

多区块链交易分发和事件处理的系统方案

刘雄文

(上海壹帐通金融科技有限公司 上海 201102)

摘要 在基于拜占庭容错协议设计的区块链系统中,系统的效率和性能较低,限制了区块链的应用与发展。提出了一种基于多区块链系统支持交易分片和事件处理的系统方案。该方案中,系统被划分为应用客户端、处理系统、各区块链 3 个层面。应用客户端用于发送交易请求和接收感兴趣的事件,处理系统进行交易的分发和事件处理,各区块链用于区块链内部处理交易请求和事件。处理系统适配区块链数量可变和不可变的两类多区块链系统,设计了选择区块链的两种策略,即映射关系表和哈希一致性,两种策略在交易分发时依据负载均衡、单调性和一致性的原则,并基于交易的业务类型和业务类型内部 ID 号把交易请求分发到合适的区块链。每个区块链与处理系统内部的交易调用客户端、事件和消息流客户端建立一一对应关系;每个应用客户端与处理系统内部的交易应用调用服务端、交易应用事件连接服务端、事件消息流服务端交互,发送交易的请求并获取交易的结果。对交易分发的方案和事件处理进行具体的设计,并对系统的整体方案进行了基本的测试和验证。

关键词 多区块链,一致性,交易分发,事件处理,消息流

中图分类号 TP311 文献标识码 A

System Solution of Multi-blockchain Transaction Dispatching and Event Handling

LIU Xiong-wen

(Shanghai OneConnect Technology Co., LTD., Shanghai 201102, China)

Abstract In the blockchain system based on the Byzantine Fault Tolerance (BFT) protocol, the efficiency and performance of system are poor, which limited the application and development of blockchain. A system solution of supporting transaction dispatching and event handling based on multiple blockchains was proposed. In the solution, the system is divided into application clients, handling system and blockchains. Application clients are used to send out transaction requests and receive the events which are interested in. Every blockchains are used to internally handle transaction requests and events. The handling system is adaptive to the system of variable number of blockchain and the system of immutable number of blockchain. Two policies of selecting blockchain, including mapping relationship table and hash consistency algorithm, are designed. According to the load balance, monotonicity and consistency, both of these two policies dispatch transactions based on the service type and the internal ID of service type. Every blockchain set up the one-to-one correspondence with transaction calling client, event and message streaming client. Every application client sends out the transaction requests and receives the transaction results by interacting with transaction application calling server, transaction application event connection server and event message streaming server. Transaction dispatcher and event handler were designed in detail, and basic test and verification were made for the whole system solution.

Keywords Multiple blockchain, Consistency, Transaction dispatching, Event handling, Message stream

1 引言

区块链是利用加密链式区块结构来验证与存储数据、利用分布式节点共识算法来生成和更新数据、执行智能合约(应用级代码等)来编程和操作数据的一种全新的去中心化基础架构与分布式计算范式^[1],它是金融科技领域的创新技术之一。区块链具有去中心化、共识机制、安全机制、公开透明、时序且不可篡改^[2]等特征。去中心化的特征是指在网络连接正常的情况下,区块链系统上的任意两个计算节点都互联互通,没有一个计算节点是中心化的节点。共识机制则用来解决闻名已久的拜占庭将军问题,拜占庭将军问题在区块链领域是指计算节点在可能不是全部可靠的基础上如何就计算结果达

成一致。安全机制是指在共识机制的基础上结合计算机安全技术使区块链安全可靠,文献^[3]提出了一种数据安全共享网络体系。公开透明的特点是指区块链系统的运作规则是公开透明的。时序和不可篡改的特征是指区块链的数据的存储具有时间戳和时间上的序列化,不可篡改,并可进行追溯和验证等。区块链可以应用到多个领域及多种场景,包括票据与供应链金融业务、证券发行与交易、客户征信与反欺诈、身份验证和电子商务等^[4]。教育领域开始使用区块链技术,基于区块链的成绩单、基于区块链技术的学习者资源管理和学习账本呈现了区块链技术在全球教育领域的发展和应用现状^[5]。基于区块链的防伪技术使系统更加安全,这利用了区块链数据的不可篡改的特性^[6]。区块链具有加密签名体制,

