نگاشت کاهش: تجزیه و تحلیل ساده شده کلان داده

چکیده

با توسعه تکنولوژی رایانه، افزایش قابل توجهی در رشد داده‌ها وجود دارد. دانشمندان با توجه به مقدار میزان افزایش نیازهای پردازش داده‌ها که در حوزه علم ایجاد می‌شوند همیشه غرق هستند. یک مسئله بزرگ در زمینه‌های مختلف استفاده از داده‌های با مقیاس بزرگ وجود دارد و این مسئله همیشه با پشتیبانی تصمیم‌گیری مواجه است. داده کاوی تکنیکی است که می‌تواند الگوهای جدیدی را از مجموعه کلان داده‌ها پیدا کند. در طی این سال‌ها تمام زمینه‌های کاربردی مورد مطالعه و بررسی قرار گرفتند و بسیاری از روش‌های داده کاوی را توسعه داده‌اند. اما در سال‌های مقدار زیادی از داده‌ها، محاسبات و تجزیه و تحلیل آنها به طور قابل توجهی افزایش یافته‌اند. در چنین موقعیتی، اکثر روش‌های داده کاوی در عمل برای دسترسی به چنین کلان داده‌هایی از دسترس خارج شدند. الگوریتم موازی/همزمان و تکنیک‌های پیاده‌سازی به طور موثر کلید ارزیابی مقیاس-پذیری و عملکرد مورد نیاز در تجزیه و تحلیل کلان داده‌ها در مقیاس بزرگ می‌باشند. تعدادی از الگوریتم‌های موازی دارای تکنیک‌های مختلف پیاده‌سازی هستند و می‌توانند: از نگاشت کاهش، MPI، بندها، و mash-up یا گردش کار که دارای ویژگی‌های عملکردی و قابلیت‌های متفاوتی هستند استفاده کنند. مدل MPI به طور موثر در محاسبه مسئله، به ویژه در شبیه‌سازی به دست می‌آید. در حقیقت استفاده از آن کار ساده‌ای نیست. نگاشت کاهش از یک مدل تجزیه و تحلیل داده‌ها در زمینه بازیابی داده‌ها است و به صورت فناوری ابر توسعه پیدا کرده است. تاکنون، چندین معماری نگاشت کاهش برای دست زدن به کلان داده‌ها توسعه پیدا کرده‌اند. معروف‌ترین آنها گوگل است. یکی دیگر از ویژگی‌های هادوپ این است که محبوب‌ترین نرم‌افزارها، نرم افزار منبع باز نگاشت کاهش است و توسط بسیاری از شرکت‌های فناوری اطلاعات مانند یاهو، فیس بوک، eBay و غیره مورد پذیرش قرار گرفته است. در این مقاله، ما به طور خاص روی هادوپ و اجرای موثر نگاشت کاهش به منظور تحلیل

پردازش تمرکز می‌کنیم.

کلمات کلیدی: کلان داده، داده کاوی، تکنیک‌های موازی کردن، HDFS، نگاشت کاهش، هادوپ

1. مقدمه

سازمان‌ها از چندین مقادیر که داده‌های ساختاری بسیار دشواری دارند و از تکنولوژی DBMS برای پردازش و تجزیه و تحلیل داده‌ها استفاده می‌کنند. این نوع مسئله با شرکت‌های مبتنی بر وب مانند فیس بوک، یاهو، گوگل و لینکدین همیشه مواجه است و نیاز به پردازش داده‌های با حجم بسیار زیاد و هزینه کافی مستر (ارباب) دارند. تعداد زیادی از این سازمان‌ها سیستم‌های غیر رسمی خود را برای غلبه بر این موضوع توسعه داده‌اند. به عنوان مثال، گوگل، نگاشت کاهش و سیستم فایل گوگل را توسعه داده است. همچنین یک سیستم DBMS به نام بیگ تیبل (جدول بزرگ) نیز ساخته شده است. امکان جستجو در میلیون‌ها صفحه نیز وجود دارد و نتایج در آن به صورت میلی ثانیه یا کمتر به منظور کمک به الگوریتم‌هایی که هر کدام از سرویس‌های جستجو را در یک چارچوب نگاشت کاهش گوگل به ارمغان می‌آورند برگردانده می‌شوند [1]. این یک مسئله چالش برانگیز در تحلیل داده‌های بزرگ نیز است. کلان داده‌ها برای کار کردن بسیار بزرگ هستند و بنابراین یک کار بزرگ برای تجزیه و تحلیل کلان داده‌ها انجام می‌شود. تکنولوژی‌های موجود در تجزیه و تحلیل کلان داده‌ها به سرعت در حال تکمیل شدن هستند و به طور قابل توجهی علاقه زیادی به رویکردهای تحلیلی مانند هادوپ، نگاشت کاهش و Hive و توسعه نگاشت کاهش در برابر ارتباط DBMS دارند [2].

استفاده از چارچوب نگاشت کاهش به طور گسترده در جهت مقابله با کلان داده‌ها بسیار موثر بوده است. در چند سال گذشته، نگاشت کاهش به عنوان رایج‌ترین نمونه محاسباتی موازی، تحلیل کلان داده‌ها به نظر می‌رسید [3].

