Scripts for Building and Querying Interval Trees with IP Geolocation Database

Posted on July 8, 2024

WhoisXML API now offers scripts for IP Geolocation Database users to easily create and use interval trees for efficient IPv4 and IPv6 geolocation data searches, including a function to check if an IP address falls within any GeoIP range.

For further information on our scripts or to submit a script idea that suits your requirements, please contact professional.services@whoisxmlapi.com.

Building the Tree

- **Purpose and Functionality:** These scripts are designed to create and utilize interval trees for efficient searching and querying of IP geolocation data. The build tree scripts target both IPv4 and IPv6 addresses, which will read data from CSV files containing IPv4 or IPv6 geolocation information, process each record, and convert IP address marks into intervals, and add them to the tree structure to build the interval tree.

- **Interval Tree Construction:** The scripts use the interval tree library to build the interval trees. For IPv4, the script iterates through the data, converting each IPv4 address into an interval between consecutive addresses. For IPv6, the script handles special cases by converting zero-value marks to the smallest possible IPv6 address, ensuring accurate tree construction and effectively handling the larger address space. This process ensures that IP address ranges are correctly segmented and added to the interval tree.

- **Serialization and Deserialization:** After building the interval tree, the scripts serialize it into

---

a pickle file using the pickle module. This allows the interval tree to be saved and reloaded efficiently, maintaining its state across different sessions and applications, which will avoid the need to rebuild the tree from scratch each time, saving time and computational resources.

Attached is an example output:

```
mengchen@vm-dev1:/mnt/nvme2/temp$ python3 build_geoipv4_interval_tree.py
Please enter the name of the geoip CSV file: geoipify_2024_06_13.csv
Please enter the name of the .pkl file to save the IPv4 interval tree: a_testing_ipv4_tree.pkl
Reading from geoipify_2024_06_13.csv and building IPv4 tree to a_testing_ipv4_tree.pkl
IPv4 interval tree construction took 862.44 seconds.
IPv4 interval tree serialized to file a_testing_ipv4_tree.pkl
Serialized IPv4 file size: 2000453527 bytes
```

Querying

- **IP Address Querying:** The scripts provide a function to check if a given IP address falls within any GeoIP ranges in the interval tree. For searching, it will convert the IP address input from string format to an integer and search the interval tree for matching intervals. If found, the corresponding geolocation data is returned; otherwise, default 'NA' values are provided.

- **Efficiency and Performance:** This script significantly improves the efficiency of IP information lookups compared to linear search methods, which makes them highly scalable and suitable for large datasets. The scripts also include time and size measurements for tree construction and serialization, providing insights into performance and storage requirements.

- **Loading and Serialization:** The ability to save and load interval trees as pickle files makes them easy to integrate into other applications and automated workflows. Developers can reload prebuilt interval trees, maintaining their state and ensuring consistent geolocation data access across different sessions. This makes these scripts highly convenient and practical for continuous use in various development environments.

Attached is an example output:

```
mengchen@vm-dev1:/mnt/nvme2/temp$ python3 check_ipgeo_intervaltree.py
Please enter the name of the .pkl file: a_testing_ipv4_tree.pkl
Loading Interval tree from a_testing_ipv4_tree.pkl
        This may take some time to load once...
Interval tree has been loaded from the file in 656.96 seconds.
Start IP Query loop
Please enter the IP address to search for: 24.43.110.82
Checking for IP address: 24.43.110.82
mark: 405499432
isp: Charter Communications (20001)
connectionType: broadband
country: US
region: Colorado
city: Cherry Creek
lat: 39.61100
lng: -104.86100
postalCode:
timezone: -06:00
geonameId: 8480048
Please enter the IP address to search for: 23.42.111.86
Checking for IP address: 23.42.111.86
mark: 388653056
isp: Akamai Technologies
connectionType:
country: TW
region: Taipei
city: New Taipei City
lat: 24.94702
lng: 121.58175
postalCode:
timezone: +08:00
geonameId: 1665148
```

--

These scripts offer developers a fast and reliable method to determine the geolocation of IP addresses. The interval tree structure allows for quick queries, enhancing the performance and efficiency of applications requiring real-time or frequent IP lookups.

Access the latest scripts on *GitHub*.