Notes (in progress) about P2P Web

Rasmus Erik Voel Jensen

2017

Abstract

Random ramblings and notes during the development of P2P Web.

WORK IN PROGRESS

These are preliminary notes, for own use.

General notes

Purpose:

- Infrastructure for no-server HTML5 apps => a decentralized trustless computer for the web

In short:

- Network topology: kademlia like, - address = hash of pubkey
- State/storage: each node stores a neighbourhood around its own address, saved in blockchain merkeltree
- Operations: changes to state are verifiable, and verified by nodes in neighbourhood
- Balance: nodes gets paid for doing tasks for the network, and can use this to buy tasks in the network. Also pay/payout for state blockchain.
- Tasks: stored, nodes assigned to tasks in deterministic random part of storage, proof-of-result stored, result stored, verification/value, balance updated.

Additional notes:

- WebPlatform: computations in webassembly. WebRTC as transport (thus modified kademlia). Crypto-algorithms from crypto.subtle.

- Neighbourhood size and amount of state per node - determined by node density (global minimum density / local density). Fixed amount of memory per node.
- Mutable references in blockchain (using balance to keep alive)
- Autonomous processes (using balance to keep alive)
- Not entire blockchain stored, only parts needed by the node
- Stake in computation tasks
- Balance/trade between processes
- Introduced 'errors' in blockchain, and bounties for finding/proving them.
- Binary/Quad merkel tree for proofs
- Pub/private-key derived from entropy source
- Task types: computation, storage, storage-transfer, find node with certain data
- Node trust / reliability proof via blockchain
- Block-tree rather than chain
- Computational task level of validation
- Result safety: added to state by any node in neighbourhood by proof of distance of computing-node to task.
- Computational task: computing time bound, and cost calculation.
- consensus algorithm: CRDT, additional data in timeinterval: after last block, before timed signature from other deterministic random node
- Tagged overlay network - opt-in part of infrastructure for tunable bandwidth requirements
- Network simulation (core optimised for low memory)
- Bandwidth optimised, - number of significant bits per node-id, stream compression, only send diffs etc.

Explore/ideas:

- Performance characteristics of current WebRTC implementations
- Performance effects of design choices for Kademlia-like algorithm on top of WebRTC (instead of UDP)
- Verifiable "computational" tasks, and economy based "computation".
- (Survey p2p overlay networks)
- WebRTC bootstrapping options (decentralised signalling server vs. actual node)
- Infrastructure deployment - bootstrap-code + load signed version of code from network, - partly test within network before full deploy.

Description of algorithm:

- nodes connected in kademlia-like structure
- regular state snapshot (blockchain merkel-dag)
    - divide-and-conqueor consensus algorithm, verifying credit

updates in neighbourhood.
    - each node stores the state of a neighbourhood around its own address, as well as the path to the root. The neighbourhood size is fixed for all, ensuring good redundancy of data for
- content of state
    - list of entities(nodes)
        * id
        * balance/credits (updated by work, tasks, cost of staying in blockchain, and transfers)
        * state (+ proof)
        * tasks - scheduled for execution - wager
        * result of previous scheduled task
        * work
            · stake
            · result
            · proof-of-work
        * state
- verifiable tasks
- task types
    - computation
    - storage
        * data
        * key/value
    - random verifications (of proof-of-stake tasks)
    - blockchain verification
- entities
    - nodes
    - nodes with stake
    - accounts (pub-key)
    - autonomous
- computational process
    - task gets stored in blockchain
    - task gets assigned to a number of bcrandom nodes
    - task gets done, and proof-of-work gets stored in the blockchain
    - task result gets released
    - result+proof-of-work get validated + signed into blockchain
    - balance is updated

Design criteria

- low bandwidth
- low memory footprint (useful for large simulation, as well as embeded systems)
- low code footprint
- tagging of hosts

- connect to arbitrary host
- foundation for other p2p applications

Roadmap

Implementation strategy:

