UNIVERSITÀ DEGLI STUDI ROMA TRE

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica

Tesi Di Laurea

Analisi, progettazione e prove sperimentali di un FulgurHub in TypeScript

Laureando

Federico Ginosa
Matricola 457026

Relatore
Alberto Paoluzzi

Correlatore
Federico Spini

Anno Accademico 2017-2018

Ad Ada Lovelace

Indice

Elenco delle figure

Capitolo 1

Introduzione

2008 pubblicazione paper di Sathosi Nakamoto

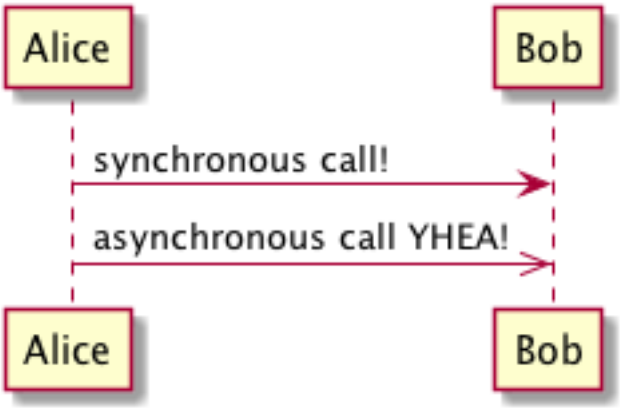

Figura 1.1: Uno schema UML realizzato con plantuml

2009 pubblicazione protocollo BitCoin

Prova [1] test

Problemi di scalabilità

Throughput teorico massimo di BitCoin

Aumento costi delle fee

Soluzioni al problema della scalabilità

Algoritmo di consenso

Sharding

OffChain

Lavoro di questa tesi

Analisi dello stato dell'arte relativa a soluzioni di scalabilità OffChain

Design e implementazione di un IPC

Analisi, progettazione e sviluppo di un FulgurHub

Prove sperimentali di Fulgur Hub

Descrizione capitoli

Capitolo 2

Background

Distributed Ledger Technologies

Il problema che risolvono le DLT

1. Transazioni trustless in un sistema distribuito
2. DLT permissionless vs DLT permissioned

Caso d'uso: scambio di asset

1. La transazione, rappresentazione dello scambio di valore
2. Il ledger, registro pubblico degli scambi di valore

Blockchain e smart contract

1. Meccanismo di consenso e Proof of Work
   (a) Descrizione generale della PoW
   (b) Problema del double spending
   (c) Controllo della generazione di asset
2. Aggiornare lo stato della blockchain con operazioni complesse basate su smart contract

Scalabilità OffChain

State channel

Payment channel

1. Architettura generale
   (a) 1 smart contract
   (b) 2 server
2. Inizializzazione
   (a) Deploy
   (b) Apertura
   (c) Join
3. Schema propose/accept

   (a) Gli endpoint

      i. Propose

      ii. Accept

   (b) Struttura di una propose

      i. Numero di sequenza

      ii. Balance A

      iii. Balance B

      iv. Firma della propose

4. Chiusura in due fasi

   (a) Richiesta di chiusura

      i. L'operazione "'close"'

   (b) Finalizzazione della chiusura

      i. Il tempo di grazia

      ii. L'operazione "'withdraw"'

Inextinguishable payment channel

1. Estensione delle struttura dati di una propose

   (a) Hash di un token

   (b) Tipologia di operazione

   (c) Tipologia di catena

2. Struttura di un token

    (a) Numero di sequenza

    (b) Valore

    (c) Scadenza

    (d) Firma del token

3. Schema attach/detach

    (a) Detach di un token OffChain

    (b) Attach di un token OnChain

4. Ricarica a caldo

Obiettivi di Fulgur Hub

Transazioni immediate

Transazioni tra più di due entità

Transazioni tra diversi hub

Autogestito

Non censurabile

FulgurHub e stato dell'arte

Lightning Network

1. Topologia di rete a confronto e censura
2. Superamento del problema di ricerca del percorso ottimo

NOCUST

1. Conferma di una transazione non immediata

Capitolo 3

Analisi

Obiettivi

Dimostrazione di fattibilità

1. Transazioni OffChain-OffChain
2. Transazioni OnChain-OnChain
3. Transazioni OffChain-OnChain
4. Transazioni OnChain-OffChain
5. Prelievi a caldo
6. Ricariche a caldo

