Uniswap v4 核心 [草稿]*

June 2023

Hayden Adams
hayden@uniswap.org

Moody Salem
moody.salem@gmail.com

Noah Zinsmeister
noah@uniswap.org

Sara Reynolds
sara@uniswap.org

Austin Adams
austin@uniswap.org

Will Pote
pote@uniswap.org

Mark Toda
mark@uniswap.org

Alice Henshaw
alice@uniswap.org

Emily Williams
emily@uniswap.org

Dan Robinson
dan@paradigm.xyz

摘要

Uniswap v4 是一种实现在以太坊虚拟机上的非托管自动做市商。通过任意代码 Hooks，Uniswap v4 提供了可自定义性，允许开发人员在 Uniswap v3 中引入的集中流动性模型中增加新功能。在 Uniswap v4 中，任何人都可以创建一个带有指定 Hooks 的新池，该 Hooks 可以在预定的池操作之前或之后运行。Hooks 可用于实现之前内置于协议中的功能，如预言机，以及以前需要在协议层面独立实现的新功能。Uniswap v4 还通过单例模式 (singleton)、闪电记账 (flash accounting) 和对原生 ETH 的支持，提高了 gas 效率和开发者体验。

1 介绍

Uniswap v4 是一种实现在以太坊虚拟机（EVM）上的自动做市商（AMM），用于实现价值的高效交换。与 Uniswap Protocol 的先前版本一样，Uniswap v4 是非托管、不可升级和无需许可的。Uniswap v4 的重点是通过可定制性和 gas 效率升级的体系结构变化，在 Uniswap v1 和 v2 中构建的 AMM 模型和 Uniswap v3 中引入的集中流动性模型的基础上进行改进。

Uniswap v1 [1] 和 v2 [2] 是 Uniswap Protocol 的前两个版本，分别实现了 ERC-20 <> ETH 和 ERC-20 <> ERC-20 的交易，两者都使用了常量乘积做市商（CPMM）模型。Uniswap v3 [3] 通过使用在有限价格范围内提供流动性的仓位，提供了更高资本效率的流动性，并引入了多个费用层级。

尽管集中流动性和费用层级增加了流动性提供者的灵活性，并允许实施新的策略，但 Uniswap v3 不足以支持随着 AMM 和市场的发展而出现的新功能。

一些功能，例如最初在 Uniswap v2 中引入的价格预言机，允许集成者利用去中心化的链上定价数据，但这以增加交易者的 gas 成本为代价，并且对集成者而言不具有可定制性。其他增强功能的想法，包括通过时间加权平均价格做市商（TWAMM）[6] 实现的时间加权平均价格订单（TWAP）、波动率预言机、限价订单或动态费用，需要重新实现核心协议，无法由第三方开发者添加到 Uniswap v3 中。

*Uniswap v4 核心白皮书的中文翻译由 WTF Academy 贡献。

此外，在以前的 Uniswap 版本中，部署新的池需要部署新的合约，其成本随着合约字节码的大小而增加。另外，涉及到与多个 Uniswap 池的交易涉及到多个合约之间的转账和冗余状态更新。自 Uniswap v2 以来，Uniswap 要求 ETH 被包装成 ERC-20，而不是支持原生 ETH。这些都带来了 gas 成本。

在 Uniswap v4 中，我们通过一些值得注意的功能改进了这一点：

- *Hooks*：Uniswap v4 允许任何人使用自定义功能部署新的集中流动性池。对于每个池，创建者可以定义一个 “Hooks 合约”，该合约在调用的生命周期的关键点执行逻辑。这些 Hooks 也可以管理池的交换费用以及向流动性提供者收取的提款费用。
- *单例模式*：Uniswap v4 摒弃了先前版本中使用的工厂模型，而是实现了一个包含所有池的单个合约。单例模式降低了池的创建成本和多跳交易（multi-hop trade）的成本。
- *闪电记账*：单例使用 “闪电记账” 机制，要求在锁定结束时池或调用者没有代币欠款。在调用过程中，代币可以用于单例内外的任意数量的操作。通过 EIP-1153 [4] 中提议的瞬态存储操作码，这种功能将变得高效。闪电记账进一步降低了跨多个池的交易的 gas 成本，并支持与 Uniswap v4 的更复杂集成。
- *原生 ETH*：Uniswap v4 恢复了对原生 ETH 的支持，并支持在 v4 池中使用原生代币进行配对。ETH 交换者和流动性提供者从转账成本更低和去除额外包装成本中受益。

以下章节详细解释了这些变化和使它们成为可能的体系结构变化。

2 HOOKS

Hooks 是在池的执行过程中在指定点执行一些开发者定义的逻辑的外部部署合约。这些 Hooks 允许集成者创建具有灵活和可定制执行的集中流动性池。

Hooks 可以修改池的参数，或添加新特性和功能。可以使用 Hooks 实现的示例功能包括：

- 在一段时间通过 TWAMM 执行大型订单
- 按指定价格成交的链上限价订单
- 随波动率变化的动态费用
- 流动性提供者的内部化 MEV 分配机制
- 实现中位数、截断或其他自定义预言机

