

Progetto di Sistemi Distribuiti

Laboratorio di Sistemi Distribuiti

Maggio 2012 - versione 2

Attenzione: le parti modificate rispetto alla versione precedente sono evidenziate in giallo.

1 Descrizione del progetto

Scopo del progetto è implementare un sistema distribuito, denominato “I-Bay”, che sarà di seguito descritto.

1.1 Descrizione del sistema

I-Bay è una nuovissima ed innovativa piattaforma che offrirà ai propri utenti la possibilità di vendere e comprare oggetti sia nuovi sia usati. L'elemento di grande novità, che cambierà per sempre il modo di acquistare online, è che gli oggetti non avranno un prezzo fisso stabilito dal venditore ma saranno a prezzo dinamico, deciso tramite delle “aste online”.

La principale differenza tra un'asta tradizionale e un'asta online è rappresentata dal fatto che le trattative si svolgono virtualmente, consentendo agli acquirenti di accedere ovunque all'asta.

Il funzionamento delle aste online è semplice: il bene viene messo all'asta ad un prezzo minimo fissato, detto “base d'asta”, al di sotto del quale non vi saranno assegnazioni. I compratori possono rilanciare e gli importi vanno ad accumularsi alla base d'asta. Per tutta la durata dell'asta i partecipanti, sono costantemente avvisati delle variazioni dei prezzi e di eventuali rilanci, in modo da poter intervenire direttamente nelle contrattazioni. Ogni puntata inoltre fa scattare un conto alla rovescia di k secondi (ad esempio $k=30s$) ed ogni nuova offerta effettuata prima dello scadere del tempo resetta il conto alla rovescia.

Si aggiudica l'asta l'ultimo l'acquirente che effettua un'offerta quando il conto alla rovescia arriva a zero.

2 Sincronizzazione

Per un corretto funzionamento dell'asta è necessario che non siano effettuate offerte con importo minore del prezzo corrente dell'oggetto nè offerte dal medesimo importo da acquirenti differenti.

Si richiede di implementare un algoritmo centralizzato che permetta la mutua esclusione per l'accesso alla variabile contenente il prezzo corrente dell'oggetto tramite un lock esplicito. Il coordinatore dell'algoritmo sarà il venditore dell'oggetto.

Analizziamo ora i componenti del sistema.

3 Le componenti del sistema

Le componenti del sistema sono tre: Seller, Buyer e AuctionHouse.

3.1 Seller

L'applicazione Seller provvede alla registrazione dell'oggetto da vendere presso la casa d'aste. Una volta che l'oggetto è stato correttamente registrato presso la casa d'aste, quest'ultima mette in contatto il venditore con gli acquirenti: l'applicazione Seller riceve quindi le connessioni da parte degli acquirenti. N.B. le comunicazioni tra Seller e i Buyer devono avvenire mediante socket.

Il venditore attende l'arrivo di un numero prefissato di acquirenti (ad. esempio 4) e dà quindi inizio all'asta: solo a questo punto gli acquirenti possono fare le proprie offerte. Come già descritto, se non sono effettuati rilanci entro un lasso di tempo fissato, l'oggetto viene assegnato all'acquirente che aveva effettuato l'ultima offerta più alta: l'applicazione Seller è responsabile di tenere traccia del tempo trascorso dall'ultima offerta valida e della chiusura dell'asta stessa.

Alla chiusura dell'asta l'applicazione Seller comunica il vincitore ed il prezzo di vendita dell'oggetto

3.2 Buyer

L'applicazione Buyer deve connettersi alla casa d'aste per ottenere la lista di oggetti acquistabili ed, una volta scelto l'oggetto desiderato, iscrive l'acquirente all'asta. L'applicazione, durante l'asta, deve sia permettere di effettuare nuove offerte tramite linea di comando, sia aggiornare l'utente di eventuali rilanci da parte degli altri acquirenti.

Al termine dell'asta l'applicazione notifica all'acquirente se è riuscito ad aggiudicarsi l'oggetto e, in caso contrario, comunica l'acquirente vincente ed il prezzo di vendita.

3.3 AuctionHouse

L'applicazione AuctionHouse rappresenta la casa d'aste, che si occupa di:

- gestire le iscrizioni dei venditori
- rendere disponibile una lista di oggetti in vendita
- gestire le iscrizioni degli acquirenti
- mettere in contatto venditori ed acquirenti

Tutte le comunicazioni dai venditori ed acquirenti verso la casa d'aste avvengono attraverso RMI. L'applicazione rende disponibili i seguenti metodi eseguibili da remoto:

```
ItemID[] getOpenAuctions();
```

```
SellerId joinAuction(int itemID);
```

```
boolean addItemToSell( String description,  
                        String sellerName, int port, double minimumPrice);
```

La classe `ItemId` rappresenta l'identificativo di ogni oggetto. Ogni oggetto viene descritto da un numero univoco, da una descrizione testuale che ne presenta le caratteristiche e dalla base d'asta.

