BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

This is the Eighth edition of the BCH Pulse developer newsletter. Existing projects have been updated. We will now be moving to a bi-monthly publication. Submissions are due in by the 1st and 15th of each month. Information can be sent to the new email bch.dev.mail@gmail.com Thanks!

Here is some of the work that is being built on BCH by the Developers. This does not take into account work done by developers who chose to remain anonymous.

Developers ~

Chris Pacia - BCHD

Recently completed:
*Updated fast sync checkpoint

Currently working on:
*Misc work on SLP

Josh Ellithorpe - BCHD

Recently completed:
*Fixed mutex issue
*Fixed memory leak in utxo cache
*Updated docker image, updated my mainnet and testnet public nodes, added new utxo set for fastsync on ipfs for 0.15.1 release

*IBD runs to test the syncing changes

Tyler Smith - BCHD

Recently completed:
*BCHD bug fixes and code review

Currently working on:
*Monitoring and alerting system for bchd gRPC network
*Trying to finish first draft spec for Avalanche based pre and post consensus

Future work:
*Gathering and implementing feedback on Avalanche spec
*Continuing Avalanche experiments, analysis, and research
*Adapting backports from btcd and dcrd into bchd

Wants help with:
*Adapting backports from btcd and dcrd into bchd

Mark Lundeberg - Bitcoin ABC

Recently completed:
*ABC implementation of November 15 network upgrade.
*CashFusion alpha.

Currently working on:
*Implementing CashFusion
*Organization & planning for May 2020 upgrade

BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

Jason Cox - Bitcoin ABC

Recently completed:
*Improved automated test coverage.
*Automation infrastructure improvements.

Currently working on:
*Better review process UX.
*More release automation and test coverage.

Future work:
*Faster infrastructure deployments.

Wants help with:
*Unit test, utility, and network backports:
https://github.com/Bitcoin-ABC/bitcoin-abc/blob/master/doc/backporting.md

Amaury Sechet - Bitcoin ABC

Currently working on:
*PSBT
*Support for Quic

Wants help with:
*Backporting from Core.
*Address indexer.
*Bitcore-compatible RPCs

Joshua Green - Bitcoin Verde

Recently completed:
*Node P2P Message for SLP Validation
*Bitcoin Verde Wallet SLP Validation via Node P2P Message
*Performance Testing for Database Partitioning
*Evaluating Database Storage (Binary Hashes)
*Resolved Blockchain Segment Corruption Bug (resulting in Node stalling after being served a malicious invalid block)

Currently working on:
*Hardening the BinaryHash changes.
*Hardening the Database Partition changes.

Future work:
*BCH Spec Documentation (Starting 12/2)

Wants help with:
*BCH Spec Documentation. (https://t.me/bitcoinverde)

BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

Andrea Suisani - Bitcoin Unlimited

Recently completed:
*Updated cashnodes.io infrastructure, backend, update to keep track of min consensus upgrade version.

Currently working on:
*Documentation: Overall description of the new CPFP algo for BU (waiting for feedback)
*Documentation: Mitigating of DS attacks leveraging delta in mempool admission policies (waiting for feedback)
*Finish up current open BU PRs:
- port avx2
- sse4 based optimization for sha256
- update BU RNG

Future work:
*Experiment with long chain of unconfirmed on mainnet
*The aim is to asses delta in DS success rate against 0-conf in case of heterogeneous mempool admission policies
*Second iteration of the Gigablock testnet experiment

Peter Tschipper - Bitcoin Unlimited

Recently completed:
*Implemented efficient Child-Pays-for-Parent algorithm for mining long unconfirmed chains

Currently working on:
*Finishing up resolving the O(n^2) post block processing issues related to long chains

Future work:
*Giga-net testing

George Bissias - Bitcoin Unlimited

Currently working on:
*More robust failure recovery for Graphene.
- Graphene blocks will fail to decode with some tunable frequency (roughly one per day). Currently, an Xthin or Compact block is requested whenever the Graphene block fails to decode. In this work, we will add a failure recovery mechanism to Graphene that leverages the data already sent to decode the block with minimal additional information from the sender.
Details can be found here: https://people.cs.umass.edu/~gbiss/graphene.sigcomm.pdf

Future work:
*Continue to develop improvements to the Graphene protocol.
*Develop a prototype of the Bobtail protocol for Bitcoin Unlimited alongside Storm.