نگاشت کاهش محبوبیت خود را زمانی که با موفقیت توسط گوگل مورد استفاده قرار گرفت به دست آورد. در واقع، این یک ابزار پردازش داده کاوی است و با خطا مقابله می‌کند و قادر است پردازش داده‌های با حجم بسیار زیاد را به موازات گره‌های محاسباتی ارائه دهد [4]. به لطف سادگی آن، مقیاس پذیری و تحمل خطا، و نگاشت کاهش در حال تبدیل شدن در همه جا هستند، و به طور قابل توجهی هر دو صنعت علمی دانشگاه را به دست آورده‌اند. ما می‌توانیم عملکرد بالا را با توقف پردازش واحدهای کوچک به پایان برسانیم و می‌توانیم به صورت

موازی چندین گره را در خوشه اجرا کنیم [5]. در چارچوب نگاشت کاهش، سیستم فایل توزیع شده (DFS) ابتدا داده‌ها را در چندین ماشین تقسیم‌بندی کرده و سپس داده‌ها را به صورت جفت شده (key,value) بیان می‌کنند. چارچوب نگاشت کاهش توابع اصلی ماشین مستر (ارباب) را اجرا می‌کند و ما ممکن است داده‌های ورودی را قبل از نگاشت توابع که پس پردازنده نام دارد و خروجی عملکرد کاهش را پردازش کنند. توابع نگاشت و کاهش به صورت دوتایی ممکن هستند یکبار یا چندین بار اجرا شوند، به این دلیل که به ویژگی‌های برنامه بستگی دارند [6]. هادوپ یک برنامه محبوب منبع باز است که مجموعه‌ای از نگاشت کاهش کلان داده‌ها را تجزیه و تحلیل می‌کند. این یک فایل سیستمی توزیع شده در سطح کاربر است و برای مدیریت منابع ذخیره-سازی در میان خوشه‌ها مورد استفاده قرار می‌گیرد [7]. با این وجود، سیستم سرعت‌های ناخواسته در مجموعه-ای از داده‌ها که کمتر تولید می‌شوند از بین می‌برد. اما سرعت قابل قبولی را با مجموعه‌ای از کلان داده‌ها که تعداد گره‌های محاسباتی را کامل می‌کند تولید می‌کند و زمان اجرای آن را 30 درصد کاهش می‌دهد و آنها را با داده کاوی و سایر روش‌های پردازش مقایسه می‌کند [8].

به طور کلی بخش 2 تکامل نگاشت، کاهش و هادوپ را بیان می‌کند. بخش 3 توضیحات مختصر کلان داده‌ها و مدل برنامه‌نویسی نگاشت کاهش را ارائه می‌دهد. بخش 4 معماری هادوپ را تنظیم می‌کند. بخش 5 روشی علمی از تکنولوژی نگاشت کاهش و هادوپ که ترکیبی از عملکرد نگاشت و کاهش هادوپ است را ارائه می‌دهد.

2. کارهای مرتبط

کلان داده‌ها به اشکال مختلف مجموعه‌ای از داده‌های بزرگ اشاره دارد و این کلان داده‌ها نیاز به سیستم‌های محاسباتی خاصی دارند تا تحلیل شوند. برای تجزیه و تحلیل کلان داده‌ها کارهای زیادی مورد نیاز است. اما، امروزه برای تجزیه و تحلیل چنین کلان داده‌هایی مسائل چالش برانگیز نیز وجود دارد. چارچوب نگاشت کاهش به تازگی توجه زیادی را برای چنین داده‌های گسترده‌ای را به کار می‌برد. نگاشت کاهش یک مدل برنامه‌نویسی و پیاده‌سازی مرتبط با پردازش و تولید مجموعه کلان داده‌ها می‌باشد و به طیف گسترده‌ای از وظایف در دنیای واقعی پاسخ می‌دهد [9]. نگاشت کاهش نمونه‌ای از ویژگی برنامه‌نویسی موازی را به سادگی ارائه می‌دهد. در عین حال، متوازن کننده و ظرفیت تحمل‌پذیری خطا به همراه این ویژگی‌ها ارائه می‌شود [10]. سیستم فایل گوگل (GFS) معمولا تحت عنوان یک سیستم نگاشت کاهش داده‌های توزیع شده را به صورت کارآمد و با

