Enhance Cloud Storage upload latency with Mobile Edge Computing and achieve SLA transparency with Hybrid-Blockchain.

MSc Research Project
Cloud Computing

Sathish Kumar Krisnamoorthy
Student ID: X20208057

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sathish Kumar Krisnamoorthy
Student ID:	X20208057
Programme:	Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Shivani Jaswal
Submission Due Date:	15/08/2022
Project Title:	Enhance Cloud Storage upload latency with Mobile Edge Computing and achieve SLA transparency with Hybrid-Blockchain.
Word Count:	5915
Page Count:	33

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	14th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhance Cloud Storage upload latency with Mobile Edge Computing and achieve SLA transparency with Hybrid-Blockchain.

Sathish Kumar Krisnamoorthy
X20208057

Abstract

Cloud computing has become the primary storage area, not only for information technology but also for individual use. In the Age of Data, the production of data is tremendous; it is important to securely persist and transmit this data efficiently. The latency is high if the transferred file size is enormous or the transfer distance is significant. In this research, using Mobile Edge computing, an attempt is made to reduce the propagation, processing, and queuing delays. The Service Level Agreement (SLA) of the proposed system is achieved through Blockchain technology. This research will use a live dataset obtained from the Amazon open data source, which contains mobile devices' network and location details, to prove the hypothesis, and the dataset contains data about 300 million devices. This dataset is preprocessed using a python script that will map them to the GeoPandas library and generate random global coordinates and their closest peer edge devices. These random coordinates were generated and were subjected to the countries' boundaries. According to the dataset, India, the USA, and China were chosen for evaluation because they were the top 3 countries with the maximum number of participating devices. This preprocessed result is passed to a Python simulator that calculates the latency improvement of this system. The results of this methodology show that compared to a datacenter located 250 KM away from a specific point, the proposed system will be faster by 107.86%, 1847.60%, 18,289.91%, 36007.63%. for transferring 1 GB, 10 GB, 100 GB, and 200 GB of data.

1 Introduction

The production of data has increased tremendously with the widespread use of internet-based applications. As a result, the secondary storage space of current mobile devices is insufficient for ordinary users, which leads them to use personal cloud storage. Even though the cloud solves the problem of data storage, the rate at which a file is uploaded depends on its size and the distance it has to travel over the network, which may be quite far in the case of a personal cloud because the service provider has a limited number of data centers. The Cloud-Centric strategy has its limitations, such as **bandwidth** constraints when transferring large files; **latency** constraints when transmitting data to distant locations; **uninterrupted connection** due to restricted centralised servers; and **security** limitations as it is managed by individual corporations. If the above-mentioned limitations are addressed, it could provide a better upload speed for cloud storage, which might benefit a plethora of personal cloud users and a few latency-sensitive applications.

With the limitation above in mind, the following research question has been formed.

"Can the upload latency of the cloud computing be improved using a decentralized cache system with mobile edge computing and hybrid blockchain ?"

In this research, the personal cloud user's mobile devices are employed as edge devices, and free storage will be provided as an incentive for their participation. These edge devices are used as level-one caches for holding the data chunks transferred to the personal cloud by the end user. The end user will have the impression that data transmission is done once it reaches the edge devices. This transmission will be faster because the edge devices will be located much closer to the end user compared to the data-centers. Later down the line, the data from the mobile edge devices can be sent to the core data center. From the literature review, most of the current personal cloud service providers do data compression on the user device to reduce network congestion, which could be avoided in this system by offloading that computation to peer devices. The research conducted based on distributed cloud storage has been discussed in the related work section, and none of them has evaluated the latency improvement in distributed cache systems with a live dataset. In this research, a recent dataset from the Ookla speed test organisation is used. The dataset contains 32 million locations, and the count of mobile devices in those locations, along with their upload and download bandwidth, has been provided. This data is used to evaluate the latency improvement of the proposed system.

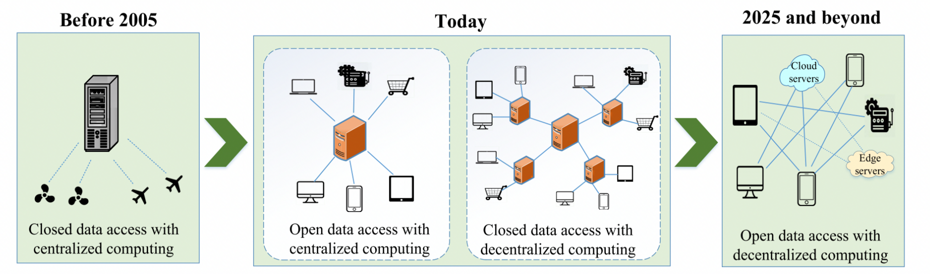


Figure 1: Past towards Future
Nguyen et al. (2020)

The security issue in the cloud-centric approach could be resolved with the help of distributed and decentralised computing. Figure 1 shows the timeline of computing, and the future is travelling towards complete open data and decentralised computing. According to Nguyen et al. (2020), the world is at *decentralised computing with closed data access*, and it is on a mission to migrate to *decentralised computing with open data access*. The projects like Inter Planetary File System (IPFS) Benet (2014), MaidSafe Irvine (2010), Sia Vorick and Champine (2014), Filecoin Benet and Greco (2018) were developed to achieve this complete decentralised network and open access, but projects like IPFS couldn't grow without peer participation, which doesn't happen overnight. The participating peers might need proper motivation. This research might be one of the stepping stones for that future. In this paper, Mobile Edge Computing and Blockchain distributed systems will be used to address the limitations mentioned above.

The key points of this research are as follows:

- Establish a distributed layered cache system using mobile edge and blockchain to improve the personal cloud latency.
- Using mobile edge computing, the compression of huge files can be offloaded to peers and improve the processing delay drastically, along with it the propagation delay is also improved as the distance between the user and the data centre is reduced.
- Calculate the improvement in latency and find the total throughput while using a distributed cache system with the help of machine learning libraries and an appropriate dataset. From the literature review, most of the similar systems do their calculations based on assumptions, but in this research, relevant dataset are used for it.
- Build a system that would lead to decentralised storage by gaining the trust of peer nodes.
- Maintain the system's network with the continuous participation of the peer nodes by providing the proper incentive for their contribution.

The subsequent section of the document is II: Related Work, where the literature survey is subdivided into subsections. This is followed by the III: Methodology Section, in which the research approaches are detailed, and the IV: Design Specifications Section, in which the research plan and experimental design strategy are discussed. Next, the implementation tools and techniques used to prove the proposal in the V: Implementation Section are followed by the results of the experiment and an evaluation of them in the VI: Evaluation Section. In Section VII, "Conclusion and Future Work," the research is summed up and the next steps are outlined.

2 Related Work

In this section, a detailed analysis of the related work is made along with the related technology used in this research project. The taxonomy of research is given in Figure 2 as per the ACM computing classification *Computing classification system* (n.d.). The related literature is grouped into sub-sections and reviewed below.

2.1 Extending Cloud to Edge Computing

According to Cisco's whitepaper on fog computing, fog computing extends the cloud computing paradigm to the network's edge, thereby enabling a new class of applications and services Bonomi et al. (2012). The advantages of fog computing are mobility, location awareness, decreased latency, and heterogeneity. Unlike the cloud, fog nodes are much closer to the end-user. It is ideal for offloading the computation cycle to fog nodes. Fog is not a substitute for the cloud but rather an extension of it that improves data management and data analytics in particular. Fog computing, like cloud computing, is a virtualization technique that provides storage, networking, and computing services between the end-user device and cloud data centers.

Edge computing is similar to fog and runs at the network's rim, except it is located further away than the fog's network rim, giving it further more availability to end-users

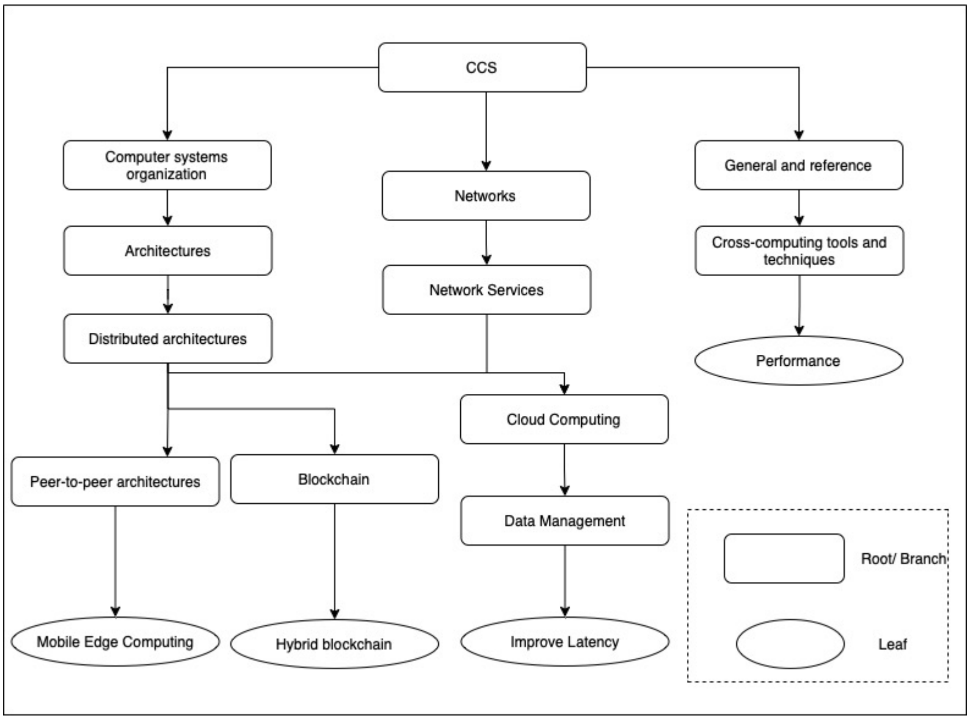


Figure 2: Taxonomy of the Topic

Shi et al. (2016). The compute power of mobile devices is increasing rapidly, with smartphones and laptops leading to the development of apps that are mobile-centric. With the rise of 5G networks, the transmission delay is further reduced and mobile users expect applications with low latency and excellent performance. As mobile devices are severely constrained by energy, it is difficult to execute computation-intensive applications on them. The most effective method for overcoming these limits is to offload computations to the cloud. However, this would increase the latency of the application, which is unacceptable to the users. In this research, mobile edge computing Mach and Becvar (2017) is utilised to offload the computations, which improves the processing delay. In this study, it is suggested that file compression be moved to the mobile edge.

