Constellation

区块链微服务运行系统

2017年11月25日

摘要

当前的区块链技术，如比特币、以太坊及其它都有系统性同步局限，从而使其不适合大量应用于当前技术架构下的分布式消费应用。共识机制和智能合约架构不能被扩大至消费级应用运行所要求的的级别。我们构想了一个抗故障，可水平扩展的分布式操作系统， 它可以作为一个移动客户端应用全部节点。因此我们提出将密码安全性共识改革为新式无服务器架构，这个架构利用Extended Trust Chain, Proof-of-Meme共识模式，基于JVM的组合式微服务智能合约，最初作为有限状态机之上的一个任务参与者. 这个架构确保了高交易吞吐量，能够使消费级分布式应用程序在Constellation上被打造。

简介

暂且不说去中心化的目标，日益中心化的网络和挖矿组织控制着网络安全，操作和保护着加密货币。随着后比特币网络的兴起，如以太坊，通过EVM和它的智能合约系统逻辑，处理金融交易的最初目的现已被扩展为提供去信任分布式计算。虽然存在这些分布协议多样化、创新性的应用，但他们都根植于当今主流软件，进入门槛很高。企业使用不熟悉的编程语言和设计方式开发，运行和维护一个活跃的分布式应用程序是一件高成本、耗时巨大的事。在当今最先进的公共区块链之上，扩展一个应用至一个中等企业的使用需求，这样的使用案例在实际上是不可能的。为了克服这些技术挑战，区块链科技需要一个自我可持续协议，它可以作为一个水平可扩展的，低成本的高效分布式操作系统。

1 数字货币类’比特币扩展协议共识 2017’ medium.com/@DCGco
2 J. Evans ’区块链是新的Linux系统，不是新的因特网’techcrunch.com

比特币是为解决金融交易分布共识问题所创造的，但是它依赖于高耗能的共识过程，即工作量证明。这聚集了具备发达的、集中计算能力的特定少数的垂直计算能力。这也将网络安全和回报置于这些同样的特定少数的手中。扩展性争辩在这些密谋团体内升级，这催生了许多比特币分支的创造，如Bitcoin Cash和 Bitcoin Gold。这些单独的政治派系极具分裂型，并危害整个生态系统。不合时宜的吞吐量要求（如1MB的区块容量），加之高耗时的算法已经使交易时间和交易费用突飞猛涨。以太坊使用类似的反ASIC工作量证明共识算法，并在朝权益证明共识机制Casper前进。在权益证明环境下，民主不平衡，那些最有钱的能够通过共识选择网络的状态。以太坊和智能合约的使用并驾齐驱，然而由于它们的异步执行，消费级分布式应用程序处于严重的瓶颈。同步是提供针对消费者产品的关键，Constellation以其微服务架构智能合约做到了这点。

什么使Constellation与众不同？

区块链发展团体一直在寻找新的分布式记账实施方法，以解决扩展性争议，但截至目前为止都没法解决这点。这个问题依然存在：我们如何在更少的时间内，以最低的成本处理更多的交易？我们提议一个水平扩展途径，它使用类似MapReduce的方法。

水平扩展是并发编程的应用；它表示随着用户的加入，交易吞吐量增加。MapReduce是一个将计算分解为简单操作的流程，这些简单计算可以被送入计算异步DAG（有向非循环图），从而增加已并发程序的效率。

Constellation协议应用了一个水平扩展区块链结构，即Extended Trust Chain，它使用了可以用于移动设备的P2P层，即gossip协议。Constellation通过微服务架构获取智能合约，微服务架构可以使高度便捷服务通过理解每个微服务的SLA（服务水平协议）和/或函数签名，镶嵌及组合至分布式应用程序之中。

3 比特币是否已经变得中心化？' www.trustnodes.com

Gossip协议可以使大型网络在高于现行区块链技术数量等级的规模上进行整体网络状态交流。在这种类型的网络里，网络里的每一成员记录其临伴，当其接收到消息时，它反过来向所有的临伴传播消息。这具备一些非常有趣的数学特性，但是在我们的看来，它使大量连接的设备可以共享网络状态，并在现今区块链应用上形成共识。（见图1）

交易吞吐量（tps）			
比特币	以太坊	IOTA(Tangle)	Hylochain(1200节点-固定邻居）
3-4	15	500-800	4000-4800

