aave No.2 updateInterestRates

https://github.com/xiaoyu1998

Resolution:2560×1440(2K)
version: 6070e82d962d9b12835c88e68210d0e63f08d035

Entries

supply
aave-v3-core/contracts/protocol/pool/pool.sol

```
SupplyLogic.executeSupply(
```

executeSupply
aave-v3-core/contracts/protocol/libraries/logic/SupplyLogic.sol

```
reserve.updateState(reserveCache);

ValidationLogic.validateSupply(reserveCache, reserve, params.amount);
reserve.updateInterestRates(reserveCache, params.asset, params.amount, 0);
IERC20(params.asset).safeTransferFrom(msg.sender,
reserveCache.aTokenAddress, params.amount);
bool isFirstSupply = IAToken(reserveCache.aTokenAddress).mint(
```

ReserveLogic

updateState
aave-v3-core/contracts/protocol/libraries/logic/ReserveLogic.sol

```
_updateIndexes(reserve, reserveCache);
_accrueToTreasury(reserve, reserveCache);
```

updateInterestRates
存款利率/借款固定利率/借款变动利率

aave-v3-core/contracts/protocol/libraries/logic/ReserveLogic.sol

```
vars.totalVariableDebt = reserveCache.nextScaledVariableDebt.rayMul(
IReserveInterestRateStrategy(reserve.interestRateStrategyAddress).calculateInterestRates(
reserve.currentLiquidityRate = vars.nextLiquidityRate.toUint128();
reserve.currentStableBorrowRate = vars.nextStableRate.toUint128();
reserve.currentVariableBorrowRate = vars.nextVariableRate.toUint128();
```

ReserveInterestRateStategy

_updateIndexes
aave-v3-core/contracts/protocol/libraries/logic/ReserveLogic.sol

```
if (reserveCache.currLiquidityRate != 0) {
    uint256 cumulatedLiquidityInterest = MathUtils.calculateLinearInterest(
    reserveCache.nextLiquidityIndex = cumulatedLiquidityInterest.rayMul(
    reserve.liquidityIndex = reserveCache.nextLiquidityIndex.toUint128();
if (reserveCache.currScaledVariableDebt != 0) {
    uint256 cumulatedVariableBorrowInterest = MathUtils.calculateCompoundedInterest(
    reserveCache.nextVariableBorrowIndex = cumulatedVariableBorrowInterest.rayMul(
    reserve.variableBorrowIndex = reserveCache.nextVariableBorrowIndex.toUint128();
```

calculateInterestRates
aave-v3-core/contracts/protocol/pool/DefaultReserveIntersetRateStrategy.sol

```
vars.totalDebt = params.totalStableDebt + params.totalVariableDebt;

vars.currentVariableBorrowRate = _baseVariableBorrowRate;
vars.currentStableBorrowRate = getBaseStableBorrowRate();
if (vars.borrowUsageRatio > OPTIMAL_USAGE_RATIO) {
    vars.currentVariableBorrowRate +=
    vars.currentStableBorrowRate +=
          _stableRateSlope1 +_
          stableRateSlope2.rayMul(excessBorrowUsageRatio);
} else {
    vars.currentVariableBorrowRate +=
    _variableRateSlope1.rayMul(vars.borrowUsageRatio).rayDiv(
    vars.currentStableBorrowRate +=
    _stableRateSlope1.rayMul(vars.borrowUsageRatio).rayDiv(
if (vars.stableToTotalDebtRatio > OPTIMAL_STABLE_TO_TOTAL_DEBT_RATIO) {
    vars.currentStableBorrowRate +=
    _stableRateExcessOffset.rayMul(excessStableDebtRatio);
vars.currentLiquidityRate = _getOverallBorrowRate(
```

所有债务

总借款率 > 最佳比例

base_v+offset+v_slope1+v_slope2x过量费率(总借款)
v_slope1+offset+slope1+slope2x过量费率(总借款)

总借款率 < 最佳比例

base_v++slope1x借款率/最佳比例
v_slope1+offset++slope1x借款率/最佳比例

固定利率借款比例 > 最佳固定利率借款比例
+过量费率(固定利率借款)

_getOverallBorrowRate
aave-v3-core/contracts/protocol/pool/DefaultReserveIntersetRateStrategy.sol

```
uint256 totalDebt = totalStableDebt + totalVariableDebt;
uint256 weightedVariableRate =
totalVariableDebt.wadToRay().rayMul(currentVariableBorrowRate);
uint256 weightedStableRate =
totalStableDebt.wadToRay().rayMul(currentAverageStableBorrowRate);
uint256 overallBorrowRate = (weightedVariableRate +
weightedStableRate).rayDiv(
```

变动利率加权
固定利率加权
(变动利率加权 + 固定利率加权)/所有债务