Documentation: Fire Detection Application

Introduction:

We've developed a fire detection application with the primary function of receiving user alerts and analyzing them to identify fires. This document outlines the process and algorithms involved in detecting and managing fire alerts.

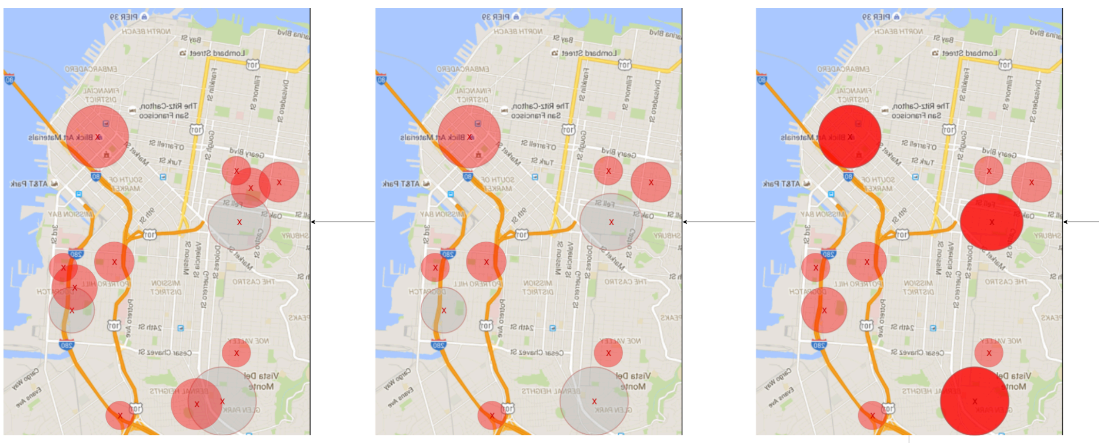
by : Ali alsuleman

Alert Reception and Analysis:

- The application receives alerts from users indicating potential fire outbreaks.
- These alerts contain coordinates and a radius denoting the area of concern.
- Upon receiving a fire alert, a dormant node is created to signify the event's occurrence in that area.

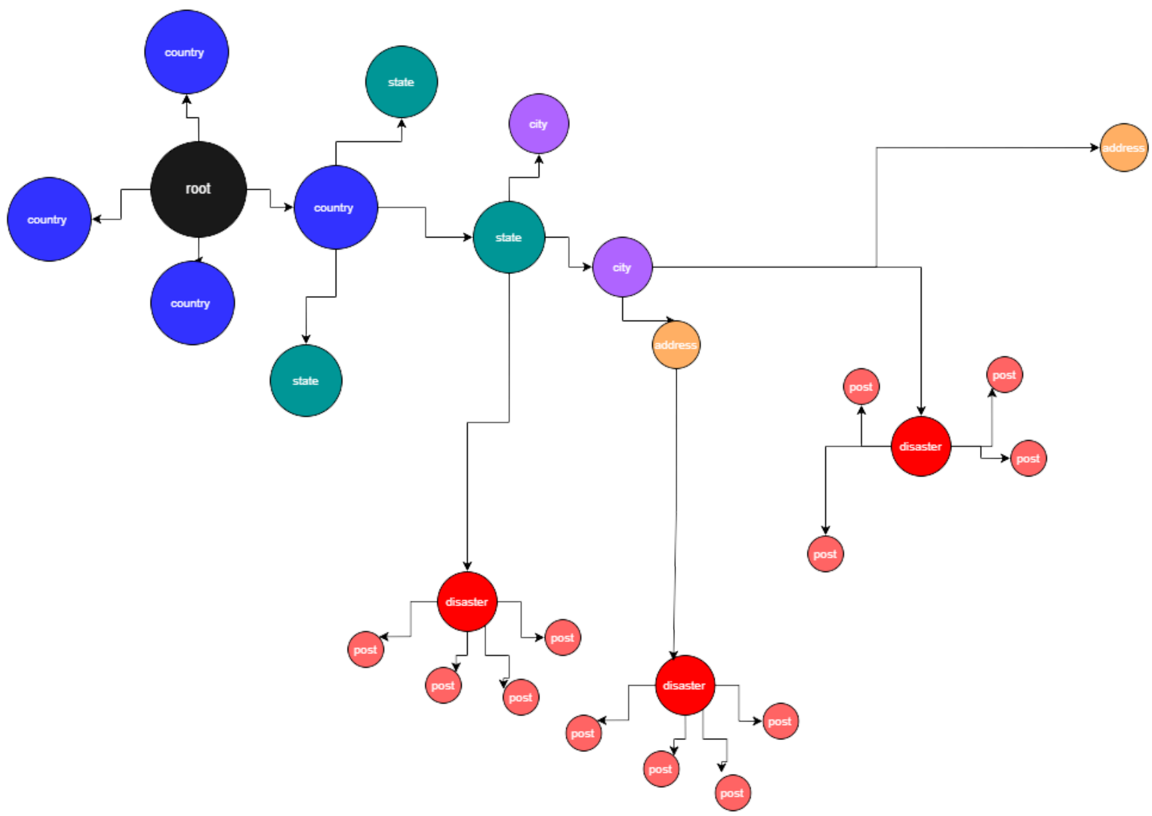
Event Analysis and Intersection:

