Aptos 区块链：安全、可扩展和可升级的 Web3 基础设施

v1.0*

2022 年 8 月 12 日

摘要

区块链作为一种新的互联网基础设施的兴起，导致开发人员以快速增长的速度部署了数万个分散的应用程序。不幸的是，由于频繁的停机、高成本、低吞吐量限制和众多安全问题，区块链的使用尚未普遍存在。为了在 web3 时代实现大规模采用，区块链基础设施需要遵循云基础设施的道路，作为一个可信、可扩展、经济高效且不断改进的平台，用于构建广泛使用的应用程序。

我们提出了 Aptos 区块链，其设计以可扩展性、安全性、可靠性和可升级性为关键原则，以应对这些挑战。Aptos 区块链在过去三年中由全球超过 350 家开发商开发[1]。它在共识、智能合约设计、系统安全、性能和分散化方面提供了新的创新。这些技术的结合将提供一个基本的构建块，将 web3 推向大众：[1]

- 第一，Aptos 区块链在本地集成并在内部使用 Move 语言，以实现快速安全的交易执行 [2]。Move prover 是用 Move 语言编写的智能合约正式验证器，它为合约不变量和行为提供了额外的保障。这种对安全性的关注使得开发人员能够更好地保护他们的软件免受恶意实体的侵害。
- 第二，Aptos 数据模型支持灵活的密钥管理和混合托管选项。这与签名前的交易透明度和实用的轻型客户端协议一起，提供了更安全、更可信的用户体验。
- 第三，为了实现高吞吐量和低延迟，Aptos 区块链在交易处理的关键阶段采用了流水线和模块化方法。具体来说，交易传播、区块元数据排序、并行交易执行、批量存储和账本认证都是同时运行的。这种方法充分利用了所有可用的物理资源，提高了硬件效率，并实现了高度并行执行。
- 第四，与其他通过要求预先了解数据读写来打破事务原子性的并行执行引擎不同，Aptos 区块链不会对开发人员施加此类限制。它可以有效地支持任意复杂事务的原子性，从而为现实世界的应用程序实现更高的吞吐量和更低的延迟，并简化开发。
- 第五，Aptos 模块化架构设计支持客户端灵活性，并针对频繁和即时升级进行了优化。此外，为了快速部署新技术创新并支持新的 web3 用例，Aptos 区块链提供了嵌入式链上变更管理协议。
- 最后，为了超越单个验证器的性能，Aptos 区块链正在试验前卫的举措：其模块化设计和并行执行引擎支持验证器内部分片，同质状态分片提供了水平吞吐量可扩展性的潜力，而不会增加节点运营商的额外复杂性。

1 介绍

在互联网的 web2 版本中，消息传递、社交媒体、金融、游戏、购物和音频/视频流等服务由控制用户数据直接访问的中心化公司提供（例如，Google、Amazon、Apple、和 Meta）。这些公司使用针对

*特别鸣谢此白皮书的翻译人员：Boyu, Container, Kusou1, Ruyisu，ywww，廖师虎，肖川, Tom, Geometryolife 等

[1]法律免责声明：本白皮书及其内容不是任何代币的出售要约或购买要约的邀请。我们发布此白皮书仅是为了接收公众的反馈和意见。本文档中的任何内容均不应被阅读或解释为 Aptos 区块链或其代币（如果有）将如何开发、使用或增值的保证或承诺。Aptos 仅概述了其当前的计划，这些计划可能会自行改变，其成功将取决于其无法控制的许多因素。此类未来陈述必然涉及已知和未知的风险，这可能导致未来期间的实际表现和结果与我们在本白皮书中描述或暗示的内容存在重大差异。Aptos 不承担更新其计划的义务。无法保证白皮书中的任何陈述将被证明是准确的，因为实际结果和未来事件可能存在重大差异。请不要过分依赖未来的陈述。

目标用例优化的特定应用软件开发基础架构，并利用云基础设施向用户部署这些应用程序。云基础设施提供对虚拟化或者物理基础设施服务的访问，例如租用的虚拟机 (VM) 和在全球数据中心（例如 AWS、Azure 和 Google Cloud）内运行的裸机硬件。因此，构建可扩展到数十亿用户的 web2 互联网服务从未像今天这样容易。然而，web2 要求用户明确信任中心化实体，这一要求让社会越来越担忧。

为了消除这种担忧，一个新的互联网时代已经开始：web3。在互联网的 web3 版本中，区块链已经出现，其提供了分散的、不可变的账本，使用户能够安全可靠地相互交互，而无需信任任何控制中介或中心化实体。类似于 web2 互联网服务和应用程序如何依赖云基础设施作为构建块，去中心化应用程序可以使用区块链作为去中心化基础设施层，从而覆盖全球数十亿用户。

