Class Project

You will be buliding a distributed key-value datastore using ZeroMQ as transport protocol.

[Node-0] [Node-1] [Node-2] [Node-3]

Cluster Adjustment

- Add Node-4
- Remove Node-0

How to launch server cluster

Format: python server_consumer.py {num_node}

```
pipenv run python server_consumer.py 4
```

How to run client

Format: python client_producer.py {num_node}

```
pipenv run python client_producer.py 4
```

Phase 1

The scope of phase 1 is to shard (PUT) the data into a list of servers. No retrieval is required. You will be implementing the following two hashing algorithms on the client side.

- Consistent hashing
- HRW hashing

Phase 2

In the phase 2, you will be adding retrieval GET by Id/key and GET all operations in both client and server sides. In addtion, the PUT method will get modified to work with new interface.

PUT

JSON Request Payload

```
{
    "op": "PUT",
    "key": "key",
    "value": "value"
}
```

GET by key

JSON Request Payload

```
{
    "op": "GET_ONE",
    "key": "key"
}
```

JSON Response Payload

```
{
    "key": "key",
    "value": "value"
}
```

GET All

JSON Request Payload

```
{
    "op": "GET_ALL",
}
```

JSON Response Payload

```
{
    "collection": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        }
    ]
}
```

Cluster Membership

In order to dynamically control the node membership, your system will integrate with Consul.

In the phase 1, we collected cluster size from the command line and mapped to nodes using this scheme of

```
server_port = "200{}".format(each_server)
```

In the phase 2, both client and server will no longer read the initial cluster size from the command line. Instead, you will be loading from Consul.

First, each node will be registered to the membership in Consul during the server boot up. Upon the server shut down, the node will be removed from the membership.

Similarly on the client side, you will first lookup the membership from Consul and then the data will be sharded across different nodes.

Cluster Adjustment

Adding and removing nodes will be supported in the consistent hashing mode only and node rebalancing--moving data from one node to another--will be handled on the client side by sending the *remove* and *add* signals to Consul.

Steps to remove node

- Pick a node to be removed.
- Re-balance data to the other nodes.

- After removal is done, send *remove* signal to Consul.

Steps to add node

- Send *add* signal to Consul.
- Server will get a push notification about the membership changes.
- Launch a new server process based on the node information given by Consul.
- Re-balance existing data to the new node.