- To analyze fire alerts, we need to identify all events within the same area that may intersect with the fire.
- Each alert triggers a database query to retrieve all alerts within the specified range corresponding to the area of fire spread.
- Intersecting events are merged based on the common area between them. (See Algorithm Explanation below)



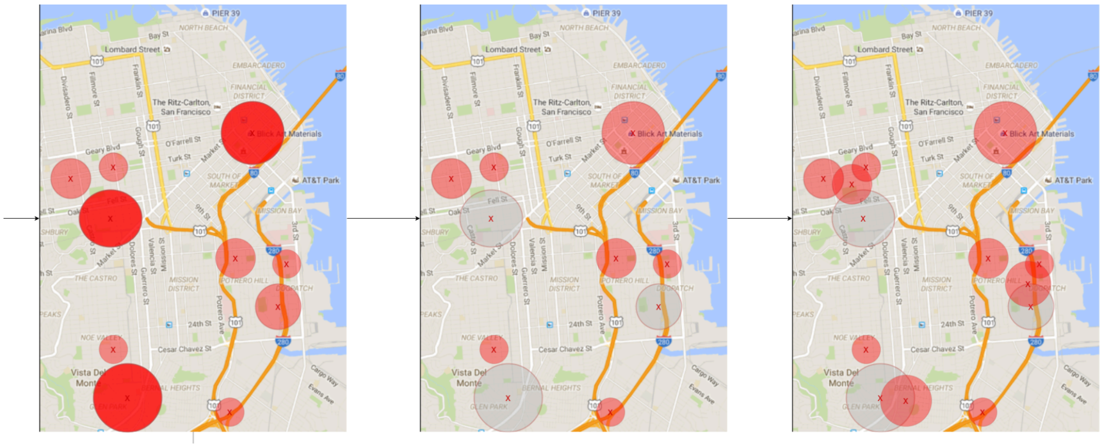
Efficient Data Retrieval:

- As repeated database queries become costly, we adopt a caching principle to utilize previously retrieved data for future analysis.
- Leveraging event metadata, we construct a tree structure from the root based on the world identity down to the smallest geographical unit (address).
- This tree structure enables efficient navigation to locate relevant event nodes, significantly reducing computational complexity. Refer to <img2> for a graphical representation.