- Mode of development
    1. Make prototype/proof-of-concept
    2. Measure performance (actual + simulated) + experiment with protocols
    3. Implement optimised version
- Levels of functionality. (subject to change)
    1. Simple p2p overlay network on top of webtechnology
    2. "Tagging" in overlay network (connections/DHT)
    3. Network state / map, - state blockchain updated divide-and-conquer.
    4. Tasks enqueued/executed in state
    5. Credit-balance in network, through proof of work
    6. Computing units, pubkey units, autonomous units/calculations.
    7. Add stake to computations for stronger security
- Auxiliary tasks:
    - Bootstrap gateway in php
    - Generate DSA from user-supplied entropy source.
    - Deployment system for new versions
    - (maybe paper Benchmark performance and limitations of WebRTC)
    - (maybe paper Review Kademlia optimisations from the viewpoint of webrtc networking model)
    - (maybe paper describing aspects of overall vision and architecture)
- Later
    - Optimisations for efficient p2p shared state (partial addresses, cached/multi-data hash proofs (quad instead of binary - same amount of data, but bette caching))

Writeups:

- Bottom-up
    - Algorithm for overlay network with taggable nodes
    - Kademlia for the Web Platform (Kademlia originally designed for IP/UDP, which have different performance characteristics than what is available on WebPlatform(WebRTC). Survey extensions / optimisations of Kademlie, and evaluate how they matches).

- Performance characteristics of p2p data on the web platform. (measure cost of initiating connection, limits / performance with many connection or data). Across the different platforms.
- Top-down
  - Design / execution model of a computational blockchain
  - Proveable tasks for for trustless computing
  - Definitions and concepts for reasoning about a p2p computing model

Possible publication targets

Ideas of places for publications:

- Open Access Journals (NB: https://doaj.org)
  - Ledger Journal
  - EAI Transactions on Scalable Information Systems
  - Computer Science (AGH)
- ACM/IEEE (pay for open access)
  - IEEE Transactions on Parallel and Distributed Systems
  - ACM Transactions on Computer Systems (TOCS)
- Conferences (in Europe)
  - 2017-10-30 2018-02-26/28 IFIP NTMS 1st International Workshop on Blockchains and Smart Contracts (BSC)
  - ~~2018-02-05 2018-06-20/22 DCAI 2018 : 15th International Conference on Distributed Con~~
  - (2018-02-11 2018-07-23/27 ACM Symposium on Principles of Distributed Computing)
- Central non-conference sources (based on related articles)
  - "The Computing Research Repository" (not a Journal, but arXiv, indexed in dblp etc.)
  - Peer-to-Peer Networking and Applications (Springer)
  - ~~Computer Networks (avoid Elsevier)~~

Literature

Various articles etc. that I should take a look add (and possibly add notes here eventually)

- (Teutsch and Reitwießner 2017)
- (Parno, Raykova, and Vaikuntanathan 2012)
- (Kaune et al. 2008)
- (Wood 2014)
- (Golem 2016)

- (Rhea et al. 2004)
- (Wang, Yang, and Chen 2010)
- (Benet 2014)
- (Maymounkov and Mazières 2002)
- (Gennaro, Gentry, and Parno 2010)
- (Medrano-Chávez, Pérez-Cortés, and Lopez-Guerrero 2015)
- (Ou et al. 2010)
- (Parno et al. 2013)
- (Baumgart and Mies 2007)
- (Jiménez, Osmani, and Knutsson 2011)
- (Surati, Jinwala, and Garg 2017)
- (Wood and Steiner 2016)
- (Schmid 2008)
- (Haas et al. 2017)

https://allquantor.at/blockchainbib/bibtex.html

Notes for later / optimised version

- page size
- typically 4K (getpagesize() is 4K on my linux, and that looks like common size via https://en.wikipedia.org/wiki/Page_%28computer_memory%29)
- webassembly 64K page size
- minimise memory usage (for ability to run large simulations).
- i.e. 64K per nodes => 100K nodes in memory simulation ~ 6G memory

Bibliography

Baumgart, Ingmar, and Sebastian Mies. 2007. "S/Kademlia: A Practicable Approach Towards Secure Key-Based Routing." In *13th International Conference on Parallel and Distributed Systems, ICPADS 2007, Hsinchu, Taiwan, December 5-7, 2007*, 1–8. doi:10.1109/ICPADS.2007.4447808.

Benet, Juan. 2014. "IPFS - Content Addressed, Versioned, P2P File System." *CoRR* abs/1407.3561. http://arxiv.org/abs/1407.3561.

Gennaro, Rosario, Craig Gentry, and Bryan Parno. 2010. "Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers." In *Advances in Cryptology – Crypto 2010: 30th Annual Cryptology Conference, Santa Barbara, ca, Usa, August 15-19, 2010. Proceedings*, edited by Tal Rabin, 465–82. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-14623-7_25.

