海联金汇 HYUNION HOLDINGS
股票代码：002537

UMF 联动优势 UNION MOBILE FINANCIAL TECHNOLOGY
海联金汇旗下企业

科技赋能普惠金融

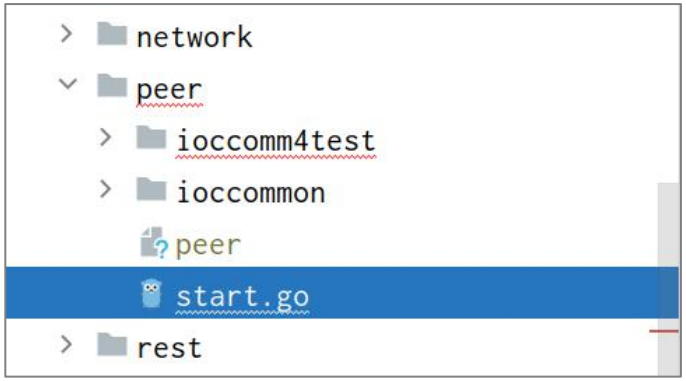
熟悉优链的代码流程

王宇

MAKE FINTECH EVERYWHERE

目录

入口



优链的程序入口：peer/start.go

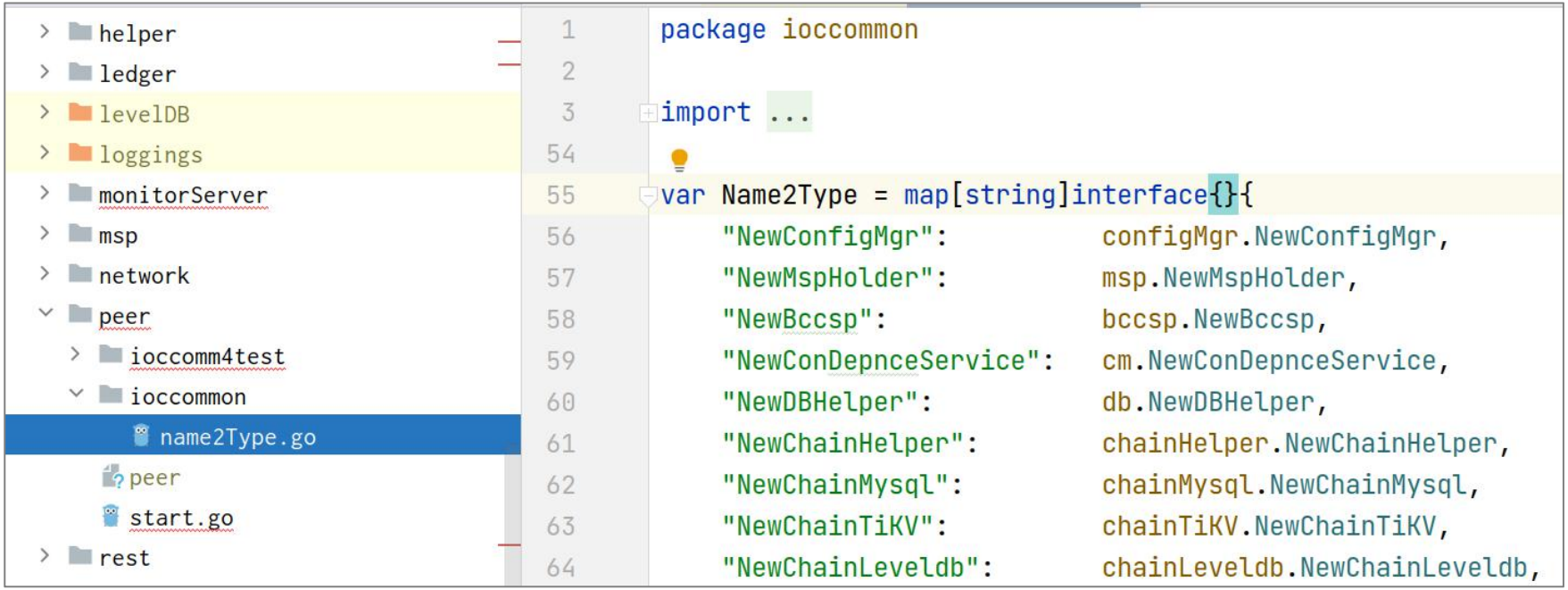
入口做了哪些事情：

1. 通过IOC启动所有程序实例
2. 启动事件服务器
3. 启动监控上报服务

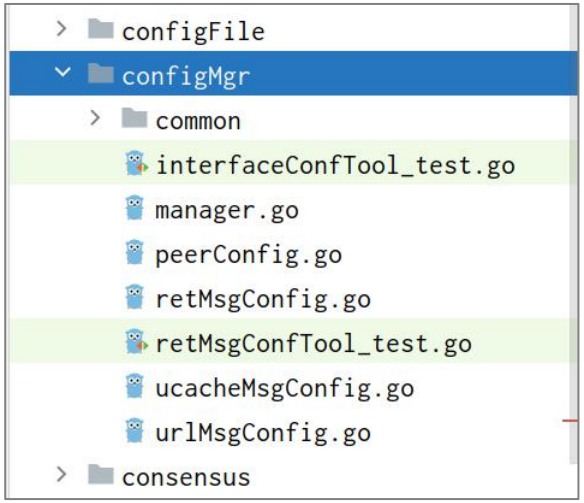
IOC

```
startLogger.Infof( format: "R-[main](%s)",  args...: "start...")
// 初始化协调器,里面根据模块依赖注入相应实例
b, err := uioc.NewBuilder(ioccommon.Name2Type)
if err != nil {...}

//config 里的日志和flogging里的日志 ?
var iocLogger = ulogging.MustGetLogger(ulogging.Uioc)
b.ConfigLogger(iocLogger)

iocCtx := b.Build()
err = iocCtx.CreateAllInstance()
if err != nil {...}

config := configMgr.Configration.GetPeerConfigViper() // 获取文

/.../

err = iocCtx.Start()
if err != nil {...}
```

IOC注入的模块


helper
ledger
levelDB
loggings
monitorServer
msp
network
peer
ioccomm4test
ioccommon
name2Type.go
peer
start.go
rest
1 package ioccommon
2
3 import ...
54
55 var Name2Type = map[string]interface{}{
56 "NewConfigMgr": configMgr.NewConfigMgr,
57 "NewMspHolder": msp.NewMspHolder,
58 "NewBccsp": bccsp.NewBccsp,
59 "NewConDepnceService": cm.NewConDepnceService,
60 "NewDBHelper": db.NewDBHelper,
61 "NewChainHelper": chainHelper.NewChainHelper,
62 "NewChainMysql": chainMysql.NewChainMysql,
63 "NewChainTiKV": chainTiKV.NewChainTiKV,
64 "NewChainLeveldb": chainLeveldb.NewChainLeveldb,

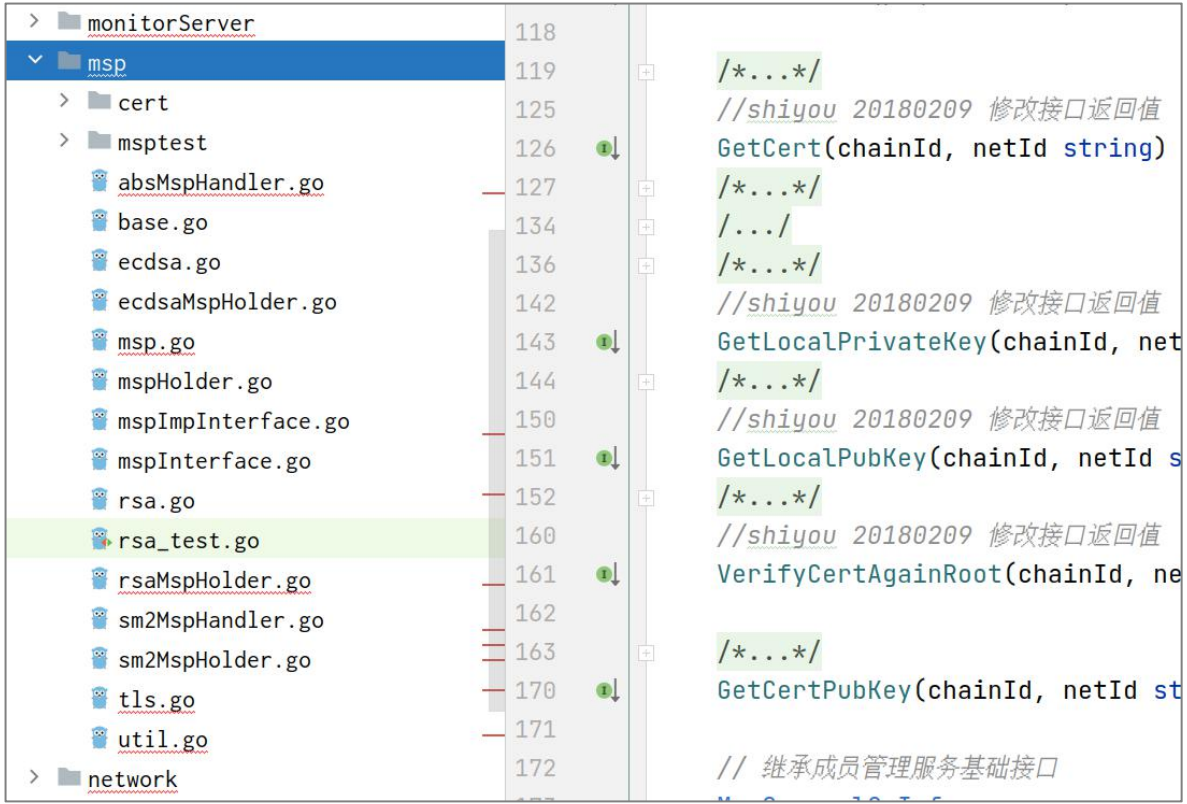
获取配置




```
import (
    "strings"
    "sync"

    "github.com/spf13/viper"
)
```