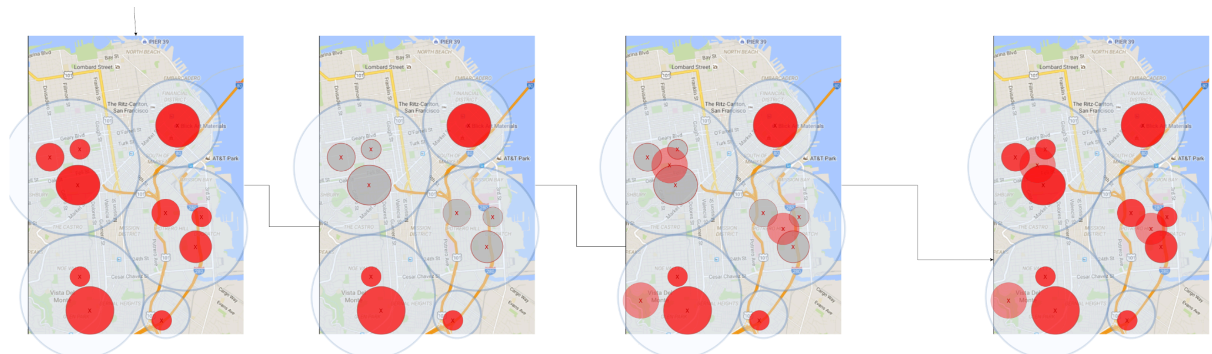
Memory Management:

- Efficient memory management is crucial, especially during high data influx periods.
- Random deletion poses challenges as it may lead to loss of important data that is intersecting with newly arrived alerts

