(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107526624 A
(43)申请公布日 2017.12.29

(21)申请号 201710585453.9

(22)申请日 2017.07.18

(71)申请人 杭州趣链科技有限公司
地址 310012 浙江省杭州市西湖区文三路199号13幢南楼501室

(72)发明人 梁秀波 邱炜伟 李启雷 李伟 尹可挺

(74)专利代理机构 杭州求是专利事务所有限公司 33200
代理人 邱启旺

(51)Int.Cl.
G06F 9/455(2006.01)
G06Q 20/40(2012.01)

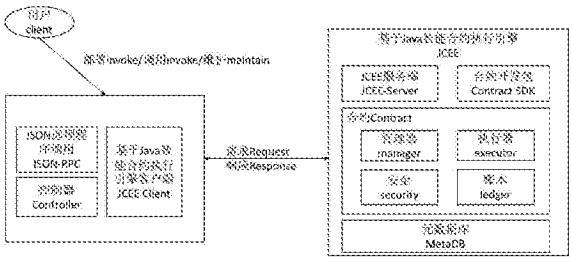
权利要求书1页 说明书3页 附图1页

(54)发明名称

一种基于Java虚拟机的智能合约执行引擎

(57)摘要

本发明公开了一种基于Java虚拟机的智能合约执行引擎JCEE，JCEE允许用户使用Java语言进行智能合约的开发，提供了智能合约完整生命周期管理。JCEE采用微服务的架构设计，分为客户端和服务端，客户端负责接收智能合约的调用请求，服务端负责具体智能合约的执行工作。JCEE的执行包括如下步骤：(1)客户端接收用户的合约调用请求并进行请求的完备性检查；(2)客户端将安全的用户请求封装并传递给JCEE服务端；(3)服务端首先进行合约的安全性检查，对安全的执行请求调用执行器进行具体的请求执行并返回执行结果。本发明提出了一种全新的智能合约执行引擎设计方案，解决了现有智能合约执行引擎通用性不足性能低下等问题。



CN 107526624 A

1.一种基于Java虚拟机的智能合约执行引擎,其特征在于,包括如下模块:

(1)基于Java虚拟机的智能合约执行引擎JCEE客户端:接收用户发送的智能合约调用请求,进行请求安全性检查、权限检查和参数合法性检查工作,该客户端模块包括JSON远程程序调用JSON-RPC模块、控制器模块以及JCEE客户端JCEE-Client模块等三个子模块,其中:

JSON-RPC模块负责接收解析来自用户的智能合约调用请求;

控制器模块维护客户端与JCEE服务端之间的通信安全以及心跳检测;

JCEE-Client模块负责具体智能合约的请求转发。

(2)JCEE服务端:JCEE服务端负责对用户智能合约的管理以及具体合约的执行工作;JCEE服务端包括JCEE服务器JCEE-Server、合约开发包Contract SDK、合约Contract模块以及元数据库MetaDB等四个子模块,其中:

JCEE-Server作为JCEE的服务器,管理同JCEE-Client模块的通信服务,将合约执行请求转发给下级处理器进行实际执行;

Contract SDK实现用户撰写智能合约相关的服务类,提供具体的合约编写支持;

Contract模块由四个独立的子模块构成:管理模块manager,负责合约生命周期的管理包括部署、执行、升级、冻结、解冻以及注销;执行模块executor:通过线程池的方式使得多个名字空间中的合约调用并行执行;安全模块security:通过基于字节码检查的机制进行合约安全性管控;账本模块ledger:为智能合约提供可以存取智能资产的功能;

MetaDB实现对合约元数据进行存储,包括合约名、合约创建时间、合约地址、合约创建者以及合约存储地址。

2.如权利要求1所述的一种基于Java虚拟机的智能合约执行引擎,其特征在于,所述的JCEE客户端中合约的调用请求通过JCEE客户端进行接收和转发,JCEE客户端可以内嵌入任何现有区块链平台中,合约的具体调用是通过客户端对用户合约调用请求的再封装。

3.如权利要求1所述的一种基于Java虚拟机的智能合约执行引擎,其特征在于,所述的JCEE服务端中安全模块中的安全检查组件通过字节码层面细粒度控制Java智能合约的安全性,以及通过线程池的方式实现多个名字空间中智能合约的并发执行。