La classe `SellerId` rappresenta l'identificativo di ogni venditore. Ogni venditore è caratterizzato da un nome (una stringa), dall'indirizzo IP e dalla porta sulla quale l'applicazione `Seller` è in ascolto. Un venditore non può vendere più oggetti contemporaneamente.

Il metodo `getOpenAuctions` restituisce l'elenco di tutti gli oggetti in vendita e la cui asta non ha ancora avuto inizio. Dopo aver invocato tale metodo, l'acquirente, se interessato, può decidere di partecipare all'asta per un oggetto tramite il metodo `joinAuction`.

Il metodo `joinAuction` registra un nuovo acquirente all'asta dell'oggetto specificato e restituisce l'identificativo del venditore. Dopo aver invocato tale metodo è compito dell'applicazione `Buyer` comunicare al venditore i propri dati. I parametri del metodo sono: l'ID che identifica l'oggetto che l'acquirente si vorrebbe aggiudicare, il nome dell'acquirente e la porta sulla quale la relativa applicazione `Buyer` rimane in attesa di connessioni. L'indirizzo IP viene ottenuto automaticamente dalla `AuctionHouse`. Se si verifica qualche problema (esempio: è già iniziata l'asta per l'oggetto desiderato), il metodo ritorna `null`.

Quando l'asta raggiunge il numero di acquirenti prefissato, l'applicazione AuctionHouse marca l'asta come già iniziata e non permette l'iscrizione di nuovi acquirenti.

N.B. I parametri `buyerName` e `port` sono opzionali.

Il metodo `addItemToSell` viene invocato dal venditore per mettere in vendita un oggetto. I parametri del metodo sono: una stringa che descrive l'oggetto da vendere, il nome del venditore, la porta sulla quale la relativa applicazione Seller rimane in attesa di connessioni e l'importo da cui far partire l'asta (ovvero la base d'asta). È compito della AuctionHouse assegnare ad ogni oggetto in vendita un identificativo univoco. Il metodo ritorna `true` se l'operazione va a buon fine; se invece si è verificato qualche problema il metodo restituisce `false`.

4 Semplificazioni e limitazioni

Si ricorda che lo scopo del progetto è dimostrare la capacità di progettare e realizzare un'applicazione distribuita. Pertanto gli aspetti non riguardanti il protocollo di comunicazione e la concorrenza sono considerati secondari.

Inoltre è possibile assumere che :

- nessun nodo (Buyer, Seller o AuctionHouse) si comporti in maniera maliziosa,
- nessun nodo (Buyer, Seller o AuctionHouse) termini in maniera incontrollata

Si gestiscano invece i possibili errori di inserimento dati da parte dell'utente.

Sebbene le librerie di Java forniscano molteplici classi per la gestione di situazioni di concorrenza, per fini didattici gli studenti sono invitati a fare esclusivamente uso di metodi e di classi spiegati durante il corso di laboratorio. Pertanto, eventuali strutture dati di sincronizzazione necessarie (come ad esempio semafori o buffer condivisi) dovranno essere implementate da zero e saranno discusse durante la presentazione del progetto.

Nonostante alcune problematiche di sincronizzazione possano essere risolte tramite l'implementazione di server iterativi, per fini didattici si richiede di utilizzare server multithread.

5 Esempio di esecuzione

sottolineato = inserito dall'utente

5.1 Buyer

Quale nome vuoi usare?

Letizia

Quale porta vuoi usare?

1234

A quale server (RMI) ti vuoi connettere (lascia vuoto per usare localhost)?

mercurio.srv.dico.unimi.it

A quale porta del server (RMI) ti vuoi connettere (lascia vuoto per usare 1099)?

1100

Ottingo lista oggetti da server... fatto!

Seleziona l'oggetto che ti interessa:

1. Vaso finto antico, prezzo base 50 euro
2. Camicia hawaiana, prezzo base 15 euro
3. Soprammobile di plastica con putti, prezzo base 150 euro

2

Venditore: Sergio

Acquirenti iscritti: Dragan, Andrea

In attesa di altri acquirenti...

Nuovo acquirente: Marco

Via dell'asta!

Inserire offerta maggiore di 15 euro:

Marco offre 18 euro

20

Hai fatto un'offerta di 20 euro!

Dragan offre 21 euro

21 euro e 1!

21 euro e 2!

Andrea offre 30 euro

30 euro e 1!

30 euro e 2!

30 euro e 3!

Peccato, non hai vinto l'asta.

Andrea si aggiudica l'oggetto, camicia hawaiana, per 30 euro.

5.2 Seller

Quale nome vuoi usare?

Andrea

Quale porta vuoi usare?

1243

A quale server (RMI) ti vuoi connettere (lascia vuoto per usare localhost)?

mercurio.srv.dico.unimi.it

A quale porta del server (RMI) ti vuoi connettere (lascia vuoto per usare 1099)?