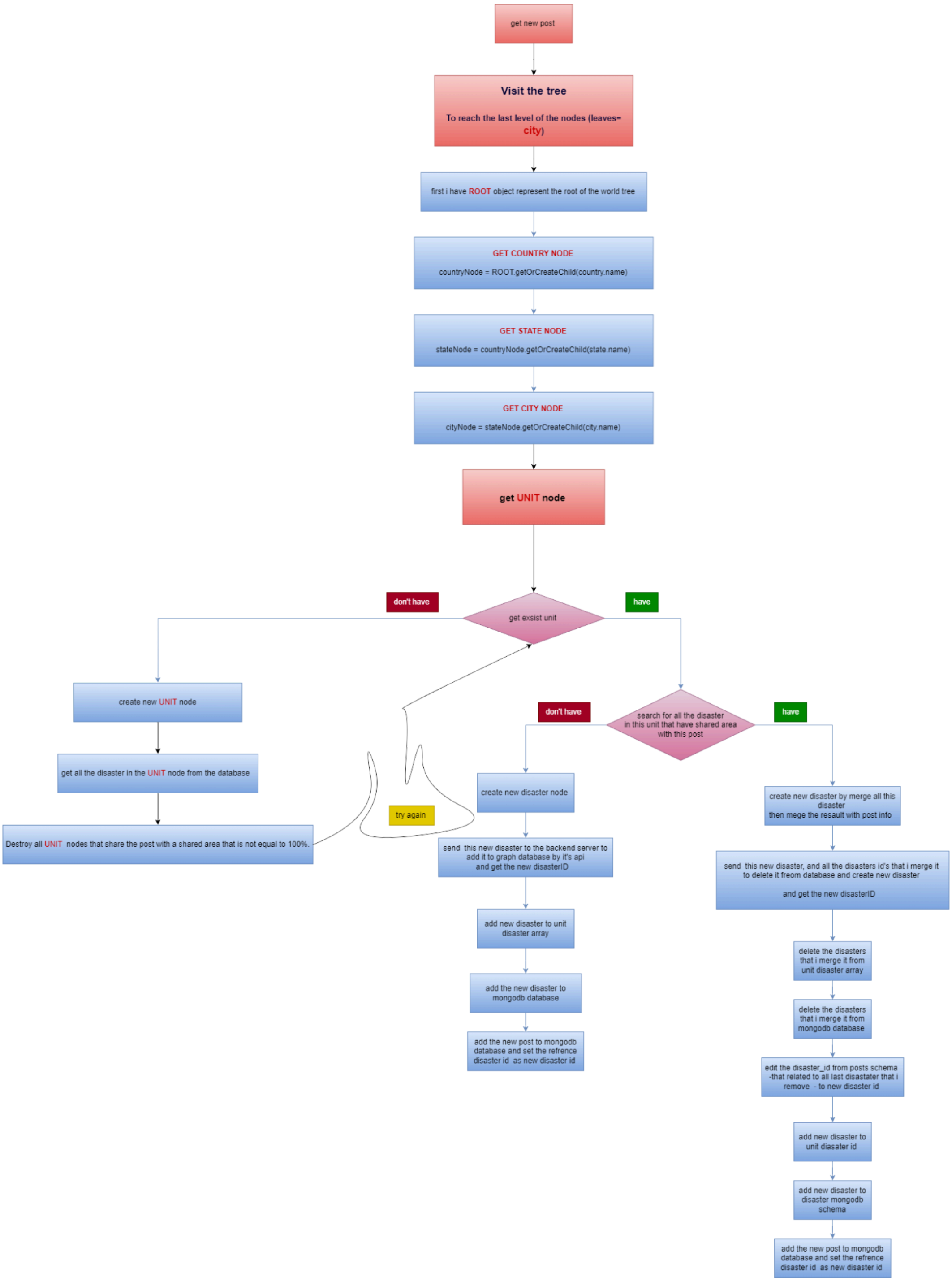


- To address this, data is partitioned into the smallest geographical units, and events within each unit are stored.
- Deletion operations are performed at the unit level, ensuring all data within the unit is either retained or deleted.
- In case of an event occurring within an existing unit, all necessary information can be accessed locally.

Deletion mechanism

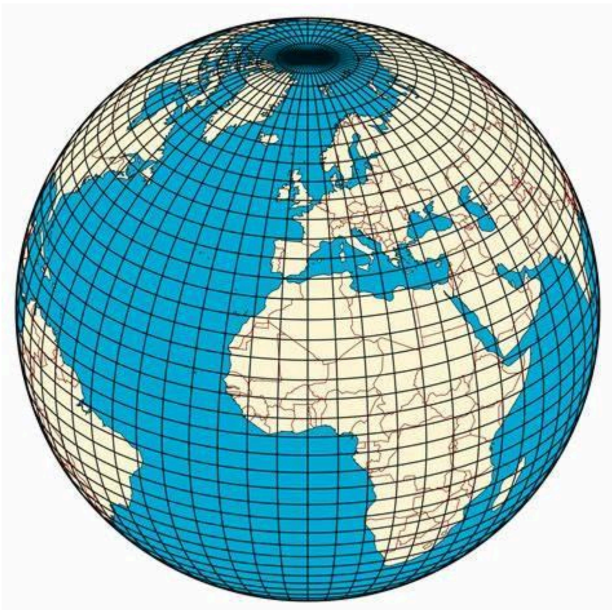


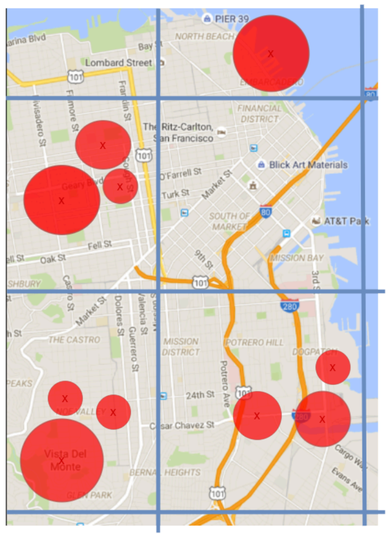
- The provided diagram illustrates the flow of data from the arrival of an alert to the creation of units and event merging.



