EOS区块链使用mongo_db_plugin将区块链的数据写入到mongo数据库中。DApp可以通过查询mongodb的集合获取自己需要的信息

使用场景

由于需要对链上数据进行大量自定义的查询，一般是通过部署自有的节点，然后使用在eos节点上使用mongo插件将链上数据同步到mongo数据库，然后查询mongo数据库，这样更直接和方便。虽然有其他得替代方案比如(Hyperion-History-API)，(history-tools)，测试下来，mongo更简单直接些。

mongo 插件配置

```
-q [ --mongodb-queue-size ] arg (=1024)
                                      The target queue size between nodeos
                                      and MongoDB plugin thread.
--mongodb-abi-cache-size arg (=2048)  The maximum size of the abi cache for
                                      serializing data.
--mongodb-wipe                        Required with --replay-blockchain,
                                      --hard-replay-blockchain, or
                                      --delete-all-blocks to wipe mongo
                                      db.This option required to prevent
                                      accidental wipe of mongo db.
--mongodb-block-start arg (=0)        If specified then only abi data pushed
                                      to mongodb until specified block is
                                      reached.
-m [ --mongodb-uri ] arg              MongoDB URI connection string, see:
                                      https://docs.mongodb.com/master/referen
                                      ce/connection-string/. If not specified
                                      then plugin is disabled. Default
                                      database 'EOS' is used if not specified
                                      in URI. Example: mongodb://127.0.0.1:27
                                      017/EOS
--mongodb-update-via-block-num arg (=0)
                                      Update blocks/block_state with latest
                                      via block number so that duplicates are
                                      overwritten.
--mongodb-store-blocks arg (=1)       Enables storing blocks in mongodb.
--mongodb-store-block-states arg (=1) Enables storing block state in mongodb.
--mongodb-store-transactions arg (=1) Enables storing transactions in
                                      mongodb.
--mongodb-store-transaction-traces arg (=1)
                                      Enables storing transaction traces in
                                      mongodb.
--mongodb-store-action-traces arg (=1)
                                      Enables storing action traces in
                                      mongodb.
--mongodb-filter-on arg               Track actions which match
                                      receiver:action:actor. Receiver,
                                      Action, & Actor may be blank to include
                                      all. i.e. eosio:: or :transfer:  Use *
                                      or leave unspecified to include all.
--mongodb-filter-out arg              Do not track actions which match
```

```
                                        receiver:action:actor. Receiver,
                                        Action, & Actor may be blank to exclude
                                        all.
```

开启插件

通过配置下面选项，能开启nodeos节点的mongo插件，将链上的数据写入到mongo数据库中。

- plugin=eosio::mongo_db_plugin
- mongodb-uri=mongodb://ip:port/EOS

控制写入哪些集合

在config.ini中必须开启以下配置，才能将数据写入mongodb的集合中：

- mongodb-store-block-states = true #如果并不是特殊需要，建议不要开启，会占用大量存储
- mongodb-store-blocks = true #如果并不是特殊需要，建议不要开启，会占用大量存储
- mongodb-store-transactions = true
- mongodb-store-transaction-traces = true #如果并不是特殊需要，建议不要开启，会占用大量存储
- mongodb-store-action-traces = true #如果并不是特殊需要，建议不要开启，会占用大量存储

选择过滤数据

通常，我们希望控制与自己相关的合约的数据可以写入mongo数据库中，滤除掉自己不相关的数据。可以使用下面的选项。

- --mongodb-filter-on 允许哪些合约，哪些action，哪些actor(交易发起者)允许写入mongo数据库
- --mongodb-filter-out 不允许哪些合约，哪些action，哪些actor(交易发起者)允许写入mongo数据库

数据集合(collections)

这些数据包括下面4类：

- block 块
- transaction 交易
- action 活动
- account 账户

```
> show collections
account_controls
accounts
action_traces
block_states
blocks
pub_keys
transaction_traces
transactions
```

其中transaction_traces，transactions, action_traces存储交易数据。DApp可以从这几个表中查询交易信息。这几个表的数据其实是冗余的。

action_traces表默认没有"trx_id"得索引，如果想根据"trx_id"查询得话，可以自行添加索引，不然会很慢。

```
db.action_traces.getIndexes()
db.action_traces.createIndex({"trx_id":1, "background":true})
```

acounts存储账户信息，需要确保accounts集合中包含有你的智能合约对应的文档，这个文档信息对应着你的智能合约部署时的abi信息，这个信息确保mongodb中action、transaction的中的action的参数data能有效解释，否则你find()查看记录时将看到的是看不懂的原始数据。

判断交易是否存在，是否不可逆

交易发出后，会返回一个交易id，在transactions表中查询这个交易的id文档是否存在，如果存在，检查该交易的的字段"irreversible"，如果为true，表示该交易已经确定不可逆的记录在区块链上。