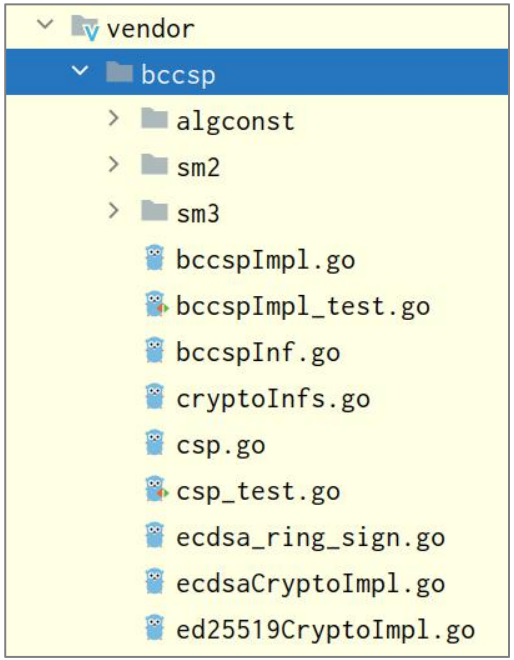
```
func newPeerConfig() (*PeerConfig, error) {
    config := viper.New()
    config.AddConfigPath( in: "./configFile/")
    config.AddConfigPath( in: "../configFile/")
    config.AddConfigPath( in: "../../configFile/")
    config.SetConfigName( in: "peer") // name of con
    config.AutomaticEnv()
    replacer := strings.NewReplacer( oldnew...: ".",
    config.SetEnvKeyReplacer(replacer)
    err := config.ReadInConfig()
    if err != nil : nil, err
```

成员管理




```
import (
    "crypto/tls"
    "crypto/x509"
    "google.golang.org/grpc"
    "google.golang.org/grpc/credentials"
    "io/ioutil"
    "uchains/api/ulogging/error"
)
```

加密服务

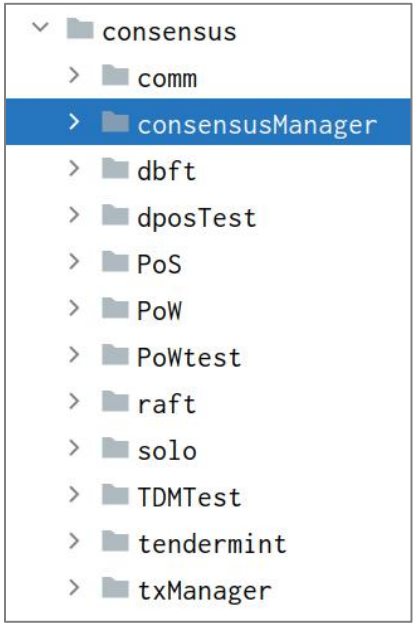


bccspInf.go

```
/*
Copyright IBM Corp. 2016 All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0
```

```
/**
*NBSP@Description:NBSP将sm2 签名后得到的r,s放到跟密钥长度有关的数组里返回
* 字节数组的长度的计算公式 ((椭圆曲线密钥长度 + 7) / 8)*2
* @param k sm2 私钥
* @param digest 被签名消息的hash值
* @param opts 签名设置，暂未用到
* @return signature 签名
* @return err 错误 if any
*NBSP@authorNBSPranshiyou
*NBSP@dateNBSP2018/2/7NBSP13:58
*/

func signSM2ByCopyRS(k *sm2.PrivateKey, digest []byte, opts SignerOpts)
    r, s, err := sm2.Sign(k, digest)
```

共识依赖的结构




```go
type ConDepnceService struct {
    ledger.LedgerServiceInf
    syncService.SyncServiceInf
    chainInterface.ChainServiceInf
    txManager.TxManagerService
    peer.PeerHandlerInf
    appInterface.AppServerInf
    msp.MspOpInf
    netMgrInterface.NetMgrService
}
```

```go
/**
 *   @PackageName:   consensusManager
 *   @Description:   共识管理模块，管理每个链所对应的共识实现
 *   @author   ChenZheng
 *   @date   2018年4月16日
 */
type ConsensusManager struct {
    sync.RWMutex
    ConsensusMap     map[string]constants.IConsensus
    ConsensusTypeMap map[string]consensusType.ConsensusType
    assetInf.IAssetMgr
    assetInf.IVoteMgr
}
```

共识启动

```
type ConsensusManagerInf interface {
    StartConsensus(chainInfo *chainComm
    StopConsensus(chainId string) error
    GetConsensusTypeByChainId(chainId s
    GetChainConsensusByChainId(chainId
}
```

```
func (c *ConsensusManager) StartConsensus(chain
    chainid := chainInfo.ChainID
    consType := consensusType.ConsensusType(cha
    logger.Infof( format: " ** ChainId=[%s], Con
    time.Sleep(time.Second * 10)
    service := uioc.GetIoCContext().Get( name: "
    c.setConsensusTypeToMap(chainid, consType)
    switch consType {
    case consensusType.PoW:...
    case consensusType.Tendermint:
        x, _ := tm.NewTendermintCore()
        x.NewConsensusCore(service, chainInfo)
        c.setChainConsensusToMap(chainid, x)
    case consensusType.PoS:...
    case consensusType.DBFT:...
    case consensusType.Solo:...
    default:...
    }
```

数据库句柄

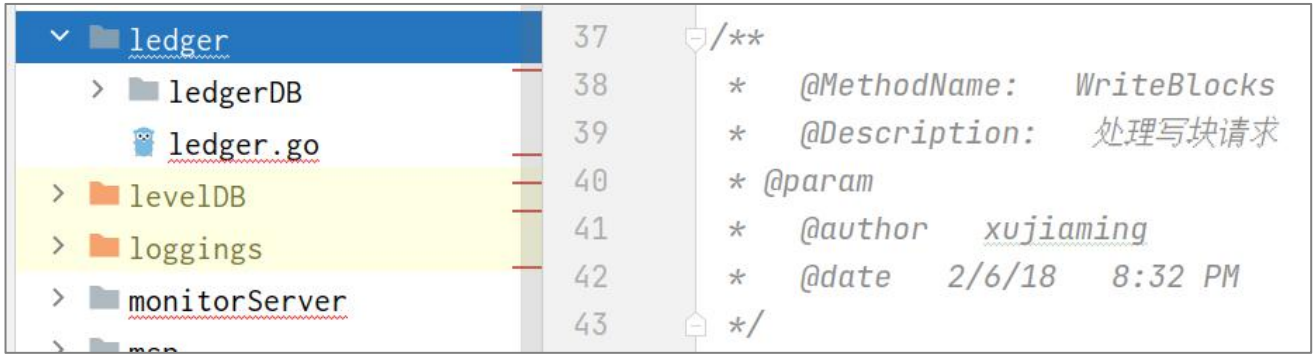

```go
type DBHandler struct {
    dbHandler map[string]interface{}
    dbLock    sync.RWMutex
}
```

```go
func (dh *DBHandler) Start() {
    config := configMgr.Configration.GetPeerConfigViper()

    //Connect
    types := config.GetString( key: "db.supportDBType")
    if types == "" : "dbtypes is nil" *

    typeArray := strings.Split(types, sep: ",")
    dbHelperLogger.Infof( format: "typeArr=%v", typeArray)
    dbHelperLogger.Debugf( format: "[dbHelper] dbType=%v",
    for _, typ := range typeArray {
        switch typ {
        case string(dbComm.MYSQLXORM):...
        case string(dbComm.ROCKSDB):

        case string(dbComm.TIKV):...
        case string(dbComm.LEVELDB):
            /*...*/
        case string(dbComm.MONGODB):...
        default:: fmt.Sprintf("invalid db type %s from cor
        }
```

链助手


```
/**
 *   @MethodName:      GetContractName
 *   @Description:    从内存中获取链对应的合约名
 * @param: chainID 链ID
 *   @author   wangjie
 *   @date   2019/8/29   11:38
 */
```

账本



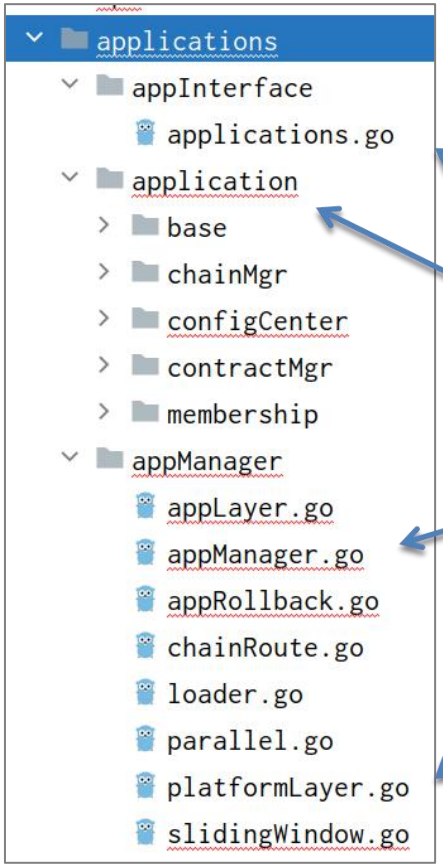

```
"NewChainMongodb":      chainMongodb.NewChainMongodb,
"NewLegder":            ledger.NewLegder,
"NewLedgerTikv":        ledgerTikv.NewLedgerTikv,
"NewLedgerMysql":       ledgerMysql.NewLedgerMysql,
"NewLedgerLeveldb":     ledgerLeveldb.NewLedgerLeveldb,
"NewLedgerMongodb":     ledgerMongo.NewLedgerMongodb,
"NewIndexHelper":       indexHelper.NewIndexHelper,
```

```
/**
 *   @MethodName:   RollbackBlocks
 *   @Description:  处理回滚请求
 * @param
 *   @author   xujiaming
 *   @date   2/6/18   8:33 PM
 */
```

