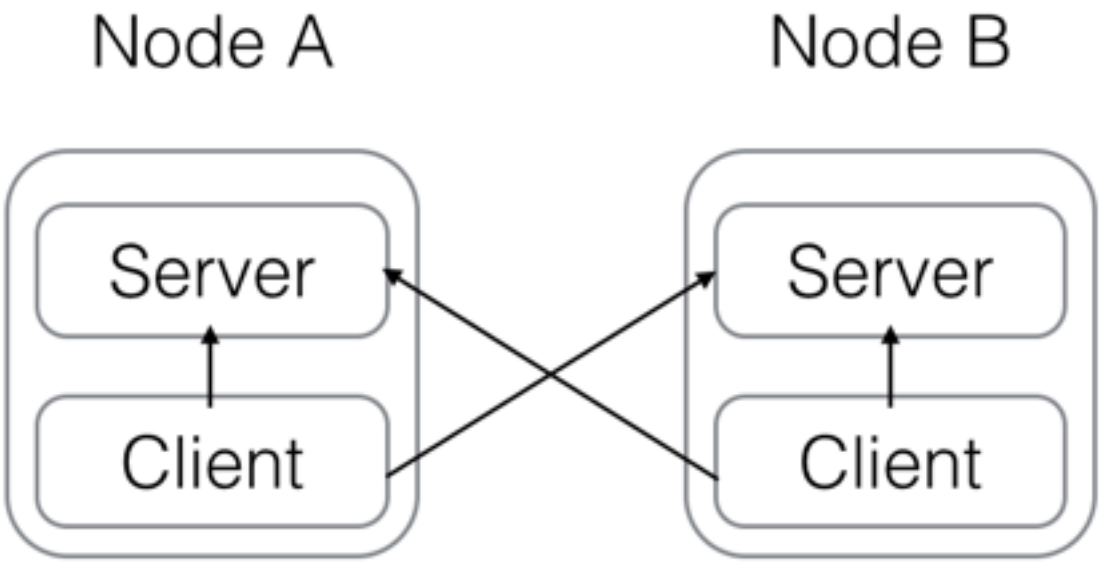
Design Documents

for A Simple Distributed Hash Table

Man Liu - October 18, 2015



Mechanism for Node Server

The mechanism for the index server is simple: the server reads the configure file and gets the port number, then turns on the port and when a client connects, it starts a new thread to handle requests. For now, the server only processes three commands: put, get and del.

Class NodeServer is for reading the configure, listening the port and starting new threads.

Class NodeServerThread is for receiving all requests from clients and returning results of requests.

Class Config is for manipulating configure.

Mechanism for Node Client

A client communicates with all eight servers. It reads configure 'servers.properties' to get server addresses and ports and connects to all servers. Then, a client calculates a hash code to decide which server stores a given key, and sends messages to the server and the backup server. So the resilience for the key/value pairs is decided by client. For now, the backup server is the next server on the list, which can be implemented as deciding by config in the future.

Class Config is for manipulating configure.

Class Commander is for running a client, which is to receive user inputs, to manage messages, and to communicate with servers.