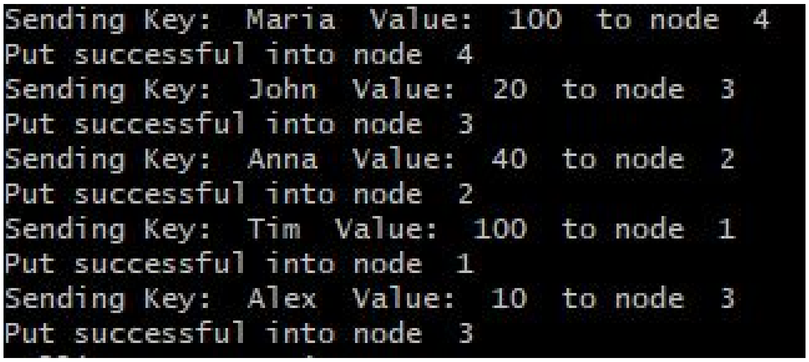
Ryan Myers, Mihir Deshmukh
CPE 469
Assignment 5: Dynamo
Dr. Maria Pantoja

Questions

1. What is the location (server name) of each one of the above keys?

   The keys are stored in the following locations:
   Maria -> Server 4
   John -> Server 3
   Anna -> Server 2
   Tim -> Server 1
   Alex -> Server 3

   This can be seen by the following image of consecutive put operations using our Dynamo implementation



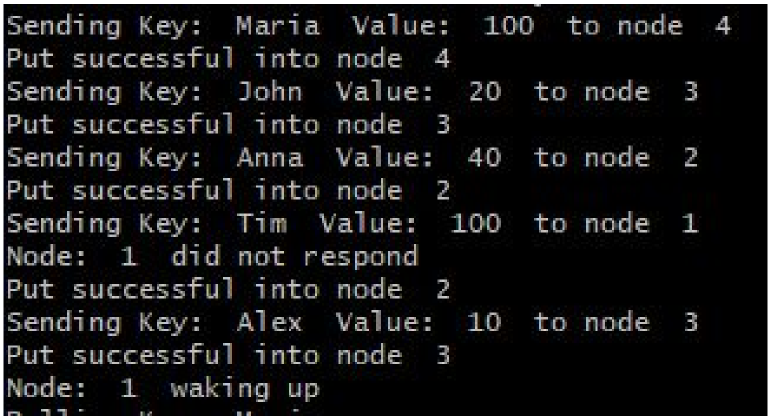
```
Sending Key:  Maria  Value:  100  to node  4
Put successful into node  4
Sending Key:  John  Value:  20  to node  3
Put successful into node  3
Sending Key:  Anna  Value:  40  to node  2
Put successful into node  2
Sending Key:  Tim  Value:  100  to node  1
Put successful into node  1
Sending Key:  Alex  Value:  10  to node  3
Put successful into node  3
```

Figure 1: Consecutive put operations on all key values with no node failures

2. What is the location (server name) of each one of the above keys if one of the servers (the second one) goes done?

   After server 1 (the second one) goes down, the keys are stored in the following locations:
   Maria -> Server 4
   John -> Server 3
   Anna -> Server 2
   Tim -> Server 2
   Alex -> Server 3

This can be seen in the following image after server 1 goes down and consecutive put operations are performed on all the keys, after which server 1 wakes up



```
Sending Key:  Maria  Value:  100  to node  4
Put successful into node  4
Sending Key:  John  Value:  20  to node  3
Put successful into node  3
Sending Key:  Anna  Value:  40  to node  2
Put successful into node  2
Sending Key:  Tim  Value:  100  to node  1
Node:  1  did not respond
Put successful into node  2
Sending Key:  Alex  Value:  10  to node  3
Put successful into node  3
Node:  1  waking up
```

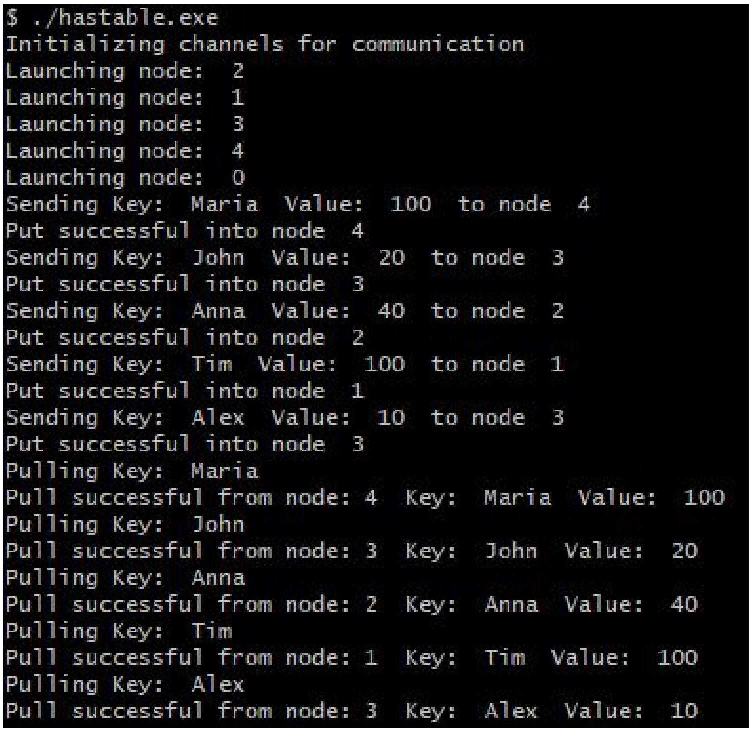
Figure 2: Consecutive put operations on all keys following the failure of node 1

