JIPET | Journal of Intelligent Procedures in Electrical Technology/ Vol. 11/ No. 41/ Spring 2020
P-ISSN: 2322-3871, E-ISSN: 2345-5594, http://jipet.iaun.ac.ir/

Implementation of Agglomerative Hierarchical Clustering Algorithm Applying Map-Reduce Parallel Approach

Fahimeh Tavakoli[1], *M.Sc.*, Faramarz Safi-Esfahani[1,2], *Assistant Professor*

1. Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran
fahimeh.tavakoli@gmail.com
2. Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran
fsafi@iaun.ac.ir

Abstract:
The map-reduce model is a method for executing large data applications. It is also a parallel programming model for writing applications that can be executed on the cloud. Organizations are increasingly producing data that is generated by business processes, user activities, website tracking, sensors, finance, accounting, and more. Data clustering algorithms are used as tools for analyzing large volumes of data. The main purpose of these algorithms is to categorize data into clusters so that the data objects in each cluster are more similar. In this paper, a dense hierarchical clustering algorithm, one of the data mining techniques, is implemented using map-reduce design and then the results of this algorithm are compared with the usual one. Experiments show that runtime decreases with increasing input data size. The runtime of the algorithm improved by 16.80% for the 200 data-point dataset and 29.26% for the dataset with 1000 data points. The percentage of CPU usage in the parallel system also increased from 22% to 94%.

Keywords: Map-reduce, Hadoop, clustering algorithms, parallel programming

Received: 22 November 2019
Revised: 2 February 2020
Accepted: 3 March 2020

Corresponding Author: Dr. Faramarz Safi-Esfahani

Citation: F. Tavakoli, F. Safi-Esfahani, "Implementation of agglomerative hierarchical clustering algorithm applying map-reduce parallel approach", Journal of Intelligent Procedures in Electrical Technology, vol. 11, no. 41, pp. 15-32, Spring 2020 (in Persian).

پیاده‌سازی الگوریتم خوشه‌بندی سلسله‌مراتبی تراکمی به‌صورت موازی با روش نگاشت-کاهش

فهیمه توکلی[۱]، دانشجوی کارشناسی‌ارشد، فرامرز صافی‌اصفهانی[۲،۱]، استادیار

۱- دانشکده مهندسی کامپیوتر- واحد نجف‌آباد، دانشگاه آزاد اسلامی، نجف‌آباد، ایران
fahimeh.tavakoli@gmail.com
۲- مرکز تحقیقات مه داده- واحد نجف‌آباد، دانشگاه آزاد اسلامی، نجف‌آباد، ایران
fsafi@iaun.ac.ir

چکیده: مدل نگاشت-کاهش یک مدل برای اجرای برنامه‌های کاربردی داده‌های بزرگ است. همچنین این مدل، یک مدل برنامه‌نویسی موازی برای نوشتن برنامه‌هایی است که می‌توانند بر روی ابر اجرا شوند. سازمان‌ها به‌طور فزاینده‌ای در حال تولید داده هستند که حاصل فرایندهای کسب‌وکار، فعالیت‌های کاربران، ردیابی وب‌سایت‌ها، حسگرها، مالی، حسابداری و غیره تولید می‌شوند. الگوریتم‌های خوشه‌بندی داده، به‌عنوان ابزاری برای تجزیه‌وتحلیل حجم زیاد داده به کار می‌روند. هدف اصلی این الگوریتم‌ها، این است که داده‌ها را در خوشه‌هایی دسته‌بندی کنند به‌طوری‌که اشیای داده در هر خوشه با یکدیگر بیشترین شباهت رادارند. در این مقاله، الگوریتم خوشه‌بندی سلسله‌مراتبی متراکم که یکی از تکنیک‌های داده‌کاوی است با استفاده از طراحی نگاشت و کاهش پیاده‌سازی شده و سپس نتایج این الگوریتم با حالت بدون نگاشت و کاهش مورد مقایسه قرار می‌گیرد. آزمایش‌های انجام‌شده نشان می‌دهد با افزایش اندازه داده‌های ورودی، زمان اجرا کاهش می‌یابد. زمان اجرای الگوریتم به روش موازی نسبت به روش ترتیبی برای مجموعه داده‌ای به‌اندازه ۲۰۰ شیء داده، ۱۶/۸ درصد و برای مجموعه داده‌ای به‌اندازه ۱۰۰۰ شیء داده، ۲۹/۲۶ درصد بهبود یافت. همچنین درصد استفاده از پردازنده کل سیستم درروش موازی از ۲۲ درصد به ۹۴ درصد ارتقاء یافت.

کلمات کلیدی: نگاشت-کاهش، هادوپ، الگوریتم‌های خوشه‌بندی داده، پردازش موازی

تاریخ ارسال مقاله: ۱۳۹۸/۹/۱
تاریخ بازنگری مقاله: ۱۳۹۸/۱۱/۱۳
تاریخ پذیرش مقاله: ۱۳۹۸/۱۲/۱۳

نام نویسنده‌ی مسئول: دکتر فرامرز صافی‌اصفهانی
نشانی نویسنده‌ی مسئول: نجف‌آباد- بلوار دانشگاه- دانشگاه آزاد اسلامی واحد نجف‌آباد- دانشکده مهندسی کامپیوتر

۱- مقدمه

بیشتر نرم‌افزارهای محاسباتی درگیر محاسبات سنگین ریاضی و یا تحلیل مقدار زیادی از داده‌ها می‌شوند. به‌عنوان‌مثال، در مسئله خوشه‌بندی داده‌ها[۱] که یکی از الگوریتم‌های داده‌کاوی[۲] است. این کار با استفاده از فقط یک پردازنده نیاز به زمان طولانی برای کامل شدن دارد. درحالی‌که در محاسبات توزیع‌شده و یا موازی یک کار بزرگ با سرعت بیشتر و زمان کمتری اجرا می‌شود. ایده اصلی، تقسیم یک مسئله بزرگ به زیر مسئله‌های کوچک‌تر است تا آنجا که زیر مسئله‌ها مستقل باشند و بتوانند توسط کارگرهای[۳] مختلف (نخ‌ها[۴] در یک هسته پردازنده، هسته‌ها در یک پردازنده چندهسته‌ای، چندین پردازنده در یک ماشین، یا چند ماشین در یک کلاستر) پردازش شوند. سپس نتایج میانی هر کارگر مجزا باهم ترکیب می‌شوند تا نتیجه نهایی را تولید کنند. الگوریتم خوشه‌بندی داده سلسله‌مراتبی تراکمی [۱]، یکی از الگوریتم‌های داده‌کاوی است که به‌صورت ترتیبی[۵] داده‌ها را دسته‌بندی می‌کند. مشکلاتی در اجرای ترتیبی این الگوریتم وجود دارد که عبارت‌اند از: زمان زیاد اجرای برنامه و حداقل شدن استفاده مفید از پردازنده و کاهش کارایی برنامه. ازاین‌رو، مسئله‌ای که وجود دارد این است که از راه‌کاری استفاده کنیم تا بتوانیم الگوریتم خوشه‌بندی داده سلسله‌مراتبی متراکم را به‌صورت موازی پیاده‌سازی کنیم تا زمان اجرای برنامه و بهره‌وری پردازنده بهبود یابند. در این تحقیق با استفاده از روش نگاشت-کاهش و به‌صورت موازی الگوریتم مذکور طوری توسعه داده‌شده است که استفاده از هسته‌های پردازنده به حداکثر رسیده و باعث کاهش زمان اجرا می‌شود. بنابراین، اگر از برنامه‌نویسی موازی مبتنی بر روش نگاشت-کاهش برای موازی‌سازی الگوریتم خوشه‌بندی داده سلسله‌مراتبی تراکمی استفاده کنیم، آنگاه زمان اجرای برنامه کاهش‌یافته و درصد استفاده از پردازنده بهتر می‌شود. آزمایش‌های انجام‌شده نشان می‌دهد، با افزایش اندازه ورودی درصد کارایی افزایش می‌یابد و کاهش بیشتری در زمان اجرا داریم. برای مثال، زمان اجرای الگوریتم به روش موازی نسبت به روش ترتیبی برای مجموعه داده‌ای به‌اندازه ۲۰۰ شیء داده، ۱۶/۸۰ درصد و برای مجموعه داده‌ای به‌اندازه ۱۰۰۰ شیء داده، ۲۹/۲۶ درصد بهبود می‌یابد. علاوه بر این، درصد استفاده از پردازنده کل سیستم درروش موازی از ۲۲ درصد به ۹۴ درصد افزایش می‌یابد.

در ادامه ساختار این مقاله به شرح ذیل است: در بخش دوم مفاهیم پایه و در بخش سوم کارهای گذشته و مرور ادبیات آورده شده است. بعدازآن، در بخش چهارم راه‌حل و شبه کد روش پیشنهادی بیان می‌شود و ارتباط مؤلفه‌های راه‌حل در قالب یک چارچوب آورده شده است. در بخش پنجم، محیط آزمایش و تنظیم‌های آن توصیف‌شده و نهایتا آزمایش‌هایی انجام‌شده شرح داده می‌شود. در بخش آخر نتیجه‌گیری و راه‌کارهای آینده بیان شده است.

۲- مفاهیم و مرور ادبیات

۲-۱- مفاهیم

۲-۱-۱- خوشه‌بندی داده

خوشه‌بندی داده‌ها یک مسئله اساسی در انواع حوزه‌های علوم کامپیوتر و زمینه‌های مرتبط با آن است. یادگیری ماشین، کاهش داده، آمار، پردازش تصویر، تشخیص الگو، شبکه و بیوانفورماتیک از خوشه‌بندی برای تحلیل داده‌ها استفاده می‌کنند. خوشه‌بندی داده‌ها تقسیم‌بندی مجموعه‌ای از داده‌ها به زیرمجموعه‌هایی از اشیا داده[۶] مشابه هم است. هر یک از این گروه‌ها یا زیرمجموعه‌ها، یک خوشه نام دارد که شامل اشیائی است که مشابه یکدیگر هستند و با اشیای سایر گروه‌ها تجانس و شباهت کمتری دارند. هدف نهایی روش‌های خوشه‌بندی داده اختصاص دادن نقاط داده به یک سیستم متناهی از خوشه‌ها است. این خوشه‌ها باهم اشتراکی ندارند و اجتماع آن‌ها هم برابر با کل مجموعه داده می‌شود [۲].