قابلیت اطمینان ذخیره می‌کند و برنامه‌های کاربردی را در یک سیستم پایگاه داده بزرگ را که مورد نیاز است ارائه می‌دهد [11]. نگاشت کاهش از طریق عنصر اولیه نگاشت و کاهش در توابع زبان‌های برنامه کاربردی انجام می‌شود [12]. در حال حاضر برخی از پیاده سازی‌ها قابل دسترس هستند: اشتراک سیستم چند هسته‌ای با حافظه [13]، پردازنده‌های چند هسته‌ای نامتقارن، پردازنده‌های گرافیکی، و خوشه‌ای ماشین‌های شبکه [14]. تکنولوژی نگاشت کاهش گوگل امکان توسعه برنامه‌های توزیع شده در مقیاس وسیع را به شیوه‌ای ساده‌تر و با هزینه کم را فراهم می‌کند. ویژگی اصلی مدل نگاشت کاهش این است که قادر است کلان داده‌ها را به صورت موازی که در میان گره‌های مختلف توزیع شده است پردازش کند [15]. نرم افزار نوین نگاشت کاهشی یک سیستم اختصاصی گوگل است و بنابراین برای استفاده از منابع باز قابل دسترس نیست. محاسبات توزیع شده نظریه عناصر اولیه نگاشت و کاهش را ساده می‌کند، سپس به زیرساخت عملکرد مورد نظر که غیربدیهی است دسترسی پیدا می‌کند [16]. یک زیرساخت کلیدی دارای نگاشت کاهش گوگل، سیستم فایل توزیع شده است و با قابلیت اطمینان بالا به داده‌ها دسترسی پیدا می‌کند [9]. با ترکیب روش زمانبندی نگاشت کاهش و سیستم فایل توزیع شده، می‌توان به راحتی به محاسبات توزیع شده به صورت موازی که بیش از هزاران گره محاسباتی دارد دست یافت؛ و پردازش داده‌ها را در مقیاس ترابایت و پتابایت و همچنین قابلیت اطمینان و بهینه‌سازی سیستم توزیع شده را می‌توان بهبود داد. ابزار نگاشت کاهش در بهینه‌سازی داده‌ها بسیار کارآیی دارد و دارای قابلیت اطمینان نیز است به این دلیل که زمان دسترسی به داده‌ها یا بارگیری از آنها را 50 تا کاهش می‌دهد [16]. گوگل اولین روش تکنیک نگاشت کاهش را تعمیم می‌دهد [17]. تکنولوژی نگاشت کاهش که اخیرا معرفی شده است از جامعه علمی نشات می‌گیرد و کلان داده‌های بزرگ را تجزیه و تحلیل می‌کند [18]. هادوپ یک برنامه منبع باز از مدل برنامه‌نویسی نگاشت کاهش است و به سیستم فایل توزیع شده هادوپ (HDFS) متکی است. اما سیستم فایل گوگل (GFS) وابسته نیست. HDFS بلوک‌های داده‌ای را با قابلیت اطمینان بالا در گره‌های مختلف قرار می‌دهد و آن‌ها را کپی می‌کند و سپس محاسبات را بعد از هادوپ در این گره‌ها انجام می‌دهد. HDFS شبیه به سیستم‌های دیگر است اما طوری طراحی شده است که در برابر خطا بسیار مقاوم است. سیستم فایل توزیع شده (DFS) هیچ سخت افزار بالایی ندارد و می‌تواند در رایانه‌ها و نرم افزارها اجرا شود. همچنین مقیاس‌پذیر نیز است و یکی از اهداف اصلی طراحی در اجرا است. همانطور که مشخص شد

HDFS مستقل از هرگونه سیستم عامل سخت افزار و نرم افزار است، بنابراین در سیستم‌های ناهمگن به راحتی قابل حمل هستند [19]. دستاورد بزرگی که توسط نگاشت کاهش حاصل شده است باعث شبیه سازی هادوپ که یک برنامه منبع باز می‌باشد شده است. هادوپ یک چارچوب منبع باز است که نگاشت کاهش را اجرا می‌کند [20]. این یک مدل برنامه‌نویسی موازی است که از یک موتور نگاشت کاهش و یک سیستم فایل که سطح کاربر را مدیریت می‌کند و در میان منابع ذخیره‌سازی خوشه تشکیل شده است [9]. حمل و نقل سراسری سیستم عامل‌های مختلف-لینوکس، Mac OS/X، FreeBSD، سولاریس و ویندوز- هر دو در جاوا نوشته شده‌اند و فقط نیاز به سخت افزار کالا دارند.

3. اهمیت کلان داده‌ها

سازمان‌ها باید سیستم عامل محاسباتی تحقیقاتی خود را برای بهبود بخشیدن مقادیر کامل کلان داده‌ها ایجاد کنند. این کار کاربران را قادر می‌سازد تا از ساختار تجزیه و تحلیل کلان داده‌ها برای استخراج داده‌های مفید که به راحتی قابل کشف هستند را استفاده کنند. اهمیت کلان داده‌ها را می‌توان به صورت زیر توصیف کرد [21]:

1) کلان داده‌ها باعث انگیزه در یک اصطلاح می‌شوند.

2) این افزایش و مشهوری از هر دو کاربر تجارت و صنعت فناوری اطلاعات به دست می‌آیند.

3) از دیدگاه تجزیه و تحلیل هنوز هم تراکم کاری و راه حل‌های مدیریتی که قبلا نمی‌توانستند از هزینه/یا محدودیت‌ها پشتیبانی کنند نشان داده شده‌اند.

4) راه حل‌ها قادر هستند تصمیم‌گیری هوشمندتری را که زمان بیشتری را برای تحلیل تکنولوژی و محصولات صرف کنند ارائه دهند.

5) تجزیه و تحلیل داده‌ها در چندین ساختار تصمیم‌گیری‌های هوشمندانه‌ای را می‌توانند اتخاذ کنند. تا به امروز، این نوع داده‌ها برای پردازش‌های پیچیده از تجزیه و تحلیل سنتی تکنولوژی‌های پردازش استفاده می‌کرده است.

6) تصمیم‌گیری‌های سریع قابلیت فعال بودن را دارند به این دلیل که راه حل‌های کلان داده‌ها از تجزیه و تحلیل سریع داده‌های دقیق با حجم بالا پشتیبانی می‌کنند.

7) در نظر گرفتن زمان سریع امکان پذیر است به این دلیل که سازمان‌ها می‌توانند داده‌های خارج از انبار داده-های سازمانی را پردازش و تجزیه و تحلیل کنند.

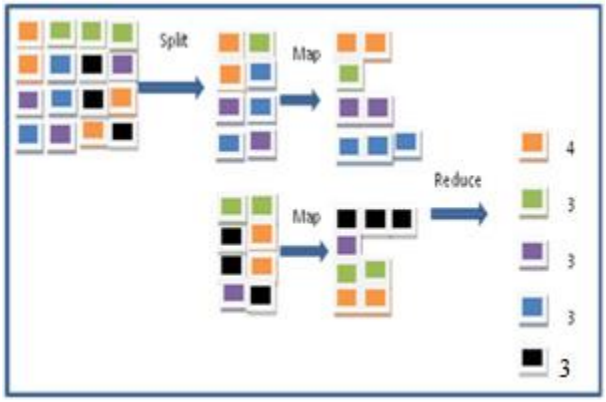
برنامه‌نویسان از مدل برنامه‌نویسی نگاشت کاهش برای بازیابی اطلاعات از کلان داده‌ها استفاده می‌کنند. ویژگی-های اصلی و مسائل مربوط به تحویل انواع مختلف مجموعه‌ای از کلان داده‌ها در جدول زیر خلاصه شده هستند. داده‌هایی که درباره تکنولوژی کلان داده‌ها هستند می‌توانند به حل آنها کمک کنند [22].