2.2 Decentralized Storage

The cloud has become the primary method for transferring data over the internet, and the majority of people save their personal data on the cloud. As it is not safe to store users' private information in some third-party storage places, In Li et al. (2018) they have developed a blockchain-based P2P network for secure distributed storage. The data is chunked, encrypted, and uploaded to the P2P network, where it is then stored in edge nodes for later retrieval. Multiple copies of data were distributed between P2P nodes and data centres using the genetic algorithm. *The propagation delay has been considerably reduced using this method.* After placing a digital signature, the file is broken into many blocks of fixed-size data packets and stored in peer-to-peer devices. The Merkle root verifies the file's integrity, as the Merkle root hash and file chunk URLs are stored in the blockchain.

The InterPlanetary File System (IPFS)Benet (2014) is one of the greatest instances of a recently built distributed peer-to-peer file management system. The IPFS uses the

GitHub concept of versioning to manage distributed files, so that all older versions of the same file are kept on the network. Along with the Merkle algorithm, it uses torrents, GitHub, and Directed Acyclic Graph technologies to achieve this stable distributed file system. This peer-to-peer version-controlled system will handle issues such as file distribution, file versioning, and accidental file loss prevention. IPFS is a new concept for a decentralised Internet infrastructure that can support a variety of applications, including globally versioned filesystems and namespaces, as well as a next-generation file-sharing system. Users may trust the content they receive without needing to trust their peers. The outdated but essential files do not go missing anymore in this system.

Both the papers above use distributed cloud storage along with blockchain, and the limitations of those are analysed and discussed further. Distributed cloud storage is an alternative to the present centralised private cloud storage. To accomplish distributed peer-to-peer storage, there must be a gradual shift in the users' participation in the peer-to-peer network, as people will not give away their internal storage for free. The proposed system suggests something that is currently relatable and realisable for user participation in the peer network. In the above papers, they employed a single public blockchain, which is not safe because everyone in the blockchain network will know which node holds which file, and there is a possibility that a specific file may be knocked down by attacking the peers that contain that data. As there is just one public blockchain network, it is impossible to store vital data on it. For this reason, a hybrid blockchain paradigm is used in this research, in which important data is stored on a tamper-resistant hashed database. To maintain the transparency of the clients, The system utilises the public blockchain, which stores the Merkley root of the database. Peers can challenge the cloud service provider's SLA for the resources used by the system at any time and confirm it using public blockchain.In addition, the above paper's simulation results for performance are mostly based on assumptions, as the storage capacity and network bandwidth of the datacenter and user nodes were assumed. There is no incentive provided for users with 17 to 26 GB of internal storage to participate in the p2p network. There is no actual evidence to support this notion, as there are still a large number of devices whose internal storage capacity is less than 16 GB. In this paper, 4 million records of different geographic boundaries were identified with their network bandwidth and latency, so the hypothesis of the proposed research can be tested with a proper dataset.

As cloud storage is a centralised entity in which users entrust their data to a third party, it is always a black box for the user, as they are unaware of the data's whereabouts. Although cloud service providers offer various SLAs, the majority of them are opaque. To ensure cloud reliability, Li et al. (2021), a two-layered public blockchain system for cloud transactions, has been proposed to ensure cloud reliability. Two blockchain network layers, Trust Authentication Blockchain (TAB) and Trading Behaviour Blockchain (TBB), are created to ensure the integrity of the participating datacenter, fog nodes, and edge nodes. The TAB is used to authenticate nodes based on their identity and behavioural information. The TBB will maintain the transactions of the participating nodes, generate a trustworthy block, and submit it to the TAB for calculating the nodes' trustworthiness. The miners of this system will be the participating users who will also maintain the double blockchain network.

According to the above report, there is no incentive for miners to engage in the blockchain network since, despite providing an incentive for participation, mobile mining is expensive due to its limited power and storage. Therefore, it is not possible for nodes to mine a public blockchain with two layers, as this could result in frequent device failure. In this research, user-mobile edge nodes are not used as miners. Instead, a hybrid blockchain is used, which drastically reduces storage and computation in peer nodes.

2.3 Blockchain contribution in decentralized system

Blockchain addresses various limitations of cloud computing, like security and transparency. This article, Sharma et al. (2020), is a literature review of the advantages of using blockchain technology along with the cloud. It shows that the combination of blockchain and cloud has a number of important uses, such as *cloud storage with blockchain-based cryptography* for token verification, authentication, etc. It is also used for data searching, data deletion, data integrity, data depublication, blockchain-based auditing, payment in cryptocurreny, and access control.

Blockchain plays a vital role in ensuring security and transparency in a decentralised system. As it uses a connected merkle root-hash chain for ensuring immutable data storage, it is commonly used for transparent auditing, as shown in Li et al. (2020). A hybrid blockchain Wang et al. (2019) is more suitable for such transparent auditing because the critical user data can be abstracted from public display. In this research, a transparent SLA is ensured with the help of hybrid blockchain.

2.4 Present day cloud storage

In Ni et al. (2021); Li et al. (2019), well-known cloud storage providers such as Google Drive, One Drive, Dropbox, Mega, Wuala, Horizon, Cloud Drive, Copy, and HubiC were evaluated based on their design and functionality. According to the report, the performance of the cloud is mostly determined by the file size and distance from the datacenter. The file size cannot be limited because it is fully based on the needs of the customers. However, this system might lower the distance to the datacenter and the propagation latency. This research is based only on this distance reduction via mobile edge computing and performance measurement. Personal cloud storage services optimize network transmission in accordance with Ni et al. (2021) in the following ways.

Chunking: A huge file is divided into many chunks of smaller bytes in order to optimise data transfer. Users whose network connection is interrupted while uploading a file can restart the upload from the point at which they were disconnected thanks to the partial submission of data made possible by chunking. It also benefits users with a sluggish network because the data packets are tiny and hence easier to upload. There is no cause for concern regarding the upload connection. Dropbox and Google Drive both employ chunking and transport their data pieces across distinct TCP connections. Google Drive and Dropbox utilise data chunks of 8 MB and 4 MB, respectively.

Bundling: If there are several small files that are less than the default chunk size, they are merged and sent. If N files are uploaded, Dropbox and Google Drive establish N TCP connections for file transmission. Other services organise the files and deliver them across fewer TCP connections by grouping them. The test results indicate that services that do not employ bundling have significant latency in sending over a long distance or when transferring a large number of small files.

Compression: The file to be transmitted to the server will be compressed in order to decrease network congestion and transfer time. The compression of the data file will also lower the cloud's storage requirements. Google Drive compresses files based on their type, since it does not compress JPEG files in order to preserve their quality. Dropbox compresses every file, no matter what kind it is, while other cloud storage services don't compress any files.

Delta Encoding: Once data has been edited, only the updated piece is transmitted to the server, rather than the complete file. As there is a trade-off between chunking, deduplication and delta encoding, no other service provider does delta encoding but Dropbox.

P2P Synchronisation: The file is sent directly between peer devices, bypassing the server. Dropbox and Copy are peer-to-peer synchronisation applications.

Table 1: Literature Review Comparison

Papers	Blockchain	Edge Computing	Decentralized System	Decentralized Cache Upload
Benet (2014)	✓	✗	✓	✗
Li et al. (2018)	✓	✓	✓	✗
Li et al. (2021)	✓	✓	✓	✗
Nguyen et al. (2020)	✓	✓	✓	✗
Wang et al. (2019)	✓	✗	✓	✗
Li et al. (2020)	✓	✗	✓	✗
Liu et al. (2016a)	✗	✓	✓	✗
Liu et al. (2016b)	✗	✓	✓	✗
Kuang et al. (2020)	✗	✓	✓	✗
Wang et al. (2014)	✗	✓	✓	✗
Zeydan et al. (2016)	✗	✓	✓	✗
THIS ONE	✓	✓	✓	✓

2.5 Content caching in distributed computing

In Liu et al. (2016a), caching over a wireless network is described, and caching is performed based on the region's most popular content. When a certain streaming material is popular and the majority of people in a particular location are consuming it, placing the data file close to the consumer will minimise propagation delay and improve latency. Due to caching at the wireless network's edge, spectral efficiency and energy efficiency have considerably increased compared to the wired network's edge, as demonstrated in this study. The downsides of this strategy are elaborated further below.

Placing particular data on a mobile edge node for a period of time is strictly dependent on the availability of the participating edge nodes. If the edge node moves to a different place, the data will no longer be accessible. Also, when popular material is posted on a particular edge node and many people are consuming it, this will result in a power drain on the edge node due to the fact that edge nodes are mobile devices with limited batteries. As caching is not firmly tied to the participating edge node, in this research project, mobile devices are used for temporary data storage as caches, which will be changed very rapidly. Significant computations will be wasted in the above system because the most popular content is always evolving and must be monitored. Changing the popular content more frequently also necessitates a costly data migration. In this research, as system cache the data temporarily, there is no need to determine the most popular material in a particular location, so there is no need to expend computational resources on it.

As mobile edge caching may substantially reduce latency and increase network performance, Liu et al. (2016b) discusses the security risks and challenges of mobile edge caching. The purpose of caching data is to reduce access time, improve latency, and eliminate the network bottleneck. The study suggests that caching may be divided into proactive caching and reactive caching. Based on the current demand, reactive caching places the most frequently utilised data at the edge nodes. In proactive caching, popular material is forecasted and cached at the edge before content demand increases. Three characteristics are crucial for both reactive and proactive caching: content placement, content distribution, and content utilization. Among the security flaws addressed in the study are the following:

Three distinct techniques for offloading data in edge nodes are presented and contrasted in Kuang et al. (2020). These three approaches offload data to edge nodes using three distinct algorithms: the Backtracking algorithm, the Genetics algorithm, and the Greedy algorithm. The following situation is considered in this paper: There are numerous users and several unloading locations. These users offload their subtasks to the offloading sites, and if the computation is intensive, they may calculate them themselves or transmit them to other offloading points. Based on the evaluation of user cost, execution time, and resource utilisation at offloading locations, the experiment result demonstrates that the greedy method is more effective.

As the streaming of multimedia information is so prevalent in the 5G age, service providers must ensure that it can be seen without buffering. Wang et al. (2014); Zeydan et al. (2016) recommend proactive content caching utilising mobile caching so that data may be sent without delay downstream. However, they have not mentioned the advantage of caching data from user devices to the cloud, which will be done in this research.

