Research Paper

Group Members:
Abu Bakar Hasnath (20301037)
Humaira Khan (20301069)
Sheikh Alima Mahbub (20101517)
Mirza Raiyan Ahmed (20101188)

Title

A comparative analysis of the cost of execution and power consumption of producing video timestamp data between a traditional stand-alone computer and fog computers

Abstract

The number of CCTV cameras being utilized by businesses all over the world is rising quickly as surveillance technology becomes more widely available. Timestamps in the recorded video are necessary so that an operator can rapidly enter an event when reviewing the video clip. CCTV cameras may now operate continuously and without pausing, producing high-quality footage thanks to recent advancements in technology. Additionally, the size and resolution of video data have both grown. These factors have made the computationally complex and costly process of producing these timestamps on a conventional computer For the sake of our paper, we have chosen to generate these timestamps using a network of Fog computer systems. We will first create these timestamps on a specific computer using the required method, and then we will carry out the same process using a network of fog computers. Only one fog node at a time will be used in a network of fog computers, and any fog node can fail independently. We will gather information about the size of video files being transferred, network bandwidth usage, and computational power used while handling video files on a single computer and subsequently using a group of fog machines. Finally, we will compare the data to see if there has been a decrease in the amount of bandwidth used, the size of the data, and the processing power required to handle the video data. We are certain that we can demonstrate the tangible advantages of using fog computers to manage CCTV camera video data as opposed to using a host PC to send the files directly to the cloud for storage.

Keywords

CCTV cameras, Video timestamp, Stand-alone computers, Fog computers, Video data resolution, Bandwidth, Computational power, Cloud storage

Introduction

An exponential rise in the amount of surveillance data has resulted from businesses all over the world quickly implementing CCTV cameras. Timestamps are essential for rapidly evaluating recorded video footage because they allow operators to jump right to important occurrences. Recent technological developments have made it possible for CCTV cameras to run constantly, producing high-quality video data of increased size and resolution. On conventional stand-alone computers, creating timestamps has become quite difficult due to the processing complexity and cost involved.

In this study, we investigate the generation of video timestamp data using fog computers to overcome these difficulties. A distributed computing paradigm called fog computing eliminates the need to send data to centralized cloud servers by moving processing and storage closer to the data source. We believe that a network of fog computers can help us generate timestamps for CCTV camera video data more quickly and affordably.

We will compare the performance of a conventional stand-alone computer with that of a fog computer network to verify our hypothesis. On a particular computer, we will begin by producing timestamps using tried-and-true techniques. To account for any independent fog node failures, we will then replicate the operation using a network of fog computers, using only one fog node at a time.

We will gather important data for our investigation, such as the size of the video files being transported, the amount of network bandwidth used, and the amount of processing power needed to handle video files. This information will be gathered as the fog computer network and the standalone computer process the video stream. We will compare these measures to see if using fog computers for CCTV camera video data management results in a decrease in bandwidth utilization, data size, and computing power.

We will use iFogSim, a Java-based framework for simulating fog computing, to simulate the fog computer network. We may track variables like latency, power usage, and network congestion using this effective tool in an end-to-end network connection. We will investigate the effectiveness and performance of fog computing environments using iFogSim, collecting information important to our comparative analysis.

A dedicated stand-alone computer with a high configuration and a potent GPU capable of performing sophisticated machine-learning algorithms will be used in the experimental setup. The standalone computer will take network-provided video footage data and process it using machine learning models to provide timestamps. Following that, these timestamps will be uploaded to a

cloud storage platform so that operators can easily access them during video analysis.

iFogSim will be used to simulate an environment in the fog computer network configuration. The network will be made up of IoT devices that will operate as CCTV cameras and feed video data to a scheduler on a fog computer. The scheduler will assess the fog computers' availability and allocate video processing work accordingly. We will shift tasks to available fog machines to accommodate any fog node outages. The video timestamp files that come from this process will be delivered back to the fog computer scheduler, which will then send them to a cloud server for storage.

We will gather information about video file sizes, network bandwidth usage, and computational power consumption during the research process for both the stand-alone computer scenario and the fog computer network. With the use of this information, we will be able to conduct a thorough examination of the advantages and effectiveness of using fog computers to manage CCTV camera video data.

