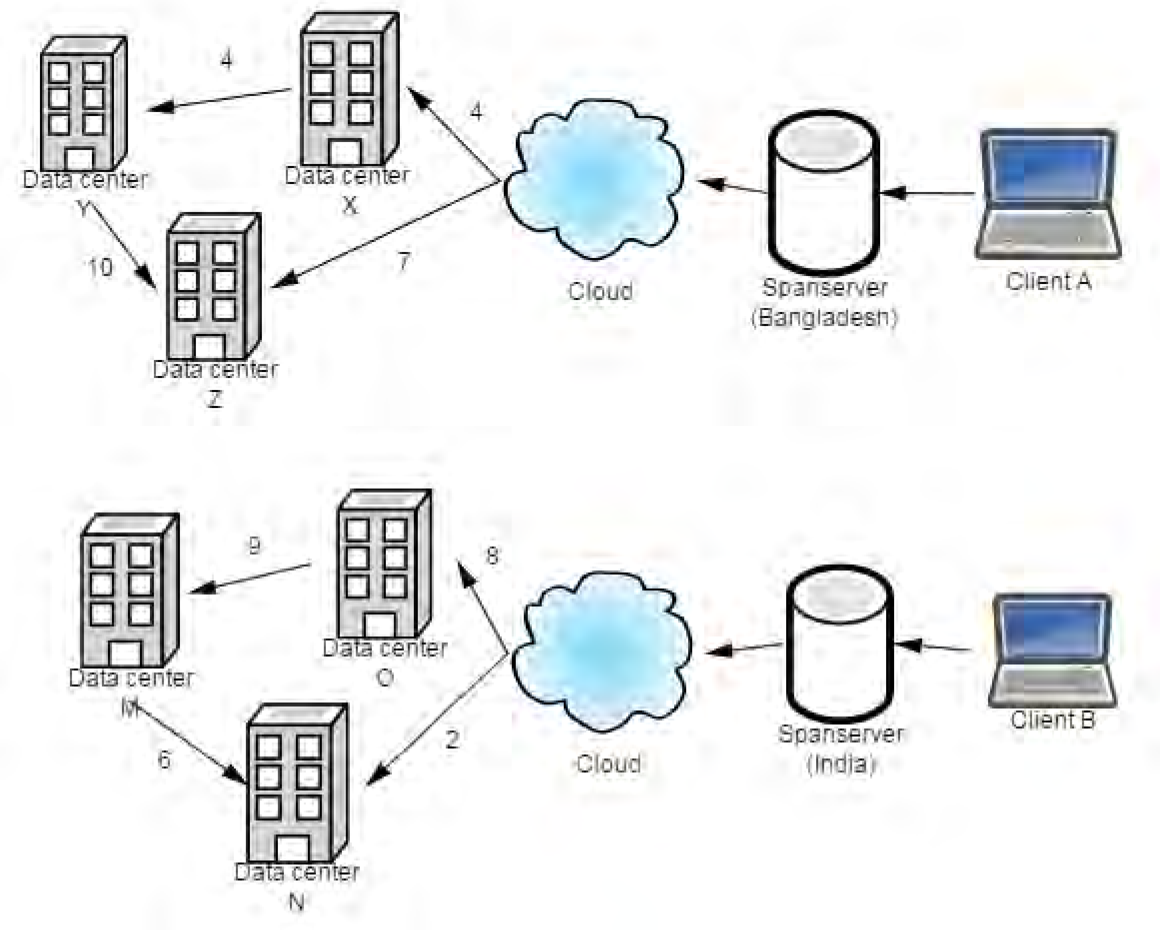

Figure 6.1.1: Proposed system's work flow

6. This threshold check will determine whether it will be efficient to wait for data center X to be free or retrieve the video from the data center Z.

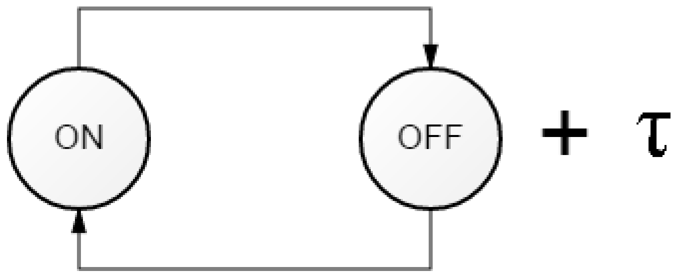

Figure 6.1.2: Threshold Wait to turn on device

7. If the threshold wait (tau) is less than the time of getting the video from data center Z, that Spanner will wait till data center X gets free. Otherwise it will get the video from data center Z.