2.6 Research Niche

All mobile edge computing and distributed storage research conducted before was performed based on an assumption. In this research, the simulation is done by using current and relevant real-time datasets along with a prototype. To ensure SLA for peer nodes, a hybrid blockchain rather than a single public blockchain is used. Most of the research conducted on distributed cloud storage and mobile edge computing is used for proactive caching of popular content. This research will focus on using them to upload the data from the end user to the cloud datacenter to reduce the propagation delay and processing delay. As shown in Table 1, this research suggests a unique method for reducing upload latency via mobile edge computing.

3 Methodology

The primary aim of this research is to create a decentralised cache system with the help of mobile edge devices. Public cloud users can participate in this system as a peer edge node, and incentive will be provided to those users for their resource usage as free cloud storage. The SLA between these peers and the cloud service provider can be ensured using a hybrid blockchain. This hybrid blockchain provides the user with trust in order for them to verify their resource usage on the public blockchain. The primary dataset used in this research analysis is the ookla speed test dataset, obtained from the AWS open dataset *Speedtest by Ookla Global Fixed and mobile network performance maps* (n.d.). According to the **CC BY-NC-SA 4.0** licence *Creative Commons Legal Code* (n.d.), the dataset used in this project may be freely utilised for non-commercial purposes. Moreover, the dataset contains no personally identifiable information about its users. The sample dataset is shown in Figure 5 and the numerical dataset split-up is shown in Figure 6. The geo location dataset is got from datahub *Country Polygons as GeoJSON* (n.d.), which is also open license under ”Open Data Commons Public Domain Dedication and License”.

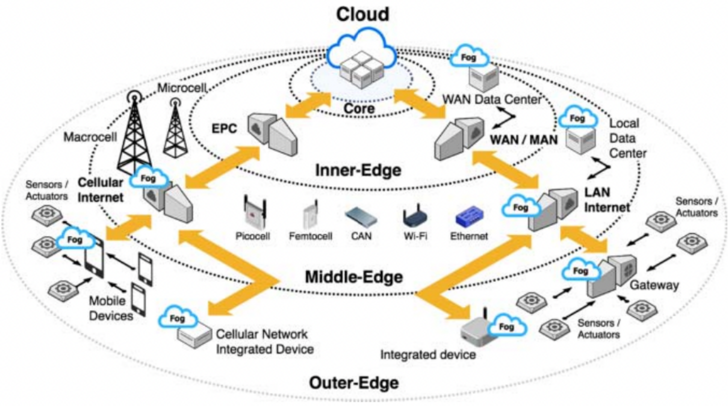


Figure 3: Motivation
Buyya (2019)

The cloud hierarchy diagram depicted in Figure 3 inspired this research idea. Instead of transmitting the data packets from the edge to the core, the data will be transferred to mobile edge devices located at the outer edge itself. By doing this, it will reduce the latency of the data transmission, which has been analysed and calculated in this research. As given in Equation 1, the latency is the combination of various delays like transmission delay, propagation delay, processing delay, and queuing delay. The distributed cache system proposed will reduce these delays, as explained below.

Transmission Delay: The transmission delay is completely dependent on the upload speed of the user end device, and it is not focused on in this research.

Processing Delay: As per the literature review, most of the cloud service providers compress the file on the end user device to reduce the transfer file size. This process will be offloaded to the mobile edge devices in this system. Huge file transmissions benefit from this offloading. In this research, for transferring a particular file, how many peer mobile edge devices are needed is calculated with the primary dataset. This device count is used to calculate the improvement in processing delay. From Equation 3, it is clear that *the improvement of compression time is directly proportional to the number of peer edge devices used to transfer the file.*

Queuing Delay: In this system the files are not sent from each user's device to a central cloud server for one-to-one file transfer. Instead, it'll send files to many peers, which will cut down on the waiting time.

Propagation Delay: The propagation delay is entirely dependent on the distance between the transmitter and the receiver. With the help of mobile edge devices, this distance can be reduced. The improvement in latency due to this reduction in distance is measured with the dataset. This primary dataset is preprocessed with the help of a preprocessing script, and the output from this script is sent to the simulator. From the simulator, it shows that India, the USA, and China are the top 3 countries with the most mobile devices in the primary dataset. *These 3 countries are used for evaluation case studies, to generate random points and calculate propagation delay, overall throughput, system cache size of this peer caching system, and ideal peer count.* If the random point is generated for the entire primary dataset, the calculation might not be accurate because the density of mobile edge peer devices will be lower in Antarctica compared to India. The outliers will be more if the whole world is considered, so the research is conducted on a country basis. For calculating the propagation delay between the random point and mobile edge peer devices, this Equation 2 is used, and for calculating the throughput Equation 4 is used.

$$Latency\ delays = Transmission + Propagation + processing + queue \tag{1}$$

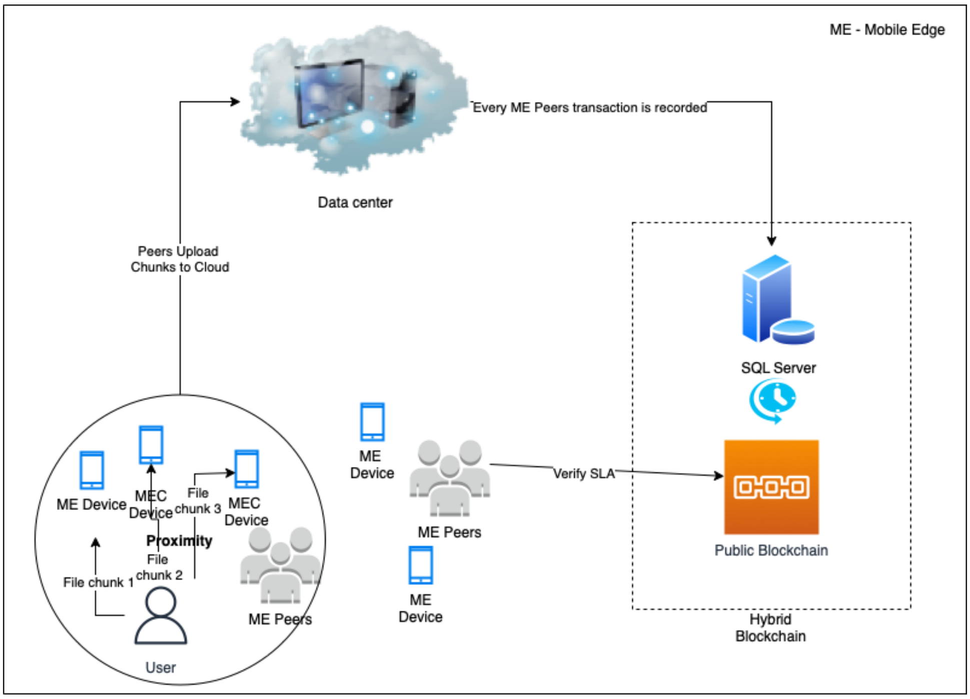
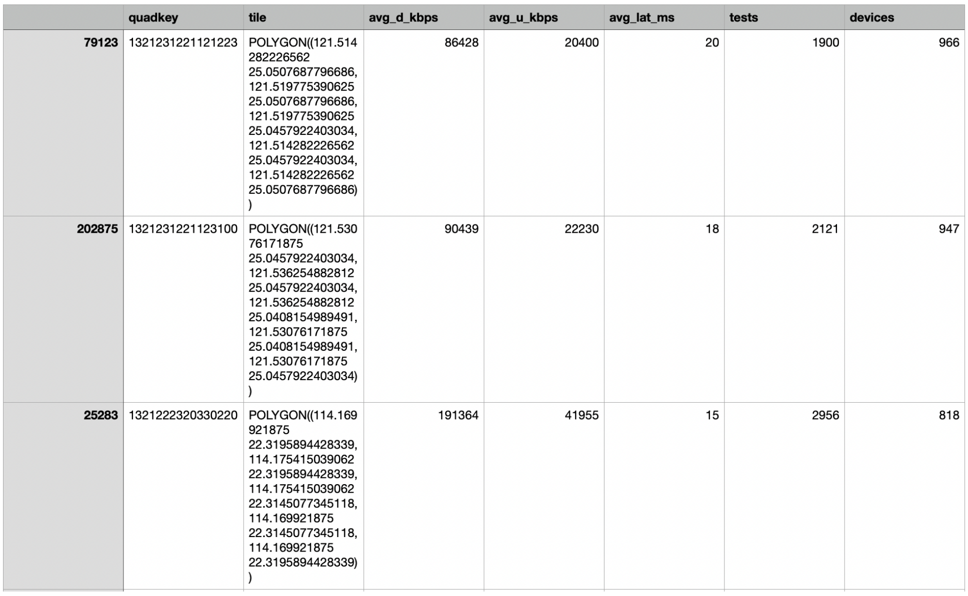
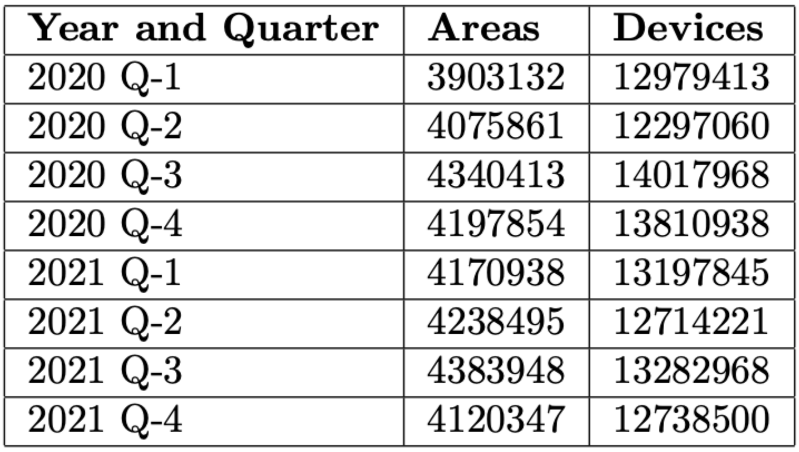


Figure 4: Overall System Flow



	quadkey	tile	avg_d_kbps	avg_u_kbps	avg_lat_ms	tests	devices
79123	1321231221121223	POLYGON((121.514282226562 25.0507687796686, 121.519775390625 25.0507687796686, 121.519775390625 25.0457922403034, 121.514282226562 25.0457922403034, 121.514282226562 25.0507687796686))	86428	20400	20	1900	966
202875	1321231221123100	POLYGON((121.53076171875 25.0457922403034, 121.536254882812 25.0457922403034, 121.536254882812 25.0408154989491, 121.53076171875 25.0408154989491, 121.53076171875 25.0457922403034))	90439	22230	18	2121	947
25283	1321222320330220	POLYGON((114.169921875 22.3195894428339, 114.175415039062 22.3195894428339, 114.175415039062 22.3145077345118, 114.169921875 22.3145077345118, 114.169921875 22.3195894428339))	191364	41955	15	2956	818