我们预计在未来为特定的 Hooks 设计撰写独立的白皮书，因为许多 Hooks 的复杂性与协议本身一样。

2.1 操作 Hooks

当有人在 Uniswap v4 上创建一个池时，他们可以指定一个 Hooks 合约。该 Hooks 合约实现了在池的执行过程中池将调用的自定义逻辑。Uniswap v4 目前支持八个此类 Hooks 回调：

- beforeInitialize/afterInitialize
- beforeModifyPosition/afterModifyPosition
- beforeSwap/afterSwap
- beforeDonate/afterDonate

Hooks 合约的地址决定了哪些 Hooks 回调会被执行。这一方法高效且表达力强，并确保即使是可升级的挂钩也遵守某些不变量。创建有效 Hooks 时有最低准则需要遵守。在图 1 中，我们描述了 beforeSwap 和 afterSwapHooks 在交换执行流程部分的工作原理。

2.2 Hooks 管理费用

Uniswap v4 允许对交换和提取流动性收取费用。

交换费用可以是静态的，也可以由 Hooks 合约动态管理。Hooks 合约还可以选择将一定比例的交换费用分配给自己。提款费用不能在池中进行本地设置。要设置提款费用，池创建者必须设置一个 Hooks 合约来管理该费用，并且，收取的提款费用会支付给 Hooks 合约。支付给 Hooks 合约的费用可以由 Hooks 合约的代码任意分配，包括支付给流动性提供者、交换者、Hooks 创建者或任何其他方。

Hooks 的功能受创建池时选择的不可变标志（flag）的限制。池创建者可以选择的费用设置有：

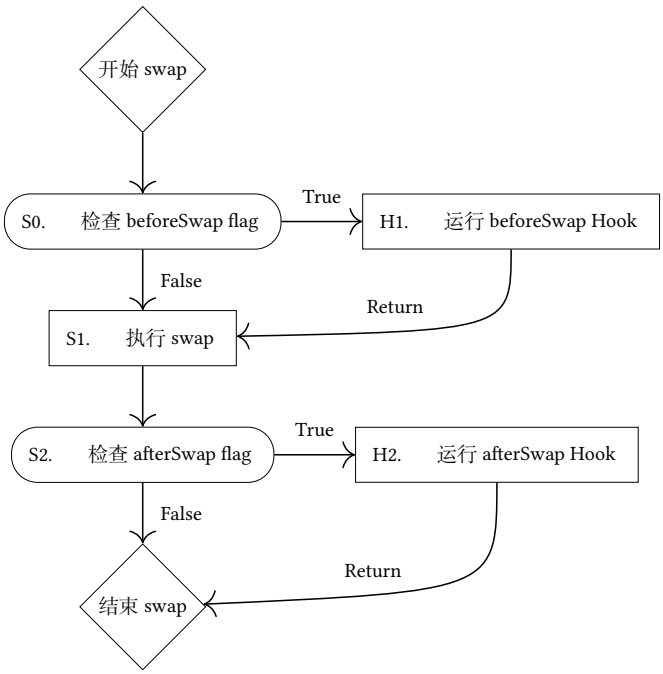

图 1: Swap Hook 流程图

- 池收取静态费用（以及该费用是多少）还是动态费用
- Hooks 是否具有收取交换费用的权限
- Hooks 是否具有收取提款费用的权限

治理可以从交换费用或提款费用中收取一定费用，如下面的治理部分所讨论的。

3 单例和闪电记账

Uniswap Protocol 的先前版本使用工厂/池模式，其中工厂为新的代币对创建单独的合约。Uniswap v4 使用单例设计模式，所有池都由单个合约管理，使得池的部署成本降低约 99%。

单例设计与 v4 中的另一个体系结构变化闪电记账相辅相成。在 Uniswap Protocol 的先前版本中，每个操作（例如交换或向池中添加流动性）都以代币转移结束。在 v4 中，每个操作都会更新一个称为`delta`的内部净余额，仅在锁定结束时进行外部转账。新的`take()`和`settle()`函数分别用于从池中借资金和存入资金到池中。通过要求池或调用者没有代币欠款，确保了池的偿付能力。

闪电记账简化了复杂的池操作，例如原子交换和原子添加。与单例模式结合使用时，它还简化了多跳交易。

在当前的执行环境中，闪电记账架构是昂贵的，因为它要在每次余额变化时进行存储更新。尽管合约保证了内部会计数据实际上从未序列化到存储中，但当超过存储退款上限，用户仍然需要支付费用 [5]。但是，由于余额必须在事务结束时为 0，因此可以使用瞬态存储实现对这些余额的记账，正如 EIP-1153 [4] 中所描绘的。