可以在公平合同签署的应用场景中解决隐私保护的问题^[7]。区块链在股权资产购买和转赠的应用场景中也有了设计与实现^[8]。区块链在众筹领域的应用场景的智能合约也有了设计^[9]，在金融领域中正在探索的应用还包括数字货币、清结算、支付、资产管理等，其他正在开展的应用领域包括物联网、通信、慈善捐赠、投票、医疗^[10]等。虽然区块链具有广阔的应用前景，但是区块链自身的运行效率还较低，需要通过各种方式来提升区块链效率^[11]。在传统的分布式系统中，有根据数据分片中热数据的所占比例作为热负载值以确定数据迁移时机^[12]来提升数据访问性能的方式，还有通过集群、分区、分片三层拆分模式，并通过读写分离、单库分表、多库分表等多种解决方案将数据按照一定的规则切分到不同的库或表中，从而提升数据库的访问性能^[13]，另外通过分片预留的方式也可提升分片的灵活性和效率，即在保证最大分片间隔的前提下，允许对资源进行分片预留。在各分片中，若存在剩余资源量小于请求预留资源量的时隙，允许用最小资源量进行预留^[14]，然而这些分片方式并不能适用到区块链的系统中，无法提升区块链的效率和性能。区块链领域有双链设计的应用，如通过对账户区块链和交易区块链分开的处理方法^[15]，但是没有从系统级别来提升区块链的效率和性能。本文结合区块链系统的实际特点，并在考虑区块链的效率和性能提升的基础上，设计和实现了多区块链的交易分发及事件处理的系统，还在超级帐本的基础上进行了验证。

2 系统运行环境

多区块链交易分发和事件处理系统的运行场景的建立包括如下步骤:1)系统管理员在系统的每个节点上安装了区块链系统的镜像，并为每个节点安装和部署智能合约(智能合约是一种应用级代码，指区块链的应用领域所部署的接口，使用者可以使用接口参数调用接口)，智能合约部署后作为系统每个节点区块链镜像文件的扩展镜像而存在。2)区块链为智能合约提供了各类外部接口，如初始化、交易的调用、交易结果的查询、区块链状态和相关信息的查询等，区块链的使用者可以通过应用客户端调用这些外部接口。在进行交易调用时，传入所调用的智能合约的名称、智能合约的路径、智能合约的方法名称、智能合约的方法参数、智能合约的名称和智能合约的路径来在系统镜像中确认和定位智能合约，智能合约的方法名称和方法参数则是进行交易的调用。3)系统的外部用户下载系统证书，通过约定的加密算法对关键字段(比如用户密码等)进行加密，把报文及报文加签后的签名发送给系统；系统安全认证通过后即可通过应用客户端使用系统进行交易。区块链的使用者可以通过应用客户端调用这些外部接口，而外部接口的使用是否成功，则分为同步等待和异步通知的方式。同步等待方式指应用客户端一直等待区块链完成执行；异步通知指应用客户端发出请求后，不等待区块链的执行完成，区块链完成执行后通知应用客户端执行结果。区块链上的事件指区块链的各类交易完成时触发的事件以及区块的数据信息和区块状态的迁移、应用客户端的注册和注销、系统拒绝等各类事件，各类事件通过消息流的方式以应用客户端的形式与区块链之间进行消息通信。目前，智能合约都部署运行在一个区块链上，区块的处理速度太慢，并且随着区块数量的增加，链接的区块数量增多，区块链上处理交易的速度变

慢，在单个区块链的内部性能达到优化的瓶颈后，采用基于交易分发和事件处理的系统来提高区块链的性能则是区块链系统设计的一种全新的方式。

3 多区块链系统

单一的区块链所能达到的效率和实现的系统性能是有限的，对于区块链上每个机构组织而言，所能提供给区块链的节点数是一个或多个，一个机构组织可以分配具有相同的计算功能和交易功能的节点到多个相同类型的区块链中。多区块链系统按业务类型分成多种业务类型的区块链，每种业务类型的区块链都包含一个或多个区块链；多区块链系统中相同业务类型如果包含多个区块链，那么每个区块链的共识结果都是被认可和接受的，一个机构组织可以分配相同的多个节点到多个区块链中。多区块链系统中每个区块链包含区块链业务类型和区块链编号的基本属性。多区块链系统中，区块链的使用者通过应用客户端来提交交易的请求，同时通过应用客户端与处理系统建立事件连接和处理事件流消息。

4 多区块链的交易分发和事件处理系统

多区块链的交易分发和事件处理的系统示意如图 1 所示。系统包含应用客户端、处理系统和多区块链。系统中多个区块链启动后，系统管理员安装和部署智能合约，然后处理系统启动并建立与每个区块链的网络连接，最后是应用客户端的启动。