一种基于Java虚拟机的智能合约执行引擎

技术领域

[0001] 本发明涉及区块链技术、智能合约引擎，尤其涉及一种基于Java虚拟机(简称JVM)的智能合约执行引擎。

背景技术

[0002] 区块链技术，区块链是一种新型去中心化协议，能安全地存储数字货币交易或其他数据，信息不可伪造和篡改，区块链上的交易确认由区块链上的所有节点共同完成。智能合约是部署在区块链上的一段可自动执行的程序，广泛意义上的智能合约包含编程语言、编译器、虚拟机、事件、状态机、容错机制等。其中，对应用程序开发影响较大的是编程语言以及智能合约的执行引擎，即虚拟机。虚拟机作为沙箱被封装起来，整个执行环境都被完全隔离。虚拟机内部执行的智能合约不能接触网络、文件系统或者系统中的其他线程等系统资源。合约之间只能进行有限调用。

[0003] 现有的智能合约执行引擎要不在合约的执行性能上达不到生产系统的要求，要不在安全性管理上存在漏洞。而智能合约直接同用户的链上资产交互，其安全性至关重要。

发明内容

[0004] 面对现有智能合约执行引擎的不足，本发明提出了一种基于Java虚拟机的智能合约执行引擎(JVM based Contract Execution Engine，以下简称JCEE)，该执行引擎允许合约编写者使用成熟的图灵完备的Java语言进行智能合约编写并且实现了字节码层面的安全性检查。

[0005] 基于Java虚拟机的智能合约引擎JCEE的设计以一种微服务的架构提供服务，主要包括客户端和服务端，服务端是智能合约执行引擎的主要模块而客户端则提供了一种访问合约执行引擎的能力。一种基于Java虚拟机的智能合约执行引擎通过模块化设计允许嵌入到多种区块链网络，具体技术方案如下：

[0006] 一种基于Java虚拟机的智能合约执行引擎，包括如下模块：

[0007] (1)基于Java虚拟机的智能合约执行引擎JCEE客户端：接收用户发送的智能合约调用请求，进行请求安全性检查、权限检查和参数合法性检查工作，该客户端模块包括JSON远程程序调用JSON-RPC模块、控制器模块以及JCEE客户端JCEE-Client模块等三个子模块，其中：

[0008] JSON-RPC模块负责接收解析来自用户的智能合约调用请求；

[0009] 控制器模块维护客户端与JCEE服务端之间的通信安全以及心跳检测；

[0010] JCEE-Client模块负责具体智能合约的请求转发；

[0011] (2)JCEE服务端：JCEE服务端负责对用户智能合约的管理以及具体合约的执行工作；JCEE服务端包括JCEE服务器JCEE-Server、合约开发包Contract SDK、合约Contract模块以及元数据库MetaDB等四个子模块，其中：

[0012] JCEE-Server作为JCEE的服务器，管理同JCEE-Client模块的通信服务，将合约执

行请求转发给下级处理器进行实际执行；

[0013] Contract SDK实现用户撰写智能合约相关的服务类，提供具体的合约编写支持；

[0014] Contract模块由四个独立的子模块构成：管理模块manager，负责合约生命周期的管理包括部署、执行、升级、冻结、解冻以及注销；执行模块executor：通过线程池的方式使得多个名字空间中的合约调用并行执行；安全模块security：通过基于字节码检查的机制进行合约安全性管控；账本模块ledger：为智能合约提供可以存取智能资产的功能；

[0015] MetaDB实现对合约元数据进行存储，包括合约名、合约创建时间、合约地址、合约创建者以及合约存储地址。

[0016] 进一步地，所述的JCEE客户端中合约的调用请求通过JCEE客户端进行接收和转发，JCEE客户端可以内嵌入任何现有区块链平台中，合约的具体调用是通过客户端对用户合约调用请求的再封装。

[0017] 进一步地，所述的JCEE服务端中安全模块中的安全检查组件通过字节码层面细粒度控制Java智能合约的安全性，以及通过线程池的方式实现多个名字空间中智能合约的并发执行。

[0018] 本发明的有益效果：

[0019] 本发明的基于JVM的智能合约执行引擎，允许用户使用图灵完备的Java语言直接进行智能合约的编写，既提高了只能合约的编写效率也提高了合约的执行效率。本发明采用基于字节码的细粒度合约安全检查机制并将合约的执行限制在虚拟机中从而实现了合约的沙箱化管理，从根本上杜绝了合约与网络、文件系统等系统资源的直接接触。此外本发明中通过线程池机制合约执行处理器设计提高了合约的整体执行效率。