单例和闪电记账使得在多个 v4 池之间更高效地进行路由成为可能，降低了流动性碎片化的成本。引入 Hooks 将大大增加池的数量，这个特性会非常有用。

4 原生 ETH

Uniswap v4 将原生 ETH 带回交易对中。虽然 Uniswap v1 严格将 ETH 与 ERC-20 代币配对，但由于实施复杂性和在 WETH 和 ETH 配对之间的流动性碎片化的担忧，Uniswap v2 中删除了原生 ETH 配对。单例和闪

电记账减轻了这些问题，因此 Uniswap v4 允许同时支持 WETH 和 ETH 配对。

原生 ETH 转账的 gas 成本约为 ERC-20 转账的一半（ETH 为 21k gas，ERC-20 约为 40k gas）。目前，Uniswap v2 和 v3 要求绝大多数用户在在 Uniswap Protocol 上交易之前（之后）将他们的 ETH 包装（解包装）为 WETH，这需要额外的 gas。

5 其他值得注意的功能

5.1 ERC1155 记账

Uniswap v4 将支持通过单例实现的 ERC-1155 代币的铸造/销毁，用于额外的代币记账。用户现在可以将代币保留在单例合约中，避免 ERC-20 转入或转出合约的。这一点对于频繁交换者或流动性提供者非常有价值，因为它们会在多个区块或交易中连续使用相同的代币。

5.2 治理更新

Uniswap v4 具有两种单独的治理费用机制，交换费用和提款费用，有着不同的机制。首先，与 Uniswap v3 类似，治理可以选择在特定池上获取特定百分比的交换费用。对于 v4，如果 Hooks 最初选择为池打开提款费用，治理还可以获取特定百分比的提款费用。与 Uniswap v3 不同，治理不控制可允许的费用层级或价格刻度间距。

5.3 Gas 减少

正如上面所讨论的，Uniswap v4 通过闪电记账、单例模式和对原生 ETH 的支持引入了有意义的 gas 优化。此外，引入了 Hooks 使得协议内嵌的价格预言机（在 Uniswap v2 和 Uniswap v3 中包含）变得不再必要，这也意味着一些池可以完全放弃预言机，并在每个区块中的第一次池交换中节省约 15k gas。

5.4 donate()

`donate()`允许用户、集成者和 Hooks 直接支付给特定范围内的流动性提供者，支付的方式可以是池中的任意一种或两种代币。此功能依赖于费用记账系统以实现高效支付，而费用支付系统仅支持池中的代币。可能的用例包括在 TWAMM 订单上给范围内的流动性提供者打赏或新类型的费用系统。

6 总结

总之，Uniswap v4 是一个非托管、不可升级且无需许可的 AMM 协议。它基于 Uniswap v3 中引入的集中流动性模型，通过 Hooks 实现了可定制的池。与 Hooks 相辅相成的还有其他体系结构变化，如单例合约，它将所有的池状态保存在一个合约中，以及闪电记账，它有效地确保池的偿付能力。其他改进包括对原生 ETH 的支持，ERC-1155 余额记账，新的费用机制以及向范围内流动性提供者捐赠的能力。

REFERENCES

[1] Hayden Adams. 2018. *Uniswap v1 Core.* Retrieved Jun 12, 2023 from https://hackmd.io/@HaydenAdams/HJ9jLsfTz

[2] Hayden Adams, Noah Zinsmeister, and Dan Robinson. 2020. *Uniswap v2 Core.* Retrieved Jun 12, 2023 from https://uniswap.org/whitepaper.pdf

[3] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. 2021. *Uniswap v3 Core.* Retrieved Jun 12, 2023 from https://uniswap.org/whitepaper-v3.pdf

[4] Alexey Akhunov and Moody Salem. 2018. *EIP-1153: Transient storage opcodes.* Retrieved Jun 12, 2023 from https://eips.ethereum.org/EIPS/eip-1153

[5] Vitalik Buterin and Martin Swende. 2021. *EIP-3529: Reduction in refunds.* Retrieved Jun 12, 2023 from https://eips.ethereum.org/EIPS/eip-3529

[6] Dave White, Dan Robinson, and Hayden Adams. 2021. *TWAMM.* Retrieved Jun 12, 2023 from https://www.paradigm.xyz/2021/07/twamm

免责声明

本文仅供一般信息目的。它不构成投资建议或购买或销售任何投资的推荐或招揽，也不应用于评估做出任何投资决策的价值。不应依赖本文提供会计、法律或税务建议或投资建议。本文反映了作者当前的观点，并不代表 Uniswap Labs、Paradigm 或其关联公司的观点，也不一定反映 Uniswap Labs、Paradigm、其关联公司

或与之相关的个人的观点。所反映的观点可能会随时更改，而无需进行更新。