BlueStone Protocol

Yiran Tao, Schowen Peng

June 28, 2020
version 1.0

Contents

1 Part I **1**
1.1 Introduction 1
1.1.1 Fixed Borrowing Rate 1
1.1.2 Customized Deposit and Loan Term 2
1.1.3 Personalized Financial Records 2
1.2 Basics Features and Implementations 2
1.3 Tradable Certificate of Deposit (CD) 4
1.4 Proof-of-Interest (POI) Consensus 4

2 Part II **5**
2.1 Deposit Weight Model 5
2.2 Borrowing Rate Model 5
2.2.1 Linear Model 5
2.2.2 Data Selecting & Loss Function 6
2.2.3 Analytical Solution 6

1 Part I

1.1 Introduction

We start our journey to design the BlueStone Protocol with one future in mind: the demand for on-chain lending services will grow exponentially as the crypto economy continues to expand. The diversity of lending use cases will require more advanced and scalable financial infrastructures.

This paper introduces a pool-based decentralized lending protocol with three building blocks: **Fixed Borrowing Rate**, **Customized Deposit and Loan Term** and **Personalized Financial Records**.

1.1.1 Fixed Borrowing Rate

At the bootstrap phase, the borrowing rate will be based on external market prices and set through oracles; an autonomous interest rate model will be developed in parallel as more deposit and loan data is collected. The lending rate will then be calculated on-chain based on the borrowing rate and the utilization of liquidity.

1.1.2 *Customized Deposit and Loan Term*

Unlike traditional loans which have due dates, most decentralized lending protocols allow their borrowers to repay at any time. The lack of commitment comes with two major downsides: 1) borrowers are not incentivized to borrow for the exact terms they need, thus hurt the overall efficiency of liquidity matching; 2) lenders can't withdraw their funds at will, as there is no guarantee that there will be enough liquidity in the protocol at the time. By Introducing terms, BlueStone is able to differentiate borrowing demands, offer favorable rates and guarantee liquidity at the same time.

1.1.3 *Personalized Financial Records*

The lack of identities and financial records has greatly impeded the potential of on-chain lending. By engaging with the BlueStone protocol, both lenders and borrowers will be producing meaningful data directly on the blockchain. For example, Dapp developers will be able to query the loan history of an Ethereum address and decide whether this address is eligible to enjoy certain benefits. In this way, prices can be targeted for different borrowers and funds can be more efficiently allocated. In addition, other parties are incentivized to develop on top of the BlueStone protocol and lay the groundwork for an open identity system.

1.2 *Basics Features and Implementations*

In Phase 0, users can interact with BlueStone Dapp, which provides basic lending and borrowing services.

- After the lenders deposit assets into the protocol, the deposits will be stored inside a `Pool`. The length M of a `Pool` represents the maximum term available to the lenders and borrowers.
- Borrowers are allowed to borrow for any term T between the range of $[1, M]$, by providing collateral accepted in the protocol; for each term he or she chooses, the rate is determined by the following on-chain function:

  $$BorrowingRate = H - \frac{H - L}{M} \times T$$

  (L, H) is the borrowing rate range.