图1：可对比的区块链吞吐量对比

能够使区块链共识具备水平可扩展性的方法之一便是Extended Trust Chain。在Extended Trust Chain中具有多种节点，每种节点在网络中都有其自己的责任和角色，异步掌管协议的不同方面。

4 R. van Renesse '基于Gossip区块链?' zurich.ibm.com

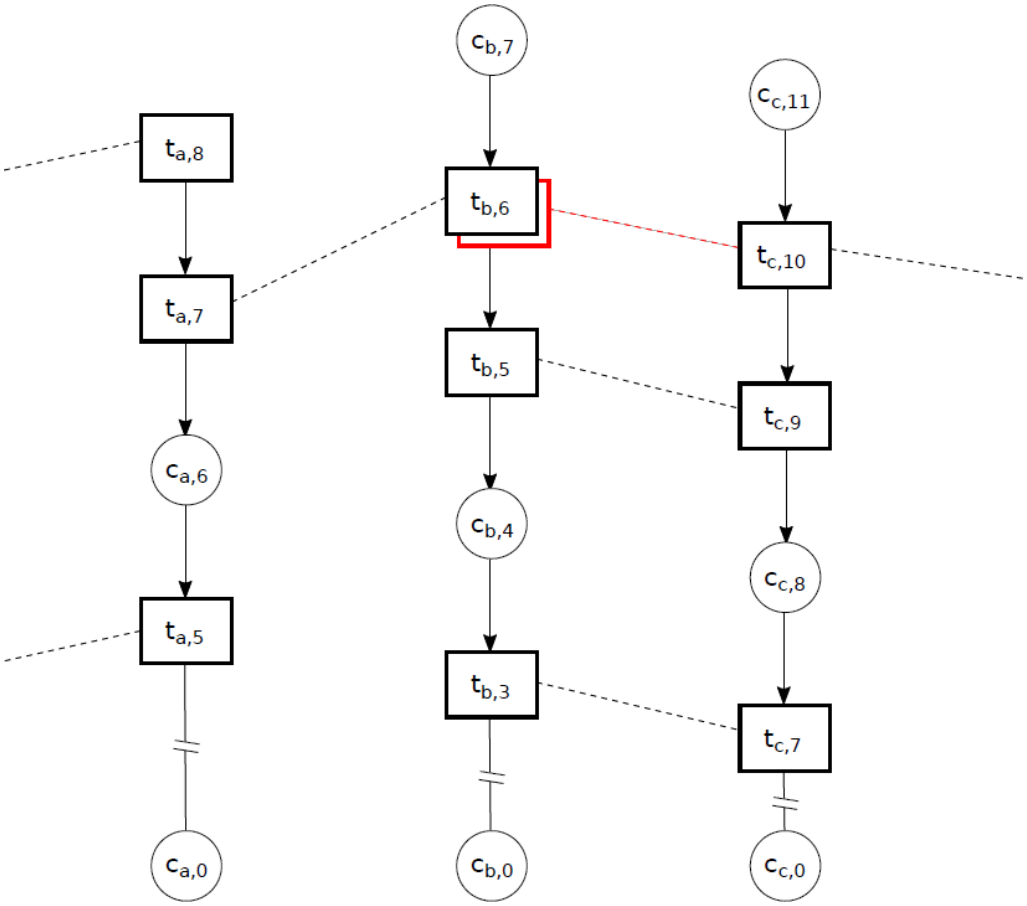

图2：Extended Trust Chain 图表，K. Cong及其他人

节点

具体而言，基本节点发送交易并存储一个成员的个体链（当交易达成时，每个体的链和网络上的其他形成一个DAG，它被以哈希函数写入线性块。）

检查点程序

节点程序对交易实施共识（存入缓冲区并以哈希函数写入检查点块），对这些交易的收集进行哈希，并变成主链中的下一块。

• • •

验证者程序

最终，验证者节点运行验证程序，存储当前链的状态，此后将链的状态/历史植入网络上的节点。

EVM vs JVM

Plasma升级至以太坊时提提出了分布式应用程序构想，如智能合约上的MapReduce操作。然而，随着其应用至EVM，它仍然进入着同样的误区，即如何以最低成本、低延迟的方式进行同步架构扩展。Solidity编程语言的构建是为了解决智能合约发展不确定性风险。确定性指导的需求来源于计算执行智能合约的成本需求，并作为网络上防御DDOS攻击和垃圾信息的防御性措施。但是，如果编译后的字节码可以在链上公证并在JVM中验证，那么确定性编译就变成了更宽松的要求。燃气消费的消减，燃气限制的规避，具备微服务复杂性的智能合约变成可能。

