滴雨科技区块链技术指南

介绍篇

介绍

一般而言，区块链是一个不变的交易分类帐，维护在*对等节点*的分布式网络中。这些节点各自通过应用已通过*共识协议*验证的交易来维护分类账的副本，交易被分组为多个区块，这些区块包括将每个区块绑定到前一个区块的哈希。

区块链的第一个也是最广为人知的应用是 比特币加密货币，尽管其他应用也紧随其后。以太坊是另一种加密货币，采用了不同的方法，整合了许多与比特币相同的特征，但增加了*智能合约*以创建用于分布式应用程序的平台。比特币和以太坊属于同一类的区块链，我们将其都归类为 *公共无许可*区块链技术。基本上，这些是对任何人开放的公共网络，参与者可以在其中进行匿名交互。

随着比特币，以太坊和其他一些衍生技术的普及，对将区块链的基础技术，分布式分类帐和分布式应用程序平台应用于更具创新性的*企业*用例的兴趣 也日益增长。但是，许多企业用例需要性能特征，即无权限的区块链技术无法（目前）提供这些性能。另外，在许多用例中，参与者的身份是一个硬性要求，例如在金融交易中，必须遵循“了解您的客户（KYC）”和“反洗钱（AML）”法规。

对于企业使用，我们需要考虑以下要求：

- 参与者必须是可识别/可识别的
- 网络需要获得*许可*
- 高交易吞吐量性能
- 交易确认的低延迟
- 与业务交易有关的交易和数据的隐私权和机密性

尽管许多早期的区块链平台目前都已进行一些改进，从而为企业使用。但 Hyperledger Fabric 从一开始就为企业使用而设计。以下各节描述了 Hyperledger Fabric（Fabric）如何与其他区块链平台区分开来，并描述了其架构决策的一些动机。

超级账本 fabric

Hyperledger Fabric 是一个开放源代码的企业级许可分布式分类帐技术（DLT）平台，设计用于企业环境，与其他流行的分布式分类帐或区块链平台相比，它提供了一些关键的区分功能。

差异化的一个关键点是 Hyperledger 是在 Linux 基金会下建立的，它本身有着悠久而非常成功的历史，它在**开放式治理**下培育开源项目，这些项目发展了强大的可持续社区和繁荣的生态系统。Hyperledger 由多元化的技术指导委员会管理，Hyperledger Fabric 项目由来自多个组织的多元化维护人员管理。自最早成立以来，它的开发社区已发展到超过 35 个组织和近 200 个开发人员。

Fabric 具有高度**模块化**和可**配置的**体系结构，可针对银行，金融，保险，医疗保健，人力资源，供应链甚至数字音乐交付等广泛的行业用例进行创新，多功能和优化。

Fabric 是第一个分布式账本平台，支持**以通用编程语言（**例如 Java，Go 和 Node.js）而非受约束的领域特定语言（DSL）**编写的智能合约。**这意味着大多数企业已经具有开发智能合约所需的技能，并且不需要其他培训来学习新的语言或 DSL。

Fabric 平台也是**经过许可的**，这意味着与公共的未经许可的网络不同，参与者是彼此认识的，而不是匿名的，因此是完全不受信任的。这意味着，尽管参与者之间可能不会*完全*信任对方（例如，他们可能是同一行业的竞争者），但网络可以在治理模型下运行，该治理模型是基于参与者之间确实存在的信任而建立的，例如处理争议的法律协议或框架。

平台最重要的区别之一是它对**可插入共识协议的**支持，该 **协议**使平台可以更有效地进行定制，以适应特定的用例和信任模型。例如，当部署在单个企业中或由受信任的机构运营时，完全拜占庭式的容错共识可能被认为是不必

要的，并且会严重拖累性能和吞吐量。在这种情况下， 崩溃容错（CFT）共识协议可能绰绰有余，而在多方，分散的用例中， 可能需要更传统的 拜占庭容错（BFT）共识协议。

Fabric 可以利用**不需要本机加密货币的**共识协议来激发昂贵的挖掘或推动智能合约的执行。避免使用加密货币会降低一些重大的风险/攻击向量，并且无需进行加密挖掘操作就意味着可以以与任何其他分布式系统大致相同的运营成本来部署该平台。

这些差异化设计功能的结合使 Fabric 在交易处理和交易确认延迟方面成为当今**性能最好的平台**之一，并实现了交易和实现的智能合约（Fabric 称为“链码”）的**隐私和机密性。**他们。让我们更详细地探讨这些差异化功能。

模块化

Hyperledger Fabric 专门设计为具有模块化体系结构。无论是可插拔共识，可插拔身份管理协议（例如 LDAP 或 OpenID Connect），密钥管理协议还是密码库，该平台的核心都经过了配置，可以满足企业用例需求的多样性。在较高的层次上，Fabric 由以下模块化组件组成：