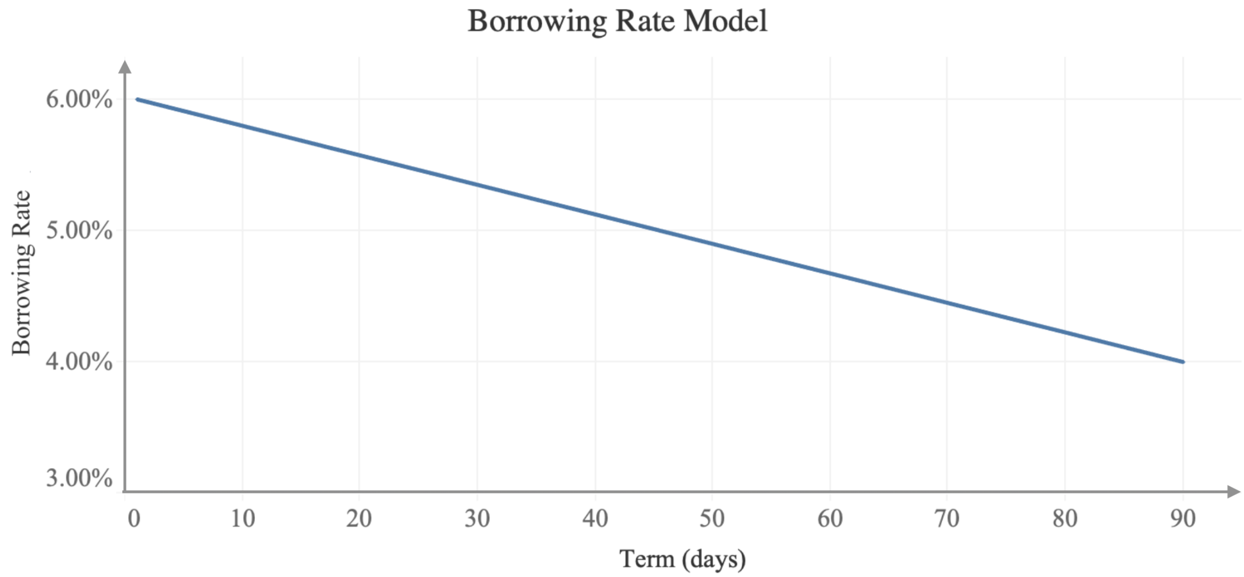

Fig 1 borrowing rate vs. borrowing term

For example in Fig.1, M is set to 90, and (L, H) is set to $(10\%, 15\%)$. Alice wants to borrow 2000 DAI for 15-day, the calculated $APR_{15} = 15\% - \frac{15\%-10\%}{90} \times 15 = 14.16\%$ and the required collateralization ratio is 150%, she needs to collateralize at least $\$2000 \times 150\% = \$3000$ worth of ETH to secure her debt position; and pay back $2000 + 2000 \times 14.16\% \times \frac{15}{365} = 2011.6384$ DAI in 15 days after this loan was originated.

- Lenders can deposit their assets (denoted as amount A) with selected term T and withdraw the principal A and accrued interest after the deposit matures. Lenders are given the flexibility to choose T between the range of $[1, M]$ as well. After a lender deposits into the protocol for term T, his or her deposit will be stored inside `Pool[T]`; this pool rotates from $\{$`Pool[M], Pool[M-1], ..., Pool[0]`$\}$ and accepts deposits at different T.To ensure the fairness of deposit interest allocation, the protocol employs a deposit weight model to reward the contribution of both amount and term:

$$W = A \times T$$

For example, if Bob deposits 100 DAI for 15-day, his deposit will be stored at `Pool[15]` and the deposit weight $W = 100 \times 15 = 1500$. After five days, Charlie deposits another 100 DAI for 10-day, his deposit will be stored in the same pool as Bobs; Charlie's $W = 100 \times 10 = 1000$. If this pool accrued 20 DAI interests after it matures, Bob will receive $20 \times \frac{1500}{2500} = 12$ DAI interest, and Charlie will receive $20 \times \frac{1000}{2500} = 8$ DAI interest.

The modularization of liquidity pool, term, borrowing rate and deposit weight model allows us to optimize the efficiency of the protocol continually. The optimization process will be discussed further in part II.

1.3 Tradable Certificate of Deposit (CD)

Certificate of Deposit is lender's claim to the underlying deposit and interest at maturity. In Phase 1, the discounting system will be available to the lenders so that they can sell their CDs to an open market for instant liquidity.

For example, if Alice deposits $100 DAI for 30-day; on Day 20, she is entitled to the $100 principal and $4 interest and decides to withdraw early; she can sell her CD to an open auction contract at an initial price of $102. If Bob and Carol start to bid this CD and Bob wins the auction at the price of $103; when the loan matures in 10 days, Bob can recoup the principal and interest of this deposit.

1.4 Proof-of-Interest (POI) Consensus

The Proof-of-Interest consensus and the POI token will be introduced in Phase 2. POI tokens can be obtained by making contributions to the protocol: initial purchasing and subsequent mining. The protocol allocates a capped amount of POI token per day to reward the liquidity providers and the interest payers.