This study aims to show the real benefits of fog computing in terms of cost efficiency, lower bandwidth consumption, and enhanced processing power for timestamp creation. The results of this study will advance the field of surveillance technology and educate decision-makers on the advantages of incorporating fog computing into their CCTV camera systems.

1 Proposed Model

In a traditional architecture, local high-end dedicated computer systems generate video timestamps from video data passed on from Network Video Recorders (NVRs) or Video Management Systems (VMS) obtained from a video sensor device like CCTV cameras. An organization places camera sensors strategically and often uses NVRs or VMS to control these video cameras for security purposes. An event or action, or anomaly from video data can be detected using traditional algorithms or deep learning-based methods [**?**]. Network Video Recorders (NVRs) and Video Management Systems (VMS) are incapable of running large deep learning-based models for event detection, and sometimes they also fail to detect events using hard-coded algorithms. Hence, a dedicated high-end computer system must run locally nonstop to produce these timestamps and export them to the cloud.

In our proposed FOG model, we plan to use many FOG servers placed at remote locations to handle the computational task of producing these timestamps and finally sending them to the cloud for storage. Our FOG model can be implemented using available consumer-grade hardware so that more organizations can implement our FOG model while keeping the cost low. Moreover, our paper contains a detailed methodology so anyone can replicate our work on their consumer-grade hardware.

[width=]figure$_{11}$.*png*

Figure 1: Comparison of the traditional cloud and FOG Model (Fig 1.1)

2 Proposed FOG-based Model Architecture

Our paper introduced a stratified model architecture characterized by each distinct task carried out into separate layers.

CCTV cameras, along with Network Video Recorders (NVRs) or Video Management Systems (VMS), together form the bottom layer (Layer 0). In this layer, controllers like NVRs and VMS receive live video feed from the sensor camera; they slice these live video feed into segments and save them in local storage premises using Network-Attached Storage (NAS) or Storage Area Networks (SANs). The control system then forwards these distinct video data files to the next layer, which contains the Dedicated Scheduler.

(Layer 1) consists of the Scheduler Computer. This dedicated local computer receives video files from the layer below, and its task is to forward this video file to an available FOG server located at a remote location. This computer will have all the necessary information about all the remote FOG servers at the next layer (Layer 2); it will first check whether a specific FOG server is available. If available to compute, the Scheduler Computer will forward the video data file to that FOG server; if not, it will send the data to the next available FOG server. This Scheduler Computer is the only high-end local computer that will always be active because this machine is the decision-taker; it handles all the FOG servers and implements intelligent scheduling algorithms to balance the load among all the FOG servers. Implementing necessary security measures like using Secure Shell (SSH) while transferring video file data between the layers is crucial[8].

Preferably (Layer 2) can be implemented in a remote location where power and data centers are cheap. FOG servers at a remote location receive video data and then implement necessary methods, such as a traditional method or deep learning algorithm, to produce timestamps and forward the timestamp to the cloud storage server, which is at the next layer.

(Layer 3) consists of the Storage Server, the final layer of our proposed FOG architecture. The cloud server receives timestamp files from the FOG server and saves them along with logs of the data transmissions; they will save the IP address and usernames of the FOG servers. An operator can retrieve timestamp files from the cloud at any time for utilization.

Dataset Analysis

1.Highway Traffic Videos Dataset

This dataset is obtained from an open source. This database is about a video of traffic on the highway in Seattle, WA. This particular video was taken over for

two days consecutively by a stationary camera. It is labeled as light, medium and heavy traffic respectively. The 1st frame of this video interferes with another video signal. So, while processing each video we will have to start from the second frame. Besides this, the cropped version of the video is provided in a MATLab file. In this dataset there are a total of 264 files of which 254 are videos. This video is provided courtesy of the Washington State Department of Transportation.

Location: Source: http://www.wsdot.wa.gov/ City: Seattle,WA Direction: Looking South Location: I-5 S 188th street Traffic: Southbound Traffic Date: 05-08-2004 to 06-08-2004

2.Smart-City CCTV Violence Detection Dataset (SCVD)

The current datasets such as - NTU CCTV-Fights dataset, Real-Life Violence Situations Dataset and some other currently used databases for violence detection comprises videos recorded from phone cameras that could reshape the distribution and focus of the CCTV based Violence detection. This dataset contains a class for weapons detection in videos. Thus making it the first weapon video dataset as other dataset for weapon detection are based mostly on the images of guns and knives. Therefore it means that this dataset modulates the fact that any handheld object which can be used to harm humans or properties would be regarded as a weapon. This dataset consists of 3 directories- Non-violence (248 videos), Violence (112 videos) and Weapon Violence (124 videos). Furthermore there are a total of 481 files.