设计以共用基础模块构成的分布式应用程序是合理的，这就如组装乐高玩具，每一片都是现成的微服务。更进一步，我们采取链上互通，他可以使区块链公司提供可供他人再使用的分布式应用程序，作为更复杂的分布式应用程序的基础模块。一个真正分散的生态系统不可能通过一整个智能合约而存在，这使得这种类型代码的重用变得不可能并且低效。

使用JVM作为分布式程序的导体在大数据社区是工业标准。消费者应用部署一系列微服务于自动规模调整组，这些组依赖于JVM所提供的配置。在这种情形下，AWS虚拟主机可提供并自动调整数千节点群。同样的扩展性可通过Constellation的微服务架构实现，它可以随时按需补充资源；从而提供一个动态可扩展分布式应用程序架构。

这种类型的分布式架构内在要求功能性编程所提供的模块度。因此，我们使用Scala编程语言打造交互端口，这些交互端口能被任何JVM语言所使用。当微服务被设计成具体类型时，可以直接使用源代码构成分布式应用程序，允许在编写时验证。开发商可以通过将可用的智能合约链在一起形成组合应用；作为一个或多个微服务/分布式应用程序的计算逻辑constellation便被编成了。Constellation的智能合约端口将内在被设计为协变和逆变型，这意味着如果我们混合不兼容的微服务或是分布式应用程序，我们将会立即知道。正因为这样，我们甚至可以创造出包含于

其它区块链之内的智能合约constellation，这是一个潜在的跨链流动性问题（如波卡链）的解决法。

考虑到我们的功能性编程ethos,我们的共识协议可以被理论化地绝对定义为hylomorphism。

5 V. Buterin 等人. https://plasma.io/plasma.pdf

6 M. Hearn 'Corda: 分布式记账' docs.corda.net

HyloChain - 共识架构

我们将我们的共识架构命名为HyloChain，因为Constellation基本的共识架构是hylomorphic。一圈共识需要上一圈的哈希区块结果，并将其作为常规交易加入交易池。交易池的填充是同型的开放式操作。一旦检查点区块被之前圈和交易填充，它将被哈希化。这就是同型开放式操作。

定义: HyloChain

A hylomorphism h : A C 可以分别就其变形 和变质的部分被定义， 定义 hylomorphism为

$$h = [(c, \oplus), (g, p)] \qquad (1)$$

变形部分可以以一元函数g : A →B来定义。A 通过重复应用或展开定义B中的元素，谓词 p : A → 布尔提供了终结条件。

变质部分可以被定义为初始值c ∈ C 的集合和二元运算符 ⊕ : B × C → C 的迭代。

7 G.Wood 等人. https://github.com/w3f/polkadot-white- paper/raw/master/PolkaDotPaper.pd

在我们的情形中，消息的gossiping是我们的变形，这些消息的哈希值和前一区块的结果是我们的变质.

如果运算符被定义为加密哈希函数, ⊕ = CHash ，g n = (n; n − 1)，p n = False（我们的链没有终结条件），做n次迭代，HyloChain可以被绝对定义为

HyloChain = [(GenesisBlock, ⊕), (g, p)]　　(2)

创始区块是我们的初始元素，当我们部署HyloChain时使用创始区块。

与Extended Trust Chain的关系

我们的共识架构HyloChain建于Extended Trust Chain架构之上，并在其之上以如下方式扩展。在Extended Trust Chain中，每一个节点作为一个账户运行，该账户保存它自己链的历史，依赖于交易顺序来验证（类似于哈希图）。交易被发起者和合约对方签名，然后广播在网络中（通过gossip）。受托人通过信用被选中，在检查点区块内对交易执行共识。共识结果再次被广播，作为正常交易被加入下一区块。

8 L BAIRD等人. http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf

9 Kelong Cong,等人. https://repository.tudelft.nl/islandora/object/uuid:86b2d4d8-642e- 4d0f-8fc7-d7a2e331e0e9?collection=education

