BlockSci

Design and applications of a blockchain analysis platform

Harry Kalodner, **Malte Möser,** Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, Arvind Narayanan

Why build a blockchain analysis tool?

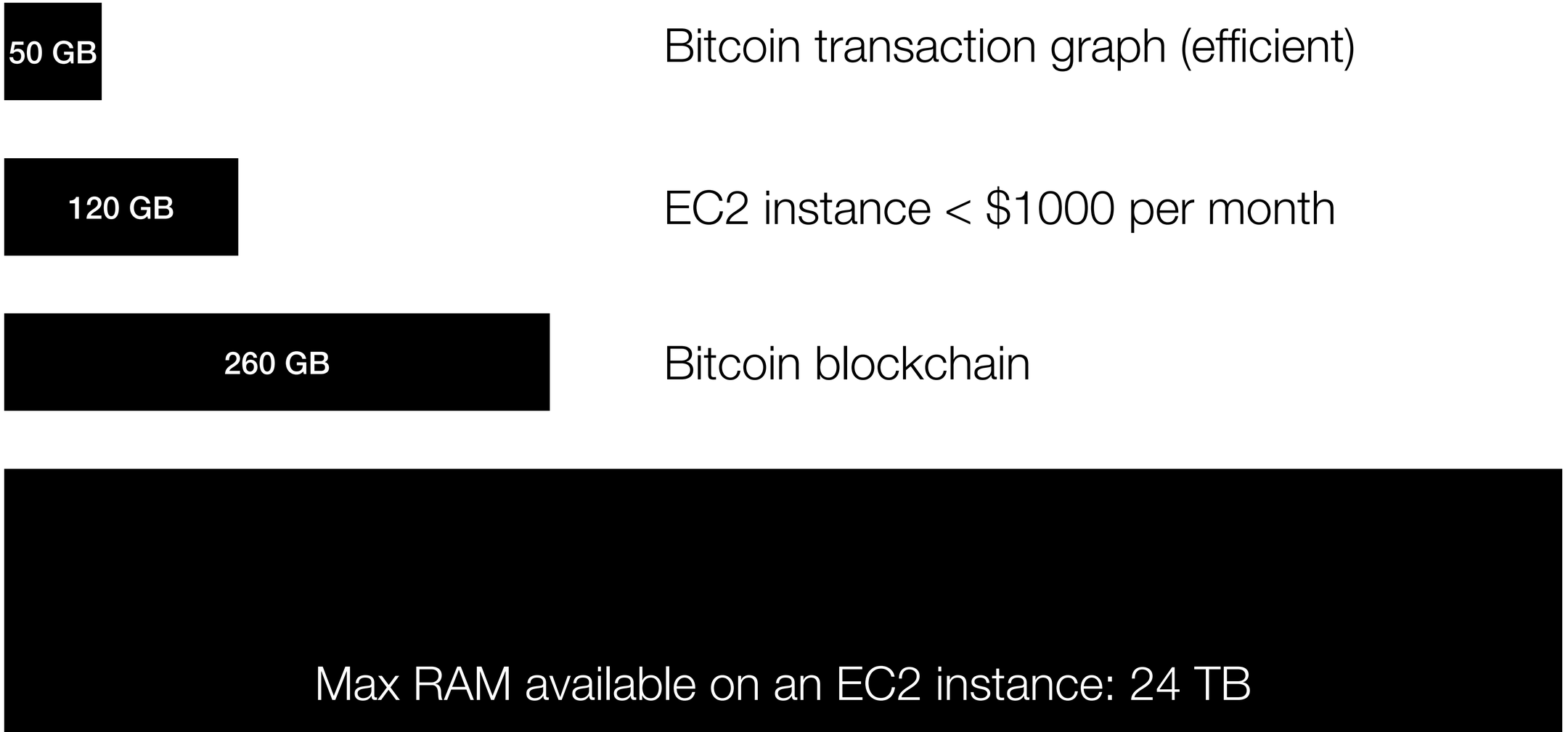
Blockchain data

- Unprecedented research corpus of financial transactions
- Interesting for scientific analyses and commercial applications

Tools to analyze blockchain data

- Commercial tools are often tailored towards specific use cases
- Lack of general-purpose tools