Figure 5: Sample Dataset



Year and Quarter	Areas	Devices
2020 Q-1	3903132	12979413
2020 Q-2	4075861	12297060
2020 Q-3	4340413	14017968
2020 Q-4	4197854	13810938
2021 Q-1	4170938	13197845
2021 Q-2	4238495	12714221
2021 Q-3	4383948	13282968
2021 Q-4	4120347	12738500

Figure 6: Dataset Describe

$$Propagation = distance\ between\ user\ and\ server/speed\ of\ light \tag{2}$$

$$ProcessingDelayImprovement = \frac{Total\ time\ for\ file\ compression}{Total\ time\ for\ file\ compression/No.\ of\ peer\ edge\ device} \tag{3}$$

$$Throughput = \sum_{j=1}^{n} \frac{ChunkLength}{Bandwidth_j}\ ,\ n = peer\ edge\ nodes \tag{4}$$

The overall system flow is given in Figure 33. In this research, the above-mentioned calculations are done by varying the transfer file size from 1 GB to 200 GB and recording them. These results are later plotted in graphs, histograms, box plots, scatter plots, probability distribution and compared with datacenters. Since the exact datacenter locations are not given by the cloud service provider, the calculations will be done for the following distances: 250 KM, 500 KM, and 1000 KM. These results are compared and drawn a conclusion about distributed peer cache system.

4 Design Specification

The research project is conducted in two phases.

In **phase 1**, the necessary calculations on the distributed cache system's performance are done by processing the primary dataset with the help of a preprocessing script and the simulator code. The architectural design of phase 1 is shown in Figure 7.

In **phase 2**, the performance improvement of the system based on any given latitude and longitude is calculated and displayed on a webpage. **MVC architecture** has been used for this website design and a simulation workflow of DAPP and peer device SLA has been done along with it. An Ethereum smart contract has been created to store the merkle hash root on the blockchain through DAPP. The architectural design of phase 2 is given in Figure 8.

The entire preprocessing script was running on an AWS instance, so it is difficult to set up the instance to run the preprocessing script each time. In order to solve this issue, an **AWS template is created with the help of a bash script for automation**. This bash-script will install all the necessary packages and software required to run the script, and it will also pull the script code from GitHub and start the script.

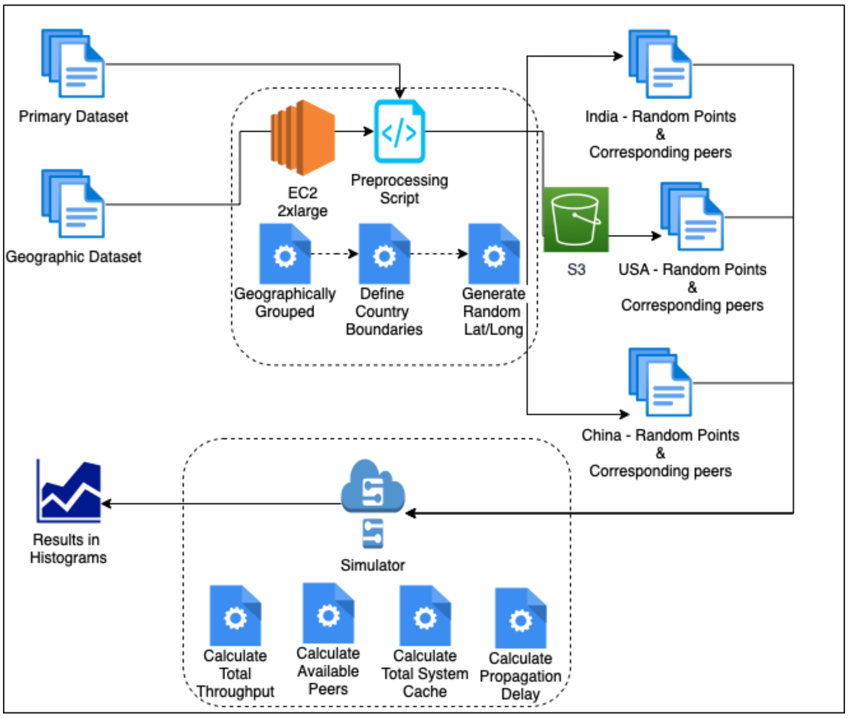


Figure 7: Phase1: Simulator Design

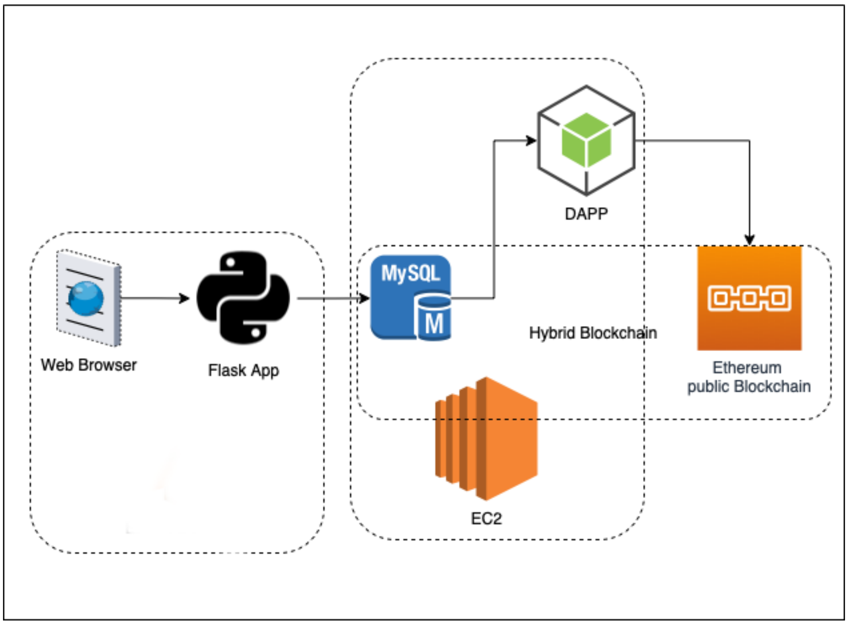


Figure 8: Phase2: Web App Design

5 Implementation

The preprocessing script is built with the Python programming language and the primary libraries used in it are Pandas for handling dataframes, Matplotlab for data visualisation and producing graphs, and Geo Pandas for managing location-related operations. The entire sequence flow of the preprocessing script is given in Figure 9. The preprocessing script is deployed to an AWS EC2 instance and 7 t2.X2Large instance type has been used. An AWS template is used for automating the instance setup and code setup. It took approximately 15 hours to generate 1000 random points along with their proximity peers within 1000 Km. These datasets have been exported as CSV files and transferred to the AWS S3 bucket. A total of 50,000 (approx) random coordinates have been collected for India and 5,000 (approx) random coordinates each for China and the USA. The simulator also uses the same libraries as the preprocessing script and takes the inputs from the S3 bucket uploaded by the preprocessing script. The sequence diagram of it is given in Figure 10.

For the web application, Python Flask has been used along with the Jinja template for the frontend. Geo Pandas and Pandas are used for handling location-based calculations and dataframe-related calculations. The primary dataset is preprocessed and used here. Once the calculation is done for demonstrating the behaviour of the system, a mock data is created in a MySql database about the peer devices. The entity relationship diagram of the database is given in Figure 12. The DAPP created in NodeJS will run every hour to pick up the transactions from the database and create a merkle root. This DAPP uses Web3 to communicate with the Ethereum smart contract. The smart contract has been developed at Remix and is written in Solidity. Metamask is used for ethereum transactions, and etherscan is used for verifying the smart contract deployment to the ethereum network. The database entries are hashed, and the root value is maintained at the merkleRoot entity. This value is updated to the Ethereum public test blockchain network with the help of the DAPP. The sequence diagram is shown in Figure 11. The

Transactions table holds all the resource usage information of the peers. The DAPP will select all of the hash column values in the transactions table for the previous hour and compute the textbfSHA2562 value using the textbfcrypto-js library, which it will store in the textbfMerkle_Root table. This value in Merkle_Root table is also uploaded to the Ethereum public blockchain network. A user can verify the SLA by checking the root hash in the blockchain and the DB, through the rest endpoints */blockchainChallenge* and */dbChallenge*.

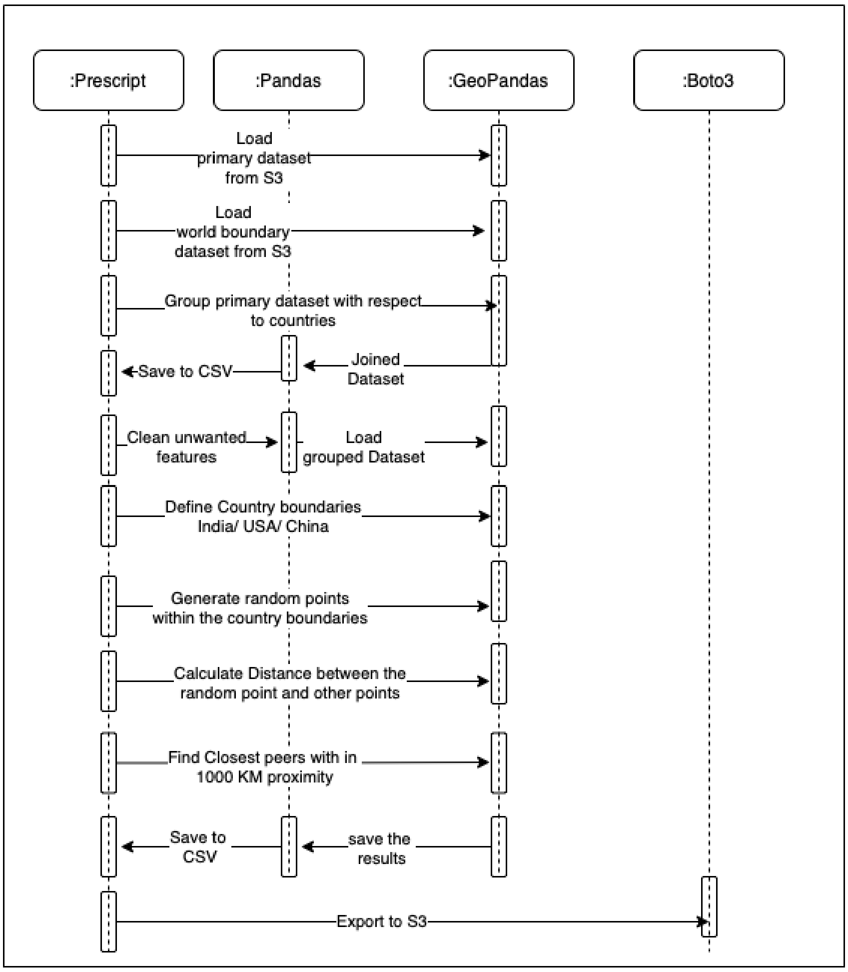


Figure 9: Prescript Sequence Diagram

6 Evaluation

The proposed system performance is compared with the traditional datacenters' performance. For this research, since the location of the cloud companies was abstracted due to security reasons, the distance between the user and the datacenter was assumed to be 250 km, 500 km, and 1000 km. The performance for transferring 1 GB, 10 GB, 100 GB, 200 GB is measured and compared. In the calculation, transmission delay is ignored because it is independent with respect to the user's network and will be the same for both systems. The equation used in the simulator is given in Equation 5 and Equation 6. The *qd* value is considered as 1 second, as it is the same in both equations. The calculation of the