索引助手


contractHelper
indexHelper
helperDB
indexHelper.go
membershipHelper
networkMgr
ledger
69 "NewLedgerLeveldb": ledgerLeveldb.NewLedgerLeveldb,
70 "NewLedgerMongodb": ledgerMongo.NewLedgerMongodb,
71 "NewIndexHelper": indexHelper.NewIndexHelper,
72 "NewIndexMysql": indexMysql.NewIndexMysql,
73 "NewIndexTiKV": indexTiKV.NewIndexTiKV,
74 "NewIndexLeveldb": indexLeveldb.NewIndexLeveldb,
75 "NewIndexMongodb": indexMongo.NewIndexMongodb,
76 "NewAppManager": appManager.NewAppManager,

APP管理器



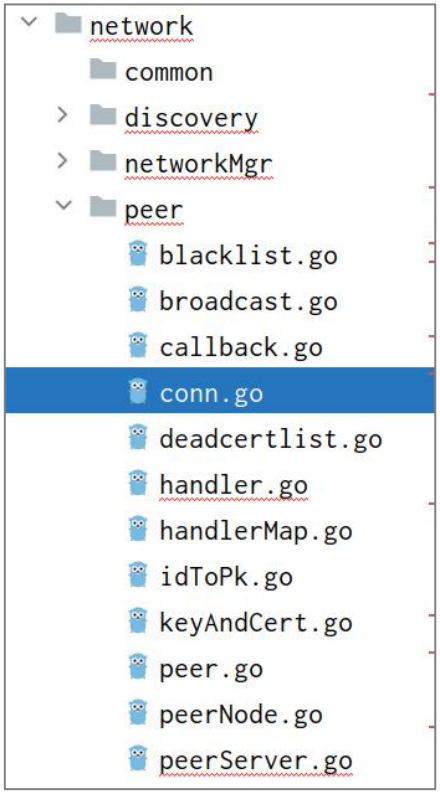
1. appInterface 接口定义了注册链、升级合约等接口

2. application 目录下的是系统链，链管理链、合约管理链、成员管理链等，包括新增的配置中心链

3. appManager.go 中初始化系统链

4. platformLayer.go 接收合约模块的请求，处理写块

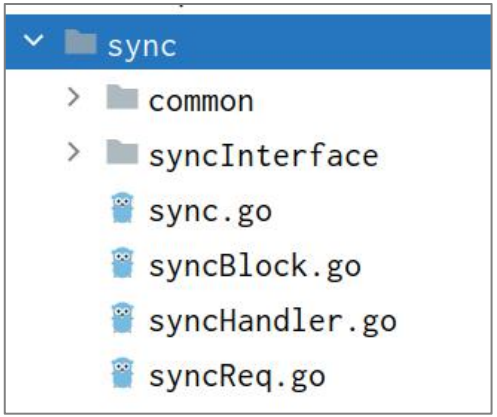
网络模块




```go
func (ps *PeerServer) SetConn(netId string, address string, con
    ps.netHandlerMap[netId].connMap.Lock()
    defer ps.netHandlerMap[netId].connMap.Unlock()
    if _, ok := ps.netHandlerMap[netId].connMap.conn[address];
    ps.netHandlerMap[netId].connMap.conn[address] = conn

    return nil
}
```

```go
type connMap struct {
    sync.Mutex
    conn map[string]*grpc.ClientConn
}
```

同步模块


sync
common
syncInterface
sync.go
syncBlock.go
syncHandler.go
syncReq.go

缓存模块

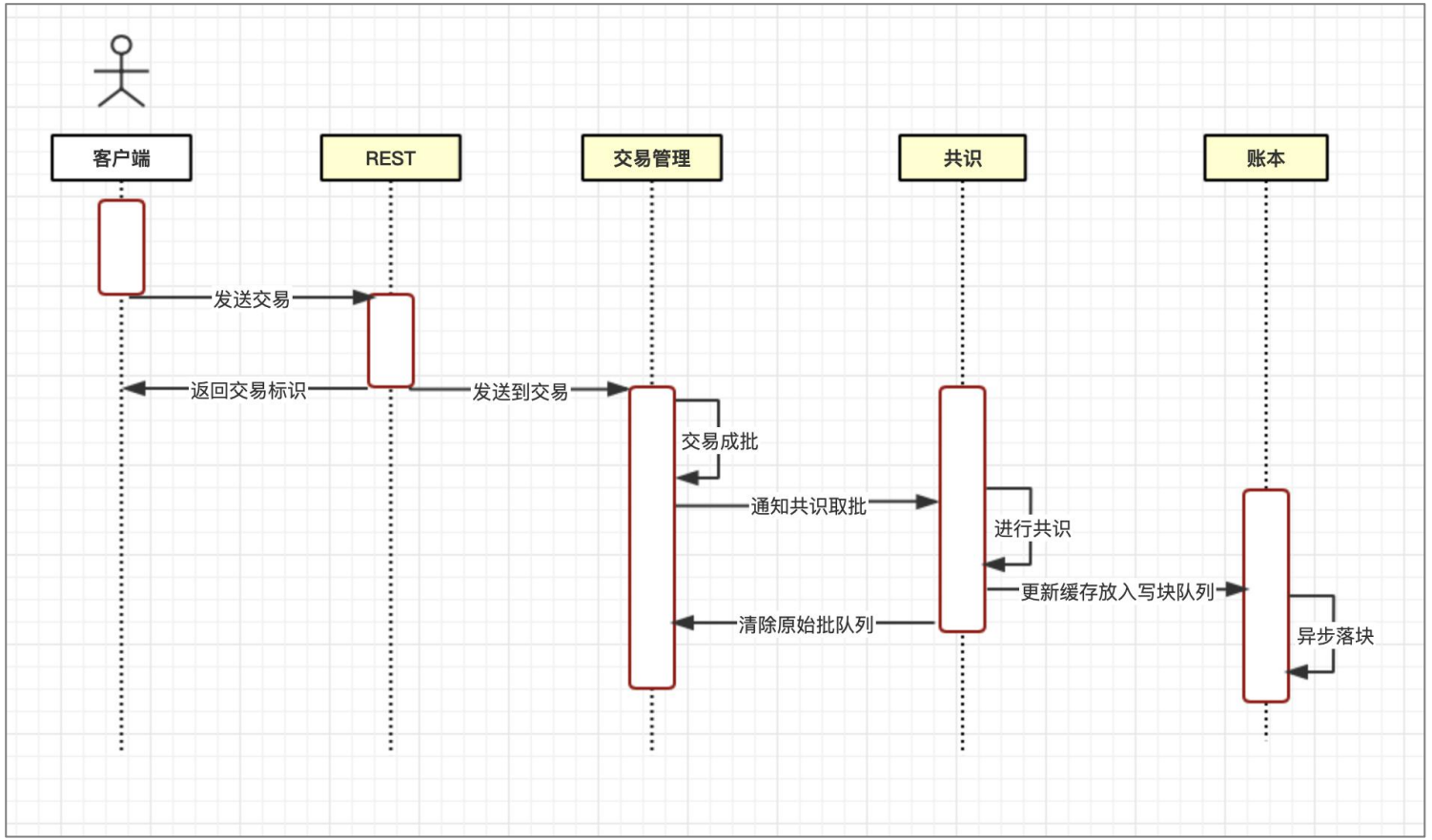

cache
cacheManager.go
uchainsCache.go
configMgr
db
helper
19 // 缓存策略||回调函数||超时
20 const (
21 Type_simple = "simple" // 简单模式 size可以为负数，
22 Type_lru = "lru" // 按照时间 size不能为负数
23 Type_lfu = "lfu" // 按照频率 size不能为负数
24 Type_arc = "arc" // 综合模式 size不能为负数
25)

优链的顶级目录


```
smallyu@smallyu-PC:~/workspace/go/src/uchains$ ls
api           consensus      externalApp  loggings       rest       umfnet
applications  db             gitpush.sh   monitorServer  script     vendor
cache         deploy         helper       msp            start.sh
configFile    deploy.tar.gz  ledger       network        sync
configMgr     eventServer    levelDB      peer           ulogging
```

目录

交易流程


客户端
REST
交易管理
共识
账本
发送交易
返回交易标识
发送到交易
交易成批
通知共识取批
进行共识
更新缓存放入写块队列
清除原始批队列
异步落块

（by 赵树伟）

启动Web服务

rest/rest.go

调用构建路由的函数

```
192  func StartRESTServer() {
193      restLogger.Info( args...: "* ====================
194      RegisterAppRestInstence() // 注册业务类型与业务实例
195      if len(MuxAppRestMap) == 0 {
196          restLogger.Errorf( format: "E-[start](%s)",
197          panic( v: "MuxAppRestMap is nil")
198      }
199      restLogger.Info( args...: "* ====================
200
201      restDelay = configMgr.Configration.GetPeerConfig
202      var iocCtx = uioc.GetIoCContext()
203      restService := iocCtx.Get( name: "serverREST").(*
204      router := buildOpenchainRESTRouter(restService)
205      restPort := configMgr.Configration.GetPeerConfig
```

