۱۴۰۱/۱۰/۳۰
چهارمین همایش ملی مهندسی کامپیوتر و نرم افزار
4TH NATIONAL CONFERENCE ON COMPUTER AND SOFTWARE ENGINEERING

# تجزیه و تحلیل کلان داده

نویسندگان : **فتح اله ابراهیمی ؛ کامبیز قناعت**

دانشگاه فنی حرفیه ای امام خامنه ای بوشهر – رشته مهندسی حرفه ای کامپیوتر

**ایمیل:**

**فتح اله ابراهیمی :** fathallah1379@gmail.com

**کامبیز قناعت :** ali.79.reza@gmail.com

## چکیده

عصر کلان داده در حال حاضر فرا رسیده است. اما تجزیه و تحلیل داده های سنتی ممکن است قادر نباشد .برای مدیریت چنین حجم زیادی از داده ها مقدور باشد. سوالی که اکنون مطرح می شود این است که روش پردازش را چگونه توسعه دهیم. یک پلت فرم با کارایی بالا برای تجزیه و تحلیل کارآمد داده های بزرگ و نحوه طراحی یک الگوریتم استخراج مناسب برای یافتن چیزهای مفید از داده های بزرگ میباشد. برای بحث عمیق در این موضوع، این مقاله با مقدمه ای کوتاه بر تجزیه و تحلیل داده ها آغاز می شود و به دنبال آن، بحث در مورد تجزیه و تحلیل داده های بزرگ برخی از مسائل مهم باز و تحقیقات بیشتر همچنین دستورالعمل‌هایی برای مرحله بعدی تجزیه و تحلیل کلان داده ارائه خواهد شد.

کلمات کلیدی: کلان داده، تجزیه و تحلیل داده ، MapReduce, MPI

## ۱- مقدمه

این دوره مربوط به داده های بزرگ است. Big Data باعث ایجاد تغییرات اساسی در تجزیه و تحلیل داده های سنتی می شود.

برای انجام هر نوع تحلیل بر روی چنین داده های حجیم و پیچیده ای، افزایش مقیاس پلت فرم های سخت افزاری قریب الوقوع می شود و اگر خواسته‌های کاربر برآورده شود، پلتفرم‌های نرم‌افزاری به یک تصمیم حیاتی تبدیل می‌شوند. در یک زمان معقول محققان روی ساختن زمان کار کرده اند. تکنیک‌های تجزیه و تحلیل داده‌ها برای کلان داده‌ها بیش از هر زمان دیگری منجر به پیوستگی شده است. چندین پلتفرم کلان داده با ویژگی های مختلف و انتخاب پلت فرم مناسب مستلزم داشتن دانش عمیق در مورد قابلیت های آن است. به ویژه، توانایی پلت فرم ها برای انطباق با داده های افزایش یافته است. پردازش درخواست‌ها نقش مهمی در تصمیم‌گیری برای ایجاد تجزیه و تحلیل مناسب دارد.

راه حل های مبتنی بر یک پلت فرم خاص برای این منظور، ابتدا به ارائه یک گزارش کامل خواهیم پرداخت که درک همه پلتفرم های کلان داده محبوبی که در حال حاضر استفاده می شوند ,در عمل مزایا و معایب هر یک از آنها را برجسته میکنیم.

## ۲- مقیاس بندی

مقیاس بندی توانایی سیستم برای انطباق با تقاضاهای افزایش یافته از نظر پردازش داده است. برای پشتیبانی از پردازش کلان داده، پلتفرم‌های مختلف مقیاس‌بندی را در موارد مختلف ترکیب می‌کنند. از منظری گسترده تر، پلتفرم های کلان داده را می توان دسته بندی کرد.

به دو نوع مقیاس بندی زیر:

مقیاس افقی: مقیاس افقی شامل توزیع حجم کار در سراسر آن است. بسیاری از سرورها که ممکن است حتی ماشین های کالایی باشند نیز همچنین به عنوان (مقیاس بیرون)، که در آن چندین ماشین مستقل به منظور بهبود با هم اضافه می شوند,

قابلیت پردازش چندین نمونه از سیستم عامل را دارند و در ماشین های جداگانه اجرا می شود.

