Graphene Improvement Proposal

Conversion from DPOS to GPOS blockchain

Contributors: Jonathan Baha'i, Matt Hunter, Robert Thomas, Jeff Maxwell

Abstract:

The original intent of Graphene was to operate as a Decentralized Autonomous Community (DAC) where DPOS enables the voting collective of core token holders to determine whom would act as Committee, Witness, and Workers within typical Graphene blockchains. However, the challenges of voter turnout continue to plague all Graphene based DPOS blockchains such as Bitshares, Steem, Peerplays, and EOS. This proposal aims to resolve this issue by attaching a rewards mechanism in the blockchain directly to the voting activity.

This protocol change will convert a Graphene blockchain from a DPOS consensus to a GPOS consensus, Gamified Proof of Stake. The CORE token will continue to operate as a core token to the Graphene blockchain essential for all core blockchain operations.

All fees that the Graphene Blockchain charges for smart contract operations and peer-to-peer transactions are currently burned as a means of deflation of the CORE token. This mechanism would have all or a portion of those fees become part of participation rewards.

Being a DPOS consensus blockchain, the importance of voter participation of these CORE holders is paramount to the security of the blockchain. In this proposal, we introduce a protocol change; where only holders of *vested* CORE tokens will receive a 'participation reward' by securing the blockchain and its governance through voting on the operations of the blockchain including things such as witnesses, committee, proxies, and worker proposals or other features that require decentralized stakeholder approval.

CORE Token holders will be able to vest or "stake" their tokens and only be eligible to vote through their vested balance after a predefined period (proxy voting will still be allowed).

Active voting in governance of the blockchain will be necessary in order to receive participation rewards. When a user participates in voting activity within the distribution period (typically 30 days), their staking time will be set to 100%. However, with each interval of

participation rewards that are paid out, there will be a staged linear decay to their voting power, and thus their reward over a predefined period which can be voted on by the CORE token holders via the Committee.

Hence, each vesting balance will have a linear weight applied to it when voting. This weight will be 100% initially, and then in alignment with the monthly participation reward will step down in linear decay towards 0% only if the user does not participate in voting. At this point, rewards will no longer be paid to the owner of the vesting balance, and their voting influence in the blockchain will cease until they participate again.

The contributors recommendation is to make this a six month period to start with the understanding that the Committee may change this parameter later if it's seen fit. Any rewards which would have been provided to participants which has decayed in their participation will be sent to the worker proposal reserve where it will be available for improvements to be made on the Graphene blockchain. This ensures that participation rewards are only shared with those that participate actively, and any rewards that went unused are later available for other types of work in the blockchain.

For other tokens within the blockchain, such as UIAs, if this feature is enabled with the participation rewards option then the participation that takes place with the CORE token will qualify the user to receive participation rewards in the UIA they carry. This ensures that those who wish to utilize the UIAs distribution mechanism for participation rewards also become active participants in the blockchains governance.

This will benefit the community by having more engaged voters and compensating those who are active users in the Graphene blockchain. The result is a more stable and active community surrounding the Graphene ecosystem, and a more secure blockchain in comparison to original DPOS consensus.

Motivation:

Key issues in Blockchain governance are: Who should have input on how a blockchain runs? Who should receive participation rewards from the Blockchain? All the accounts? Only the witnesses? How can we use incentives to increase the activity of the blockchain to make it more secure?

Compensating users who have vested or 'staked' their tokens with the ability to vote and receive revenue from the blockchain aligns the need for participation in voting on the various

operations of the DPOS blockchain, with the desire of the user to be compensated for the work that goes into the activity. Voting in DPOS blockchains has been found to be difficult due to the amount of research necessary to make voting choices. Those that are involved in the Blockchain receive more of a "say" in what will happen; decreasing the apathy that affects all governance of a Blockchain.

Assumptions:

In order to utilize this mechanism the Graphene blockchain must implement the profit sharing code provided through Peerplays in 2016. Other changes will need to be made changing the ‘profit sharing’ to ‘participation rewards’ for governance. A "vested" user is one who has a "vested balance" in the blockchain for a length of time that is equal to or longer than the "vested period" (i.e. 30 days which is equal to when participation rewards are paid out).

This change to the Blockchain will not cause issues for the wallet and other pieces that use the Blockchain. However, some modifications will need to be made to the Graphene Core Wallet and any other wallets which enable voting in order to accommodate properly reporting separate vesting balances and decay periods in order to have the end user aware of their current voting status.

There also needs to be support for UIA management of features and showing of participation reward payouts. There will be a slight increase in the load of the maintenance interval due to the vote decay mechanism, while there will be a slight decrease in the load due to the reduction of participation reward calculations.

Advantages:

- This protocol will benefit active members of the Graphene blockchain community by rewarding them for participation at every distribution interval.
- Voters who are vested will be more engaged in the community and ensure the Graphene blockchain operates as it was intended (Decentralized Autonomous Community).
- Members balances will be better protected in the event of personal private key hack, attempts to withdraw balances that are vested would not be possible and likely responded to in time before an attempt was successful.

- The GPOS consensus ensures a higher degree of security for the blockchain through rewarding active voting.
- Rewards are solely based on distribution of actual volumes generated by blockchain activity and not through ongoing token generation the same way block rewards are produced for witnesses.
- Incentives for active end users utilization of blockchain operations are better aligned.

