Nama : Hamzah Fatihulhaq
NIM : 110319213

SUMMARIZE

CASPER POS

Selama beberapa tahun terakhir telah ada banyak penelitian tentang blockchain berbasis “bukti kepemilikan” (PoS) algoritma konsensus. Dalam sistem PoS, blockchain menambahkan dan menyetujui blok baru melalui proses di mana siapa pun yang memegang koin di dalam sistem dapat berpartisipasi, dan pengaruh yang dimiliki agen sebanding dengan jumlah koin (atau "taruhan (stacke)") yang dimilikinya. Ini adalah alternatif yang jauh lebih efisien untuk "penambangan" proof of work (PoW) dan memungkinkan blockchain untuk beroperasi tanpa perangkat keras dan biaya listrik penambangan yang tinggi. Ada dua aliran pemikiran utama dalam desain PoS. *proof of stake* pertama, berbasis rantai, meniru *proof of work* dan menampilkan rantai blok dan mensimulasikan penambangan dengan secara acak memberikan hak untuk membuat blok baru untuk pemangku kepentingan (*stakeholders*).

Peserta protokol mengikuti protokol dengan jujur, maka, terlepas dari latensi jaringan, algoritme tidak dapat menyelesaikan blok yang saling bertentangan. Penggunaan kembali algoritma BFT untuk proof of stake pertama kali diperkenalkan oleh Tendermint, dan memiliki inspirasi modern seperti, Casper yang mengikuti penggunaan BFT, meskipun dengan beberapa modifikasi.

Casper the Friendly Finality Gadget adalah overlay di atas mekanisme proposal dengan mekanisme yang mengusulkan blok1. Casper bertanggung jawab untuk menyelesaikan blok-blok ini, pada dasarnya memilih rantai unik yang mewakili transaksi kanonik dari buku besar. Casper memberikan keamanan, tetapi keaktifan tergantung pada mekanisme proposal yang dipilih. Artinya, jika penyerang sepenuhnya mengendalikan mekanisme proposal, Casper melindungi dari penyelesaian dua pos pemeriksaan yang saling bertentangan, tetapi penyerang dapat mencegah Casper menyelesaikan pos pemeriksaan di masa mendatang.

Casper memperkenalkan beberapa fitur baru yang belum tentu didukung oleh algoritma BFT:

- **Accountability**. Jika validator melanggar aturan, Casper dapat mendeteksi pelanggaran dan mengetahui validator mana yang melanggar aturan. Akuntabilitas memungkinkan untuk menghukum validator yang salah, memecahkan masalah "tidak ada yang dipertaruhkan" yang mengganggu PoS berbasis rantai. Hukuman untuk pelanggaran aturan adalah seluruh deposit validator. Penalti maksimal ini merupakan pembelaan terhadap pelanggaran protokol. Karena keamanan *proof of stake* didasarkan pada ukuran penalti, yang dapat diatur jauh melebihi keuntungan dari hadiah penambangan, *proof of stake* memberikan insentif keamanan yang lebih kuat daripada *proof of work*.
- **Dynamic validators**. Cara aman untuk set validator untuk berubah dari waktu ke waktu.
- **Defenses**. Cara pertahanan terhadap serangan revisi jarak jauh serta serangan di mana lebih dari validator berhenti offline, dengan mengorbankan asumsi sinkronisitas tradeoff yang sangat lemah.
- **Modular overlay**. Desain Casper sebagai overlay membuatnya lebih mudah untuk diterapkan sebagai peningkatan ke *proof of work* yang ada.

The Casper Protocol

Di dalam Ethereum, mekanisme proposal pada awalnya akan menjadi bukti rantai kerja yang ada, menjadikan versi pertama Casper sebagai sistem PoW/PoS hybrid. Di versi mendatang mekanisme proposal PoW akan diganti dengan yang lebih efisien. Misalnya, kita dapat membayangkan mengubah proposal blok menjadi semacam Skema penandatanganan blok round-robin PoS.