مقیاس عمودی: مقیاس عمودی شامل نصب پردازنده های بیشتر، بیشتر است. حافظه و سخت افزار سریعتر، معمولاً در یک سرور واحد است. همچنین به عنوان "scale up" شناخته میشود و معمولاً شامل یک نمونه واحد از یک سیستم عامل است.

با توجه به اینکه افزایش مقیاس به صورت عمودی می تواند مدیریت و نصب را آسان کند.

## ۳- شبکه های همتا به همتا ( peer to peer )

شبکه های همتا به همتا شامل میلیون ها ماشین متصل به یک شبکه است. این است یک معماری شبکه غیرمتمرکز و توزیع شده که در آن گره ها در شبکه ها وجود دارند, (معروف به همتایان) در خدمت و همچنین مصرف منابع هستند. این نوع شبکه یکی از قدیمی ترین شبکه های توزیع شده است. پلتفرم های محاسباتی موجود به طور معمول، رابط ارسال پیام (MPI) است. طرح ارتباطی مورد استفاده در چنین تنظیماتی برای برقراری ارتباط و تبادل داده ها بین همسالان هر گره می‌تواند نمونه‌های داده را ذخیره کند و مقیاس‌بندی عملاً انجام می‌شود. (نامحدود (می تواند میلیون ها گره باشد)).

گلوگاه اصلی در چنین تنظیماتی در ارتباط بین گره های مختلف بوجود می آید. گره های پخش پیام‌ها در شبکه‌های همتا به همتا ارزان‌تر است اما انباشته داده ها/نتایج بسیار گران تر است. علاوه بر این، پیام ها از طریق شبکه ارسال می شوند. به شکل یک درخت پوشا با یک گره دلخواه به عنوان ریشه که در آن پخش آغاز شده است.

MPI که پارادایم استاندارد ارتباط نرم افزاری مورد استفاده در این شبکه است. چندین سال است که استفاده می شود و به خوبی تثبیت شده و به طور کامل اشکال زدایی شده است.

یکی از ویژگی های اصلی MPI می توان به فرآیند حفظ حالت اشاره کرد، یعنی فرآیندها می توانند زنده بمانند تا زمانی که سیستم اجرا شود و نیازی به خواندن دوباره و دوباره همان داده ها نباشد. مانند سایر فریمورک‌ها مانند MapReduce (توضیح داده شده در بخش Apacheهادوپ»). تمام پارامترها را می توان به صورت محلی حفظ کرد. از این رو، برخلاف MapReduce، MPI به خوبی برای پردازش تکراری مناسب است. یکی دیگر از ویژگی های MPI سلسله مراتبی است. پارادایم ارباب/برده هنگامی که MPI در مدل master-slave مستقر می شود، Slave ماشین می تواند استاد سایر فرآیندها شود. این می تواند بسیار مفید باشد برای تخصیص منابع پویا که در آن بردگان مقادیر زیادی داده برای پردازش دارند. MPI برای بسیاری از زبان های برنامه نویسی در دسترس است. که شامل روش های ارسال و دریافت پیام ها و داده ها است. برخی از روش های دیگر موجود با MPI عبارتند از "پخش"، که برای پخش داده ها یا پیام ها بر روی تمام گره ها و 'Barier' استفاده می شود و روش دیگری است که می تواند مانع ایجاد کند و اجازه می دهد تمام فرآیندها همگام شوند

و قبل از ادامه به یک نقطه خاص برسیم. اگرچه به نظر می رسد MPI برای توسعه الگوریتم هایی برای تجزیه و تحلیل داده های بزرگ عالی است، اما چند اشکال عمده دارد یکی از اشکالات اصلی عدم تحمل خطا است.

از آنجایی که MPI مکانیزمی برای رسیدگی به خطاها ندارد. هنگامی که در بالای شبکه های همتا به همتا استفاده می شود،

که یک سخت افزار کاملاً غیر قابل اعتماد است، یک شکست گره می تواند باعث ایجاد این مشکل شود و کل سیستم خاموش شود و کاربران باید نوعی تحمل خطا را پیاده سازی کنند. مکانیزم درون برنامه برای جلوگیری از چنین شرایط ناگواری با سایر چارچوب ها مانند Hadoop (که نسبت به تحمل خطا قوی هستند) به طور گسترده محبوب شده است و MPIدیگر به طور گسترده مورد استفاده قرار نمی گیرد.