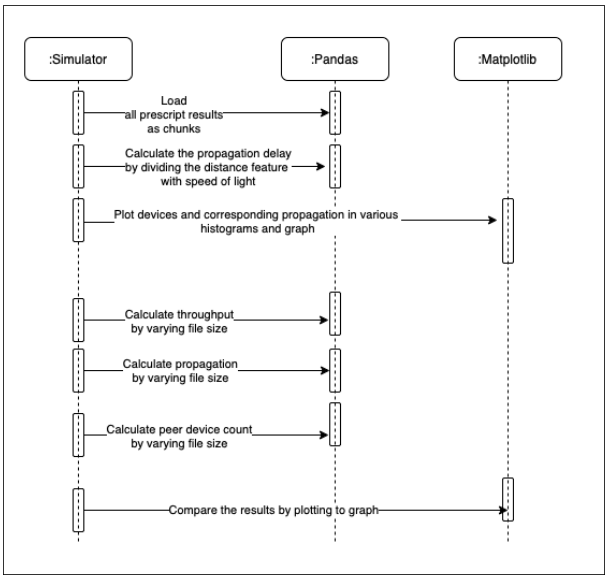


Figure 10: Simulator Sequence Diagram

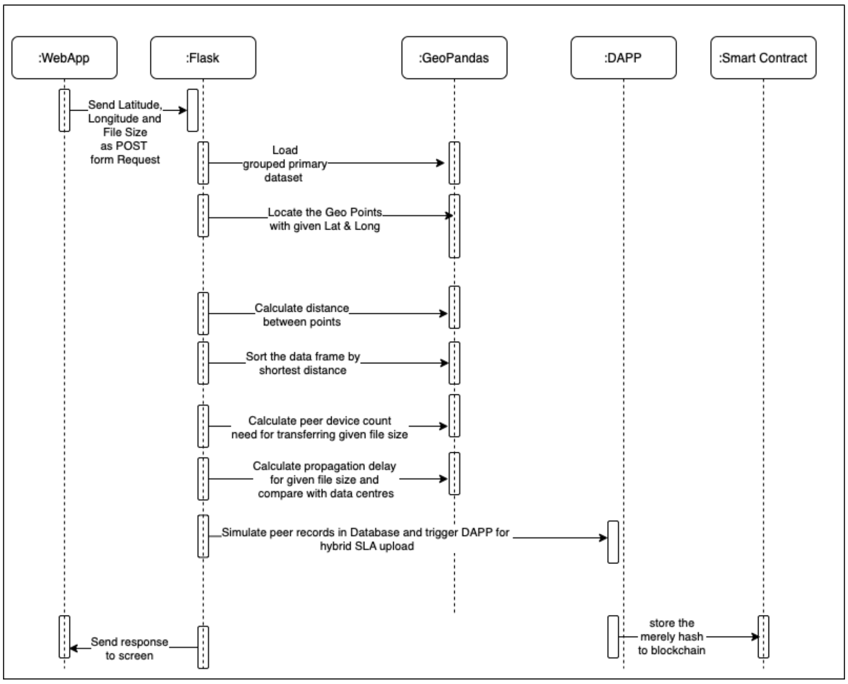


Figure 11: WebApp Sequence Diagram

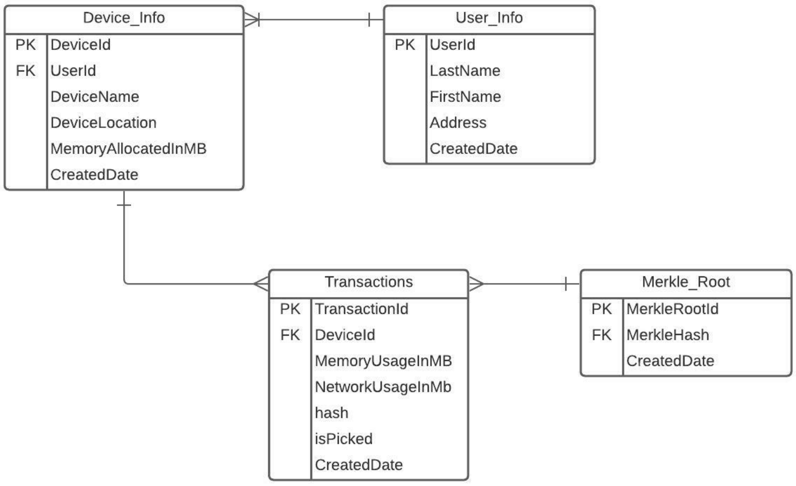


Figure 12: ER Diagram

compression time *ct* of various files is done in the arm processor and set to 0.08 seconds per MB for user devices and 0.2 seconds per MB for peer devices. The peer devices are mobile and will have less computational power than the users' desktop computers. For that reason, the compression time for the data centre is much less than the proposed system in the calculation. In the pre-script, a random coordinate is generated with respect to the country's boundary and a list of proximity peers is collected in a CSV file.

$$System\ Performance = \sum_{i=1}^{n} \frac{pd_i}{sl} + \frac{fs * ct}{nd} + \frac{qd}{nd} \tag{5}$$

$$Datacenter\ Performance = \frac{dd}{sl} + fs * ct + qd \tag{6}$$

where,
pd - Peer Distance
sl - Speed of Light
fs - file size
ct - Compression time
nd - No of Devices
qd - Queuing delay
dd - Datacenter distance

6.1 Case Study 1: India

51,490 random points have been generated within the Indian boundary and evaluated. From Figure 13 it is clear that the time difference between transferring a 1 GB file and 200 GB is only 0.1 minute, which is very small. In Figure 14, Figure 15, Figure 16, Figure 17, comparison of the proposed system and the traditional datacenter located at 250 Km distance is done. The performance of datacenters that were 500 km, 1000 km, or 250 km away was almost the same; the only difference was in the decimal point. So the data centre located at 250 km is taken for comparison.

The detailed description of the resultant dataset for transferring 1 GB, 10 GB, 100 GB, and 200 GB files is given in Table 2, Table 3, Table 4, Table 5. The propagation delay for transferring 100 GB and 200 GB files between the proposed system and datacenters located 250, 500, and 1000 kilometres apart is compared in Figure 18 and Figure 19. When it comes to moving 1 GB, 10 GB, 100 GB, and 200 GB files, the proposed system is 109.93%, 1848.23%, 18818.61%, and 36418.31% faster than a traditional datacenter.

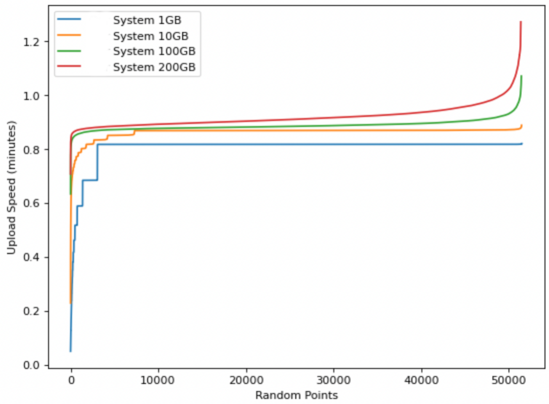


Figure 13: Proposed System Performance

Table 2: Data Transmission Comparison 1 GB - India

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
count	51490.0	51490.0	51490.0	51490.0	51490.0
mean	5.18	6269.93	0.003	48.20	101.19
std	2.00	24223.20	0.006	3.72	2.00
min	5.0	40.56	0.0	2.95	101.00
25%	5.0	416.11	0.00069	49.00	101.00
50%	5.0	1016.04	0.0016	49.00	101.00
75%	5.0	2940.05	0.0034	49.00	101.00
max	123.0	512331.93	0.20	49.18	219.00

Table 3: Data Transmission Comparison 10 GB - India

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	47.43	38587.18	0.054	51.71	1007.43
std	2.11	61483.41	0.06	1.46	2.11
min	47.0	810.48	0.0	13.69	1007.00
25%	47.0	8032.23	0.02	52.07	1007.00
50%	47.0	16545.31	0.036	52.09	1007.00
75%	47.0	39250.64	0.063	52.12	1007.00
max	189.0	733630.71	1.198	53.26	1149.00

Table 4: Data Transmission Comparison 100 GB - India

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	470.37	275775.61	1.20	53.23	10070.38
std	3.81	195185.47	1.01	1.12	3.81
min	469.0	12016.48	0.0	37.97	10069.00
25%	469.0	129135.56	0.58	52.65	10069.00
50%	469.0	223599.70	0.95	53.03	10069.00
75%	470.0	377743.84	1.46	53.56	10070.00
max	649.0	1728705.90	12.27	64.21	10249.00

Table 5: Data Transmission Comparison 200 GB - India

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	939.85	498429.07	3.08	55.15	20139.85
std	4.65	302129.03	2.36	2.40	4.65
min	938.0	30774.44	0.0	42.42	20138.00
25%	938.0	271210.89	1.59	53.67	20138.00
50%	938.0	437747.57	2.51	54.58	20138.00
75%	940.0	664540.61	3.77	55.87	20140.00
max	1165.0	2447664.57	24.10	76.27	20365.00

