Jollar – A Jolie Based Criptocurrency

Stefano Pio Zingaro

April 21, 2018

Indice

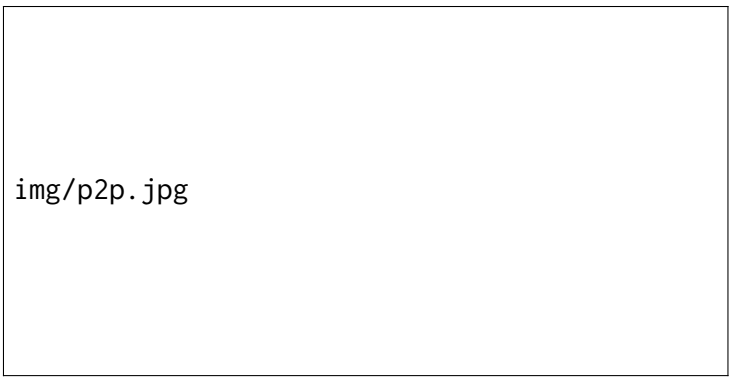
Abstract

Il progetto si propone di creare un sistema di scambio elettronico decentralizzato, in cui gli utenti effettuano transazioni certificate dalla rete stessa, senza il bisogno di un organo centrale garante (come ad esempio una banca). Il progetto è sviluppato seguendo i principi della programmazione orientata ai (micro)servizi, in cui la comunicazione tra i processi è implementata nel linguaggio visto a lezione: Jolie.

1 Descrizione del progetto

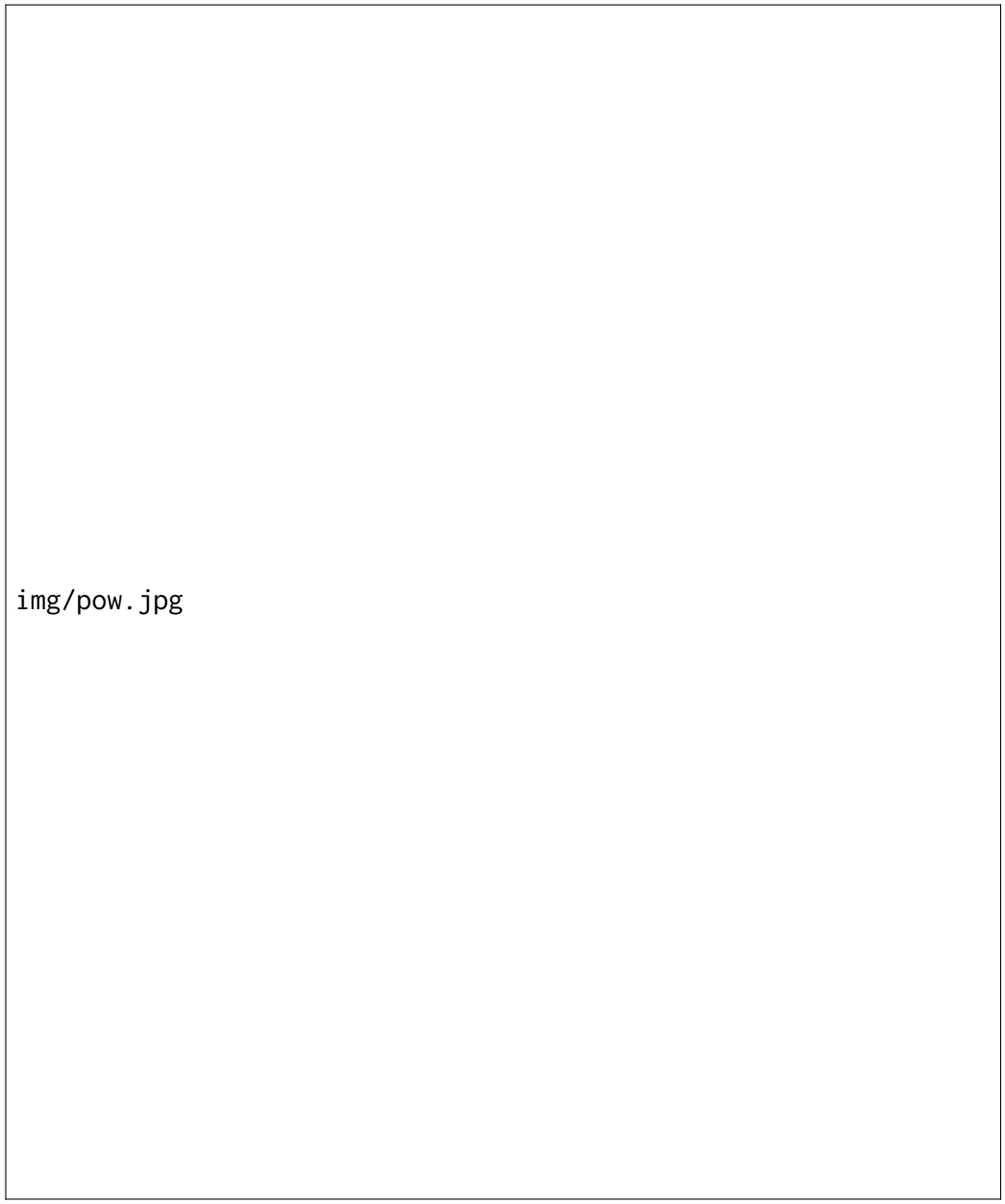
In generale, il sistema consiste in una rete *Peer-To-Peer* (**P2P**) che utilizza tecniche di *Proof-of-Work* (**POW**) per registrare l'elenco delle transazioni