**۴- آپاچی هادوپ**

Apache Hadoop یک چارچوب متن باز برای ذخیره و پردازش مجموعه داده های بزرگ با استفاده از خوشه های سخت افزار کالا است. Hadoop به گونه‌ای طراحی شده است که تا صدها عدد برسد و حتی هزاران گره تحمل خطای بالایی دارد. اجزای مختلف یک پشته Hadoop در شکل ۱ نشان داده شده است. پلت فرم Hadoop شامل موارد زیر است :

دو جزء مهم:

سیستم فایل توزیع شده (HDFS) , یک سیستم فایل توزیع شده است که برای ذخیره سازی استفاده می شود. داده ها در سراسر خوشه ماشین آلات کالا هستند در حالی که در دسترسی بالا و تحمل خطای آن نیز زیاد است. خوشه Hadoop YARN یک لایه مدیریت منابع است و کارها را در سرتاسر زمان بندی می کند.

MapReduce

مدل برنامه نویسی مورد استفاده در Hadoop MapReduce است که توسط گوگل پیشنهاد شده است. MapReduce طرح اصلی پردازش داده مورد استفاده است در Hadoop که شامل تقسیم کل کار به دو بخش است که به عنوان نقشه‌بردار شناخته می‌شوند و کاهنده ها در سطح بالا، نقشه‌برداران داده‌ها را از HDFS می‌خوانند، آن‌ها را پردازش می‌کنند و

برخی نتایج میانی را برای کاهنده ها ایجاد می کند. کاهش دهنده ها برای تجمع استفاده می شوند. نتایج میانی برای تولید خروجی نهایی که دوباره در HDFS هستند نوشته می شود.

یک کار معمولی Hadoop شامل اجرای چندین نقشه‌بردار و کاهش‌دهنده در سراسر مختلف است.گره ها در خوشه یک نظرسنجی خوب در مورد MapReduce برای پردازش داده های موازی است.

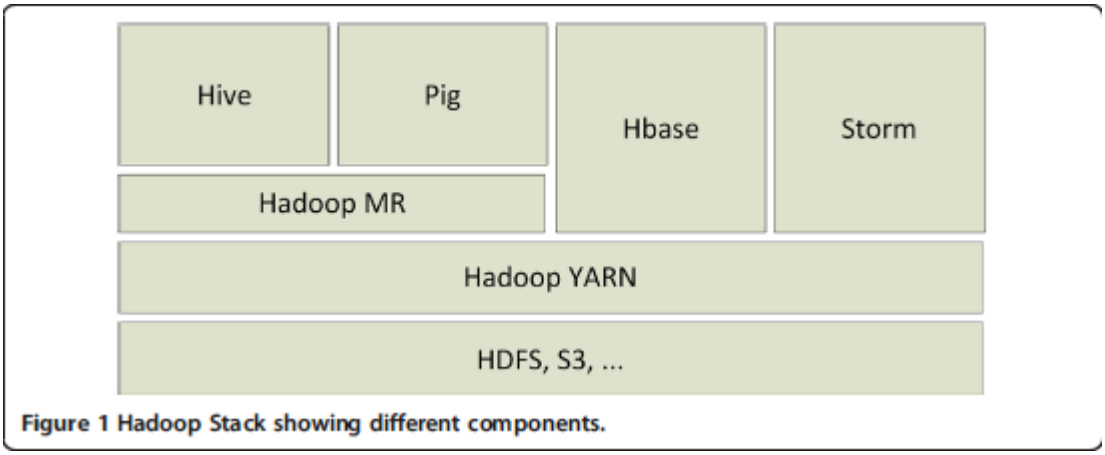

Figure 1 Hadoop Stack showing different components.