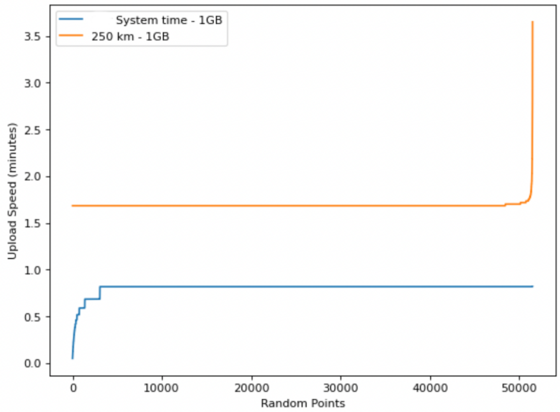


Figure 14: Comparison of the latency experienced by the user when transferring 1 GB data

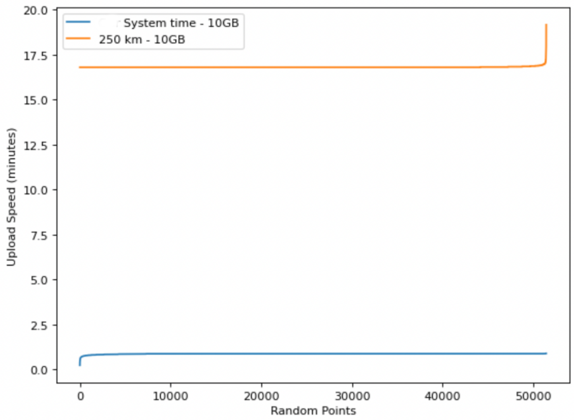


Figure 15: Comparison of the latency experienced by the user when transferring 10 GB data

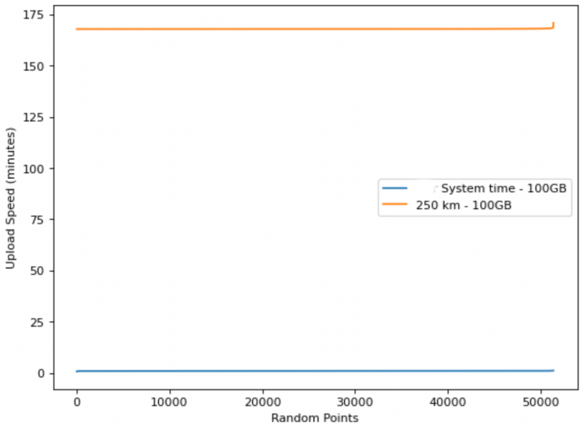


Figure 16: Comparison of the latency experienced by the user when transferring 100 GB data

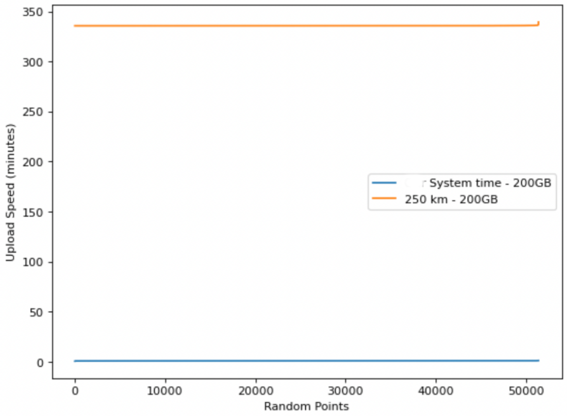


Figure 17: Comparison of the latency experienced by the user when transferring 200 GB of data

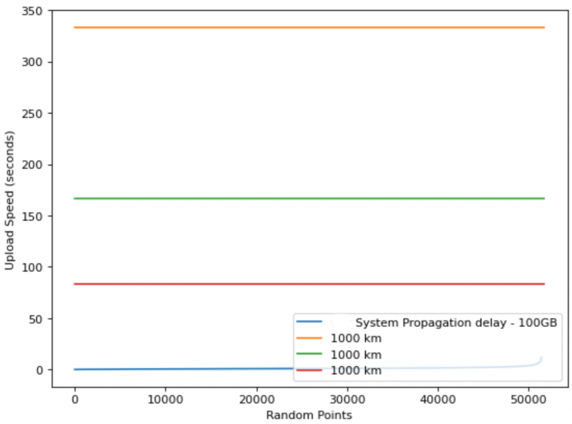


Figure 18: Comparison of propagation delay for transferring 100 GB data

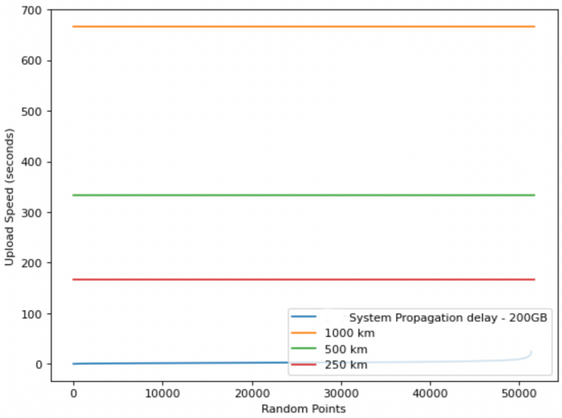


Figure 19: Comparison of propagation delay for transferring 200 GB data

6.2 Case Study 2: USA

Since computing country-specific random coordinates takes more time, for the USA, only 5124 points are generated and evaluated. By checking with India's dataset, 5000 points and 50,000 points don't change the result much. So, to reduce the cost of running the simulator in EC2 X2-Large instances, the random point is reduced to 5000. For transferring 1 GB, 10 GB, 100 GB, and 200 GB files, the proposed system outperforms traditional datacenters by 106.81%, 1868.53%, 18098.69%, and 36184.66%, respectively.

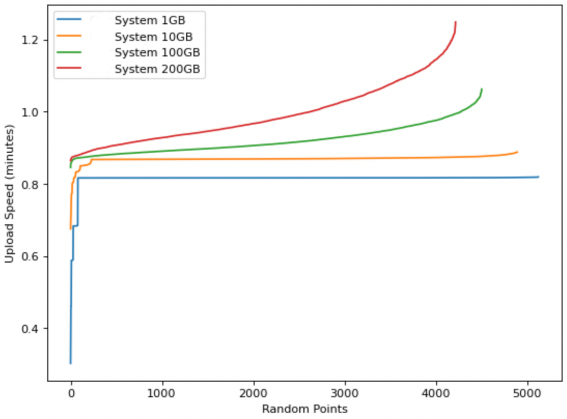


Figure 20: Proposed System Performance

Table 6: Data Transmission Comparison 1 GB - USA

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
count	5124.0	5124.0	5124.0	5124.0	5124.0
mean	5.03	7104.28	0.02	48.85	101.03
std	0.27	23815.26	0.02	1.50	0.27
min	5.00	18.23	0.00	18.14	101.00
25%	5.00	418.29	0.00	49.00	101.00
50%	5.00	1158.58	0.01	49.01	101.00
75%	5.00	3816.61	0.02	49.02	101.00
max	14.00	324320.68	0.17	49.17	110.00

Table 7: Data Transmission Comparison 10 GB - USA

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	47.10	49323.05	0.20	52.16	1007.10
std	0.64	73238.29	0.21	0.65	0.64
min	47.00	307.01	0.00	40.51	1007.00
25%	47.00	8794.99	0.06	52.12	1007.00
50%	47.00	21507.76	0.13	52.18	1007.00
75%	47.00	55026.16	0.27	52.32	1007.00
max	61.00	687935.43	1.29	53.30	1021.00

Table 8: Data Transmission Comparison 100 GB - USA

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	469.34	352452.27	3.19	55.33	10069.34
std	1.41	245170.93	2.32	2.33	1.41
min	469.00	6329.08	0.02	50.74	10069.00
25%	469.00	165938.72	1.42	53.57	10069.00
50%	469.00	294466.65	2.53	54.67	10069.00
75%	469.00	479133.39	4.52	56.68	10069.00
max	492.00	1934900.52	11.59	63.77	10092.00

Table 9: Data Transmission Comparison 200 GB - USA

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	938.36	657495.90	7.33	59.49	20138.36
std	1.38	381132.84	4.74	4.74	1.38
min	938.00	13443.64	0.09	51.89	20138.00
25%	938.00	377534.78	3.64	55.80	20138.00
50%	938.00	581771.76	6.16	58.31	20138.00
75%	938.00	875164.37	10.37	62.51	20138.00
max	964.00	2504007.66	22.76	74.93	20164.00

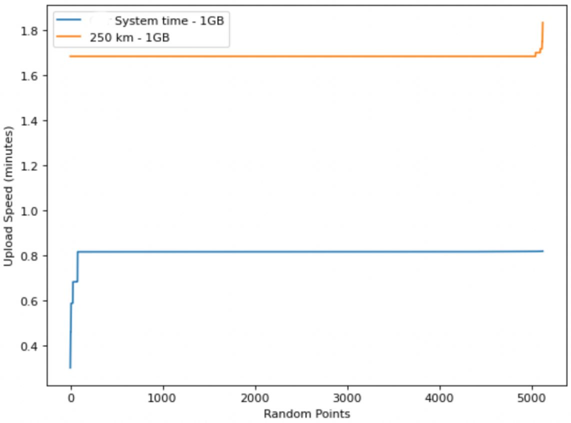


Figure 21: Comparison of the latency experienced by the user when transferring 1 GB data

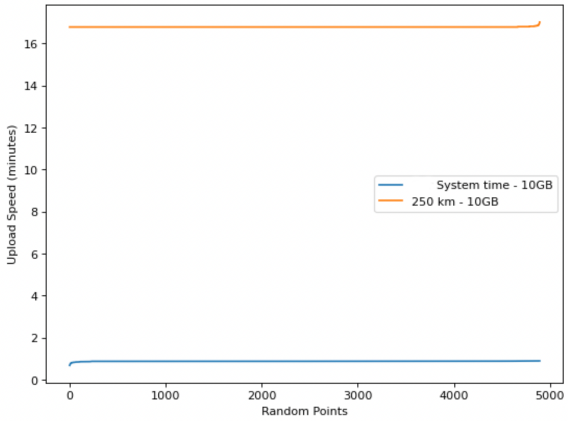


Figure 22: Comparison of the latency experienced by the user when transferring 10 GB data