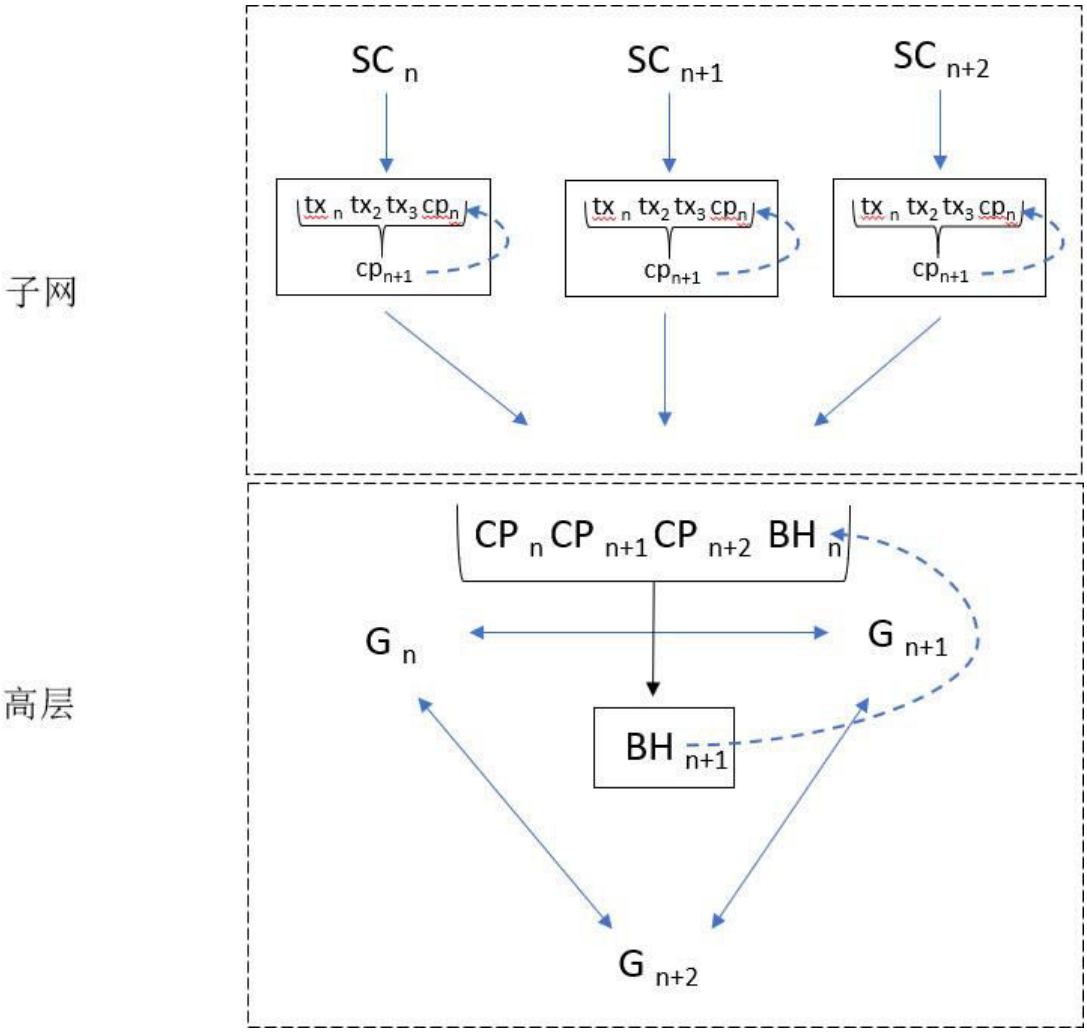

图3: HyloChain

容错

Constellation的HyloChain容错遵循传统的拜占庭共识模型。当其所控制的共识参与者接近1/310 时，敌对者控制网络共识的概率趋近1/2。为了确保网络安全，在最大化交易吞吐量并减少延迟的情况下，我们可以使用这些限定，选定特殊参数来调试容差。

