Testing GRPC, Redis framework in all microservices

Setup

I will use grpcurl to simulate grp request to the various endpoints. Here is how to set it up:

1. Install: sudo snap install --edge grpcurl
2. Eg. list all endpoints at 9090: grpcurl -plaintext localhost:9090 list

Driver Service

Driver side redis cache testing:

1. Simply go to redis container and execute redis-cli command inside.

docker exec -ti redis redis-cli

2. Now use: KEYS * command to list out all keys in the cache. Should list out
    - drivers
    - Location-location-map
    - nearby-station

- If any key is not present, the redis cache isn't initialized properly. Check .py code which does this.
- Initially, drivers cache is empty map. Drivers are added by manually later.
    GET drivers -> returns empty {}
- Location-location-map is having all edges from each vertex to every other vertex. It's a fully connected map
    GET location-location-map -> returns long long list of mappings.
- Nearby-station is having city point -> nearby metro station mapping for k% of total city points.

Driver CRON job testing

The current CRON job is set to tick every 2 minutes in driverService business logic. The distance per tick is currently set to 10 units. If all is working well, driver-service logs and redis should reflect that. So, let's check that out.

Steps:

1. **Describe** what to send in GRP call to PostDriver service (adds a driver) using:

grpcurl -plaintext localhost:9090 describe com.metrocarpool.driver.PostDriver

2. A driver was added to the redis cache via this GRP curl command:

```
grpcurl -plaintext \
  -d '{ "driverId": 202, "routePlaces": ["b1","b2","b3","b4","b5"], "finalDestination": "b5",
"availableSeats": 3 }' \
  localhost:9090 com.metrocarpool.driver.DriverService/PostDriverInfo
```

3. Check what the output after running command in 2 is expected like using **describe**:

```
grpcurl -plaintext localhost:9090 describe com.metrocarpool.driver.DriverStatusResponse
```

4. Execute redis-cli in docker container of redis and set it on MONITOR:

```
docker exec -ti redis redis-cli
>MONITOR
```

OUTPUT:

```
127.0.0.1:6379> MONITOR
OK
1763280241.907299 [0 172.21.0.7:60794] "GET" "drivers"
1763280278.785557 [0 172.21.0.7:60794] "GET" "drivers"
1763280278.788413 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280278.849477 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
ace\":\"b2\",\"timeToNextPlace\":216.000000000,\"distanceToNextPlace\":18.0,\"finalDestination\":\"b5\",\"lastSeenMetroStation\":\"\"}}"
1763280362.353028 [0 172.21.0.7:60794] "GET" "drivers"
1763280362.356271 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280362.401121 [0 172.21.0.7:60794] "GET" "nearby-stations"
1763280362.407183 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
ace\":\"b2\",\"timeToNextPlace\":96.000000000,\"distanceToNextPlace\":8.0,\"finalDestination\":\"b5\",\"lastSeenMetroStation\":\"ME2\"}}"
1763280482.467544 [0 172.21.0.7:60794] "GET" "drivers"
1763280482.470216 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280482.497917 [0 172.21.0.7:60794] "GET" "nearby-stations"
1763280482.500311 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
ace\":\"b3\",\"timeToNextPlace\":132.000000000,\"distanceToNextPlace\":11.0,\"finalDestination\":\"b5\",\"lastSeenMetroStation\":\"ME2\"}}"
1763280602.417042 [0 172.21.0.7:60794] "GET" "drivers"
1763280602.421448 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280602.485904 [0 172.21.0.7:60794] "GET" "nearby-stations"
1763280602.489490 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
ace\":\"b3\",\"timeToNextPlace\":12.000000000,\"distanceToNextPlace\":1.0,\"finalDestination\":\"b5\",\"lastSeenMetroStation\":\"ME2\"}}"
1763280722.533773 [0 172.21.0.7:60794] "GET" "drivers"
1763280722.537701 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280722.575509 [0 172.21.0.7:60794] "GET" "nearby-stations"
1763280722.579010 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
```

```
Terminal  Local  Ubuntu  Ubuntu (2)
1763280602.489490 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
ace\":\"b3\",\"timeToNextPlace\":12.000000000,\"distanceToNextPlace\":1.0,\"finalDestination\":\"b5\",\"lastSeenMetroStation\":\"ME2\"}}"
1763280722.533773 [0 172.21.0.7:60794] "GET" "drivers"
1763280722.537701 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280722.575509 [0 172.21.0.7:60794] "GET" "nearby-stations"
1763280722.579010 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
ace\":\"b4\",\"timeToNextPlace\":60.000000000,\"distanceToNextPlace\":5.0,\"finalDestination\":\"b5\",\"lastSeenMetroStation\":\"ME1\"}}"
1763280842.496868 [0 172.21.0.7:60794] "GET" "drivers"
1763280842.499909 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280842.540403 [0 172.21.0.7:60794] "GET" "nearby-stations"
1763280842.542478 [0 172.21.0.7:60794] "SET" "drivers" "{\"202\":{\"availableSeats\":3,\"routePlaces\":[\"b1\",\"b2\",\"b3\",\"b4\",\"b5\"],\"nextPl
ace\":\"b5\",\"timeToNextPlace\":96.000000000,\"distanceToNextPlace\":8.0,\"finalDestination\":\"b5\",\"lastSeenMetroStation\":\"ME1\"}}"
1763280962.630532 [0 172.21.0.7:60794] "GET" "drivers"
1763280962.633279 [0 172.21.0.7:60794] "GET" "location-location-map"
1763280962.661663 [0 172.21.0.7:60794] "GET" "nearby-stations"
1763280962.666577 [0 172.21.0.7:60794] "SET" "drivers" "{}"
1763281082.642591 [0 172.21.0.7:60794] "GET" "drivers"
1763281202.751651 [0 172.21.0.7:60794] "GET" "drivers"
1763281322.073636 [0 172.21.0.7:60794] "GET" "drivers"
```

Meaning:

What each step means
- Initial SET: route b1→b5; nextPlace b2; distance 18; time 216s (18×12s/unit); lastSeen empty.
- Tick 1: distance to b2 drops 18→8; time 96s; lastSeen becomes ME2 (b1 has ME2 in CSV).
- Tick 2: reaches b2 (uses remaining 8) and moves 2 units into b2→b3 (13 total) → remaining 11; time 132s; nextPlace b3.
- Tick 3: distance to b3 11→1; time 12s.
- Tick 4: reaches b3 (consumes 1), enters b3→b4 (14 total) → remaining 5; time 60s; lastSeen flips to ME1 (b3/b4 have ME1).
- Tick 5: reaches b4 (consumes 5), enters b4→b5 (13 total) → remaining 8; time 96s; nextPlace b5.
- Tick 6: arrives at finalDestination b5 → evicted → drivers "{}".

Implied segment lengths (consistent with logs)

- b1→b2: 18
- b2→b3: 13
- b3→b4: 14
- b4→b5: 13

Sanity checks that pass
- timeToNextPlace = distanceToNextPlace × 12s (with 10 units per 120s tick).
- lastSeenMetroStation updates at nodes with entries in CSV (ME2 at b1, ME1 at b3/b4).
- State advances across segments and evicts at destination.