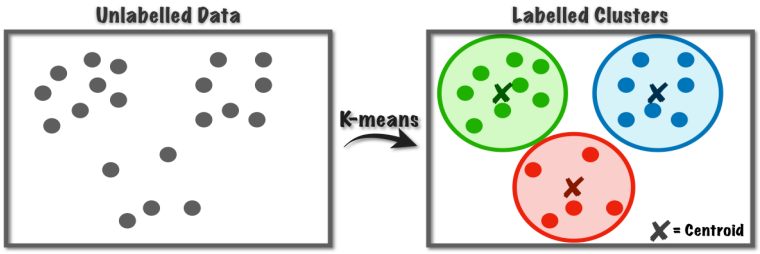
PRL 2023/24 - Implementace paralerního algoritmu K-means

Autor: Bc. Petr Pouč

Datum: 14. 4. 2023

1 Rozbor algoritmu

Cílem projektu bylo implementovat paralerní algoritmus pro shlukování vstupních dat, konkrétně 4-means. Algoritmus rozděluje data do disjunktních skupin (clustery) na základě jejich vzdáleností od středu konkrétní skupiny.



Obrázek 1: Příklad k-means (source [1]).

Program načítá data z binárního souboru, následně inicializuje prvotní vrcholy všech 4 shluků. Dále distribuuje 1 hodnoty všechny procesy pomocí funkce `MPI_Scatter`. Každý proces zároveň získá hodnoty všech počátečních vrcholů. Vrcholy jsou zaslány funkcí `MPI_broadcast`. Proces dále určí, do kterého shluku hodnota patří a výsledek odešle ostatním procesům. Vrcholy shlůků jsou poté přepočítány.

Tento proces se neustále opakuje, dokud se vrcholy shluků neustálí, tedy nedojde ke konvergenci.

2 Časová a prostorová složitost

Načtení vstupních dat má lineární časovou složitost `O(n)`, neboť operace závisí na velikosti vstupu. Zasílání hodnot pomocí funkce MPI_scatter probíhá tak, že rootovský proces vždy zašle jednu hodnotu,časová složitost opět závisí na celkovém počtu hodnot, z tohoto důvodu je časová složitost lineární `O(n)`, to samé platí pro broadcastování jednotlivých vrcholů. Hledání nejbližšího clusteru na základě vzdálenosti mezi vrcholem a hodnotou má konstatní časovou složitost `O(1)`.