Blockchain data is “small”



Goals for BlockSci

Performance

in-memory database

domain-specific optimizations

Capabilities

supports different blockchains

analytic tools included

Usability

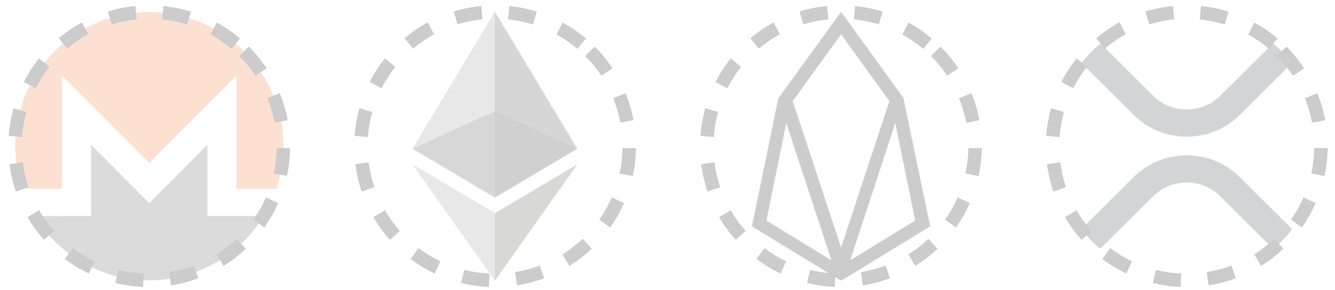
C++ / Python

fluent interface

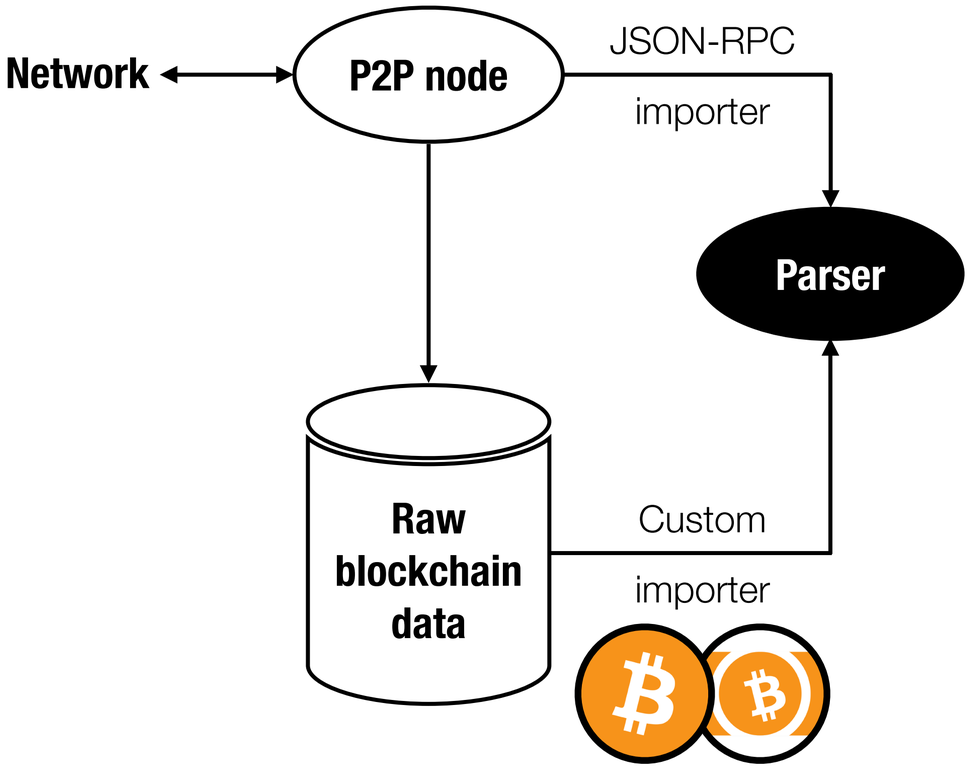
Which blockchains should we support?

