Simulazione DEUS Carsharing P2P

MODELLO

Il modello che si è andato a simulare è quello di un carsharing peer to peer, ovvero ogni persona automunita può mettere a disposizione la propria automobile per dare la possibilità di utilizzo a terze parti sotto una ricompensa oraria in denaro.

Il progetto in questione si pone l'obiettivo di simulare la differenza di remunerazione fra un sistema di prenotazione privo di particolari meccanismi di accettazione rispetto al sistema di prenotazione svolto nel paper scientifico "A simulation study of Peer-to-Peer carsharing" (Hampshire 2011).

La simulazione avviene attraverso il simulatore ad eventi discreti DEUS scritto in linguaggio Java, che fornisce la possibilità di avere comodi Log dei risultati e di generare file compatibili con GnuPlot per graficare i risultati della simulazione.

Viene simulata la presenza di 20 macchine (10 dedicate al test con prenotazione senza meccanismo di selezione e altre 10 dedicate al test con prenotazione secondo modello Hampshire) all'interno di una zona rappresentata come un quadrato di dimensioni 100 x 100. Ogni macchina avrà le proprie caratteristiche fra cui le coordinate (x, y) e un array di 24 elementi che rappresenta la disponibilità oraria nelle 24 ore (0 macchina libera, 1 macchina prenotata).

Per ogni macchina verrà associata un numero randomico di richieste di prenotazione (con distribuzione di Poisson a media 5 e media 10) che includeranno due caratteristiche: il tempo di inizio del noleggio e la durata del noleggio. La durata media è stata studiata in paper scientifici precedenti e si attesta sulle 3,93 h, da cui è stato successivamente approssimato, facendola diventare una distribuzione geometrica con media 4.

Si suppone, per facilitazione, che le richieste di prenotazione dei veicoli che arrivano nella giornata che viene simulata siano tutte destinate al giorno successivo.

Alla fine di questa relazione è presente una sezione 'risultati' che mostra la variazione di guadagno percentuale dell'utilizzazione e del ricavo giornaliero delle macchine a seconda del meccanismo di prenotazione implementato.

CARSHARING.XML

Il seguente file XML di configurazione per il simulatore si compone di quattro elementi chiave:

- Eventi
- Processi
- Nodi
- Simulazione

EVENTI

Sono stati dichiarati tre eventi per rappresentare:

- La creazione delle automobili (evento Birth)

- La richiesta da parte dei clienti per ogni automobile (Evento Request)
- Log del sistema per avere un feedback sul risultato dell'elaborazione. (Evento di Log)

Ogni evento ha associato una classe implementata in Java che lo gestisce.

PROCESSI

Ad ogni evento è stato associato un processo di natura diversa a seconda dell'esigenza specifica che andava soddisfatta. L'evento di creazione iniziale delle automobili e della parte di Log del sistema è stato associato ad un semplice processo periodico rettangolare con cadenza di un'unità per entrambi, che si svolgono in differenti momenti della simulazione.

L'evento di richiesta di prenotazione è stato associato ad un processo di Poisson rettangolare per simulare al meglio l'arrivo di clienti con distribuzione di Poisson con tempo medio inizialmente di 5 minuti e poi con 10 minuti. Il tempo medio è stato scelto cercando di far arrivare un numero equo di richieste giornaliere per ogni macchina.

Ogni processo ha una sua data di inizio e di fine specificata attraverso specifici parametri.

NODI

È presente un solo nodo specifico che rappresenta la macchina. Al suo interno viene dichiarato come parametro la dimensione del lato della zona in cui dovranno essere generate casualmente le automobili. Ad esso viene associata la classe Car.java.

SIMULAZIONE

Viene specificata la durata massima possibile della simulazione assieme alla dichiarazione del seed (nel caso non venga inserito a mano diversamente in fase di compilazione) dal quale prenderanno origine i numeri casuali creati. Inoltre, sono specificati quali dei processi dichiarati precedentemente dovranno essere eseguiti nella simulazione. Il tempo della simulazione è stato impostato facendo si che un'unità di tempo simulato corrisponda ad 1 minuto nella vita reale. Quindi una giornata di 24 ore (1440 minuti) corrisponderà a 1440 unità temporali simulate.