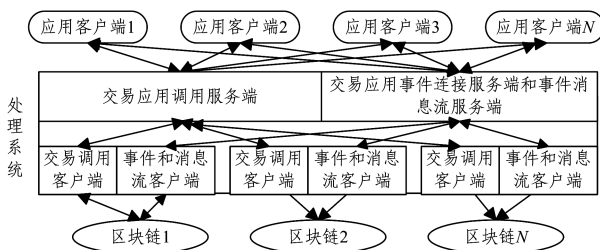


图 1 多区块链的交易分发和事件处理系统

多区块链系统的用户通过应用客户端来发起交易。多区块链系统提供多个应用客户端，应用客户端是可扩展的，可以根据应用需要动态增加或减少，应用客户端的数量与多区块链系统中区块链的数量没有关联。多区块链系统提供配置文件、初始化文件和数据库表的方式来设置多区块链中每个区块链的属性，多区块链系统的系统管理员可以设置任意一种方式来使用。处理系统衔接多个应用客户端和多个区块链，处理系统对于每个区块链提供了客户端的功能，对于每个应用客户端则作为其服务端的功能而存在。处理系统在多区块链中的每个区块链启动后，以每个区块链为服务端，基于每个区块链都建立客户端，包括交易调用客户端、事件和消息流客户端，这样可避免有应用客户端执行交易和事件消息处理时才建立客户端而引起的额外网络连接负载。同时处理系统也向每个应用客户端提供统一的服务端，包括交易应用调用服务端、交易应用事件连接服务端和事件消息流服务端。其中交易应用事件连接服务端指应用客户端向处理系统进行事件连接时，处理系统提供事件连接服务，并通过事件流服务端来分配事件消息流的通信管道地址。系统用户在使用应用客户端发送交易请求时，会增加两个参数，包括业务类型和业务类型内部的 ID 号；处理系统通过智能合约的调用接口中提供的

附加的业务类型和业务类型内部的 ID 号,基于处理系统的交易分发处理策略,选择合适的交易调用客户端并分发到合适的区块链,同时事件消息流服务端也会基于交易分发处理策略选择合适的交易调用客户端和事件消息流客户端;区块链在响应事件消息时,会把事件消息流发送给处理系统中合适的事件和消息流客户端。同一个应用领域存在一个或多个区块链,这里的业务类型指区块链的应用领域类型,如资产交易、股权众筹、智能合约、证券交易、电子商务等;业务编码指应用领域内部的细分,它们是相同业务领域的内部划分。应用客户端在交易时调用带参数的执行方法(`invoke`)、带参数的查询方法(`query`)等,这里的交易包括交易的执行和交易的查询等。

4.1 多区块链中选择区块链的策略

多区块链的总的属性定义为数量可变区块链和数量不可变的区块链。区块链的选择是指接收到应用客户端的交易请求后,选择哪个区块链执行交易请求,处理系统为保证系统的处理的一致性,对区块链进行了统一编号。多区块链系统选择哪个区块链来进行交易的策略设计如下。

基于区块链的数量是否可变采用不同的方法。如图 2 所示,系统在启动时会读取区块链总的属性并读取各区块链相关的配置文件或数据库表。区块链总的属性指区块链总的数量是否可变,各区块链配置信息包括区块链的类型、区块链相同类型的数量、相同区块链的每个区块链的编号、区块链接收交易报文的 IP 地址和端口、区块链接收事件连接的 IP 地址和端口;其中接收交易报文的 IP 地址和接收消息事件的 IP 地址是相同的,但是端口不同。如果区块链总的数量不变,则对区块链进行分类统计后,初始化区块链数据信息。无论应用客户端的数量是否变化,处理系统与区块链的连接都基于业务类型和业务类型内部的编号进行哈希并保持负载均衡性、单调性和一致性,即区块链被均匀连接使用及同一个应用客户端的请求都是传送到相同编号的区块链,同时新的请求也是基于负载均衡分配到未使用的区块链。如果区块链总的数量可变,则初始化区块链映射表。映射表是指在应用客户端输入的参数中选择业务类型,业务类型内部的 ID 号映射到相同业务类型的具有某一个编号的区块链上;映射表保存在磁盘存储器里并建立内存缓存,以便于映射表的更新保存和读取操作。在区块链的相关数据信息获取后,处理系统与每个区块链进行通信连接,包括交易调用的连接和事件消息流的连接,建立每个区块链对应的处理系统的客户端,客户端包括交易调用客户端、事件和消息流客户端。多区块链中选择区块链的方案策略以确保应用客户端的数量和区块链的数量改变时不会引起可用的区块链之间的数据迁移,即不会导致一个区块链的计算的结果数据迁移到另外一个可用的区块链上。对于区块链的总数增加的场景,基于单调性的特性,新的交易请求会分发给新的区块链去执行;对于区块链的总数减少的场景,被减少的区块链会从映射表中移除,不再作为交易分发的选择,交易请求会基于负载均衡和单调性的原则进行分发,也会基于当前可用的区块链保持一致性。处理系统对每个区块链的网络连接进行了监控,当区块链网络连接中断时,不再作为可用的区块链,而区块链网络恢复时,则再次作为可用的区块链。处理系统自身通过集群模式消除了单点故障。