Spark: نسل بعدی پارادایم تجزیه و تحلیل داده ها

Spark یک الگوی نسل بعدی برای پردازش کلان داده است که توسط محققین توسعه یافته است. دانشگاه کالیفرنیا در فکر این که یک جایگزین برای Hadoop بسازد, که Spark را طراحی کرده اند. برای غلبه بر محدودیت های ورودی/خروجی دیسک و بهبود عملکرد سیستم های قبلی مورد استفاده قرار میگیرد. ویژگی اصلی Spark که آن را منحصر به فرد می کند، توانایی آن در اجرای حافظه است. محاسبات این اجازه را می دهند تا داده ها در حافظه پنهان شوند، بنابراین محدودیت سربار دیسک Hadoop برای کارهای تکراری را حذف میکند. اسپارک یک موتور عمومی برای پردازش داده در مقیاس بزرگ که از جاوا، اسکالا و پایتون برای کارهای خاصی از آن پشتیبانی می کند. آزمایش شده است تا ۱۰۰× سریعتر از Hadoop MapReduce زمانی که داده ها می توانند در یادگیری ماشین برای داده کاوی بزرگ ,پتانسیل یادگیری ماشین برای تجزیه و تحلیل داده ها را می توان به راحتی در ادبیات اولیه یافت. متفاوت از طراحی الگوریتم داده کاوی برای مشکلات خاص و الگوریتم های یادگیری ماشین را می توان برای مسائل مختلف استخراج و تجزیه و تحلیل استفاده کرد. زیرا آنها معمولاً به عنوان الگوریتم "جستجو" راه حل مورد نیاز استفاده می شوند. از آنجایی که اکثر الگوریتم های یادگیری ماشینی را می توان

برای یافتن یک راه حل تقریبی استفاده کرد, برای مسئله بهینه سازی، می توان آنها را برای اکثر مسائل تجزیه و تحلیل داده استفاده کرد.

مسائل تجزیه و تحلیل داده ها را می توان به عنوان یک مسئله بهینه سازی فرموله کرد. مثلا، الگوریتم ژنتیک، یکی از الگوریتم های یادگیری ماشین، نه تنها می تواند در حل مشکل خوشه بندی مورد استفاده قرار گیرد ، همچنین می توان از آن برای حل الگوی کاوی مکرر استفاده کرد.

پتانسیل یادگیری ماشین صرفاً برای حل مسائل مختلف نیست. همنین برای مشکلات داده کاوی در اپراتور تجزیه و تحلیل داده های KDD نیز میباشد. همچنین دارای پتانسیل افزایش است.

طبق مطالعه اخیر:  نشان می دهد که برخی از الگوریتم های استخراج سنتی، روش های آماری، راه حل های پیش پردازش و حتی رابط کاربری گرافیکی برای چندین نماینده اعمال شده است.

ابزارها و پلتفرم ها برای تجزیه و تحلیل داده های بزرگ, نتایج به وضوح آن ماشین را نشان می دهد. الگوریتم های یادگیری یکی از بخش های ضروری تجزیه و تحلیل داده های بزرگ خواهند بود. یکی از مشکلات در استفاده از روش های یادگیری ماشین فعلی برای تجزیه و تحلیل داده های بزرگ مشابه است. الگوریتم های بسیاری از الگوریتم های داده کاوی سنتی که به صورت متوالی طراحی شده اند.

محاسبات متمرکز با این حال، یکی از راه حل های ممکن، ساخت آنها است. کار برای محاسبات موازی خوشبختانه، برخی از الگوریتم های یادگیری ماشین (به عنوان مثال، الگوریتم‌های مبتنی بر جمعیت) اساساً می‌توانند برای محاسبات موازی استفاده شوند که

برای چندین سال نشان داده شده است، مانند نسخه محاسباتی موازی الگوریتم ژنتیک. متفاوت از GA سنتی، همانطور که در شکل a۹ نشان داده شده است، جمعیت از

الگوریتم ژنتیک مدل جزیره، یکی از GAهای موازی، می تواند به چند زیرجمعیت تقسیم شود.

همانطور که در شکل b۹ نشان داده شده است. این بدان معنی است که می توان جمعیت های فرعی را به آنها اختصاص داد

رشته ها یا گره های کامپیوتری مختلف برای محاسبات موازی، با یک اصلاح ساده GA بیان میشود, به همین دلیل، در [۱۲۳]، کیران و بابو توضیح دادند که چارچوب برای توزیع شده است.

الگوریتم داده کاوی هنوز نیاز به جمع آوری اطلاعات از رایانه های مختلف دارد. گره ها همانطور که در شکل ۱۰ نشان داده شده است، طراحی رایج الگوریتم داده کاوی توزیع شده است:

به شرح زیر: هر الگوریتم ماینینگ بر روی یک گره کامپیوتری (کارگر) انجام خواهد شد که داده‌های منسجم محلی خود را دارد، اما نه کل داده‌ها برای ساختن یک مفهوم جهانی.