Equation:

- P (Request Server | Device On) * ∂ (Device)

  I have generated this equation to develop my proposed system and show results.

6.2 System Flowchart:

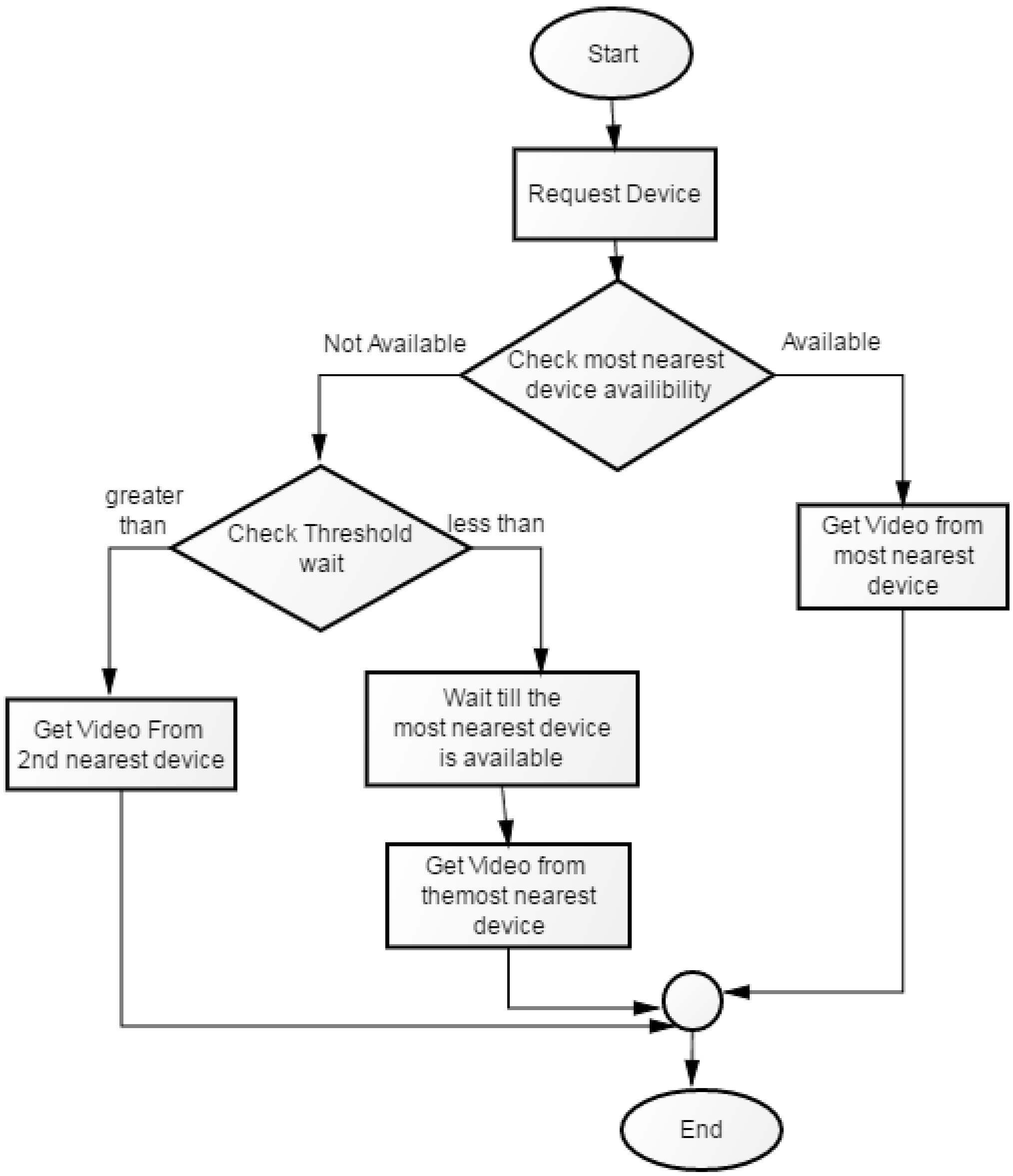

Figure 6.2.1: System Flow-Chart

Chapter 7

7.1 Results:

In order to generate results first we have taken a random graph considering the fact that each node of the graph contains replicated data that the client has requested. In the graph, we fixed a client node that request video from the graph. In the graph, I have included distance metric among the nodes.

Then I ran the algorithm to find the shortest path from the graph. The shortest paths are being stored in a stack. On top of the stack stays the closest node after that the second closest node and so on.