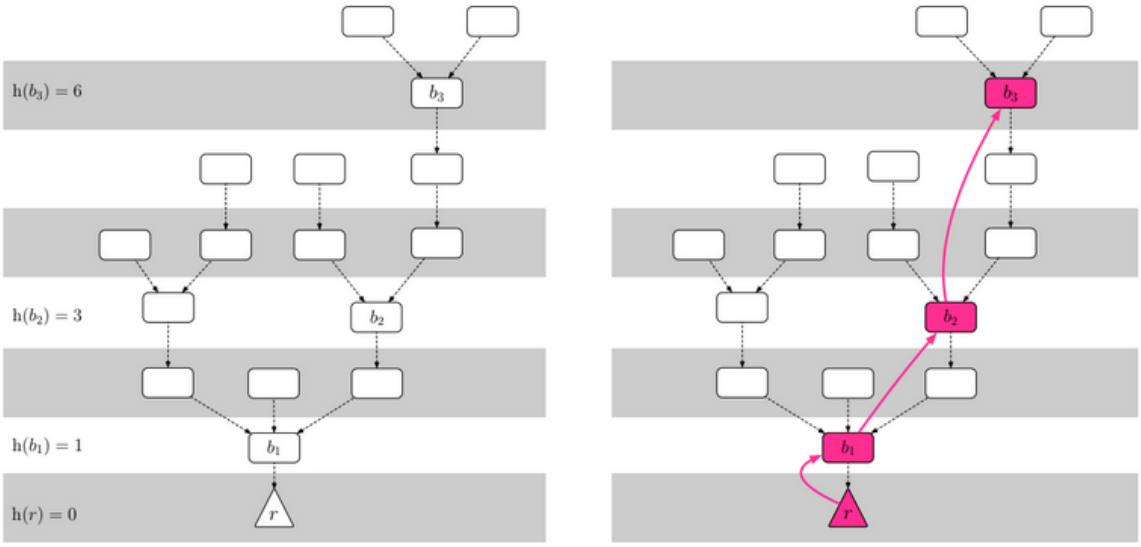
Dalam versi Casper yang sederhana ini, diasumsikan bahwa ada seperangkat validator dan mekanisme proposal yang tetap (misalnya, mekanisme proposal bukti kerja yang sudah dikenal) yang menghasilkan blok anak dari blok yang ada, membentuk pohon blok yang terus tumbuh. Dari akar pohon biasanya disebut “genesis block”.

Dalam keadaan normal, kami berharap bahwa mekanisme proposal biasanya akan mengusulkan blok satu demi satu dalam daftar tertaut (yaitu, setiap blok "induk" memiliki tepat satu blok "anak"). Tetapi dalam kasus latensi jaringan atau serangan yang disengaja, mekanisme proposal pasti terkadang akan menghasilkan banyak anak dari orang tua yang sama. Tugas Casper adalah memilih satu anak dari setiap orang tua, sehingga memilih satu rantai kanonik dari pohon balok.

Daripada berurusan dengan pohon blok penuh, untuk tujuan efisiensi Casper hanya mempertimbangkan subpohon dari pos pemeriksaan yang membentuk pohon pos pemeriksaan. Blok genesis adalah pos pemeriksaan, dan setiap blok yang tingginya di pohon blok (atau nomor blok) adalah kelipatan tepat 100 juga merupakan pos pemeriksaan. “Tinggi pos pemeriksaan” sebuah balok dengan tinggi balok 100 k adalah k; ekuivalen, tinggi h(c) dari pos pemeriksaan c adalah jumlah elemen dalam rantai pos pemeriksaan yang membentang dari c (non-inklusif) ke akar di sepanjang tautan induk.

Setiap validator memiliki deposit; ketika validator bergabung, depositnya adalah jumlah koin yang disimpan. Setelah bergabung, setoran masing-masing validator naik dan turun dengan hadiah dan penalti. Bukti keamanan pasak berasal dari ukuran setoran, bukan jumlah validator, jadi untuk sisa makalah ini, ketika kami mengatakan “2/3 validator”, kami mengacu pada fraksi tertimbang setoran; yaitu, satu set validator yang jumlah jumlah setorannya sama dengan 2/3 dari total ukuran setoran dari seluruh set validator.

Validator dapat menyiarkan pesan suara yang berisi empat informasi: dua pos pemeriksaan s dan t bersama dengan ketinggiannya h(s) dan h(t). Kami mengharuskan s menjadi nenek moyang dari t di pohon pos pemeriksaan, jika tidak, suara dianggap tidak valid. Jika kunci publik validator tidak ada dalam set validator, suara dianggap tidak sah. Bersama dengan tanda tangan dari validator, kami akan menulis suara ini dalam bentuk hv, s, t, h(s), h(t)i.