附图说明

[0020] 图1是基于Java虚拟机的智能合约执行引擎的架构图；

[0021] 图2是基于Java虚拟机的智能合约执行引擎的请求执行流程图。

具体实施方式

[0022] 下面根据附图和具体实施例详细描述本发明，本发明的目的和效果将变得更加明显。

[0023] 如图1所示，本发明的基于Java虚拟机的智能合约执行引擎，包括如下模块：

[0024] (1)基于Java虚拟机的智能合约执行引擎JCEE客户端：接收用户发送的智能合约调用请求，进行请求安全性检查、权限检查和参数合法性检查工作，该客户端模块包括JSON远程程序调用JSON-RPC模块、控制器模块以及JCEE客户端JCEE-Client模块等三个子模块，其中：

[0025] JSON-RPC模块负责接收解析来自用户的智能合约调用请求；

[0026] 控制器模块维护客户端与JCEE服务端之间的通信安全以及心跳检测；

[0027] JCEE-Client模块负责具体智能合约的请求转发；

[0028] (2)JCEE服务端：JCEE服务端负责对用户智能合约的管理以及具体合约的执行工作；JCEE服务端包括JCEE服务器JCEE-Server、合约开发包Contract SDK、合约Contract模块以及元数据库MetaDB等四个子模块，其中：

[0029] JCEE-Server作为JCEE的服务器，管理同JCEE-Client模块的通信服务，将合约执行请求转发给下级处理器进行实际执行；

[0030] Contract SDK实现用户撰写智能合约相关的服务类，提供具体的合约编写支持；

[0031] Contract模块由四个独立的子模块构成：管理模块manager，负责合约生命周期的管理包括部署、执行、升级、冻结、解冻以及注销；执行模块executor：通过线程池的方式使得多个名字空间中的合约调用并行执行；安全模块security：通过基于字节码检查的机制进行合约安全性管控；账本模块ledger：为智能合约提供可以存取智能资产的功能；

[0032] MetaDB实现对合约元数据进行存储，包括合约名、合约创建时间、合约地址、合约创建者以及合约存储地址。

[0033] 所述的JCEE客户端中合约的调用请求通过JCEE客户端进行接收和转发，JCEE客户端可以内嵌入任何现有区块链平台中，合约的具体调用是通过客户端对用户合约调用请求的再封装。

[0034] 所述的JCEE服务端中安全模块中的安全检查组件通过字节码层面细粒度控制Java智能合约的安全性，以及通过线程池的方式实现多个名字空间中智能合约的并发执行。

[0035] 下面模拟一个合约调用的过程来说明基于Java虚拟机的智能合约执行引擎的具体执行过程。首先用户向JCEE-Client发送一个JS对象标记JSON的请求，JCEE-Client模块的JSON远程程序调用JSON-RPC模块接收到用户请求并封装成内部的交易transaction；接着该transaction会发送到JCEE-Client子模块，该模块会进行交易transaction的安全性以及参数的完整性能进行初步的安全性检查；安全性检查之后JCEE-Client通过远程程序调用协议将合约调用发送给JCEE-Server端，JCEE-Server端对调用进行封装并就分装的任务按照名字空间进行派发。最后基于线程池的任务执行器对这些派发的任务进行具体的执行。至此完成一个合约的调用。