Supported: design similar to Bitcoin (to varying degree)

Unsupported: different design and/or interface

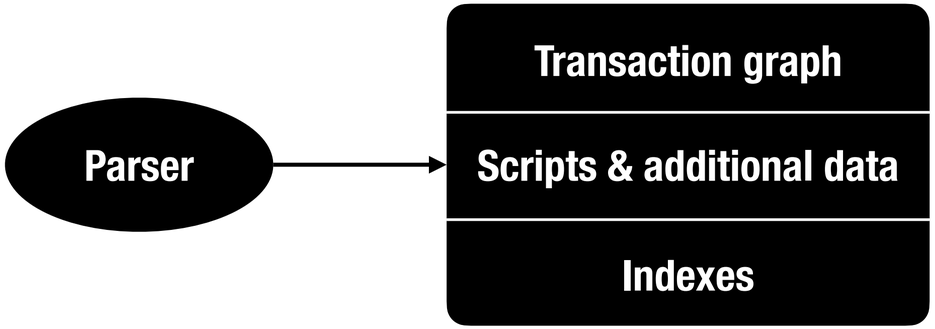


Building an efficient parser



- Incremental updates
- Parallelized pipeline to resolve arbitrary transaction ordering in blocks
- Optimized address lookups
- Multi-chain mode

Data layout: tradeoff between memory efficiency and performance



BlockSci Data

- Core transaction graph data optimized for sequential analyses
- Additional data stored in hybrid format, loaded on-demand
- Address data is deduplicated across different address types
- Indexes for common lookups

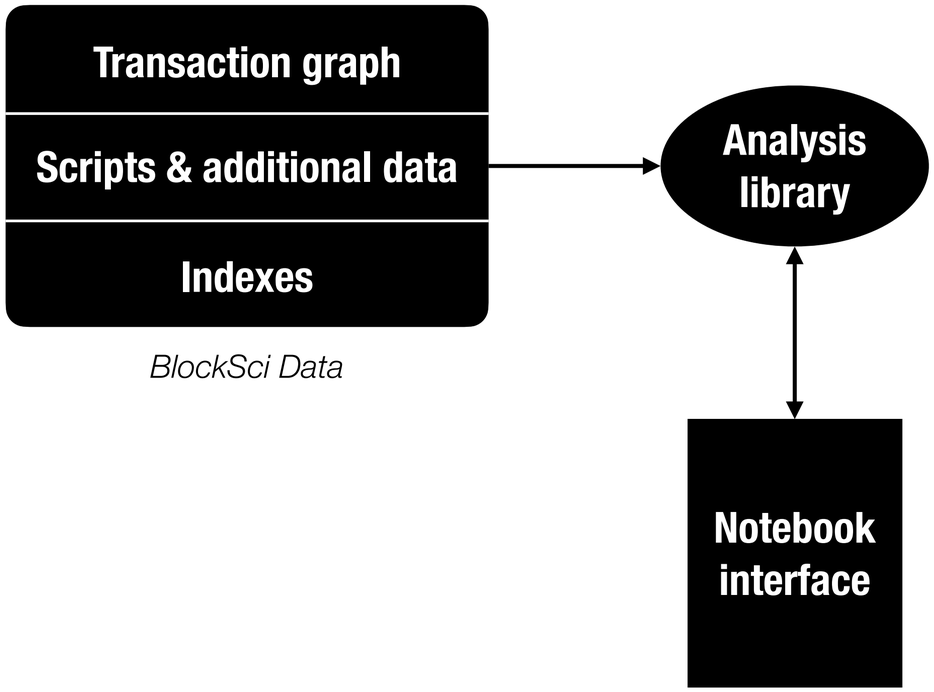
BlockSci is fast

Iterating over transaction inputs and outputs in C++

11.3 seconds	single-threaded
0.9 seconds	multithreaded (16 vCPUs)

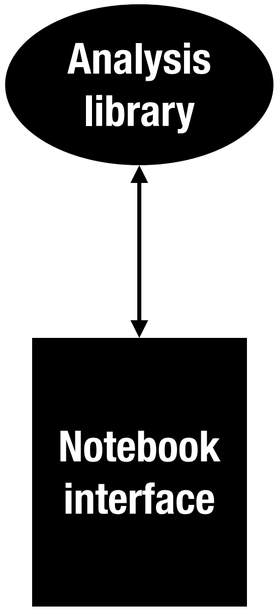
Iterating over 610,695 Bitcoin blocks on an r5.4xlarge EC2 instance (16 vCPUs, 2.5 GHz Intel Xeon Platinum 8175M, 128 GiB memory, 800 GiB EBS volume)