构建路由1

rest/rest.go

```
99
100  // 构建router & 向router中注册url
101  func buildOpenchainRESTRouter(s *ServerREST) *web.Router {
102
103      s.genesis = &genesisTx{GTx: make(map[string]*pbNetwork.
104      GlobalIMap = &globalIInfo{}
105      GlobalIMap.GlobalIMapInfo = make(map[string]string)
106      server := ServerREST{...}
```

从配置文件读取路由

```
174      router.Post( path: "/UChains/posVote", server.PosVote)
175      /************************** app chain ***********************
176      // 将interfaceConfig.yaml中url & 合约配置db中url，注册到router
177      AppRegisterRouter(router)
178      //用于可信区块链测试，验证节点具有平滑升级功能
179      router.Get( path: "/UChains/testupdate/", server.TestUpdate)
```

构建路由2

rest/

registerRouter.go

```
55  /**
56   *   @MethodName:   AppRegisterRouter
57   *   @Description:   将配置文件中 & 合约注册的url，接口对应的url注册到路由
58   *  1. 服务启动时，router要注册的数据来源：配置文件 & db
59   *  2. 合约的rest接口有变化时，重新调用func AppRegisterRouter (router *web
60   * @param router
61   *   @author   zhaozj
62   *   @date   18-2-22   上午10:25
63   */
64  func AppRegisterRouter(router *web.Router) {
65      restLogger.Debugf( format: "R-[AppRegisterRouter](start|reload)")
```

注册路由到框架

```
124             loggerInfo := fmt.Sprintf("H-[AppRegisterRouter]->[router](#
125             restLogger.Info(loggerInfo)
126             if method == string(configMgrComm.POST) {
127                 router.Post(urlConfInfo, (*ServerREST).BaseRestHandler)
128             }
129             if method == string(configMgrComm.GET) {
130                 router.Get(urlConfInfo, (*ServerREST).BaseRestHandler)
```

构建路由3

rest/

baseRestHandler.go

代码抽象

```
26  /**
27   *   @MethodName:    BaseRestHandler
28   *   @Description:   处理所有router注册的url请求
29   * @param rw
30   * @param req
31   *   @author    zhaozj
32   *   @date    18-2-9    下午4:16
33   */
34  func (s *ServerREST) BaseRestHandler(rw web.Res
35
```

```
202             }
203             result := restConst.FormatInsertResRe
204             encoder.Encode(result)
205             return
206         }
207         // call template func
208         appRestHandler.ProcessTemplate(paraMap, e
209     }
```

构建路由4

rest/

restTemplate.go

代码抽象

```
203     /**
204      *   @MethodName:   RestProcessTemplate
205      *   @Description:   通用Rest接口模板
206      * @param paraMap
207      *   @author   zhaozj
208      *   @date   18-2-9   上午10:58
209      */
210     func (this *AppRestTemplate) ProcessTemplate
211         chainId, _ := paraMap[string(restConst.C
212         Tla = time.Now()
```

```
255             encoder.Encode(result)
256             return
257         }
258
259         // 4. 根据请求模式的差异处理
260         this.ReqModeHandler(paraMap, encoder)
261     }
```

构建路由5

rest/

restTemplate.go

代码抽象

```
142  /**
143   *   @MethodName:   ReqModeHandler
144   *   @Description:   根据请求模式的差异处理
145   * @param paraMap
146   * @param encoder
147   *   @author   zhaozj
148   *   @date   18-2-9   上午10:58
149   */
150  func (this *AppRestTemplate) ReqModeHandler(p
151      restLogger.Debugf( format: "R-[ReqModeHandl
152      method, _ := paraMap[string(cgComm.METHOD
```

```
185
186      // 2. 处理POST请求
187      if method == string(cgComm.POST) {
188          restLogger.Debugf( format: "C-[ha
189          this.handlerPostReq(chainId, que
190      }
191
```

构建路由6

rest/

restTemplate.go

调用提交交易的方法

```
428    // 处理PUT请求
429    func (this *AppRestTemplate) handlerPutReq(cha
430        // 如果queryId == 0xFF代表业务不需要查询任何数据
431        const typeRequest = "PUT"
432        // metrics.CommitTransaction.WithLabelValu
433        if queryId != 0xFF {
434            queryResult := this.AppService.Query(c
435            if queryResult.ErrMsg != "" && queryRe
436                restLogger.Warningf( format: "E-[har
437                result := restConst.FormatQueryRes
438                    restConst.RetCode(queryResult.
439                    queryResult.ErrMsg,  data: "")
440                encoder.Encode(result)
441                return
442            }
443        }
444        reqip := paraMap[string(cgComm.REQIP)]
445        result := this.commitTx(paraMap, chainId)
446        restLogger.Debugf( format: "RT-[handlerPutRe
447        encoder.Encode(result)
```

构建路由7

rest/

restTemplate.go

```
317
318  func (this *AppRestTemplate) commitTx(payload
319      restLogger.Debugf( format: "OP-[%s]",  args.
320      const typeRequest = "CommitTransaction"
321      // metrics.CommitTransaction.Inc()
322      //  metrics.CommitTransaction.WithLabelVa
```

把交易提交到交易管理器

```
398      //duyong add :测试rest层tps 不执行下面交易管理中接收到交易后的处理
399      //return  restConst.FormatInsertResResult(restConst.CODE_S
400
401      restReturn := this.TxManagerService.RecvTxMsg(txMsg)
402      if restReturn.ErrCode != string(restConst.CODE_STATUS_OK)
403          return restConst.FormatInsertResResult(restConst.RetCo
404      }
```

接收交易1

consensus/

txManager/

txManager.go

代码抽象

```
345  /**
346   *   @MethodName:      RecvTxMsg
347   *   @Description:   接收rest层提交过来的交易
348   * @param: msg
349   *   @author   wangjie
350   *   @date   2019/7/11   11:15
351   */
352  func (m *MultiChainTxManager) RecvTxMsg(msg
353      chainTxManager := m.getChainTxManager(m
354      if chainTxManager == nil {
355          txManagerLogger.Errorf( format: "(tx
356          return &appComm.RetQueryResult{ErrC
357      }
358
359      return chainTxManager.RecvTxMsg(msg)
360  }
```

接收交易2

consensus/

txManager/

multiType.go

代码抽象

```
350     *   @MethodName:        RecvTxMsg
351     *   @Description:
352     * @param:
353     *   @author    wangjie
354     *   @date    2019/7/11    10:43
355     */
356   func (ctm *ChainTxManager) RecvTxMsg(
357       txType, err := strconv.Atoi(msg.P
358       if err != nil {
359           txManagerLogger.Errorf( forma
360               msg.Payload[restConst.CHA
361           return &appComm.RetQueryResul
362       }
363       txManager := ctm.GetTxManager(txT
364
365       if txManager == nil {
366           txManagerLogger.Errorf( forma
367               msg.Payload[restConst.CHA
368           return &appComm.RetQueryResul
369       }
370       return txManager.RecvTxMsg(msg)
371   }
```

接收交易3

consensus/

txManager/

chainTxManager.go

把交易发送到通道（异步）

```
152  /**
153   *   @MethodName:       RecvTxMsg
154   *   @Description:    接收rest提交过来的交易
155   * @param:
156   *   @author    wangjie
157   *   @date    2019/7/11    11:35
158   */
159  func (m *TxManager) RecvTxMsg(msg *pb.Transa
160      select {
161      case m.recvTxMsgChan <- msg:
162          txPoolEnabled := m.txPoolEnabled
163          result := &appComm.RetQueryResult{}
164          if txPoolEnabled {
165              result = <-m.restReChan
166          } else {
167              result.ErrCode = string(restCons
168          }
169          return result
170      default:
```

处理交易1

consensus/

txManager/

chainTxManager.go

接收到交易

```
302  /**
303   *   @MethodName:      recvTxService
304   *   @Description:   接收交易处理服务
305   * @param:
306   *   @author   wangjie
307   *   @date   2019/7/11   11:35
308   */
309  func (m *TxManager) recvTxService() {
310
311      timerManager := new(TimerManager)
312      timerManager.timers = time.NewTimer(m.batchTi
313      timerManager.timersAvail = false
314
315      m.timerManager = timerManager
316      m.originalBatchs.timerManager = timerManager
317      for {
318          select {
319          case ocMsg := <-m.recvTxMsgChan:
320              //time.Sleep(time.Second) //慢点从交易
321              m.processTxMessage(ocMsg)
322          case <-m.originalBatchs.timerManager.time
323              txManagerLogger.Infof( format: "* chai
324              m.originalBatchs.closeCurrentBatch( i
325          }
```

处理交易2

consensus/

txManager/

chainTxManager.go

代码抽象

```
606  /**
607   *   @MethodName:      processTxMessage
608   *   @Description:   处理提交过来的交易
609   * @param:
610   *   @author   wangjie
611   *   @date   2019/7/11   11:40
612   */
613  func (m *TxManager) processTxMessage(peerMsg
614
615      if peerMsg == nil {
616          if m.txPoolEnabled {
617              m.restReChan <- &appComm.RetQuery
618          }
619          return fmt.Errorf( format: "peerMsg is
620      }
621
622      err := m.AddTx(peerMsg)
623      if err != nil {
624          txManagerLogger.Errorf( format: "add re
625      }
626      return err
627  }
```