(b) The height function

(c) The justified chain $r \rightarrow b_1 \rightarrow b_2 \rightarrow b_3$

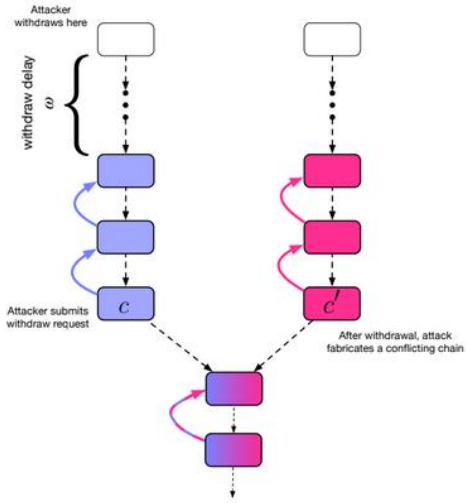
Enabling Dynamic Validator Sets

Himpunan validator harus dapat berubah. Validator baru harus dapat bergabung, dan validator yang ada harus dapat keluar. Untuk mencapai ini, didefinisikan dinasti sebuah blok. Dinasti blok b adalah jumlah pos pemeriksaan akhir dalam rantai dari root ke induk blok b. Ketika pesan deposit calon validator termasuk dalam blok dengan dinasti d, maka validator akan bergabung dengan set validator pada blok pertama dengan dinasti d + 2. Kami menyebut d + 2 dinasti awal validator ini, DS(v).

Stopping Attacks

Ada dua serangan terkenal terhadap sistem proof-of-stake: long range revisions dan catastrophic crashes

1. **Long Range Revisions**
   Penundaan penarikan setelah dinasti akhir validator memperkenalkan asumsi sinkronisitas antara validator dan klien. Setelah koalisi validator telah menarik simpanan mereka, jika koalisi tersebut memiliki lebih dari 2/3 simpanan di masa lalu, mereka dapat menggunakan supermayoritas historis mereka untuk menyelesaikan pos pemeriksaan yang bertentangan tanpa takut dipotong (karena mereka telah menarik uang mereka) . Ini disebut serangan revisi jarak jauh.
   Dalam istilah sederhana, long-range attacks dicegah dengan aturan pilihan garpu untuk tidak pernah mengembalikan blok yang telah diselesaikan, serta harapan bahwa setiap klien akan "masuk" dan mendapatkan tampilan lengkap terkini dari rantai di beberapa frekuensi reguler (misalnya, sekali per 1-2 bulan). Garpu "long range revision" yang menyelesaikan blok yang lebih lama dari itu akan diabaikan begitu saja, karena semua klien telah melihat blok yang diselesaikan pada ketinggian itu dan akan menolak untuk mengembalikannya.



2. Castastrophic Crashes

Misalkan > 1/3 validator crash-fail pada saat yang sama—yaitu, mereka tidak lagi terhubung ke jaringan karena partisi jaringan, kegagalan komputer, atau validator itu sendiri berbahaya. Secara intuitif, mulai saat ini, tidak ada tautan supermayoritas yang dapat dibuat, dan dengan demikian tidak ada pos pemeriksaan di masa mendatang yang dapat diselesaikan.

Kita dapat memulihkan dari ini dengan melembagakan "kebocoran tidak aktif" yang secara perlahan menguras deposit validator mana pun yang tidak memilih pos pemeriksaan, sampai akhirnya ukuran depositnya berkurang cukup rendah sehingga validator yang memberikan suara adalah supermayoritas. Rumus yang paling sederhana adalah sesuatu seperti "di setiap zaman validator dengan ukuran setoran D gagal memilih, ia kehilangan D p (untuk 0 < p < 1)", meskipun untuk mengatasi crash besar lebih cepat rumus yang meningkatkan tingkat kebocoran di peristiwa rentetan panjang blok yang belum selesai mungkin optimal.

Kebocoran tidak aktif memperkenalkan kemungkinan dua pos pemeriksaan yang saling bertentangan diselesaikan tanpa validator yang dipotong (seperti pada Gambar 6), dengan validator hanya kehilangan uang hanya pada satu dari dua pos pemeriksaan. Asumsikan validator dibagi menjadi dua subset, dengan subset VA voting pada rantai A dan subset VB voting pada rantai B. Pada rantai A, deposit VB akan bocor, dan sebaliknya, menyebabkan setiap subset memiliki supermajority pada rantai masing-masing, memungkinkan dua pos pemeriksaan yang saling bertentangan untuk diselesaikan tanpa ada validator yang secara eksplisit dipotong (tetapi setiap subset akan kehilangan sebagian besar deposit mereka di salah satu dari dua rantai karena kebocoran). Jika situasi ini terjadi, maka setiap validator harus memilih pos pemeriksaan akhir apa pun yang dilihatnya terlebih dahulu.

Algoritma yang tepat untuk pulih dari berbagai serangan ini tetap menjadi masalah terbuka. Untuk saat ini, kami menganggap validator dapat mendeteksi perilaku yang jelas-jelas menyimpang (misalnya, tidak menyertakan bukti) dan secara manual membuat " minority soft fork". Garpu minoritas ini dapat dilihat sebagai blockchain dalam dirinya sendiri yang bersaing dengan rantai mayoritas di pasar, dan jika rantai

mayoritas benar-benar dioperasikan oleh penyerang jahat yang berkolusi maka kita dapat berasumsi bahwa pasar akan menyukai garpu minoritas.