10 M. Pease, R. Shostak 和 L. Lamport, '在出现差错时达成协议', Journal of the ACM (JACM), vol. 27, no. 2

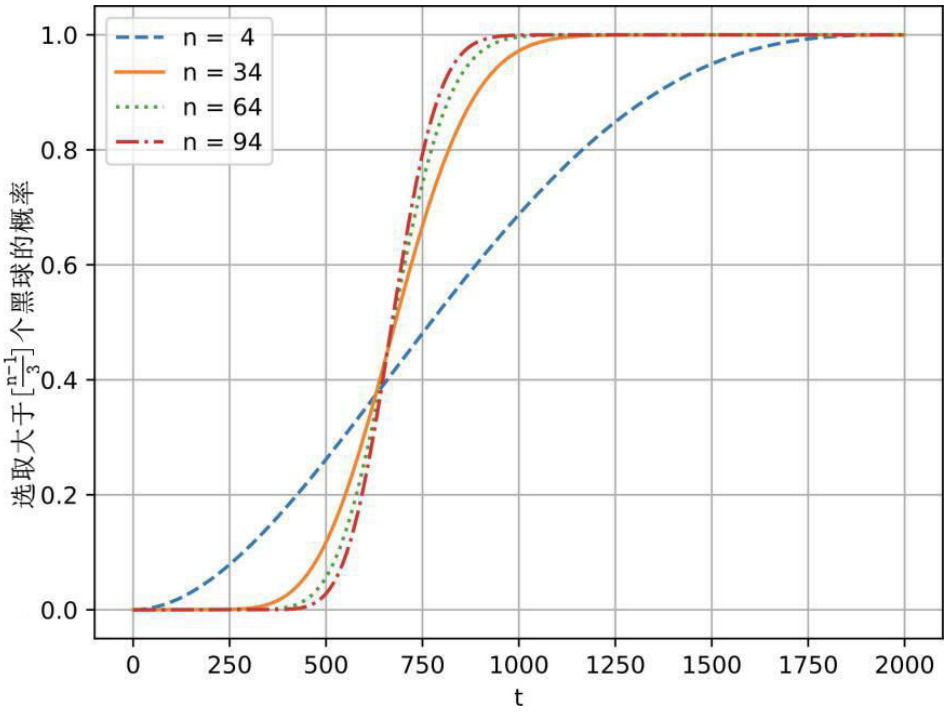

图 4: 固定值2000 K群体大小下，针对不同t（t是拜占庭因子的数目）值，选取值大于（n-1）/3（指黑球数，来自超几何分布的解释）时的概率值，K. Cong 等人

另外，基于我们的拜占庭共识程序（我们倾向于使用ACS，如M. Ben-Or等人所使用的，及Honeybadger BFT所使用的）和双重交易签名，网络分叉是不可能的。不正确共识的唯一结果是来自检查点区块的单独交易检查，如在共识过程中的交易丢失。这不会导致资本损失，简单的向网络再次发送便可消减此影响。

考虑到水平扩展，吞吐量和参与者的数量是线性关系，容错的线性增加引起交流成本的多项式增加。因此，我们必须确定共识参数，以保证在限定容错下的最大化吞吐量和交易确认时间。在这样的限定下我们如何扩展网络？我们的方法是将我们网络上的区块量分解为子网层级（见图1），每个层级执行其自己的共识，每个层级的共识结果像冒泡一样到达上一层，上一层以正常交易处理这些检查区块。这种方法和链纤维很类似(在验证时也被使用），然而不同的是我们通过对局部敏感哈希降维的方法增加交易池的大小。因为这个架构的作用是为了消减共识的复杂性，正常情况为O(n3)或是每个节点的二次。Honeybadger BFT中的共识算法可以被减少至每个委托人O(1) ，我们的复杂性变为

• • •

11 pp 50, K. Cong,等人.

$$O(n) \rightarrow O(m) : m << n \qquad (3)$$

这是子网中节点数的次线性。

在这种情况下，我们将这些检查区块称为局部敏感哈希。每一个子网形成一个局部敏感区块哈希，并发总给母网（星系，如下述）。这些母网在下一轮共识时将这些局部敏感区块哈希以普通交易处理。为了增强网络的性能，子网中邻居的选择仅依取决于最小化延迟，因为每个节点记录其自己的交易历史。

在每一层级中，采用和gossiped信息传递一样的自类似架构，允许同一节点通过仅改变节点所提供的资源参与到每一层。

Constellation中的所有节点可能“休息”，例如它们可以在任何时候加入或离开网络。一旦子网达到如上所述的交易吞吐量阀限和密码安全级别（facilitators的数量VS参与者数量），进入的节点必须形成一个新的子网。因此，随着成员的离开和进入网络，新的子网被不断动态分配。如下所述的自类似结构和我们的受托人选择模式是为了确保基于网络资源的动态自动扩展，产生一致性的吞吐量。展示这种架构显示出的无标度网络固有的幂律是没有意义的，它因在复杂系统，如分布式计算中容差的应用而臭名昭著。