جدول 1: خلاصه‌ای از ویژگی‌های اصلی، چالش‌ها و پاسخ‌های تکنولوژی مربوط به تحویل انواع مختلف کلان داده‌ها در زیر نشان داده شده است

جنبه	ویژگی‌ها	پاسخ به پرسش‌ها و مهارت‌ها
حجم صدا	میزان داده‌های تولید شده در سال-های گذشته به شدت افزایش یافته است. با این حال، در عمل جنبه چالش برانگیز کمتری دارد.	اینترنت باعث افزایش قابل ملاحظه‌ای در تولید داده‌های سراسری شده است. پاسخ به این وضعیت از طریق تعمیم راه حل‌ها براساس ابر بوده است. رویکرد پایگاه داده noSQL یک پاسخ برای ذخیره کوئری و حجم زیادی از داده‌ها به صورت توزیع شده در نظر گرفته است.
سرعت	تولید داده‌های با سرعت بالا در حال افزایش است و داده‌های تولید شده باید در فریم‌های کوتاه‌تری جمع آوری شوند.	میلیون‌ها دستگاه روزانه به صورت متصل (گوشی‌های هوشمند) اضافه می‌شوند و نه تنها باعث افزایش حجم بلکه سرعت نیز می‌شود. برای دستیابی به یک مزیت رقابتی، شرکت‌های جهانی سیستم‌های پردازش داده را به عنوان یک نیاز ضروری در نظر گرفتند.
گونه	انفجار فرمت‌های داده‌ای که با داده-های ساختار متن منابع داده‌ای بسیار متفاوت است وجود دارد.	روش‌های جاری که برای جمع‌آوری و تجزیه و تحلیل داده‌های غیر ساخت یافته یا نیمه ساخت یافته به کار می‌روند، کاملا مخالف شیوه مدل رابطه‌ای کوئری هستند. این واقعیت منجر به تکامل انواع جدیدی از ذخیره داده شده است و قادر است از مدل‌های انعطاف پذیر داده پشتیبانی کند.
مقدار	تا همین اواخر، تمرکز بیشتر بر روی ثبت حجم زیادی از داده‌ها بود با این حال نگرانی عمده‌ای در نحوه تسخیر آنها وجود داشت.	تکنولوژی‌های کلان داده‌ها باعث ایجاد، و استفاده کردن از حجم زیادی از داده‌ها شده است. در عمل، زمانی که داده‌های ناقص به داده‌های حاوی مقادیر که دارای چالش‌هایی هستند تبدیل می‌شوند می‌توانند در تصمیم‌گیری‌ها و یا سایر شرایط تجارت مورد استفاده قرار گیرند.

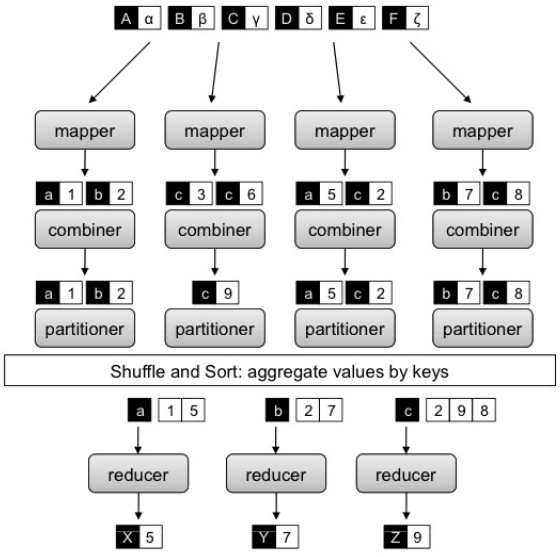
3.1 مدل برنامه‌نویسی نگاشت کاهش

نگاشت کاهش طوری طراحی شده است که برنامه‌نویسان بتوانند از آن به جای کاربران تجاری استفاده کنند. این یک مدل برنامه‌نویسی است، نه یک زبان برنامه‌نویسی. این مدل برنامه‌نویسی برای راحتی، عملکرد و کنترل کلان داده‌ها طراحی شده است. مراحل مربوط به کار نگاشت کاهش را می‌توان به صورت زیر نشان داد:



شکل : مراحل نگاشت کاهش به منظور پردازش پایگاه داده

برنامه‌های کاربردی که شامل نمایه‌سازی و جستجو، تجزیه و تحلیل نمودار، و متن، یادگیری ماشین، تبدیل داده‌ها است با استفاده از SQL استاندارد DBMSهایی که آسان نیستند را ایجاد می‌کند. در چنین مناطقی ماهیت رویه نگاشت کاهش به راحتی برای برنامه‌نویسان ماهر قابل درک است. همچنین این مزیت را هم دارد که توسعه‌دهندگان مجبور نیستند از اجرای محاسبات موازی استفاده کنند-و به صورت شفاف در سیستم بکار گرفته می‌شوند. اگرچه نگاشت کاهش برای برنامه‌نویسان طراحی شده است، با این حال برنامه‌های غیربرنامه‌نویسی می‌توانند از برنامه‌های نگاشت کاهش از پیش ساخته شده و توابع کتابخانه‌ای بهره ببرند [3]. معماری نگاشت کاهش را می‌توان به صورت زیر نشان داد:



شکل 2: نگاشت کاهش به همراه ترکیب کننده، و تقسیم کننده

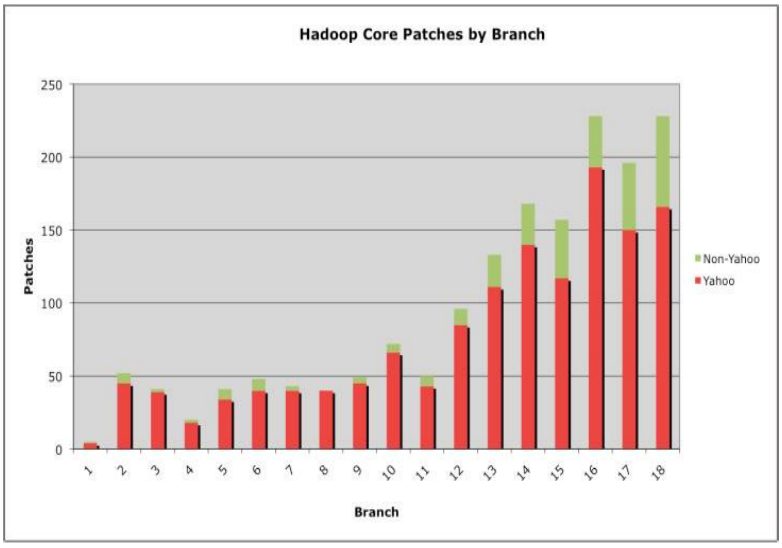
جدول 2: توصیف نگاشت کننده‌ها، کاهنده‌ها، تقسیم کننده‌ها و ترکیب کننده‌ها

نیاز به تولید تعداد دلخواهی از جفت‌های میانی	نگاشت کننده‌ها
اعمال تمام مقادیر میانی که مرتبط با کلید میانبر است.	کاهنده‌ها
وظیفه اصلی آن تقسیم فضای کلیدی میانجی است، سپس برای جفت کردن مقادیر میانی به کاهنده‌ها تخصیص داده می‌شود.	تقسیم کننده‌ها
ترکیب کننده‌ها یک مسئله بهینه‌سازی هستند (اختیاری). قبل از انجام فاز بهم زدن و مرتب‌سازی، اجازه می‌دهد که داده‌های محلی جمع‌آوری شوند. اساسا، ترکیب کننده‌ها برای ذخیره مورد استفاده قرار می‌گیرند، به عنوان مثال برنامه شمارش لغات	ترکیب کننده‌ها

برنامه‌های نگاشت کاهش معمولا در جاوا نوشته می‌شوند. آنها همچنین می‌توانند به زبان‌های دیگر مانند C++، پایتون، روبی و غیره کدگذاری شوند. این برنامه‌ها ممکن است داده‌های ذخیره شده در فایل‌ها و سیستم‌های پایگاه داده را پردازش کنند. به عنوان مثال در گوگل، نگاشت کاهش در بالای سیستم فایل گوگل (GFS) اجرا می‌شود.

4. تنظیم مسائل

هادوپ: یاهو! اولین عامل اصلی در سال 2006 شده است



شکل 3: عامل اصلی هادوپ

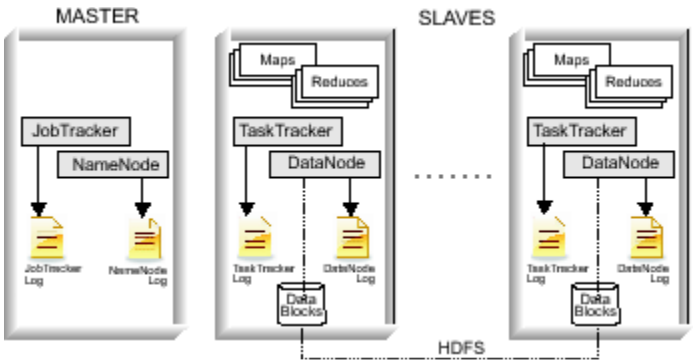
آپاچی هادوپ شامل چندین مولفه است. مواردی که در یک پایگاه داده و پردازش تحلیلی مورد توجه هستند عبارتند از [23]:

سیستم فایل توزیع شده هادوپ (HDFS)، نگاشت کاهش، Pig، Hive، Hbase، اسکوپ

HDFS می‌تواند یک منبع یا سیستم فایل برای برنامه‌های نگاشت کاهش باشد. این بهترین درخواست برای فایل‌های بسیار بزرگ می‌باشد. استفاده از داده‌های تکراری برای دسترسی به داده‌ها در HDFS امکان‌پذیر است. اما این باعث افزایش ذخیره‌سازی مورد نیاز برای مقابله با داده‌ها می‌شود. چارچوب نگاشت کاهش هادوپ به توزیع نگاشت فرآیند کمک می‌کند بنابراین داده‌های HDFS به برنامه محلی نیاز پیدا می‌کنند. برای پردازش، تمام فایل‌های خروجی ایجاد می‌شوند و توسط فرآیند نگاشت و برنامه کاهش بیشتر حرکات و دسترسی به داده-های گره داخلی را انجام می‌شوند. در زمان اجرا، هر دو برنامه نگاشت و کاهش ، داده‌های انجام شده را به سیستم فایل محلی ارسال می‌کنند تا بتوانند از سربارگذاری تکرار HDFS جلوگیری کنند. HDFS از خوانندگان متعدد و یک نویسنده (MROW) پشتیبانی می‌کند. مکانیسم شاخص در HDFS قابل دسترس نیست، از این رو، برای خواندن برنامه‌ها به منظور اسکن محتوای کامل یک فایل مناسب و بهتر است. در HDFS، محل واقعی داده‌ها برای برنامه‌ها و نرم افزار خارجی شفاف است.