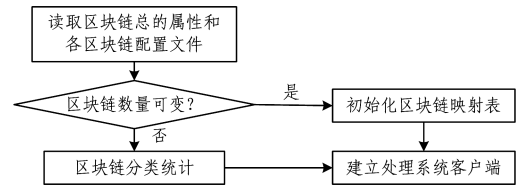


图 2 多区块链中选择区块链的策略

4.2 多区块链的分发方案

处理系统灵活地分发智能合约,选择合适的区块链进行事件连接和事件消息流的处理,即根据应用客户端参数灵活动态地分配,没有静态地分配和指定。多区块链进行分发时,通过业务类型和业务类型内部的 ID 号进行区分。不同的业务通过业务类型来区分,相同业务类型通过业务类型内部的 ID 号来区分业务,业务类型和业务 ID 作为处理系统进行分发时新增的参数,在参数传递到处理系统时,处理系统首先解析业务类型,然后在对应业务类型中的区块链集合中,基于负载均衡性、单调性和一致性的原则,结合业务类型内部的 ID 号来计算出合适的区块链编号所对应的处理系统内部的客户端。处理系统在进行分发时,考虑两个维度的分发,包括交易应用调用的分发和事件连接及事件数据流的分发。多区块链的交易调用分发系统包括交易应用调用服务器和交易调用客户端。交易应用调用服务端用来监听应用客户端的交易请求,交易调用客户端用来与每个区块链建立一对一的通信连接。多区块链的事件连接及事件数据流的分发处理的原则和交易应用调用的分发处理的原则一样,基于分发原则和方式选择合适的区块链编号对应的处理系统内部的事件和消息流客户端。在应用客户端发送交易请求时,用交易的方法函数和参数、交易的执行方法函数(`invoke`)来执行交易,交易的查询方法函数(`query`)则可以用来查询区块链上的资产情况以及区块链的状态信息等。以两种简化的业务类型(包括简化的资产交易和简化的股权众筹)为例子进行分发的系统方案的设计说明。简化的资产交易示例流程:用户在交易前部署的资产拥有人 A 拥有资产数量 1000000;资产拥有人 B 拥有资产数量 100;调用函数(`invoke`)执行交易,交易的参数包括可以用来设定资产转移的方向即从哪个资产拥有人转给哪个资产拥有人、转移的资产的数量等,例如资产拥有人 A 转给资产拥有人 B,转移的资产数量是 100;查询函数(`query`)用来查询交易资产的余额以及资产的其他数据信息,如果调用函数(`invoke`)执行相同交易 100 次,则资产拥有人 A 转给资产拥有人 B 的总的数量是 10000,那么资产拥有人 A 剩余的资产是 1000000 减去 10000,即通过查询函数(`query`)查询到资产拥有人 A 的资产是 990000,查询到资产拥有人 B 的资产是 10100。简化的股权众筹交易示例流程:用户通过应用客户端在交易前部署股权投资人 A 的信息数据(股权投资人 A 的唯一标识符 ID、股权投资人 A 的份额百分比、股权投资人 A 的份额、股权投资人 A 的投资金额),股权投资人 B 的信息数据(股权投资人 B 的唯一标识符 ID、股权投资人 B 的份额百分比、股权投资人 B 的份额、股权投资人 B 的投资金额),股权投资人 C 的信息数据(股权投资人 C 的唯一标识符 ID、股权投资人 C 的份额百分比、股权投资人 C 的份额、股权投资人 C 的投资金额)。调用函数(`invoke`)执行股权交易,交易的参数可以用来设定股权转移的方向即从哪个股权投资人转给哪个股权投资人、转移的份额的数量;查询函数(`query`)可以用来查询资产拥有人的最新的股权信息,参数可以是股权投资人

