

Ordinamento distribuito nel mondo reale

Matteo Dell'Amico

Università di Genova

Sommario

In questo laboratorio, mostreremo un modo di introdurre gli algoritmi distribuiti di ordinamento distribuito, un tema estremamente complesso, facendo leva sull'intuizione che gli studenti hanno nello svolgere compiti manipolando in squadra un oggetto fisico: un mazzo di carte numerate. Questa attività è preferibilmente diretta a partecipanti alle prime armi con l'informatica per introdurre il tema degli algoritmi, e dura all'incirca mezz'ora.

1 Avvertenza

Consiglio a chi vuole partecipare all'attività di laboratorio di non leggere oltre, perché l'attività è concepita per persone che arrivano senza sapere cosa succederà.

2 Motivazione

Chi è esperto di algoritmi spesso fa con profitto leva sull'intuizione per capirli. Chi invece sta imparandoli, soprattutto quando è alle prime armi, spesso fa fatica a collegare le proprie intuizioni con un mondo che viene visto come astratto e distante.

Lo scopo di questo laboratorio è quello di mostrare agli studenti che sono in grado di usare intuizioni molto pratiche e apparentemente lontane da quello che studiano per risolvere un problema complesso: l'ordinamento distribuito.

3 Attività

In questo laboratorio dividiamo i partecipanti in squadre da 4-6 persone, diamo ad ogni squadra un mazzo mischiato di carte numerate da 1 a 100 e chiediamo loro di restituircelo ordinato, il più rapidamente possibile, cronometrando. Questo viene fatto senza dare ai partecipanti il tempo di coordinarsi prima di far partire il cronometro. Non si specifica se il mazzo deve venire restituito in ordine crescente o decrescente, ed entrambi i risultati sono considerati accettabili.

Dopo aver fatto questo primo esercizio, si dice ai partecipanti che l'attività verrà ripetuta identica, ma questa volta avranno il tempo di decidere che strategia adottare, mentre si verifica che i mazzi siano ordinati correttamente e li si rimischiano; il tempo del secondo tentativo viene nuovamente cronometrato.

Una terza e ultima iterazione dell'esercizio consiste nel creare gruppi più grossi (ad esempio 10-15 persone) e chiedere a questi gruppi più grossi di ordinare un singolo mazzo mischiato, sempre cronometrando.

I tempi ottenuti dai gruppi vengono marcati su una lavagna per stimolare chi partecipa a cercare di superare gli altri gruppi.

Alla fine chi facilita l'attività fa una breve spiegazione in cui collega il problema e le soluzioni che i partecipanti stessi hanno trovato ai problemi di ordinamento distribuito in informatica.

4 Risultato atteso

In questa sezione, riporto i risultati ottenuti dall'aver proposto questa attività a studenti che hanno partecipato a due stage presso l'Università di Genova nell'ambito del Percorso per le Competenze Trasversali e l'Orientamento (PCTO) nel 2023 e 2024.

Questo problema ha paralleli ampi e piuttosto evidenti con l'ordinamento distribuito in informatica, e ho notato che i gruppi riescono in maniera piuttosto affidabile a scoprire senza bisogno di suggerimenti algoritmi efficienti. Il secondo tentativo ottiene quasi sempre un risultato notevolmente migliore del primo, mostrando che gli algoritmi efficienti fanno effettivamente la differenza.

Una soluzione molto gettonata, ed efficiente, è quella di dividere il mazzo in un mazzetto per persona. I partecipanti dividono in mucchi comuni per decine le carte del loro mazzetto e successivamente ordinano ciascuno le decine. Questa soluzione può essere vista come una parente piuttosto stretta del *bucket sort* distribuito.

È molto raro che raddoppiare il numero di partecipanti che svolge l'attività abbia grossi effetti sul tempo necessario per finire l'attività. Questo effetto può essere usato per introdurre temi come *speedup* e scalabilità, mostrando come raddoppiare il numero di lavoratori dimezza il tempo necessario per risolvere il problema solo per alcuni casi di problemi (in particolare, vale la pena di raccontare di quelli conosciuti come *embarrassingly parallel*, che hanno costi limitatissimi di comunicazione e sincronizzazione).

Altri temi che emergono naturalmente dall'attività sono, ad esempio:

- come trovare un algoritmo di ordinamento efficiente: capita che alcuni gruppi adottino algoritmi lenti (ad esempio una variante dell'*insertion sort* in cui si cerca l'1, poi il 2, poi il 3...);
- come trovarne una versione distribuita efficiente: spesso capita che alcuni gruppi usino algoritmi poco propensi a venire parallelizzati (ad esempio per via di un controllo centralizzato che crea un collo di bottiglia);
- come distribuire il lavoro in maniera efficace: spesso ci sono partecipanti che si trovano impossibilitati ad aiutare, "sprecando" il loro tempo; questo è un problema ricorrente nella programmazione distribuita;
- come limitare il costo di comunicazione e sincronizzazione: spesso capita che i gruppi scelgano soluzioni che li costringono a passare buona parte del loro tempo a comunicare/sincronizzarsi con altri invece di portare avanti il lavoro.

L'idea è che un'attività di questo tipo possa servire a dare un punto di riferimento più vicino all'intuizione quando si parla dei problemi che verranno poi visti in maniera più formale in seguito.