Running Code

Our code can be compiled using the command:
go build hashtable.go

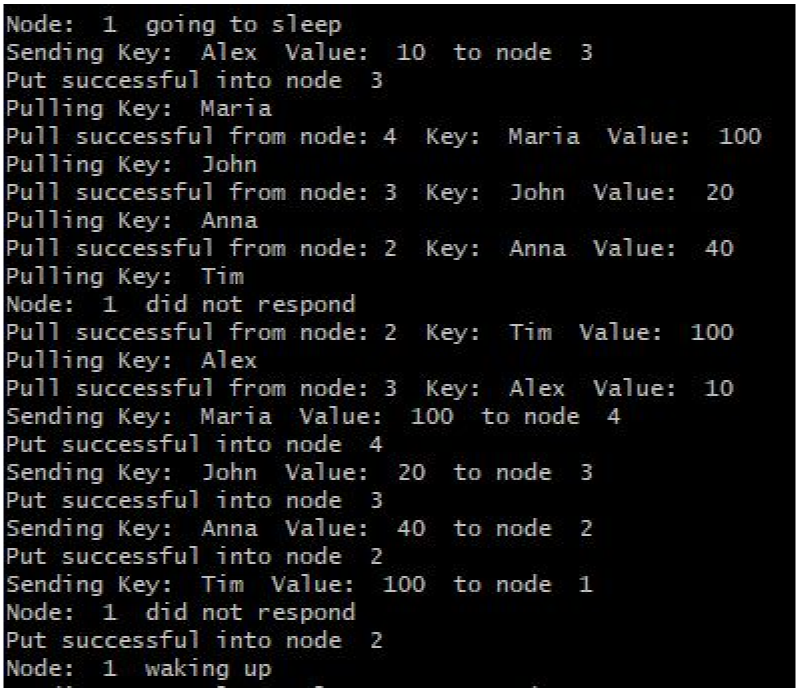
The code can then be run by using the command:
./hashtable.exe

The following images demonstrate our Dynamo protocol running. The program output shows when the user is trying to put or pull a key from the server network and where the value is eventually inserted or received from. It also shows when a node goes to sleep and consequently if a put or pull failed from said node.



```
$ ./hastable.exe
Initializing channels for communication
Launching node:  2
Launching node:  1
Launching node:  3
Launching node:  4
Launching node:  0
Sending Key:  Maria  Value:  100  to node  4
Put successful into node  4
Sending Key:  John  Value:  20  to node  3
Put successful into node  3
Sending Key:  Anna  Value:  40  to node  2
Put successful into node  2
Sending Key:  Tim  Value:  100  to node  1
Put successful into node  1
Sending Key:  Alex  Value:  10  to node  3
Put successful into node  3
Pulling Key:  Maria
Pull successful from node: 4  Key:  Maria  Value:  100
Pulling Key:  John
Pull successful from node: 3  Key:  John  Value:  20
Pulling Key:  Anna
Pull successful from node: 2  Key:  Anna  Value:  40
Pulling Key:  Tim
Pull successful from node: 1  Key:  Tim  Value:  100
Pulling Key:  Alex
Pull successful from node: 3  Key:  Alex  Value:  10
```

Figure 3: Start of Dynamo protocol, all nodes are alive and can be seen fulfilling successful put and pull requests



```
Node:  1  going to sleep
Sending Key:  Alex  Value:  10  to node  3
Put successful into node  3
Pulling Key:  Maria
Pull successful from node: 4  Key:  Maria  Value:  100
Pulling Key:  John
Pull successful from node: 3  Key:  John  Value:  20
Pulling Key:  Anna
Pull successful from node: 2  Key:  Anna  Value:  40
Pulling Key:  Tim
Node:  1  did not respond
Pull successful from node: 2  Key:  Tim  Value:  100
Pulling Key:  Alex
Pull successful from node: 3  Key:  Alex  Value:  10
Sending Key:  Maria  Value:  100  to node  4
Put successful into node  4
Sending Key:  John  Value:  20  to node  3
Put successful into node  3
Sending Key:  Anna  Value:  40  to node  2
Put successful into node  2
Sending Key:  Tim  Value:  100  to node  1
Node:  1  did not respond
Put successful into node  2
Node:  1  waking up
```

Figure 4: Example of node 1 going to sleep and subsequent put and pull operations being forwarded to node 2 after no response from node 1