处理交易3

consensus/

txManager/

chainTxManager.go

代码抽象

```
642  /**
643   *   @MethodName:      AddTx
644   *   @Description:   添加交易到当前批
645   * @param:
646   *   @author   wangjie
647   *   @date   2019/7/11   11:40
648   */
649  func (m *TxManager) AddTx(tx *pb.Transaction) error {
650      if tx == nil {
651          return fmt.Errorf("#{"Transaction is nil"}")
652      }
653      //去重
654      txManagerLogger.Debugf( format: "* chainID(%s).(add
655
656      err := m.originalBatchs.AddTxInCurrentBatch(tx)
657      if err != nil {
658          return err
659      }
660
661      return nil
662  }
```

处理交易4

consensus/

txManager/

originalBatches.go

代码抽象

```
437  /**
438   *   @MethodName:       AddTxInCurrentBatch
439   *   @Description:    添加交易到当前批中
440   * @param:
441   *   @author    wangjie
442   *   @date    2019/7/11    11:22
443   */
444  func (o *OriginalBatchs) AddTxInCurrentBatch(tx *pb.Transact
445      txManagerLogger.Debugf( format: "batchSize:[%v]", o.batch
446      var json = jsoniter.ConfigCompatibleWithStandardLibrary
```

```
476      }
477
478      txManagerLogger.Debugf( format: "runningTimer
479      _, err = o.currentBatch.AddTxInBatch(tx)
480      if err != nil : err ↵
483      return nil
484  }
```

处理交易5

consensus/

txManager/

chainTxManager.go

将交易添加到原始批队列

```
642  /**
643   *   @MethodName:       AddTx
644   *   @Description:    添加交易到当前批
645   * @param:
646   *   @author    wangjie
647   *   @date    2019/7/11    11:40
648   */
649  func (m *TxManager) AddTx(tx *pb.Transaction) error
650      if tx == nil {
651          return fmt.Errorf("#{"Transaction is nil"}")
652      }
653      //去重
654      txManagerLogger.Debugf( format: "* chainID(%s).(a
655
656      err := m.originalBatchs.AddTxInCurrentBatch(tx)
657      if err != nil {
658          return err
659      }
```

处理原始批1

consensus/

txManager/

txManager.go

初始化的时候启动了三个服务

```
180  /**
181   *   @MethodName:      startChainServer
182   *   @Description:   启动每个链的处理交易，广播批，处理批等服务
183   * @param: chainInfo 链配置信息
184   *   @author   wangjie
185   *   @date   2019/7/11   11:09
186   */
187  func (m *MultiChainTxManager) startChainServer(chainInfo
188      txManagerLogger.Infof( format: "(txManager).(chainID(%
189      chainTxManager := m.getChainTxManager(chainInfo.Chain
190      if chainTxManager == nil {
191          txManagerLogger.Errorf( format: "(MultiChainTxMana
192          return
193      } else {
194          err := chainTxManager.initBatchQueueAndConsensus(
195          if err != nil {
196              txManagerLogger.Errorf( format: "initBatchQueu
197              return
198          }
199          chainTxManager.recvTxService()
200          chainTxManager.handleBatchService()
201          chainTxManager.broadcastBatchService()
202      }
```

处理原始批2

consensus/

txManager/

multiType.go

代码抽象

```
546  /**
547   *   @MethodName:      handleBatchService
548   *   @Description:
549   * @param:
550   *   @author   wangjie
551   *   @date   2019/7/11   10:51
552   */
553  func (ctm *ChainTxManager) handleBatchService() {
554      ctm.RLock()
555      defer ctm.RUnlock()
556      for _, txMgr := range ctm.txManager {
557          go txMgr.handleBatchService()
558      }
559  }
```

处理原始批3

consensus/
txManager/
chainTxManager.go

通知共识模块处理批

```
422  /**
423   *   @MethodName:      handleBatchService
424   *   @Description:   批处理服务
425   * @param:
426   *   @author   wangjie
427   *   @date   2019/7/11   11:39
428   */
429  func (m *TxManager) handleBatchService() {
```

```
457
458          case batch := <-m.recvBatchMsgChan:
459              t1 := time.Now()
460              txManagerLogger.Debugf( format: "*
```

```
542                  //txManagerLogger.Infof("* chainID(%s) set batch %s t
543                  txManagerLogger.Infof( format: "* chainID(%s).(set bat
544                      m.chainID, batch.BatchId, originalBatch.GetBatchI
545
546                  m.originalBatchs.AddBatchCount()
547
548                  m.reverseNotify.NotifyCreateOriginalBatch(m.chainID)
549              }
550          }
```

处理原始批4

consensus/

txManager/

txManager.go

真的通知共识模块处理批

```
521  /**
522   *   @MethodName:      NotifyCreateOriginalBatch
523   *   @Description:   通知共识可以获取批
524   * @param: chainID 链ID
525   *   @author   wangjie
526   *   @date   2019/7/11   11:19
527   */
528  func (m *MultiChainTxManager) NotifyCreateOriginalBatch(
529      routeChainID := m.GetLastRouteChainID(chainID)
530      txManagerLogger.Infof( format: "NotifyCreateOriginalB
531      chainTxManager := m.getChainTxManager(routeChainID)
532      if chainTxManager == nil {
533          txManagerLogger.Errorf( format: "(txManager) chai
534          return
535      }
536      select {
537      case chainTxManager.notify <- struct{}{}:
538          txManagerLogger.Debugf( format: "chainID(%s).(not
539      default:
540          txManagerLogger.Infof( format: "chainID(%s).(noti
541      }
542  }
```

共识的启动1

consensus/

tendermint/

tendermint.go

```
71  /**
72   *   @MethodName:
73   *   @Description:   启动tendermint共识
74   * @param
75   *   @author   ranshiyou
76   *   @date   2018/4/25   11:32
77   */
78  func (tdmCore *TendermintCore) NewConsensusCore(coor
79      tendermintLogger.Infof( format: "*** Start Tenderm
80
```

```
110
111     bftMgr.MsgHandler = handler
112     tdmCore.BftMgr = bftMgr
113
114     bftMgr.Start()
115     tdmCore.Coor.SetNotify(hand
116
117     return nil
```

启动 BFT 管理器

共识的启动2

consensus/

tendermint/

common/

service.go

代码抽象

```
123  // Start implements Service by calling OnStart (
124  // returned if the service is already running or
125  // stopped service, you need to call Reset.
126  func (bs *BaseService) Start() error {
127      bs.Logger.Infof( format: "test bg in start %d
128      defer func() {
129          bs.Logger.Infof( format: "test bg out sta
130      }()
131      if atomic.CompareAndSwapUint32(&bs.started,
132          if atomic.LoadUint32(&bs.stopped) == 1 {
133              bs.Logger.Error(Fmt( format: "Not sta
134              return ErrAlreadyStopped
135          }
136          bs.Logger.Info(Fmt( format: "Starting %v"
137          err := bs.impl.OnStart()
138          if err != nil {
139              // revert flag
140              bs.Logger.Errorf( format: "OnStart er
141              atomic.StoreUint32(&bs.started, 0)
142              return err
143          }
144          return nil
```

共识的启动3

consensus/

tendermint/

msgHandler/

routineMgr.go

启动交易监听服务

```
122  func (rm *RoutineMgr) Start(h *TDMMsgHandler) {
123      go h.DeliverMsg(rm.MsgDeliverChan)
124      rm.SetActive(MsgDeliver)
125
126      go h.BroadcastMsg(rm.BroadCastMsgChan)
127      rm.SetActive(BroadCast)
128
129      go h.Syncer.syncServer(h.processOuterMsg, rm.SyncServerCha
130      rm.SetActive(SyncServer)
131
132      go h.checkBackward.Monitor(rm.BackwardMonitorChan) // add
133      rm.SetActive(BackwardMonitor)
134
135      go h.CheckFork.recvForkSearch(rm.RecvForkChan) // add for
136      rm.SetActive(RecvFork)
137
138      go h.CheckFork.Monitor(rm.ForkMonitorChan) // add for moni
139      rm.SetActive(ForkMonitor)
140
141      if h.Member.inbgGroup() {
142          // 监听是否有交易
143          go h.TxServer.CheckTxsAvailable(h, rm.TxsAvailChan)
144          rm.SetActive(TxsAvail)
145
```

共识接收交易1

consensus/

tendermint/

msgHandler/

txService.go

收到交易，处理

```
179    // 监听是否有交易
180    func (ts *TxServer) CheckTxsAvailable(tdm *TDMMsgHandler, quit
181        timer := time.NewTicker(10 * time.Second)
182        latCheckHeight := tdm.Height
183
184        for {
185            select {
186            case <-ts.coor.NotifyGetOriginalBatch(ts.chainID):...
197            case <-ts.notify:
198                ts.logger.Debugf( format: "ts.notify out")
199                if ts.coor.GetOriginalBatchsLen(ts.chainID) > 0 {
200                    select {
201                    case ts.txsAvailable <- tdm.Height:
202                        ts.logger.Infof( format: "NewRound need to e
203                    default:
204                        ts.logger.Infof( format: "last propose not c
205                    }
206                } else {
```

共识接收交易2

consensus/

tendermint/

msgHandler/

msgdeliver.go

代码抽象

```
44  func (h *TDMMsgHandler) DeliverMsg(quit chan struct{}) {
45      var peerMsg *txComm.PeerMessage
46      var err error
47      /*rs := h.RoundState*/
48      set := make(map[uint64]bool)
49
50      for {
51          select {
52          case height := <-h.TxServer.GetTxsChan():
53              h.Logger.Infof( format: "txsAvailable in")
54              //defer h.logger.Infof("txsAvailable out")
55              if !h.checkContinue() {
56                  continue
57              }
58              h.Logger.Infof( format: "#(%d-%d) (msgDeliver)
59              h.handleTxsAvailable(height)
60
61          case peerMsg = <-h.PeerMsgQueue:
62              h.Logger.Infof( format: "PeerMsgQueue in")
```