Dimostrare la scalabilità architetturale

Descrizione generale dell'architettura

Lo smart contract

Il client

L'hub

Casi d'uso

Apertura di un canale

1. Pre condizioni
2. Descrizione delle interazioni

Pagamento OnChain-OnChain

1. Pre condizioni
2. Descrizione delle interazioni
3. Gestione delle eccezioni
    (a) Credito insufficiente del client OnChain

Pagamento OffChain-OffChain

1. Pre condizioni
2. Descrizione delle interazioni
3. Gestione delle eccezioni
   (a) B non invia la ricevuta di pagamento ad A
   (b) Generazione di una miriade di token
   (c) L'hub non permette di attaccare un token
   (d) L'hub non permette di staccare un token
   (e) A si rifiuta di regolare un trasferimento nei confronti dell'hub
   (f) Tentativo di pagamento con un token scaduto
   (g) Mancanza di cooperazione nel ricevere un pagamento

Pagamento OffChain-OnChain

1. Pre condizioni
2. Descrizione delle interazioni

Pagamento OnChain-OffChain

1. Pre condizioni
2. Descrizione delle interazioni

Prelievo a caldo

1. Pre condizioni
2. Descrizione delle interazioni

Ricarica a caldo

1. Pre condizioni
2. Descrizione delle interazioni

Chiusura di un canale

1. Pre condizioni
2. Descrizione delle interazioni

Riscossione dei pending token

1. Pre condizioni
2. Descrizione delle interazioni
3. Gestione delle eccezioni
    (a) Tentativo di ritirare un pending token già usato

Capitolo 4

Progettazione e sviluppo

Le motivazioni tecnologiche

La blockchain: Ethereum

1. Supporto degli smart contract
2. Ambiente di sviluppo maturo
   (a) Solidity
   (b) Ganache
   (c) Web3

Il linguaggio di programmazione: TypeScript

1. Supporto di web3
2. Tipizzazione forte

Il database lato server: Redis

1. Throughput considerevole in scrittura
2. Customizzazione delle qualità nei limiti del teorema CAP
   (a) Consistenza
   (b) Disponibilità
   (c) Sharding

Il database lato client: LevelDB

Lo smart contract

Implementazione in Solidity

Interfaccia in TypeScript

Il client

RPC privata

1. Join di un hub
2. Trasferimento OnChain-OnChain
3. Detach di un token OffChain-OffChain
4. Detach di un token OnChain-OffChain

5. Invio della PoD
6. Redimere un pending token
7. Attach di un token OnChain
8. Regolazione di un pagamento OffChain
9. Invio della ricevuta di pagamento

Endpoint pubblici

1. Ricezione di una PoD
2. Ricezione di una ricevuta di pagamento

Gestione degli eventi asincroni

1. Il monitor
2. Gli eventi
   (a) Detach di un token OnChain
   (b) Ricezione di una PoD

Hub

Endpoint pubblici

1. Ricezione di una propose

2. Ricezione di una ricevuta di pagamento

Gestione degli eventi asincroni

1. Il monitor
2. Gli eventi
    (a) Join di un utente
    (b) Chiusura di un canale
    (c) Ritiro di un pending token

Capitolo 5

Prove sperimentali

Gli obiettivi

Verifica delle performance delle transazioni OffChain

Verifica della scalabilità delle transazioni OffChain

L'approccio adottato

Benchmark server

1. Deploy dell'ambiente di collaudo basato su Docker Swarm
2. Esecuzione del benchmark
   (a) Transazioni seriali
   (b) Transazioni concorrenti

(c) Simulazione della latenza di rete

Il throughput lato client

Risultati

1. Al variare della RAM
   (a) Tabella
   (b) Grafico
2. Al variare della CPU
   (a) Tabella
   (b) Grafico

Il throughput lato hub

Risultati

1. Al variare della RAM
   (a) Tabella
   (b) Grafico
2. Al variare della CPU
   (a) Tabella

Considerazioni sulle performance

Considerazioni sulla scalabilità

Replicare l'hub

Replicare redis

Capitolo 6

Conclusioni e sviluppi futuri

Autogestione finanziaria dell'hub

Denominazione degli endpoint sulla base della valuta

[1] Joseph Poon and Thaddeus Dryja. 2016. The bitcoin lightning network: Scalable off-chain instant payments. *draft version 0.5* 9, (2016), 14.