in un archivio pubblico, detto **blockchain**. Una rete P2P, come ad esempio quella di Emule[1], è una rete i cui nodi non sono **client** o **server** fissi, ma prendono forma di entità equivalenti o "paritarie".



La PoW, è un algoritmo che viene utilizzato per raggiungere un accordo decentralizzato tra diversi nodi nel processo di aggiunta di un blocco specifico alla blockchain. Tale algoritmo produce valori che vengono utilizzati per verificare che sia stata eseguita una notevole quantità di lavoro. Nell'immagine riportata qui sotto viene visualizzato un esempio di utilizzo di tecniche POW nel mondo delle criptovalute (in particolare quella di Ethereum ma generalizzabile a qualunque altra che usa POW).

[1] www.emule-project.net/home/perl/general.cgi?l=18



La **blockchain** è una struttura dati ordinata, una lista concatenata di blocchi contenenti transazioni. L'intera struttura della blockchain può essere conservata in un file, oppure in un database. Un **block**, o blocco, è un contenitore, una struttura dati a sua volta, che aggrega transazioni che devono essere incluse in un *ledger*[2] pubblico e condiviso, la blockchain. Il blocco è composto da un *header*, che contiene i *metadata*, e dalla lista di transazioni

[2]termine che indica il libro mastro dei contabili, viene tradotto come registro.

che ne compongono la maggior parte della grandezza.

1.1 Le Transazioni

1.2 Il Server Timestamp

1.3 La Proof-of-Work

Per implementare un server timestamp basato su rete P2P, abbiamo bisogno di usare un sistema di proof-of-work[3]. Questo metodo consiste nell'obbligare i nodi che vogliono scrivere un blocco a cercare un valore che sia difficile da trovare e di cui sia facile controllarne la correttezza da parte degli altri nodi che vogliono validare la scrittura.

- **Indice**: Un numero primo p deve essere generato, una buona pratica sarebbe quella di generare un numero primo abbastanza grande, quindi si consiglia l'utilizzo di tipi `Long`. I `Long` in Java, quindi anche in Jolie, hanno un limite superiore pari a $9,223,372,036,854,775,807$. Oltre questo numero il sistema semplicemente termina di produrre prove.
- **Definizione**: $f_p(x) \equiv \sqrt{x} \bmod p$, cioè equivale ad estrarre tutte le radici quadrate modulo il numero primo p. Il dominio di f_p è $\mathbb{Z}_p$.
- **Verifica**: Dati x, y controllare che $y^2 \equiv x \bmod p$.

Il meccanismo di controllo prevede solo una moltiplicazione, mentre non esiste un metodo veloce per l'estrazione delle radici modulo un numero primo, che non richieda meno di $\log p$ moltiplicazioni. Ne consegue che maggiore sia la lunghezza di p, maggiore sarà lo sforzo computazionale necessario, e quindi maggiore la differenza di tempo per valutare f_p e per verificarne la correttezza.