进入提案阶段

consensus/

tendermint/

msgHandler/

txService.go

进入提案处理

```
397  // 处理交易
398  func (h *TDMMsgHandler) handleTxsAvailable(height uint64) {
399      h.mtx.Lock()
400      defer h.mtx.Unlock()
401
402      // 是共识组成员，进入
403      if h.Member.inbgGroup() {
404          //if h.Round == 0 {
405          //  h.enterPropose(height, 0)
406          //} else {
407          //  h.enterPropose(height, h.Round + 1)
408          //}
409          h.enterPropose(height, h.Round)
410      }
411  }
```

选提案人

consensus/

tendermint/

msgHandler/

enterProposeHandler.go

```
11  // 进入条件
12  // Enter (CreateEmptyBlocks): from enterNewRound(height,round)
13  // Enter (CreateEmptyBlocks, CreateEmptyBlocksInterval > 0 ): afte
14  // Enter (!CreateEmptyBlocks) : after enterNewRound(height,round),
15  func (h *TDMMsgHandler) enterPropose(height uint64, round int32) {
16      if round >= 2 && round < 4 {
17          h.Logger.Warningf( format: "#(%d-%d-%v) enterPropose (%d-%d)
```

```
76      //selected, val := h.isProposerByVRF(height, round)
77      selected, val := h.ProposeHandler.GetIsProposerFunc()(h, h.Validators, h.La
78      if val == nil {
79          h.Logger.Warningf( format: "enterPropose | #(%d-%d-%v) no proposer selec
80          h.Logger.Infof( format: "enterPropose | #(%d-%d-%v) no proposer selected
81          return
82      }
83
84      if selected {
85          h.Logger.Infof( format: "enterPropose | #(%d-%d-%v) enterPropose Our tur
86          //h.scheduleTimeout(time.Second*5, height, round, types.RoundStepPropos
87          //todo  for test
88          //if h.Height%2 == 0 && h.Member.GetPeerIDToString() == "org3-vp3" {
89          //  h.Logger.Infof("i am a bad man, do not propose")
90          //} else {
91          //  h.ProposeHandler.decideProposal(height, round)
92          //}
93          h.ProposeHandler.decideProposal(height, round)
94      } else {
```

决定提案

关于选提案人的问题1


```
61  func isProposerByVrf(h *msgHandler.TDMMsgHandler, val *types.ValidatorSet, VRFValue []byte, heig
62      if round != 0 {
63          peer, err := h.ProposerInfo.GetRoundProposer(round - 1)
64          if err != nil {
65              h.Logger.Warningf( format: "GetRoundProposer err %s", err.Error())
66          } else {
67              h.Logger.Infof( format: "isProposerByVrf | peer=",peer, "height=", height,"round=", r
68              h.Logger.Infof( format: "isProposerByVrf | len(h.ProposerInfo.GetCurrentBlacklist())="
69              if (len(h.ProposerInfo.GetCurrentBlacklist()) >= len(val.Validators)) {
70                  h.ProposerInfo.Reset()
71              }
72              h.ProposerInfo.AddCurrentBlacklist(peer)
73          }
74      }
75      changeTarget := h.ProposerInfo.NeedToChangeProposer(round)
76      var proposer *types.Validator
77      if changeTarget {
78          h.Logger.Infof( format: "isProposerByVrf | CurrentBlacklist %v BlackList %v", h.ProposerI
79          proposer = vrf.CalcProposerByVRF(VRFValue, val,
80              h.ProposerInfo.GetCurrentBlacklist(), h.ProposerInfo.GetBlackList(), height, round)
81      } else {
```

关于选提案人的问题2


```
47      defer func() {
48          // Done enterPropose:
49          h.updateRoundStep(round, types.RoundStepPropose)
50
51          // If we have the whole proposal + POL, then goto Prevote now.
52          // else, we'll enterPrevote when the rest of the proposal is received (in AddProposalBlockPart),
53          // or else after timeoutPropose
54          if h.ProposeHandler.isProposalComplete() {
55              h.timeState.EndConsumeTime(h)
56              h.enterPrevote(height, h.Round)
57          }
58      }()
59      // If we don't get the proposal and all block parts quick enough, enterPrevote
60      h.scheduleTimeout(h.AddTimeOut(timeoutPropose), height, round, types.RoundStepPropose)
```

决定提案1

consensus/

tendermint/

msgHandler/

makeProposeMsg.go

调用发送网络消息方法

```
28    // 通过VRF 抽选出提案人
29    //func (ph *ProposeHandler) isProposerByVRF(height uint
30    //
31    //  ph.logger.Debugf("#(%v-%v) calc (%d-%d) proposer pre
32    //  proposer := vrf.CalcProposerByVRF(ph.tdm.LastCommitS
33    //  return bytes.Equal(ph.tdm.Member.GetAddr(), proposer
34    //}
35
36    func (ph *ProposeHandler) decideProposal(height uint64,
```

```
74        for i := 0; i < blockParts.Total(); i++ {
75            part := blockParts.GetPart(i)
76            ph.logger.Infof( format: "#(%v-%v) (block
77            //打印每个拆分块的大小
78            blkPartMsg, err := ph.buildBlockPartMsgF
79            if err != nil {
80                ph.logger.Errorf( format: "#(%v-%v) b
81                return
82            }
83
84            ph.tdm.sendInternalMessage(blkPartMsg)
85
86            /*h.BroadcastMsgToAll(blkPartMsg)*/
87        }
```

决定提案2

consensus/

tendermint/

msgHandler/

handler.go

代码抽象

```
372  // send a msg into the receiveRoutine regarding our own
373  func (h *TDMMsgHandler) sendInternalMessage(msg *txComm
374      h.Logger.Debugf( format: "#(%v-%v) (sendInternalMessa
375      select {
376      case h.InternalMsgQueue <- msg:
377      default:
378          // NOTE: using the go-routine means our votes c
379          // be processed out of order.
380          // attempt push to internalMsgQueue in receiveR
381          h.Logger.Info( args...: "Internal msg queue is fu
382          go func() { h.InternalMsgQueue <- msg }()
383      }
384  }
```

决定提案3

consensus/

tendermint/

msgHandler/

msgdeliver.go

```
44  func (h *TDMMsgHandler) DeliverMsg(quit chan struct{}) {
45      var peerMsg *txComm.PeerMessage
46      var err error
47      /*rs := h.RoundState*/
48      set := make(map[uint64]bool)
49
50      for {
51          select {
52          case height := <-h.TxServer.GetTxsChan():...
60
61          case peerMsg = <-h.PeerMsgQueue:...
108
109         case internalMsg := <-h.InternalMsgQueue:
110             //h.logger.Infof("InternalMsgQueue in")
```

调用处理方法

```
131             h.Logger.Infof( format: "(msgDeliver) %s recv internalM
132             err = h.processInnerMsg(tdmMsg, h.Member.GetPeerID())
133             if err != nil {
134                 h.Logger.Errorf( format: "(msgDeliver) %s process i
135             }
```

决定提案4

consensus/

tendermint/

msgHandler/

msgdeliver.go

代码抽象

```
533  /**
534   *   @MethodName:
535   *   @Description:   处理节点内部的消息
536   * @param
537   *   @author   shiyou
538   *   @date   2018/5/11   14:32
539   */
540  func (h *TDMMsgHandler) processInnerMsg(tdmMsg *tdm.TDMMessage,
541
542      var err error
543      switch tdmMsg.Type {
544      case tdm.TDMType_MsgProposal:
545          err = h.ProposeHandler.HandleInnerProposalMsg(tdmMsg)
546      case tdm.TDMType_MsgVote:...
550
551      case tdm.TDMType_MsgBlockPart:
552          err = h.ProposeHandler.HandleInnerBlockPartMsg(tdmMsg)
553
554      default:
555          h.Logger.Warningf( format: "(processOuterMsg) type %v no
556      }
```

决定提案5

consensus/

tendermint/

msgHandler/

proposeMsgHandler.go

代码抽象

```
127  func (ph *ProposeHandler) HandleInnerBlockPartMsg(tdmMsg
128      blkPartMsg := tdmMsg.Msg.(*tdmProto.TDMMessage_Block
129      ph.logger.Infof( format: "#(%v-%v) HandleInnerBlockPar
130
131      blkPart := ph.buildTypeBlkPartFromNetMsg(blkPartMsg)
132
133      //TODO shiyou 什么时候verify
134      _, err := ph.addProposalBlockPart(blkPartMsg.Height,
135      if err != nil {
136          ph.logger.Errorf( format: "(msgHandler) HandleInne
137          return nil
138      }
139
140      return nil
141  }
```

进入预投票

consensus/

tendermint/

msgHandler/

proposeMsgHandler.go

进入预投票

```
164  // NOTE: block is not necessarily valid.
165  // Asynchronously triggers either enterPrevote (before
166  func (ph *ProposeHandler) addProposalBlockPart(height u
167      // Blocks might be reused, so round mismatch is OK
168      if ph.tdm.Height != height : false, nil ↵
171
```

```
244          // Move onto the next step
245          ph.logger.Infof( format: "#(%v-%v-%v) propos
246
247          ph.tdm.enterPrevote(height, ph.tdm.Round)
248          //}
249          /* todo xujiaming fixv1
250          else if h.Step == types.RoundStepCommit {
```

