Notes on AAVE - can go in Appendix for details and in the main text.

Aave general description

Aave is one of the most used and battle tested lending protocols in DeFi. Before Aave version 1 (V1) the decentralised lending strategy was peer to peer (P2P), resulting in a direct lender - borrower matching handled by a smart contract. A traditional finance instrument to represent this is and OTC contract to which the Defi community added, at first, a layer for decentralisation and peers matching. The P2P approach enables the credit process in a decentralised environment but is not flexible and is illiquid cause the charged rate and lended amount in a potential loan must be published waiting for an interested counterpart.

Aave V1 breaks the P2P approach by creating lending pools where lenders deposit a cryptocurrency accepted by the protocol for the specific pool in exchange of a proxy token, receiving algorithmically calculated interest and borrowers borrow against the pool providing in exchange a collateral, chosen among the accepted ones from the protocol. Borrowing can be at variable or fix rate and each borrowing position has an health factor (Hf) which potentially triggers, based on an algorithm, a liquidation for an under collateralised loan (Hf below a calculated threshold). In traditional finance this can be similar to a margin call with the addition of an upgradable protocol driving the calls based on mathematical functions proposed through a governance process.

Aave version 2 enhance version 1 as gives the possibility of upgrading the proxy tokens given to lenders, reduces gas inefficiencies and simplifies code and architecture.

V1 has the merit of flashloans introduction and V2 allows to use them for collateral trading, which means swapping collateral without closing and reopening a position, loan repayments, margin trading, debt swaps and margin deposits. The smart contract handling the lending pool including flashloan calls is deployed on ethereum and can be tracked. It is the smart contract Lending Pool (contract address in ethereum main network :0x7d2768dE32b0b80b7a3454c06BdAc94A69DDc7A9)

The Lending Pool includes calls which allow to deposit, redeem, borrow, repay, swap rate, liquidate, calling flash loans among others. In this paper we focus on calls to LendingPool,flashloans from ethereum inception to block number ….. (corresponding to date: )

Flashloans

Flash loans are unique in the blockchain environment and allow to perform various operation using a loan without the need of a collateral, provided that the loan is paid, interests included, within the same ethereum transaction (1). Flashloans are the main topic of this paper therefore is critical to clarify how they work. It is extremely common when reading documentation or articles to get confused by the statement that the repayment should happen within the same ethereum transaction: this can be misleading as with ethereum transaction, in this case is intended a transaction which can hold hundreds of transactions. To avoid any confusion we recall the definition of transactions and block. Transactions are intended to be cryptographically signed instructions sent from one Ethereum account to another where the account can be a public address associated to private keys typically controlled by a person or a contract account associated to a smart contract and controlled by code. A block instead is a component of the chain, containing among the others a set of transactions and the information to rebuild the chain itself. The nature of blockchain permits the creation of flashloans cause if a set of operations, in this case inner transaction within the flashloan one, is not ending with the full repayment, all the operations will be aborted by not being added to the block, therefore not ending in the blockchain. The original name of flashloans, One Block Borrows, though semantically not immediate avoided any technical confusion. The described behaviour is comparable to a rollback in a traditional relational database where the atomic operation is the full bunch of inner transaction wrapped by the flash loan one.

Flashloans are not enabling the user to get an infinite loan. The constraint are the liquidity of the pool, as a bigger amount than the pool size itself cannot be accessed, and borrowing enabled for such pool. An exception to the full repayment of the loan before the block addition is that a debt position, if allowed, is opened by the end of the transaction.

Appendix ideas

For the appendix I can add the Aave architecture.