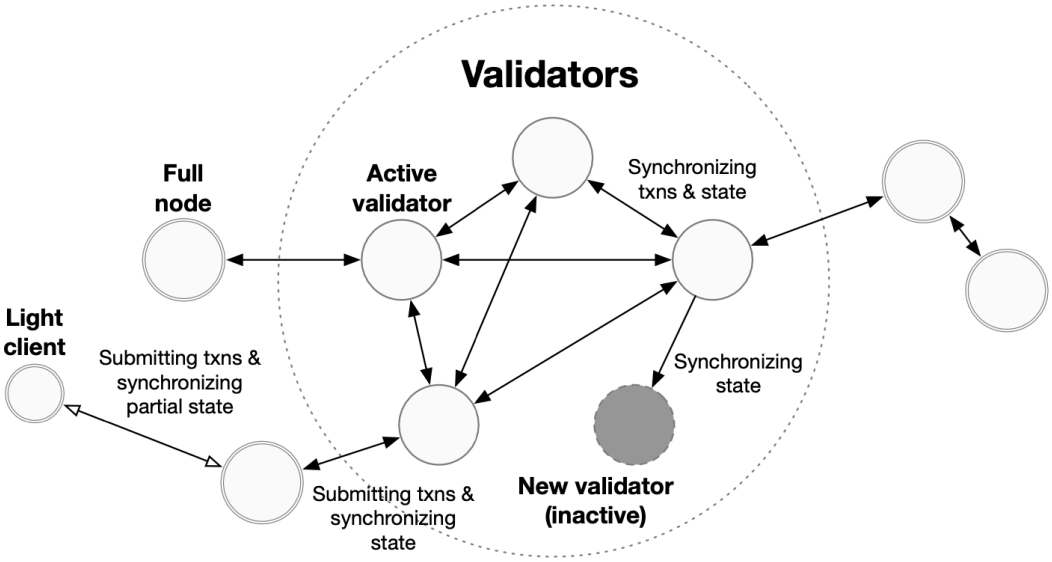
然而，尽管当今存在许多区块链，但 web3 尚未被广泛应用 [3]。虽然技术不断推动行业发展，但现有区块链不可靠，对用户收取高额交易费用，吞吐量受限，因安全问题经常遭受资产损失，并且无法支持实时响应。与云基础设施如何使 web2 服务达到数十亿相比，区块链尚未使 web3 应用程序达到相同的效果。

2 Aptos 的愿景

Aptos 的愿景是提供一个区块链，可以为 web3 带来主流应用，并增强去中心化应用程序的生态系统，以解决现实世界的用户问题。我们的使命是通过提供灵活的模块化区块链架构来提升区块链可靠性、安全性和性能方面的最新技术水平。该体系架构应支持频繁升级、快速采用最新技术进步，以及对新兴用例提供一流的支持。

我们设想一个使用它进行社区管理和运营的去中心化、安全和可扩展的网络。当全球基础设施需求增F时，区块链的计算资源会横向和纵向扩展以满足这些需求。随着新用例和技术进步的出现，网络应该在不中断用户的情况下频繁且无缝地升级。对基础设施的担忧应该逐渐消失。开发人员和用户将可以访问许多不同的选项，用于密钥恢复、数据建模、智能合约标准、资源使用权衡、隐私和可组合性。用户知道他们的资产是安全的、始终可用的，并且可以以接近成本的费用进行访问。任何人都可以安全、轻松、不变地与全球不受信任的各方进行交易。区块链与云基础设施一样无处不在。

为了实现这一愿景，必须取得重大的技术进步。我们在过去三年中构建、开发、推进和部署 Diem 区块链（Aptos 区块链的前身）的经验证明，网络可以在不中断其客户端的情况下不断升级其协议 [4]。2020 年初，Diem 主网部署到十几家节点运营商，拥有多个钱包提供商。在接下来的一年里，我们的团队发布了两次重大升级，改变了共识协议和核心框架。两次升级都在用户没有停机的情况下完成。借助 Aptos 区块链，我们在 Diem 区块链的启发下，对技术堆栈进行了一系列彻底的改进，同时还将安全、透明和频繁的升级作为核心功能。特别是，我们强调了交易处理的新方法（如第 7 节所述）以及去中心化和网络治理的新方法。



随着 Aptos 区块链的不断改进和发展，我们将发布本白皮书的更新版本，其中包含我们协议和设计选择的最新迭代。在本文档的其余部分，我们将描述 Aptos 区块链的当前状态以及未来计划。

3 概述

如图 1 所示，Aptos 区块链由一组验证器组成，它们使用拜占庭容错 (BFT)、股权证明共识机制共同接收和处理来自用户的交易。代币持有者在其选定的验证器中锁定或质押代币。每个验证器的共识投票权重与质押其中的数量成正比。验证器可以是活跃的并参与共识。同样，如果验证器没有足够的权益参与、轮换出验证器集、在同步区块链状态时选择离线，或者由于历史性能不佳而被共识协议视为不参与，验证器也可能处于非活动状态。客户端是系统中需要提交交易或查询区块链状态和历史的任何部分。客户可以选择下载并验证由验证器签名的查询数据证明。全节点是从验证器或网络中的其他全节点复制交易和区块链状态的客户端。他们可以根据需要选择修剪交易历史和区块链状态以回收存储。轻型客户端仅维护当前的一组验证器，并且可以安全地查询部分区块链状态，通常是来自完整节点。钱包是轻型客户端的常F示例。为了满足广泛采用安全、快速、可靠和可升级的 web3 基础架构的需求，Aptos 区块链建立在以下核心设计原则之上：

- 通过一种新的智能合约编程语言 Move [5]，快速安全地执行，具有简单的可审计性和机械分析性。Move 起源于 Aptos 区块链的前身，并随着该项目的发展而不断进步。
- 通过批处理、流水线和并行化的事务处理方法，实现极高的吞吐量和低延迟。
- 新颖的并行事务处理，通过 Block-STM 有效地支持任意复杂事务的原子性，这与需要预先了解要读取和写入的数据位置的现有并行执行引擎不同。
- 通过快速的权益权重验证器集和声誉跟踪实现性能优化以及去中心化。
- 将可升级性和可配置性作为一流的设计原则，以拥抱新用例和最新技术。
- 模块化设计支持严格的组件级测试，以及适当的威胁建模和无缝部署，所有这些都确保了高度安全和可靠的操作。