预投票处理1

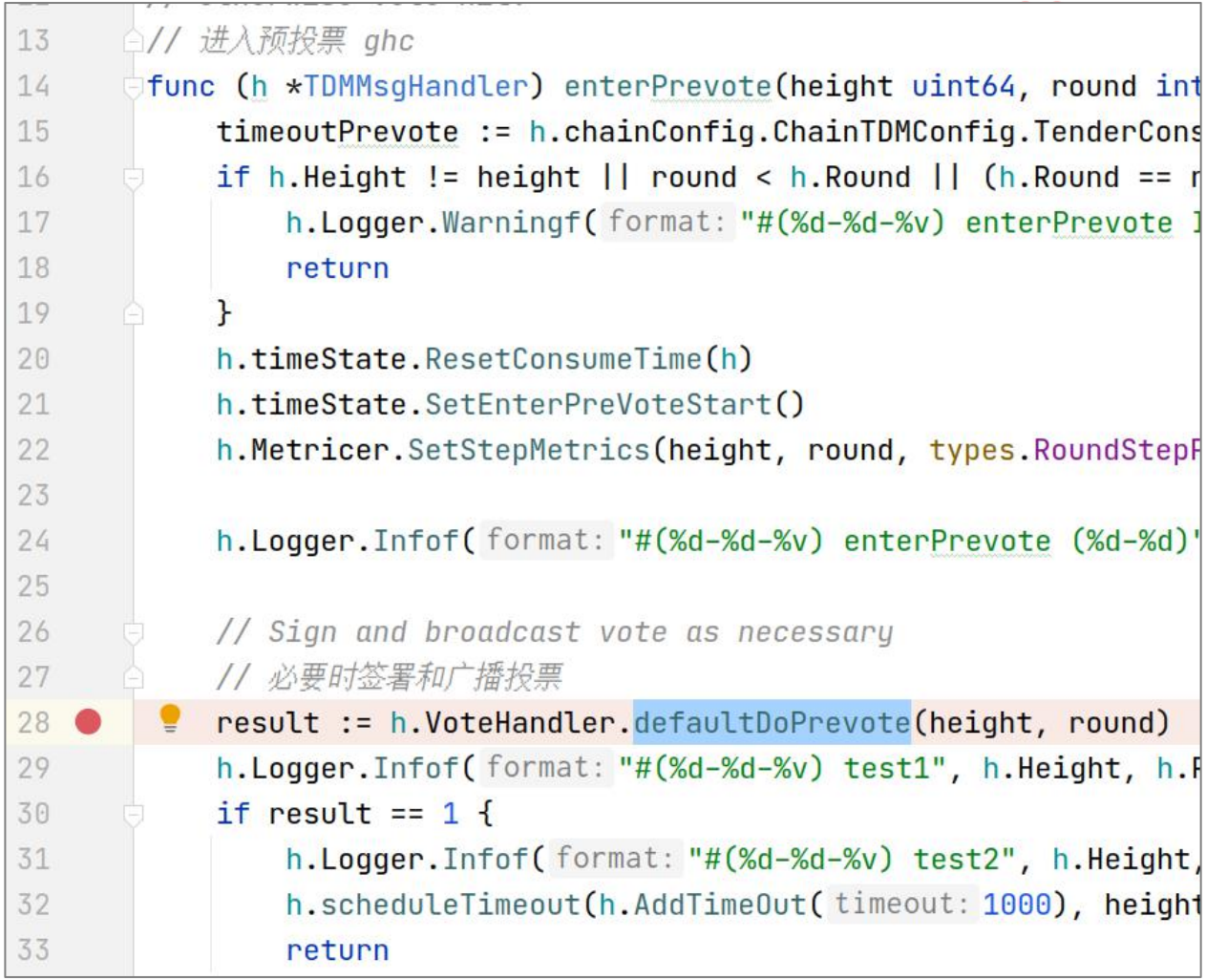
consensus/

tendermint/

msgHandler/

enterPrevoteMsgHandler.go

调用投票处理方法



```
13  // 进入预投票 ghc
14  func (h *TDMMsgHandler) enterPrevote(height uint64, round int
15      timeoutPrevote := h.chainConfig.ChainTDMConfig.TenderCons
16      if h.Height != height || round < h.Round || (h.Round == r
17          h.Logger.Warningf( format: "#(%d-%d-%v) enterPrevote
18          return
19      }
20      h.timeState.ResetConsumeTime(h)
21      h.timeState.SetEnterPreVoteStart()
22      h.Metricer.SetStepMetrics(height, round, types.RoundStepP
23
24      h.Logger.Infof( format: "#(%d-%d-%v) enterPrevote (%d-%d)"
25
26      // Sign and broadcast vote as necessary
27      // 必要时签署和广播投票
28      result := h.VoteHandler.defaultDoPrevote(height, round)
29      h.Logger.Infof( format: "#(%d-%d-%v) test1", h.Height, h.R
30      if result == 1 {
31          h.Logger.Infof( format: "#(%d-%d-%v) test2", h.Height,
32          h.scheduleTimeout(h.AddTimeOut( timeout: 1000), height
33          return
```

预投票处理2

consensus/

tendermint/

msgHandler/

makeVoteMsg.go

交易检测

```
18  func (vh *VoteHandler) defaultDoPrevote(height uint64,
19      // 测试更换共识组，投nil票，测试完成注释掉
20      /*h.signAddVote(types.VoteTypePrevote, nil, types.
21      return*/
```

```
69          // checkType判断
70          if !vh.tdm.IsCheck() {
71              /*bitMap ,err:=vh.tdm.getBitMap(vh.chainID,vh.tdm.Height,ckDat
72              if err!=nil{
73                  vh.logger.Errorf("vh.tdm.getBitMap err %v=",err)
74                  return 0
75              }*/
76              bitMap, _, _, _ := vh.tdm.TxServer.TxsCheckBeforeAfter(txs)
77              vh.logger.Debugf( format: "#(%v-%v) (defaultDoPrevote) bitMapLe
78              for index := uint32(0); index < bitMap.GetAllLen(); index++ {
79                  if util.CompareByteAndBool(bitMap.Get(index), b: false) {
80                      tx := txs[int(index)]
81                      if int(index) < len(txs) && tx != nil {
82                          vh.logger.Warningf( format: "#(%v-%v) (defaultDoPre
83                      }
```

等待预投票信息

consensus/

tendermint/

msgHandler/

msgdeliver.go

```
425  /**
426   *   @MethodName: handleTimeout
427   *   @Description:   某一阶段超时后的处理
428   * @param
429   *   @author   ranshiyou
430   *   @date   2018/4/26   16:55
431   */
432  func (h *TDMMsgHandler) handleTimeout(ti timeoutInfo,
433      h.Logger.Info( args...: "Received tock", "timeout"
434      if ti.Height != h.Height || ti.Round < h.Round ||
435          h.Logger.Info( args...: "Ignoring tock because
436          /*h.scheduleTimeout(time.Second*10, rs.Height
437          return
438      }
```

```
456      case types.RoundStepPrevote:
457          h.timeState.EndConsumeTime(h)
458          h.enterPrevoteWait(ti.Height, ti.Round)
459      case types.RoundStepPrevoteWait:
```

等待预投票信息

进入预提交阶段1

consensus/
tendermint/
msgHandler/
enterPrevoteMsgHandler.go

进入预提交阶段

```
47  //Enter: any +2/3 prevotes at next round.
48  func (h *TDMMsgHandler) enterPrevoteWait(height uint64, round int32) {
49      timeoutPrevoteWait := h.chainConfig.ChainTDMConfig.TenderConsensus.T
50      if h.Height != height || round < h.Round || (h.Round == round && type
51          h.Logger.Warning(cmn.Fmt( format: "#(%v-%v-%v) enterPrevoteWait (%
52          return
53      }
54
55      h.Logger.Infof(cmn.Fmt( format: "#(%v-%v-%v) enterPrevoteWait (%v-%v)"
56      h.Metricer.SetStepMetrics(height, round, types.RoundStepPrevoteWait)
57
58      defer func() {
59          // Done enterPrevoteWait:
60          h.updateRoundStep(round, types.RoundStepPrevoteWait)
61      }()
62
63      // Wait for some more prevotes; enterPrecommit
64      h.scheduleTimeout(h.AddTimeOut(timeoutPrevoteWait), height, round, ty
65  }
```

进入预提交阶段2

consensus/

tendermint/

msgHandler/

msgdeliver.go

```
425  /**
426   *   @MethodName: handleTimeout
427   *   @Description:   某一阶段超时后的处理
428   * @param
429   *   @author   ranshiyou
430   *   @date   2018/4/26   16:55
431   */
432  func (h *TDMMsgHandler) handleTimeout(ti timeoutInfo,
433      h.Logger.Info( args...: "Received tock", "timeout"
434      if ti.Height != h.Height || ti.Round < h.Round ||
435          h.Logger.Info( args...: "Ignoring tock because
436          /*h.scheduleTimeout(time.Second*10, rs.Height
437          return
438      }
```

```
456      case types.RoundStepPrevote:
457          h.timeState.EndConsumeTime(h)
458          h.enterPrevoteWait(ti.Height, ti.Round)
459      case types.RoundStepPrevoteWait:
460          //cs.eventBus.PublishEventTimeoutWait(cs
461          h.enterPrecommit(ti.Height, ti.Round)
```