The difficulty of mining will take into account the utilization ratio, the deposit supplied and the interest repaid within that day. The basic formulas of mining are as follows:

$$POI_{lender} \sim f_1(\frac{Deposit_{lender}}{Deposit_{total}}, UtilizationRatio)$$

$$POI_{borrower} \sim f_2(\frac{Interest_{borrower}/Interest_{total}}{UtilizationRatio})$$

This mechanism is designed to achieve a high level of liquidity and maintain the market in equilibrium by rewarding contributors dynamically.

The primary utility of the POI token is to access discounted interest rates. By staking POI tokens, holders can not only enjoy better rates, but also auction their positions to other borrowers. For example, Alice stakes some POI tokens into the protocol and receives 5% discount off the borrowing APR; she can either borrow assets with a 95% rate of the current APR, or rent out the discount position to Bob and receive 2% carry in return. In this way, the volatility risk and the interest benefit of holding POI tokens are shared by different parties in the system.

Therefore, players with various incentives are encouraged to create positive externalities: lenders earn interest by providing liquidity to the pool, borrowers are rewarded with the POI token for paying interest; POI holders share the risk of the system and make profits from renting out their privileges. These stakeholders will add to the network effect of the POI protocol and make decentralized governance possible. In the long run, we are open to explore the usage of POI for on-chain governance. Eg., POI holders are held accountable for the stability of the system through voting on risk parameters, including minimum collateralization ratios, deposit/loan terms, base interest rates, and etc.

2 Part II

2.1 Deposit Weight Model

As discussed above, the purpose of deposit weight model is to ensure the fairness of deposit interest allocation. The "vanilla" model $W = A \times T$ will be employed at first; sophisticated models will be developed after more deposit data are collected. The mathematics explanations behind this process are as follows:

A precise weight distribution function can be defined as $F(A_1, T_1; A_2, T_2; A_3, T_3; \dots)$, the value of the function F are in the range $[0, 1]$. $F_i(A_1, T_1; A_2, T_2; A_3, T_3; \dots)$ represents the deposit weight of user i and F_i meets the identity equation: $\Sigma_{i=1}^{n} F_i = 1$, n is the total number of users in a single liquidity pool. Thus the basics model can be written in the following format:

$$F_i = \frac{A_i \times T_i}{\Sigma_{i=1}^{n}(A_i \times T_i)}$$

Another potential solution of $F_i(A_1, T_1; A_2, T_2; A_3, T_3; \dots)$ could be `softmax function`, where

$$F_i = \frac{\exp A_i \times T_i}{\Sigma_{i=1}^{n} \exp\left(A_i \times T_i\right)}$$

In addition to the two options above, the weight distribution function can be iterated through real data training. First, a loss function needs to be defined, to indicate the "badness" of model performance; the most common loss function is the **mean square error**:

$$E = \frac{1}{2n} \Sigma_{i=1}^{n} (\tilde{F_i} - F_i)^2$$

F_i is the output of the weight distribution function, $\tilde{F_i}$ is the real deposit data we collects, and n is the total number of users in a single liquidity pool. The model performs better if the value of loss function is closer to zero and the purpose of the training is to find a function F, which can minimize the value of E.

2.2 Borrowing Rate Model

The on-chain borrowing rate model can simulate the real worlds rates autonomously. At first, a naive linear regression is employed to model the relationship between the on-chain rate and the real world rate.

2.2.1 Linear Model

Currently, these four factors are considered in the equation: maximum term M , loan term T, borrowing rate upper bound H, and lower bound L.

$$BorrowingRate = W_1 \times H + W_2 \times L + W_3 \times T + W_4 \times M + B$$

$BorrowingRate \in [L, H]$.

W_1, W_2, W_3, W_4 are weights, B is bias.

$BorrowingRate(BR)$ is the model output which simulates real borrowing rate.

