Move双子星：Sui与Aptos如何挑战区块链格局

作者：YBB Capital Researcher Zeke

## 前言

近期市场愈发冷淡，圈内不少OG也开始对行业存在的意义产生动摇。说点我个人的感想，我始终觉得过去许多伟大的愿景之所以被“证伪”，是因为这些愿景从一开始就不能逻辑自洽。在非金融应用之外的Dapp，总是试图以强调去中心化价值的方式来掩盖产品本身并不够优秀。但事实是，他们让我不要相信谷歌、推特、YouTube，转而相信他们的多签钱包和单机服务器足够安全。很多愿景并不是被证伪，而是从来没有被真正的验证过。我依旧相信大部分愿景就算没那么宏伟，也都有其意义，它们也许只是需要足够的底层去支撑。最终在去中心化以及媲美Web2的良好体验上，至少可以提供一者。就像从前TON与Solana同样被人所轻视，但如今在多方面的表现上正在逐步追上老大哥，承载应用的公链需要创新，这在每个周期都会推动行业进步。所以，我们在今天将要探寻一个长期以来一直没被重视的公链类型，Move系。

## 一、Move

Move 编程语言最初诞生于 Meta 的遗弃项目 Diem（最初称为 Libra），该项目最初旨在创建一个更稳定、更受监管的稳定币，作为其元宇宙愿景的基底。 然而，事与愿违的是，该项目遭到了来自全球监管机构的强烈反对和持续的压力。 监管机构担心 Diem 的规模和 Facebook 庞大的用户群可能会对金融稳定、货币政策和数据隐私构成威胁，拜登政府牵头的施压，最终导致 Meta 不得不放弃 Diem 项目。

但值得庆幸的是，Diem的核心并没有被遗弃，从原团队中分拆出的不同派系依旧在坚持对Move的挖掘与开发，到如今已演变为我们所熟知的Move双子星Sui、Aptos。除此之外，还有尚在襁褓之中的Linera（借鉴Move的Rust公链），近期在大力宣发的Movement等多个公链项目。

那么为何一个腰斩项目的余温能有如此之大呢？Move作为Web2头部大厂对于区块链编程语言的一份答卷，其功底不必多说，在设计上也围绕现有区块链编程语言（尤其是Solidity）做了许多性能与安全问题上的反思，其设计目标是为资产管理和访问控制执行量身定制一个类型系统。我个人总结为简单三点：

安全性: Move 语言设计的大前提在于安全性，它使用静态类型检查和资源管理来防止常见的安全漏洞，例如溢出错误和重入攻击。对比于其他语言虚拟机，Move支持多种安全功能，下图引用Nansen的对比。



可组合性: 支持模块化和可组合性，允许开发人员轻松地创建和组合不同的智能合约，从而构建更复杂的应用程序。

性能: Move 语言的虚拟机经过优化（支持并行、内存管理、编译器优化），可以高效地执行智能合约，从而提高交易速度和吞吐量。

在模块化EVM公链充斥市场的当下，Move其实算一种勇敢的尝试。我所说的上诉三点，你可能也在很多公链项目的介绍里看过类似的，我更建议亲身去体验才能具像化这些文字。

## 二、Sui



### 2.1 架构

作为双子星之一的Sui，在上线初期一直因空投问题和代币释放方式而饱受诟病。但抛开这些问题，只说项目本身。Sui至少在性能和体验方面是足够优秀的，并且在游戏上的表现极为出色，这都离不开其自身为主流采用所进行改良的架构设计。在这里我先简单描述一下Sui在架构上的创新：

对象存储模型：该组件是Sui对Move改良后的核心组件，所谓对象储存模型是将数据存储为独立的对象，每个对象都有唯一的标识符。与传统的数据库系统不同，对象存储模型没有固定的数据结构，可以存储各种类型的数据，例如文本、图像、视频、音频。这种模型允许并行执行和水平扩展（添加节点来扩展存储容量），Sui在设计上是围绕着该模型展开的。