Location: Source: Youtube Date: 10-01-2022 to 09-5-2022

Video Timestamp Generation

We will generate timestamps that are primarily highlights from CCTV video footage data using classic machine learning models such as CNN (Convolutional Neural Networks) and RNN (Recurrent Neural Networks).

Comparative Analysis

Data obtained from the fog computer network using iFogSim and data obtained from the standalone computer will be compared. We will analyze the changes or improvements in the fog computer network's bandwidth utilization, data size, and processing power. The advantages and effectiveness of using fog computers to manage CCTV camera video data will be discussed.

Literature Review

The use of fog computing in Smart Transportation Systems (STSs) is discussed in this research, which also highlights the necessity of policy-driven security management in these settings. In addition to IoT verticals, an orchestration

layer, and an abstraction layer, it proposes a fog computing architecture. To address security issues, the paper introduces a policy management module and improves the current architecture with elements like the Policy Decision Engine, Application Administrator, and Policy Enforcer. The OpenAZ framework and XACML-based policies are used in a testbed to assess policy enforcement in an STS scenario. In addition to outlining upcoming work on resolving policy conflicts and enhancing the framework's capabilities, the article emphasizes the significance of policy management for secure collaboration in fog computing.[1]

To extend cloud services to Internet of Things (IoT) devices, fog computing presents security and privacy problems. It highlights the need for creative solutions and ongoing research by identifying issues in network services, data processing, and IoT device privacy. Authentication, access control, lightweight protocols, and trust management have all been suggested as solutions, but issues with heterogeneity, scalability, and compatibility still exist. The paper highlights the significance of cutting-edge methods and specialized approaches to guarantee strong security and privacy in the Fog computing environment, taking into consideration its particular features and resource limitations.[2]

This article examines the idea of fog computing, which permits IoT devices to run applications at the network edge. It addresses security and privacy problems as well as the benefits of fog computing in several fields. The essay focuses on the importance of additional research in extending the Fog computing paradigm, creating models for Fog devices, and enhancing service mobility across heterogeneous platforms. Overall, it demonstrates the promise of fog computing to deliver low latency, increased quality of service, and location awareness at the network edge.[3]

As a replacement for cloud computing, fog computing introduces services to the network edge for latency-sensitive IoT applications. The benefits of fog computing are covered in this article, along with the work being done by the OpenFog Consortium with the backing of organizations like Cloudlet and Intel. It emphasizes research initiatives in scalable authentication, privacy preservation, access control, and fog forensics and focuses on the security and privacy concerns particular to fog computing. The paper highlights the need for creative solutions to deal with these issues and identifies open research topics in forensics, intrusion detection, trust, authentication, and privacy. Overall, the research emphasizes how crucial it is to safeguard data in fog computing settings to assure its continued growth in the IoT industry.[4]

This study proposes a framework for fog-based health monitoring that uses wearables with integrated sensors and fog computing to gather patient data. The framework uses cloud storage and fog gateways for effective healthcare services, together with real-time data processing and data security safeguards. Fog computing's benefits in clinical decision-making and healthcare monitoring, such as decreased latency and increased reliability, are emphasized. In IoT-based healthcare systems, the suggested framework offers a safe and scalable approach for real-time health monitoring.[5]

To detect threats in fog computing, this work offers a distributed deep learning (DL) technique dubbed CAVID. Using the NSL-KDD dataset, the study

compares the performance of deep learning (DL) models versus shallow learning (SL) models. Accuracy and scalability are increased by using pre-trained stacked autoencoders (SAE) for feature engineering and Softmax for classification. The analysis demonstrates that DL models perform better than conventional ML architectures, providing higher detection rates and fewer false positives. The study demonstrates the potential of DL in tackling cybersecurity concerns in the IoT and fog computing domains and helps to design better security architectures for fog computing. Future work will involve analyzing additional datasets and models, training on dispersed IoT networks, and enhancing precision.[6]