معماری HDFS

معماری HDFS شامل گره‌های ارباب/برده است و گره ارباب گره نام و گره برده گره داده نام دارد. HDFS فقط شامل تک گره نام ارباب و تعداد زیادی گره داده (برده) در میان خوشه و معمولا در هر گره است. HDFS یک فضای نامی (شبیه بسته‌ای در جاوا) را برای ذخیره داده‌های کاربران اختصاص می‌دهد. یک فایل ممکن است به یک یا چند بلوک داده تقسیم شود و این بلوک‌های داده ممکن است در مجموعه‌ای از گره‌های داده نگهداری شوند. گره نام اطلاعات ضروری فراداده را که در مورد نحوه اتصال بلوک‌ها به یکدیگر است و نحوه ذخیره بلوک‌ها در گره نام را نشان می‌دهد. نیازهایی که توسط کلاینت برای خواندن و نوشتن سیستم فایل ساخته شده بودند به صورت مستقیم توسط گره داده پردازش شده بودند، در حالی که عملیات فضای نام مانند باز کردن، بستن و تغییر نام دایرکتوری‌ها توسط گره‌های نام انجام می‌شود. مسئولیت‌های گره نام و گره‌های داده باید مربوط به فعالیت‌های خاصی مانند ایجاد بلوک داده، تکرار و حذف باشد [20]. معماری HDFS (سیستم فایل توزیع شده هادوپ) در زیر نشان داده شده است [23]:



شکل 4: یک مدل ساده از چندین گره خوشه هادوپ

نمونه‌ای از HDFS که دارای یک ماشین اختصاصی است فقط گره نام را اجرا می‌کند. معمولا هر یک از ماشین-ها در خوشه یک نمونه از نرم افزار گره داده را اجرا می‌کنند، و این معماری به شما اجازه می‌دهد که چندین گره داده را در یک ماشین اجرا کنید. گره نام مربوط به محل ذخیره‌سازی فراداده و کنترل است، در صورتی که گره نام مربوط به موارد ذکر شده نباشد اطلاعات کاربر هرگز مدیریت نمی‌شود. گره نام از یک نوع خاص ورود به سیستم، به نام EditLog، و برای پایداری فراداده استفاده می‌کند.

توسعه هادوپ

اگرچه هادوپ یک پیاده‌سازی خالص در جاوا است، با این حال ما می‌توانیم از آن به دو روش مختلف استفاده کنیم. ما می‌توانیم از یک جریان موجود در API یا از لوله‌های هادوپ استفاده کنیم. گزینه دوم این امکان را فراهم می‌کند تا برنامه‌های هادوپ با استفاده از C++ ایجاد شوند. از این رو، ما روی سوابق تمرکز خواهیم کرد. هدف اصلی طراحی هادوپ فراهم کردن ذخیره‌سازی و ارتباطات موجود در بسیاری از ماشین‌های همگن است. برنامه‌نویسان لینوکس را به عنوان سیستم عامل اولیه خود برای توسعه و آزمایش انتخاب کردند؛ از این رو، اگر علاقمند به کار با هادوپ در ویندوز باشید، لازم است نرم افزار جداگانه‌ای را برای تقلید محیط پوسته نصب کنید. هادوپ می‌تواند به سه روش مختلف بستگی به نحوه توزیع فرآیندها اجرا شود [24]:

- حالت مستقل: این حالت به صورت پیش فرض با هادوپ ارائه می‌شود. همه این موارد در یک فایل جاوا اجرا می-شوند.
- حالت شبه کد توزیع شده: در اینجا، هادوپ برای اجرای بر روی یک ماشین واحد، با انواع دمون‌های هادوپ به عنوان فرآیندهای مختلف جاوا پیکربندی می‌شود.

- حالت خوشه‌ای یا توزیع شده: در اینجا، یک دستگاه در خوشه به عنوان گره نام نامگذاری می‌شود و دستگاه دیگری به عنوان Job Tracker تعیین می‌شود. فقط یک گره نام در هر خوشه قرار می‌گیرد و فضای نام، فایل سیستمی فراداده و کنترل دسترسی را مدیریت می‌کند. دومین گره نام نیز می‌تواند به قابلیت تحمل‌پذیری خطا دستیابی پیدا کند. بقیه دستگاه‌های موجود در خوشه هر دو به عنوان گره نام و Task Tracker عمل می‌کنند. گره داده، داده‌های سیستم را نگه می‌دارد؛ و هر گره داده حافظه محلی را ذخیره می‌کند یا هارددیسک محلی آن را مدیریت می‌کند. Task Tracker عملیات نگاشت و کاهش را انجام می‌دهد.