Development of Search Algorithm :

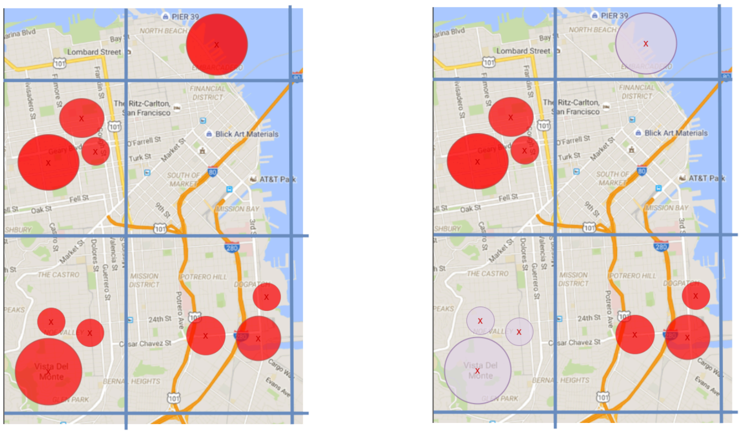
- In the search algorithm development, our main objective is to efficiently locate a specific range of values representing points near the current alert (the inactive event). If we can index these coordinates so that each value corresponds to a single unit in memory, it will significantly reduce complexity. Instead of dynamically creating units based on space and location, we propose creating fixed units. This involves dividing the Earth into units approximately one square kilometer in size, which are then indexed in memory. Rather than using lines of longitude and latitude in degrees, we measure coordinates in kilometers. This conversion is achieved by multiplying the coordinates by the constant factor representing the number of kilometers in each degree, which is 110. Thus, for any alert, obtaining the corresponding unit simply requires multiplication by this constant, providing coordinates in kilometers rather than degrees , and ignoring the decimal part of both the length and width coordinates, we can map the index to the appropriate unit. This optimization reduces the complexity of the tree search to O(1).





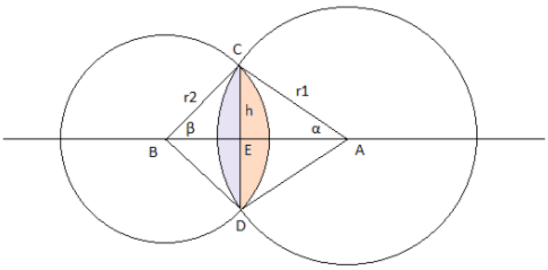
Memory Management:

Deletion mechanism In fixed slicing



Algorithm Explanation:

- The algorithm for merging intersecting events involves identifying the common area between any two intersecting events.
- The percentage of this area relative to the smallest event's area is calculated.
- If this ratio exceeds the minimum intersection threshold, the events are merged.
- Increasing the value of the minimum threshold enhances the accuracy of fire location detection.



Let $CE = h$ and note that $x = AB = AE + EB$, i.e.

$$x = \sqrt{r_2^2 - h^2} + \sqrt{r_1^2 - h^2}$$

Square to get $x^2 + r_2^2 - r_1^2 = 2x\sqrt{r_2^2 - h^2}$; square again to obtain

$$h = \frac{1}{2x}\sqrt{2x^2r_1^2 + 2x^2r_2^2 + 2r_1^2r_2^2 - x^4 - r_1^4 - r_2^4}$$

Then, the circle sector angles are, respectively

$$\alpha = \sin^{-1}\frac{h}{r_1}, \quad \beta = \sin^{-1}\frac{h}{r_2}$$

and the purple and orange areas are respectively the differences between the corresponding circle sectors and triangles, i.e.

$$S_a = \alpha r_1^2 - h\sqrt{r_1^2 - h^2}, \quad S_b = \beta r_2^2 - h\sqrt{r_2^2 - h^2}$$

Thus, the area enclosed by the two circles is

$$Area = S_a + S_b = r_1^2 \sin^{-1}\frac{h}{r_1} + r_2^2 \sin^{-1}\frac{h}{r_2} - xh$$

where h is given above.

Conclusion:

This documentation provides insights into the fire detection application's functionality, including alert reception, event analysis, data retrieval, and memory management. By implementing efficient algorithms and memory strategies, we ensure accurate and timely fire detection.