۲-۱-۲- مدل برنامه‌نویسی نگاشت-کاهش

مدل نگاشت-کاهش یک مدل برنامه‌نویسی موازی است که توسط گوگل معرفی‌شده است و هدف از طراحی آن این است که پردازش داده‌ها را به‌صورت موازی بر روی کلاسترهای عظیم ساده کند. در این مدل برنامه‌نویسی کاربر محاسبات را با دو تابع به نام‌های نگاشت و کاهش مشخص می‌کند. کتابخانه نگاشت-کاهش به‌طور خودکار محاسبات را موازی می‌کند و موضوعاتی

مانند توزیع داده، تعادل بار و تحمل خطا را کنترل می‌کند. پلتفرم هادوپ که پیاده‌سازی کد-باز از نگاشت-کاهش است به زبان جاوا نوشته‌شده و محصول شرکت آپاچی است. هدف اصلی نگاشت-کاهش موازی‌سازی محاسبات بر روی ماشین‌های کلاستر شده است. هادوپ هم همانند گوگل از ماشین‌های زیاد در یک کلاستر برای توزیع کردن پردازش داده‌ها استفاده می‌کند. این مقاله مروری بر مدل برنامه‌نویسی نگاشت-کاهش و برنامه‌های آن است. تابع نگاشت جفت (مقدار/کلید) را به‌عنوان ورودی می-گیرد و خروجی‌ها یک مجموعه‌ای از جفت‌های میانی (مقدار/کلید) است. تابع کاهش هم یک کلید را به‌عنوان ورودی می‌گیرد و یک لیست از مقادیر را به آن اختصاص می‌دهد. مقادیر ورودی برای تابع کاهش به‌طور خودکار از نتایج میانی گروه‌بندی می‌شوند. هر دو تابع نگاشت و کاهش دو پارامتر ورودی دارند یک مجموعه از جفت‌های مقدار/کلید (مجموعه داده ورودی) و یک تابع مرتبه اول تعریف‌شده توسط کاربر (تابع کاربر). توابع نگاشت و کاهش تابع کاربر را روی زیرمجموعه‌هایی از مجموعه داده ورودی به کار می‌برند. درنتیجه همه زیرمجموعه‌ها به‌طور مستقل توسط تابع تعریف‌شده توسط کاربر پردازش می‌شوند. توابع نگاشت و کاهش در اینکه چطور این زیرمجموعه‌ها را از مجموعه داده ورودی تولید می‌کنند و همچنین پاس دادن آن‌ها به تابع کاربر الحاق شده متفاوت هستند. تابع نگاشت هر جفت (مقدار/کلید) از مجموعه داده ورودی خودش را به یک زیرمجموعه اختصاص می‌دهد بنابراین همه جفت‌ها به‌طور مستقل توسط تابع کاربر پردازش می‌شوند. تابع کاهش همه جفت-های (مقدار/کلید) از مجموعه داده ورودی را به‌وسیله کلیدها گروه‌بندی می‌کند. هر گروه یک زیرمجموعه منحصربه‌فرد می‌شود که پس‌ازآن توسط تابع تعریف‌شده توسط کاربر پردازش می‌شود [۳].

۲-۱-۳- الگوریتم خوشه‌بندی سلسله‌مراتبی

خوشه‌بندی سلسله‌مراتبی[۷] یک سلسله‌مراتب یا یک درخت از خوشه‌ها ایجاد می‌کند که به گراف دندروگرام[۸] معروف بوده و هر گره خوشه شامل خوشه‌های فرزند است. انواع الگوریتم خوشه‌بندی سلسله‌مراتبی شامل الف) تراکمی[۹] (پایین به بالا)که در آن ابتدا هر نقطه داده به‌صورت یک گروه در نظر گرفته می‌شود و تا زمانی که فقط یک خوشه وجود داشته باشد دوگروهی که بیشترین تجانس و تشابه را باهم دارند ادغام می‌شوند ب) تقسیم‌کننده[۱۰] (بالا به پایین): در این الگوریتم در ابتدا همه نقاط داده به‌صورت یک گروه در نظر گرفته می‌شود و تا زمانی که هر نقطه داده در گروه مخصوص به خودش قرار بگیرد گروهی که کم‌ترین تجانس و تشابه رادارند به دو گروه تقسیم می‌شود. شبه کد (۱) الگوریتم ترتیبی سلسله‌مراتبی با روش تراکمی را نشان می‌دهد. در طول مرحله ادغام، فقط نزدیک‌ترین همسایه از هر خوشه باید بررسی شود. ازاین‌رو، تنها فاصله تا نزدیک‌ترین خوشه ذخیره می‌شود. پیچیدگی فضای کل الگوریتم خوشه‌بندی سلسله‌مراتبی تراکمی $O(n)$ است [۴].

ورودی: N تعداد اشیا داده، M تعداد قسمت‌ها، T تعداد ابعاد اشیاء داده - خروجی: خوشه‌ها

۱) مجموعه خوشه‌ها $C= \{C_1, C_2 \dots C_k\}$ مجموعه اشیاء داده $N= \{n_1, n_2 \dots n_n\}$

۲) هر نقطه داده به یک خوشه تخصیص داده می‌شود. $C \leftarrow \{ c_i=\{x_i\} \mid x_i \in D\}$

۳) محاسبه ماتریس فاصله $\Delta \leftarrow \{ d(x_i,x_j) \mid x_i \text{ و } x_j \in D \}$

۴) تکرار می‌کنیم:

۵) پیدا کردن نزدیک‌ترین جفت خوشه‌ها به‌طوری‌که $c_i, c_j \in C$

۶) خوشه‌ها را ادغام می‌کنیم $C_{ij} \leftarrow c_i \cup c_j$

۷) بروز کردن و محاسبه ماتریس فاصله Δ جدید

۸) تا زمانی که $|C| = 1$ یعنی C تنها شامل یک خوشه شود.

شبه کد (۱): الگوریتم ترتیبی خوشه‌بندی سلسله‌مراتبی با روش تراکمی

Pseudo code (1): The pseudo code of agglomerative hierarchical clustering

۲-۱-۴- روش‌های مختلف برای تعیین فاصله بین دو خوشه

متدهای مبتنی بر گراف: این متدها از گرافی از نقاط در هر خوشه برای تعیین فاصله درون خوشه استفاده می‌کنند [۵]. درروش اتصال منفرد[۱۱] فاصله بین دو خوشه حداقل فاصله بین دونقطه است به‌طوری‌که هر یک از دونقطه در یک خوشه است که در رابطه (۱) نشان داده‌شده است. درروش لینک متوسط[۱۲] فاصله بین هر دو خوشه میانگین فاصله بین هر جفت از نقاط

است به‌طوری‌که هر جفت نقاط هرکدام یک نقطه در هر خوشه دارند که در رابطه (۲) نمایش داده‌شده است. درروش لینک کامل[۱۳] فاصله بین دو خوشه حداکثر فاصله بین دونقطه است به‌طوری‌که هریکی از نقاط در یک خوشه است که در رابطه (۳) مشخص‌شده است.

$$d(S_1;S_2) = \min ||S_1 - S_2|| \quad (۱)$$

$$d(S_1;S_2) = \sum ||S_1 - S_2||2 \quad (۲)$$

$$d(S_1;S_2) = \max ||S_1 - S_2||2 \quad (۳)$$

متدهای هندسی: این متدها یک مرکز خوشه را برای هر خوشه تعریف می‌کند و از آن‌ها برای تعیین فاصله بین خوشه‌ها استفاده می‌کند [۵]. درروش مرکز جرم[۱۴] مرکز خوشه مرکز جرم همه نقاط در خوشه است و فاصله اقلیدسی[۱۵] برای فاصله بین مرکز جرم‌های خوشه می‌شود. در روش میانه[۱۶] مرکز جرم خوشه میانگین بدون وزن همه مرکزهای دو خوشه تجمیع شده برای تشکیل آن است. همچنین درروش حداکثر واریانس[۱۷] فاصله بین دو خوشه مجموع مربعات فواصل هر نقطه به مرکز خوشه است.

۲-۱-۵- تجانس لنس-ویلیامز[۱۸]

رابطه (۴) توصیف‌کننده روش خوشه‌بندی سلسله‌مراتبی تراکمی است که دربرگیرنده روش‌های منفرد[۱۹]، کامل[۲۰]، متوسط[۲۱]، مرکزی[۲۲]، میانه[۲۳]، یک‌جانبه[۲۴] و متوسط وزنی[۲۵] است. در این تحقیق برای ایجاد ماتریس فاصله در زمان پیاده‌سازی الگوریتم خوشه‌بندی سلسله‌مراتبی متراکم روش لینک منفرد[۲۶] بکار برده شده است که کمترین فاصله بین اشیای داده را در نظر می‌گیرد [۶,۷]. جدول (۱) مقدار پارامترها در روش‌های مختلف را نشان می‌دهد که درنهایت فرمول برای متد لینک منفرد در رابطه (۵) مشخص‌شده است. پارامترهای c_i, c_j, c_k تعداد تکرارها در خوشه را نشان می‌دهند.

$$d(I \cup j, k) = a_i * d[I, k] + a_j * d[j, k] + b * d[I, j] + g * d[I, k] - d[j, k] \quad (۴)$$

$$\text{Distance}(i \cup j, k) = \min\{d(i,k), d(j,k)\} \quad (۵)$$

Table (1): Parameters G, b, a_j and A_i for different methods

جدول (۱): پارامترهای A_i،a_i،b،G برای متدهای مختلف

روش	A_i	a_j	b	G
منفرد	0.5	0.5	0	-0.5
کامل	0.5	0.5	0	0.5
متوسط	ci/(ci+cj)	cj/(ci+cj)	0	0
مرکزی	ci/(ci+cj)	cj/(ci+cj)	-	-
میانه	0.5	0.5	-0.25	0
یک‌جانبه	(ci+ck)/(ci+cj+ck)	(cj+ck)/(ci+cj+ck)	-ck/(ci+cj+ck)	0
متوسط وزنی	0.5	0.5	0	0

۲-۱-۶- شاخص و فاصله جاکارد

در این الگوریتم از معیار شاخص جاکارد[۲۷] [۸] که به ضریب تشابه جاکارد[۲۸] نیز معروف است استفاده می‌شود. ضریب جاکارد برای مقایسه تشابه و تجانس مجموعه داده‌های محدود بکار می‌رود. درواقع این ضریب یک نوع اندازه‌گیری اطلاعات نامتقارن[۲۹] بر روی متغیرهای دودویی[۳۰] و غیر دودویی است. ضریب تشابه جاکارد در دو نمونه داده محدود A و B و با متغیرهای غیر دودویی به‌صورت اشتراک این دو مجموعه تقسیم‌بر اجتماع آن دو به دست می‌آید که در رابطه (۶) نشان داده‌شده است. اگر دو مجموعه A و Bخالی باشند آنگاه J(A,B)=1 می‌شود. همچنین رابطه (۷) همواره برقرار است. همچنین فاصله جاکارد[۳۱]، عدم

تجانس[۳۲] بین دو مجموعه داده را اندازه‌گیری می‌کند. درواقع این معیار مکمل ضریب جاکارد است و با کم کردن شاخص جاکارد از عدد یک به دست می‌آید که در رابطه (۸) نشان داده شده است.

