Cardano SL Wallet

Web API Benchmarking

`API v0`

Middleware Team

Contents

Benchmark Environment

Table 1.1: User Story

Issue	Owner	Sprint
CBR-23	Denis Shevchenko	Cardano #57: Novaya Nadezhda

Table 1.2: Computer

OS	CPU	RAM
Debian Linux 9.3 (64bit)	Core i7-7500U @ 2.70GHz	16 GB DDR4 2133 MHz

Table 1.3: Bench Tool

Package	Version
gauge	0.2.1

Table 1.4: Code

Feature Branch	Base Branch
`feature/cbr23-wallet-bench`	`master`

Table 1.5: Build

Script	RTS Options	Use Nix
`build/cardano-sl.sh`	`-N2`	No

Table 1.6: Launch

Script	Number of nodes	Connect to Mainnet
`launch/demo-with-wallet-api.sh`	4	No

Benchmark Launch

Launch

Example of complete command:

```
$ stack bench cardano-sl-wallet --benchmark-arguments  \
    "--tls-pub-cert=$PWD/scripts/tls-files/ca.crt      \
     --tls-priv-key=$PWD/scripts/tls-files/server.key  \
     --wal-conf=$PWD/wallet/bench/config/Wallets.yaml  \
     --ep-conf=$PWD/wallet/bench/config/Endpoints.csv"
```

Run:

```
$ stack bench cardano-sl-wallet --benchmark-arguments "--help"
```

to see description of supported arguments.

Configuration

There are two different configuration files:

1. `Wallets.yaml`
   Contains wallets, accounts and addresses we are using during benchmarking.
2. `Endpoints.csv`
   Contains a list of Wallet Web API endpoints we want to benchmark. By default all listed benchmarks will be launched sequentially, one by one.

Preparing

To make benchmarking more realistic, wallet database was generated, using `dbgen` tool. Please follow these instructions[1] to reproduce it on your local computer.

As a result we have 1 wallet with 80k addresses in it. Most of these addresses contains constant amount of money.

[1] https://iohk.myjetbrains.com/youtrack/issue/CSL-2249#comment=93-17408

GetHistory

Table 3.1: Empty tx history

Wallets	Accounts	Addresses	**Time, ms**	**Mean, ms**
1	1	1	**101.1**	**101.8**
1	2	80k	**637**	**641.2**

Table 3.2: Non-empty tx history, 80k addresses

Wallets	Accounts	Transactions	**Time, ms**	**Mean, ms**
1	2	435	**691.3**	**882.5**
1	2	1.7k	**1106**	**1091**

GetWallet

Table 4.1: Empty tx history

Wallets	Accounts	Addresses	**Time, ms**	**Mean, ms**
1	1	1	**126.9**	**126.1**
1	2	80k	**3641**	**3407**

Table 4.2: Non-empty tx history, 80k addresses

Wallets	Accounts	Transactions	**Time, ms**	**Mean, ms**
1	2	435	**3625**	**3435**
1	2	1.7k	**3588**	**3513**

GetWallets

Table 5.1: Empty tx history

Wallets	Accounts	Addresses	**Time, ms**	**Mean, ms**
1	1	1	**124.9**	**125.2**
1	2	80k	**3115**	**3256**

Table 5.2: Non-empty tx history, 80k addresses

Wallets	Accounts	Transactions	**Time, ms**	**Mean, ms**
1	2	435	**3485**	**3488**
1	2	1.7k	**3924**	**3623**

IsValidAddress

Table 6.1: Empty/non-empty tx history

Wallets	Accounts	Addresses	**Time, ms**	**Mean, ms**
1	1	1	**91.72**	**90.95**
1	2	80k	**92.32**	**89.60**

NewAddress

Table 7.1: Empty/non-empty tx history

Wallets	Accounts	Addresses	**Time, ms**	**Mean, ms**
1	1	1	**130.5**	**129.8**
1	2	80k	**138.8**	**132.4**

GetSyncProgress

Table 8.1: Empty/non-empty tx history

Wallets	Accounts	Addresses	**Time, ms**	**Mean, ms**
1	1	1	**92.99**	**91.01**
1	2	80k	**96.81**	**90.41**

NewPayment

Table 9.1: Empty tx history before starting

Wallets	Accounts	Addresses	**Time, ms**	**Mean, ms**
1	2	80k	**7680**	**8115**

Table 9.2: Non-empty tx history, 80k addresses

Wallets	Accounts	Transactions	**Time, ms**	**Mean, ms**
1	2	435	**8142**	**8171**
1	2	1.2k	**8252**	**8589**
1	2	1.7k	**9242**	**8382**