• 水平吞吐量可扩展性，同时保持分散化，其中分片是向用户公开的一流概念，也是编程和数据模型的原生概念。

第 4 节解释了开发人员如何与 Aptos 区块链中与 Move 交互。第 5 节描述了逻辑数据模型。第 6 节详细介绍了 Aptos 区块链如何通过强大的验证方法实现安全的用户体验。第 7 节描述了围绕流水线、批处理和并行化的关键性能创新。第 8 节详细介绍了不同客户端与其他节点同步状态的各种选项。第 9 节描述了我们的社区所有权和治理计划。最后，第 10 节讨论了在保持去中心化的同时未来的方向。

4 Move 语言

Move 是一种新的智能合约编程语言，强调安全性和灵活性。Aptos 区块链使用 Move 的对象模型来表示其账本状态（参见第 5.5 节），并使用 Move 代码（模块）对状态转换规则进行编码。用户提交的交易可以发布新模块、升级现有模块、执行模块内定义的输入函数，或者包含可以直接与模块公共接口交互的脚本。

Move 生态系统包含编译器、虚拟机和许多其他开发工具。Move 受到 Rust 编程语言的启发，该语言通过线性类型等概念明确了数据的所有权。Move 强调资源稀缺性、保存性和访问控制。移动模块定义了每个资源的生命周期、存储和访问模式。这确保了像 Coin 这样的资源不会在没有适当凭证的情况下产生，不会被重复使用，也不会消失。

即使存在不受信任的代码，Move 也利用字节码验证器来保证类型和内存安全。为了帮助编写更受信任的代码，Move 包含了一个正式的验证器，即 Move 验证器 [6]，它能够根据给定规范验证 Move 程序的功能正确性，该规范是用集成到 Move 中的规范语言制定的。

除了用户账户和相应的账户内容外，账本状态还包含 Aptos 区块链的链上配置。此网络配置包括一组活跃的验证器、质押属性以及 Aptos 区块链中各种服务配置。Move 对模块可升级性和全面可编程性的支持实现了无缝配置更改，并支持对 Aptos 区块链本身的升级（两组升级已多次执行，且在私有主网上实现零停机时间）。

Aptos 团队进一步增强了 Move，支持更广泛的 web3 用例。如后面 5.5 节所述，Aptos 区块链支持细粒度资源控制。这不仅支持执行的并行化，而且还实现了与访问和修改数据相关的近乎固定的成本。此外，Aptos 区块链提供了建立在细粒度存储之上的表支持，允许在单个帐户中存储大规模数据集（例如，大量的 NFT 集合）。此外，Aptos 支持完全在链上表示的共享或自治帐户。这允许复杂的去中心化自治组织 (DAO) 协作共享帐户，并将这些帐户用作异构资源集合的容器。

5 逻辑数据模型

Aptos 区块链的账本状态代表了所有账户的状态。账本状态使用一个无符号 64 位整数进行版本控制，该整数对应于系统已执行的交易数量。任何人都可以向 Aptos 区块链提交交易以修改账本状态。在执行交易时，会生成交易输出。交易输出包含零个或多个操作来操纵账本状态（称为写入集）、结果事件向量（参见第 5.1.1 节）、消耗的 gas 量和执行的交易状态。

5.1 交易（Transaction）

已签名的交易包含以下信息：

- 交易验证器：发送者使用包含一个或多个数字签名的交易验证器来确保交易是否经过验证。
- 发件人地址：发件人的帐户地址。
- 有效负载：有效负载要么指链上现有的入口函数，要么包含要作为内联字节码（称为脚本）执行的函数。此外，一组输入参数被编码为字节数组。对于点对点的事务执行，输入包含接收者的信息和转移给他们的金额。
- Gas 价格（以指定的货币/Gas 单位表示）：这是发送方愿意为执行交易而为每单位 Gas 支付的金额。Gas 是一种支付计算、网络和存储费用的方式。gas 单位是计算的抽象度量，没有内在的现实世界价值。
- 最大 gas 量：最大 gas 量是交易在中止之前允许消耗的最大 gas 单位。账户必须至少有 gas 价格乘以最大 gas 量，否则交易将在验证期间被丢弃。
- 序号：交易的序号。这必须与交易执行时存储在发件人帐户中的序列号匹配。成功执行交易后，帐户序列号会增加以防止重放攻击。
- 到期时间：交易停止有效的时间戳。
- 链 ID：标识此交易有效的区块链，提供进一步保护为用户防止签名错误。

在每个版本 i，状态变化由元组 (Ti,Oi,Si) 表示，分别包含交易、交易输出和结果账本状态。给定一个确定性函数 Apply，执行具有账本状态 Si-1 的交易 Ti 会产生事务输出 Oi 和一个新的账本状态 Si。即，Apply(Si−1, Ti) → Oi, Si 。

