Background

Will Martino (me)

• Focus mostly on consensus + formal verification
• At JPM: Juno Lead
• SEC: Founding Tech Lead for the Distributed Ledger Technology Working Group

Kadena

• Founded in 2016 by the former Lead Engineers for JP Morgan Blockchain group
• Announcing our public blockchain project (Launch ETA Q4’18)
# Kadena Stack

<table>
<thead>
<tr>
<th></th>
<th>Public Blockchain</th>
<th>Enterprise Blockchain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consensus Mechanism</td>
<td><strong>Chainweb</strong></td>
<td><strong>ScalableBFT</strong></td>
</tr>
<tr>
<td>Smart Contracts</td>
<td><strong>Pact Smart Contract Language</strong></td>
<td></td>
</tr>
<tr>
<td>Development Tooling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Atom/Emacs Integration
- Unit Tests
- Local Dev Server
- Opt-In Hindley-Milner Type Interference
- Formal Verification
Chainweb

Overview

Chainweb
• Parallel-, egalitarian-chain PoW protocol
• Same energy, higher throughput, lower latency

Key Features
1. Peer chains cross reference each other’s previous Merkle roots
2. Trustless destructive cross-chain coin transfers through SPV at the smart contract level
Background

Similar/Prior Art

• Blockrope/Betacoin
  Cross-referencing chains “twisted” together

• GHOST/Decor+
  Including orphaned header’s for increased security/lower confirmation times
Two-Step

- **Delete** coins on Chain A. Specify:
  - Delete Acct on Chain 1 & Amount
  - Create Chain & Create Acct

- **Create** requires a Merkle proof of deletion
  - Smart contract runs SPV, gets Amount and Acct from proof
  - Consume Delete’s Transaction ID
Cross-Chain Destructive Coin Transfers

Create Chain

Delete Chain

<table>
<thead>
<tr>
<th>Block Height</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chain 1 (Create Chain)</td>
<td>Alice: 0</td>
<td>Alice: 0</td>
<td>Alice: 0</td>
<td><strong>Alice: 10</strong></td>
</tr>
<tr>
<td>Chain 2 (Delete Chain)</td>
<td>Alice: 10</td>
<td><strong>Alice: 0</strong></td>
<td>Alice: 0</td>
<td>Alice: 0</td>
</tr>
</tbody>
</table>
My parent and my peer reference refer to my grandparent

My parent and my peer reference refer to my peer’s grandparent
Traditional Blockchain

Merkle Propagation by Layer

Base Graph

Whole Network

Graph Edge Path for a Single Example Block

Header Reference for a Single Example Block
Two Chain Configuration

Cross-Chain Header Reference

Graph Edge Dictating Peer Reference Requirements
Three Chain Configuration
Terms

• **Vertex**: “node” in the graph
• **Edge**: path in the graph
• **Degree**: number of edges a vertex has
• **Order**: number of vertices in a graph
• **Diameter**: the length of the longest, shortest path between vertices
Base Graph as Braid Pattern

- Every vertex is a chain
- Every edge dictates what peer’s previous Merkle roots are required, in addition to their own previous root, when constructing a new header

We use known solutions to the degree-diameter problem for our base graphs
Ten Chain Configuration
Twenty Chain Configurations
Chainweb

**Throughput**

All Known Degree-Diameter Solutions Work

- Assuming 10 tx per sec per chain
- Range from 10 to 10,000,000,000 chains
- We aim to launch between 1k-10k tx per sec

**Congestion Free**

- Crypto Kitties only clogs a subset of chains
Past and Future Merkle Cones
Merkle Cones

- What a reorg impacts
- Dictate the limit of how far ahead/behind chains can get
- Hashrate naturally pools to the lagging chain

Merkle Mass ($\mu_2$)

- Merkle cones have mass, buries a block in cryptographic assurance
- Increases “exponentially” from 1 to order
Hypothesis

• Once a block’s future Merkle cone is formed, a transaction is “confirmed”

Calculations

• Work in progress & looking for collaborators:
  • Attacking the full braid
  • Closed Form Merkle Cone Attack
  • Monte Carlo Simulation
Chainweb Considerations

Configurable Confirmation Latency
- Block Time * Base Graph Diameter

Bandwidth
- Heavy: Whole Chainweb Block Stream
- Light: Single Chain Block Stream
- Light: Chainweb Network Headers Stream

Assurance is captured in the header stream
Considerations

Subset Replication and Mining

- Possible to mine a subset of the network

The Role of Large Mining Pools

- Hashrate load balancers
- Header stream network providers
- Layer consensus providers
Considerations

Cross-Chain Account to Account Transfers

• Take the same amount of time (diameter) as a same-chain transfer

• Well formed Delete is enough, Create at your leisure

Load Balancing Smart Contracts

• Cross-chain transfer tools work for contract assets as well
Thank You!

Will Martino will@kadena.io

http://kadena.io