Golem. 2016. "The Golem Project - Crowdfunding Whitepaper." http://www.golemproject.net/doc/DraftGolemProjectWhitepaper.pdf.

Haas, Andreas, Andreas Rossberg, Derek L. Schuff, Ben L. Titzer, Michael Holman, Dan Gohman, Luke Wagner, Alon Zakai, and J. F. Bastien. 2017. "Bringing the Web up to Speed with Webassembly." In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, Barcelona, Spain, June 18-23, 2017*, 185–200. doi:10.1145/3062341.3062363.

Jiménez, Raúl, Flutra Osmani, and Björn Knutsson. 2011. "Sub-Second Lookups on a Large-Scale Kademlia-Based Overlay." In *2011 IEEE International Conference on Peer-to-Peer Computing, P2P 2011, Kyoto, Japan, August 31 - September 2, 2011*, 82–91. doi:10.1109/P2P.2011.6038665.

Kaune, S., T. Lauinger, A. Kovacevic, and K. Pussep. 2008. "Embracing the Peer Next Door: Proximity in Kademlia." In *2008 Eighth International Conference on Peer-to-Peer Computing*, 343–50. doi:10.1109/P2P.2008.36.

Maymounkov, Petar, and David Mazières. 2002. "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric." doi:10.1007/3-540-45748-8_5.

Medrano-Chávez, Adán G., Elizabeth Pérez-Cortés, and Miguel Lopez-Guerrero. 2015. "A Performance Comparison of Chord and Kademlia Dhts in High Churn Scenarios." *Peer-to-Peer Networking and Applications* 8 (5): 807–21. doi:10.1007/s12083-014-0294-y.

Ou, Zhonghong, Erkki Harjula, Otso Kassinen, and Mika Ylianttila. 2010. "Performance Evaluation of a Kademlia-Based Communication-Oriented P2P System Under Churn." *Computer Networks* 54 (5): 689–705. doi:10.1016/j.comnet.2009.09.022.

Parno, Bryan, Jon Howell, Craig Gentry, and Mariana Raykova. 2013. "Pinocchio: Nearly Practical Verifiable Computation." In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, ca, Usa, May 19-22, 2013*, 238–52. doi:10.1109/SP.2013.47.

Parno, Bryan, Mariana Raykova, and Vinod Vaikuntanathan. 2012. "How to Delegate and Verify in Public: Verifiable Computation from Attribute-Based Encryption." In *Theory of Cryptography: 9th Theory of Cryptography Conference, Tcc 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, edited by Ronald Cramer, 422–39. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-28914-9_24.

Rhea, Sean C., Dennis Geels, Timothy Roscoe, and John Kubiatowicz. 2004.

"Handling Churn in a DHT." In *Proceedings of the General Track: 2004 USENIX Annual Technical Conference, June 27 - July 2, 2004, Boston Marriott Copley Place, Boston, Ma, USA*, 127–40. http://www.usenix.org/publications/library/proceedings/usenix04/tech/general/rhea.html.

Schmid, Stefan. 2008. "Dynamics and Cooperation: Algorithmic Challenges in Peer to Peer Computing." PhD thesis, ETH Zurich. https://www.net.t-labs.tu-berlin.de/~stefan/PhDStefan.pdf.

Surati, Shivangi, Devesh C. Jinwala, and Sanjay Garg. 2017. "A Survey of Simulators for P2p Overlay Networks with a Case Study of the P2p Tree Overlay Using an Event-Driven Simulator." *Engineering Science and Technology, an International Journal* 20 (2): 705–20. doi:http://dx.doi.org/10.1016/j.jestch.2016.12.010.

Teutsch, Jason, and Christian Reitwießner. 2017. "A Scalable Verification Solution for Blockchains." https://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf.

Wang, C., N. Yang, and H. Chen. 2010. "Improving Lookup Performance Based on Kademlia." In *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing*, 1:446–49. doi:10.1109/NSWCTC.2010.111.

Wood, Gavin. 2014. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." *Ethereum Project Yellow Paper*. https://github.com/ethereum/yellowpaper.

Wood, Gavin, and Jutta Steiner. 2016. "Trustless Computing—The What Not the How." In *Banking Beyond Banks and Money: A Guide to Banking Services in the Twenty-First Century*, edited by Paolo Tasca, Tomaso Aste, Loriana Pelizzon, and Nicolas Perony, 133–44. Cham: Springer International Publishing. doi:10.1007/978-3-319-42448-4_8.