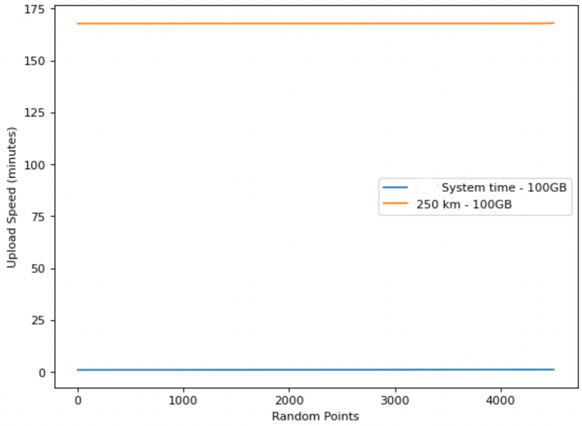


Figure 23: Comparison of the latency experienced by the user when transferring 100 GB data

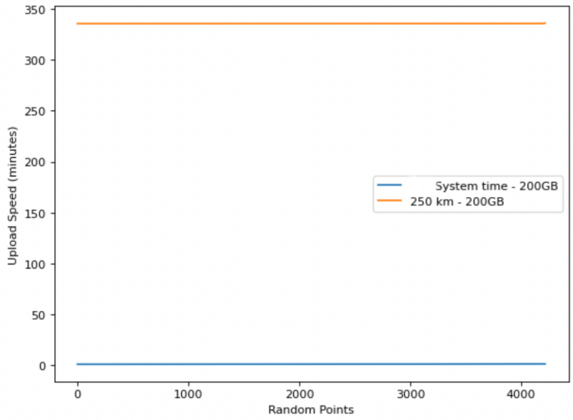


Figure 24: Comparison of the latency experienced by the user when transferring 200 GB of data

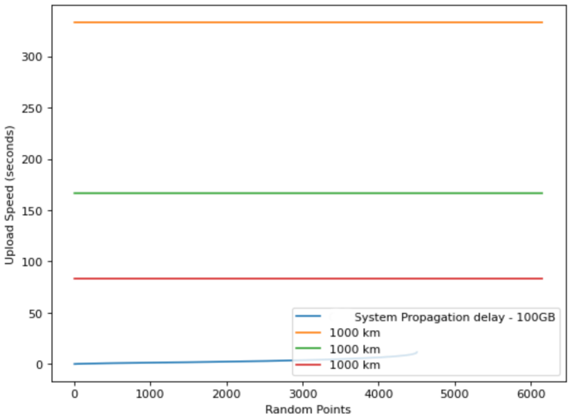


Figure 25: Comparison of propagation delay for transferring 100 GB data

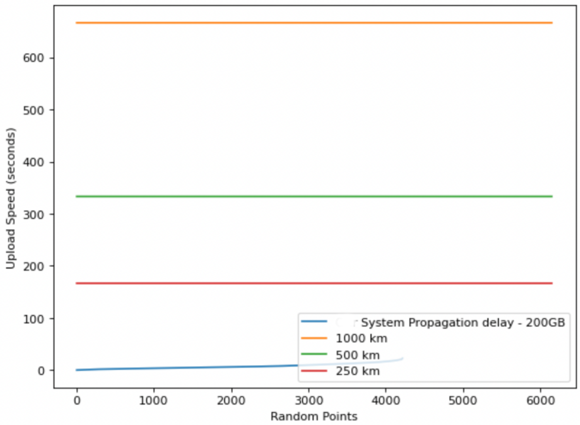


Figure 26: Comparison of propagation delay for transferring 200 GB data

6.3 Case Study 3: China

The results and comparison for random coordinates generated in China are given below. Due to cost and time constraints, even for China, only 5430 points are used for simulation. The average performance of the system over traditional data centres for transferring 1 GB, 10 GB, 100 GB, and 200 GB files is 106.85%, 1826.05%, 17952.43%, 33419.94%.

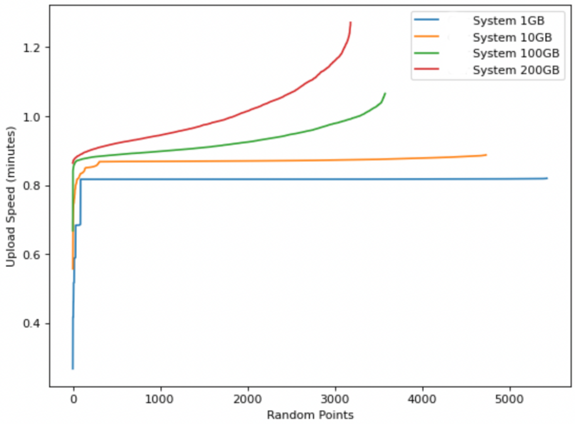


Figure 27: Proposed System Performance

Table 10: Data Transmission Comparison 1 GB - China

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
count	5430.00	5430.00	5430.00	5430.00	5430.00
mean	5.03	1714.24	0.03	48.84	101.03
std	0.30	14132.90	0.03	1.61	0.30
min	5.00	12.76	0.00	16.00	101.00
25%	5.00	139.78	0.00	49.00	101.00
50%	5.00	256.11	0.01	49.01	101.00
75%	5.00	483.79	0.04	49.04	101.00
max	16.00	274387.11	0.17	49.17	112.00

Table 11: Data Transmission Comparison 10 GB - China

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	47.16	12519.22	0.30	52.21	1007.16
std	0.92	36556.40	0.28	0.86	0.92
min	47.00	283.13	0.00	33.41	1007.00
25%	47.00	1877.28	0.08	52.13	1007.00
50%	47.00	3016.26	0.19	52.24	1007.00
75%	47.00	6231.15	0.45	52.49	1007.00
max	75.00	376910.95	1.28	53.24	1035.00

Table 12: Data Transmission Comparison 100 GB - China

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	469.65	79891.49	3.67	55.78	10069.65
std	3.71	99342.22	2.57	2.60	3.71
min	469.00	3773.72	0.02	40.07	10069.00
25%	469.00	22153.93	1.62	53.75	10069.00
50%	469.00	38969.04	2.95	55.06	10069.00
75%	469.00	92617.07	5.25	57.40	10069.00
max	636.00	798289.43	12.42	63.94	10236.00

Table 13: Data Transmission Comparison 200 GB - China

	devices	**throughput**	**propagation delay**	**system upload time**	**upload time 250 km**
mean	938.78	145445.06	7.94	60.08	20138.78
std	2.88	155567.17	5.04	5.04	2.88
min	938.00	8480.34	0.09	51.83	20138.00
25%	938.00	46803.34	3.94	56.08	20138.00
50%	938.00	84919.91	6.70	58.84	20138.00
75%	938.00	177058.32	11.09	63.22	20138.00
max	1017.00	1174916.38	24.15	76.32	20217.00

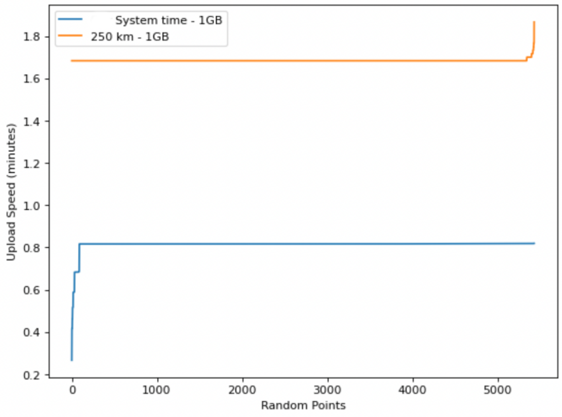


Figure 28: Comparison of the latency experienced by the user when transferring 1 GB data

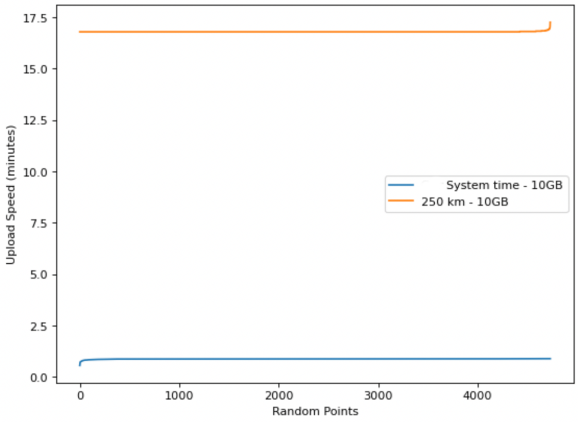


Figure 29: Comparison of the latency experienced by the user when transferring 10 GB data

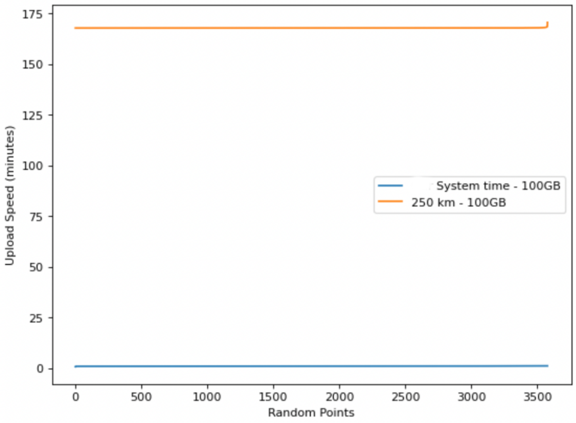


Figure 30: Comparison of the latency experienced by the user when transferring 100 GB data

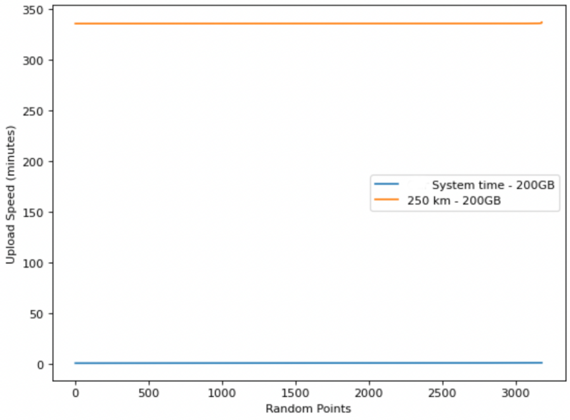


Figure 31: Comparison of the latency experienced by the user when transferring 200 GB of data

