The exercise is designed for you to show off your ability to design and execute a module e2e, understand your coding style, and test your coding skills.

Introduction

We're developing a network scanner that can scan a whole network. The scanner gets a (wide) range of IP addresses and can handle the load of concurrent scans.

The system's architecture is built as distributed nodes located around the world. Each node listens to a queue. Each message in the queue contains an IP address to scan. The node scans the IP address and sends the result to the results queue.

Specification

Your job is to implement a simplified scan node. For every IP, the scanner will:

1. check to see if there's a live host on that IP

2. for each port in ScanRequest.ports, test to see if the port is open.

3. for each open port, determine if the service' protocol on that port is HTTP or not.

4. push the result to the response queue.

Messages Structure:

The Request message structure is as follows:

```
from typing import Protocol, List, Optional

class ScanRequest(Protocol):
    id: int #same as in the request
    ipv4: str
    ports: List[int]
```

The Result should fit in the following schema:

```
class ScanResult(Protocol):
    id: int
    is_alive: bool
    ports: List[PortResult]
class PortResult(Protocol):
    port: int
    is_open: bool
    is_http: Optional[bool]
```

You can assume the queue implementation is given to you. Call the get(message: ScanRequest) function to receive the next message from the queue, and send(message: ScanResult) to push the scan result to the results queue. The rest of the implementation is up to you!

Some guidance:

1.Plan for maximum performance. assume your node can use all the machine's resources. Don't leave a CPU cycle as NOP ;).

2. Test your code! create a mock queue and a backlog of messages, send them to your node and fetch the result. Show that your node is working as planned. Bonus: time how many IPs/sec your implementations can do.

Have Fun!