# Sistemi Distribuiti e Cloud Computing - A.A. 2021/22
# Progetto B1: Algoritmo di clustering $k$-means in stile MapReduce e in Go

Docente: Valeria Cardellini
Dipartimento di Ingegneria Civile e Ingegneria Informatica
Università degli Studi di Roma "Tor Vergata"
cardellini@ing.uniroma2.it

## Requisiti del progetto

Lo scopo del progetto è realizzare nel linguaggio di programmazione Go un'applicazione distribuita che implementi l'algoritmo di clustering $k$-means [4] in versione distribuita secondo il paradigma di computazione MapReduce [3].

L'algoritmo $k$-means è un algoritmo di clustering che permette di suddividere un insieme di punti in $k$ gruppi o cluster sulla base della distanza euclidea [4]. Una delle euristiche più note e semplici per implementare $k$-means è quella di Lloyd, che trova in modo iterativo il centroide di ciascun cluster in cui è partizionato l'insieme di punti e quindi riassegna i punti ai cluster in base al centroide più vicino. Nella fase iniziale dell'euristica, i centroidi vengono inizializzati in modo casuale. Il numero di iterazioni può essere determinato a priori oppure dipendere dalla convergenza, utilizzando come criterio di convergenza la minimizzazione della somma dei quadrati all'interno del cluster (Within-Cluster Sum of Squares, WCSS).

L'implementazione distribuita dell'euristica di Lloyd secondo il paradigma di computazione MapReduce richiede di implementare un algoritmo iterativo, in cui ciascuna iterazione è composta da una fase di Map seguita da una fase di Reduce (si vedano le slide da 35 a 46 in [2]). Nella fase di Map, avviene la classificazione dei punti: ciascun mapper in parallelo riceve in input un sottoinsieme (o chunk) di punti dell'insieme di partenza e per ciascuno di questi punti calcola la distanza euclidea tra il punto ed i $k$ centroidi, identificando così il centroide che minimizza la distanza e al cui cluster il punto viene assegnato. Nella fase di Reduce, avviene il calcolo dei nuovi centroidi: ciascun reducer in parallelo riceve in input tutti i punti assegnati ad un determinato cluster e calcola il valore del centroide di quel cluster. Varianti dell'implementazione sopra descritta (incluso $k$-means++, che adotta una tecnica randomizzata per migliorare la fase iniziale dell'euristica) sono discusse in [1]. Per realizzare l'applicazione, si può estendere l'esercizio in Go già sviluppato durante il corso.

L'applicazione distribuita deve soddisfare i requisiti elencati di seguito.

- Realizzare un'architettura master-worker, in cui il master distribuisce il carico di lavoro tra i nodi worker, che implementano i mapper ed i reducer.
- Implementare $k$-means in modo distribuito secondo il paradigma di computazione MapReduce.
- Per il deployment dell'applicazione, si richiede di usare container Docker, di fornire i relativi file per la creazione delle immagini e di effettuare il deployment dell'applicazione su una istanza EC2 utilizzando il grant AWS a disposizione.

- *Opzionale*: gestire il crash di un mapper durante la computazione.

Si richiede di progettare l'applicazione ponendo particolare cura al soddisfacimento dei requisiti sopra elencati e rendendo facilmente configurabili gli eventuali parametri relativi all'applicazione e al suo deployment.

Si richiede infine di valutare le prestazioni dell'applicazione implementata al variare del numero di punti e del numero di mapper e reducer utilizzati, analizzando i risultati ottenuti nella relazione.

È possibile usare librerie e tool di supporto allo sviluppo del progetto, non sovrapposti con gli scopi del progetto; le librerie ed i tool usati devono essere esplicitamente indicati e brevemente descritti nella relazione.

## Scelta e consegna del progetto

Il progetto è dimensionato per essere realizzato da **1** studente. Essendo un progetto di tipo di B, il voto del progetto peserà il 25% della valutazione complessiva dell'esame.

Per poter sostenere l'esame nell'A.A. 2021/22, **entro il 26/8/2022** è necessario prenotarsi per il progetto, comunicando alla docente in una email avente come oggetto **[SDCC scelta progetto]** le seguenti informazioni:

- nome, cognome e numero di matricola;
- progetto scelto.

Nel caso in cui il numero di prenotazioni per il progetto scelto abbia raggiunto la soglia massima prevista, sarà necessario effettuare una nuova scelta tra i progetti ancora disponibili.

Eventuali modifiche nella scelta del progetto devono essere tempestivamente comunicate alla docente e con lei concordate. Non è possibile cambiare in corso di svolgimento la tipologia del progetto (ad es. passare da progetto di tipo B a progetto di tipo A).

Per ogni comunicazione via email è necessario specificare **[SDCC]** nell'oggetto della email. Il progetto è valido **solo** per l'A.A. 2021/22 e deve essere consegnato **entro il 30/11/2022**. La prova d'esame scritta deve essere superata **entro la sessione autunnale 2021/22** (appelli di settembre 2022).

La consegna del progetto deve avvenire almeno 5 giorni lavorativi prima della data (da concordare con la docente) in cui si intende sostenere la discussione del progetto. La consegna del progetto consiste nell'invio di una email alla docente avente come oggetto **[SDCC consegna progetto]** e con il corpo della mail contenente un link a spazio di Cloud storage o repository (e.g., Dropbox, Google Drive, GitHub) che contiene:

1. il codice sorgente (opportunamente commentato);
2. relazione (in formato pdf), senza codice;
3. breve howto per l'installazione, la configurazione e l'esecuzione dell'applicazione.

La relazione contiene:

- la descrizione dettagliata dell'architettura dell'applicazione e delle scelte progettuali effettuate, opportunamente motivate;

- la descrizione dell'implementazione realizzata;
- la descrizione delle eventuali limitazioni riscontrate;
- l'indicazione della piattaforma software usata per lo sviluppo dell'applicazione (incluse eventuali librerie).

Si consiglia di redarre la relazione in forma di articolo scientifico di lunghezza massima pari a 5 pagine, usando il formato ACM proceedings (`https://www.acm.org/publications/proceedings-template`) oppure il formato IEEE proceedings (`https://www.ieee.org/conferences_events/conferences/publishing/templates.html`).

## Valutazione del progetto

I principali criteri di valutazione del progetto saranno:

1. rispondenza ai requisiti;
2. originalità;
3. organizzazione del codice (leggibilità, modularità, ...);
4. organizzazione, chiarezza e completezza della relazione.

## Riferimenti bibliografici


[1] M. Bodoia. Mapreduce algorithms for k-means clustering, 2016. `https://stanford.edu/~rezab/classes/cme323/S16/projects_reports/bodoia.pdf`.

[2] V. Cardellini. MapReduce and Hadoop, 2022. `http://www.ce.uniroma2.it/courses/sabd2122/slides/MapReduce%26Hadoop.pdf`.

[3] J. Leskovec, A. Rajaraman, and J. Ullman. *Mining of Massive Datasets, chapter 2*. Cambridge University Press, 3rd edition, 2020. `http://infolab.stanford.edu/~ullman/mmds/ch2n.pdf`.

[4] J. Leskovec, A. Rajaraman, and J. Ullman. *Mining of Massive Datasets, chapter 7*. Cambridge University Press, 3rd edition, 2020. `http://infolab.stanford.edu/~ullman/mmds/ch7.pdf`.