因果排序： 确保交易的执行顺序符合因果关系，避免出现数据冲突和不一致性。这使得 Sui 能够处理大量并发交易，并保持数据的一致性。

Narwhal 和 Bullshark 共识引擎： Sui 使用 Narwhal 和 Bullshark 作为其共识引擎，Narwhal 负责交易排序和验证，其工作原理是通过维护本地交易池，根据交易的因果关系进行排序并广播，确保所有节点都拥有相同且有效的交易顺序。而Bullshark在收到Narwhal排序后的交易列表时会对交易列表进行投票，并使用拜占庭容错共识来确保所有节点都对交易列表达成一致。

Sui Move： Sui 在 Move 语言的基础上进行了扩展，增加了新的功能，例如支持 NFT、资产管理和数据存储。

Sui 框架： Sui 提供了一套完整的框架，帮助开发者快速构建和部署应用程序。该框架包括各种工具和库，例如 Sui 钱包、Sui SDK 和 Sui CLI 。

Sui 的架构设计使其能够处理大量并发交易，并保持高速度、低费用和安全性。同时，Sui 的 Move 语言和 Sui 框架也为开发者提供了强大的工具，帮助他们构建安全、可扩展和用户友好的应用程序。

### 2.2 共识

Sui 区块链使用一种名为 Mysticeti 的共识机制，它是一种基于拜占庭容错 (BFT) 的共识，旨在优化低延迟和高吞吐量。

Mysticeti 允许多个验证器并行提出区块，从而充分利用网络带宽并提供审查阻力。此外，该协议仅需三轮消息即可从 DAG （有向无环图）中提交区块，与 pBFT 相同，并匹配理论最小值。提交规则允许并行投票和认证区块领导者，从而进一步减少中位数和尾部延迟。提交规则还可以容忍不可用的领导者，而不会显著增加提交延迟。

Mysticeti 在 Sui 主网上线之前，已经在测试网上运行了三个月，其显著成果包括延迟降低了 80%。现在，Sui 网络可以每秒处理数万笔交易，端到端延迟远低于一秒。

Sui 区块链还使用一种称为委托权益证明 (DPoS) 的特定类型的权益证明共识。当涉及共享对象的交易（称为复杂交易）发生时，Sui会采用上文所诉的Narwhal & Bullshark共识引擎进行交易排序。与其他BFT共识机制的公链相比，Sui的优劣势可总结为六点：

优势：

低延迟和高吞吐量： Mysticeti 协议通过并行区块提议和优化消息传递流程，显著降低了共识延迟，并提升了网络吞吐量。 这使得 Sui 区块链能够处理每秒数万笔交易，端到端延迟远低于一秒；

审查阻力： Mysticeti 协议允许多个验证器并行提出区块，从而提高了网络的审查阻力；

容忍不可用的领导者： 提交规则允许容忍不可用的领导者（ 当领导者节点出现故障时，系统会自动选举一个新的领导者来接替其职责），而不会显着增加提交延迟。

劣势：

复杂性： Mysticeti 协议的设计相对复杂，需要更深入的技术理解才能完全掌握其运作机制；

安全性： 虽然 Mysticeti 协议在测试网上表现出色，但其安全性仍需在实际应用中得到进一步验证；

可扩展性： Mysticeti 协议的扩展性仍需进一步观察，以确保其能够适应未来不断增长的网络规模和交易量。

### 2.3抽象账户

Sui的抽象账户模型（Account Abstraction）是一种允许用户以更简单、更安全的方式管理其账户和交易的机制。它通过将账户和交易逻辑从底层区块链协议中抽象出来，实现了更高层次的账户管理和交易处理。

在Sui的抽象账户模型中，账户不再是简单的公钥-私钥对，而是具有更丰富的属性和行为的对象。每个账户都有一个唯一的标识符，称为账户ID，该ID与账户的公钥和私钥对相关联。

