ScaleMax:image search

ahijaz,mhijaz,arman_mohammadi

1-**Introduction**:

ScaleMax:image search is a distributed system for searching for images that uses AI, and web crawling to answer users queries. Users specify a querying term and the system uses its pre-built index to return a list of image links that is tagged with the term. Building the tags using IMAGGA “image recognition API, the system is scalable and each time a node is booted it adds itself to the group of nodes doing work. Our systems consist of crawling, indexing, querying, and client nodes, and we use mapReduce for automatically parallelizing and executing on a large number of nodes. Our system scalability comes from the fact that we can boot a node at any point of time, and have it start receiving jobs from the map reduce framework.

2, Example:

At first we run our nodes and give them roles using the following command:

./distribution.js --ip '127.0.0.1' --port 8111 --crawler
./distribution.js --ip '127.0.0.1' --port 8110 --crawler
./distribution.js --ip '127.0.0.1' --port 8113 --indexer
./distribution.js --ip '127.0.0.1' --port 8114 --querier
./distribution.js --ip '127.0.0.1' --port 8115 --client

In this example, we have booted two crawling nodes. Then in either node command line we can start the crawling process with this command:
crawl $urls.txt”

,where urls.txt is a file that includes links to start the crawling process from. This command will automatically send image links to indexer, so we can run the following command on indexer cli to start the indexing process:
index

After that on a client node with port 8115 we can run the command
query $term
And it will output a list of images that were tagged with the term using AI

Example output:

> query food
[

'https://s.yimg.com/uu/api/res/1.2/GQq_iR6VE53GjIKAUNpB.Q--~B/Zmk9c3RyaW07aD0yODQ7cT04MDt3PTUzNjthcHBpZD15dGFjaHlvbg--/https://media.zenfs.com/en/reuters.com/51ab2ce67e9241e1a50cb6e053d83fb3.cf.jpg'
]
query cow
[

'https://s.yimg.com/uu/api/res/1.2/GQq_iR6VE53GjIKAUNpB.Q--~B/Zmk9c3RyaW07aD0yODQ7cT04MDt3PTUzNjthcHBpZD15dGFjaHlvbg--/https://media.zenfs.com/en/reuters.com/51ab2ce67e9241e1a50cb6e053d83fb3.cf.jpg'
]
query animal
[

'https://s.yimg.com/uu/api/res/1.2/GQq_iR6VE53GjIKAUNpB.Q--~B/Zmk9c3RyaW07aD0yODQ7cT04MDt3PTUzNjthcHBpZD15dGFjaHlvbg--/https://media.zenfs.com/en/reuters.com/51ab2ce67e9241e1a50cb6e053d83fb3.cf.jpg'
]

query cute
[

'https://s.yimg.com/uu/api/res/1.2/ZeNu1uBB8oU2eGpVLBMafw--~B/Zmk9c3RyaW07aD0yODQ7cT04MDt3PTUzNjthcHBpZD15dGFjaHlvbg--/https://s.yimg.com/os/creatr-uploaded-images/2024-05/4b251d50-07c5-11ef-bd6f-fde99765962f.cf.jpg',

'https://s.yimg.com/uu/api/res/1.2/GQq_iR6VE53GjIKAUNpB.Q--~B/Zmk9c3RyaW07aD0yODQ7cT04MDt3PTUzNjthcHBpZD15dGFjaHlvbg--/https://media.zenfs.com/en/reuters.com/51ab2ce67e9241e1a50cb6e053d83fb3.cf.jpg'
]

3, Discussion:

1-The system comes with some limitations. Especially the indexer because Imagga has rate limits. Imagga has a 1000 rate limit per day, and other rate limits of queries per ip, so we aren't able to index a lot of images. We found this fact interesting since we couldn't get around that even when we tried to add a timeout(sleep) before each request. Aside from that, the results we got from Imagga were enough to do some queries and prove that our concept works. This limitation can be addressed in the future by upgrading the Imagga from the free plan.

2-The optimal number of urls to send in one message was among the things we experimented with, sending each url in a message caused ECONNRESET, so we have implemented the SIZE constant as discussed above to solve the error. We have tried values 10, 100, and a 100 can cause heap out of memory sometimes, while 10 rarely causes an ECONNRESET. This can be addressed by increasing the memory of nodes on Amazon Aws.

3- Having a unique visited.txt per crawling node was an interesting choice. Another approach we thought of, was having a visited.txt file on the coordinator node and each time a node wants to check if a node is visited or add it to visited.txt to file, it can send a request to the coordinator node. However, that is a lot of messages exchanged in the network, and since each node will have unique urls at the beginning of each mapReduce jobs, we thought that crawling a url visited by another node will happen rarely and even if it did, it won't affect the system correctness.