2.2.2 *Data Selecting & Loss Function*

The real world borrowing data $H, L, T, M, \tilde{BR}$ are collected to minimize the errors. $Borrow\tilde{ing}Rate(\tilde{BR})$ is the real borrowing rate we collects.

Suppose that the number of sample data collected is n; for the i^{th} sample $BR_i = W_1 \times H_i + W_2 \times L_i + W_3 \times T_i + W_4 \times M_i + B$ and its matrix form is:

$$(BR_1, BR_2, \dots BR_n) = (W_1, W_2, W_3, W_4) \begin{bmatrix} H_1 & \dots & H_n \\ L_1 & \dots & L_n \\ T_1 & \dots & T_n \\ M_1 & \dots & M_n \end{bmatrix} + (B, B, \dots B)$$

Again, the most common loss function, **mean square error**, is employed to express the errors between BR and $\tilde{BR}$. It can be written as follow:

$$E = \frac{1}{2n} \Sigma_{i=1}^{n} (\tilde{BR_i} - BR_i)^2 = \frac{1}{2n} [\tilde{BR_i} - (W_1 \times H_i + W_2 \times L_i + W_3 \times T_i + W_4 \times M_i + B)]^2$$

An optimal set of mode parameters is needed, referred as W_1, W_2, W_3, W_4, B, to make the loss value E minimum.

2.2.3 *Analytical Solution*

Since the model and loss function are relatively straightforward, the minimum value of the loss function can be derived directly. This type of solution is called analytical solution.

The loss function can be expressed in the matrix multiplication form and H_i, L_i, T_i, M_i are rewritten as X_{1i}, X_{2i}, X_{3i}, X_{4i} respectively. For the convenience of calculation, W is denoted as $\bar{W} = (W_1, W_2, W_3, W_4, B)$, and a row of 1s are added at the end of the X.

$$\bar{X} = \begin{bmatrix} H_1 & \dots & H_n \\ L_1 & \dots & L_n \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix},$$

$$\begin{aligned} E &= \frac{1}{2n}\Sigma_{i=1}^{n}(\tilde{BR}_i - BR_i)^2 \\ &= \frac{1}{2n}\Sigma_{i=1}^{n}[\tilde{BR}_i - (W_1 \times H_i + W_2 \times L_i + W_3 \times T_i + W_4 \times M_i + B)]^2 \\ &= \frac{1}{2n}\Sigma_{i=1}^{n}[\tilde{BR}_i - (W_1 \times X_{1i} + W_2 \times X_{2i} + W_3 \times X_{3i} + W_4 \times X_{4i} + B)]^2 \\ &= \frac{1}{2n}(\tilde{BR} - \bar{W} \times \bar{X})(\tilde{BR} - \bar{W} \times \bar{X})^T \end{aligned}$$

$\times$ is the matrix multiplication, ^{T} is the transpose of the matrix. Let derivative of loss function E equals 0, then $\bar{W}$ is inferred as below:

$$\begin{aligned} \frac{\partial E}{\partial \bar{W}} &= \nabla_{\bar{W}}(\frac{1}{2n}(\tilde{BR} - \bar{W} \times \bar{X})(\tilde{BR} - \bar{W} \times \bar{X})^T) \\ &= \frac{1}{n}[\nabla_{\bar{W}}(\tilde{BR} - \bar{W} \times \bar{X})] \times (\tilde{BR} - \bar{W} \times \bar{X})^T \\ &= -\frac{1}{n}\bar{X} \times (\tilde{BR} - \bar{W} \times \bar{X})^T \\ &= 0, \\ \bar{W} &= \tilde{BR} \times \bar{X}^T \times (\bar{X} \times \bar{X}^T)^{-1} \end{aligned}$$

As a result, the optimal values of W and B are obtained. These two values can make the loss function E reach the minimum value, and $BorrowingRate$ can fit the real world rate better.

$$\bar{W} = \tilde{BR} \times \bar{X}^T \times (\bar{X} \times \bar{X}^T)^{-1}$$

$$\bar{W} = (W_1, W_2, W_3, W_4, B)$$

More complex parameters will be considered in the future, such as, the borrowing amount, the utilization of liquidity pools and etc.