5.1.1

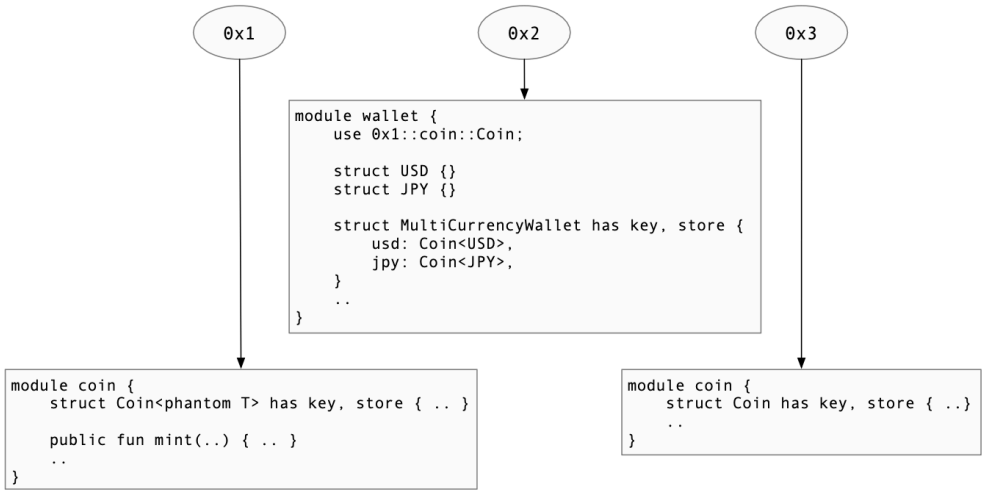
在交易执行期间发出事件。每个 Move 模块都可以定义自己的事件并选择在执行时何时发出这些事件。例如，在硬币转移过程中，发送者和接收者的账户都会分别发出 SentEvent 和 ReceivedEvent。这些数据存储在账本中，可以通过 Aptos 节点进行查询。每个注册的事件都有一个唯一的键，该键可用于查询事件详情。

发射到同一个事件键的多个事件会产生事件流，一个事件列表，每个条目包含从 0 开始的顺序增加的数字、类型和数据。每个事件都必须由某种类型定义。可能存在由相同或相似类型定义的多个事件，尤其是在使用泛型时。事件具有关联的数据。对于 Move 模块开发人员，一般原则是包含所有必要的数据，以了解在执行更改数据并发出事件的事务之前和之后对底层资源的更改。

交易只能产生事件，不能读取事件。这种设计允许交易执行仅取决于当前状态和交易输入，而不是历史信息（例如，先前生成的事件）。

5.2 账户

每个帐户都由一个唯一的 256 位值标识，称为帐户地址。当从现有账户发送的交易调用 *create_account(addr)* 移动函数时，会在账本状态下创建一个新账户（参见第 5.5 节）。这通常发生在交易尝试将 Aptos 代币发送到尚未创建的帐户地址时。为方便起见，Aptos 还支持 transfer(from, to, amount) 函数，如果该账户在转账之前不存在，该函数会隐式创建一个账户。



要创建一个新帐户，用户首先生成一个签名密钥对：(vk, sk)。接下来，使用与签名方案标识符 (ssid) 连接的公共验证密钥 vk 的加密哈希 H 导出给定签名方案的新帐户地址：其中 addr = H(vk, ssid)。

在地址 addr 创建新帐户后，用户可以使用私有签名密钥 sk 对要从 addr 帐户发送的交易进行签名。用户还可以轮换 sk，以主动更改 sk 或响应可能的妥协。这不会更改帐户地址，因为帐户地址在创建期间仅从公共验证密钥派生一次。

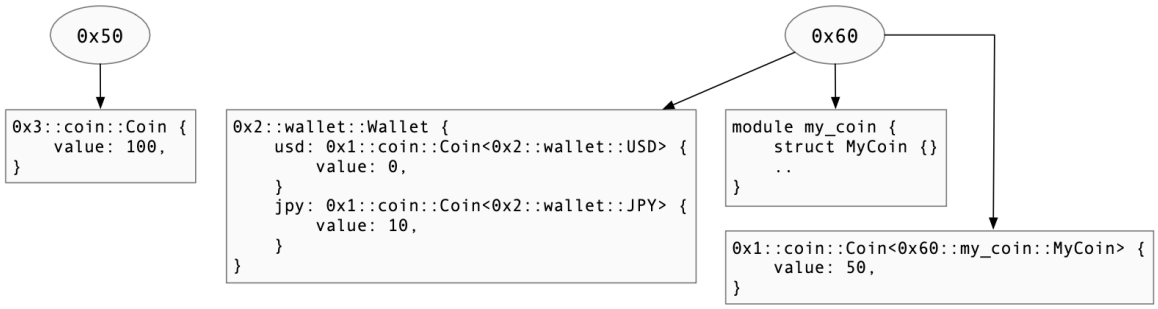
Aptos 区块链不会将账户与真实世界的身份相关联。用户可以通过生成多个密钥对来创建多个帐户。由同一用户控制的帐户彼此之间没有内在联系。但是，单个用户仍然可以在单个钱包中管理多个帐户，以进行简单的资产管理。这种灵活性为用户提供了匿名性，同时我们为未来的版本试验了保护隐私的原语。单个用户或一组用户拥有的多个帐户也提供了增加执行并发性的通道，如第 7.4 节所述。

5.3 Move 模块

Move 模块包含声明数据类型（结构）和过程的 Move 字节码。它由声明模块的帐户地址和模块名称来标识。例如，图 2 中第一个货币模块的标识符是 0x1::coin。一个模块可以依赖于其他链上模块，如图 2 中的钱包模块所示，从而实现代码重用。