CLASSE BIRTHEVENT

Classe assegnata all'evento birth nel quale attraverso il processo rettangolare analogo vengono creati e quindi istanziati ad ogni intervallo unitario una macchina situata in posizione spaziale diversa in un quadrato di lato 100.

CLASSE CAR

Classe che rappresenta l'astrazione fisica della macchina. Estende la classe Node di DEUS. Si compone dei seguenti attributi:

- Coordinate spaziali x e y
- Array d di 24 elementi che rappresenta il tempo disponibile per la prenotazione

Troviamo poi diversi metodi fra cui:

- Getter and Setter per impostare e aggiornare dall'esterno i valori degli attributi

- Costruttore e Inizializzazione dell'oggetto
- Clone, nel quale vengono specificate le differenze che dovranno avere gli oggetti della stessa classe fra di loro una volta istanziati.

CLASSE REQUEST

Classe collegata al processo di richiesta, da parte di un utente, di una prenotazione. Estende la classe Event che ci obbliga a richiamare il metodo run (), nel quale vengono specificate le azioni che vogliamo vengano intraprese durante l'esecuzione del processo. Il processo è basato sull'arrivo delle prenotazioni con una distribuzione di Poisson con media iniziale impostata a 5 e successivamente a 10 minuti.

Qui possiamo trovare la gestione dell'accettazione o meno di un cliente a seconda della macchina scelta. Ad ogni evento di richiesta viene generato un numero casuale fra 0 e il numero di macchine per selezionare l'automobile preferita, un numero casuale con media geometrica pari a 4 per rappresentare la durata media della prenotazione e un altro numero casuale fra 0 e 23 per rappresentare l'orario di inizio noleggio.

Le richieste vengono processate in parallelo attraverso due algoritmi differenti. Uno accetta tutte le richieste compatibili senza fare però nessuna selezione particolare, mentre l'altro algoritmo accetta le richieste solo con durata >= 4 ore fino a che non ha riempito almeno 2/3 del tempo disponibile per il noleggio, dopodiché procede a riaccettare anche prenotazioni con durata inferiore per cercare di massimizzare l'utilizzazione totale.

In ognuno dei due algoritmi vengono effettuati controlli preventivi prima di accettare qualsiasi prenotazione che verificano che il tempo di inizio più la durata di noleggio non oltrepassi le 24 ore della giornata o che non si sovrapponga a noleggi già assegnati.

Ad ogni ora di noleggio è stato poi assegnato un costo orario che il cliente pagherà al noleggiatore della macchina. Il costo è stato scelto in base a studi scientifici precedenti e imposto pari a 5 dollari orari.

Il confronto tra i risultati dei due algoritmi si trova in fondo a questa relazione nell'apposita sezione.

CLASSE LOG

Anche la classe Log estende la classe Event. Infatti, anch'essa è comandata da un processo rettangolare con cadenza unitaria che va a scrivere su un file di Log diverse informazioni utili al fine della simulazione. In questo caso nel metodo run() ad ogni minuto del giorno (quindi 1440 volte) vengono scritti su file il tasso di riempimento dell'auto e il relativo ricavo in dollari, sia dell'algoritmo senza restrizioni sia dell'algoritmo secondo il paper di Hamspshire descritto precedentemente.

RISULTATI

Dai risultati mediati con **10 seed diversi** possiamo notare come con i parametri impostati, l'algoritmo di Hampshire di selezione delle prenotazioni garantisca in entrambi i casi (distribuzione di Poisson con media 5 e media 10) un ricavo medio superiore rispetto ad una accettazione casuale per ogni singola giornata. Più precisamente, analizzando l'intera flotta di macchine dedicate al carsharing

P2P, nel caso con media di arrivi a 5 minuti con algoritmo senza selezione sulla prenotazione abbiamo un ricavo giornaliero medio di **826,2 $** mentre con l'algoritmo di Hampshire il ricavo medio sale a **873,1 $**. Un incremento delle entrate pari al **5,6%**. Questo grazie ad una migliore utilizzazione dei veicoli che passa dal **68,8%** al **72,7%**. Il ricavo viene calcolato semplicemente moltiplicando le ore di utilizzo delle singole macchine per la tariffa oraria, impostata a 5$.

Nel caso a media 10 minuti con algoritmo senza selezione sulla prenotazione abbiamo un ricavo giornaliero medio di **690,6 $** mentre con l'algoritmo di Hampshire il ricavo medio sale a **715,6 $**. Un incremento delle entrate pari al **3,6%**. L'utilizzazione in questo caso passa da **57,5%** a **59,6%**.