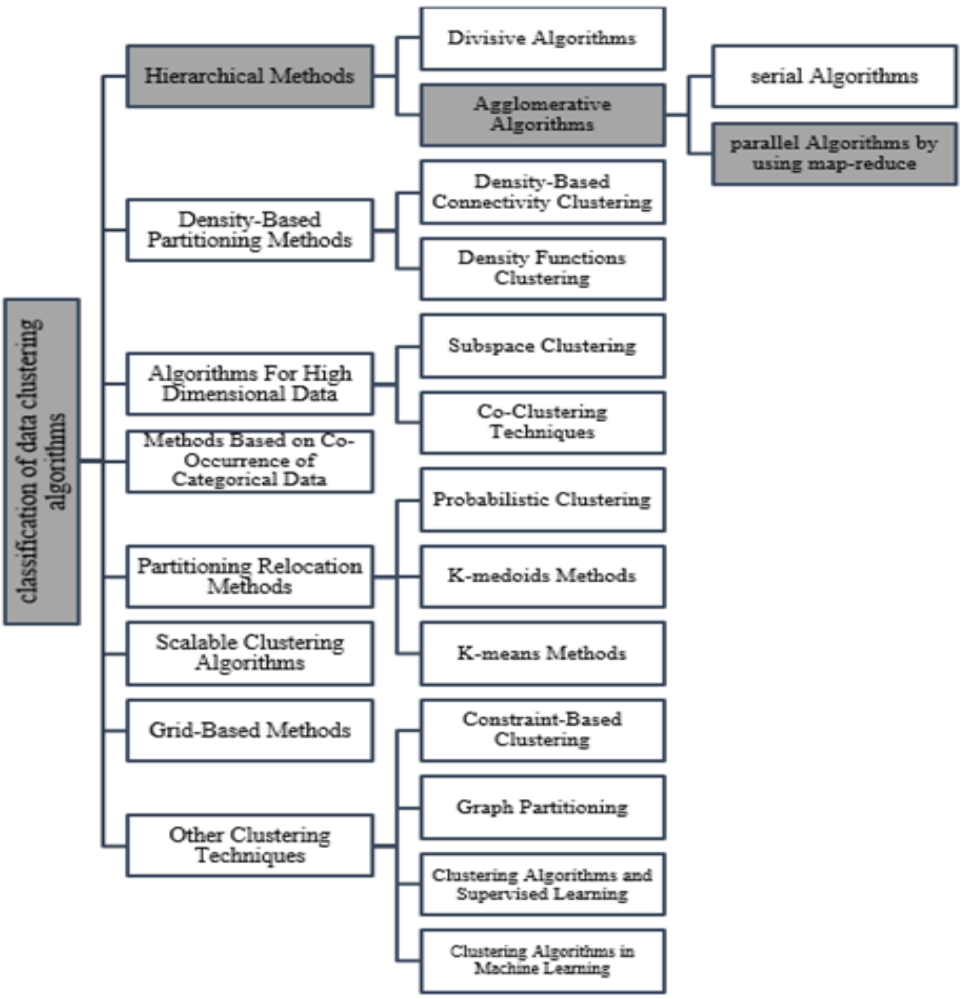
$$J(A,B)=\frac{|A\cap B|}{|A\cup B|} \quad (۶)$$

$$0\le J(A,B)\le 1 \quad (۷)$$

$$d_j(A,B)=1-J(A,B)=\frac{|A\cup B|-|A\cap B|}{|A\cup B|} \quad (۸)$$

۲-۲- مرور ادبیات

الگوریتم‌های خوشه‌بندی داده در شکل (۱) نشان داده‌شده است که الگوریتم‌های خوشه‌بندی داده را به چندین دسته تقسیم می‌کند. یکی از این دسته‌ها، روش‌های سلسله‌مراتبی است که در سطح اول شکل بارنگ تیره مشخص‌شده است. این روش، نیز به‌نوبه خود به دودسته تقسیم می‌شود: الگوریتم‌های تقسیم‌کننده و الگوریتم‌های متراکم. الگوریتم‌های متراکم در سطح دوم نمودار بارنگ تیره قابل‌شناسایی است. الگوریتم‌های سلسله‌مراتبی متراکم هم به روش ترتیبی و هم به روش موازی قابل پیاده-سازی هستند. در این تحقیق نیز تمرکز ما بر روی موازی‌سازی الگوریتم سلسله‌مراتبی متراکم با روش نگاشت-کاهش است. در ادامه به توضیح تحقیقات انجام‌شده توسط این الگوریتم خواهیم پرداخت.



شکل (۱): تقسیم‌بندی روش‌های خوشه‌بندی داده از دیدگاه داده‌کاوی

Figure (1): Taxonomy of clustering methods in data mining

تاکنون تحقیقات متعددی در حوزه خوشه‌بندی به همراه به‌کارگیری روش نگاشت-کاهش صورت گرفته است [۹،۱۰] یا به توزیع کردن پردازش این الگوریتم‌ها پرداخته‌اند [۱۱]. از بین این تحقیقات برخی روی استفاده از روش‌های فرا اکتشافی در موازی‌سازی خوشه‌بندی با استفاده از روش نگاشت-کاهش تمرکز کرده‌اند [۱۵-۱۲]. همچنین برخی به استفاده از نگاشت-کاهش در روش‌های داده‌کاوی پرداخته‌اند [۱۶]، خوشه‌بندی اسناد [۱۷] و یا از این روش روی داده‌های بزرگ استفاده‌شده است [۹،۱۸] و یا هم به داده‌های بزرگ و هم مسائل درون خوشه‌ای پرداخته‌اند [۱۹]. به‌کارگیری نگاشت-کاهش در خوشه-بندی داده‌های جریانی هم کاربرد داشته است [۲۰]. در تحقیق [۲۱] یک روش خوشه‌بندی موازی برای الگوریتم میانگین k (k-means) بنام PKMeans معرفی شده است. جایی که داده‌ها بین چندین پردازنده تقسیم می‌شوند که می‌تواند مراکز خوشه[۳۳] به‌دست‌آمده در تکرارهای قبلی را در نظر گرفته و نمونه‌های داده را به خوشه مربوطه تخصیص می‌دهند. درواقع در این الگوریتم تابع نگاشت، فرآیند تخصیص هر نمونه داده را به نزدیک‌ترین مرکز خوشه اجرا می‌کند. درحالی‌که تابع کاهش، فرآیند بروز کردن مراکز خوشه جدید را انجام می‌دهد. در [۱۵] مورد مطالعاتی از خوشه‌بندی داده‌های فیلم نت‌فلیکس[۳۴] با استفاده از سه الگوریتم خوشه‌بندی در چارچوب نرم‌افزاری نگاشت-کاهش فراهم‌شده است. نویسندگان در [۲۲]، از خوشه-بندی سلسله‌مراتبی برای گروه‌بندی کاربران اینترنت با کاهش دادن حجم عظیمی از صفحات سیاهه[۳۵] تا صد گیگابایت را استفاده می‌کنند. بااین‌حال، با اجرای الگوریتم موازی مشکلاتی از قبیل حافظه ناکافی و هم‌زمان اجرای طولانی آزاردهنده می-باشند. مقاله آن‌ها یک روش خوشه‌بندی سلسله‌مراتبی کارا برای کاهش مجموعه داده‌های عظیم با استفاده از مدل نگاشت-کاهش نشان می‌دهد که از دو روش بهینه‌سازی استفاده می‌کند: به‌روزرسانی دسته‌ای[۳۶] برای کاهش زمان محاسبات و هزینه-های ارتباطی در بین نودهای کلاستر ۲) روش مبتنی بر هم رخدادی[۳۷] برای کاهش بعد بردارهای ویژگی و حذف ویژگی‌های نویز[۳۸]. در این مطالعه تجربی نشان می‌دهد که روش اول به‌طور قابل‌توجهی می‌تواند باعث کاهش سربار ارتباطات و ورودی/خروجی شود و همچنین باعث کاهش زمان اجرای کل به حدود ۱/۱۵ شود. روش دوم به‌طور مؤثر ویژگی‌ها را ساده کرده و نیز دقت خوشه‌بندی سلسله‌مراتبی بهبودیافته است. پژوهشگران در [۲۳] الگوریتم خوشه‌بندی توزیع‌شده را پیاده‌سازی کردند که DisCo نام دارد و برای برنامه‌هایی در حوزه متن‌کاوی[۳۹]، فیلترینگ هم کارانه[۴۰]، بیوانفورماتیک و گراف کاوی[۴۱] کاربرد دارد. مدل برنامه‌نویسی نگاشت-کاهش در این الگوریتم برای پیش‌پردازش داده‌های توزیع‌شده و خوشه‌بندی آن‌ها استفاده می‌شود. خوشه‌بندی سلسله‌مراتبی به‌طور وسیع در انجمن‌های داده‌کاوی موازی و توزیع‌شده موردمطالعه قرارگرفته است. نویسندگان در [۲۴] یک روشی بنام DisCo که یک پیاده‌سازی موازی از الگوریتم خوشه‌بندی سلسله‌مراتبی ارائه می-کنند که بر اساس پیوند تنها[۴۲] حداقل واریانس بر روی پردازنده‌های SIMD[۴۳] را فراهم کرده‌اند. اجرای غیر موازی این الگوریتم پیچیدگی زمانی $O(n^2)$ ایجاد می‌کند. در [۲۵] الگوریتم‌های موازی برای خوشه‌بندی سلسله‌مراتبی را با روش‌های منفرد، متوسط‌گیری، میانه‌گیری، کامل و پیوند مرکز[۴۴] انجام می‌دهد درحالی‌که در تحقیق فعلی از روش منفردگرا برای اجرای الگوریتم موازی خوشه‌بندی سلسله‌مراتبی استفاده می‌شود. پژوهشگران در [۲۶] یک الگوریتم قطعی به نام EREW برای خوشه‌بندی سلسله‌مراتبی ارائه دادند که مبتنی بر گراف کامل و الگوریتم‌های درخت پوشا[۴۵] با حداقل فاصله اقلیدسی است. این الگوریتم می‌تواند n تا شیء را بر روی O(n) پردازنده و در زمان O(n2/p) کلاستر بندی کند. مؤلفین در [۲۷] الگوریتم خوشه‌بندی سلسله‌مراتبی را روی گذرگاه‌های نوری[۴۶] توسعه داده و الگوریتم‌هایی ارائه می‌دهند که در بدترین حالت کارایی را تضمین می‌کند. در این الگوریتم از روش منفردگرا که کمترین فاصله بین خوشه‌ها را محاسبه می‌کند استفاده می‌شود و معیار فاصله اقلیدسی را برای اندازه‌گیری فاصله به کار می‌برد. اما درروش الگوریتم موازی ارائه‌شده در تحقیق جاری، معیاری که برای اندازه‌گیری میزان تشابه بین خوشه‌ها استفاده می‌شود ضریب جاکارد است. چارچوب بنام pPOP در [۲۸] ارائه‌شده که نسخه موازی روش POP[۴۷] است و روی معماری چندپردازنده با حافظه اشتراکی پیاده‌سازی شده است. نتایج آزمایش‌ها نشان می‌دهد که pPOP نزدیک به‌سرعت خطی عمل می‌کند و بهتر از الگوریتم‌های موازی موجود هم در زمان پردازنده و هم‌مقدار حافظه موردنیاز است. در [۲۹] نویسندگان روش خوشه‌بندی BIRCH را برای پایگاه داده‌های خیلی بزرگ توسعه داده‌اند. این روش ویژگی خوشه‌بندی[۴۸] را تعریف می‌کند که اطلاعات را در یک کلاستر خلاصه می‌کند. مجموعه داده با یک درخت +B به نام درخت CF نمایش داده می‌شوند که از واردکردن نقاط داده جدید پشتیبانی می‌کند. روش خوشه‌بندی پیشنهادی در این مقاله

مبتنی بر فاصله است و فرض بر آن است که اشیای داده از قبل ایجادشده است. نویسندگان در [۳۰] الگوریتم خوشه‌بندی ROCK را توسعه دادند جایی که تابع تجانس sim (pi; pj) نزدیکی بین نقاط داده و همسایه‌ها را اندازه‌گیری می‌کند که این تابع به‌صورت یک جفت نقطه تعریف‌شده که تجانس بین دونقطه بزرگ‌تر از حد یک آستانه است. در رابطه $sim(pi; pj) \geq \Theta$ اگر دونقطه اشتراک با تعداد زیادی از همسایه‌ها داشته باشند آن‌ها درون یک کلاستر گذاشته می‌شوند. مقاله تحقیقی [۳۱] الگوریتم خوشه‌بندی CHAMELEON را توسعه داده که تشابه دو کلاستر را بر اساس یک مدل پویا اندازه‌گیری می‌کند که باعث می‌شود کشف کلاسترهای همگن ساده‌تر شود. محققین در [۳۲] یک الگوریتم خوشه‌بندی سلسله‌مراتبی را بر روی داده های ناهمگن توزیع‌شده توسعه می‌دهد. نویسندگان در [۳۳] یک روش بنام RACHET برای ادغام خوشه‌بندی سلسله‌مراتبی از مجموعه داده‌هایی که به‌صورت یکنواخت توزیع شده‌اند را توسعه می‌دهند. هر سایت یک گراف دندروگرام بر اساس داده‌های محلی تولید می‌کند و سپس آن را به یک سایت محلی انتقال می‌دهد. برای کاهش هزینه‌های ارتباطی، شرح کامل از یک کلاستر را در یک دندروگرام نمی‌فرستند ولی در عوض یک تقریب از هر کلاستر فرستاده می‌شود که متشکل از آمارهای مختلف توصیفی است. برای مثال تعداد نقاط داده در هر کلاستر، به‌طور میانگین فاصله اقلیدسی از هر نقطه داده تا مرکز جرم باشد. محققین در [۳۴] یک الگوریتم برای ترکیب چارچوب نگاشت-کاهش و روش مقداردهی اولیه سلول عصبی از الگوریتم VPSOM[۴۹] برای خوشه‌بندی متن استفاده کرده است. در این روش، محققین از روش نگاشت-کاهش برای پردازش مجموعه داده‌های خود به‌صورت موازی استفاده می‌کنند.