1100

Inserisci descrizione oggetto:

Vaso finto antico

Inserisci base d'asta:

50

Attendo acquirenti...

Nuovo acquirente: Letizia

Nuovo acquirente: Laura

Nuovo acquirente: Christian

Nuovo acquirente: Daniela

Via dell'asta!

Letizia offre 52 euro

Daniela offre 60 euro

Nessuna offerta per 20 secondi

60 euro e 1!

Attendo 5 s...60 euro e 2!

Attendo 5 s...60 euro e 3!

Chiudo l'asta.

Hai venduto il tuo oggetto per 60 euro!

Daniela si aggiudica l'oggetto: vaso finto antico.

6 Presentazione del progetto

Il progetto è da svolgere individualmente. Durante la valutazione del progetto verrà richiesto di mostrare alcune parti del programma, verrà verificata la padronanza del codice presentato, verrà verificato il corretto funzionamento del programma e verranno inoltre poste alcune domande di carattere teorico inerenti gli argomenti trattati nel corso (parte di laboratorio). E' possibile presentare il progetto sul proprio computer.

Il codice sorgente dovrà essere consegnato al docente prima della discussione del progetto. Per la consegna, è sufficiente archiviare il codice in un file zip, rinominato con il proprio numero di matricola (es. 654321.zip) ed effettuare l'upload dello stesso tramite il sito <http://upload.dico.unimi.it>. Sarà possibile effettuare la consegna a partire da una settimana prima della data di ogni appello. La consegna deve essere tassativamente effettuata entro le 23:59 del secondo giorno precedente quello della discussione (es. esame il 13

mattina, consegna entro le 23.59 dell'11). Eventuali modifiche apportate al progetto dopo il termine della consegna devono essere segnalate al docente prima della discussione.

Si invitano inoltre gli studenti ad utilizzare, durante la presentazione, le istruzioni `Thread.sleep()` al fine di mostrare la correttezza della sincronizzazione del proprio programma. Si consiglia vivamente di analizzare con attenzione tutte le problematiche di sincronizzazione e di rappresentare lo schema di comunicazione fra le componenti. Questo schema deve rappresentare il formato dei messaggi e la sequenza delle comunicazioni che avvengono tra le componenti in occasione delle varie operazioni che possono essere svolte. Tale schema sarà di grande aiuto, in fase di presentazione del progetto, per verificare la correttezza della parte di sincronizzazione distribuita.

Nel caso in cui il docente si accorga che una parte significativa del progetto consegnato non è stata sviluppata dallo studente titolare dello stesso, lo studente in causa: a) dovrà superare nuovamente tutte le prove che compongono l'esame del corso. In altre parole, se l'esame è composto di più parti (teoria e laboratorio), lo studente dovrà sostenere nuovamente tutte le prove in forma di esame orale (se la prova di teoria fosse già stata superata, questa verrà annullata e lo studente in merito dovrà risostenerla). b) non potrà sostenere nessuna delle prove per i 2 appelli successivi. Esempio: supponiamo che gli appelli siano a Febbraio, Aprile, Giugno, Luglio, e che lo studente venga riconosciuto a copiare all'appello di Febbraio. Lo studente non potrà presentarsi agli appelli di Aprile e Giugno, ma potrà sostenere nuovamente le prove dall'appello di Luglio in poi. Il docente si riserva la possibilità di assegnare allo studente in causa lo svolgimento di una parte integrativa o di un nuovo progetto.

7 Parti facoltative

7.1 Prima parte

La mutua esclusione descritta nella Sezione 2 deve essere gestita in modo distribuito. Si chiede di ottenere la mutua esclusione tramite un algoritmo ad anello. Le comunicazioni tra Buyer devono avvenire mediante socket.

7.2 Seconda parte

Il sistema richiede che ogni acquirente e venditore invii e riceva parecchi messaggi durante lo svolgimento dell'esperimento. Creare una nuova socket per l'invio di ogni singolo messaggio è una soluzione che richiede un certo overhead della rete, in quanto le operazioni di stabilimento e di chiusura di connessione, seppur trasparenti, richiedono l'invio di pacchetti di rete aggiuntivi. Si modifichino le applicazioni Buyer e Seller in modo che sia mantenuta una connessione persistente per le comunicazioni tra di essi.

8 Aggiornamenti

Qualora fosse necessario, il testo dell'attuale progetto verrà aggiornato al fine di renderne più semplice l'interpretazione. Le nuove versioni del progetto verranno pubblicate sul sito del corso. Si consiglia agli studenti di controllare regolarmente il sito.

Al fine di incentivare la presentazione del progetto nei primi appelli disponibili, lo svolgimento della parte facoltativa 1 diventa obbligatoria a partire dall'appello di Settembre (incluso), mentre entrambe le parti facoltative (1 e 2) saranno obbligatorie negli appelli successivi.