****

**مقدمه ای بر YARN و بررسی هادوپ**

**چکیده**

تجزیه و تحلیل کلان داده­ها و مقدار زیادی از داده­ها در سال­های اخیر به یک بینش جدید تبدیل شده­اند. روز به روز داده­ها با سرعت چشمگیری در حال رشد هستند. یکی از تکنولوژی­های کارآمد و موثر که با کلان داده­ها برخورد می­کند هادوپ است و در این مقاله آنها را مورد بحث و بررسی قرار می­دهیم. هادوپ، برای پردازش از کلان داده­ها، که از مدل زمانبندی نگاشت کاهش است استفاده می­کند. هادوپ از زمانبندی­های مختلفی برای اجرای موازین شغلی استفاده می­کند. زمانبندی به طور پیش فرض (First In First Out) زمانبندی FIFO است. زمانبندی با گزینه­های مختلف پیشدستی و غیرپیشدستی توسعه پیدا می­کند. نگاشت کاهش از محدودیت­هایی که دست پیدا کرده است عبور می­کند. بنابراین برای غلبه بر محدودیت­های نگاشت کاهش، نسل بعدی نگاشت کاهش با عنوان YARN توسعه پیدا می­کند (منابع انتقال دهنده دیگر). بنابراین، این مقاله یک بررسی از هادوپ، و استفاده از چند روش زمانبندی و به طور مختصر مقدمه­ای بر YARN را ارائه می­دهد.

**کلمات کلیدی:** هادوپ، HDFS، نگاشت کاهش، زمانبندی، YARN

**1- مقدمه**

در حال حاضر سناریو با اینترنت چیزهای زیادی تولید می­کند و به طور عمده برای هوش تجاری تجزیه و تحلیل می­شوند. در منابع مختلف کلان داده­ها سایت­های شبکه­های اجتماعی، حسگرها، تراکنش برنامه­های کاربردی سازمانی/پایگاه داده­ها، دستگاه­های تلفن همراه، داده­های تولید شده، مقدار زیادی داده­های تولید شده از فیلم­های با کیفیت بالا و منابع بسیاری وجود دارد. برخی از منابع این داده­ها ارزش حیاتی دارند و برای توسعه کسب و کار بسیار مفید می­باشند. بنابراین یک سوال مطرح می­شود چگونه چنین مقادیر عظیمی از داده­ها می­توانند استفاده کنند؟ علاوه بر این، هیچگونه توقف داده­ای در آن وجود ندارد. درخواست­های زیادی برای بهبود تکنیک­های مدیریت کلان داده­ها وجود دارد. پردازش کلان داده­ها را می­توان با استفاده از محاسبات توزیع شده و مکانیسم­های پردازش موازی انجام داد. هادوپ ]1[ یک سیستم عامل محاسباتی توزیع شده در جاوا است که شامل ویژگی­هایی شبیه به سیستم فایلی گوگل و نمونه­ای از برنامه­نویسی نگاشت کاهش است. چارچوب هادوپ توسعه­دهندگان را از مسیر حل مسائل مسدود می­کند و به آنها این امکان را می­دهد که روی مسائل محاسباتی خود تمرکز کنند و مسائل مربوط به راه­اندازی چارچوب را به طور ذاتی انجام دهند.

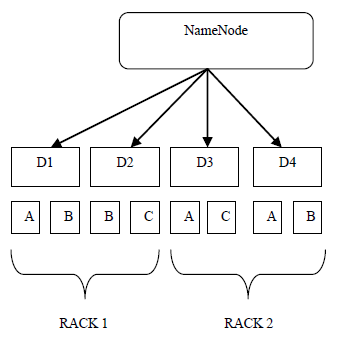
در بخش دوم ما در مورد جزئیات دو مورد مهم هادوپ HDFS و نگاشت کاهش بحث می­کنیم. در بخش سوم ما درباره برنامه­های هادوپ بحث می­کنیم. بخش چهارم برخی از انواع زمانبندی مورد استفاده در هادوپ و بهبود زمانبندی را مورد بحث قرار می­دهیم. بخش پنجم بیشتر در مورد جنبه­های فنی هادوپ بحث می­کنیم. بخش ششم روی نمونه نسل بعدی هادوپ YARN متمرکز می­شویم. سرانجام در بخش هفتم به منابع هادوپ رجوع می­کنیم.

