Análisis Real Time de transaciones de Bitcoin

Autor: Hernán Cortés Pastor

Fecha:13/12/2020

Contenido

1 Introducción

1.1 Transacciones de Bitcoin.

Las transacciones de bitcoin son entendidas como los envíos de bitcoins de una persona a otra usando la red Bitcoin. En este punto, todas estas transacciones no son más que registros guardados en la cadena de bloques (blockchain). El mismo principio también se aplica al resto de criptomonedas como Ethereum, Dash o Bitcoin Cash.

Para realizar dichas transacciones necesitamos de un cliente para la criptomoneda, mejor conocido como monedero o wallet. Estos no son más que una pieza de software que nos permite administrar nuestros fondos y gestionar nuestras direcciones. Gracias a ellos podemos enviar y recibir criptomonedas. La dirección publica funciona como una dirección de correo o un número de cuenta bancaria a donde vamos hacer o recibir la transacción.

Los elementos que conforman una transacción son los siguientes:

- **Entradas (inputs).** Las entradas son las referencias a una salida de una transacción pasada que no ha sido empleada en ninguna otra transacción. Estas nos permiten confirmar la procedencia de los activos que se utilizarán en una transacción. Y son las que contienen la dirección (address) donde originalmente se recibieron los bitcoins.
- **Salidas (outputs).** Estas contienen la dirección (address) a la cual se realiza la transferencia y la cantidad enviada. Una transacción puede contener más de una salida.
- **Identificador (TXid).** Cada transacción realizada tendrá su propio hash. Este hash se genera a partir de las entradas y las salidas. Este valor es el que permite identificar una transacción de forma única e irrepetible dentro de una blockchain.

Técnicamente, las transacciones son archivos JSON como el que se muestra a continuación:


```
{
    "op": "utx",
    "x": {
        "lock_time": 0,
        "ver": 1,
        "size": 192,
        "inputs": [
            {
                "sequence": 4294967295,
                "prev_out": {
                    "spent": true,
                    "tx_index": 99005468,
                    "type": 0,
                    "addr": "1BwGf3z7n2fHk6NoVJNkV32qwyAYsMhkWf",
                    "value": 65574000,
                    "n": 0,
                    "script": "76a91477f4c9ee75e449a74c21a4decfb50519cbc245b388ac"
                },
                "script":
"483045022100e4ff962c292705f051c2c2fc519fa7e0c8fd4c5fe539184a30fe35a2f5fccf7ad62054cad29360d871f8187d"
            }
        ],
        "time": 1440086763,
        "tx_index": 99006637,
        "vin_sz": 1,
        "hash": "0857b9de1884eec314ecf67c040a2657b8e083e1f95e31d0b5ba3d328841fc7f",
        "vout_sz": 1,
        "relayed_by": "127.0.0.1",
        "out": [
            {
                "spent": false,
                "tx_index": 99006637,
                "type": 0,
                "addr": "1A828tTnkVFJfSvLCqF42ohZ51ksS3jJgX",
                "value": 65564000,
                "n": 0,
                "script": "76a914640cfdf7b79d94d1c980133e3587bd6053f091f388ac"
            }
        ]
    }
}
```

1.2 Objetivo

Una de las características fundamentales del Blockchain es que permite acceder a todas las transacciones realizadas alrededor del mundo en el instante. Desde la siguiente web se proporciona un WebSocket API Real-Time (público) que es posible consumir por cualquier usuario.

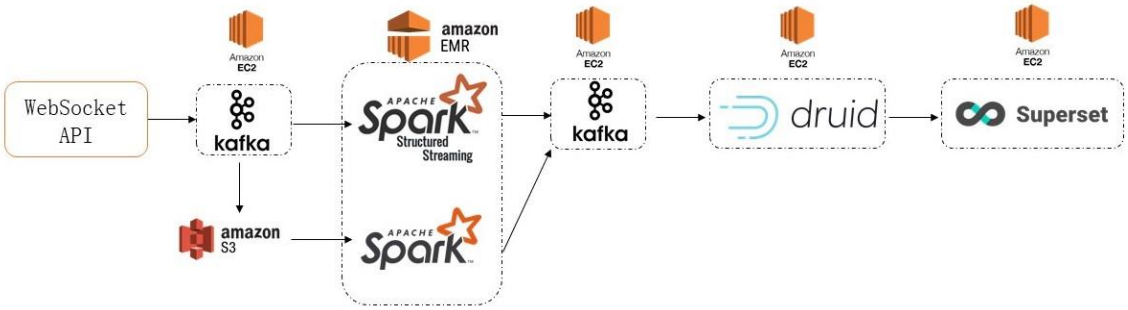
https://www.blockchain.com/api/api_websocket

Surge por tanto la oportunidad de realizar el procesamiento de estas transacciones en tiempo real con el objetivo de realizar un análisis del mercado en el momento actual.