1.4 La Rete Peer-To-Peer

1.5 Il Network Visualizer

Il Network Visualiser è un tool amministrativo da terminale per il monitoraggio del sistema.

[3] `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/pvp.ps`

2 Consegna del Progetto

Il progetto deve essere sviluppato utilizzando il linguaggio Jolie. Non ci sono requisiti riguardo ai protocolli (*protocol*) e i media (*location*) utilizzati per realizzare la comunicazione tra i componenti dello Shop. L'obiettivo riguarda la gestione concorrente di risorse condivise. L'importante è dimostrare che il comportamento implementato gestisca i tipici problemi di concorrenza su risorse condivise (e.g., il tipico problema `Readers-Writers`).

N.B. Uno dei punti di valutazione riguarda l'uso del parallelismo nel progetto. Ad esempio usare execution sequential per prevenire scritture (e letture) parallele è una soluzione al problema considerato, ma rappresenta anche una notevole perdita in termini di concorrenza, incidendo negativamente sulla valutazione del progetto.

2.1 Report

Il report, scritto in word o LaTeX, dovrà avere le seguenti caratteristiche formali:

1. Dovrà avere lunghezza di quattro o cinque pagine (da 4 a 5 pagine equivalgono da 8 a 10 facciate).
2. Dovrà essere scritta in singola colonna con font di grandezza **12pt**.
3. Deve essere consegnato in formato PDF.

In contenuto del report avrà discussioni in particolare sui seguenti argomenti:

- la struttura del progetto (la divisione delle funzionalità tra i servizi e la loro gerarchia);
- le scelte più importanti fatte sul progetto e come sono state sviluppate, con esempi di codice;
- i problemi principali riscontrati, le alternative considerate e le soluzioni scelte;
- le istruzioni per eseguire il progetto e quelle per eseguire una demo di esecuzione.

Questo punto è particolarmente importante, scrivete le specifiche come se tutto il programma dovesse essere eseguito da una persona che non sa nulla del progetto, dopo almeno 2 anni dal momenti in cui è stato scritto.

2.1.1 Demo

La demo conterrà almeno 5 nodi della rete, 1 server timestamp, 1 Network Visualizer per osservare l'esecuzione dei comandi. La sequenza di esecuzione suggerita è la seguente:

1. avvio del Server Timestamp;
2. avvio del Network Visualizer;
3. avvio in cascata dei nodi.

A questo indirizzo è disponibile un canovaccio, in markdown, con la struttura del report. Pandoc[4] permette di creare PDF da markdown. È possibile scrivere il report nel formato preferito (.doc, .tex), l'importante è che il PDF generato rispetti la struttura del modello.

2.2 Gruppi

I gruppi possono essere costituiti da un minimo di 3 a un massimo di 5 persone. I gruppi che intendono svolgere questo progetto, devono comunicare via email a `stefanopio.zingaro@unibo.it` entro il **10 Maggio 2018** la composizione del gruppo. L'email deve avere come oggetto **GRUPPO LSO** e contenere:

1. Nome del gruppo;
2. Una riga per ogni componente del gruppo, con cognome, nome e matricola;
3. Un indirizzo email di riferimento a cui mandare le notifiche al gruppo, sarà poi suo incarico trasmetterle agli altri membri.

Email di esempio con oggetto **GRUPPO LSO**
NomeGruppo

- *Zingaro, Stefano, 123456*
- *Pallino, Pinco, 234567*
- *Banana, Joe, 345678*

[4] `http://cs.unibo.it/~sgiallor/teaching/project/current/report_template.md`

Referente: stefano.zingaro@studio.unibo.it

Chi non comunicherà la composizione del gruppo entro il **10 Maggio 2018** non potrà consegnare il progetto. Nel caso, chi non riuscisse a trovare un gruppo, lo comunichi il prima possibile, entro e non oltre il **7 Maggio 2018**, a stefano.zingaro@studio.unibo.it con una mail con oggetto **CERCO GRUPPO LSO**, specificando:

1. Cognome, Nome, Matricola, Email;
2. Eventuali preferenze legate a luogo e tempi di lavoro (si cercherà di costituire gruppi di persone con luoghi e tempi di lavoro compatibili)

Email di esempio con oggetto **CERCO GRUPPO LSO**

- *Zingaro, Stefano, 123456,* stefano.zingaro@studio.unibo.it

Preferisco trovarci nei pressi del dipartimento, tutti i giorno dopo pranzo.