一个模块必须在一个帐户中唯一命名，即每个帐户最多可以声明一个具有任何给定名称的模块。例如，图 2 中地址 0x1 的帐户无法声明另一个名为 coin 的模块。另一方面，地址为 0x3 的账户可以声明一个名为 coin 的模块，该模块的标识符为 0x3::coin。请注意，0x1::coin::Coin 和 0x3::coin::Coin 是不同的类型，不能互换使用，也不能共享公共模块代码。相比之下，0x1::coin::Coin<0x2::wallet::USD> 和 0x1::coin::Coin<0x2::wallet::JPY> 是同一泛型类型的不同实例，不能互换使用，但可以共享通用模块代码。

模块会被分组到位于相同地址的包中。该地址的所有者将包作为一个整体在链上发布，包括字节码和包元数据。包元数据决定了一个包是可以升级还是不可变的。对于可升级的包，在允许升级之前执行兼容性检查：必须更改现有的入口点函数，并且不能将资源存储在内存中。但是，可以添加新的功能和资源。Aptos 框架由 Aptos 区块链的核心库和配置组成，被定义为可定期升级的模块包（参见第 9.2 节）。



5.4 资源

与模块类似，帐户地址也可以具有与之关联的数据值。在每个帐户地址中，值按其类型作为键，每种类型最多有一个值属于该帐户。图 3 提供了一个例子。地址 0x50 保存单个值，其中 0x3::coin::Coin 是完全限定类型。0x3 是 coin 模块存放的地址，coin 是模块名称，Coin 是数据类型名称。泛型类型的值也是允许的，不同的实例被视为不同的类型。这对于可扩展性至关重要，允许不同的实例共享相同的功能代码。更改、删除和发布值的规则在定义数据类型的模块中进行编码。

Move 的安全和验证规则防止其他代码或实体直接创建、修改或删除其他模块中定义的数据类型的实例。

在一个地址下最多具有每个类型的一个顶级值可能起初听起来很有限。但是，这在实践中不是问题，因为程序员可以将包含其他数据的包装器类型定义为内部字段，从而避免任何限制。图 3 中的 Wallet 结构是如何使用包装器类型的示例。

还应注意，并非所有数据类型都可以存储在链上。为了使数据实例符合顶级值，数据类型必须具有键能力。同样，嵌套值也需要存储能力。具有这两种能力的数据类型也称为资源。

5.5 账本状态

从 Move 虚拟机（Move VM）的角度来看，每个账户都由一组值和键值数据结构组成。这些数据结构称为表条目，并以二进制规范序列化格式 (BCS) 存储。这种数据布局使开发人员能够编写智能合约，这些智能合约可以有效地处理跨大量账户复制的少量数据，以及存储在少数账户中的大量数据。移动模块的存储方式与帐户数据类似，但在独立的命名空间下。创世账本状态定义了初始账户集及其在区块链初始化时的关联状态。

在发布时，Aptos 区块链将由一个单一的账本状态表示。然而，随着采用的增加和技术的发展，Aptos 将扩大分片的数量以增加吞吐量（即启用多个分类帐状态）并支持跨分片移动或访问资产的交易。每个账本状态都将维护特定分片的所有链上资产，并为相同的账户模型提供细粒度的键值数据存储，为存储访问提供近乎固定的成本。

6 安全的用户体验

为了覆盖数十亿互联网用户，web3 用户体验必须安全且易于访问。在下面的部分中，我们描述了 Aptos 区块链提供的几项创新，旨在实现这一目标。

6.1 事务可行性保护

签署交易意味着签署者授权交易由区块链提交和执行。有时，用户可能会无意中签署交易，或者没有充分考虑交易可能被操纵的所有方式。为了降低这种风险，Aptos 区块链限制了每笔交易的可行性，并保护签名者免受无限有效性的影响。Aptos 区块链目前提供三种不同的保护——发送者的序列号、交易到期时间和指定的链标识符。

- 对于每个发送者的账户，一个交易的序列号只能提交一次。结果，发送者可以观察到，如果当前账户序列号 一个事务的序列号，那么要么 t 已经被提交，要么 t 永远不会被提交（因为 t 使用的序列号已经被另一笔交易）。
- 区块链时间以高精度和频率（通常为亚秒级）推进，详见第 7.3.1 节。如果区块链时间超过了事务 t 的过期时间，那么类似地，要么 t 已经被提交，要么 t 永远不会被提交。
- 每笔交易都有一个指定的链标识符，以防止恶意实体在不同的区块链环境（例如，跨测试网和主网）之间重放交易。

6.2 基于 Move 的密钥管理

如第 5.2 节所述，Aptos 帐户支持密钥轮换，这是一项重要功能，可帮助降低与私钥泄露、远程攻击以及可能破坏现有加密算法的未来进展相关的风险。此外，Aptos 账户也足够灵活，可以启用新的混合托管模式。在一种这样的模型中，用户可以将轮换帐户私钥的能力委托给一个或多个保管人和其他受信任的实体。然后，Move 模块可以定义一个策略，使这些受信任的实体能够在特定情况下轮换密钥。例如，实体可能由许多受信任方持有的 k-out-of-n 多重签名密钥表示，并提供密钥恢复服务以防止用户密钥丢失（例如，目前 20