2 Desarrollo y tecnologías

2.1 Arquitectura de solución.

En el siguiente esquema, se muestra la arquitectura y las tecnologías empleadas.



Como se puede ver se trata de una arquitectura Lambda.

- Job Spark Structured Streaming realiza el procesamiento en tiempo real consumiendo los datos de Kafka, procesándolos y volviendo a dejarlos de nuevo en Kafka para que sean consumidos por la base de datos analítica Druid y finalmente por la herramienta de visualización Superset.
- Job Spark obtiene los datos históricos que se han persistido en el servicio de almacenamiento S3 de AWS. Una vez procesados en Spark se dejan en Kafka para que sean consumidos por la base de datos analítica Druid y finalmente por la herramienta de visualización Superset. Reutilizando así parte de la estructura de Streaming.

2.2 Detalle de tecnologías empleadas.

2.2.1 Kafka y Kafka Connect

Apache Kafka es una plataforma distribuida de transmisión de datos que permite publicar, almacenar y procesar flujos de registros, y suscribirse a ellos, en tiempo real.

Kafka Connect es una herramienta proporcionada por Kafka que permite el streaming de datos entre la plataforma y otros sistemas de datos.

En este proyecto, primeramente, usamos Kafka para la ingesta en tiempo real de todas transacciones en bruto que son consumidas por procesos de Spark Structured Streaming.

Además de esto, se usará Kafka Connect para persistir todos los datos en un bucket de S3 y que puedan ser usados posteriormente en procesos batch.

Por otro lado, los resultados de los procesos de Spark se vuelcan a otros topics de Kafka para su posterior uso en Apache Druid.

2.2.2 S3

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento.

En este proyecto se usa S3 con el objetivo de persistir todas las transacciones recibidas y poder acceder a ellas posteriormente mediante procesos batch.

2.2.3 Spark

Apache Spark es un framework de programación diseñada para el procesamiento de datos masivos de forma distribuida y rápida.

En este proyecto se usa Spark Structured Streaming con el objetivo de procesar las transacciones validando, enriqueciendo, filtrando, agrupando etc. los datos antes de ser enviados a Druid.

Por otro lado, se usa Spark en procesos batch, con el objetivo de realizar el procesamiento de datos históricos.

2.2.4 Druid

Apache Druid se describe como una base de datos analítica en tiempo real para grandes volúmenes de datos.

En este proyecto, se usará esta herramienta para obtener los datos de diferentes topics de Kafka, realizar el procesamiento y agrupamiento necesario y dejarlos listos para ser consumidos por una herramienta de visualización.

2.2.5 Superset

Apache Superset es una aplicación de software de código abierto para la exploración y visualización de datos capaz de manejar grandes cantidades de datos.

En este proyecto, usaremos Superset conectado a Druid para obtener los datos y elaborar un dashboard de visualización de datos.

2.2.6 AWS EC2 y EMR.

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable.

Amazon EMR es una plataforma de clúster administrada que simplifica la ejecución de trabajo de Big Data con Spark.

El proyecto se desplegará en la nube AWS. Para ello se usarán diferentes instancias EC2. Además, los diferentes Jobs de Spark se lanzarán desde el servicio administrado EMR de AWS.

3 Instalación y despliegue

Se crea un repositorio en GitHub con el desarrollo del proyecto. En el fichero 'Readme.md' del repositorio se detalla la instalación y despliegue de este proyecto.