5. آزمایشات

نوشتن برنامه کاربردی نگاشت کاهش هادوپ

بهترین راه برای درک و کار کردن با هادوپ این است که از طریق فرآیند نوشتن برنامه، نگاشت کاهش هادوپ پیاده سازی شود. ما با یک برنامه ساده نگاشت کاهش کار می‌کنیم و این برنامه می‌تواند بسیاری از رشته‌ها را معکوس کند. مثالی که در زیر ارائه شده است از طریق تعدادی از مراحل ابتدا تمام داده‌ها را به گره‌های مختلف تقسیم می‌کند، عملیات را انجام می‌دهد تا داده‌ها معکوس شوند و نتیجه رشته‌ها را باهم مرتبط می‌سازد و سپس نتایج را تولید می‌کند. این نرم افزار فرصتی را برای بررسی تمام مفاهیم اصلی هادوپ فراهم می‌کند. ابتدا، ما در مراحل زیر نگاهی به اعلان و وارد کردن بسته می‌اندازیم. بسته در کلاس رشته‌ای com.javaworld.mapreduce قرار دارد. این را می‌توان در دو مجموعه به صورت زیر نشان داد:

First set of Imports

```
package com.javaworld.mapreduce;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.StringTokenizer;
import java.io.*;
import java.net.*;
import java.util.regex.MatchResult;
```

Second set of Imports

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.mapredu.JobClient;
import org.apache.hadoop.mapredu.JobConf;
import org.apache.hadoop.mapredu.MapredBase;
import org.apache.hadoop.mapredu.Mapper;
import org.apache.hadoop.mapredu.OutputCollector;
import org.apache.hadoop.mapredu.Reducer;
import org.apache.hadoop.mapredu.Reporter;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
```

اولین مجموعه اعلان‌ها برای کلاس‌های استاندارد جاوا است و دومین مجموعه برای پیاده‌سازی نگاشت کاهش است. بازخوانی کلاس با توسعه org.apache.hadoop.conf.Configured و اجرای رابط org.apache.hadoop.until.Tool آغاز می‌شود.

نگاشت و کاهش

حالا شما می‌توانید به پیاده‌سازی نگاشت کاهش واقعی پرش پیدا کنید. دو کلاس داخلی عبارتند از: نگاشت: شامل عملکردی برای پردازش جفت‌های کلیدی ورودی به منظور تولید جفت خروجی کلیدی است.

Map class

```
public static class Map extends MapredBase
      implements   Mapper<LongWritable,   Text,   Text,
Text>
{
 private Text inpText   = new Text();
 private Text reverText = new Text();
 public void map(LongWritable key, Text inputs,
 OutputCollector<Text, Text> output,
 Reporter reporter) throws IOException {
 String inputString = inputs.toString();
 int length = inputString.length();
 StringBuffer reverse = new StringBuffer();
 for(int i=length-1; i>=0; i--)
     {
       reverse.append(inputString.charAt(i));    }
 inputText.set(inpString);
reverseText.set(reverse.toString());
output.collect(inpText,reverText);
}   }
```

Now, it is required to combine all such outputs.
This job is done with the reduce()method of the Reduce class as shown in the steps below:

Reduc.redu()