12 M. Ben-Or 和 R. El-Yaniv.弹性-恒定时间内的最佳交互一致性. 分布式计算, 16(4):249?262, 2003

13 Ravasz, E.; Barab´asi (2003).”复杂网络中的分层组织”.Phys.Rev. E. 67: 026112.

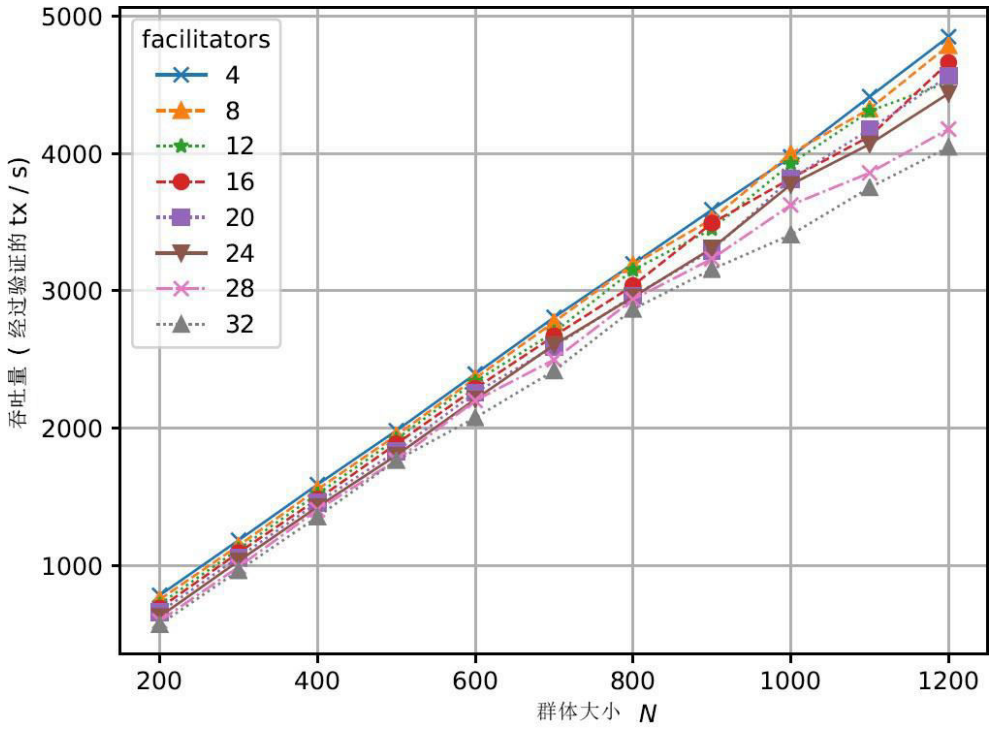

图5: HyloChain 吞吐量 vs 群体, K. Cong等人.

受托人选择模式: Proof-of-Meme

一个好的受托人选择系统可以提高Trust Chains中的容差。因此我们提出Proof-of-Meme，包括节点参于受托人选择的历史。Meme是一个单位，代表一种可以传播的模仿想法或是行为。在我们的情形中，它代表整个Constellation中得到奖励的善意行为，应被模仿以提高系统中节点的整体信用。Proof-of-Meme是英才管理，而工作量证明是财阀管理。

在我们的意义上，Meme是对应于每个节点帐户的特征向量；在最简单的情况下，它是作为确定性机器学习算法输入的浮点数矩阵。我们跟随REGRET的足迹，它使用特征的本体来描述在网络工作中的节点的信誉；技术层面上，这是特征空间的张量积。对应的确定性机器学习模型使用本体（Meme）作为其特征空间，来确定节点的信用分数及所指节点被选中参于共识的可能性。

14 pp 50, K. Cong,等人.

15 J. Sabater, REGRET: 社交社会的信用模式

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.7554rep=rep1type=pdf

Meme是信用分数的基础，信用分数赋予了被选中共识的可能性

在异步共识机制下提高BFT，概率性的受托人选择被在这样的环境下进行了大量研究。特别是在 A. Biryukov等人的GURU研究中表明，拜占庭共识中1/3 恶意节点的典型容错可以被提升至（在许多情况下可以超过）1/2。在我们的Proof-of-Meme受托人选择中，我们采纳了他们的验证框架。

和Extended Trust Chain关联的一个工作量区块体解决了sybil抗信用建模问题。具体而言，P. Otte使用了NetFlow算法来抵御sybi攻击，和一个修正的PageRank，即Temporal PageRank来跟新信用状态。Constellation将首先复制Temporal PageRank，然后在实施分析时尽其所能地改进。