Sui的抽象账户模型包括以下几个关键组件：

账户对象（Account Object）：账户对象是Sui中账户的基本单位。每个账户对象都有一个唯一的账户ID，并包含账户的属性和行为；

账户数据（Account Data）：账户数据是账户对象的核心组成部分。它包含账户的基本信息，例如账户ID、公钥和私钥对等；

交易上下文（Transaction Context）：交易上下文是Sui中交易的基本单位。它包含交易的相关信息，例如交易ID、账户ID和交易数据等；

账户逻辑（Account Logic）：账户逻辑是Sui中账户的行为和规则的集合。它定义了账户如何处理交易和管理其状态。

Sui的抽象账户模型通过以下几个步骤来处理交易：

交易创建：用户创建一个交易，并将其发送到Sui网络；

交易验证：Sui网络验证交易的有效性和完整性；

账户查找：Sui网络根据交易中的账户ID查找对应的账户对象；

账户逻辑执行：Sui网络执行账户逻辑来处理交易和更新账户状态；

交易确认：Sui网络确认交易的结果，并将其写入区块链。

简单来说，Sui的抽象账户模型是一种创新机制，它简化了账户管理和交易处理，使应用更像应用。

### 2.4 游戏

一个公链能否突出重围，首先要沉淀与积累，之所以在上文中把Move称为一次勇敢的尝试，有两点原因：一是模块化概念泛化的时代下原生Move系（即Move双子星）算是对Layer1最后的尝试了，基本属于逆势而行，但近期多条异构链的崛起，也许在证明模块化不是唯一的答案。二是重制一条公链并采用新的编程语言这种举动，你可以把它想象为在当前的手机市场中想要重制一个系统去挑战iOS和安卓，未来的道路必定充满荆棘。Move系在未来几年是否能像Solana一样发光、发热，自身选择的发展方向将尤为关键。Sui对于这个问题的答案是，游戏。

游戏是Web3的重要入口之一，但绝大部分公链对游戏的支撑并不好，这是因为区块链自出生起基本就是围绕金融设计的，又因去中心化的构造性能低下，所以天生不适合游戏。不过Sui并不一样，它的模型既适合DeFi应用也适合非金融应用与游戏。正如上文所诉在Sui中，一切都是对象。游戏或是应用程序具有层次关系的复杂资产，在 Sui 上，对象可以拥有其他对象（资产可以拥有资产）。假设您正在玩一个英雄角色的游戏，并且该英雄角色有一个库存，它还有属于该角色的其他数字资产。Sui 可以以其他区块链无法做到的方式准确地对这些数据层次结构进行建模。因此，它使开发人员有机会表达他们想要构建的应用程序，而无需解决链的基本限制。

除此之外Sui还在积极开展与传统Web2巨头的合作，从去年与四大韩国游戏巨头中的三家（Netmarble、NHN和NCSoft）达成合作伙伴关系。到今年和Tiktok合作开发链游与SocialFi项目，Sui正在将传统巨头引入Web2。

## 三、Aptos



Aptos 作为另一个基于 Move 语言的 Layer 1 区块链，同样致力于构建高性能、可扩展的 Web3 基础设施。其架构设计与 Sui 有着许多相似之处，但也展现出一些独特的特点。

### 3.1 架构

1.模块化设计： Aptos采用模块化架构，允许开发者独立开发和升级不同模块，从而提高开发速度和灵活性；

2.并行执行引擎 (Block-STM)： 与其他需要预先声明数据依赖关系的区块链不同，Aptos 的并行执行引擎无需事先了解数据位置即可并行处理交易，从而提高吞吐量并降低延迟；

3.流水线交易处理： Aptos将交易处理分为传播、元数据排序、批处理存储等多个阶段，并通过流水线方式并行执行这些阶段，从而最大化吞吐量并降低延迟；

4.Move 编程语言： Aptos 使用 Move 编程语言，相比于Sui引入的创新，Aptos所做的更多是将其完善。比如规范语言、引入更强大的函数支持、自定义能力；

5.灵活的状态同步： 允许节点选择不同的状态同步策略，例如同步完整历史记录或仅同步最新状态，从而提高节点的灵活性；

6.AptosBFT 共识机制： AptosBFT 是 Aptos 使用的拜占庭容错共识机制，它通过优化验证者之间的通信和同步来提高吞吐量并降低延迟。与Sui相比只能算DiemBFT 的改进版本，在效率和抗崩溃恢复上做了一定改良，所以在这里只简单说明一下。

Aptos 的架构设计使其能够处理大量并发交易，并保持高速度、低费用和安全性。同时，Aptos 的 Move 语言和 Aptos 框架也为开发者提供了强大的工具，帮助他们构建安全、可扩展和用户友好的应用程序。

