AntidoteClient_Wrapper

Antidote Features

- Supports high-level replicated data types (counters, sets, maps)
- Multiple servers in geo-distributed locations
- Fully-replicated data centers in the network core
- Relation between updates to different objects

Antidote Working

- Data is shared among the servers using consistent hashing and organized in a ring
- Read/Write Transaction
    - Server hosting a copy of the data
    - Multiple objects
        - Contact servers that have access by the transaction
    - Master-less design
- Replication of updates
    - Cure protocol
    - Asynchronous replication
- API is there for connecting to an Antidote instance and accessing data
- Connection is initiated to the Data center using its **Address** and **Port**
- Object can be crated and updates by the client using **Counter**
- BObj identifies the object
- Object is identified by its **Key** and **Bucket**

AntidoteClient_Wrapper

(Assumptions on the initial phase, can be changed as the project moves forward)

- Acts as a bridge between Docker and YCSB
- Input
    - From: **Docker module**
    - The Address and port
- Output
    - To: **YCSB module**
    - **Read()** - Read() of YCSB
    - **Write()** - Insert() of YCSB
    - **Update()** - Update() of YCSB
    - The table in which these operations are performed
    - Value and Data Type
    - Key