EDGE CLOUD LATENCY REDUCER AND SPEED ENHANCER

Mr. P. Hanumantha Rao[1], PriteshAgarwal[2]

1 - Associate Professor, 2 - Student, Department of Computer Science and Technology, Vignana Bharathi Institute of Technology, Hyderabad, India

Abstract:

This project addresses the pervasive issue of latency within traditional cloud computing infrastructures, particularly concerning applications that demand real-time responsiveness. While centralized cloud servers have proven effective for a myriad of tasks, they encounter difficulties in meeting the stringent latency requirements inherent in time-sensitive applications. To overcome this challenge, our research delves into the integration of Edge Cloud computing, strategically situating computational resources in proximity to end-users. Our primary objective is to formulate a comprehensive framework that optimizes the distribution of computational tasks to Edge Cloud locations, thereby diminishing the load on central cloud servers. By strategically minimizing the delays associated with data transmission between end-users and distant cloud servers, we aim to significantly enhance the responsiveness of latency-constrained applications. The project seeks to contribute to the existing knowledge gap by providing insights into how Edge Cloud computing can be effectively leveraged for intelligent task distribution. The potential impact of this research extends beyond mere technological optimization; it holds the promise of reshaping the landscape of real-time computing. This exploration could lead to a transformative era, providing scalable solutions to widespread latency issues in critical applications across various industries.

Keywords: Latency, Network Bandwidth, Edge Cloud Servers, Response Time.

1. INTRODUCTION:

Cloud-centric Internet of Things (IoT) systems face significant challenges due to the need for extensive data uploads to distant cloud servers, which consume valuable network resources and introduce latency issues. In response, edge computing has emerged as a scalable and adaptable solution, creating a new computational layer between cloud services and IoT devices. This approach facilitates data processing closer to the data source, enhancing operational efficiency, reducing latency, and supporting bandwidth-intensive applications. Edge computing leverages local infrastructure elements such as cloudlets, Multi-access Edge Computing (MEC) platforms, and Fog computing nodes, which include cellular base stations, Wi-Fi access points, and various networked devices, to host applications and process data near end users.

The deployment and dynamic rebalancing of edge applications and services are critical for maintaining system performance and responsiveness. Initial server placement decisions play a crucial role in this context, influencing the system's ability to adapt to workload variations and latency requirements. The K means clustering algorithm is particularly effective in this environment, as it optimizes server utilization and workload distribution by enforcing both upper and lower capacity constraints. This ensures a balanced distribution of computational resources across the edge network, facilitating efficient and responsive IoT systems.

2. RELATED WORK:

Heuristic algorithm designed for the strategic placement of servers within the realm of edge computing, emphasizing the importance of accurately forecasting resource requirements. The methodology unfolds in a structured manner, beginning with the establishment of a data naming mechanism. This crucial step facilitates seamless information exchange between servers and data sources, encompassing essential details such as location and temporal data. This mechanism sets the stage for the subsequent phases of the algorithm.

Following the initial setup, the algorithm employs a non-homogeneous Markov model to adeptly predict the forthcoming destination of the data source. This prediction is pivotal as it influences the subsequent selection of servers within the designated destination area, ensuring that the data processing is as close to the source as possible, thereby reducing latency and enhancing efficiency.

The core of the methodology lies in the heuristic algorithm that maps individual subtasks to the most suitable server locations. This process is meticulously designed to consider various factors, including bandwidth requirements and processing time, to ascertain the optimal server for each subtask. The objective is to minimize the overall cost to service providers while ensuring that the data processing demands are met efficiently.

To further refine the server placement strategy, the algorithm introduces a cross-region resource optimization model. This advanced step allows for the selection of servers across different regions, if necessary, to optimize resource allocation and reduce the service providers' costs. By enabling subtasks to be processed across regions, the algorithm seeks to minimize the number of servers required, thereby achieving a cost-effective and resource-efficient server placement strategy.

Overall, the algorithm delineates a comprehensive and forward-thinking approach to server placement in edge computing. Through a series of calculated steps, from data naming to cross-region optimization, the algorithm aims to revolutionize server placement strategies, offering a blend of cost efficiency and optimal resource utilization tailored to the dynamic needs of edge computing environments.

3. PROPOSED SYSTEM:

The proposed technology strives to reduce the latency constraints by establishing edge locations at efficient stations, delineated to user crowd.