https://github.com/hernancortespastor/Transactions-BTC

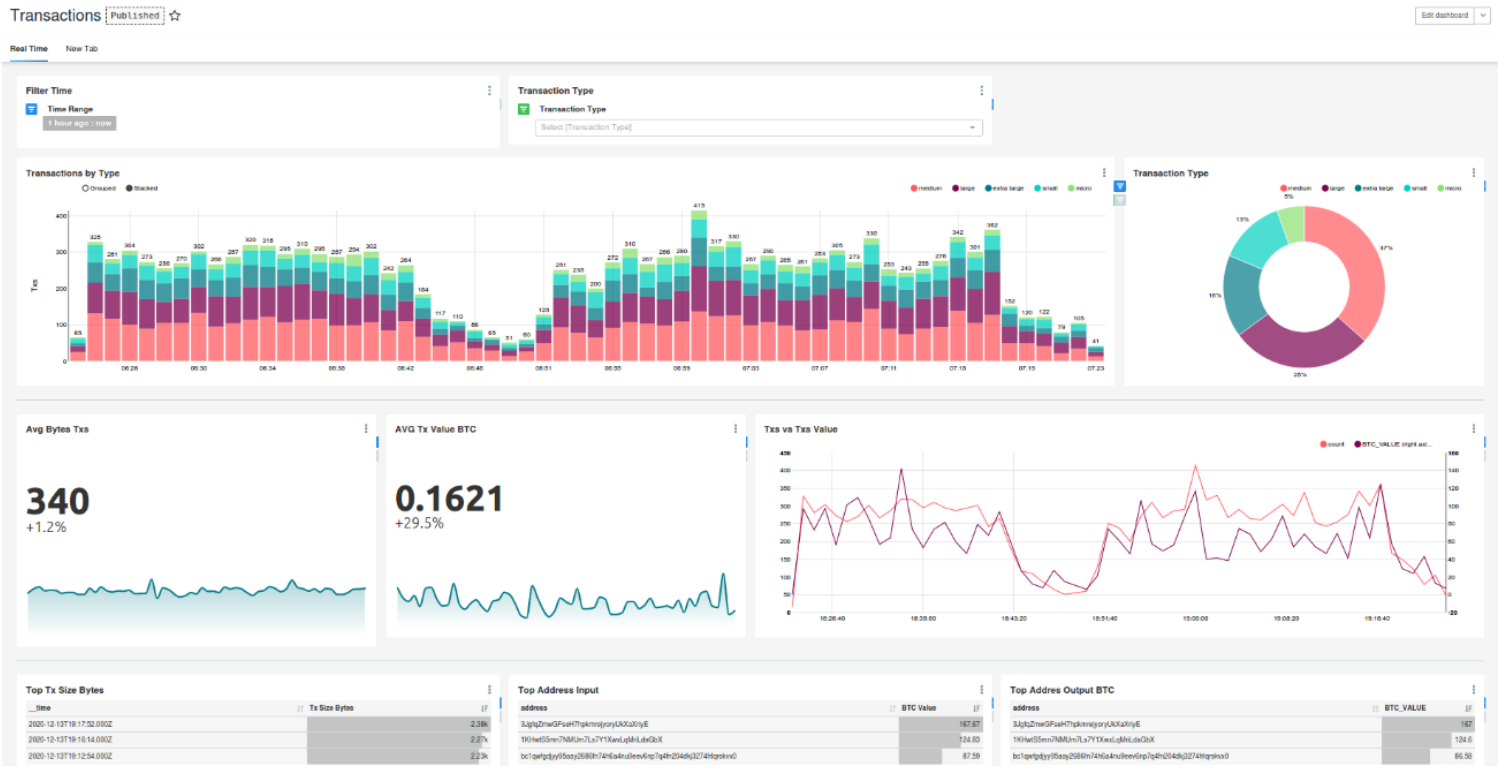
4 Resultados.

4.1 KPIs

Se definen los siguientes KPIs, a partir de estos se elaboran las visualizaciones para poder analizar los datos.

1- Número de transacciones por valor de BTC transferido.
2- Promedio de tamaño en bytes de transacciones.
3- Promedio de valor de BTC total de transacciones.
4- Mayor transacción tamaño de transacción.
5- Direcciones top de envío.
6- Direcciones top de Recepción.

A partir de estos KPIs se elaboran diferentes visualizaciones en SUPERSET y se agrupan en un dashboard.



5 Mejoras propuestas

1- Usar el servicio gestionado MSK de AWS en lugar de una instancia de EC2 genérica.
2- Automatizar despliegue del proyecto en AWS.
3- Enviar los resultados del Job de Spark a otra base de datos en lugar de Kafka y Druid.
4- Añadir nuevos KPIs y visualizaciones.
5- Monitorizar y optimizar rendimiento de Jobs de Spark.