我们通过Proof-of-Meme选择受托人，基于分值使用信用，结合每个节点作为单独账户的职能，通过增加透明性，激发良好行为，使我们的无需许可式网络具备私有网络的优点。每个节点（和传递的微服务）的所有交易和共识历史都公开认证。这使我们具备当下科技所忽视的信任，它通过交易费用和不平等的共识机制来加强良好的行为。

Proof-of-Meme释义

我们已经讨论了三种可解决分布式公司工作量本体，它们协同解决分布式计算中的问题。以下是我们受托人选择的构成。我们将跟随REGRET所使用的方法，开发Meme本体，它将作为我们信用分值的特征空间。我们将GURU的验证框架整合至我们的受托人选择方法。更多具体东西将在开发程序中被加进去。为受托人选择打造一个信用本体，培养一个信用分值确定模型并实施抽样算法是我们开发的关键部分。

16 Guru: 分布式共识协议的通用信用模型, A. Biryukov 等人l. https://eprint.iacr.org/2017/671.pdf

17 P.Otte,”分布式系统中抗Sybi信用机制https://repository.tudelft.nl/islandora/object/uuid:17adc7bd-5c82-4ad5-b1c8- a8b85b23db1f/datastream/OBJ/

我们现在详细讲述这种分值如何在受托人选择中使用。每一个功能代表一个节点对网络的效用（在共识中它是否是明显错误的，它能为共识贡献多少存储，诸如此类等等）。通过将其传递至一个函数中，Meme可以被转换为数字值，就是如上我们所说的被培养的模型，也是我们如何量化Meme信用的方法。为了提供一种促进新节点改进其Meme的机制，在固定大小间隔下绘制Meme得分分布概率图；分布图展示了在特定的被选间隔中Meme的得分。从计算的角度来看，这是将节点从新聚类为子程序，实施聚类算法。这意味着在放宽facilitator数量的同时保持同级别的容差和确认时间，最大化吞吐量。信用良好的Meme将具备优先选择权，但是心得Meme也将拥有参与的机会，在网络上积累其信用。执行记录将在链上被公证，提供错误资源或是以不正确的方式运行的Meme将消减其信用，高性能的可信Meme将增加其信用。在诱导的情形下，我们的受托人选择理念如下：

1) 执行共识
2) 哈希区块被输入确定性算法，生成每个受托人的Meme得分。这是在最高共识层(Galaxy)执行的，负责选择facilitators。
3) 这个常量被用来对分布进行洗牌T(基于其效能在间隔之间移动Meme)。
4) 上一个区块哈希值（上一个共识结果）被用来对每个间隔里的内容进行排序，每个间隔中的前N个（根据概率分布）被这一轮选中。行为错误或是可证明是恶意的（在记录中可证明的）共识参与者中的Meme将会被抛至下次参与共识选举。

随着Meme信用承载更多价值，它将取代交易费。微服务可以选择信用好的Meme作为服务寄宿，这比执行共识获取手续费更加容易盈利。与其在高延迟时增加价格，低信用分的Meme将会被制约，它们的交易将以低优先级处理。像这样的Meme可以提供所具备的资源（参与共识），从而增加其自己的吞吐量并解决网络带宽问题。

获取你的Meme

Constellation将不会有网络交易费，反而信用将在我们的机制中防御延迟和恶意攻击。为了发展我们的网络，在最初的创始区块之后，我们将以新铸代币形式提供网络奖励（为ICO参与者分配代币），随着收益递减至特定日期，代币铸造将停止。新铸代币将以其各自的Meme信用进行分配。

设想一个销售点系统的情形，特别是自动售货机。依靠Constellation运行的自动售货机可以实现为通过共识选择参与者的微服务。它的Meme信用将随着时间增长，即使在高流量下也能确保较短的交易时间。这很重要，因为现实世界的应用需要大的吞吐量来提供具备竞争力的服务。这意味着即使一个自动售货机的顾客只有很低的Meme信用，即使在高峰时段，自动售货机的高信用确保了顾客方便地得到他们的物品。