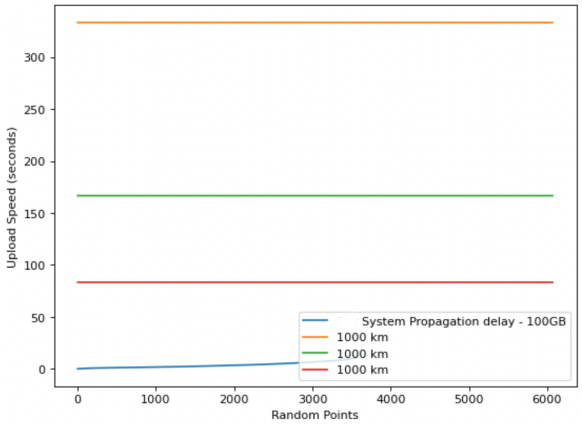


Figure 32: Comparison of propagation delay for transferring 100 GB data

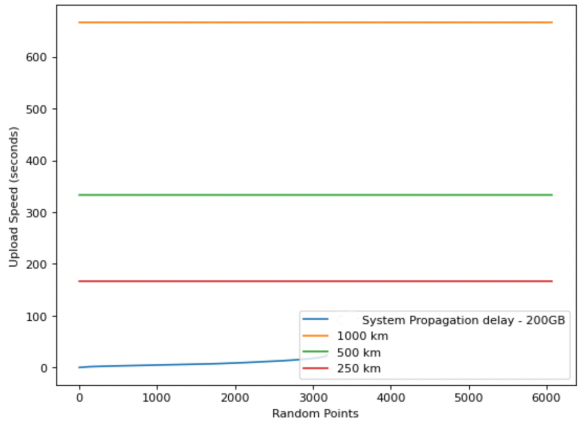


Figure 33: Comparison of propagation delay for transferring 200 GB data

6.4 Discussion

The evaluation result for the case study above clearly shows that the system completely dominates the traditional datacenter when the file size grows. Among the three countries taken for evaluation, the performance of India is better than the other two countries because of the high and dense mobile population. At any given point, the distance between proximity peer devices is very small for India. The performance comparison between the countries is given in Figure 34 and Figure 35. The average performance of our system over traditional data centres for transferring 1 GB, 10 GB, 100 GB, and 200 GB files is 107.86%, 1847.60%, 18,289.91%, 36007.63% (This is the average value of the top 3 countries).

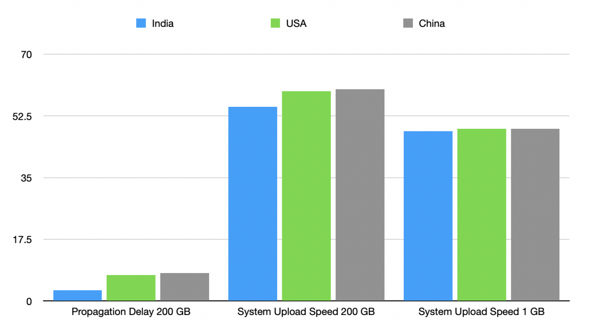


Figure 34: Comparison between Countries

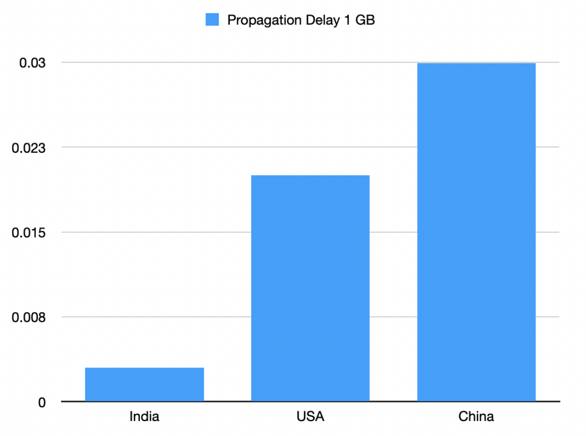


Figure 35: Comparison between Countries

7 Conclusion and Future Work

The results from the research clearly show that the performance of a distributed cache system is better than that of a traditional data center. In this research, instead of theoretically measuring the performance, a live dataset of 30 GB was preprocessed with the help of a python script running on 7 AWS T2-EC2-X2large instances and produced 500 GB of data. This 500 GB of data is stored in S3 and later fetched and passed as input to the simulator written in Python. The simulator produces the results in the form of dataframes and graphs. A website for checking the performance of the system at any given coordinate around the world is being developed. This website is integrated into the DAPP for simulating the transparent SLA. As part of the proposed hybrid blockchain network, the DAPP built in NodeJS will send the merkle root hash to the Ethereum Ropsten test network.

In the evaluation, only the top 3 countries with a high device count are considered. In the future, the same simulator code could be used to calculate the performance of this system in other countries as well. In this proposed system, the cache is used only for uploading the content from the user to the datacenter. But, this system could also be used for content caching, and the performance of that could be calculated in the future.

In this research, the system's advantage over propagation delay and processing delay was the primary focus. In the future, the actual queuing delay improvement through this system might be calculated, along with the advantage of sending data from one user to many peers instead of from one user to one server, could also be calculated.

References

Benet, J. (2014). Ipfs-content addressed, versioned, p2p file system, *arXiv preprint arXiv:1407.3561* .

Benet, J. and Greco, N. (2018). Filecoin: A decentralized storage network, *Protoc. Labs* pp. 1–36.

Bonomi, F., Milito, R., Zhu, J. and Addepalli, S. (2012). Fog computing and its role in the internet of things, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, Association for Computing Machinery, New York, NY, USA, p. 13–16. CORE Ranking=A^*, Address =Helsinki Finland 17 August 2012.
URL: *https://doi.org/10.1145/2342509.2342513*

Buyya, R. (2019). *Hierarchy of Fog and Edge Computing (Page 38).*, Wiley Series on Parallel and Distributed Computing, p. 38–38.

Computing classification system (n.d.). url=https://dl.acm.org/ccs.
URL: *https://dl.acm.org/ccs*

Country Polygons as GeoJSON (n.d.). url=https://datahub.io/core/geo-countries.
URL: *https://datahub.io/core/geo-countries*

Creative Commons Legal Code (n.d.). url=https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.
URL: *https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode*

Irvine, D. (2010). Maidsafe distributed file system.

Kuang, L., Gong, T., OuYang, S., Gao, H. and Deng, S. (2020). Offloading decision methods for multiple users with structured tasks in edge computing for smart cities, *Future Generation Computer Systems* **105**: 717–729. JCR Impact Factor 2021: 7.138.
URL: *https://www.sciencedirect.com/science/article/pii/S0167739X19323325*

Li, J., Wu, J. and Chen, L. (2018). Block-secure: Blockchain based scheme for secure p2p cloud storage, *Information Sciences* **465**: 219–231. JCR Impact Factor 2021: 6.795.
URL: *https://www.sciencedirect.com/science/article/pii/S0020025518305012*

Li, J., Wu, J., Jiang, G. and Srikanthan, T. (2020). Blockchain-based public auditing for big data in cloud storage, *Information Processing Management* **57**(6): 102382. JCR Impact Factor 2021: 6.222.
URL: *https://www.sciencedirect.com/science/article/pii/S0306457320308773*

Li, W., Wu, J., Cao, J., Chen, N., Zhang, Q. and Buyya, R. (2021). Blockchain-based trust management in cloud computing systems: a taxonomy, review and future directions, *Journal of Cloud Computing* **10**(1): 35. JCR Impact Factor 2021: 3.222.
URL: *https://doi.org/10.1186/s13677-021-00247-5*

Li, Z., Zhang, Y., Liu, Y., Xu, T., Zhai, E., Liu, Y., Ma, X. and Li, Z. (2019). A quantitative and comparative study of network-level efficiency for cloud storage services, *ACM Trans. Model. Perform. Eval. Comput. Syst.* **4**(1). JCR Impact Factor 2021: 2.17.
URL: *https://doi.org/10.1145/3274526*

Liu, D., Chen, B., Yang, C. and Molisch, A. F. (2016a). Caching at the wireless edge: design aspects, challenges, and future directions, *IEEE Communications Magazine* **54**(9): 22–28. JCR Impact Factor 2021: 9.31.

Liu, D., Chen, B., Yang, C. and Molisch, A. F. (2016b). Caching at the wireless edge: design aspects, challenges, and future directions, *IEEE Communications Magazine* **54**(9): 22–28. JCR Impact Factor 2021: 5.938.

Mach, P. and Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading, *IEEE Communications Surveys Tutorials* **19**(3): 1628–1656. JCR Impact Factor 2021: 25.249.

Nguyen, D. C., Pathirana, P. N., Ding, M. and Seneviratne, A. (2020). Integration of blockchain and cloud of things: Architecture, applications and challenges, *IEEE Communications Surveys Tutorials* **22**(4): 2521–2549. JCR Impact Factor 2021: 25.249.

Ni, J., Zhang, K. and Vasilakos, A. V. (2021). Security and privacy for mobile edge caching: Challenges and solutions, *IEEE Wireless Communications* **28**(3): 77–83. JCR Impact Factor 2021: 11.979.

Sharma, P., Jindal, R. and Borah, M. D. (2020). Blockchain technology for cloud storage: A systematic literature review, *ACM Comput. Surv.* **53**(4). JCR Impact Factor 2021: 10.282.
URL: *https://doi.org/10.1145/3403954*

Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L. (2016). Edge computing: Vision and challenges, *IEEE Internet of Things Journal* **3**(5): 637–646. JCR Impact Factor 2021: 9.471.

Speedtest by Ookla Global Fixed and mobile network performance maps (n.d.). url=https://registry.opendata.aws/speedtest-global-performance/.
URL: *https://registry.opendata.aws/speedtest-global-performance/*

Vorick, D. and Champine, L. (2014). Sia: Simple decentralized storage, *Retrieved May* **8**: 2018.

Wang, X., Chen, M., Taleb, T., Ksentini, A. and Leung, V. C. (2014). Cache in the air: exploiting content caching and delivery techniques for 5g systems, *IEEE Communications Magazine* **52**(2): 131–139. JCR Impact Factor 2021: 9.619.

Wang, X., Yu, P., Yu, G., Zha, X., Ni, W., Liu, R. P. and Guo, Y. J. (2019). A high-performance hybrid blockchain system for traceable iot applications, *in* J. K. Liu and X. Huang (eds), *Network and System Security*, Springer International Publishing, Cham, pp. 721–728. CORE Ranking=B, Address =Sapporo, Japan, December 15–18, 2019.

Zeydan, E., Bastug, E., Bennis, M., Kader, M. A., Karatepe, I. A., Er, A. S. and Debbah, M. (2016). Big data caching for networking: moving from cloud to edge, *IEEE Communications Magazine* **54**(9): 36–42. JCR Impact Factor 2021: 9.619.