调用预提交处理方法

预提交投票

consensus/

tendermint/

msgHandler/

enterPrecommitMsgHandler.go

发送预提交的投票消息

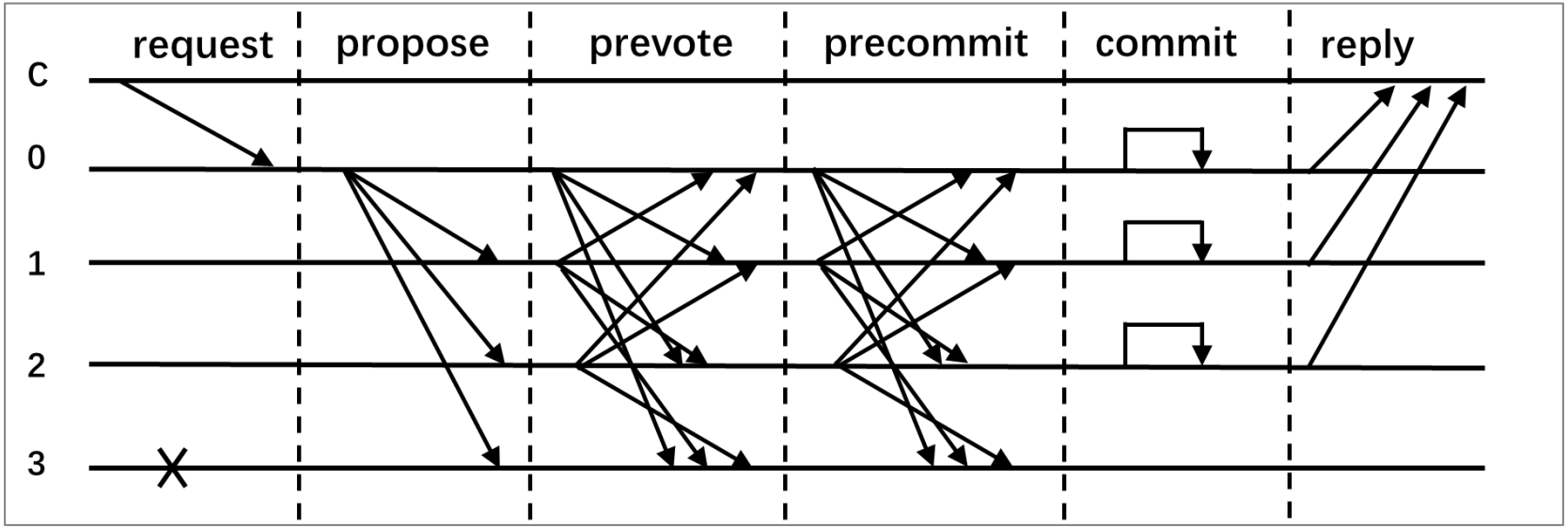
```
7
8   func (h *TDMMsgHandler) enterPrecommit(height
9       timeoutPrecommit := h.chainConfig.ChainTD
10      if h.Height != height || round < h.Round
11          h.Logger.Debugf(cmn.Fmt( format: "#(%v
```

```
38      blockID, ok := h.Votes.Prevotes(round).TwoThirdsMajority()
39      if !ok {
40          h.Logger.Debugf( format: "#(%v-%v-%v) enterPrecommit (%v
41          h.VoteHandler.applyAddVote(types.VoteTypePrecommit, has
42          return
43      }
44      //TODO ChenZheng 2018-5-14 内部事件通知
```

```
59      // If +2/3 prevoted for proposal block, stage and precommit it
60      if h.ProposalBlock.HashesTo(blockID.Hash) {
61          // Validate the block.
62          h.LockedRound = round
63          h.LockedBlock = h.ProposalBlock
64          h.LockedBlockParts = h.ProposalBlockParts
65
66          h.VoteHandler.applyAddVote(types.VoteTypePrecommit, blockI
67          h.Logger.Infof( format: "#(%v-%v-%v) enterPrecommit: +2/3 p
68          return
69      }
```

交易流程

Tendermint



预提交投票处理

consensus/

tendermint/

msgHandler/

voteMsgHandler.go

进入提交块

```
78  func (vh *VoteHandler) addVote(vote *types.Vote, peerID
79      vh.logger.Infof( format: "#(%v-%v) addVote (%v-%v) %
```

```
135     case types.VoteTypePrevote:...
163     case types.VoteTypePrecommit:
164         precommits := vh.tdm.Votes.Precommits(vote.Round)
165         vh.logger.Infof( format: "#(%v-%v) alreadyPrecommits
166         blockID, ok := precommits.TwoThirdsMajority()
167         if ok {
168             if len(blockID.Hash) == 0 {
169                 vh.logger.Warningf( format: "#(%v-%v) addVot
170                 vh.tdm.enterNewRound(height, vote.Round+1)
171             } else {
172                 vh.tdm.enterNewRound(height, vote.Round)
173                 vh.tdm.enterPrecommit(height, vote.Round)
174                 vh.tdm.enterCommit(height, vote.Round)
175
176                 /* todo xujiaming fixv1
177                 if h.config.SkipTimeoutCommit && precommits
```

写块

提交块之后开始写块，包括调用合约的AppProcsss

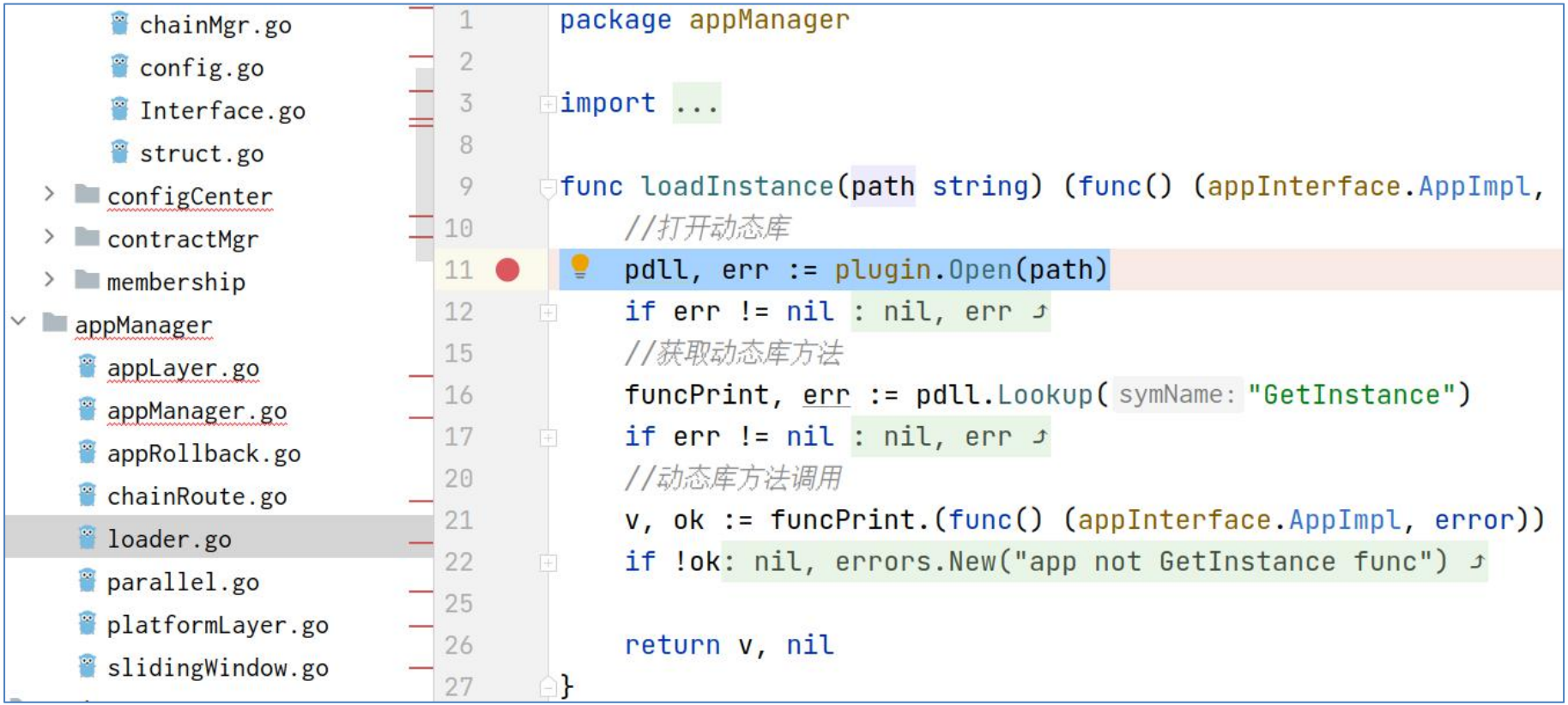
题外

```
if selected {
    h.Logger.Infof( format: "enterPropose | #(%d-%d-%v) enterPropose Our turn to propose,require:%s,i am %s", h.He
    //h.scheduleTimeout(time.Second*5, height, round, types.RoundStepPropose)
    //todo  for test
    //if h.Height%2 == 0 && h.Member.GetPeerIDToString() == "org3-vp3" {
    //  h.Logger.Infof("i am a bad man, do not propose")
    //} else {
    //  h.ProposeHandler.decideProposal(height, round)
    //}
    h.ProposeHandler.decideProposal(height, round)
} else {
    //  h.scheduleTimeout(time.Second*50, height, round, types.RoundStepPrecommitWait)
    if val == nil {...}
    h.Logger.Infof( format: "enterPropose | #(%d-%d-%v) enterPropose  Not our turn to propose,require:%s,i am %s",
}
```

tail -f peer.log | grep "ur turn"　　（ by 赵树伟 ）

目录

加载合约



目录

感谢聆听　敬请指正