的唯一标识符。应用客户端在交易资产时,调用函数(invoke)和查询函数(query)都需要增加额外的参数,包括交易的业务类型和交易的业务类型内部的 ID 号。交易类型为资产交易类型、股权众筹类型等,交易业务类型内部的 ID 号可以根据实际业务需要任意选择,比如资产交易可以用资产交易的数量范围确定业务类型内部的 ID 号;股权众筹的业务类型内部的 ID 号可以用股份的名称 ID 作为业务类型内部的 ID 号。这样在已有参数的基础上,通过扩展的参数来选择交易调用客户端、事件和消息流客户端,并基于负载均衡性和一致性的原则保证了区块链数据处理的一致性。若业务类型相同并且业务 ID 号相同,那么总是在同一个可用的区块链上进行交易和事件处理。

4.3 多区块链的事件流的处理系统

区块链上的事件指区块链的各类交易完成时触发的事件以及包括区块的数据信息和区块状态的变化事件,应用客户端的注册、注销和系统拒绝等各类事件,各类事件在事件连接后以消息流的方式在应用客户端、处理系统和区块链之间进行传递。在多区块链的环境中,多个应用客户端对多个区块链发起智能合约的交易请求,同时应用客户端注册感兴趣的事件,应用客户端和区块链的事件消息流是双向的读操作和写操作。其中交易请求、事件连接和事件消息流是处理系统和同一个可用的区块链处理系统通信,同时处理系统和同一个可用的应用客户端之间通信。事件消息流服务端,用于接收所述应用客户端发送的事件消息流获取请求,将所述事件消息流获取请求发送至所述交易请求所分配的区块链。事件和消息流客户端,用于当所属区块链执行智能合约的交易时,获取对应的事件消息流,将所述事件消息流通过事件连接时建立的事件流服务端返回至应用客户端。事件包括应用客户端与交易应用调用服务端通信时触发的事件、应用客户端与事件消息流服务端通信时触发的事件,以及区块链上部署的智能合约在执行交易时所触发的事件。例如,注册、注销和拒绝等以及交易状态和区块链状况等。传统的方式中,区块链上部署的智能合约可以通过同步或异步的方式向服务器返回注册、注销和拒绝等通信事件的信息。交易状态等用户感兴趣事件的消息,用户也可以通过应用客户端主动查询得到。本系统方案实施中,多个应用客户端与多个区块链之间通过事件消息流的方式进行通信,并以异步的方式返回事件消息。在事件消息流服务端与应用客户端建立事件连接之后,应用客户端可以向事件消息流服务端注册感兴趣的事件,例如,交易状态和区块链状态等。应用客户端向交易应用调用服务端发送交易请求时,还可以向事件消息流服务端发送事件消息流获取请求,交易应用调用服务端根据交易请求中携带的业务类型和业务类型内部的业务 ID 号选择相应的区块链。处理系统内部的交易调用客户端将应用客户端的交易请求分配至已选择的区块链。事件消息流服务端将应用客户端发送的事件消息流的获取请求转发至事件和消息流客户端,事件消息流客户端将事件消息流获取请求发送至处理交易的区块链。也就是说,交易请求与事件消息流的获取请求被分配至同一个区块链。当区块链执行智能合约的交易时,事件流客户端采用异步的方式获取事件消息流。事件和消息流客户端将获取到的事件消息流返回至事件消息流服务端,事件消息流服务端将事件消息流发送至对应的应用客户端。由此不需再通过应用客户端对感兴趣的事件进行查询,可以直接获取

用户感兴趣事件的消息,为用户提供了方便。多个应用客户端与多个区块链之间可以通过双向事件消息流的方式进行通信。事件消息流服务端可以并发读取多个应用客户端的事件消息流获取请求,并且通过事件和消息流客户端将读取到的事件消息流的获取请求发送至对应的区块链。事件和消息流客户端也可以通过并发的形式读取区块链的事件消息流,通过事件消息流服务端将事件消息流并发送送至应用客户端。通过并发的形式对应用客户端与区块链之间的事件消息流进行读写操作,能够有效提高应用客户端与区块链之间的通信效率,并发包括进程并发处理、线程并发处理以及协程的并发处理等综合的模式,对其进行了设计与实现。