مقایسه بین مرتبط‌ترین کارهای گذشته در مقایسه با الگوریتم‌های خوشه‌بندی سلسله‌مراتبی موازی در جدول (۲) آورده شده است و از دیدگاه‌های مختلف نظیر محیط عملیات، معیار تشابه، ساختمان داده مورداستفاده، داده ورودی و روش پیاده‌سازی مورد مقایسه قرار گرفته‌اند. روش پیشنهادی این پژوهش نیز در سطر آخر آورده شده است که با یک سیستم چندپردازنده با حافظه اشتراکی پیاده‌سازی شده است و از معیار فاصله جاکارد برای اندازه‌گیری میزان تشابه اشیای داده (از نوع متن و رشته‌ای) و از روش نگاشت-کاهش برای خوشه‌بندی سلسله مراتبی تراکمی اشیای داده استفاده شده است و درنهایت، داده‌ها در قالب یک گراف دندروگرام شکل گرفته است.

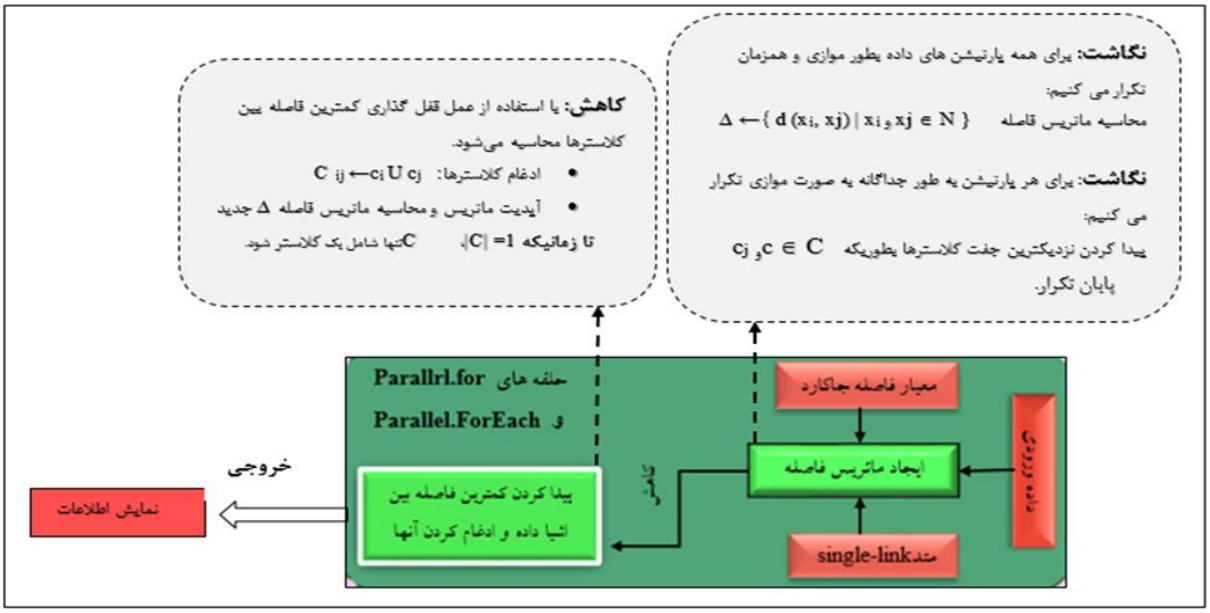
Table (2): The comparison of distributed and parallel clustering methods

جدول (۲): مقایسه بین الگوریتم‌های خوشه‌بندی سلسله‌مراتبی توزیع‌شده و موازی

روش/ویژگی	محیط عملیاتی	معیار تشابه	ساختمان داده	داده ورودی
[۲۴]	پردازنده‌های SIMD	حداقل واریانس	درخت پوشا	نقاط داده در فضا
[۲۵]	محیط توزیع‌شده	حداقل واریانس	-	نقاط داده در فضا
[۲۶]	محیط توزیع‌شده	فاصله اقلیدسی	گراف کامل	اشیا داده
[۲۸]	چندپردازنده با حافظه اشتراکی	فاصله اقلیدسی	گراف دندروگرام	نقاط داده در فضا
[۲۹]	محیط توزیع‌شده	-	درخت B+	نقاط داده در فضا
[۳۵]	محیط توزیع‌شده	تابع تعریف‌شده $sim(pi; pj) \geq \Theta$	-	نقاط داده در فضا
[۳۰]	محیط توزیع‌شده	اندازه‌گیری بر اساس مدل پویا	گراف دندروگرام	داده‌های همگن
[۳۱]	محیط توزیع‌شده	فاصله اقلیدسی	–	داده‌های ناهمگن
[۳۲]	محیط توزیع‌شده	فاصله اقلیدسی	گراف دندروگرام	نقاط داده در فضا
[۳۳]	محیط توزیع‌شده	-	-	خوشه‌بندی متن
روش پیشنهادی	چندپردازنده با حافظه اشتراکی	فاصله جاکارد	گراف دندروگرام	اشیا داده از نوع متن و رشته‌ای

۳- راه‌حل پیشنهادی

شکل (۲) چارچوب خوشه‌بندی داده سلسله مراتبی تراکمی به روش موازی و با استفاده از مدل نگاشت-کاهش را نمایش می-دهد که شامل دو بخش نگاشت و کاهش است. در بخش اول داده‌ها به‌طور تصادفی به چندین بخش تقسیم می‌شوند. سپس بر روی هر قسمت عمل نگاشت صورت می‌گیرد و داده‌های هر قسمت به‌طور موازی به هسته‌های پردازنده نگاشت می‌شود. پس از به دست آوردن کمترین فاصله بین اشیا در هر قسمت الگوریتم وارد بخش دوم می‌شود و عمل کاهش با پیدا کردن کوچک‌ترین فاصله بین اشیا داده صورت می‌گیرد. شبه کد (۲) مربوط به الگوریتم موازی پیشنهاد شده برای خوشه‌بندی سلسله مراتبی تراکمی است. فرض شده که N تا شیء داده داریم که هرکدام از این اشیا دارای مختصات T بعدی هستند به این معنا که هر شیء داده T تا ویژگی دارد. چون تعداد اشیا داده N تا است و M هم تعداد قسمت‌ها است پس تعداد اشیا داده تخصیص داده شده برای هر قسمت N/M می‌شود. اگر N/M مساوی صفر نباشد قسمت آخر مابقی اشیا داده را می‌گیرد.



شکل (۲): چارچوب ارائه‌شده برای پیاده‌سازی الگوریتم خوشه‌بندی سلسله‌مراتبی متراکم بر اساس روش نگاشت-کاهش

Figure (2): The presented framework to map-reduce implementation of agglomerative hierarchical clustering

ورودی: N تعداد اشیا داده، M تعداد قسمت‌ها، T تعداد ابعاد اشیاء داده - خروجی: خوشه‌ها

مجموعه خوشه‌ها $C= \{C_1, C_2 \dots C_k\}$ مجموعه اشیاء داده $N= \{n_1, n_2 \dots n_n\}$

۱) داده‌های ورودی به M قسمت تقسیم می‌شوند.

۲) محاسبه تعداد اشیا داده در هر قسمت به‌طوری‌که S_m تعداد اشیاء داده در هر قسمت باشد.

۳) اگر $N \bmod M \neq 0$ شد باقیمانده اشیا داده در نظر گرفته شود.

۴) نگاشت: برای هر قسمت داده به‌طور موازی تکرار می‌کنیم:

محاسبه ماتریس فاصله $\Delta \leftarrow d(x_i, x_j) \mid x_i$ و $x_j \in N\}$

۵) نگاشت: به‌طور موازی تکرار می‌کنیم:

۶) پیدا کردن نزدیک‌ترین جفت خوشه‌ها به‌طوری‌که $c_i, c_j \in C$

۷) پایان تکرار

۸) کاهش: با استفاده از قفل‌گذاری داده‌ها کمترین فاصله بین خوشه‌ها محاسبه می‌شود.

۹) ادغام خوشه‌ها $C_{ij} \leftarrow c_i U c_j$

۱۰) آپدیت کردن و محاسبه ماتریس فاصله Δ جدید تا زمانی که $|C| =1$، C تنها شامل یک خوشه شود.

شبه کد (۲): الگوریتم موازی خوشه‌بندی سلسله‌مراتبی تراکمی

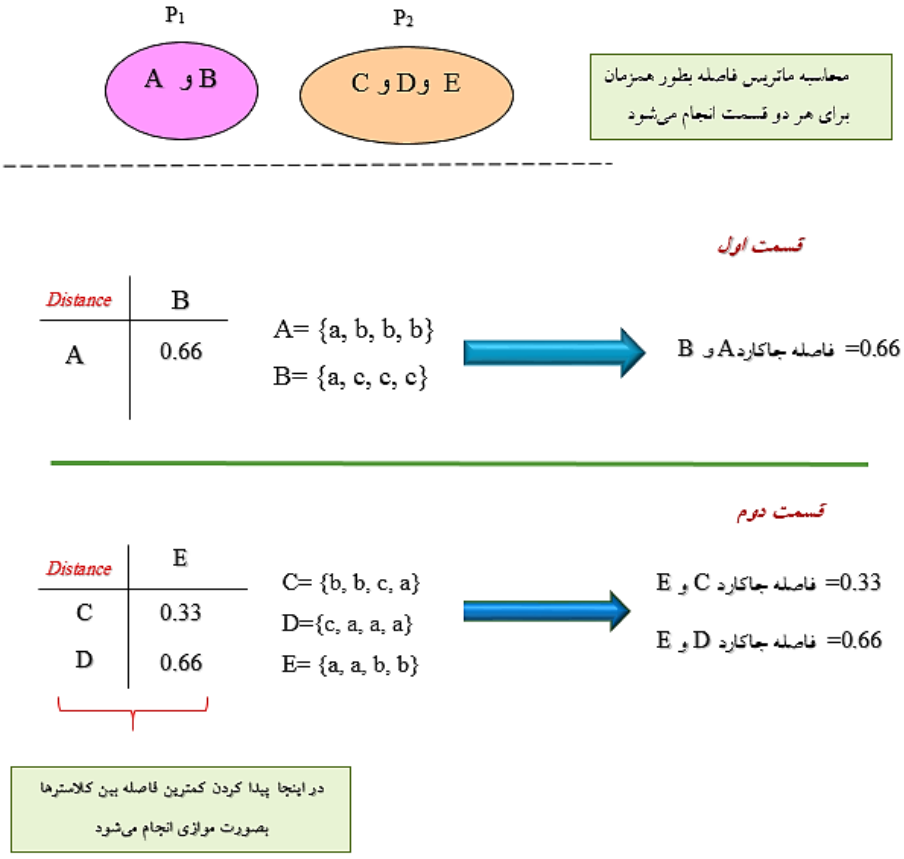
Figure (2): The pseudo code of agglomerative hierarchical clustering method