**2- هادوپ**

هادوپ یک چارچوب طراحی شده از کلان داده­ها است و سیستم­های معمولی کلان داده­ها را می­تواند سازماندهی کند. هادوپ داده­ها را در مجموعه­ای از دستگاه­ها توزیع می­کند. قدرت واقعی هادوپ به صورت مقیاس­پذیری صدها یا هزاران کامپیوتر که هر کدام شامل چندین هسته پردازنده هستند می­باشد. بسیاری از شرکت­های بزرگ معتقدند که ظرف چند سال بیش از نیمی از داده­ها جهان در هادوپ ذخیره خواهند شد ]2[. علاوه بر این، هادوپ با ماشین مجازی ترکیب می­شود و نتایج بیشتری را ارائه می­دهد. هادوپ عمدتا شامل موارد زیر است: 1) سیستم فایل توزیع شده هادوپ (HDFS): سیستم فایل توزیع شده برای دستیابی به فضای ذخیره­سازی و تحمل خطا است و 2) نگاشت کاهش هادوپ یک مدل برنامه­نویسی قدرتمند موازی است که مقدار زیادی داده را از طریق محاسبات توزیع شده در میان خوشه­ها پردازش می­کند.

**A. سیستم فایل توزیع شده هادوپ-HDFS**

سیستم فایل توزیع شده هادوپ ]3[ ]4[ یک سیستم فایل منبع باز است که به طور خاص برای بررسی فایل­های بزرگی نمی­توانند سیستم فایل سنتی را سازماندهی کنند طراحی شده است. مقدار زیادی از داده­ها تقسیم، تکثیر می­شوند و در میان ماشین­های متعدد وجود دارند. تکرار داده­ها، محاسبات را سریع و قابلیت اطمینان را تسهیل می­کنند. به همین دلیل است که سیستم HDFS نیز می­­تواند به عنوان یک سیستم فایل توزیع شده نامگذاری شود و این بدین معنی است که اگر یک کپی از داده­ها خراب باشد یا به طور خاص نتوانسته باشد از گره نام در محل ذخیره­سازی داده­ها استفاده کند از عمل کپی تکراری استفاده می­کند. در حال حاضر این عمل اطمینان حاصل می­کند که بدون هرگونه اختلالی انجام شود.



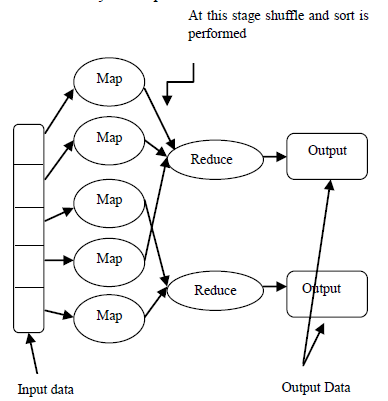
شکل 1: HDFS

HDFS دارای معماری ارباب/برده است. معماری HDFS در شکل 1 نشان داده شده است. در شکل الفبای A,B,C بلوک داده­ها را نشان می­دهد و D عدد مربوط به گره داده را نشان می­دهد. HDFS اکوسیستم با قابلیت تحمل بالا را ارائه می­دهد. تک گره نام همراه با گره داده در یک خوشه معمولی HDFS وجود دارد. گره نام، مسئولیت سرور اصلی فضای نامی سیستم فایل را مدیریت می­کند و دسترسی کلاینت­ها به فایل­ها را کنترل می­کند. فضای نامی ایجاد، حذف و اصلاح فایل­ها توسط کاربران را ثبت می­کند. نگاشت­های گره نام بلوک­های داده و گره­های داده و عملیات فایل­های سیستمی، تغییر نام و بسته شدن فایل­ها و دایرکتوری­ها را مدیریت می­کنند. همه این دستورات براساس گره نام هستند، و گره­های داده عملیاتی را بر روی بلوک­های داده­ای مانند ایجاد، حذف و تکرار انجام می­دهند. اندازه بلوک 64 مگابایت است و به 3 نسخه تقسیم می­شود. دومین نسخه در ردیف محلی در حالی که سومین نسخه در یک ردیف دور ذخیره می­شود. ردیف چیزی جز مجموعه­ای از گره­های داده نیست.

**B. نگاشت کاهش هادوپ**

نگاشت کاهش ]5[ ]6[ یک تکنولوژی بسیار مهم است که توسط گوگل ارائه شده است. نگاشت کاهش یک مدل برنامه­نویسی ساده شده و یک مولفه اصلی هادوپ برای پردازش موازی با میزان داده­های وسیع است. برنامه­نویسان بارگذاری مسائل را راحت به دست می­آورند با این حال به صورت آزادانه روی توسعه نرم افزار تمرکز می­کند. نمودار نگاشت کاهش هادوپ در شکل 2 نشان داده شده است. دو تابع پردازش داده مهم وجود دارند که عبارتند از نگاشت کاهش برنامه­نویسی که شامل نگاشت و کاهش است.

