Università di Pisa

Università di Pisa, Dipartimento di Informatica

Laboratorio di Reti - A.A. 2021-2022

Docente: Federica Paganelli

Relazione progetto Winsome

Nicola Vetrini
23 dicembre 2021

Indice

1 Introduzione

Lo scopo del progetto è stata la realizzazione di una rete sociale, Winsome, basata sulla condivisione di contenuti tra un insieme di utenti.
Ciascun utente alla registrazione specifica una lista di tag a cui è interessato ed ha la possibilità, in seguito, di seguire altri utenti che hanno almeno un tag in comune. Per ogni utente vi è quindi un blog personale, nel quale egli può pubblicare dei post, ed un feed contenente i post degli utenti che segue. Ciascun utente può anche decidere di votare positivamente o negativamente un post, commentarlo o effettuarne il rewin (analogo del retweet, seppur con limitazioni).
Winsome ha inoltre un meccanismo di reward sia per i creatori di post che hanno delle interazioni, sia per chi commenta o vota un post; i reward sono calcolati periodicamente, nella valuta di Winsome: i wincoin, che possono essere convertiti in altre valute elettroniche (il server supporta solamente la conversione bitcoin).

2 Server Winsome

2.1 Argomenti da riga di comando

aaa

2.2 Configurazione

Il server è configurato attraverso un file JSON che viene cercato, nell'ordine, ai seguenti path:

1. il path passato attraverso l'opzione `-c <path>`
2. `config.json` nella directory corrente
3. `./data/WinsomeServer/config.json` il file di configurazione di default

Se tutte le opzioni precedenti non contengono un file di configurazione valido allora il server termina immediatamente.
I parametri configurabili sono i seguenti:

dataDir : Il path alla directory contenente i dati del server, ovvero la directory che contiene il file `users.json`, la directory `blogs`, etc ...

registryPort : La porta del registry RMI nel quale inserire lo stub per la registrazione. Il valore deve essere un intero nel range $[0, 65535]$

serverSocketAddress : Nome host o indirizzo IP del ServerSocket creato dal server per accettare le connessioni dai client (IP pubblico del server nel caso generale, ma in questo caso localhost)

serverSocketPort : Porta alla quale deve essere legato il socket sopracitato, nel range $[0, 65535]$

minPoolSize : Intero (> 0) che rappresenta il numero di core thread nella threadpool per la gestione delle richieste del server

maxPoolSize : Intero ($x : INT\_MIN \leq minPoolSize \leq x < INT\_MAX$) che rappresenta il numero massimo di thread che possono essere gestiti contemporaneamente dalla threadpool

workQueueSize : Dimensione della coda di task che la threadpool può accumulare in attesa che un thread del pool sia libero, in accordo con la politica di gestione delineata dalla documentazione di ThreadPoolExecutor

retryTimeout : long (> 0) che rappresenta il numero di millisecondi che l'handler per la gestione delle richieste rifiutate dal threadpool attende prima di provare a sottomettere nuovamente la richesta

Non è necessario specificare tutti i parametri nel file, poiché quelli assenti assumeranno i valori di default definiti nella classe `WinsomeServer/ServerConfig.java`. Il file JSON viene deserializzato utilizzando Jackson con ObjectMappper in un'istanza della classe menzionata, ed un riferimento ad essa viene passato alla creazione del server, nel main.
Nella classe è in realtà presente anche il campo (non serializzato) configFile, che serve soltanto per determinare, nella funzione `getServerConfiguration()` in `main()`, se era stato settato con l'opzione -c un path da riga di comando per il file di configurazione, che prende la precedenza rispetto a quelli di default.

2.3 Esecuzione

Il codice del server Winsome è contenuto all'interno del package WinsomeServer; la classe ServerMain in tale package contiene il metodo main, quindi è quella da eseguire per far partire il server.

2.3.1 Thread e gestione della concorrenza

Il server Winsome, come già menzionato, è multithreaded: vi è un thread principale, il cui compito è l'inizializzazione del server e del registry; dal thread principale è mandato in esecuzione un istanza di WinsomeServer, sottoclasse di Thread, che si occupa di gestire lo smistamento delle richieste attraverso un selector. Le richieste sono passate ad una threadpool le cui dimensioni minime e massime sono fissate dal file di configurazione. All'avvio, inoltre, vengono attivati tanti thread quanti sono gli utenti letti dal file `users.json` per deserializzare dal file il contenuto del loro blog

3 Client Winsome

client

4 Compilazione ed esecuzione

4.1 Compilazione

4.2 Esecuzione

4.3 Documentazione

È disponibile nel makefile un target (make doc) per generare con javadoc la documentazione delle classi realizzate, visualizzabile tramite browser.
Con l'opzione -link di javadoc si ottengono link cliccabili alla documentazione delle classi della libreria standard Java e delle librerie utilizzate, anche se ciò potrebbe introdurre un leggero ritardo nella generazione, per cui le righe contenenti -link possono essere tolte, se necessario.

5 Librerie utilizzate

Le librerie esterne utilizzate dal software sono state incluse, sotto forma di file .jar, nella cartella *libs*

5.1 Apache Commons CLI

È stata utilizzata la libreria Apache Commons CLI versione 1.5.0 per il parsing degli argomenti da riga di comando sia in WinsomeServer che in WinsomeClient.

5.2 Jackson

All'interno del software, soprattutto nella componente server, è stata utilizzata ripetutamente la serializzazione e deserializzazione tra classi ed oggetti JSON scritti su file. A tale scopo ho scelto di utilizzare la libreria Jackson, versione 2.9.7.