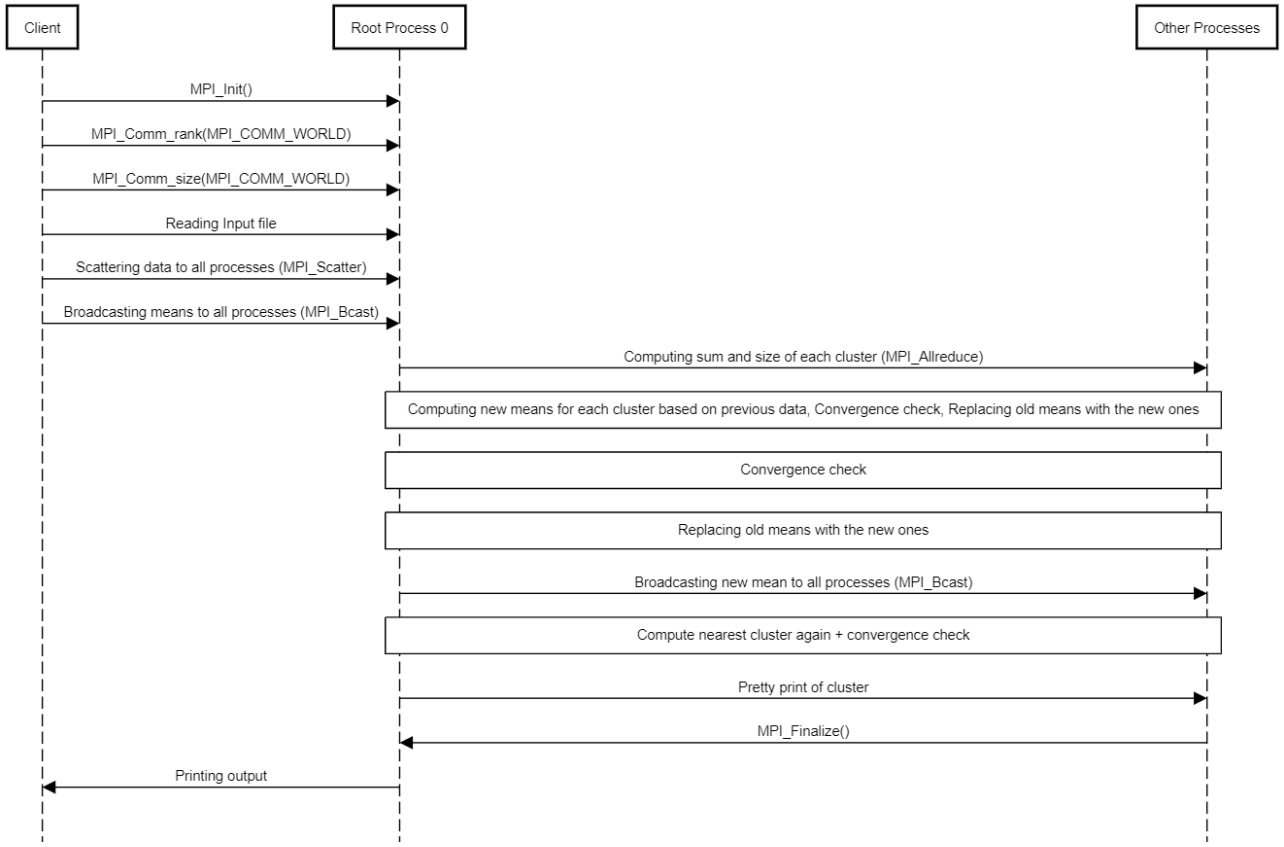
Cyklus, který se opakuje dokud nedojde ke konvergenci má pro k ¡ 5 časovou složitost lineární `O(n)` [2].//

Prostorovou složitost lze získat zkoumáním množství paměti používané každým procesem v daném čase. Výpočet nového clusteru vyžaduje O(1) místa na proces, nebo O(nprocs) místa celkem. V tomto algoritmu každý proces odesílá a přijímá jedno celé číslo pomocí MPI_Scatter a MPI_Allreduce. Prostorová složitost těhto operací by měla být taktéž uměrná velikosti vstupu, tedy lineární `O(n)`, konkrétně `O(nprocs)`.

Velikost polí pro výpočet nových clusterů je pevná a rovná se hodnotě n_means. Prostor potřebný k uložení těchto polí je tedy O(n_means). Stejná složitost platí i pro pole, ve kterém jsou uloženy nově vypočtené vrcholy clusterů.

Celková prostorová složitost daného kódu je tedy `O(nprocs + n_means + nprocs*n_means + n)`, což lze zjednodušit na `O(nprocs*n_means + n)`, kde nprocs je celkový počet procesů, n je počet vstupních čísel obsažených v binárním souboru, n_means představuje počet shluků.

3 Komunikační protokol



Obrázek 2: Sekvenční diagram paralerní implementace algoritmu k-means

4 Závěr

Algoritmus 4-means používá paralerizaci pomocí MPI knihovny, jenž umožňuje vhodně rozdělit výpočet mezi několik procesů. Algoritmus je tedy velmi efektivní i pro větší množství dat. Naopak pro malé množství dat, by mohla být paralerní verze nevhodná, pro režii distribuci dat.

Reference

[1] Jeffares, A. (2019) K-means: A complete introduction, Medium. Towards Data Science. Available at: https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c (Accessed: April 15, 2023).

[2] How slow is the K-means method? - theory.stanford.edu (no date). Available at: https://theory.stanford.edu/ sergei/papers/kMeans-socg.pdf (Accessed: April 14, 2023).