با استفاده از عمل نگاشت، تمام قسمت‌ها به‌طور موازی ماتریس فاصله را محاسبه می‌کنند. همچنین برای محاسبه کمترین فاصله بین دو کلاستر درون هر قسمت، مجدداً عمل نگاشت صورت می‌گیرد و محاسبات به‌صورت موازی انجام می‌شود. با توجه به ماتریس‌های فاصله که در هر قسمت به‌طور جداگانه محاسبه می‌شود و کوچک‌ترین فاصله بین دو کلاستر به دست می‌آید با استفاده از عمل کاهش و با به کار بردن عمل قفل‌گذاری، کمترین فاصله بین خوشه‌ها محاسبه‌شده و دو کلاستر با کمترین فاصله باهم ادغام می‌شوند و ماتریس فاصله جدید بروز می‌شود و درنهایت این الگوریتم با به دست آمدن تنها یک خوشه خاتمه می‌یابد. به‌عنوان یک مطالعه موردی فرض کنیم پنج شئ داده N={A, B, C, D, E} مطابق جدول (۳) داشته باشیم. هر شئ داده دارای مختصات چهاربعدی است (T=4) و همچنین تعداد قسمت‌ها دو باشد (M=2). پس اشیاء داده به‌صورت زیر قابل‌تقسیم می‌باشند: P1={A, B} و P2={C, D, E}. با توجه به انجام مراحل الگوریتم در شکل (۳)، کمترین فاصله بین دو شیء داده ۳۳/. است که فاصله بین C و E است. بنابراین، این دو شیء داده باهم ادغام‌شده و تشکیل یک کلاستر را می‌دهند. سپس مجموعه داده‌ها بروز می‌شوند و ماتریس فاصله جدید دوباره محاسبه می‌شود. این عمل تا آنجا ادامه دارد که تمام مجموعه داده‌ها تشکیل یک کلاستر را دهند.

Table (3): Data points applied in the case study

جدول (۳): اشیای داده مورداستفاده در مطالعه موردی

اشیا داده	مختصات ۱	مختصات ۲	مختصات ۳	مختصات ۴
A	a	b	b	b
B	a	c	c	c
C	b	b	c	a
D	c	a	a	a
E	a	a	b	b

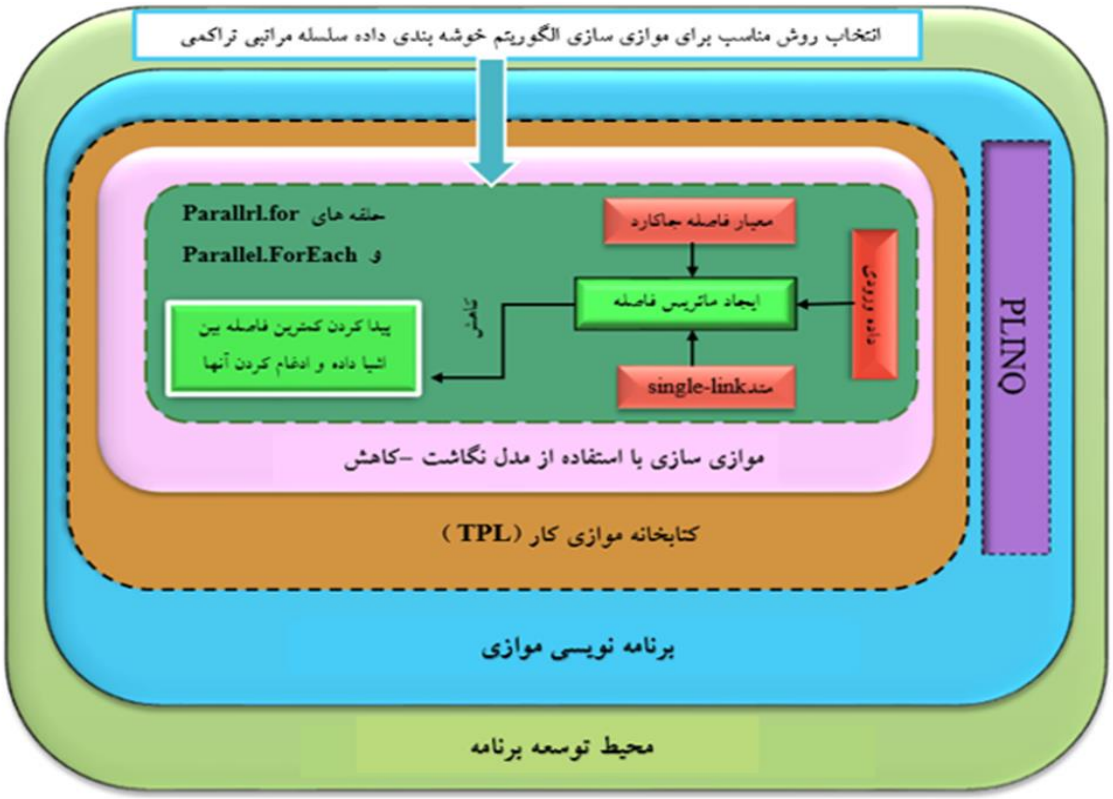


شکل (۳): مراحل طی شده در مطالعه موردی

Figure (3): The steps of the case study

۴- محیط آزمایش، تنظیمات، مجموعه داده‌ها و آزمایش‌ها

طبق شکل (۴) محیط توسعه برنامه، چارچوب نرم‌افزاری پیاده‌سازی شده بر اساس امکانات دات نت (.NET) است که توسط شرکت مایکروسافت توسعه داده‌شده و بر روی سیستم عامل‌های ویندوز قابل اجرا است و شامل کتابخانه‌ای از کلاس‌های مورد نیاز برای نوشتن برنامه‌ها است. برای نوشتن برنامه به‌صورت موازی، از کتابخانه موازی کار (TPL) که در نسخه ۴,۰ این چارچوب وجود دارد استفاده شده است. همچنین برای موازی‌سازی، حلقه‌های Parallel.For و Parallel.ForEach را به کار گرفته‌شده است. در پایان، از ابزار ویژولایزر هم‌زمانی۵۰ برای ارزیابی نتایج آزمایش‌ها و نظارت بر پارامترهای مورد سنجش و تجزیه و تحلیل آن‌ها استفاده شده است. آزمایش‌های این تحقیق بر روی یک ماشین مجازی چندپردازنده و با حافظه اشتراکی صورت گرفته است در جدول (۴) مشخصات ماشین آورده شده است.



شکل (۴): مراحل توسعه الگوریتم ترتیبی خوشه‌بندی سلسله‌مراتبی تراکمی به‌صورت موازی

Figure (4): Developing steps of the parallel agglomerative hierarchical clustering algorithm

Table (4): The particulars of the applied machine

جدول (۴): مشخصات ماشین مورداستفاده در آزمایش‌ها

مدل پردازنده	Intel core i7-4702MQ 2.20 GHz
تعداد هسته‌های فیزیکی	۴ عدد
تعداد نخ‌ها یا هسته‌های منطقی	۸ عدد
مقدار حافظه اصلی	۸ گیگابایت
سیستم‌عامل	ویندوز ۸

۴-۱ معرفی مجموعه‌های داده

در این تحقیق از دو نوع مجموعه داده برای آزمایش‌ها استفاده شده است. مجموعه داده اول، داده‌هایی از نوع عددی هستند و بنابراین معیار تشابه برای خوشه‌بندی داده‌ها استفاده از فاصله اقلیدسی بوده که مبتنی بر موقعیت نقاط داده‌ها در فضا است. همچنین این داده‌ها در رقابت‌های داده‌کاوی KDD نیز مورداستفاده قرارگرفته است. این مجموعه داده شامل ۵۰۰۰ نقطه داده با تعداد ابعاد ۷۸ است. درحالی‌که مجموعه داده دوم داده‌هایی هستند که درون خود برنامه به‌صورت تصادفی تولید می‌شوند. این داده‌ها از نوع غیر عددی و از نوع رشته یا متن می‌باشند که درون خود برنامه به‌صورت تصادفی تولید می‌شوند و معیار خوشه‌بندی آن‌ها استفاده از ضریب جاکارد مبتنی بر ویژگی‌های اشیاء داده است نه موقعیت آن‌ها در فضا.

۴-۲ آزمایش‌ها

همان‌طور که در فصل‌های قبل گفته شد هدف این تحقیق موازی‌سازی الگوریتم خوشه‌بندی سلسله مراتبی به روش تراکمی و همچنین اندازه‌گیری کارایی این الگوریتم به روش‌های ترتیبی و موازی با مقایسه بین زمان‌های اجرا آن‌ها است. در این تحقیق برای ارزیابی کارایی الگوریتم خوشه‌بندی سلسله‌مراتبی به روش تراکمی چهار آزمایش بر روی مجموعه داده‌های مختلف صورت گرفته است. در این تحقیق هر آزمایش یک‌بار به روش ترتیبی و یک‌بار به روش پیشنهادی انجام شده است. در مجموعه آزمایش اول، زمان اجرای الگوریتم به‌صورت ترتیبی و موازی مقایسه شده است. در آزمایش دوم، درصد به‌کارگیری پردازنده بر اساس هسته‌های پردازنده به روش‌های ترتیبی و موازی مورد تجزیه‌وتحلیل قرارگرفته است. در آزمایش سوم، وضعیت نخ‌های برنامه در اجرای ترتیبی و اجرای موازی ازنظر تعداد هسته‌های پردازنده که در اختیاردارند بررسی‌شده است و در آزمایش چهارم درصد استفاده از پردازنده کل سیستم درروش ترتیبی و موازی مورد بررسی قرار گرفته است.

۴-۲-۱- آزمایش اول: مقایسه زمان اجرا درروش ترتیبی و موازی

مطابق جدول (۵) در آزمایش اول، در ابتدا زمان اجرای برنامه به روش ترتیبی و با مجموعه داده‌های غیر عددی (متنی) که در سایزهای مختلف هستند اندازه گرفته‌شده است. در ادامه برای ارزیابی روش پیشنهادی، طبق جدول (۶) زمان اجرای این برنامه به روش موازی و با مجموعه داده‌های غیر عددی (متنی) که در سایزهای مختلف هستند اندازه گرفته شده است. با بررسی وضعیت اجرای الگوریتم به‌وسیله این دو روش(ترتیبی و موازی) می‌توان نتیجه گرفت که هر چه اندازه داده ورودی بیشتر باشد اجرای الگوریتم به روش موازی بهینه‌تر است. در جدول (۷)، زمان اجرای الگوریتم برای هر دو روش پیشین (ترتیبی) و روش پیشنهادی (موازی) آورده شده و همچنین میزان بهبود الگوریتم اندازه‌گیری شده است.