Come si può notare dai grafico sottostanti che confrontano i risultati dei due algoritmi, l'Hampshire con media a 5 minuti restituisce un tasso di riempimento più lento solo nella prima metà della giornata, mentre con media a 10 minuti il guadagno rispetto all'algoritmo "Random" risulta meno accentuato e il tasso di riempimento aumenta solo verso la fine della giornata. Entrambi però, garantiscono mediamente una resa superiore sia in termini di utilizzazione media delle automobili sia, di conseguenza, in termini di ricavi alla fine di ogni giornata.

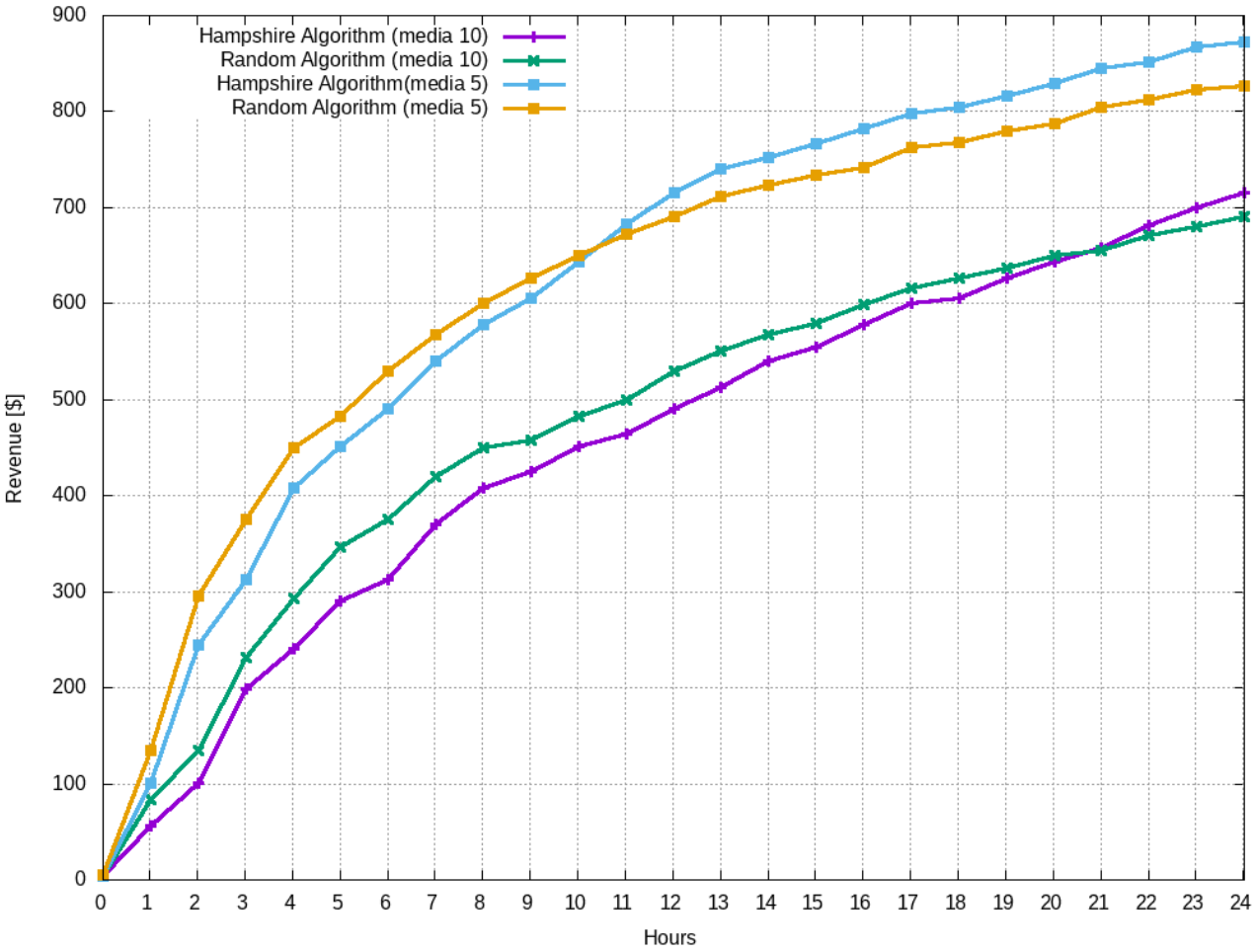

Figura 1: Confronto ricavo giornaliero fra i due diversi algoritmi

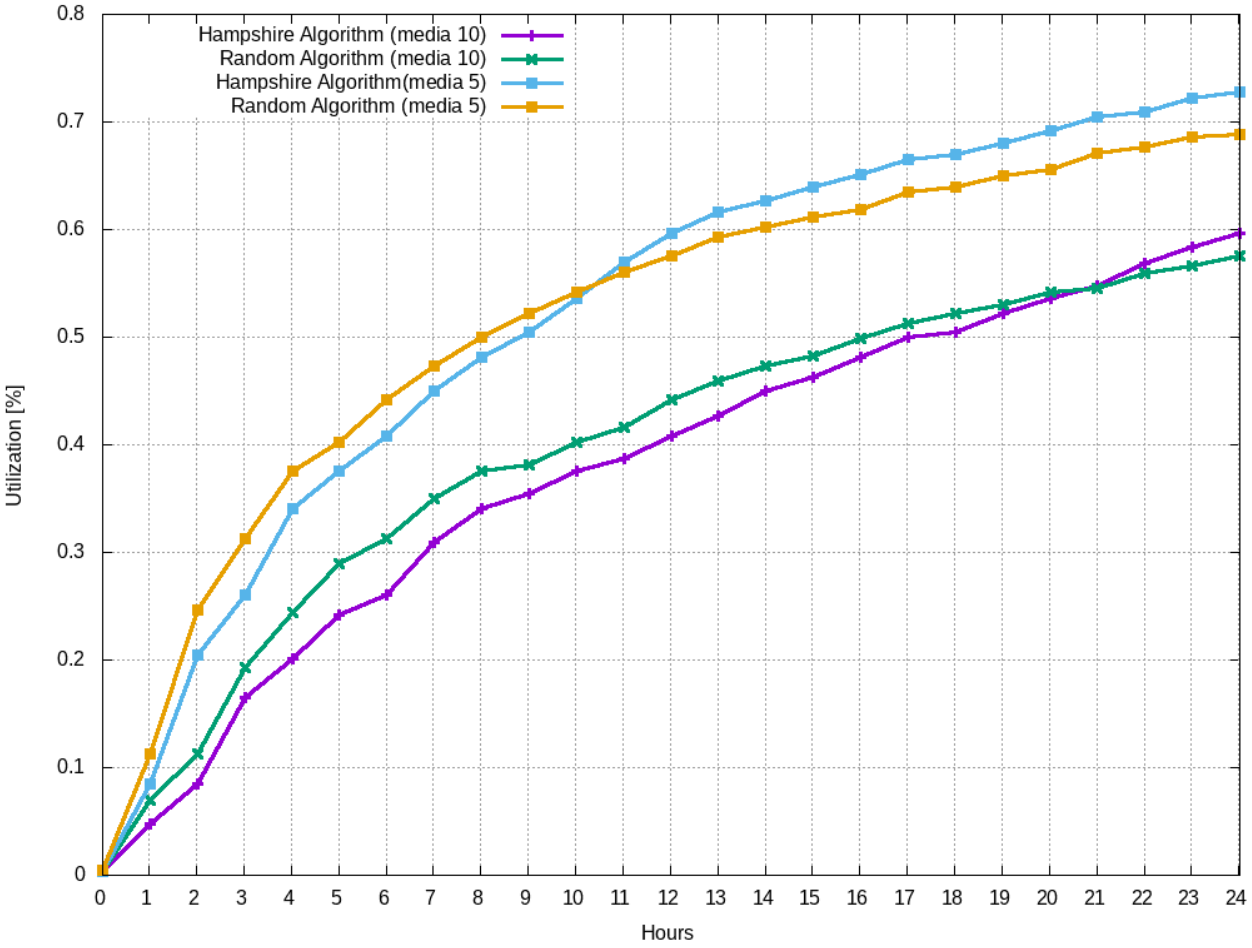

Figura 2:Confronto utilizzazione giornaliera fra i due diversi algoritmi

Riferimento bibliografico: A simulation study of Peer-to-Peer carsharing (Hampshire 2011)