6.3 提前验证的事务透明性

今天，钱包对其签署的交易几乎没有透明度。因此，用户通常很容易被诱骗签署可能窃取资金并造成毁灭性后果的恶意交易。即使对于需要枚举每笔交易访问的所有链上数据的区块链也是如此。因此，目前几乎没有用户保护措施，使用户容易受到各种各样的攻击。为了解决这个问题，Aptos 生态系统为交易预执行提供服务：一种预防措施，在签署之前向用户（以人类可读的形式）描述他们的交易结果。将此与已知的先前攻击历史和恶意智能合约相结合将有助于减少欺诈。此外，Aptos 还使钱包能够在执行期间对交易进行限制。违反这些约束将导致交易被中止，以进一步保护用户免受恶意应用程序或社会工程攻击。

6.4 实用的轻客户端协议

仅仅依靠 API 提供商的 TLS/SSL 证书在区块链客户端和服务器之间建立信任并不能充分保护客户端。即使存在有效证书，钱包和客户也无法保证所提供数据的真实性和完整性。图 4：事务处理生命

周期。所有阶段都是完全独立的，并且可以单独并行。给他们。因此，API 提供商可能会返回不正确或恶意的区块链数据，从而欺骗第三方并执行双花攻击。为了防止这种情况，Aptos 提供状态证明和轻客户端验证协议，钱包和客户端可以使用这些协议来验证不受信任的第三方服务器提供的数据的有效性。此外，通过利用第 7.6.2 节中基于时间戳的状态证明，轻客户端始终可以确保帐户状态的新鲜度（例如，在几秒钟内）的严格限制，并且只需要跟踪网络配置的变化（纪元变化）或使用当前受信任的检查点（航路点）来保持最新状态 [8]。通过结合高频时间戳和廉价的状态证明，Aptos 区块链为客户提供了更高的安全保证。此外，Aptos 节点还公开了丰富的高性能存储接口，可以进一步微调以允许订阅针对特定数据和链上帐户的证明。轻客户端可以利用这一点来保留最少的可验证数据，而无需运行完整节点或处理大量事务。

7 流水线、批处理和并行交易处理

为了最大限度地提高吞吐量，增加并发性，并降低工程复杂性，Aptos 区块链上的交易处理被分为不同的阶段。每个阶段都是完全独立的和单独的可并行化，类似于现代的超标量处理器架构。这不仅提供了显著的性能优势，而且使 Aptos 区块链能够提供验证者-客户互动的新模式。比如说。

- 当特定事务已包含在一批持久交易中时，可以通知客户端。持久且有效的交易极有可能立即提交。
- 当订购了一批持久交易时，可以通知客户。为了减少确定已执行交易输出的延迟，客户端可以选择在本地执行交易，而不是等待验证者远程完成执行。
- 客户端可以选择等待验证者执行经过验证的交易，并对经过验证的结果执行状态同步（例如，参见第 8 节）。

Aptos 模块化设计有助于加快开发速度并支持更快的发布周期，因为更改可以针对单个模块，而不是单个单体架构。同样，模块化设计还提供了将验证器扩展到单台机器之外的结构化路径，提供访问额外的计算、网络和存储资源。图 4 显示了整个事务的生命周期不同的处理阶段

7.1 批处理

批处理是一项重要的效率优化，是 Aptos 区块链中每个操作阶段的一部分。在交易传播过程中，每个验证者将交易分组为批次，在共识过程中将批次组合成块。执行、存储和账本认证阶段也分批工作，以提供重新排序、减少操作（例如，重复计算或签名验证）和并行执行的机会。

将交易分组可能会导致少量延迟，例如，在执行分发之前等待 200 毫秒来累积一批交易。但是，批处理在最长等待时间和最大批量大小方面很容易配置，使去中心化网络能够自动优化延迟和效率。批处理还允许有效的费用市场对交易进行优先排序，并避免来自过分热心的客户的 (DoS) 攻击。

7.2 持续的交易传播

根据 Narwhal Tusk[9] 的主要观点，Aptos 区块链的交易传播与共识脱钩。验证者不断地将成批的交易流向对方。同时利用所有可用的网络资源。由验证者 v 分发的每批交易都被持久化，并将批次摘要的签名发回给 v。在第 7.3 节中定义，任何 2f+1 个在批次摘要上的加权签名形成一个可用性证明 (PoAv)。可用性证明（PoAv）。这样的证明保证了至少有 f+1 个加权的诚实验证者已经存储了该批次，因此所有诚实的验证者都能在执行前检索到它。无限持续的交易批次可以打开一个 DoS 攻击的载体，

导致验证者用完存储并崩溃。为了防止这种情况，每批交易都有一个相关的时间戳。该批交易的时间戳允许在每个验证器上进行有效的垃圾回收。此外，还有一个单独的每个验证器配额机制被设计用来保护验证器不被耗尽空间即使在最极端的情况下，例如在潜在的拜占庭攻击下。批次也有大小限制，在同意坚持到稳定的存储之前就已经验证了。最后。一些优化以消除重复和缓存交易，减少存储成本，并确保执行与并行执行引擎的整合。

7.3 区块源数据排序

7.3.1 Blockchain time

7.4 Parallel transaction execution

7.4.1 Parallel data model

7.4.2 Parallel execution engine

7.5 Batch storage

7.6 Ledger certification

7.6.1 Ledger history certification

7.6.2 Periodic state certification

8 状态同步

9 社区的所有权

Aptos 区块链将由一个多元化的社区拥有、运营和管理，原生的 Aptos token 会被用在交易、网络费用、治理协议升级的投票、链上 (on-chain) 和链下 (off-chain) 的过程，并通过权益证明模型 (proof-of-stake model) 来提高区块链的安全性。Aptos token 经济学的完整说明将在未来的出版物中发布。