Table (5): The execution time of the sequential method

جدول (۵): زمان اجرای الگوریتم به روش ترتیبی

اندازه مجموعه داده	زمان اجرا
۱۰۰	۲/۸۲ میلی‌ثانیه
۲۰۰	۱۰/۴ ثانیه
۵۰۰	۸۲/۵ ثانیه
۱۰۰۰	۶/۶۳ دقیقه

Table (6): The execution time of the parallel method

جدول (۶): زمان اجرای الگوریتم به روش موازی

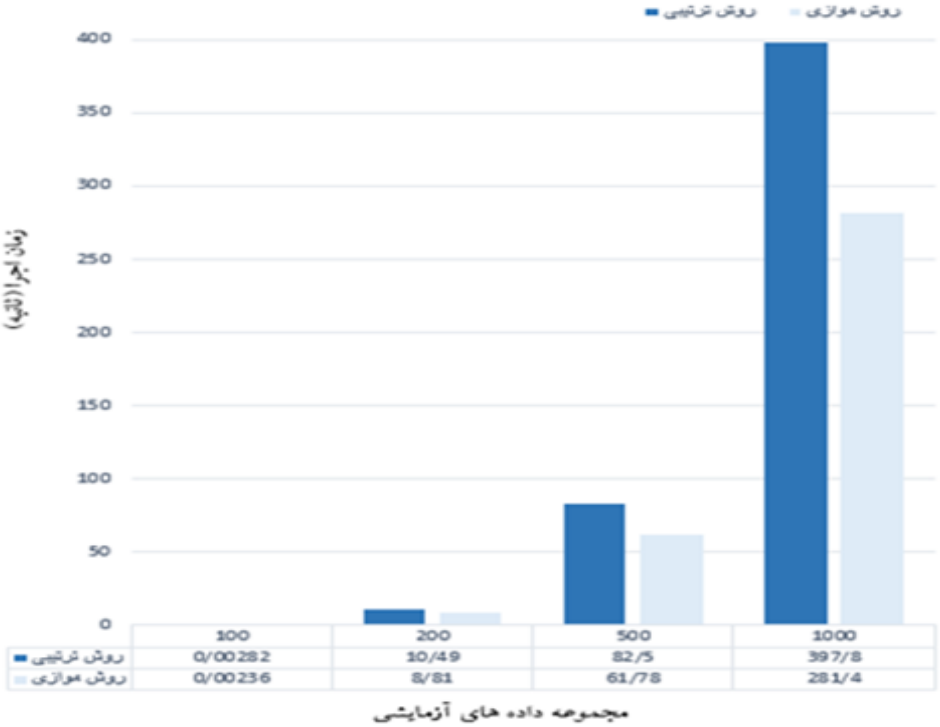
اندازه مجموعه داده	زمان اجرا
۱۰۰	۲/۳۶ میلی‌ثانیه
۲۰۰	۸/۸۱ ثانیه

۵۰۰	۶۱/۷۸ ثانیه
۱۰۰۰	۴/۶۹ دقیقه

Table (7): The comparison of sequential and parallel methods

جدول (۷): مقایسه کارایی بین روش‌های ترتیبی و موازی

مجموعه داده	زمان اجرا درروش ترتیبی	زمان اجرا به روش موازی	میزان کاهش زمان اجرا
۱۰۰	۲/۸۲ میلی‌ثانیه	۲/۳۶ میلی‌ثانیه	۱۶/۳۱درصد
۲۰۰	۱۰/۴۹ ثانیه	۸/۸۱ ثانیه	۱۶/۸۰درصد
۵۰۰	۸۲/۵ ثانیه	۶۱/۷۸ ثانیه	۲۵/۱۱درصد
۱۰۰۰	۶/۶۳ دقیقه	۴/۶۹ دقیقه	۲۹/۲۶درصد



شکل (۵): مقایسه زمان اجرای روش‌های ترتیبی و موازی

Figure (5): The execution time comparison of sequential and parallel methods

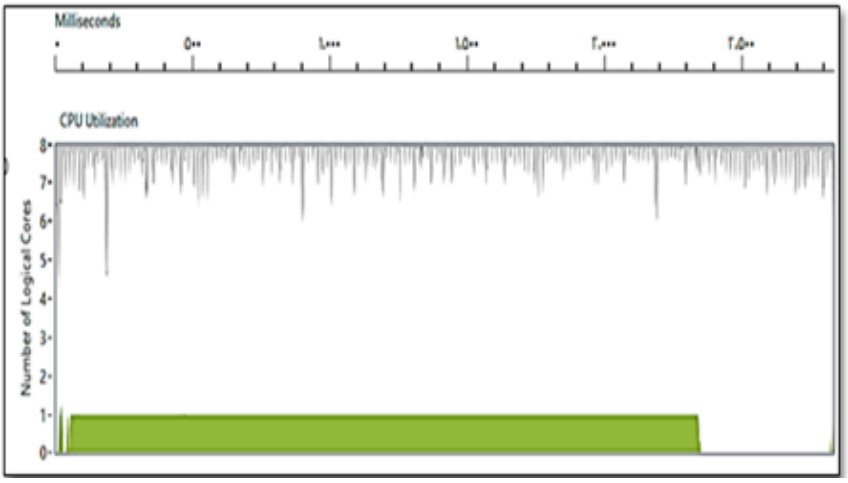
درواقع برای مجموعه داده‌های بزرگ‌تر موازی‌سازی بیشتری صورت می‌گیرد. با افزایش اندازه داده ورودی، اجرای الگوریتم به روش موازی به‌صرفه‌تر است و میزان کاهش زمان اجرا محسوس‌تر است. همچنین در شکل (۵) میزان بهبود الگوریتم (میزان کاهش زمان اجرا) روش پیشنهادی(موازی) نسبت به الگوریتم روش پیشین (ترتیبی) نمایش داده شده است.

۴-۲-۲- آزمایش دوم: مقایسه درصد به‌کارگیری پردازنده بر اساس هسته‌های پردازنده درروش ترتیبی و موازی

در این آزمایش درواقع میانگین استفاده از هسته‌های پردازنده توسط ۱) فرآیندهای اصلی ۲) فرآیندهای سیستمی[۵۱] ۳) فرآیندهای بیکار[۵۲] و ۴) فرآیندهای دیگر که در طول زمان آزمایش در حال اجرا بر روی سیستم هستند و مشخص نیست که کدام هسته در هر زمان فعال است. این تصویر با تقسیم کردن زمان پروفایل کار به بخش‌های کوتاه‌مدت ایجاد می‌شود. برای هر بخش، نمودار گرافیکی میانگین تعداد نخ‌های فرآیند که در حال اجرا بر روی هسته‌های منطقی در آن فاصله زمانی هستند نشان داده می‌شود. تعداد هسته‌های منطقی بر روی محور عمودی و زمان اجرای برنامه بر روی محور افقی نشان داده‌شده است. شکل (۶) اجرای الگوریتم ترتیبی را در مجموعه داده صدتایی و در زمان ۲,۸۲ میلی‌ثانیه نشان می‌دهد و شکل (۷)

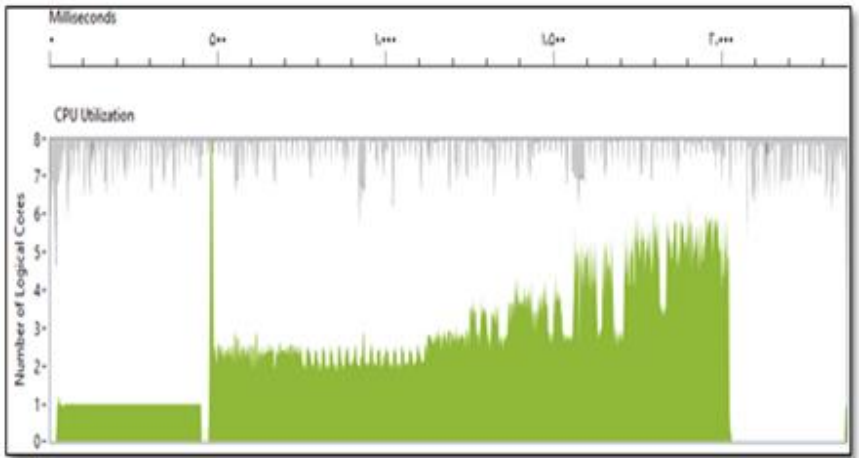
اجرای الگوریتم موازی در مجموعه داده صدتایی و در زمان ۲,۳۶ میلی‌ثانیه را نشان می‌دهد. درروش ترتیبی برنامه تنها از یک هسته پردازنده برای اجرای برنامه استفاده می‌کند و مابقی پردازنده برای پردازش فرآیندهای سیستمی و فرآیندهای بیکار و دیگر فرآیندها تخصیص داده می‌شود. ولی اجرای این برنامه به روش موازی از تعداد بیشتر هسته‌های پردازنده برای اجرای کار استفاده می‌کند. درواقع این تصاویر درجه هم‌زمانی[۵۳] را در برنامه و در طول زمان نشان می‌دهد که از آن‌ها می‌توان برای شناسایی نواحی که نیاز به تنظیم کارایی یا موازی‌سازی دارند استفاده کرد.

نمای هسته‌ها: شکل (۸) اجرای الگوریتم ترتیبی را در مجموعه داده هزارتایی و در زمان ۶/۶۳ دقیقه نشان می‌دهد. نخ اصلی برنامه که به رنگ تیره مشخص است برای اجرا در بین هسته‌های پردازنده سوئیچ می‌کند و عملاً هیچ دو هسته‌ای به‌طور هم‌زمان در زیر همدیگر قرار ندارند. شکل (۹) وضعیت هسته‌ها به روش موازی را نشان می‌دهد. برنامه در ابتدا به‌صورت ترتیبی و با یک هسته پردازنده اجرا می‌شود ولی در ادامه برنامه به‌صورت موازی اجرا می‌شود و تراکم نخ‌های برنامه در این روش بیشتر است زیرا برخلاف روش ترتیبی تمام هسته‌های پردازنده را به‌صورت موازی برای اجرای برنامه بکار می‌گیرد. هم‌چنین در طول فرآیند تحلیل پروفایل کار، ابزار ویژولایزر هم‌زمانی همه رویدادها[۵۴] و تعویض متن‌های[۵۵] هر نخ برنامه را بررسی می‌کند.