5 系统测试和验证

系统基于超级账本 Hyperledger Fabric 1.0.0 的区块链进行了系统测试和验证,结果如表 1 所列,建立了区块链数量可变和不可变的两类多区块链系统,在区块链数量可变的前提下基于业务类型和业务类型内部的 ID 号建立映射关系,在区块链数量不可变的前提下基于业务类型和业务类型内部的 ID 号进行一致性哈希算法。两类多区块链在处理系统内部都实现了负载均衡性、单调性、一致性等特性。每个区块链和处理系统内部的交易调用客户端、事件和消息流客户端进行连接;应用客户端和处理系统内部的交易应用调用服务端、交易应用事件连接服务端、事件消息流服务端进行连接。通过应用客户端发送高并发和大数据量的交易请求。

表 1 系统测试和验证结果

类型	TPS
单区块链	1000
静态多区块链(链的数量为 5)	4200
多态多区块链(最大的链的数量为 5)	4500

测试结果为:在单一区块链上的性能是 1000TPS,在数量为 5 的多区块链上的性能是 4200TPS,在区块链数量可变最大数量为 5 的区块链上的性能是 4500TPS,在交易请求的同时也实现了负载均衡性、单调性和一致性,处理系统内部的事件处理正常工作,能接收到对交易请求响应的事件消息流。测试的结果符合预期,系统方案大幅提升了区块链的性能。

结束语 本方案的重点在于对多区块链的交易分发和事件处理的系统方案的设计与开发。系统方案考虑区块链数量可变和不可变的两类多区块链系统,并进行了不同的设计与实现;两类系统都保证了负载均衡性、单调性和一致性;同时支持高并发和大数量的交易请求。交易分发机制保证了系统性能的大幅提升,事件处理机制保证了交易状态和区块链状况的异步通知,并保证了多区块链上的事件的正确处理,同时异步通知方式也保证了不会出现同步阻塞引起的系统性能问题。未来进一步的工作可以适配以太坊等其他类型的区块链系统。

参考文献

- [1] 袁勇,王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016,42(4):482.
- [2] 董宁,朱轩彤. 区块链技术演进及产业应用展望[J]. 信息安全研究, 2017,3(3):200-210.
- [3] 王继业,高灵超,董爱强,等. 基于区块链的数据安全共享网络体系研究[J]. 计算机研究与发展, 2017,54(4):742-749.

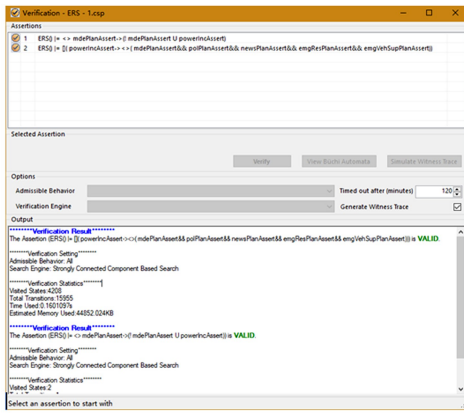


图5 结果界面

图5给出了两条行为约束的验证结果均为“Valid”。由此可以确定该电力突发公共事件应急处置系统中的消息执行与预期一致,说明该应急处置系统的设计是合理的。

结束语 针对协同业务过程的建模和分析,首先,本文使用有线状态自动机建模参与组织的业务过程,通过“集中式”异步通信模型,将业务过程组合为协同业务过程;其次,提出行为约束的声明式模板,用于定义协同业务过程需要满足的行为约束规约;最后,使用模型检测技术,借助PAT工具,实现了对协同业务过程行为的自动验证。

