Swap Pallet

Parami Devs devs@parami.io

March 11, 2022

Abstract

We implement swap pallet for assets pricing and trading using the constant product market maker model (aka, $x \times y = k$ model)[1], and additional farming algorithm to encourage investors to add liquidity into the pool.

I. Introduction

Our swap algorithm is following the design of Uniswap v1[2], the swap pallet provides an interface for seamless exchange of assets on Parami blockchain. By eliminating unnecessary forms of rent extraction and middlemen it allows faster, more efficient exchange. Where it makes tradeoffs, decentralization, censorship resistance, and security are prioritized.

In additional, the swap pallet provide the ability to farm assets with a farming curve. To do so, inspired by Uniswap v3, the swap pallet use NFT for liquidity provider tokens.

II. Goals

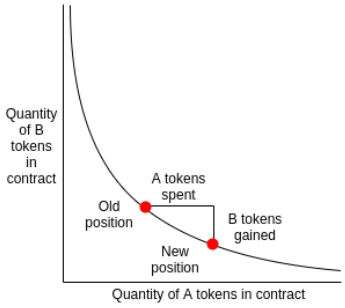
With the design of Parami Protocol, the goals of the swap pallet are:

- Provide pricing for each KOL NFT fragment / DAO coin
- Join liquidity polls to collect fess on pairs
- Liquidity-sensitive automated pricing
- Liquidity-based farming of assets
- Low fee cost

III. Implementation

i. Pricing

To achieve the goals, the swap pallet implementated constant product formula[3] for Liquidity-sensitive automated pricing.



Following the $x \times y = k$ model, when sell Δx tokens, the user will get Δy tokens, such that $x \times y = (x + \Delta x) \times (y - \Delta y)$, thus the price $(\Delta x / \Delta y)$ is the function of x/y.

With a fee $\rho = 0.003$, the token reserves are updated as follows:

$$x'_\rho = x + \Delta x = (1 + \alpha)x = \frac{1 + \beta(\frac{1}{\gamma} - 1))}{1 - \beta}x,$$

$$y'_\rho = y - \Delta y = \frac{1}{1 + \alpha\gamma}y = (1 - \beta)y$$

where $\alpha = \frac{\Delta x}{x}$, $\beta = \frac{\Delta y}{y}$, $\gamma = 1 - \rho$.

When get input price, with 0.3% fee, the price is implemented as follows:

$$(997 * \Delta x * y) / (1000 * x + 997 * \Delta x)$$

When get output price, with the same fee, the price is implemented as follows:

$$(1000 * x * \Delta y) / (997 * (y - \Delta y)) + 1$$

ii. Liquidity

An investor can mint liquidity by depositing both AD3 and asset.

When add liquidity, it takes as input $\Delta e > 0$ and updates the state as follows:

$$(e, t, l) \xrightarrow{\Delta e} (e', t', l')$$

where

$$\begin{aligned} e'' &= e + \Delta e && = (1+\alpha)e, \\ t'' &= t + \left\lceil \frac{\Delta e \times t}{e} \right\rceil + 1 && = (1+\alpha)t + 1, \\ l'' &= l + \left\lceil \frac{\Delta e \times l}{e} \right\rceil && = (1+\alpha)l, \\ \alpha &= \frac{\Delta e}{e} \end{aligned}$$

When burn liquidity, it takes as input $0 < \Delta l < l$ and updates the state as follows:

$$(e, t, l) \xrightarrow{\Delta l} (e', t', l')$$

where

$$\begin{aligned} e'' &= e - \left\lceil \frac{\Delta l \times e}{l} \right\rceil && = (1-\alpha)e, \\ t'' &= t - \left\lceil \frac{\Delta l \times t}{l} \right\rceil && = (1-\alpha)t, \\ l'' &= l - \Delta l && = (1-\alpha)l, \\ \alpha &= \frac{\Delta l}{l} \end{aligned}$$

iii. Farming

Additionally, an investor can earn from liquidity, via the farming algorithm.

In Parami Protocol, an asset has a maximum supple of 10,000,000 tokens, while only 3,000,000 tokens were minted at initial. The rest 7,000,000 tokens will be minted via farming in 3 years, on Parami Protocol, there're $P = 3 * 365.25 * (60000/12000 * 60 * 24)$ blocks.

The farming curve follows $f(x) = kx + b$ model.

$$\begin{aligned} &\int_l^u kx + b \,\mathrm{d}\, x \\ &\int kx + b \,\mathrm{d}\, x = F(x) = \frac{k}{2}x^2 + bx \end{aligned}$$

By design, we will mint 100 tokens in the first block, while the decimal digit is 18 on Parami Protocol, $B = 100,000,000,000,000,000,000,000$, $R = 7,000,000,000,000,000,000,000,000$.

So that

$$\begin{aligned} \int_0^P f(x) \,\mathrm{d}\, x &= R \\ \int_0^P f(x) \,\mathrm{d}\, x &= F(P) - F(0) \\ &= \frac{k}{2}P^2 + B * P, \\ \frac{k}{2} &\approx 12562772015768 \end{aligned}$$

Inspired by Uniswap v3[4], with the liquidity token minted time l stored in the NFT, we can calculate the reward by:

$$\int_l^u f(x) \,\mathrm{d}\, x = F(u) - F(l)$$

Each time an investor claim the reward, the pallet updates the state as follows on current block number h:

$$(r, l) \xrightarrow{h} (r', l')$$

where

$$\begin{aligned} r' &= 0, \\ l' &= h \end{aligned}$$

References

[1] Vitalik Buterin "Improving front running resistance of x*y=k market makers" March 2018

[2] Hayden Adams "Uniswap Whitepaper" February 2020

[3] Yi Zhang, Xiaohong Chen, and Daejun Park "Formal Specification of Constant Product Market Maker Model and Implementation" October 2018

[4] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson "Uniswap v3 Core" March 2021