Specification:

With the change to the blockchain some functionality will need to be added to support the proposed protocol change. Distribution logic will need to be changed to compensate just those accounts with vested balances. Chain libraries will need to reflect this change, as well. Current ability to vote and enable proxy voting should be replicated based on your vested balance, in order to qualify for the blockchain rewards mechanism. This helps the user differentiate their account between an active and an inactive one. Change the vested period to match the payout schedule (this can be changed by the community). Voter decay mechanism would be the only new operations introduced as part of this proposal. Graphene documentation will be updated to reflect the proposed change.

To keep things flexible, the distribution (dividends) feature should be modified in a way that only *EVER* allows to distribute according to vesting balance *IF and only if* the ‘target asset’ is CORE. Then, we can still have other distribution schemes that do not require vesting. Maybe have this as a flag on the asset.

For calculation of fee distribution:

db_maint.cpp:schedule_pending_dividend_balances

Current Purpose:
Schedules payouts from a dividend distribution account to the current holders of the dividend-paying asset. This takes any deposits made to the dividend distribution account since the last time it was called, and distributes them to the current owners of the dividend-paying asset according to the amount they own.

Changes necessary:

The method currently considers all users with balances, and all users with vested balances, then loops through calculating how to distribute the dividends. This would be reworked to only consider the users with a vested interest.

The current payout loop at 917:

```
for (const account_balance_object& holder_balance_object :
boost::make_iterator_range(holder_balances_begin,
holder_balances_end))
                {
```

needs to be updated to not consider all accounts with balances, but only those with vested balances. Most of this should be possible by simply replacing the original pull of all accounts with balances at the start of the method, near line 751:

```
auto holder_balances_begin =
balance_index.indices().get<by_asset_balance>().lower_bound(boost::make_
tu ple(dividend_holder_asset_obj.id));
  auto holder_balances_end =
balance_index.indices().get<by_asset_balance>().upper_bound(boost::make_
tu ple(dividend_holder_asset_obj.id, share_type()));
```

with the logic appearing just below this point to instead get the list of accounts with nonzero vesting balances.

db_maint.cpp:process_dividend_assets

Current Purpose:

Iterates through the dividend assets to be paid out, and calls schedule_pending_dividend_balances (discussed above) handle the distribution.

Changes necessary:

For increased efficiency, changes could be made at this level - since the schedule_pending_dividend_balances wouldn't need the full list of accounts with regular non-zero balances, but only those with non-zero vested balances.

It passes this list of accounts around line 1065:

```
schedule_pending_dividend_balances(db, dividend_holder_asset_obj,
dividend_data, current_head_block_time,balance_index,
vbalance_index, distributed_dividend_balance_index,
pending_payout_balance_index);
```

this could be reworked to only pass in the accounts with vested balances.

The loop beginning at 1108:

```
for (auto pending_balance_object_iter =
pending_payouts_range.first; pending_balance_object_iter !=
pending_payouts_range.second; )
```

uses the pending_payout_balance_index - which is the result of:

```
db.get_index_type<pending_dividend_payout_balance_for_holder_object_inde
x> ()
```

It would appear that this db function will need to be updated to use an index of accounts with vested balances vs all accounts with non-zero balances.

For Vote Calculations:

db_maint - struct vote_tally_helper

Changes Needed:

Currently this calculates the voting_stake by taking into account total balance, balance tied up in orders, etc. and then adds on the vesting stake. For this change only the vesting stake would be used.

db_maint:update_top_n_authorities

Current Purpose:

Calculate the top asset holders for the purposes of adjusting vote weighting

Changes Needed:

Do these calculations using vested asset balances instead of account balances.

vote_count.hpp - vote_counter struct

should take into account vested stake when calculating

account_object.hpp - property

```
share_type total_core_in_orders;
```

Tracks the assets that are tied up in ordering, to be deducted from vote weight. If switching to vested-asset weighted voting, this shouldn't be necessary.

Wallet changes:

The CLI and GUI wallets will need to reflect the need to vest for voting participation. The CLI will need to be updated with a new function that specifically places the balance in the predefined vesting balance that is intended to participate in voting. The GUI will require some changes to enable this process as well. The most ideal location would be to have it as the first tab available under 'Vote" prior to voting on other current elements of the blockchain including the option to set proxies, vote for witnesses, and advisors.

CLI

-The Wallet CLI should remain unaffected. Votes placed for a committee member will only be counted if the user has a vested balance. Votes reside within your user/account and are only counted/weighted during the vote tally process - at which point votes from members without a vested balance would be ignored.

Estimates for Work

C++ changes to the blockchain:

25-30 days to:

- Setup test environment and accounts and develop scenarios to thoroughly test the results of the changes
- Make changes to the blockchain code and redeploy to the server
- Work through the designed tests to ensure functionality
- Work through major issues as they arise
- Have code ready for more thorough testing by QA

5 days to:

- Have QA test and work through issues as discovered

Graphene GUI app:

5 days to:

- Make changes to the React/Redux app, adding in new tabs and functionality to better indicate to users how voting works with vested balances

3 days to:

- Process through QA, resolve minor issues as discovered

References:

1) Bitshares blockchain repository. https://github.com/bitshares/bitshares-core
2) BSIP-019 Draft: "Introducing profit sharing/dividends to Bitshares" https://steemit.com/bitshares/@cm-steem/bsip-019-draft-introducing-profit-sharing-dividends-to-bitshares
3) Peerplays Profit Sharing Code. https://steemit.com/bitshares/@xeroc/peerplays-providing-code-for-profit-sharing-1467622607-0030067