When client requests for a video my algorithm delivers video from the closest node. If the closest node is busy it checks the threshold wait for the closest node. If the wait is less than the time of retrieving data from second closes node from the stack, it waits until the first closest node becomes available. If not, then it retrieves the video from the second closest node in the stack.

This program is being compared to another program which also generates a random graph. But in that program the client receives video from a random node irrespective of distance metric.

After comparing these two systems I found out that my proposed solution is better than the program which chooses data centers randomly to retrieve video.

After running the efficiency test with respect to distance and time I found out that my proposed solution is **73%** efficient and the program with which I compared is **70%** efficient.

My findings prove that, my proposed solution is **3%** more efficient.

7.2 Graph:

From the generated results, I plotted a graph to show the efficiency of my proposed solution through graphical representation.

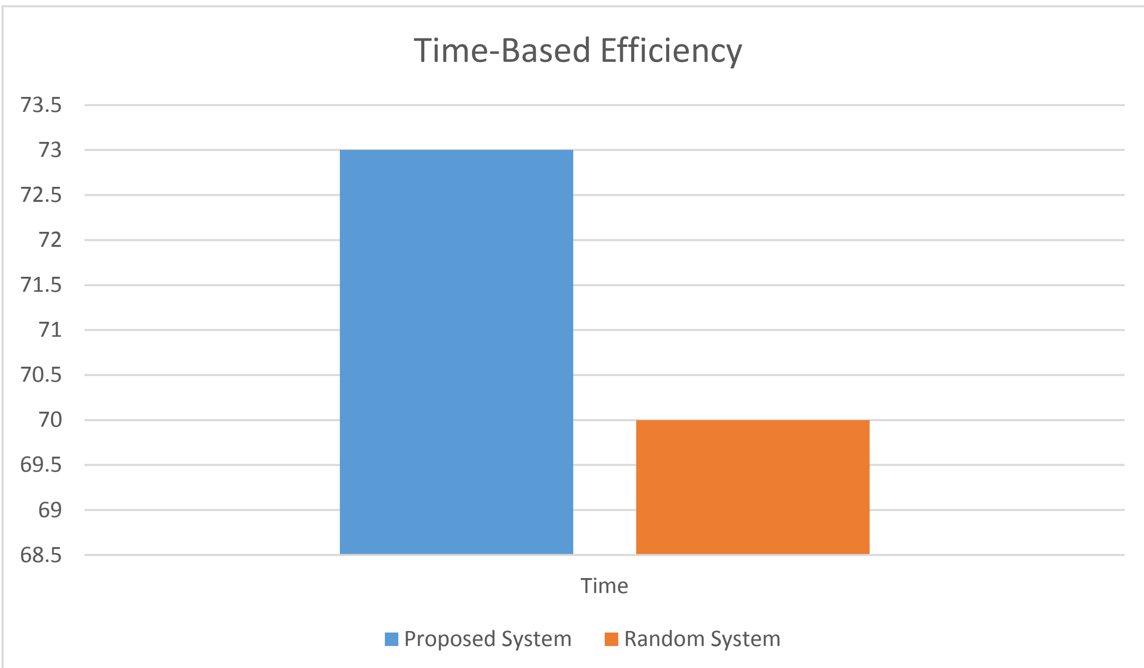

Figure 7.2.1: Time-Based Efficiency results

Chapter 8:

Like most of other proposed Video Management System by various researchers, my proposed solution also contains some limitations and for which some future works need to be done.

8.1 Limitations:

As I have already mentioned in the paper that I am proposing to use the concept of Google Spanner for Distributed Video Management System. Google Spanner is a revolutionary invention and their paper is worth reading. I tried to create a prototype of my proposed solution but creating an actual Spanner is not possible for me alone because of its complexity. That is the biggest limitation of my thesis. I want to show my proposed solution more efficient and effective.

It would have been easy for me if Spanner had free API but at present Google is using Spanner internally to handle Bigdata and the API is not free yet. They are trying to add more features to it so that it can minimize all the problems of handling Bigdata. Since it is a latest innovation and a new idea it will take time for the programmers to make it.

8.2 Future Works:

Overcoming the limitations are the future works of my thesis. The progresses that I would like to make in my thesis in future are-

- I would like to create a complete Spanner and use it in Video Management system.
- I want to like compare it then with the existing system and show its efficiency.
- After that, if any modification in the Spanner makes it better for the Video Management System, I would like to do that.

Finally, as a student of Computer Science and Engineering I would like to give my contribution and take my thesis to the next level.