本文的工作关注协同业务过程的建模和行为验证,未来的工作将重点考虑行为验证面临的状态爆炸问题。

参考文献

- 代飞,莫启,林雷蕾,等.结合Petri网和Pi演算的协同业务过程建模[J].计算机科学与探索,2015,9(6):692-706.
- 卢亚辉,明仲,张力.业务过程协同模式的研究[J].计算机集成制造系统,2011,17(8):1570-1579.
- YU W Y, YAN C G, DING Z J, et al. Modeling and verification of online shopping business processes by considering malicious behavior patterns[J]. IEEE Transactions on Automation Science and Engineering, 2016, 13(2): 647-662.
- SHAHRIARI K, HESSAMI A G, JADIDI A, et al. An approach toward a conceptual collaborative framework based on a case study in a wood supply chain[J]. IEEE Systems Journal, 2015, 9(4): 1-10.
- 曾庆田,鲁法明,刘聪,等.基于Petri网的跨组织应急联动处置系统建模与分析[J].计算机学报,2013,36(11):2290-2302.
- 何蒲,于戈,张岩峰,等.区块链技术与应用前瞻综述[J].计算机科学,2017,44(4):6.
- 许涛.区块链技术在教育教学中的应用与挑战[J].现代教育技术,2017,27(1):110-111.
- 安瑞,何德彪,张韵茹,等.基于区块链技术的防伪系统的设计与实现[J].密码学报,2017,4(2):199-208.
- 田海博,何杰杰,付利青.基于公开区块链的隐私保护公平合同签署协议[J].密码学报,2017,4(2):187-198.
- 夏新岳.基于区块链的股权资产购买和转赠设计与实现[D].内蒙古:内蒙古大学,2016:29-36.
- 黄洁华,高灵超,许玉壮,等.众筹区块链上的智能合约设计[J].信息安全研究,2017,3(3):211-219.
- 张波.国外区块链技术的运用情况及相关启示[J].金融科技时,2016(5):35.
- 黄征,李祥学,来学嘉,等.区块链技术及其应用[J].信息安全研究,2017,3(3):237-245.
- 冯超政,蒋溢,何军,等.基于冷热数据的MongoDB自动分片机制[J].计算机工程,2017,43(3):7-10.
- 王亚玲,杨超,章名尚.数据库系统应用分片中间件[J].计算机系统应用,2015,24(10):76-78.
- 吴黎兵,党平,聂雷,等.一种可分片预留接纳控制算法研究[J].计算机研究与发展,2014,51(6):1201-1204.
- 蔡维德,郁莲,玉荣,等.基于区块链的应用系统开发方法研究[J].软件学报,2017,28(6):1474-1487.
- KESTEN Y, PNUELI A, RAVIV L O. Algorithmic verification of linear temporal logic specifications[J]. Lecture Notes in Computer Science, 1999, 1443(1443): 1-16.
- CS Department NUS. PAT: Process Analysis Toolkit [EB/OL]. [2013-09-13]. <http://www.patroot.com>.
- AALST W. Modeling and analyzing interorganizational workflows[C]//Proc of the 1st IntConf on Application of Concurrency to System Design. Los Alamitos, CA: IEEE Computer Society, 1998: 262-272.
- ZHANG L, LU Y, XU F. Unified modelling and analysis of collaboration business process based on Petri nets and Pi calculus[J]. IET Software, 2010, 4(5): 303-317.
- ZENG Q T, LU F M, LIU C, et al. Modeling and verification for cross-department collaborative business processes using extended Petri nets[J]. IEEE Trans on Systems, Man, and Cybernetics: Systems, 2015, 45(2): 349-362.
- 葛季栋,胡海洋,周宇,等.一种基于不变量的工作流协同模型分解方法[J].计算机学报,2012,35(10):2169-2181.
- 邓水光,李莹,吴健,等. Web 服务行为兼容性的判定与计算[J].软件学报,2007,18(12):3001-3014.
- AALST W, PESIC M, SCHONENBERG H. Declarative workflows: Balancing between flexibility and support[J]. Computer Science-Research and Development, 2009, 23(2): 99-113.
- MONTALI M. Specification and verification of declarative open interaction models-A logic-based approach[J]. Springer Science & Business Media, 2010, 56(1): 47-76.
- HIDEBRANDT T, MUKKAMALA R. Declarative event-based workflow as distributed dynamic condition response graphs[C]//Electronic Proceedings in Theoretical Computer Science (EPTCS) 69: Proc of PLACES 2010. Sydney, Australia: EPTCS, 2011: 59-73.
- AWAD A, WEIDLICH M, WESKE M. Visually specifying compliance rules and explaining their violations for business process[J]. Journal of Visual Languages & Computing, 2011, 22(1): 30-55.
- BAIER C, KATOEN J P. Principles of model checking[M]. Cambridge: MIT Press, 2008.
- SUN J, LIU Y, DONG J S. Model Checking CSP Revisited: Introducing a Process Analysis Toolkit[M]//Leveraging Applications of Formal Methods, Verification and Validation. Springer Berlin Heidelberg, 2008: 307-322.

(上接第583页)