Analysis library



- Provides static view on data that can be updated on disk
- Map-reduce abstractions
- Forensic capabilities (clustering, change address heuristics)
- Can collect P2P network timestamps

Programmer interface



Querying the blockchain in Python

```
[tx for block in chain for tx in block if tx.fee > 10**7]
```

Fluent interface: expressiveness & speed

- Queries are specified in Python, executed in C++
- Many operators: select, filter, group_by, min/max, …

```
chain.blocks.txes.where(lambda tx: tx.fee > 10**7).to_list()
```

Example application

Unfortunate effects of multisignature use

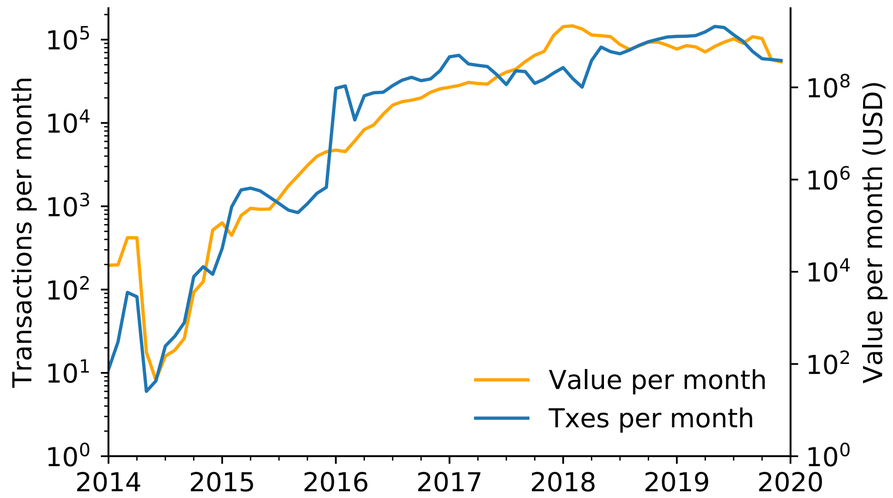
Example: drawbacks of multisig

Confidentiality

- Transactions specify list of keys and threshold
- Publicly exposes access control changes

Privacy

Security



Example: drawbacks of multisig

Confidentiality

Privacy

- Different output type allows to distinguish multisig from single-key users
- May identify change output in up to 122 million transactions