دانش پس از هر الگوریتم استخراج، مدل محلی خود را پیدا می کند، مدل محلی از هر گره کامپیوتری باید ادغام شده و در یک مدل نهایی برای نمایش ادغام شود.

دانش کامل کیران و بابو [۱۲۳] نیز اشاره کردند که ارتباط در هنگام استفاده از این نوع چارچوب محاسباتی توزیع شده، گلوگاه خواهد بود.

بو و همکاران [۱۲۴] هنگام تلاش برای اعمال الگوریتم‌های یادگیری ماشین، برخی مسائل تحقیقاتی را پیدا کردند.

به پلتفرم های محاسباتی موازی به عنوان مثال، نسخه اولیه کاهش نقشه ,چارچوب از "تکرار" (به عنوان مثال، بازگشت) پشتیبانی نمی کند. اما خبر خوب این است که برخی از آثار اخیر به این مشکل توجه زیادی داشته و سعی در رفع آن داشته اند. مشابه

به راه حل هایی برای افزایش عملکرد الگوریتم های سنتی داده کاوی، یکی از راه حل های ممکن برای افزایش عملکرد یک ماشین

الگوریتم یادگیری استفاده از CUDA، یعنی یک GPU، برای کاهش زمان محاسباتی داده است.

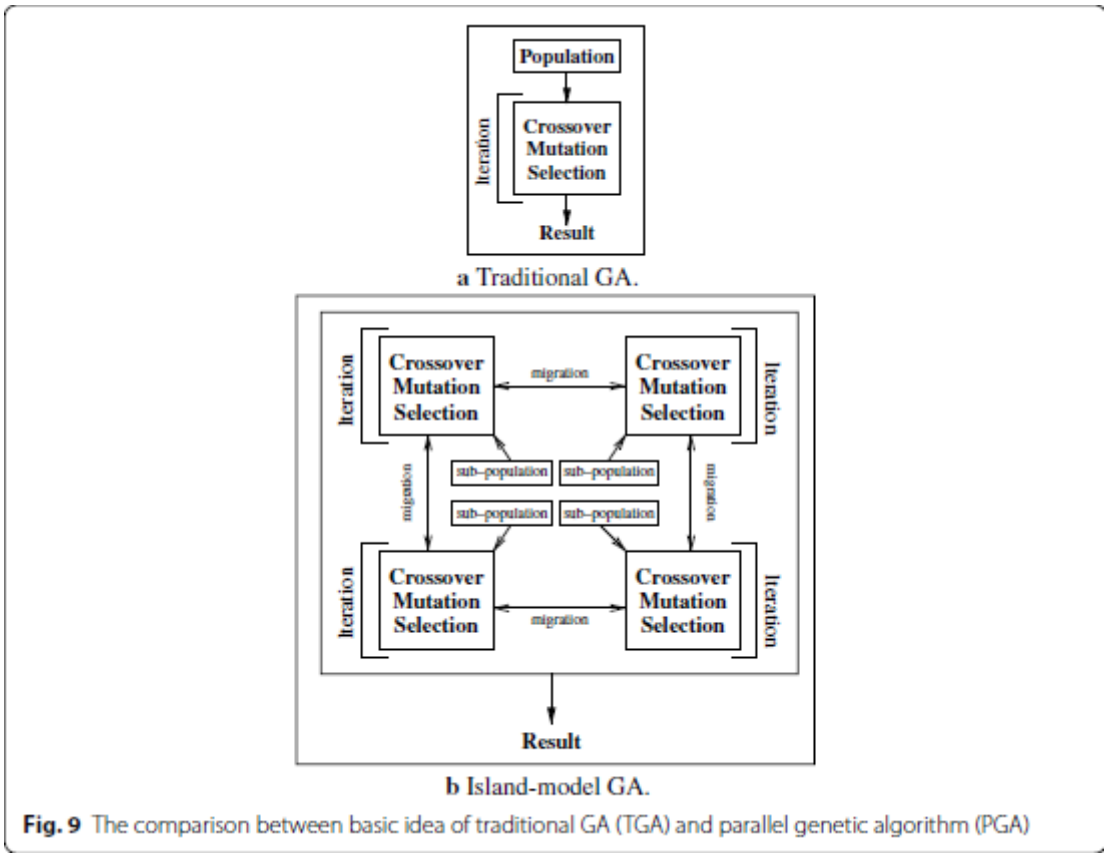


**Fig. 9** The comparison between basic idea of traditional GA (TGA) and parallel genetic algorithm (PGA)

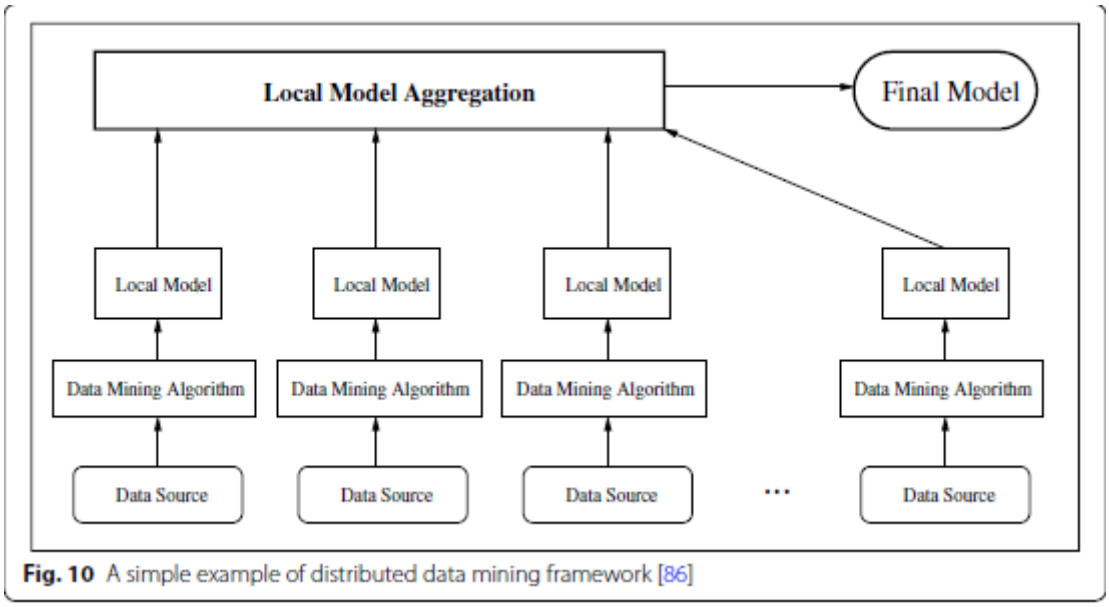