9.1 交易和网络的费用

所有 Aptos 的交易都有 gas 单位价格 (在 Aptos token 中指定)，这使得验证者可以优先处理网络中价值最高的交易。除此之外，在流水线模型的每个阶段，都有多个机会放弃低价值的交易 (这让区块链可以在系统容量大时有效地运行)。随着时间的推移，将部署网络费用，以确保使用 Aptos 区块链的成本与实际的硬件部署、维护和节点操作的成本成比例。此外，开发者将有机会在设计应用程序的时候在算力、存储和网络之间权衡不同的成本。

9.2 网络的治理

Aptos 区块链上的每一个重要特性的更改及改进都将经历这几个阶段: 提议、实现、测试和部署。这样的流程让有关人士及利益相关者能够有机会去提供反馈、探讨共同关心的问题和提出建议，最后一个阶段，即部署，通常会有两个步骤。首先会将带有新功能的软件版本部署到每个结点，然后这个功能将

被启用，例如，通过特定的标志或链上 (on-chain) 配置变量节点运营人员的每个软件部署必须是向后兼容的，以确保新的软件与支持的版本可以互操作。部署一个新的软件版本的过程的过程可能会持续多天，以将不同时区的运营人员和任何外部问题考虑到。一旦有足够数量的节点被升级，新功能的启用就可以由一个同步点触发。一旦有足够数量的节点升级，新功能的启用可以由一个同步点触发，比如一个商定的区块高度或纪元变化。在紧急条件下（例如，当停机时间不可避免时），可以通过节点运营人员的手动和强制改变来启用新功能。在最坏的情况下，在网络中形成硬分叉。与其他区块链相比，Aptos 区块链对其配置进行链上编码。每个验证者都能够去同步区块链的当前状态，并基于当前的链上的值自动选择正确的配置（例如，共识协议和 Aptos 框架版本）。由于这一功能，Aptos 区块链的升级是无缝和即时的。为了给启用过程提供灵活性和可配置性，Aptos 区块链将支持链上治理，代币持有者可以根据他们持有的代币权重进行投票。链上投票协议是公开的、可验证的，并且可以是即时的。随着时间的推移，链上治理可以部署在整个升级管理过程中。

举个例子：1. Token 持有者在链上就过渡到新的后量子签名方案进行投票 2. 开发人员实施和验证新的签名方案，并创建一个新的软件版本 3. 验证者升级他们的软件到新的版本 3. Token 持有者在链上就启用一个新的签名方案进行投票，链上的配置被更新，并且该变化生效。

作为一个开源项目，Aptos 区块链将依赖于强大的社区反馈，并使用链上治理来管理适当的流程。在某些情况下，可能仍然需要链外升级，但随着时间的推移，这种情况将减少。

9.3 权益证明共识

要参与 Aptos 区块链上的交易验证，验证者必须拥有最低要求的 Aptos 代币抵押。在交易传播过程中，抵押金额成比例影响 2f+1 抵押金额 PoAv，以及在区块元数据排序过程中的投票权重和领导者选择。验证者决定他们自己和他们各自的投保人之间的奖励分配。直压着们可以选择任意数量的验证者，将他们的代币押在其中，以获得预先商定的奖励分配。在每个纪元结束时，验证者和他们各自的抵押者将通过相关的链上 Move 模块收到他们的奖励。任何拥有足够抵押的验证者运营人员都可以自由加入 Aptos 区块链。所有的参数包括所需的最低股权，都可以由第 9.2 节中描述的链上启用程序来设置。第 9.2 节所述的链上启用程序来设置。

10 性能

如第 7 节所述，Aptos 区块链能够通过其并行、批量优化和模块化的交易处理管道实现最佳吞吐量和硬件效率。额外的性能举措，如共识升级、delta 写入、交易提示和关键路径缓存，将继续增加吞吐量并随着时间推移提高效率。今天，区块链的吞吐量通常以每秒交易量来衡量。然而，鉴于各种交易和基础设施的成本和复杂性差异很大，这样的方法比较系统是不精确。交易延迟也同样存在缺陷，因为在不同的实验中，提交到最后的起点和终点是不同的。此外，有些系统需要对交易的输入和输出有先验知识，并强制将逻辑交易分割成较小的、不太复杂的交易。分割交易会导致糟糕的用户体验，并人为地影响延迟和吞吐量，而不考虑开发者想要完成的任务。与此相反，Aptos 的方法是让开发者能够不受限制地自由构建，并根据实际使用情况测量吞吐量和延迟。自由构建，并根据真实世界的用例而不是合并交易来测量吞吐量和延迟。Aptos 区块链将继续优化单个验证者的性能，以及试验将更多验证者添加到网络的扩展技术。这两个方向都有明显的取舍。任何具有并行执行能力的区块链都可以通过更强大的硬件或将每个验证器构造成一个单独的机器集群支持额外的并发。然而，全局验证者的数量是有实际限制的，它

