

Split COMP rewards distribution and bug fixes

Updated as of block [19333212](#) at 2/29/2024, 7:36:35 AM ET

- ID: 62
- Proposer: [0xc8A69971DAa3C3ADd85Ab0d0AF297515769ddfFC](#)
- Start Block: 13285618 (9/23/2021, 9:52:46 PM ET)
- End Block: 13305328 (9/26/2021, 10:57:11 PM ET)
- Targets: [0x3d9819210A31b4961b30EF54bE2aed79B9c9Cd3B](#) ;
[0x374ABb8cE19A73f2c4EFAd642bda76c797f19233](#)

Forum Post

Forum post is present here: [Forum Post](#)

Table of contents

- [Proposal Text](#)
- [Checks](#)
 - [Checks Compound Proposal Details](#)  Passed

Proposal Text

Split COMP rewards distribution and bug fixes

This proposal completes RFP 15: Dynamic COMP reward distribution as well as fixes a few bugs.

Split COMP rewards distribution

Problem

At the moment, the COMP rewards rate for any single market is applied at the same rate for both suppliers and borrowers. This creates undesirable market conditions such as, but not limited to, negative interest rates when borrowing various assets.

An example is the WBTC market - we want to distribute COMP rewards for suppliers, but since this distribution rate applies at the same rate for both borrowers and suppliers, we end up with WBTC being borrowed at a negative interest rate. The same applies to most non-stablecoin markets.

Solution

This proposal changes the Comptroller logic to have two different COMP distribution rates for each and every market - borrow-side (`compBorrowSpeeds`) rate and supply-side (`compSupplySpeeds`) rate.

The function `Comptroller._setCompSpeeds(CToken[],uint256[],uint256[])` is introduced to enable governance to set these differing rates.

Example usage: `Comptroller._setCompSpeeds([cUSDC],[0],[1e18])` - will set the COMP rewards for supplying USDC at 0 COMP per block, and for borrowing at 1 COMP per block.

COMP rewards distribution bug fixes

This proposal fixes the following bugs:

- Market state indices were not being set when COMP rewards are added to already active markets. [\[Bug_1\]](#)
- COMP rewards were distributed for periods preceding when the distribution rate is set. [\[Bug_2\]](#)
- Borrowers were having to poke their borrow to accrue rewards if they borrowed before rewards rates were set. [\[Bug_3\]](#)

Technical changes

- `Comptroller#compSpeeds` storage variable is no longer being used and has been effectively deleted. It's replaced by `Comptroller#compBorrowSpeeds` and `Comptroller#compSupplySpeeds`. This proposal copies the current rates into the new storage variables.
- `Comptroller#_setCompSpeed(CToken,uint256)` function has been removed and replaced by `Comptroller#_setCompSpeeds(CToken[],uint256[],uint256[])`. This new function allows COMP rewards rates for multiple markets to be set with a single call.
- An upgrade hook initializes all non-initialized market state indices (initial index is 1e36).
- When new markets are added, their state indices are now properly initialized (used to use lazy initialization which caused many problems).

Other notable changes

- Depositing now uses slightly less gas after a user's initial deposit.
- Calling `Comptroller#claimComp` unfortunately uses a lot more gas to account for uninitialized borrow state indices. It's recommended to specify which markets (and which sides) to claim COMP in rather than claiming across all markets simultaneously.
- Flywheel tests have been updated and expanded upon to prevent the bugs fixed in this proposal from occurring again.
- These changes have been live on the Ropsten testnet for nearly a month and everything is working as expected.

Credits

- Arr00: initial code supporting split COMP rewards; flywheel test re-writes
- Getty: bugs 1 and 2 diagnosis help and disclosure
- Elee: bugs 1 and 2 diagnosis help, fixing, and disclosure
- Coburn: bug 3 identification; test case
- Compound Labs (especially Jared): reviewing, guidance, general help, and coordination
- TylerEther (me): full implementation of split COMP rewards; testing infrastructure changes; bugs 1 and 2 identification, diagnosis, fixing, and disclosure; bug 3 fix; overall coordination; general testing; addition of many new test cases; proposal



References

- [GitHub PR](#)
- [Proposal simulation](#)
- [Comptroller compSpeed bug disclosure](#)
- [Forums discussion](#)
- [Etherscan - contract](#)

Checks

Checks Compound Proposal Details Passed

Info:

- 1-  Function `_setPendingImplementation(address)` is removed from `0x3d9819210a31b4961b30ef54be2aed79b9c9cd3b` contract
- 2-  Upgrade of the Compound Comptroller contract to a new implementation [0x3d9819210A31b4961b30EF54bE2aeD79B9c9Cd3B](https://github.com/compound-finance/comound/blob/master/contracts/CDPManager.sol) from [0x374abb8ce19a73f2c4efad642bda76c797f19233](https://github.com/compound-finance/comound/blob/master/contracts/CDPManager.sol).