例如：

发交易

```
$cl  push action hello hi '["alice"]' -p alice  -f
executed transaction:
ffbe2e13510ba97c7e0c5e5d774b125ae37834719e9dc4ed3fb1af074d60e73d  128 bytes  407
us

#    eosio.null <= eosio.null::nonce            "7f59cf87b5ae0500"

#         hello <= hello::hi                    {"user":"alice"}
```

刚发完交易后，

```
db.transactions.find({"trx_id":"ffbe2e13510ba97c7e0c5e5d774b125ae37834719e9dc4ed
3fb1af074d60e73d"}).pretty()
{
"_id" : ObjectId("5f55f80697631990b143fcd8"),
"trx_id" : "ffbe2e13510ba97c7e0c5e5d774b125ae37834719e9dc4ed3fb1af074d60e73d",
"accepted" : true,
"actions" : [
    {
        "account" : "hello",
        "name" : "hi",
        "authorization" : [
            {
                "actor" : "alice",
                "permission" : "active"
            }
        ],
        "data" : {
            "user" : "alice"
        },
        "hex_data" : "0000000000855c34"
    }
],
"context_free_actions" : [
```

```
    {
        "account" : "eosio.null",
        "name" : "nonce",
        "authorization" : [ ],
        "data" : "7f59cf87b5ae0500"
    }
],
"context_free_data" : [ ],
"createdAt" : ISODate("2020-09-07T09:06:14.505Z"),
"delay_sec" : 0,
"expiration" : "2020-09-07T09:06:44",
"implicit" : false,
"max_cpu_usage_ms" : 0,
"max_net_usage_words" : 0,
"ref_block_num" : 11771,
"ref_block_prefix" : NumberLong("4139172678"),
"scheduled" : false,
"signatures" : [

"SIG_K1_Kc1ia7NcNKCLadYnfvemHFghRnYKTmfYiDEhTvNQ4WMHTh17N1HZ8Rhv2iJny59eJvwKjUCr
nEFgYXzD6qN9SVn52C5sy1"
],
"signing_keys" : {
    "0" : "EOS7HxPMkfyL69PqLXduP9YfuvVad8e3Nry6ryDGaJ2u8BKB2zUUm"
},
"transaction_extensions" : [ ]
}
```

约60秒后

```
db.transactions.find({"trx_id":"ffbe2e1351 0ba97c7e0c5e5d774b125ae37834719e9dc4ed
3fb1af074d60e73d"}).pretty()
{
"_id" : ObjectId("5f55f80697631990b143fcd8"),
"trx_id" : "ffbe2e1351 0ba97c7e0c5e5d774b125ae37834719e9dc4ed3fb1af074d60e73d",
"accepted" : true,
"actions" : [
    {
        "account" : "hello",
        "name" : "hi",
        "authorization" : [
            {
                "actor" : "alice",
                "permission" : "active"
            }
        ],
        "data" : {
            "user" : "alice"
        },
        "hex_data" : "0000000000855c34"
    }
],
"context_free_actions" : [
    {
        "account" : "eosio.null",
        "name" : "nonce",
```

```
        "authorization" : [ ],
        "data" : "7f59cf87b5ae0500"
    }
],
"context_free_data" : [ ],
"createdAt" : ISODate("2020-09-07T09:06:14.505Z"),
"delay_sec" : 0,
"expiration" : "2020-09-07T09:06:44",
"implicit" : false,
"max_cpu_usage_ms" : 0,
"max_net_usage_words" : 0,
"ref_block_num" : 11771,
"ref_block_prefix" : NumberLong("4139172678"),
"scheduled" : false,
"signatures" : [

"SIG_K1_Kc1ia7NcNKCLadYnfvemHFghRnYKTmfYiDEhTvNQ4WMHTh17N1HZ8Rhv2iJny59eJvwKjUCr
nEFgYXzD6qN9SVn52C5sy1"
],
"signing_keys" : {
    "0" : "EOS7HxPMkfyL69PqLXduP9YfuvVad8e3Nry6ryDGaJ2u8BKB2zUUm"
},
"transaction_extensions" : [ ],
"block_id" : "02972dff562ea2b47df66654c5a606e92e5ce3d102970d0a8f338567b9ca1566",
"block_num" : 43462143,
"irreversible" : true,
"updatedAt" : ISODate("2020-09-07T09:07:13.018Z")
}
```

比较前后可以发现，一段时间后增加了下面的字段。

```
"block_id" : "02972dff562ea2b47df66654c5a606e92e5ce3d102970d0a8f338567b9ca1566",
"block_num" : 43462143,                #交易记录在这个块中
"irreversible" : true,                 #交易不可逆
"updatedAt" : ISODate("2020-09-07T09:07:13.018Z")  #写入mongodb的时间，不是交易发生的
时间。
```

业务查询与统计

一个业务的交易可以包含多个action，可以通过action_traces集合来查询统计业务量，指定查询条件如account，actor等，但查询会很慢，因为对业务数据没有索引，需要自己根据不同的业务在不同字段上建索引，才能快速查询统计。过多的索引会降低mongo数据库的性能，占用更多的存储空间。