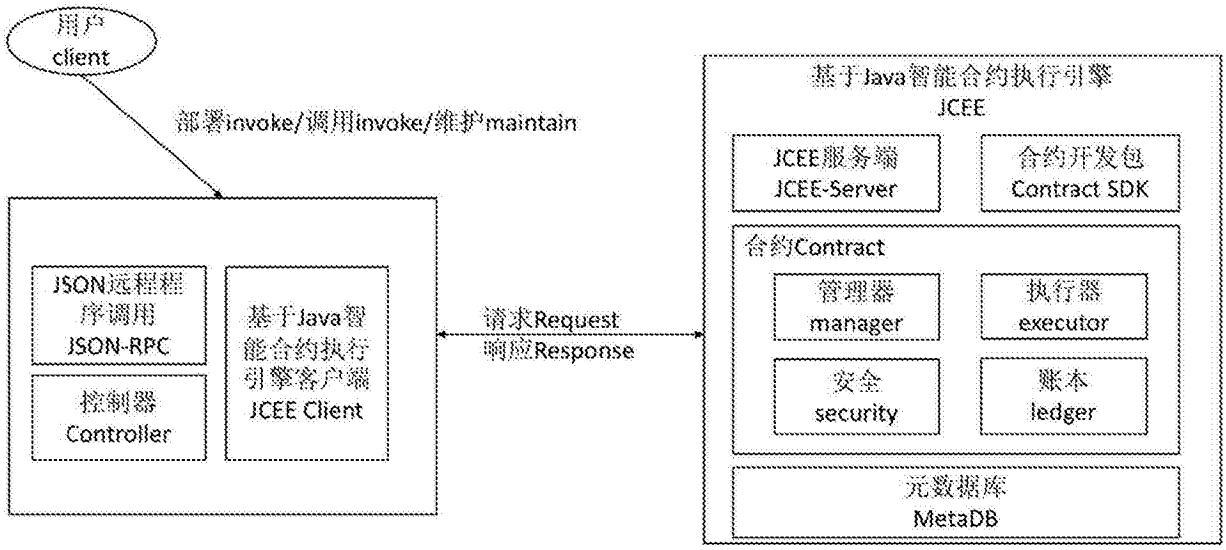

用户
client
部署invoke/调用invoke/维护maintain
JSON远程程序调用
JSON-RPC
控制器
Controller
基于Java智能合约执行引擎客户端
JCEE Client
请求Request
响应Response
基于Java智能合约执行引擎
JCEE
JCEE服务端
JCEE-Server
合约开发包
Contract SDK
合约Contract
管理器
manager
执行器
executor
安全
security
账本
ledger
元数据库
MetaDB

图1

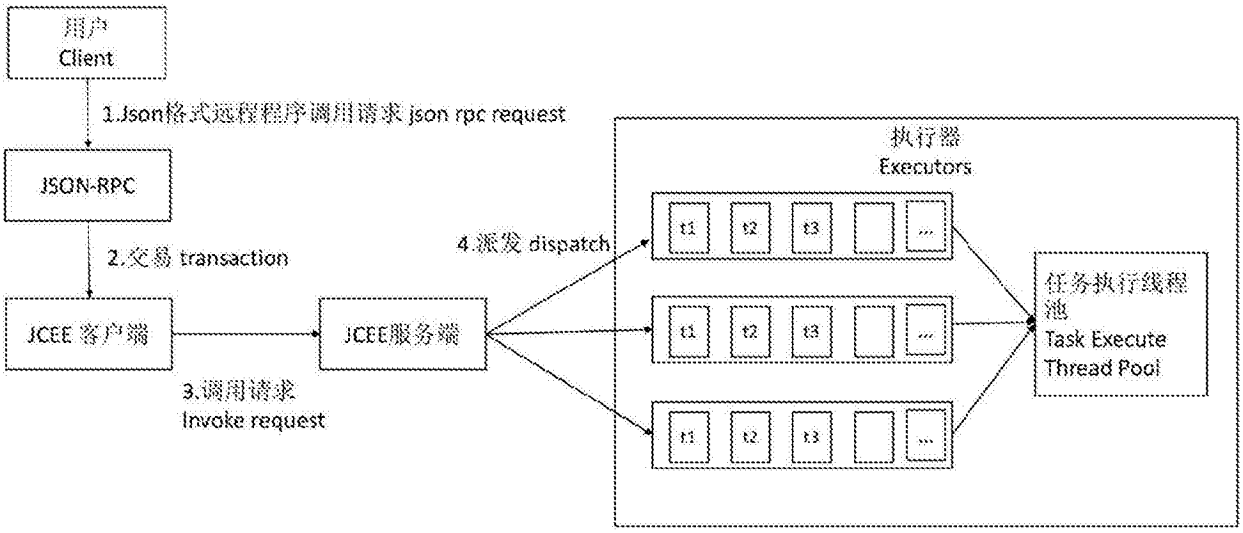

用户
Client
1.Json格式远程程序调用请求 json rpc request
JSON-RPC
2.交易 transaction
JCEE 客户端
3.调用请求
Invoke request
JCEE服务端
4.派发 dispatch
执行器
Executors
t1
t2
t3
...
任务执行线程池
Task Execute
Thread Pool

图2