```
public static class Reduc extends MapRedBase
      implements Reducer<Text, Text, Text, Text> {
      public void reduc(Text key, Iterator<Text> values,
                  OutputCollector<Text, Text> output,
                  Reporter reporter) throws IOException {
        while (values.hasNext()) {
          output.collect(key, values.next());
        }   }       }
```

Reduc: Includes functionality for collecting output from parallel map processing and outputting that collected data.

6. مشارکت ما

به تازگی، در برخی از آزمایشات کشف شده است که برنامه‌های کاربردی‌ای که از هادوپ استفاده می‌کرده‌اند در مقایسه با برنامه‌های مشابه که از پایگاه داده‌های موازی استفاده می‌کرده‌اند کارایی کمتری دارند. هدف اصلی ما این است که بهینه‌سازی HDFS و به طور قابل توجهی عملکرد کلی چارچوب نگاشت کاهش را که باعث افزایش کارایی کل برنامه‌های نگاشت کاهش در هادوپ می‌شود را فراهم کنیم. ممکن است نتیجه نهایی نگاشت کاهش در مقابل پایگاه داده موازی هیچ تغییری نداشته باشد، با این حال رویکرد جدید هادوپ و نگاشت کاهش مطمئنا امکان مقایسه دقیق‌تر مدل‌های برنامه‌نویسی را فراهم خواهد کرد. اگرچه هادوپ قابلیت‌های داخلی را برای نمایش نگاشت و کاهش فراهم می‌کند، با این حال هیچ ابزار ساخته شده‌ای برای تنظیم چارچوب، که بتواند موانع عملکرد را بدون انعطاف نگه دارد وجود ندارد. این مقاله تعاملات بین هادوپ و ذخیره‌سازی را بازیابی می-

کند. در اینجا، ما بسیاری از موانع عملکرد را که به طور مستقیم به کد برنامه مربوط نمی‌شود (یا سبک برنامه-نویسی نگاشت کاهش)، بلکه به واسط زمانبندی کار و سیستم‌های توزیع شده تمام برنامه‌های هادوپ مربوط می‌شود را توضیح دادیم. HDFS همزمان می‌تواند به طور قابل توجهی با استفاده از برنامه ورودی/خروجی زمانبندی شود و می‌تواند حفظ قابلیت حمل و نقل را بهبود ببخشد. بهبودهای بیشتر می‌توانند با کاهش پراکندگی و حافظه نهان و کاهش هزینه قابل حمل انجام شوند. هادوپ از قابلیت انتقال برای پشتیبانی کاربران و کاهش پیچیدگی نصب و راه اندازی استفاده می‌کند. این نتایج نمونه‌ای گسترده از محاسبات موازی است.

7. نتیجه‌گیری

کلان داده‌ها و فناوری‌ها می‌توانند مزایای قابل توجهی را برای کسب و کار به ارمغان بیاورند. اما استفاده‌های فوق‌العاده از این تکنولوژی‌ها برای یک سازمان و برای کنترل مجموعه‌های ناهمگن داده‌ها به منظور بررسی بیشتر بسیار دشوار است. اثرات چندگانه استفاده از کلان داده‌ها وجود دارند. برای روبرو شدن با رقابت‌های شدید شرکت‌های خصوصی، آنها از یک پتانسیل بزرگ پشتیبانی کردند. بعضی از جنبه‌ها نیاز به پیگیری دارند تا ما بتوانیم نتایج را به موقع از کلان داده‌ها بدست آوریم، به این دلیل که استفاده دقیق از کلان داده‌ها می‌تواند به گسترش، نوسازی و اثربخشی کل بخش‌ها منجر شود. برای اینکه بتوانید مزایای کلان داده‌ها را استخراج کنید، این بسیار مهم است که بدانید مدیریت و استفاده مجدد از منابع داده از جمله تاثیر داده‌های کانتری و ایجاد برنامه‌های کاربردی و مورد اطمینان چگونه حاصل می‌شوند. مهم این است که بهترین روش را به منظور استفاده از فیلتر کردن/و یا تجزیه و تحلیل داده‌ها ارزیابی کنید. برای پردازش تحلیلی بهینه‌سازی، هادوپ با نگاشت کاهش مورد استفاده قرار می‌گیرد. در این مقاله، ما اصول برنامه‌نویسی نگاشت کاهش را با چارچوب هادوپ منبع باز ارائه کردیم. این یک چارچوب فوق‌العاده از سرعت پردازش مقادیر زیادی از داده‌های هادوپ است که از طریق پروسه‌های توزیع شده و پاسخ‌های بسیار سریع ارائه می‌شود. این می‌تواند برای رفع نیازهای مختلف توسعه مورد پذیرش قرار گیرد و می‌تواند با افزایش تعداد گره‌های موجود به منظور پردازش مقیاس‌پذیری شود. قابلیت امتداد و سادگی چارچوب، و متمایزکننده‌های کلیدی که یک ابزار امیدوار کننده است را برای پردازش داده‌ها ایجاد می‌کند.

References

1. J R Swedlow, G Zanetti, C Best. Channeling the data deluge. Nature Methods, 2011, 8: 463-465
2. G C Fox, S H Bae, et al. Parallel Data Mining from Multicore to Cloudy Grids. High Performance Computing and Grids workshop, 2008
3. Maitrey S, Jha. An Integrated Approach for CURE Clustering using Map-Reduce Technique. In Proceedings of Elsevier, ISBN 978-81-910691-6-3,2nd August 2013].
4. D. DeWitt and M. Stonebraker. MapReduce: A major step backwards. The Database Column, 1, 2008.
5. Apache. Apache Hadoop. http://hadoop.apache.org, 2010.
6. Y. Kim and K. Shim. Parallel top-k similarity join algorithms using MapReduce. In ICDE, 2012.
7. Jeffrey Shafer, Scott Rixner, and Alan L. Cox. The Hadoop Distributed Filesystem: Balancing Portability and Performance. DOP is March 30,2010.
8. Moturi, Maiyo. Use of MapReduce for Data Mining and Data Optimization on a Web Portal.Published in International Journal of Computer Applications (0975 – 8887) Volume 56– No.7, October 2012].
9. Jeffrey Dean et al. Mapreduce: Simplified data processing on large clusters. In Proceedings of the 6th USENIX OSDI, pages 137–150, 2004.
10. S. Ghemawat et al. The google file system. ACM SIGOPS Operating Systems Review, 37(5):29–43, 2003.
11. C. Ranger et al. Evaluating mapreduce for multi-core and multiprocessor systems. In Proceedings of the 2007 IEEE HPCA, pages 13–24, 2007.
12. Yoo, R. M., Romano, A.K. and Kozyrakis, C. 2009. Phoenix Rebirth: "Scalable MapReduce on a Large-Scale Shared-Memory System". Proceedings of the 2009 IEEE International Symposium on Workload Characterization, pp. 198-207.
13. Rafique, Mustafa. M. 2009. "Supporting MapReduce on Large-Scale Asymmetric Multi-Core Clusters". ACM SIGOPS Operating Systems Review, Vol. 43, 2, pp. 25-34.
14. J. Dean et al. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1):107– 113, 2008.
15. Kyong, Lee, Choi, Chung, Moon. Parallel Data Processing with MapReduce: A Survey. Published in SIGMOD Record, December 2011 (Vol. 40, No. 4).
16. B. Panda, J. Herbach, S. Basu, and R. J. Bayardo, "Planet: Massively parallel learning of tree ensembles with mapreduce,"PVLDB, vol. 2, no. 2, pp. 1426–1437, 2009.
17. J. Dean et al. MapReduce: a flexible data processing tool. Communications of the ACM, 53(1):72–77, 2010.
18. Jaliya Ekanayake, Shrideep Pallickara, and Geoffrey Fox, MapReduce for Data Intensive Scientific Analyses. In Fourth IEEE International Conference on eScience (978-0-7695-3535-7/08) eScience, 2008.
19. "GFS, MapReduce, and Hadoop" (Geeking with Greg, June 2006).
20. http://hadoop.apache.org/common/docs/current/hdfs design.html, 2009.

21. MapReduce and the Data Scientist Colin White, BI Research January 2012.
22. Big Data: A New World of Opportunities. NESSI White Paper, December 2012
23. Tomasz Wiktor Wlodarczyk, Yi Han, Chunming Rong: Performance Analysis of Hadoop for Query Processing. AINA Workshops 2011:507-513.
24. W. Tantisiriroj, S. Patil, and G. Gibson. Data-intensive file systems for internet services: A rose by any other name. Technical report, Carnegie Mellon University, 2008.