```go
package main

import (
	"flag"
	"fmt"
	"log"
	"net"
	"net/rpc"
	"strconv"
	"strings"
	"sync"
	"time"

	"github.com/marcelloh/fastdb"
)

type Node struct {
	id       int
	peerIds  []int
	leaderId int

	electionMutex     sync.Mutex
	isElectionRunning bool

	internalListener net.Listener
	internalServer   *rpc.Server

	leaderListener net.Listener
	leaderServer   *rpc.Server

	database *Database
}

type Database struct {
	db    *fastdb.DB
	mutex sync.Mutex
}

type InternalRPC struct {
	node *Node
}

type LeaderRPC struct {
	node *Node
}

func main() {
	nodeID := (flag.Int("id", 1, "Node ID"))
	peersStr := flag.String("peers", "", "Comma-separated list of peer IDs")
	flag.Parse()

	if *peersStr == "" {
```

```
53                  log.Fatal("Peers must be specified via -peers")
54          }
55          var peerIDs []int
56          for _, p := range strings.Split(*peersStr, ",") {
57                  pid, err := strconv.Atoi(strings.TrimSpace(p))
58                  if err ≠ nil {
59                          log.Fatalf("Invalid peer ID: %s", p)
60                  }
61                  if pid ≠ *nodeID {
62                          peerIDs = append(peerIDs, pid)
63                  }
64          }
65
66          store, err := fastdb.Open(":memory:", 100)
67          if err ≠ nil {
68                  message := fmt.Sprintf("[Node %d] Failed to open db", nodeID)
69                  log.Fatal(message)
70          }
71          sharedDataBase := Database{db: store}
72          node := &Node{
73                  id:       *nodeID,
74                  peerIds:  peerIDs,
75                  leaderId: -1,
76
77                  isElectionRunning: false,
78
79                  database: &sharedDataBase,
80          }
81
82          node.startInternalServer(&sharedDataBase)
83
84          time.Sleep(5 * time.Second)
85
86          node.startHeartbeatRoutine()
87
88          select {}
89  }
90
```