Wants help with:
Feedback on:
https://bitco.in/forum/threads/buip131-bobtail-prototype-extending-storm-on-bitcoin-unlimited.24903

BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

Justin Holmes - Bitcoin Unlimited

Currently working on:
*Usability improvements to the transaction rate graph.

Pokkst - Crescent Cash

Recently completed:
*Crescent Cash rewrite. UI is now separated into multiple activities and app runs a lot smoother.
*A lot of new features including UTXO management, key management, key signing, key verifying, and more.
*tipbitcoin.cash now has Badger Button support so people can easily tip with Badger Wallet for both BCH and SPICE.

Future work:
*Crescent Cash update with bug fixes and Samsung DeX support

Calin Culianu - Electron Cash

Recently completed:
*4.0.11 release of Electron Cash

Currently working on:
*Contemplating beginning to merge SLP and Electron Cash main into 1 app

Future work:
*Replace ElectrumX server with something not as slow

Wants help with:
*Skilled person who can tolerate pain.. to also contribute to Electron Cash and learn the codebase.

Imaginary_Username - Electron Cash

Recently completed:
*Doing reviews for misc projects.

Currently working on:
*Reusable address: Working closely with both bchd and Harry Barber for a layer on top of bitcoind. Prototype server: https://github.com/hlb8122/prefix-server/tree/dev/grpc

Future work:
*Assist with double-spend proof testing and implementation
*Investigate Avalanche and Storm pre-consensus mechanisms

Wants help with:
*Reusable address: Wallet implementation! Any help from people who have any confidence in low-level transaction-making is greatly appreciated.

Spec here: https://github.com/imaginaryusername/Reusable_specs/blob/master/reusable_addresses.md

BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

Jonald Fyookball - Electron Cash

Recently completed:
*Alpha version of Cash Fusion Wallet for internal testing

Currently working on:
*CashFusion

James Cramer - SLPDB / Electron Cash SLP

Recently completed:
*SLP security audit
*SLPDB regtest end to end tests added and associated patches
*Various SLP library updates (e.g., slpjs, slp-validate)

Currently working on:
*EC SLP vNext release
*SLP bounties / blogging website

Future work:
*SLPDB capacity improvements

Wants help with:
*Smart contracts for SLP minting https://github.com/simpleledgerinc/slp-mint-contracts

AlwaysAn0n - CashShuffle

Wants help with:

*CashShuffle library unit tests and performance audit

Chris Troutner - Bitcoin.com

Recently completed:
*rest.bitcoin.com is now much faster, with more speed and scaling improvements coming.

*account.bchjs.cash and api.bchjs.cash is a prototype pay-to-play REST API based on this video presentation: https://www.youtube.com/watch?v=oFa8Q2OCSaw. Everything is open source if anyone would like to duplicate. Contact me for details: trout@bitcoin.com

*The server-side software for uncensorable website publishing using BCH and IPFS has had some major upgrades: troutsblog.com/about for details

Currently working on:
slp-cli-wallet has some bugs: https://github.com/christroutner/bch-cli-wallet

Future work:
More improvements to the token-liquidity app powering the SLP token exchange on psfoundation.cash is getting a major overhaul.

Wants help with:
Would love help with bugs in slp-cli-wallet: https://github.com/christroutner/bch-cli-wallet/issues. People can reach out on the Permissionless Software Foundation Telegram channel: https://t.me/permissionless_software

BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

Jason Dreyzehner - Bitauth

Recently completed:

*Released a Bitauth IDE guide:
https://blog.bitjson.com/how-to-write-custom-bitcoin-scripts-in-bitauth-ide-10216b3eb09c
(Help and feedback: https://t.me/bitauth_ide)

*Released CashChannels:
https://blog.bitjson.com/cashchannels-recurring-payments-for-bitcoin-cash-3b274fbfa6e2

Currently working on:

*Developing and testing application protocols for CashChannels
(https://blog.bitjson.com/evaluate-and-debug-bitcoin-cash-scripts-in-javascript-3e182136000d)