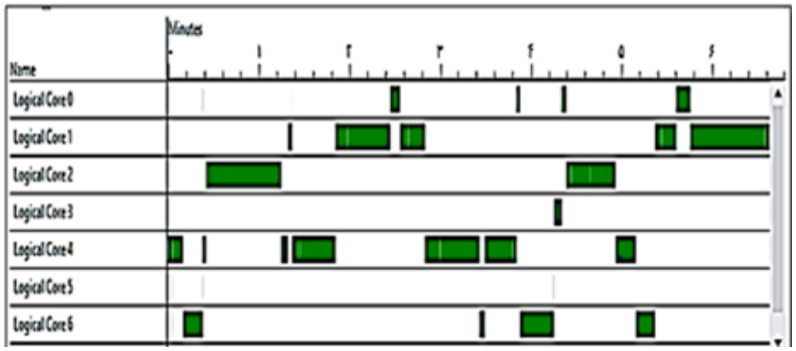
شکل (۶): میزان استفاده از پردازنده درروش ترتیبی

Figure (6): The CPU usage in sequential method



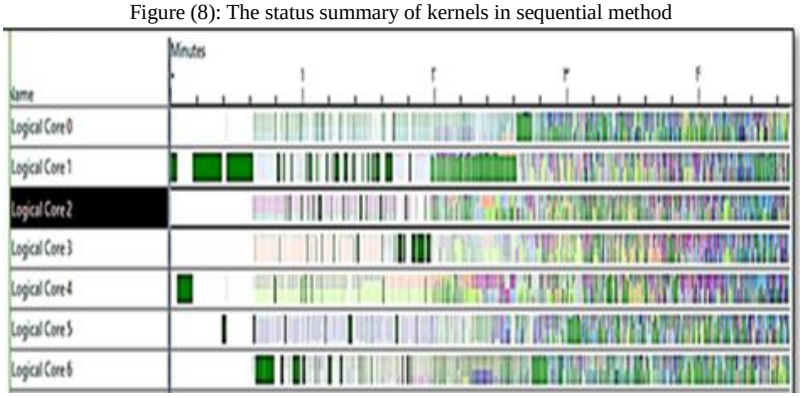
شکل (۷): میزان استفاده از پردازنده درروش موازی

Figure (7): The CPU usage in parallel method



شکل (۸): وضعیت هسته‌ها درروش ترتیبی

Figure (8): The status summary of kernels in sequential method



شکل (۹): وضعیت هسته‌ها درروش موازی

Figure (9): The summary status of kernels in parallel method

۴-۲-۳- آزمایش سوم: مقایسه روش ترتیبی و موازی از دیدگاه زمان اجرا با تعداد نخ متغیر

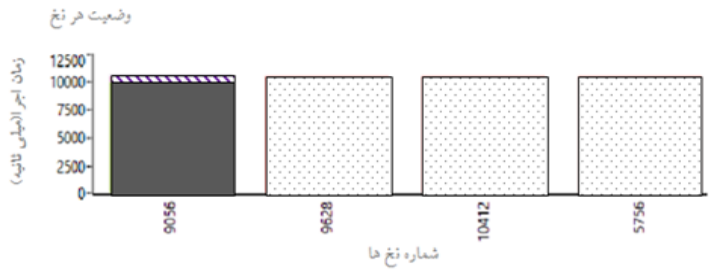
نمای نخ‌ها: در شکل (۱۰) و شکل (۱۱) وضعیت نخ‌ها موقع اجرای آزمایش‌ها به‌صورت ترتیبی و موازی آورده شده است. با استفاده از این نما، می‌توانیم تشخیص دهیم که آیا نخ‌ها در حال اجرا هستند یا اینکه به دلیل مسائلی مانند همگام‌سازی[۵۶]، ورودی/ خروجی، پایان یافتن بازه زمانی و یا دلایل دیگر مسدود شده‌اند. کانال‌های نخ: در ابزار هم‌زمانی یک کانال نخ وضعیت یک نخ را بارنگ منحصربه‌فردی نشان می‌دهد. با استفاده از اسم کانال تابع شروع برای آن نخ نمایش داده می‌شود. ابزار هم‌زمانی نخ‌های متعددی را کشف می‌کند. شایع‌ترین انواع نخ‌ها در جدول (۸) آورده شده است.

Table (8): Common types of threads

جدول (۸): شایع‌ترین انواع نخ

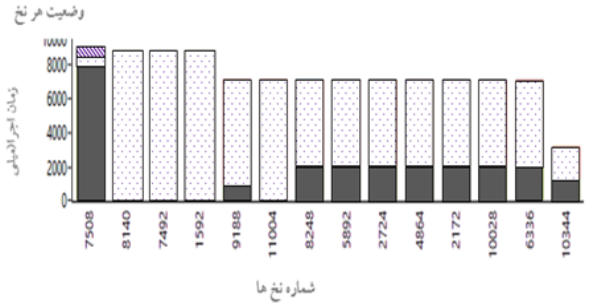
نخ اصلی	نخی که برنامه را شروع می‌کند.
نخ کارگر	نخی که توسط نخ اصلی برنامه ایجاد می‌شود.
زبان مشترک زمان اجرا[۵۷]	نخ کارگری که توسط سیستم زمان اجرای زبان مشترک ایجاد می‌شود.
اشکال‌زدای کمکی[۵۸]	نخ کارگری که توسط اشکال‌زدای ویژوال استودیو ایجاد می‌شود.
نخ RPC[۵۹]	نخی که به‌عنوان نخ RPC ایجادشده است.
استخر نخ[۶۰]	نخی که توسط استخر نخ CLR ایجاد می‌شود

با استفاده از نمای نخ‌ها در ابزار هم‌زمانی می‌توانیم درجه هم‌زمانی را تشخیص دهیم که این امر به موازی‌سازی بیشتر برنامه کمک می‌کند. همچنین موضوعات مربوط به تعادل بار برای اجرای موازی در این ابزار قابل‌بررسی است. هنگامی‌که نخ‌ها اجرا می‌شوند، ابزار هم‌زمانی می‌تواند نمونه‌هایی را جمع‌آوری کند. همچنین در نمای نخ‌ها، می‌توان تحلیل کرد که در طول یک بخش از اجرا کدام کد توسط یک یا چند نخ اجرا می‌شود.



شکل (۱۰): خلاصه‌ای از وضعیت نخ‌ها به روش ترتیبی
Figure (10): The status summary of sequential threads

در شکل‌ها نخ‌های اصلی برنامه به رنگ تیره مشخص هستند. همان‌طور که درروش ترتیبی نشان داده‌شده است نخ اصلی فقط بر روی یک هسته اجرا می‌شود و بقیه نخ‌ها که بارنگ روشن مشخص هستند مسئله همگام‌سازی را بر عهده‌دارند. درروش موازی، نخ اصلی برنامه که بارنگ تیره قابل‌نمایش است در بین هسته‌های پردازنده سوییچ می‌کند و از تمام قابلیت پردازنده برای اجرا استفاده می‌کند.



شکل (۱۱): خلاصه‌ای از وضعیت نخ‌ها به روش موازی
Figure (11): The status summary of parallel threads

۴-۲-۴- آزمایش چهارم: مقایسه روش ترتیبی و موازی از دیدگاه درصد استفاده از پردازنده کل سیستم

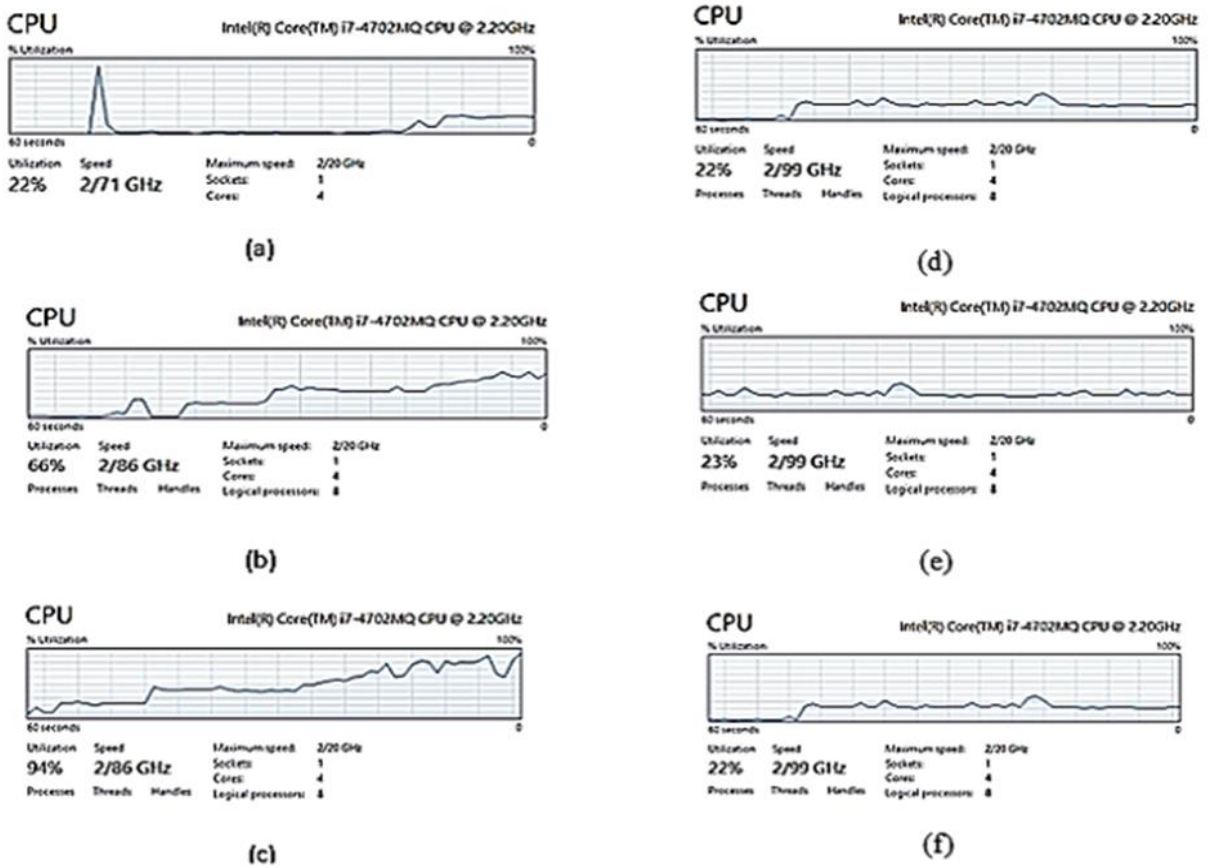
در شکل (۱۲-a,b,c) با مشاهده وضعیت پردازنده درروش موازی می‌توان متوجه شد که استفاده از پردازنده متغیر است. در ابتدای اجرای برنامه، استفاده از پردازنده ۲۲ درصد است و در ادامه باگذشت زمان اجرا، استفاده از پردازنده به ۶۶ درصد و درنهایت به ۹۴ درصد رسیده است. درواقع در اجرای برنامه به روش موازی بهره‌وری پردازنده به‌مرور افزایش پیدا می‌کند. درصورتی‌که در شکل (۱۲-d,e,f) با مقایسه وضعیت پردازنده کل سیستم می‌توان مشاهده کرد که موقع اجرای برنامه به روش ترتیبی وضعیت پردازنده ثابت است و استفاده از پردازنده یکنواخت است. دو واقع وضعیت استفاده از پردازنده تغییر محسوسی ندارد و به‌طور میانگین ۲۲درصد مواقع پردازنده در حال فعالیت بوده و حدود ۷۸ درصد مواقع پردازنده بیکار است.

۵- نتیجه‌گیری و راه‌کارهای آینده

در آزمایش‌های انجام‌شده کارایی الگوریتم خوشه‌بندی سلسله‌مراتبی تراکمی با روش پیشنهادی که مبتنی بر مدل نگاشت-کاهش است با روش پیشین که به‌صورت ترتیبی بود مورد ارزیابی قرار گرفت. برای این منظور، چهار مجموعه آزمایش برای بررسی میزان بهبود این الگوریتم انجام شد. هر آزمایش یک‌بار به روش قبلی و یک‌بار به روش پیشنهادی انجام شد. در این آزمایش‌ها از مجموعه داده‌های تصادفی برای ارزیابی استفاده شد. زمان اجرای کارها درروش پیشنهادی و درروش قبلی اندازه‌گیری شد. در این آزمایش‌ها زمان اجرای الگوریتم به روش موازی نسبت به روش ترتیبی برای مجموعه داده‌ای به‌اندازه ۲۰۰ شیء داده، ۱۶/۸۰درصد و برای مجموعه داده‌ای به‌اندازه ۱۰۰۰ شیء داده، ۲۹/۲۶درصد بهبود یافت. با بررسی وضعیت اجرای الگوریتم به‌وسیله این دو روش می‌توان نتیجه گرفت که هر چه اندازه داده ورودی بیشتر باشد اجرای الگوریتم بهینه‌تر است و کاهش زمان اجرا محسوس‌تر است. درواقع، برای مجموعه داده‌های بیشتر به خاطر نوع ساختار برنامه موازی‌سازی

بیشتری صورت می‌گیرد و بهره‌وری پردازنده افزایش می‌یابد. بر اساس آزمایش‌ها انجام‌شده، درروش پیشنهادی هر چه به سمت ریشه گراف دندروگرام نزدیک می‌شویم، عمل موازی‌سازی بیشتری صورت گرفته و عملکرد الگوریتم پیشنهادی بهتر خود را نشان می‌دهد. درصد استفاده از پردازنده کل سیستم درروش ترتیبی و موازی نیز مورد مقایسه قرار گرفت.