### 3.2 Block-STM

我们在这里扩展说一下Aptos的核心创新并行执行引擎Block-STM：

Block-STM 的核心原理：

预设顺序执行: Block-STM 依赖于区块中交易的预设顺序，所有交易必须按照这个顺序执行才能保证最终状态的一致性；

乐观并发控制: Block-STM 会乐观地并行执行交易，假设不会发生冲突。乐观并发控制基于“冲突少见”的假设，允许事务在不加锁的情况下访问和修改数据。它认为多个事务同时发生冲突的概率很低，因此可以先进行修改，并在最后提交前检查是否真的发生了冲突；

多版本数据结构: 为了支持乐观并发控制，Block-STM 使用多版本数据结构来存储数据。每个写入操作都会创建一个新的数据版本，而读取操作则会访问对应版本的數據；

验证和重试: 在执行完一个交易后，Block-STM 会验证其读取的数据版本是否仍然有效。如果验证失败，说明发生了冲突，该交易会被标记为无效并重新执行；

协作调度: Block-STM 使用一个协作调度器来协调各个线程的执行和验证任务，以最大化并行度。

Block-STM 的工作流程:

交易分组: 将区块中的交易分组，分配给不同的线程并行执行；

乐观执行: 每个线程乐观地执行分配给它的交易，并记录每个交易的读写集；

验证: 当一个线程完成一个交易的执行后，会验证其读取集中的数据版本是否仍然有效；

重试: 如果验证失败，说明发生了冲突，该交易会被标记为无效并重新执行；

提交: 所有交易都验证通过后，将结果写入区块链状态，完成交易的提交。

Block-STM 的优势:

高吞吐量: 通过乐观并发控制和协作调度，Block-STM 可以充分利用多核处理器的性能，实现高吞吐量；

低延迟: 由于交易可以并行执行，Block-STM 可以显著降低交易的确认时间；

安全性: Block-STM 的预设顺序执行和验证机制保证了最终状态的一致性和安全性。

简单来说，Block-STM 是一种高效的并行交易执行引擎，它结合了乐观并发控制、多版本数据结构和协作调度等技术，在保证安全性和正确性的前提下，最大化了区块链的吞吐量。

### 3.3 抽象账户

不同于Sui的抽象账户那么直接，Aptos所支持的抽象维度更有限一些，也没有具体的预定义标准，其抽象账户能力主要体现在以下几个方面：

模块化账户管理： 使用 Move 模块来定义和管理账户，开发者可以创建自定义模块来实现不同的账户类型和功能。

灵活的密钥管理：允许用户使用不同的密钥对账户进行不同的操作，例如使用一个密钥进行交易签名，使用另一个密钥进行账户管理。

可编程的交易验证：开发者可以在 Move 模块中定义自定义的交易验证逻辑，例如多签、限额等，以满足不同的应用场景。

### 3.4 与微软的合作

相比于Sui比较侧重游戏的发展，Aptos并没有特定的发展目标，其口号为最适合生产的区块链。比较值得一提的是Aptos目前正在与微软合作，旨在将微软的AI技术引入区块链中。当前两者的第一款合作产品Aptos Assistant已经在官方页面上线，该产品是Aptos网络所构建的生成式AI助手，而后续的AI产品还将在几个月后陆续公布。

## 四、Move系

虽然近期Sui的表现良好，但相较于EVM系及Solana、Ton等异构链来说。Move的崛起还需要一定时间的沉淀，当前双子星Sui和Aptos虽然顶着明星光环，也确实在技术上有所突破，但Move生态的整体规模和活跃度仍不及其他成熟生态。开发者数量、应用种类和用户规模都需要时间积累。从外部合作到运营来看，两者又有比较浓厚的Web2思维，缺乏一些Web3基因，各种合作项目在圈内一直属于不温不火的状态。

但就Move系的潜力而言，在多方面都是值得深挖的，也有一些开发者也已经注意到Move未来的价值。正如前言所诉，目前也已经有将Move引入ETH Layer2的项目出现，未来的Move系也将会在ETH二层生态中发光发热，当前要做的更多是如何将Move系打出去。