Security

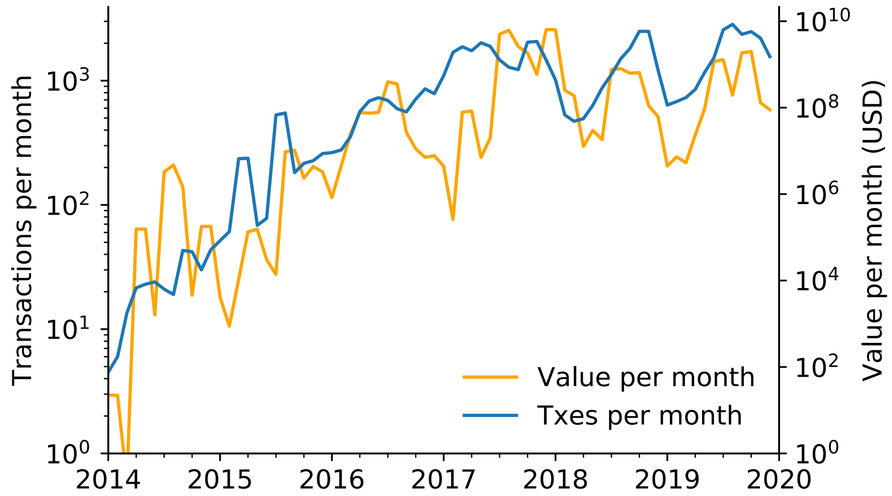
Example: drawbacks of multisig

Confidentiality

Privacy

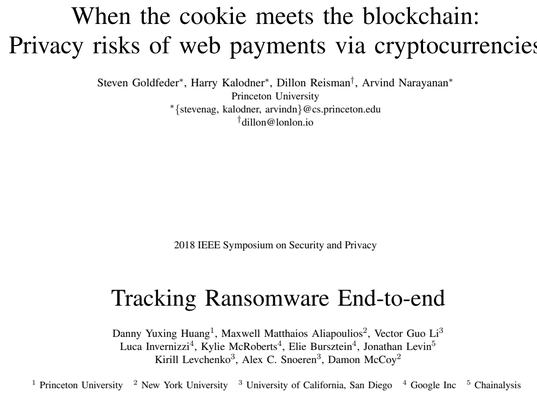
Security

- Pattern: Multisig → Regular → Multisig
- Temporary reduction in security



BlockSci has been widely used

- At least 9 peer-reviewed articles and 6 preprints
- Educational tool for workshops and lectures
- Foundation for more specialized blockchain analysis tools
- Active on GitHub



When the cookie meets the blockchain:
Privacy risks of web payments via cryptocurrencies

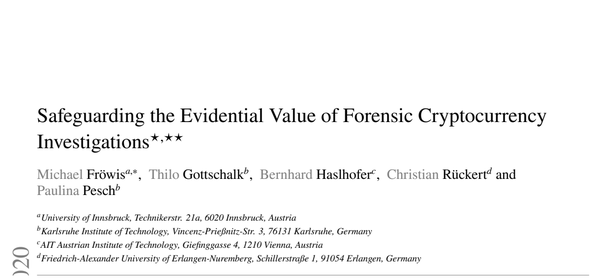
Steven Goldfeder*, Harry Kalodner*, Dillon Reisman†, Arvind Narayanan*
Princeton University
*{stevenag, kalodner, arvindn}@cs.princeton.edu
†dillon@lonlon.io

2018 IEEE Symposium on Security and Privacy

Tracking Ransomware End-to-end

Danny Yuxing Huang[1], Maxwell Matthaios Aliapoulios[2], Vector Guo Li[3]
Luca Invernizzi[4], Kylie McRoberts[4], Elie Bursztein[4], Jonathan Levin[5]
Kirill Levchenko[3], Alex C. Snoeren[3], Damon McCoy[2]

[1] Princeton University [2] New York University [3] University of California, San Diego [4] Google Inc [5] Chainalysis



Safeguarding the Evidential Value of Forensic Cryptocurrency Investigations⋆,⋆⋆

Michael Fröwis[a,*], Thilo Gottschalk[b], Bernhard Haslhofer[c], Christian Rückert[d] and Paulina Pesch[b]

[a]University of Innsbruck, Technikerstr. 21a, 6020 Innsbruck, Austria
[b]Karlsruhe Institute of Technology, Vincenz-Prießnitz-Str. 3, 76131 Karlsruhe, Germany
[c]AIT Austrian Institute of Technology, Giefinggasse 4, 1210 Vienna, Austria
[d]Friedrich-Alexander University of Erlangen-Nuremberg, Schillerstraße 1, 91054 Erlangen, Germany

universität innsbruck

Blockchain-Daten analysieren mit

BlockSci

"A high-performance tool for blockchain science and exploration"

Martin Plattner, Universität Innsbruck
martin@mplattner.at

MALTE MÖSER

HERZLICH WILLKOMMEN ZUM BLOCKSCI WORKSHOP

More details in our paper!

- Architecture and optimizations
- Performance measurements and comparison
- Multi-chain mode
- Three more applications
  - Cross-chain privacy
  - Effectiveness of transaction fee estimation
  - Velocity of cryptocurrencies

BlockSci: Design and applications of a blockchain analysis platform

ARTIFACT EVALUATED usenix ASSOCIATION PASSED

Harry Kalodner* *Princeton University*
Malte Möser* *Princeton University*
Kevin Lee *Princeton University*
Steven Goldfeder *Cornell Tech*
Martin Plattner *University of Innsbruck*
Alishah Chator *Johns Hopkins University*
Arvind Narayanan *Princeton University*

Abstract

Analysis of blockchain data is useful for both scientific research and commercial applications. We present BlockSci, an open-source software platform for blockchain analysis. BlockSci is versatile in its support for different blockchains and analysis tasks. It incorporates an in-memory, analytical (rather than transactional) database, making it orders of magnitudes faster than using general-purpose graph databases. We describe BlockSci's design and present four analyses that illustrate its capabilities, shedding light on the security, privacy, and economics of cryptocurrencies.

