vacuumlabs

Cardano/Chain indexers

What, Why, How

Martin Mikšaník

What

Provide inspection data of the blockchain, including blocks, transactions, UTxOs, datums, and more.

Input UTxO is passed by reference - data are missing

Online inspectors:
https://cexplorer.io/
https://cardanoscan.io/

Why

- Get the balance of ADA: Retrieve all UTxOs and sum the amounts for the filtered address.
- Get a list of NFTs in a collection.
- Get a list of active contracts.
- Get owners of tokens
- Get datum by hash
- Get address of script

- Get a list of contracts for a specific user: Datum inspection is required.
- Get detailed information about contract dependency: More UTxOs are needed.

How

- Ask directly cardano nodes
- External services - BlockFrost
- Full indexers - DBSync, Carp
- Thin indexers - Kupo
- Custom indexer - ???

Ask cardano node

- Difficult to request a foreign node; a local copy is required.
- Queries are very slow because nodes are not optimized for this use case.
- Analytics is challenging and requires re-storing data in a more suitable format.

vacuumlabs

External service like BlockFrost

- Very easy to use
- No backend required

- Third-party service with vendor lock.
- Does not support more complex queries.
- How can analytics be performed without locally re-storing data?

vacuumlabs

Full indexers like DBSync/Carp

- Easy to use.
- Lot of resources are required
- The entire blockchain is stored in the database, enabling support for complex queries.
- No need to re-fetch nodes when service changes occur.

Questions:

- Custom triggers and plugins?
- How can application data be mapped to this database?
- How can real-time data (e.g., transaction information) be retrieved?
- Is it possible to develop our application as a plugin?
- What happens when schema changes are required?

Thin indexers like Kupo

- Easy to use
- Stored only UTxO matching filter

- Re-fetch from node is needed for service changes
- Custom triggers/plugins?
- Similar like Full Indexer

Filtered block/transactions

- Cardano node is required
- You can write custom set of filter for block/transaction/utxo
- You can feed MQ, APIs, etc.
- Custom plugin with WASM
- Search in datums?

Custom indexer

- Cardano node is required
- Can be integrated to our application
- Can be optimized for most typical use-case (memory, performance, etc.)
- No so difficult to write it
- It is possible to use the interface for passing transaction to mempool, etc.

Notes

- Rollback are always possible
- Finding the ideal height to avoid rollback is unsafe!!!
- It is not good idea to "mutate" blockchain data because of rollback
- Pass rollbacks to database "foreign keys and cascade"
- Bulk is faster then single insert (lot of queries, index regeneration, lot of subqueries)
- Normalized form of database are not always faster because of id/uuid lookup
- Store meta information (slot, hash) for all blocks is good idea because of the range query of node when reindexing is needed.

Links

- https://cardanoscan.io/
- https://cexplorer.io/

- https://developers.cardano.org/docs/get-started/cardano-node/running-cardano/
- https://blockfrost.io/
- https://github.com/IntersectMBO/cardano-db-sync
- https://github.com/dcSpark/carp
- https://github.com/CardanoSolutions/kupo

- https://github.com/txpipe/oura
- https://github.com/txpipe/pallas
- https://github.com/miksax/CryptoAcademy2024