بر اساس تجربیات به‌دست‌آمده در هنگام انجام این تحقیق ارائه راهکارهای مشابه برای سایر الگوریتم‌های خوشه‌بندی پیشنهاد می‌گردد. ارائه راه‌حل‌های نگاشت-کاهش به همراه استفاده از امکانات پردازش گرافیکی جهت بالا بردن درجه توازی نیز می‌تواند از کارهای آینده باشد. به‌کارگیری متد پیشنهادی در این تحقیق برای الگوریتم‌های دسته‌بندی داده‌ها می‌تواند برای تحقیقات آینده در نظر گرفته شود.



شکل (۱۲): وضعیت پردازنده سیستم درروش موازی (a,b,c) و روش ترتیبی (d,e,f)

Figure (12): The processor status in the parallel method (a, b, c) and the sequential method (d, e, f)

References

مراجع

[1] E. Begoli and J. Horey, "Principles for Effective Knowledge Discovery from Big Data", Proceeding of the IEEE/IFIP, Helsinki, Finland, pp. 1-4, Aug 2012 (doi: 10.1109/WICSA-ECSA.212.32).

[2] A. Jain and R. Dubes," Algorithms for Clustering Data", NJ: Prentice-Hall, Inc., January 1988.

[3] Jain, A.K., Murty, M.N. and Flynn, P.J., "Data clustering: a review", ACM computing surveys (CSUR), vol. 31, no. 3, pp. 264-323, 1999 (doi: 10.1145/331499.331504).

[4] Dean, J. and Ghemawat, S., "MapReduce: simplified data processing on large clusters", Communications of the ACM, 51(1), pp.107-113 (doi: 10.1145/1327452.1327492).

[5] Rohlf, F.J., "Hierarchical clustering using the minimum spanning tree Computer", The Computer Journal, vol. 16, pp. 93-95, 1973.

[6] F. Murtagh, "Multidimensional Clustering Algorithms," Physica-Verlag, 1985 .

[7] J. Ward, "Hierarchical grouping to optimize an objective function," Journal of the American Statistical Association, vol. 58, pp. 236-244, 1963 .

[8] G. Lance and W. Williams, "A General Theory of Classificatory Sorting Strategies 1. Hierarchical Systems", The Computer Journal, vol. 9, no. 4, pp. 373-380, 1967 .
[9] W. Pengcheng, H. Fangcheng, L. Li, S. Chuanfu and J. Li, "Research on large data set clustering method based on MapReduce", Neural Computing and Applications, vol. 32, p. 93–99, 2020 (doi: 10.1007/s00521-018-3780-y)
[10] V. S. Bawane and S. M. Kale, "Clustering algorithms in map reduce: A Review", International Journal of Computer Applications, vol. 975, pp. 15-18, 2015.
[11] D. C. Radhika. P. Lalita., "Distributed clustering for big data with map reduce," IOSR Journal of Computer Engineering, vol. 19, no. 3, pp. 25-28, May 2017 (doi: 10.9790/0661-1903032528).
[12] S. Alshammari, M. Binti Zolkepli, B. A. Rusli, "Genetic algorithm based parallel k-means data clustering algorithm using MapReduce programming paradigm on Hadoop environment (GAPKCA)", International Conference on Soft Computing and Data Mining, SCDM 2020: Recent Advances on Soft Computing and Data Mining, Langkawi, Malaysia, Jan. 2020.
[13] Y. Miao, J. Zhang, H. Feng, L. Qiu and Y. Wen, "A fast algorithm for clustering with map reduce", vol. 7951, pp. 25-38, Berlin, Heidelberg: Lecture Notes in Computer Science, 2013 .
[14] P. P. Anchalia, A. Koundinya and S. Nk, "MapReduce design of K-means clustering algorithm", Proceeding of the IEEE/ ICISA, Suwon, South Korea, pp. 1-5, June 2013 (doi: 10.1109/ICISA.2013.6579448).
[15] W. Zhao, H. Ma, Q. He, "Parallel K-means clustering based on MapReduce", Cloud Computing, CloudCom 2009. Lecture Notes in Computer Science, vol. 5931, p. 674–679, Berlin Heidelberg: Springer-Verlag LNCS, 2009 (doi: 10.1007/978-3-642-10665-1_71).
[16] J. Ragaventhiran, D. M. Kavitha, "Map-optimize-reduce: CAN tree assisted FP-growth algorithm for clusters-based FP mining on Hadoop", Future Generation Computer Systems, vol. 103, pp. 111-122, Feb. 2020 (doi: 10.1016/j.future.2019.09.041).
[17] T. Habib Sardar, Z. Ansari, "An analysis of MapReduce efficiency in document clustering using parallel K-means algorithm", Future Computing and Informatics Journal, vol. 3, no. 2, pp. 200-209, Dec. 2018 (doi: 10.1016/j.fcij.2018.03.003).
[18] M. A. B. Haj-Kacem, C.-E. Ben N'cir. N. Essoussi., "One-pass MapReduce-based clustering method for mixed large-scale data", Journal of Intelligent Information Systems, vol. 52, pp. 619–636, June 2019 (doi: 10.1007/s10844-017-0472-5).
[19] S. Chowdam., N. Kasiviswanath. P. C. Reddy., "Clustering large datasets using K-means modified inter and intra clustering (KM-I2C) in Hadoop", Journal of Big Data, vol. 4, no. 1, pp. 1-27, 2017 (doi: 10.1186/s40537-017-0087-2).
[20] D. Teffer, R. Srinivasan, J. Ghosh, "AdaHash: Hashing-based scalable, adaptive hierarchical clustering of streaming data on MapReduce frameworks", International Journal of Data Science and Analytics, vol. 8, pp. 257–267, 2019 (doi: 10.1007/s41060-018-0145-7)
[21] "wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Jaccard_index.
[22] M. Ngazimbi, "Data clustering using MapReduce", PhD Thesis, Boise State University, 2009.
[23] T. Sun, C. Shu, F. Li, H. Yu, L. Ma, Y. Fang, "An efficient hierarchical clustering method for large datasets with Map-Reduce", Proceeding of the IEEE/PDCAT, pp. 494-499, 2009 (doi: 10.1109/PDCAT.2009.46).
[24] S. Papadimitriou, J. Sun, "DisCo: Distributed co-clustering with MapReduce: A case study towards petabyte-scale end-to-end mining", Proceeding of the IEEE/ICDM, Pisa, Italy, Dec. 2008 (doi: 10.1109/ICDM.2008.142).
[25] E. M. Rasmussen and P. Willett, "Efficiency of hierarchical agglomerative clustering using the ICL distributed array processor", Journal of Documentation, vol. 45, no. 1, pp. 1-24, 1989 (doi: 10.1108/eb026836).
[26] C. F. Olson, "Parallel Algorithms for Hierarchical Clustering", Parallel Computing, vol. 21, no. 8, pp. 1313-1325, Aug. 1995 (10.1016/0167-8191(95)00017-I).
[27] Z. Li, K. Li, D. Xiao, L. Yang, "An adaptive parallel hierarchical clustering algorithm", Proceeding of the HPCC, Springer, Berlin, Heidelberg, pp. 97-107, 2007 (doi: 10.1007/978-3-540-75444-2_15).
[28] S. Rajasekara, "Efficient parallel hierarchical clustering algorithms", IEEE Trans. on Parallel and Distributed Systems, vol. 16, no. 6, pp. 27-34, June 2005 (doi: 10.1109/TPDS.2005.72).
[29] M. Dash, S. Petrutiu, P. Scheuermann, "Efficient parallel hierarchical clustering", IEEE Trans. on Parallel and Distributed Systems, vol. 16, no. 6, pp. 497-502, 2005 (doi: 10.1109/TPDS.2005.72).
[30] S. Guha, R. Rastogi, K. Shim, "ROCK: A robust clustering algorithm for categorical attributes", Information System, vol. 32, no. 8, pp. 345- 366, 2000 (doi: 10.1016/S0306-4379(00)00022-3).
[31] G. Karypis, E. Han, V. Kumar, "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling", Computer, vol. 32, no. 8, pp. 68-75, Aug. 1999 (doi: 10.1109/2.781637).
[32] E. L. Johnson, H. Kargupta, "Collective, hierarchical clustering from distributed, heterogeneous data", Large-Scale Parallel Data Mining, pp. 221-244, 2000 (doi: 10.1007/3-540-46502-2_12).

[33] N. Samatova, G. Ostrouchov, A. Geist and A. Melec, "RACHET: An efficient cover-based merging of clustering hierarchies from distributed datasets", Distributed and Parallel Databases, vol. 11, no. 2, pp. 157-180, 2002 (doi: 10.1023/A:1013988102576).

[34] H. Gao, J. Jiang, L. She, Y. Fu, "A new agglomerative hierarchical clustering algorithm implementation based on the map reduce framework", International Journal of Digital Content Technology and its Applications, vol. 4, no. 3, 2016.

[35] T. Zhang, R. Ramakrishnan, M. Livny, "BIRCH: An efficient data clustering method for very large databases", SIGMOD, vol. 25, no. 2, Association for Computing Machinery, Jun. 1996 (doi: 10.1145/235968.233324).

[36] "Apache Hadoop," [Online]. Available: http://hadoop.apache.org. [Accessed 2019].

زیرنویس‌ها:

1. Data clustering
2. Data mining
3. Workers
4. Threads
5. Sequential
6. Data objects
7. Hierarchical clustering
8. Dendrogram
9. Agglomerative
10. Divisive
11. Single connection
12. Average link
13. Complete link
14. Crime center
15. Euclidean
16. Middle method
17. Maximum variance
18. Lance williams
19. Single
20. Complete
21. Average
22. Centroid
23. Median
24. Ward
25. Weighted average
26. Single link
27. Jaccard index
28. Jaccard similarity coefficient
29. Asymmetric information
30. Binary variables
31. Jaccard distance
32. Dissimilarity
33. Cluster centers
34. Netflix
35. Log
36. Batch updating
37. Co-occurrence
38. Noise
39. Text mining
40. Collaborative filtering
41. Graph mining
42. Single linkage
43. Single Instruction Multi Data
44. Centroid linkage
45. Spanning tree
46. Reconfigurable optical busses
47. Partially overlapping partitioning-POP
48. Clustering features – CF
49. Vector processing self-organizing model
50. Concurrency visualizer
51. System process
52. Idle process
53. Degree of concurrency
54. Events
55. Context switches
56. Synchronization
57. Common language runtime-CLR
58. Debugger
59. Remote Procedure Call
60. Thread pool