Karol Trzeszczkowski - Plugins

Recently completed:

*Released Simple Escrow Electron Cash Plugin
https://github.com/KarolTrzeszczkowski/Electron-Cash-Simple-Escrow-Plugin

*Updated Mecenas Plugin, introduced new types of Mecenas contract
https://github.com/KarolTrzeszczkowski/Mecenas-recurring-payment-EC-plugin/releases

Currently working on:

*Paper wallet manager plugin for Electron Cash,
*Covenant applications for SLP security operations,
*Server automating Mecenas recurring payments,

Future work:

*Writing covenant contract spec and writing down thoughts about good practices (help appreciated),
*Support for CashChannels in Mecenas plugin,
*Web service helping wallets implement covenant contract support

BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

Tendo Pein - Spedn

Recently completed:

*Introducing an array type with syntax [type; length], for example [Sig; 3] signatures, [byte; 10] message. The compiler type-checks the lengths so for example a type of message . message expression will be inferred as [byte; 20].

*Syntax for tuple assignment now allows its items to be of different type: (int a, Sig b, PubKey c) = expr;
*Array elements can be accessed with x[i] syntax.

*Introducing a bit type. Only arrays of bits are useful, as they represent a type of checkbits argument in the checkMultiSig function which was upgraded for Schnorr support. Bit array literal is also introduced, ex. [bit; 5] checkbits = 0b00110. checkMultiSig accepts an additional checkbits argument, as described in Nov 15 hard-fork spec.

*For a byte array of unknown size there is [byte] type which replaces the former bin type.
*Introducing (UTF-8) string literals, ex. [byte] message = “Hello, World”;.
*Introducing custom type declarations (type aliases) which can be placed before contract declarations and then used as any other type in the contract.

*Ex. type Message = [byte; 10];. Actually, Sig, DataSig, PubKey, Ripemd160, Sha1, Sha256, Time and TimeSpan are defined internally as aliases.

*Introducing separator; statement that compiles to OP_CODESEPARATOR.
*Introducing fail; statement that compiles to OP_RETURN.
*Introducing checkSize(x) function that returns true if the runtime size of a byte array matches declared type.
*Variable names can now contain underscores, ex. [byte] my_string.

Currently working on:

Exploring possible code optimization algorithms.

Future work:

*Macros
*Pattern matching
*IDE support

Tobias Ruck - SLPDEX

Recently completed:

Cashcontracts-rs, which makes working with transactions and contracts much simpler:
https://github.com/slpdex/cashcontracts-rs

Currently working on:

*Refactoring Cirrus (previously slpdexdb/cryptopandasdb) to be just a single component with interfaces:
https://github.com/slpdex/cirrus

*Experimenting with building a CashAssemblyinterpreter for Nimbus
*Be.Cash

Wants help with:

*async/await stuff for Nimbus (anyone who’s a Rust developer could help)
*Feedback/review once a paper has been published
https://github.com/slpdex/cashcontracts-rs

BCH Open-Source Pulse: Vol #8

Period ending 30/11/2019

Rosco Kalis - CashScript

Recently completed:

*Released CashScript v0.2.2 with some usability improvements and improved compatibility with SLP. Presented on BCH smart contracts at the BCH London meetup, hopefully got some people excited.

Currently working on:

*Finishing writing an article about smart contracts on ETH, BTC, and BCH. Collaborating with some people on CashScript use-cases.

Future work:

*Improve covenants workflow in CashScript.

Shammah Chancellor - CashWeb Keyserver

Currently working on:

*Backend services for CashWeb

Wants help with:

*Web design (currently looking for a web designer)

No submission this issue -

***Fernando Pelliccioni**

***Gabriel Cardona**

***Tom Zander**

***Axel Gembe**

***Antony Zegers**

***Darguval**

***Andrew Stone**

We invite **any developer** (working on BCH) who isn't featured in this issue to self report / submit what you are working on for the next issue of BCH Open-Source Pulse.

We're looking for:

*Recently completed

*Current work

*Future work (stuff you'd like to get to after you're done with your current work)

*Anything you'd like some help with

Please email us at bch.dev.mail@gmail.com