Le persone senza un gruppo verranno raggruppate il prima possibile **e non sarà possibile modificare i gruppi formati**.

2.3 Consegna e Date

Lo sviluppo del progetto avviene col supporto di *Git*[5], così come la consegna del progetto. I gruppi devono creare un repository su GitLab[6] seguendo la procedura descritta sotto.
Creazione del gruppo:

- ogni membro del gruppo crea un account su GitLab;
- il referente del gruppo:
  - crea un progetto cliccando sul + in alto a destra nella schermata principale di GitLab, inserendo il nome "LabSO_NomeGruppo" (dove NomeGruppo è il nome registrato in precedenza del gruppo) e cliccando su **Create Project**;
  - una volta che il progetto è stato creato, aggiunge membri al progetto andando su **Settings** > **Members**;
  - aggiunge ogni membro del gruppo come **Developer**, cercandoli in base allo username registrato su GitLab;
  - aggiunge l'utente "@stefanopiozingaro" come **Reporter**.

[5] http://rogerdudler.github.io/git-guide/index.it.html
[6] http://gitlab.com

Al momento della consegna, il repository dovrà contenere i sorgenti del progetto e la relazione, nominata **REPORT_LSO.pdf**. È possibile inserire un file di nome **README** che riporti istruzioni su come lanciare gli eseguibili. Per effettuare la consegna, il **referente del gruppo** crea un **Tag** del progetto. Per creare un **Tag** del progetto:

1. nella pagina di progetto su GitLab, cliccare sulle voci **Repository** > **Tags** > **New Tag**;
2. digitare come **Tag Name** il nome **Consegna** e lasciare gli altri campi vuoti;
3. cliccare su **Create Tag** per eseguire la creazione del **Tag** di consegna.

Una volta creato il Tag, inviare una email di notifica di consegna con soggetto **CONSEGNA LSO - NOME GRUPPO** a `stefanopio.zingaro@unibo.it`. Il report va anche consegnato in forma cartacea nella casella del prof. Sangiorgi (piano terra del Dipartimento di Informatica, a fianco del suo ufficio). Ci sono due date di consegna disponibili:

- le 23.59(UTC+1) di Lunedí 2 Luglio 2018;
- le 23.59(UTC+1) di Lunedí 17 Settembre 2018.

Come data di consegna, farà fede la data di creazione del Tag su GitLab. In seguito alle consegne, verranno fissate data e ora della discussione del progetto compatibilmente coi tempi di correzione. La data di discussione verrà notificata ai gruppi tramite la mail di riferimento.

2.3.1 Valutazione

Il progetto verrà valutato mediante una discussione di gruppo, con tutti i componenti del gruppo presenti. Non è possibile per i componenti di un gruppo effettuare la discussione in incontri separati. Al termine della discussione, ad ogni singolo componente verrà assegnato un voto in base all'effettivo contributo dimostrato nel lavoro di progetto. La valutazione del progetto è indipendente dal numero di persone che compongono il gruppo.

2.3.2 Note Importanti

- non si accettano email o richieste di eccezioni sui progetti con motivazioni legate a esigenze di laurearsi o di non voler pagare le tasse per un altro anno.

- chi copia o fa copiare, anche solo parte del progetto, si vedrà invalidare completamente il progetto senza possibilità di appello. Dovrà quindi rifare un nuovo progetto l'anno successivo. Durante la correzione verrà utilizzato un tool per la ricerca di codice copiato.

2.4 Domande sul Progetto e Ricevimento

Per le domande sul progetto si consiglia di usare il newsgroup del corso. Risposte corrette alle domande dei colleghi sul newsgroup verranno valutate positivamente in sede di esame. È possibile chiedere informazioni o un ricevimento via email, specificando la motivazione della richiesta, a `stefanopio.zingaro@unibo.it`.