- 可插拔*排序服务 ordering service* 在交易顺序上达成共识，然后将区块广播给同级。
- 可插拔*成员资格服务 MSP 提供商*负责将网络中的实体与加密身份相关联。
- 可选的*对等 gossip 服务*通过向其他对等点订购服务来分发输出的块。
- 智能合约（“链码”）在容器环境（例如 Docker）中运行以进行隔离。它们可以用标准编程语言编写，但不能直接访问分类帐状态。
- 账本可以配置为支持各种 DBMS。
- 可插拔的认可和验证策略实施，可以针对每个应用程序进行独立配置。

业界普遍认为，没有“一个区块链可以全部统治”。可以通过多种方式配置 Hyperledger Fabric，以满足多种行业用例的不同解决方案要求。

许可与无许可的区块链

在未经许可的区块链中，几乎任何人都可以参与，每个参与者都是匿名的。在这种情况下，除了一定深度之前的区块链状态是不可变的之外，别无其他信任。为了减轻这种信任的缺乏，无许可的区块链通常采用“开采的”本机加密货币或交易费来提供经济激励，以抵消基于“工作证明”（拜占庭式）的形式的拜占庭容错共识的参与所产生的特殊成本。

另一方面，获得许可的区块链在一组已知，已识别且经常经过审核的参与者中运作一个区块链，该参与者在产生一定程度信任的治理模型下运行。许可的区块链提供了一种方法来保护一组具有共同目标但可能不会完全相互信任的实体之间的交互。通过依赖参与者的身份，许可的区块链可以使用更传统的崩溃容错（CFT）或拜占庭容错（BFT）共识协议，这些协议不需要昂贵的挖掘。

另外，在这种许可的情况下，参与者通过智能合约有意引入恶意代码的风险得以降低。首先，参与者是相互了解的，并且遵循针对网络和相关交易类型建立的认可政策，所有活动（无论是提交应用程序交易，修改网络配置还是部署智能合约）都记录在区块链上。除了完全匿名之外，还可以根据治理模型的条款轻松地确定有罪的一方并处理事件。

智能合约

智能合约或 Fabric 称之为“链码”的功能，是一种受信任的分布式应用程序，可从区块链和对等方之间的潜在共识中获得安全性/信任。这是区块链应用程序的业务逻辑。

适用于智能合约的三个要点，尤其是应用于平台时：

- 网络中同时运行许多智能合约，
- 它们可以动态部署（在许多情况下，任何人都可以），并且
- 应用程序代码应被视为不受信任，甚至可能是恶意的。

现有的大多数具有智能合约功能的区块链平台都遵循一种 排序执行架构，其中共识协议为：

- 验证并订购交易，然后将其传播到所有对等节点，
- 然后，每个对等方依次执行事务。

排序执行架构几乎可以在所有现有的区块链系统中找到，从以太坊等公共/非许可平台 （基于 PoW 的共识）到 Tendermint， Chain 和 Quorum 等许可平台。

在以排序执行架构运行的区块链中执行的智能合约必须具有确定性。否则，可能永远无法达成共识。为了解决非确定性问题，许多平台要求以非标准的或领域特定的语言（例如 Solidity）编写智能合约，以便消除非确定性操作。这阻碍了广泛采用，因为它要求开发人员编写智能合约来学习一种新语言，并可能导致编程错误。

此外，由于所有事务由所有节点顺序执行，因此性能和规模受到限制。智能合约代码在系统中的每个节点上执行的事实要求采取复杂的措施来保护整个系统免受潜在的恶意合约的侵害，以确保整个系统的弹性。

一种新方法

Fabric 为事务引入了一种新的架构，我们称之为 **execute-order-validate**。它通过将事务流分为三个步骤来解决排序执行模型面临的弹性，灵活性，可伸缩性，性能和机密性挑战：

- *执行*交易并检查其正确性，从而认可该交易，
- 通过（可插入）共识协议*订购*交易，以及
- *在将*交易提交到分类账之前，根据特定于应用程序的背书策略*验证*交易

这种设计与订单执行范式完全不同，Fabric 在达成交易的最终协议之前执行交易。

在 Fabric 中，特定于应用程序的背书策略指定需要哪些对等节点或其中的对等节点保证给定智能合约的正确执行。因此，每个交易仅需要由满足交易认可策略所必需的对等节点的子集执行（认可）。这允许并行执行，从而提高了系统的整体性能和规模。该第一阶段还消除了任何不确定性，因为不一致的结果可以在订购前滤除。

因为我们已经消除了不确定性，所以 Fabric 是第一个启用使用标准编程语言的区块链技术。在 1.1.0 版本中，可以使用 Go 或 Node.js 编写智能合约，而在后续版本中还计划支持其他流行语言，包括 Java。