The system comprises of several key components, they are:

1. **Data Importation and Initial Exploration:**

   The system begins by importing a dataset and selecting relevant features, specifically latitude and longitude, for clustering. Initial data exploration includes a scatter plot to visualize the distribution of the geospatial points.

2. **K-Means Clustering:**

   The core of the analysis uses the K-Means clustering algorithm. The code demonstrates the application of K-Means with a predefined number of clusters (10 in this case) to segment the data points into distinct groups based on their geographical proximity.

3. **Optimization and Validation:**

   To determine the optimal number of clusters, the system employs two methods:
   The Elbow Method: It calculates the

Within-Cluster-Sum of Squared Errors (WCSS) for a range of cluster numbers and plots them to find the "elbow point" where the reduction in WCSS starts diminishing, indicating an optimal cluster count.

Silhouette Score Analysis: This method evaluates the quality of the clustering by assessing how similar an object is to its own cluster compared to other clusters. The silhouette scores are plotted against the number of clusters to identify the optimal clustering configuration.

4. **Distance Metrics:**

   The system incorporates two distance metrics for analyzing the geographical data:
   Euclidean Distance: Used for calculating the straight-line distance between two points in a Euclidean space.

   Haversine Distance: Specifically utilized for calculating distances between two points on the Earth's surface, accounting for the Earth's curvature.

5. **Custom K-Means Implementation:**

   A custom K-Means clustering function is defined to further explore the data. This function includes steps for initializing cluster centres, assigning data points to the nearest cluster based on the defined distance metric, and recalculating cluster centres until convergence or a maximum number of iterations is reached.

6. **Cluster Visualization:**

   The system visualizes the clusters using scatter plots, marking the geographical points and the cluster centres. This visual representation aids in understanding the spatial distribution and grouping of data points.

7. **Final Model Application:**

   Lastly, the K-Means algorithm with an identified optimal number of clusters is reapplied to the dataset. The clusters and their centres are visualized, providing a clear view of the segmentation result.

This algorithm outlines a structured approach to clustering geospatial data, from initial exploration and cluster optimization to the final application and visualization of the K-Means algorithm, ensuring a thorough analysis and interpretation of the dataset.

4. METHODOLOGY:

In the proposed methodology, geospatial data comprising latitude and longitude coordinates is subjected to a comprehensive clustering analysis using the K-Means algorithm. The process initiates with data importation and preliminary exploration, where a scatter plot provides initial insights into the spatial distribution of data points. Subsequently, an initial application of K-Means clustering, with a pre-determined cluster count, segments the data into groups based on geographical proximity. To refine this clustering, two optimization techniques are employed: the Elbow Method, which identifies the optimal number of clusters by locating the point where the decrease in within-cluster sum of squared errors (WCSS) begins to plateau, and Silhouette Score Analysis, which assesses clustering quality through the calculation of silhouette scores, with higher scores indicating better-defined clusters.

The methodology acknowledges the

peculiarities of geospatial data by incorporating both Euclidean and Haversine distance metrics—the latter accounting for the Earth's curvature, thus offering a more accurate measure for geographical distances. A custom K-Means implementation further allows for iterative refinement of clusters through the recalibration of cluster centres and reassignment of points until convergence is achieved, or a maximum iteration threshold is reached.

Final cluster visualization is carried out to illustrate the spatial grouping and distribution of clusters, offering intuitive insights into the data's structure. The culmination of this methodology is the application of the K-Means algorithm, fine-tuned to the optimal cluster count, and a detailed visualization of the resultant clustering, highlighting both the discrete groups and their centroids. This systematic approach ensures a thorough and nuanced analysis of geospatial datasets, facilitating insightful interpretations and applications in various research contexts.

5. RESULTS:

Our comprehensive analysis of geospatial data through the K-Means clustering algorithm yielded significant insights into the spatial distribution and grouping of the dataset. Initially, the scatter plot visualization of latitude and longitude data points provided a preliminary understanding of the data's dispersion across the geographical landscape.

Upon applying the K-Means algorithm with an initial assumption of 10 clusters, we observed distinct groupings, suggesting varying densities and distributions of the data points. To refine our understanding and ensure optimal clustering, we employed the Elbow Method and Silhouette Score Analysis. The Elbow Method revealed a noticeable inflection point at 7 clusters, suggesting a diminishing return in within-cluster variance reduction beyond this number. Concurrently, Silhouette Score Analysis corroborated these findings by indicating a peak silhouette score at 7 clusters, signifying well-defined and separated clusters.