与验证者操作者的成本和复杂性是相称的。serverless 数据库在云服务中的兴起和流行，体现了很少有实体能够有效地部署和维护这些类型的复杂分布式系统。

10.1 同质化的状态分片

最初，Aptos 区块链将以一个单一的账本状态推出。随着时间的推移，Aptos 网络将采取一种独特的方法来实现横向可扩展性，同时仍然保持去中心化。这将通过多个分片账本状态来实现，每个分片都提供同质化的 API 和分片是作为一流的概念。Aptos 代币将用于所有分片上的交易费、押金和治理。数据可以通过同构桥在分片之间传输。用户和开发者可以根据自己的需要选择自己的分片方案。例如，开发人员可以提出一个新的分片，或在现有的分片内对用户进行集群，以实现分片内的高连接。此外，分片可能有不同的系统特性。一个分片可能是计算优化的固态硬盘 (SSD)，而另一个分片可能是为大型硬盘优化的，计算特性较低。通过在不同的分片之间提供硬件灵活性，开发者可以利用适当的系统特性来满足他们的应用。总之，同质状态分片提供了水平吞吐量扩展的潜力，允许开发人员在分片中使用单一的通用状态进行编程，并使钱包能够为其用户轻松纳入分片数据。这提供了显著的性能优势，以及单一统一的 Move 智能合约平台的简单性。

11 参考文献

[1] “Aptos-core,” 2022. [Online]. Available: https://github.com/aptos-labs/aptos-core

[2] “Move,” 2022. [Online]. Available: https://github.com/move-language/move

[3] D. Matsuoka, C. Dixon, E. Lazzarin, and R. Hackett. (2022) Introducing the 2022 state of crypto report. [Online]. Available: https://a16z.com/tag/state-of-crypto-2022/

[4] Z. Amsden, R. Arora, S. Bano, M. Baudet, S. Blackshear, A. Bothra, G. Cabrera, C. Catalini, K. Chalkias, E. Cheng, A. Ching, A. Chursin, G. Danezis, G. D. Giacomo, D. L. Dill, H. Ding, N. Doudchenko, V. Gao, Z. Gao, F. Garillot, M. Gorven, P. Hayes, J. M. Hou, Y. Hu, K. Hurley, K. Lewi, C. Li, Z. Li, D. Malkhi, S. Margulis, B. Maurer, P. Mohassel, L. de Naurois, V. Nikolaenko, T. Nowacki, O. Orlov, D. Perelman, A. Pott, B. Proctor, S. Qadeer, Rain, D. Russi, B. Schwab, S. Sezer, A. Sonnino, H. Venter, L. Wei, N. Wernerfelt, B. Williams, Q. Wu, X. Yan, T. Zakian, and R. Zhou, “The libra blockchain,” 2019. [Online]. Available: https://developers.diem.com/papers/the-diem-blockchain/2020-05-26.pdf

[5] S. Blackshear, E. Cheng, D. L. Dill, V. Gao, B. Maurer, T. Nowacki, A. Pott, S. Qadeer, D. R. Rain, S. Sezer, T. Zakian, and R. Zhou, “Move: A language with programmable resources,” 2019. [Online]. Available: https://developers.diem.com/papers/diem-move-a-language-with-programmable-resources/2019-06-18.pdf

[6] D. Dill, W. Grieskamp, J. Park, S. Qadeer, M. Xu, and E. Zhong, “Fast and reliable formal verification of smart contracts with the move prover,” in Tools and Algorithms for the Construction and Analysis of Systems, D. Fisman and G. Rosu, Eds. Cham: Springer International Publishing, 2022, pp. 183–200.

[7] N. Popper. (2021) Lost passwords lock millionaires out of their bitcoin fortunes. [Online]. Available: https://www.nytimes.com/2021/01/12/technology/bitcoin- passwords- wallets- fortunes.html

[8] The Diem Team, "State synchronization and verification of committed information in a system with reconfigurations,"2020. [Online]. Available: https://github.com/aptos-labs/aptos- core/blob/main/documentation papers/lbft-verification/lbft-verification.pdf

[9] G. Danezis, L. Kokoris-Kogias, A. Sonnino, and A. Spiegelman, "Narwhal and tusk: A dag-based mempool and efficient bft consensus," in Proceedings of the Seventeenth European Conference on Computer Systems, ser. EuroSys ' 22. New York, NY, USA: Association for Computing Machinery, 2022, p. 34–50. [Online]. Available: https://doi.org/10.1145/3492321.3519594

[10] The Diem Team, "Diembft v4: State machine replication in the diem blockchain," 2021. [Online]. Available: https://developers.diem.com/papers/diem-consensus-state-machine-replication-in-the-diem- blockchain/2021-08-17.pdf

[11] S. Cohen, R. Gelashvili, L. Kokoris-Kogias, Z. Li, D. Malkhi, A. Sonnino, and A. Spiegelman, "Be aware of your leaders,"CoRR, vol. abs/2110.00960, 2021. [Online]. Available: https://arxiv.org/abs/2110.00960

[12] A. Spiegelman, N. Giridharan, A. Sonnino, and L. Kokoris-Kogias, "Bullshark: Dag bft protocols made practical," in Proceedings of the 20th Conference on Computer and Communications Security (CCS), ser. CCS ' 22. Los Angeles, CA, USA: Association for Computing Machinery, 2022.

[13] R. Gelashvili, A. Spiegelman, Z. Xiang, G. Danezis, Z. Li, Y. Xia, R. Zhou, and D. Malkhi, "Block-stm: Scaling blockchain execution by turning ordering curse to a performance blessing," 2022. [Online]. Available: https://arxiv.org/abs/2203.06871

[14] J. Lind, "The evolution of state sync: The path to 100k+ transactions per second with sub-second latency at aptos," 2022. [Online]. Available: https://medium.com/aptoslabs/52e25a2c6f10