私隐与保密

正如我们已经讨论的那样，在一个利用 PoW 作为共识模型的公共，无许可的区块链网络中，交易在每个节点上执行。这意味着合同本身或所处理的交易数据都不会保密。每个事务及其实现的代码对于网络中的每个节点都是可见的。在这种情况下，我们已经将合同和数据的机密性换成了 PoW 交付的拜占庭容错共识。

对于许多业务/企业用例而言，缺乏机密性可能会成为问题。例如，在供应链合作伙伴网络中，可能会为某些消费者提供优惠价格，以巩固关系或促进额外销售。如果每个参与者都能看到每份合同和交易，那么就不可能在完全透明的网络中维持这种业务关系–每个人都希望获得优惠的价格！

再举一个例子，考虑证券行业，在该行业中，建立仓位（或出售仓位）的交易者不希望竞争对手知道这一点，否则他们将寻求介入游戏，削弱了交易者的竞争能力。

为了解决出于满足企业用例需求的目的而缺乏隐私和机密性的问题，区块链平台采用了多种方法。所有人都有其取舍。

加密数据是提供机密性的一种方法。但是，在利用 PoW 达成共识的无许可网络中，加密数据位于每个节点上。如果有足够的时间和计算资源，则可能会破坏加密。对于许多企业用例而言，其信息可能遭到破坏的风险是无法接受的。

零知识证明（ZKP）是为解决此问题而正在探索的另一个研究领域，这里的权衡是，目前计算 ZKP 需要大量时间和计算资源。因此，在这种情况下的权衡是为了保密。

在可以利用替代形式的共识的许可上下文中，人们可能会探索一些将机密信息仅分配给授权节点的方法。

Hyperledger Fabric 是允许的平台，可通过其通道架构实现机密性。基本上，Fabric 网络上的参与者可以在参与者的子集之间建立一个“通道”，该通道应被授予特定交易集的可见性。将此视为网络覆盖。因此，只有那些参与频道的节点才能访问智能合约（链码）和交易的数据，从而保留了两者的隐私和机密性。

为了改善其隐私和机密性功能，Fabric 增加了对私有数据的支持，并在未来开发可用的零知识证明（ZKP）。

随着它的可用，将对此进行更多介绍。

可插拔共识

事务的顺序被委托给模块化组件以实现共识，该组件在逻辑上与执行事务并维护分类帐的对等方分离。具体来说就是订购服务。由于共识是模块化的，因此可以根据特定部署或解决方案的信任假设量身定制其实现。这种模块化体系结构允许平台依赖完善的工具包来进行 CFT（崩溃容错）或 BFT（拜占庭容错）排序。

Fabric 当前提供两种 CFT 订购服务实现。第一是基于 etcd 库的的 Raft 协议。另一个是 Kafka（内部使用 Zookeeper）。有关当前可用排序服务的信息，请查看有关订购的概念性文档。

还要注意，它们不是互斥的。Fabric 网络可以具有支持不同应用程序或应用程序需求的多种排序服务。

性能和可伸缩性

区块链平台的性能可能会受到许多变量的影响，例如交易规模，区块规模，网络规模以及硬件限制等。Hyperledger 社区目前正在性能和规模工作组内制定一套措施草案。，以及称为 Hyperledger Caliper 的基准测试框架的相应实现 。

尽管这项工作仍在继续发展，应被视为衡量区块链平台性能和规模特征的权威，但 IBM Research 的一个团队发表了一篇 同行评审论文，评估了 Hyperledger Fabric 的体系结构和性能。本文提供了关于 Fabric 架构的深入讨论，然后使用 Hyperledger Fabric v1.1 的预发行版报告了团队对该平台的性能评估。

研究团队所做的基准测试工作为 Fabric v1.1.0 发行版带来了许多性能改进，使平台的整体性能比 v1.0.0 发行版提高了一倍以上。

结论

对区块链平台的任何认真评估都应在其短名单中包括 Hyperledger Fabric。

结合起来，Fabric 的差异化功能使其成为用于许可区块链的高度可扩展系统，支持灵活的信任假设，使该平台能够支持从政府，金融，供应链物流，医疗保健等广泛的行业用例。多得多。

Hyperledger Fabric 是 Hyperledger 项目中最活跃的项目。该平台周围的社区正在稳步增长，并且每个后续版本提供的创新远远超过了其他任何企业区块链平台。

v2.0 Alpha 中的新增功能

关于 Alpha 版本

Hyperledger Fabric v2.0 的 Alpha 版本允许用户试用两个令人兴奋的新功能-新的 Fabric 链码生命周期和 FabToken。提供 Alpha 版本是为了向用户提供新功能的预览，而不是用于生产中。此外，没有对 v2.0 Alpha 发行版的升级支持，也没有从 Alpha 发行版到 v2.x 未来版本的预期升级支持。