P2P架构

前一章节初步接触了我们的P2P层架构，现在详细阐述。

Star

Star是constellation中最基础的东西。用户可被视为设备上的Star节点，交易通过这个Star实体发出，和网络直接互动。每一个Star包含一个局部链，局部链由其在网络上的历史组成。这个局部链用来强化编号，可被公钥识别，公钥的私密被用来签署交易。Star本身就是轻量级的客户机，用户可以使用它来和网络交互，Star和移动设备兼容。然而，Star可以根据其意愿参与共识，这能使其Meme信用增加，然后加入Star集，我们称之为星团。

星团 Star Clusters

星团是一系列被选中参与共识的Star。Star的总数由上限决定，这个上限将在上文所诉的充分实验和数据分析后确定。当我们达到这个阀限，一个新的星团被创造，它们每个都形成了局部敏感哈希区块。这些局部敏感哈希区块将被以普通交易对待，通过星系哈希化写入黑洞。

星系 Galaxies

星系和Extended Trust Chain中的验证者的角色是同型的，它同时作为一个自动扩展组，为新的星团提供资源并维持Meme信用。Meme信用通过查阅共识执行记录进行计算。区块所提出的内容的元数据，延迟和故障都被以局部敏感哈希区块传送至星系。一旦星系接收到数据，在选择新样本进行下一轮共识之前，Meme信用被更新。星系作为验证者，为移除无效交易和在星团中确定受托人s提供了真实性的来源。星系运行的元数据存储在黑洞中。如果Star的信誉超过了阀限，它或可获得一星系节点运行的权利。

黑洞 Blackhole

黑洞是哈散列的局部敏感哈希区块的区块。称它为区块链中的区块也一样。星系存储整个区块链的历史。

微服务智能合约

高度可用，弹性，分部式系统在无服务器架构上茁壮成长。就分布式操作系统而言，分布式微服务网络可以做到这一点。因此在Constellation中智能合约本身是运行在JVM上的微服务。它们可以发送交易，作为合约对方签名，并执行共识。以相应的Meme使微服务本身像Star一样运行，为商定的部分提供服务。它们可以在以太坊和Counterparty中扮演和智能合约一样的角色，另外，它们可以利用JVM生态系统中存在的代码库提供更多的复杂逻辑。如果这些微服务是以有形的服务水平协议，或是更好，以类型签名打造的，它们的组合逻辑可以直观的被镶嵌或组成于分布式应用程序之中。这正是MapReduce开始起作用的地方。智能合约微服务可以被设计为

具备发送和接受数据能力的模型，从而提高计算复杂性（基于我们的异步架构），也能被改换用途，用于新的应用程序。

关于以上美丽的比喻，值得注意的是Constellation的分布式应用程序就是constellation本身。因为每一个微服务就是一颗Star，一系列链接和/或组合的微服务可以通过程一条线—应用程序连接起来。当画出来时，很平常地就能注意到，这些形成了一个星群。

Constellation作为区块链运行系统

以上无服务器架构是分布式操作系统的一个例子。运行系统的目的是为运用基础的硬件资源提供一个交互端口。考虑到高层编程语言和直观的用户交互界面，这应用开发最基本的。分布式操作系统的不同在于，它旨在为分布计算群的基础资源提供一个交互界面。区块链内在地是一个分布系统，为了利用基层群的全能力，编写区块链可兼容的应用需要一个分布式操作系统。一般而言，操作系统需要程序来维持基础硬件的状态，这就是内核。通过无服务器架构来构建一个分布式操作系统是可行的。按我们的设想，在MicroOs中，操作系统就是一系列微服务。从应用开发商的视角来外推，像乐高积木一样，微服务是用Constellation来构建整体应用的积木。这样做的目的是能使一个非技术个人像堆积木一样使用现成的微服务来构建应用。加之直观的用户界面，这为非技术用户提供了一个无缝入口，使其能够在Constellation上开发分布式应用程序。

正如上文所述，使个人无缝的（非技术性地）将一个想法实现到应用之中，或许可以说这种能力是从人类知识向现实物质世界的势能演变。正因为如此，鉴于那些能够实现我们的目标和想法的人类知识势能，我们决定将我们的货币命名为noOs。

结论

我们提出将密码安全性共识改革为新式无服务器架构，从而使主流应用可以使用区块链科技。我们的方法应用了局部敏感散列的降维技术作为分治机制，使Trust Chain能够在整个网络建立。我们对Meme的充分使用使密码安全更深一层，为淘汰交易费用提供了可能。此外，我们还为我们的分布式架构提供了一个天体隐喻，描绘了分布式应用程序与自然界中可观察到的自相似扩张形态之间的直观关联。