The comparison and contrast of the Cloud, Edge, and Fog computing paradigms are highlighted in this systematic literature study. It concentrates on the security and privacy elements of these paradigms, outlining the major difficulties and weaknesses as well as the appropriate defenses. To safeguard sensitive data and maintain system integrity, the evaluation emphasizes the significance of comprehensive security measures, including access control, encryption, authentication, and data integrity. To create novel solutions that effectively preserve security and privacy while being adapted to the particular traits of each paradigm, more research is required.[7]

Conclusion

In generating video timestamp data for CCTV cameras, this research article compares the execution costs and power consumption of a conventional stand-alone computer versus a network of fog computers. The study emphasizes the potential benefits of utilizing fog computers in managing CCTV camera video data by examining the benefits and effectiveness of fog computing. Data on video file sizes, network bandwidth usage, and computing power consumption for both the stand-alone computer and fog computer network scenarios are collected using the suggested methods. The study seeks to use comparative analysis to show how fog computing is more affordable and effective, providing decision-makers in the field of surveillance technology with useful information and encouraging the use of fog computing for enhanced CCTV camera systems.

References

[1] DSouza, C., Ahn, G.-J., & Taguinod, M. (2014). Policy-driven security management for fog computing: Preliminary framework and a case study. In Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014) (pp. 16-23). Redwood City, CA, USA: IEEE. doi: 10.1109/IRI.2014.7051866.

[2] Alzoubi, Y. I., Osmanaj, V. H., Jaradat, A., & Al-Ahmad, A. (2021). Fog computing security and privacy for Internet of Things applications: State-of-the-art. Security and Privacy, 4, e145. https://doi.org/10.1002/spy2.145

[3] Stojmenovic, I., & Wen, S. (2014). The Fog computing paradigm: Scenarios and security issues. In 2014 Federated Conference on Computer Science and Information Systems (pp. 1-8). Warsaw, Poland. doi: 10.15439/2014F503

[4] Mukherjee, M., Islam, R., & Misra, S. (2017). Security and Privacy in Fog Computing: Challenges IEEE Access, 5, 19293-19304. doi: 10.1109/ACCESS.2017.2749422

[5] Kumari, A., Tanwar, S., Tyagi, S., & Kumar, N. (2018). Fog computing for Healthcare 4.0 environment: Opportunities and challenges. Computers Electrical Engineering, 72, 1-13. ISSN 0045-7906. https://doi.org/10.1016/j.compeleceng.2018.08.015

[6] Rawat, R., Chakrawarti, R. K., Vyas, P., Gonzáles, J. L., Sikarwar, R., & Bhardwaj, R. (2023). Intelligent Fog Computing Surveillance System for Crime and Vulnerability Identification and Tracing. International Journal of Information Security and Privacy (IJISP), 17(1), 1–25. http://doi.org/10.4018/IJISP.317371

[7] Abbas, N., Yau, K.-L. A., Liang, K., & Tang, Z. (2023). A Systematic Literature Review on Security and Privacy in Cloud, Edge, and Fog Computing Paradigms. IEEE Access, 11, 48323-48345. doi: 10.1109/ACCESS.2023.3052345

[8] Carnegie Mellon University School of Computer Science. (n.d.). Secure shell (SSH) - SCS computing facilities - Carnegie Mellon University. *Secure Shell (SSH) - SCS Computing Facilities - Carnegie Mellon University.* Retrieved from https://computing.cs.cmu.edu/security/security-ssh

[9] Aremu, T., Zhiyuan, L., Alameeri, R., Khan, M., & Saddik, A. E. (2023). SSIVD-Net: A Novel Salient Super Image Classification & Detection Technique for Weaponized Violence.

[10] Chan, A. B., & Vasconcelos, N. (2005). Probabilistic Kernels for the Classification of Auto-Regressive Visual Processes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. San Diego).

[11] Chan, A. B., & Vasconcelos, N. (2005). Classification and Retrieval of Traffic Video using Auto-Regressive Stochastic Processes. In *Proceedings of 2005 IEEE Intelligent Vehicles Symposium* (pp. Las Vegas, June).

[12] Biradar, K. M., Gupta, A., Mandal, M., & Vipparthi, S. K. (2019, June 11). Challenges in time-stamp-aware anomaly detection in traffic videos. *arXiv.org.* https://arxiv.org/abs/1906.04574