一、题目

1.实现一个基于一致性哈希分区算法的分布式键值 (Key-Value) 数据存储系统系统要提供一个客户端SDK，该SDK支持两类操作：

（1）put(key, value)：写入一个键值对，key是定长字符串；value是长度受限的变长字符串

（2）get(key)：返回指定key对应的value。如果不存在，则返回null

2.在客户端SDK中：以key为输入，利用一致性哈希算法计算出负责存储该key的节点标识，然后将put或get操作发送给该节点。

3.不同节点可以在单机上用不同的进程 (在不同的端口号监听)或虚拟机模拟

4.将服务器节点标识列表作为客户端SDK的初始化参数。

5.键值数据保存在内存中就可以。

6.支持增删节点。

二、设计说明

1.需要进行节点和数据的存储，需要进行两个接口的设计，接口1:实现节点的增删改查，设计以下功能函数：

1. **void** addNode(String node, **int** port) ;
2. **int** removeNode(String node) ;
3. String getNode(String key) ;
4. **int** getHash(String key);
5. List<String> outputAllNodes();

接口2:实现节点内数据的增删改查，设计以下功能函数：

1. **void** put(String key, String value);
2. String get(String key) ;
3. **void** remove(String key);
4. List<String> outputall();

2.需要通过客户端输入的信息进行节点的注册以及不同端口获取引用方法，客户端需要通过接口3连接注册中心，进行注册

接口3功能函数：

1. **void** RegisterNewNode(String ip, **int** port)；

3.节点、数据信息存储具有以下特点：客户端停止进程后数据不丢失、不重制，需要将**数据信息**存储到接口2所连接的文件内，定义：

1. **private** Map<String, String> data;//键值存储

**节点信息**以及**节点引用方法**存储到接口1所连接的文件内，定义：

1. List<No> node\_list = **new** ArrayList<No>();//节点使用方法列表
2. **private** TreeMap<Integer, String> ring;//一致性哈希环

只有当服务器重新启动时，数据初始化。

4.节点、数据存储流程：

**节点**：

客户端client接收用户存储请求，接收节点A的ip 🡪 通过接口3连接注册中心，注册节点A 🡪 通过接口1连接节点存储中心，调用addNodes函数，addNodes函数调用getHash函数获得哈希值，放入ring中存储； 通过lookup方法获得节点调用方法，放入node\_list中存储 🡪 获取节点A的前一个节点Q，将Q存储数据分担给节点A

**数据：**

客户端client接收用户存储请求，接收数据B的key, value 🡪 通过接口1调用put函数，put函数调用getNode函数获得数据B应该存放在节点C中，并从node\_list中获取节点C引用方法 🡪 通过接口2，调用put函数，将数据存放到节点C的map中存储

三、代码实现

1、服务器远程接口

（1）接口1:Store

|  |
| --- |
|  |

（2）接口2:Node

|  |
| --- |
|  |

（3）接口3:Re

|  |
| --- |
|  |

2、注册中心

先注册初始4个节点，再注册一个哈希环，通过Store store = new ConsistentHashingKVStore(nodes)中的nodes将初始4节点绑定到哈希环内

|  |
| --- |
|  |

ConsistentHashingKVStore接收到初始节点信息后，将节点放入哈希环ring内，并把节点引用方法也存起来

|  |
| --- |
|  |

注册中心内，接口Re可调用注册中心注册新节点方法

|  |
| --- |
|  |

3、函数方法

**增加节点：根据节点的哈希值，获得前一个节点的数据，存入Map<String, String> alldata中，增加新节点，循环先去调用remove除掉节点的alldata数据，再调用put增加alldata，可以达到分担前一个节点数据的作用**

|  |
| --- |
|  |

增加节点前，在client端调用Re接口注册新节点

|  |
| --- |
|  |

**删除节点：获取删除节点数据存入Map<String, String> alldata中，重新将数据put入其他节点**

|  |
| --- |
|  |

**输出所有节点：**

|  |
| --- |
|  |

**增加数据、删除数据**函数

Put：需要先通过接口Store的put方法获得节点，再通过接口No的put方法存放信息

|  |
| --- |
|  |

Get和Remove同理

在No接口中，对存储的数据进行操作

|  |
| --- |
|  |

**求哈希值：直接调用java中的.hash得到数据不分散，于是利用MD5求哈希值，将哈希值范围压缩到0-1000**

|  |
| --- |
|  |

四、实验结果

启动注册中心：

|  |
| --- |
|  |

**增删节点操作：**

**删除节点rmi://127.0.0.2之后，其数据data1转移到rmi://127.0.0.4中，增加节点rmi://127.0.0.2之后，数据data1又转移回来。**

|  |
| --- |
|  |
|  |