Fabric 链代码生命周期

Fabric 2.0 Alpha 引入了针对链码的分散式治理，并引入了一种新过程，可在您的对等节点上安装链码并在通道上启动它。新的 Fabric 链码生命周期允许多个组织在可用于与分类账交互之前就链码的参数达成一致，例如链码认可策略。新模型在以前的生命周期中提供了一些改进：

- **多个组织必须同意链码的参数：**在 Fabric 的 1.x 版本中，一个组织可以为所有其他渠道成员设置链码的参数（例如，认可策略）。新的 Fabric 链码生命周期更加灵活，因为它既支持集中式信任模型（例如先前生命周期模型的模型），也支持分散模型，这些模型需要足够数量的组织才能在背书策略上生效。
- **更安全的链码升级过程：**在先前的链码生命周期中，升级事务可能由单个组织发出，这给尚未安装新链码的渠道成员带来了风险。新模型仅在足够数量的组织批准升级后才允许升级链码。
- **更容易的背书策略更新：** Fabric 生命周期允许您更改背书策略，而无需重新打包或重新安装链码。用户还可以利用新的默认策略，该策略需要频道中大多数成员的认可。在从渠道中添加或删除组织时，该策略会自动更新。
- **可检查的链代码包：** Fabric 生命周期将链代码打包在易于阅读的 tar 文件中。这使得检查链码包和协调跨多个组织的安装变得更加容易。
- **使用一个程序包在通道上启动多个链码：上一个**生命周期使用安装链码包时指定的名称和版本定义了通道上的每个链码。现在，您可以使用单个 chaincode 程序包，并在相同或不同的通道上以不同的名称多次部署它。

使用新的 Chaincode 生命周期

使用以下教程开始新的链码生命周期：

- 运营商的链码：提供安装和定义链码所需步骤的详细概述，以及新模型可用的功能。
- 建立您的第一个网络：如果您想立即开始使用新的生命周期，则 BYFN 教程已更新，可以使用对等生命周期链代码 CLI 在示例网络上安装和定义链代码。
- 在 Fabric 中使用私有数据：已更新，以演示如何 在新的链码生命周期中使用私有数据集合。
- 背书策略：了解新的生命周期如何使您可以将渠道配置中的策略用作链码背书策略。

限制与限制

v2.0 Alpha 版本中新的 Fabric 链码生命周期尚未完成。具体来说，请注意 Alpha 版本中的以下限制：

- 尚不支持 CouchDB 索引
- 通过服务发现尚无法发现使用新生命周期定义的链码

这些限制将在 Alpha 版本发布后解决。

Fab 令牌

Fabric 2.0 Alpha 还为用户提供了轻松将资产表示为 Fabric 通道上的令牌的功能。FabToken 是一个令牌管理系统，它使用 Hyperledger Fabric 提供的身份和成员资格基础结构，使用未用交易输出（UTXO）模型来发行，转移和兑换令牌。

- token / FabToken：本操作指南详细介绍了如何在 Fabric 网络上使用令牌。该指南还包括有关如何使用命令/令牌 CLI 创建和传输令牌的示例。

Alppine 镜像

从 v2.0 开始，Hyperledger Fabric Docker 映像将使用 Alpine Linux，这是一种面向安全的轻量级 Linux 发行版。这意味着 Docker 映像现在要小得多，提供了更快的下载和启动时间，并且占用了主机系统上更少的磁盘空间。Alpine Linux 在设计时就从头开始考虑安全性，Alpine 发行版的极简性质极大地降低了安全漏洞的风险。

Raft Ordering 服务

在 v1.4.1 中引入了 Raft，它是一种基于 etcd 中 Raft 协议的实现的崩溃容错（CFT）订购服务 。筏遵循“领导者和跟随者”模型，其中（每个通道）选举一个领导者节点，并将其决策复制到跟随者。与基于 Kafka 的订购服务相比，筏订购服务应更易于设置和管理，其设计允许遍布全球的组织为分散式订购服务贡献节点。

- 订购服务：描述了订购服务在 Fabric 中的作用，并概述了当前可用的两种订购服务实现：Kafka 和 Raft。
- 配置和操作 Raft 订购服务：显示部署 Raft 订购服务时的配置参数和注意事项。
- 设置订购节点：描述了部署订购节点的过程，而与订购服务的实现方式无关。
- 建立您的第一个网络：已更新，可让您将 Raft 订购服务与示例网络一起使用。

发行说明

发行说明为使用新版本的用户提供了更多详细信息，以及指向完整发行版更改日志的链接。

Fabric v2.0.0-alpha 发行说明。

Fabric CA v2.0.0-alpha 发行说明。