Conclusion:

Managing the increasing number of videos in the cloud is an important issue in the world of Computer Science. In today's world, it's all about data and taking this important fact into consideration I started my research and came up with the idea of distributed video management system using the concept of Spanner. Now a day's video is used in all aspects of our lives. So, its availability is an important matter.

That's why I propose to use the concept of Google Spanner in distributed video management system. In my proposed solution, I also did some modification and added new features which made my proposed system unique and efficient. Throughout the paper, I talked about videos, various types of existing video management systems, how Spanner is better than the existing system and so on.

At the end, I also compared and showed efficiency results that my proposed solution is better. I also mentioned the limitations and future works of my thesis at the end of my paper.

To conclude it can be said that, taking a tour in the world of data management system was a pleasant journey. In future, I would love to give my contribution in the research field of distributed video management system.

REFERENCES

1. Z. Shen, "Large-Scale Sensor-Rich Video," China, 2013

2. Cisco Systems, Inc. Cisco Visual Networking Index: Forecast and Methodology, 2010-2015. White Paper, 2011.

3. http://www.theverge.com/2016/4/28/11526294/snapchat-10-billion-daily-video-views-users-stories.

4. YouTube Statistics'. http://www.youtube.com/t/press statistics, 2012.

5. J. Sheehy, "There is No Now," *acmqueue.*

6. J. Clark, "http://www.zdnet.com/," September 2012. [Online]. Available: http://www.zdnet.com/article/google-reveals-spanner-the-database-tech-that-can-span-the-planet/.

7. T. Edward, "http://searchengineland.com/," July 2015. [Online]. Available: http://searchengineland.com/youtube-ranking-factors-getting-ranked-second-largest-search-engine-225533.

8. "http://highscalability.com/," March 2008. [Online]. Available: http://highscalability.com/youtube-architecture.

9. N. Subbaraman, "https://www.quora.com," November 2013. [Online]. Available: https://www.quora.com/What-is-the-architecture-behind-Snapchat.

10. A. H. Payberah, "Megastore and Spanner".

11. J.C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang & D. Woodford, "Spanner: Google's Globally-Distributed Database" *(*PDF*), Proceedings of OSDI 2012, Google*, retrieved 18 September 2012.

12. X. Liu, D. Zhao, L. Xu, W. Zhang, J. Yin & X. Chen, "A Distributed Video Management Cloud Platform using Hadoop," *IEEE*, vol. 3, 2015.

13. M. Kim, Y. Cui, S. Han & H. Lee, ``Towards efficient design and implementation of a Hadoop-based distributed video transcoding system in cloud computing environment," *Int. J. Multimedia Ubiquitous Eng.*, vol. 8, no. 2, pp. 213_224, 2013.

14. X.-Y. Zhao, Y. Yang, L.-L. Sun & Y. Chen, ``Hadoop-based storage architecture for mass MP3 files," *J. Comput. Appl.*, vol. 32, no. 6, pp. 1724_1726, 2012.

15. C.-F. Lin, S.-M. Yuan, M.-C. Leu & C.-T. Tsai, ``A framework for scalable cloud video recorder system in surveillance environment," in *Proc. 9th Int. Conf. Ubiquitous Intell. Comput., 9th Int. Conf. Auto. Trusted Comput. (UIC/ATC)*, Sep. 2012, pp. 655_660.

16. C. Liu, K. Fan, Z. Yang & J. Xiu, ``A distributed video share system based on Hadoop," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2014, pp. 587_590.

17. Y. Liu, S. Liang, M. Zhao & Y. Sun, ``Hadoop applications in video monitoring," in *Advanced Materials Research*, vols. 1030_1032. Dürnten, Switzerland: Trans Tech Publications, 2014, pp. 1717_1720.

18. S. Higginbotham, "https://gigaom.com," September 2012. [Online]. Available: https://gigaom.com/2012/09/17/googles-spanner-a-database-that-knows-what-time-it-is/.

19. S. Murat & K. Demirbas, "Beyond TrueTime: Using AugmentedTime for Improving Spanner".

20. J. Rao & E. J .Shekita, S. Tata, "Using Paxos to Build a Scalable, Consistent and Highly Available Datastore".

21. G. Farnum, "https://www.quora.com," April 2013. [Online]. Available: https://www.quora.com/What-are-the-implications-of-a-concept-like-TrueTime-api-from-Google-Spanner-in-distributed-systems-design.

22. Sandeep, "Google Spanner," 2012

23. Amazon. Amazon DynamoDB. 2012.

.