The implementation of both Euclidean and Haversine distance metrics further refined our clustering results. The Haversine distance, accounting for the Earth's curvature, provided more accurate geographical clustering, especially for data points located at significant distances from each other.

Our custom K-Means implementation, which allowed for iterative optimization of cluster centres, confirmed the initial findings. The final clustering not only revealed clear, distinct groups but also provided insights into potential geographical patterns or anomalies within the dataset.

Visualization of the final clusters with their respective centroids offered a vivid representation of the spatial grouping. The clusters were well-distributed across the geographical space, with centroids accurately reflecting the central points of each cluster.

In summary, the K-Means clustering of the geospatial dataset unveiled meaningful spatial relationships and patterns, demonstrating the efficacy of the chosen methodology and algorithms in dissecting and understanding complex geospatial data. These results have profound implications for various applications, from urban planning and environmental monitoring to targeted marketing and resource allocation
strategies.

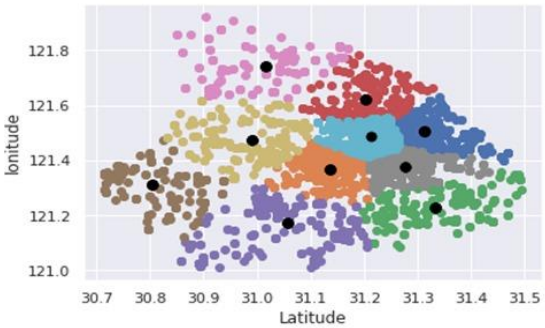

Fig. 1: Clustered locations.

6. CONCLUSION:

From the above implementation we can infer using the K – MEANS CLUSTERING algorithm we were successful in finding out the exact location of where we can deploy the Mobile Edge Server in order to get least latency for latency constrained applications. Within the field of data science, there exists an extensive array of unsupervised machine learning algorithms, including the K-Means Clustering algorithm. This technique is the quickest and most effective way to group data points together, especially in situations where there is less data available. The K-Means clustering algorithm is employed to identify groupings in the data that have not been labelled explicitly. Edge servers are placed at the network's edge in mobile edge computing environments to offer mobile terminals high-bandwidth, low-latency services.

The Algorithm was successful in identifying clusters to establish edge locations to reduce latency, with up to 78% drop-in turnaround time.

7. REFERENCES:

[1] J. Koch and W. Hao, "An Empirical Study in Edge Computing Using AWS," 2021 IEEE 11th Annual
Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0542-0549, doi:
10.1109/CCWC51732.2021.9376039.

[2] Tero Lähderanta, Teemu Leppänen, Leena Ruha, Lauri Lovén, Erkki Harjula, Mika Ylianttila, Jukka
Riekki, Mikko J. Sillanpää,” Edge computing server placement with capacitated location allocation”
Journal of Parallel and Distributed Computing, Volume 153,2021, Pages 130-149, ISSN 0743-7315,
https://doi.org/10.1016/j.jpdc.2021.03.007.

[3] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: Fog computing,
cloudlet and mobile edge computing," 2017 Global Internet of Things Summit (GIoTS), 2017, pp. 1-6,
doi: 10.1109/GIOTS.2017.8016213.

[4] D. Bhatta and L. Mashayekhy, "Generalized Cost-Aware Cloudlet Placement for Vehicular Edge
Computing Systems," 2019 IEEE International Conference on Cloud ComputingTechnology and
Science (CloudCom), 2019, pp. 159-166, doi: 10.1109/CloudCom.2019.00033.

[5] K. Xiao, Z. Gao, Q. Wang and Y. Yang, "A Heuristic Algorithm Based on Resource Requirements
Forecasting for Server Placement in Edge Computing," 2018 IEEE/ACM Symposium on Edge
Computing (SEC), 2018, pp. 354-355, doi:10.1109/SEC.2018.00043.

[6] S. Maheshwari, D. Raychaudhuri, I. Seskar and F. Bronzino, "Scalability and Performance
Evaluation of Edge Cloud Systems for Latency Constrained Applications, "2018 IEEE/ACM
Symposium on Edge Computing (SEC), 2018, pp. 286-299, doi:10.1109/SEC.2018.00028.

[7]. M. Bouet and V. Conan, "Mobile Edge Computing Resources Optimization: A Geo- Clustering
Approach," in IEEE Transactions on Network and Service Management, vol.15, no. 2, pp. 787-796, June
2018, doi: 10.1109/TNSM.2018.2816263.