CS 168: Blockchain and Cryptocurrencies

Mining Pools

Prof. Tom Austin

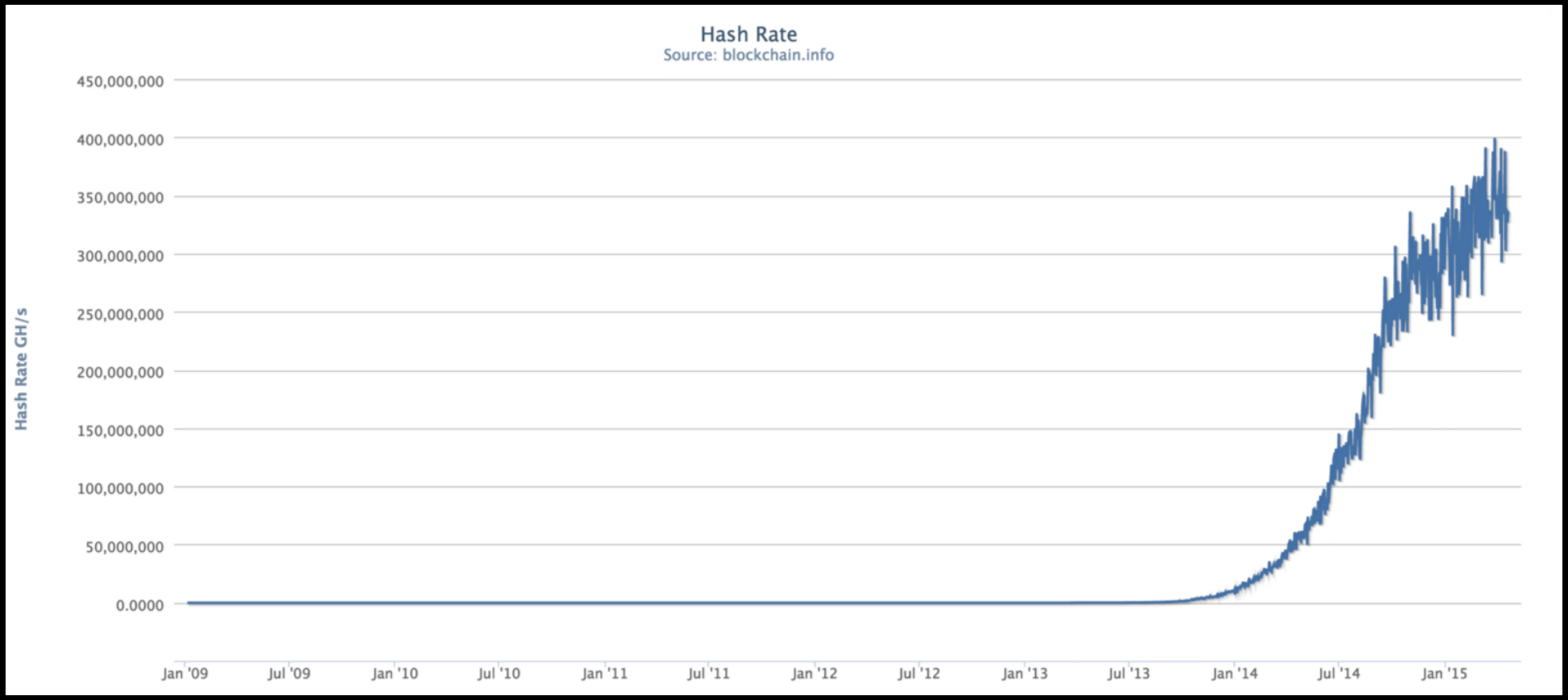
San José State University

Review: Bitcoin mining

- Miners verify transactions
  - Must find a proof-of-work.
  - Reward: newly generated bitcoins, plus transaction fees.
- One-CPU-one-vote
- Key to Bitcoin's decentralization

Bitcoin Mining Rig, 2009

Bitcoin Hash Rate Over Time



Bitcoin Mining Rig(s), today

Application-Specific Integrated Circuit (ASIC)

- Speed up mining
- Expensive
- Off-the-shelf hardware cannot compete
  - On the other hand, neither can botnets
- **Result: mining is less decentralized**

Mining Pools

- Share rewards for steadier payout
  - Less BTC now > more BTC later
- Operator collects fee for coordinating
- A variety of mining pool schemes exist
- Meni Rosenfeld, *Analysis of Bitcoin Pooled Mining Reward Systems*, 2011.

Problem:

How does the operator trust its miners?

Mining Pool "Shares"

- Coordinator determines blocks.
- Miners report "shares" to coordinator.
  - "Near misses"
  - Easier target (about 1000x easier)
- So share of mining rewards are determined by a PoW.

Pay-Per-Share (PPS)

- Operator *immediately* pays for shares
- Operator absorbs cost of paying shares
- Immune to pool hopping attacks
  - Discussed later
- Higher operator overhead (fees)

Lab, Part 1: PPS Mining Pool in SpartanGold

- Download
  - driver.js
  - pool-operator.js
  - pool-miner.js
- Run driver.js to see the simulation
- Review operator and pool operator code

Proportional Reward (PROP)

- Finding shares earns *future* rewards
- Everyone gets paid when proof is found
- Vulnerable to *pool hopping attack*

Pool Hopping Attack

A miner can:

- Find a share early in the search for a new block.
- Continue to benefit from the mining work of other miners in the pool.
- Switch to a new pool.

Rewards exceed the rewards of playing fairly.

Pool Hopping Example
(in-class)

Lab, Part 2: Implement PROP Pool Operator

- Extend `PoolOperator` to make `PropPoolOperator`
- Override `rewardMiner` and `payRewards` methods to implement PROP strategy.

Pay Per Last N Shares (PPLNS)

Same as PROP, except …

- Rewards are only paid to the last N shares
- Introduces time element, but keeps operator overhead low
- Dis-incentivizes pool hopping
- Dominant approach today

Other attacks

- *Sabotage* (aka vigilante attack)
  - Miner submits shares, but throws away proofs
  - Miner benefits when pool wins
  - Pool does not benefit from miner's hashing power
- *Lie in wait* attack. A miner in a pool:
  - Finds a valid block, but withholds it
  - Searches for a bunch of additional shares
  - Announces block after gathering extra shares

Lab, Part 3: Implement PPLNS Pool Operator

- Extend `PoolOperator` to make `PplnsPoolOperator`
- Override `rewardMiner` and `payRewards` methods to implement PPLNS strategy.