**Fig. 10** A simple example of distributed data mining framework [86]

**۵- نتیجه گیری**

در این مقاله، ما مطالعات مربوط به تجزیه و تحلیل داده ها را از تجزیه و تحلیل داده های سنتی مرور کردیم به تجزیه و تحلیل داده های بزرگ اخیر از دیدگاه سیستم، فرآیند KDD استفاده می شود.

چارچوبی برای این مطالعات است و در سه بخش ورودی، تحلیل، و خروجی از منظر چارچوب و پلت فرم تجزیه و تحلیل داده های بزرگ، بحث ها بر روی مسائل عملکرد محور و نتیجه گرا متمرکز هستند. از دیدگاه مسئله داده کاوی، این مقاله به معرفی مختصری از داده ها و الگوریتم های کلان داده کاوی که از خوشه بندی، طبقه بندی و الگوهای مکرر تشکیل شده است. برای درک بهتر تغییرات ایجاد شده توسط بزرگ داده‌ها، این مقاله بر تجزیه و تحلیل داده‌های KDD از پلتفرم/چارچوب ها تمرکز دارد. داده کاوی مسائل باز در مورد محاسبات، کیفیت نتیجه نهایی، امنیت و حریم خصوصی سپس مورد بحث قرار می گیرند تا توضیح دهند که با کدام مسائل باز ممکن است مواجه شویم . در نهایت GPU (CUDA) میتواند افزایش سرعت چشم گیری در پردازش داده های کلان طبق نتایج بدست آمده و آزمایشات به ما بدهد.

**منابع :**

1. https://scholar.google.com/
2. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0030-3
3. https://link.springer.com/article/10.1186/s40537-014-0008-6