1 Introduction

Public blockchains constitute an unprecedented research corpus of financial transactions. Bitcoin's blockchain alone is 260 GB as of December 2019.[1] This data holds the key to measuring the privacy of cryptocurrencies in practice, studying user behavior with regards to security and economics, or understanding the non-currency applications that use the blockchain as a database.

We present BlockSci, a software platform that enables the science of blockchains. It addresses three pain points of existing tools: poor performance, limited capabilities, and a cumbersome programming interface. Compared to the use of general-purpose graph databases, BlockSci is hundreds of times faster for sequential queries and substantially faster for all queries, including graph traversal queries. It comes bundled with analytic modules such as address clustering, exposes different blockchains through a common interface, collects "mempool" state and imports exchange rate data, and gives the programmer a choice of interfaces: a Jupyter notebook for intuitive exploration and C++ for performance-critical tasks. In contrast to commercial tools, BlockSci is not tailored to specific use cases such as criminal investigations or insights for cryptocurrency traders. Instead, by providing efficient and convenient programmatic access to the full blockchain data, it enables a wide range of reproducible, scientific analyses.

BlockSci's design starts with the observation that blockchains are append-only databases; further, the snapshots used for research are static. Thus, the ACID properties of transactional databases are unnecessary. This makes an in-memory analytical database the natural choice. On top of the obvious speed gains of memory, we apply a number of tricks such as converting hash pointers to actual pointers and deduplicating address data, which further greatly increase speed and decrease the size of the data. We plan to scale vertically as blockchains grow, and we expect that this will be straightforward for the foreseeable future, as commodity cloud instances currently offer up to a *hundred times* more memory than required for loading and analyzing Bitcoin's blockchain. Avoiding distributed processing is further motivated by the fact that blockchain data is graph-structured, and thus hard to partition effectively. In fact, we conjecture that the use of a traditional, distributed transactional database for blockchain analysis has infinite COST (Configuration that Outperforms a Single Thread) [1], in the sense that no level of parallelism can outperform an optimized single-threaded implementation.

BlockSci comes with batteries included. First, it is not limited to Bitcoin: a parsing step converts a variety of blockchains into a common, compact format. Currently supported blockchains include Bitcoin, Bitcoin Cash, Bitcoin SV, Litecoin, and Zcash (Section 2.1). A multi-chain mode optimizes for user-friendly and memory-efficient analyses of forked blockchains together with their parent chain. Smart contract platforms such as Ethereum are outside our scope.

Second, BlockSci includes a library of useful analytic tools, such as identifying special transactions (e.g., CoinJoin) and linking addresses to each other based on well-known heuristics, including across forked chains (Section 2.4). Third, BlockSci can record the time of transaction broadcasts on the peer-to-peer network and expose them through the same interface. Similarly, we make (historical and current) data on

*These authors contributed equally to this work.
[1] All numbers in this paper are current as of December 2019, and analyses of the Bitcoin blockchain as of block height 610,695, unless stated otherwise.

Thank you

Code & questions

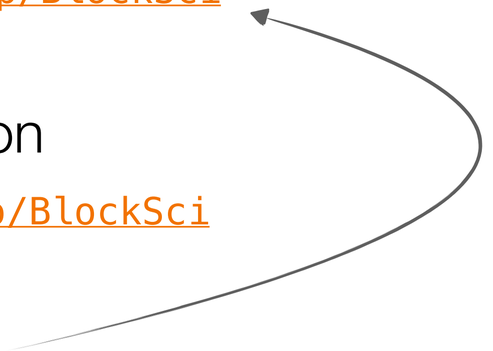
https://github.com/citp/BlockSci

Documentation

https://citp.github.io/BlockSci

Contact

blocksci@lists.cs.princeton.edu



Technical & how-to questions

BlockSci

Design and applications of a blockchain analysis platform

Harry Kalodner, **Malte Möser,** Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, Arvind Narayanan