داده­های اصلی فاز نگاشت را به عنوان ورودی معرفی می­کنند و طبق زمانبندی ­ای که توسط برنامه­نویسان انجام شده نتایج را پردازش و تولید می­کنند. برنامه­های نگاشت موازی در یک زمان اجرا می­شوند. اولا، داده­های ورودی به بلوک­های با اندازه ثابت تقسیم می­شوند و برنامه­های نگاشت موازی را اجرا می­کنند. خروجی نگاشت مجموعه­ای از جفت مقادیر کلیدی هستند که هنوز هم دارای خروجی متوسط نیز هستند. این جفت­ها در طی کار کاهش فشرده می­شوند. فقط یک کلید توسط هر برنامه کاهش پذیرفته می­شود و براساس این کلید پردازش انجام خواهد شد. سرانجام خروجی به فرم جفت­های کلیدی درخواهد آمد.



شکل 2: نگاشت کاهش هادوپ

چارچوب نگاشت کاهش هادوپ شامل یک گره ارباب است که به عنوان JobTracker و به عنوان گره­های TaskTrackers کار می­کند. کاربر تعدادی از برنامه­های نگاشت و کاهش را در درون آنها تغییر شکل می­دهد. این برنامه­ها به TaskTrackers تخصیص داده می­شوند. TaskTracker به خوبی اجرای برنامه­ها و پایان آنها را زمانی که تمام برنامه­ها انجام می­شوند رسیدگی می­کند؛ و به کاربر در مورد اتمام کار اخطار می­دهد. HDFS تحمل خطا و قابلیت اطمینان را با ذخیره­سازی و جایگزین کردن ورودی­ها و خروجی­های هادوپ ارائه می­دهد.

به طور کلی چارچوب هادوپ HDFS و نگاشت کاهش است. همچنین اکوسیستم هادوپ شامل پروژه­های مختلفی است که در ]7[ ]8[ مورد بحث قرار گرفته­اند:

* آپاچی Hbase: ستون شی­گرا، ذخیره داده­های کلیدی توزیع شده غیرربطی روی سیستم عامل HDFS اجرا می­شوند. مقیاس بیرونی به طور افقی در خوشه محاسباتی توزیع شده طراحی می­شود.
* آپاچی Hive: یک انبار داده،­ زیرساختی روی هادوپ به منظور ارائه کردن خلاصه­ای از داده­ها و کوئری، و تجزیه و تحلیل مجموعه­ای از داده­ها که روی فایل­های سیستمی ذخیره شده­اند را می­سازند. این انبار داده­ای کوئری بی­درنگ را طراحی نمی­کند، بلکه می­تواند فایل­های متنی، و فایل­های دنباله­دار را پشتیبانی کند.
* آپاچی Pig: این آپاچی مکانیسم موازی با سطوح بالا را برای برنامه­نویسی کارهای نگاشت کاهش به منظور اجرای خوشه­های هادوپ ارائه می­دهد. زبان اسکریپت به عنوان Pig لاتین استفاده می­شود، و زبان جریان داده­ای داده­ها را به صورت روش موازی پردازش می­کند.
* آپاچی Zookeeper: واسط یک برنامه کاربردی است (API) که این امکان را فراهم می­کند که پردازش داده­های توزیع شده در سیستم­های بزرگ به منظور همگام­سازی هر یک از روش­ها است و داده­های سازگار با درخواست­های کلاینت را ارائه می­دهد.
* آپاچی Sqoop: یک ابزار که به طور موثر انتقال داده­های حجیم بین هادوپ و ذخیره داده­های ساخت­یافته مانند پایگاه داده­های ارتباطی طراحی می­کند.
* آپاچی فلوم: یک سرویس توزیع شده که میزان داده­های ورودی را فراهم، جمع­آوری، و انتقال می­دهد.

**3. برنامه­های کاربردی هادوپ**

برنامه­های کاربردی هادوپ به صورت زیر ارائه شده­اند که عبارتند از ]9[ ]10[ ]11[:

* تجزیه و تحلیل انواع مختلف جریان کلیک/و یا ورودی
* تحلیل بازاریابی
* پیشگزینی سفر آنلاین
* کشف و ذخیره­های انرژی
* زیرساخت مدیریت
* بازیابی خطا
* توجه به سلامتی
* انواع مسیرهای مختلف داده شامل موقعیت جغرافیاییی داده، داده­های حسگر و دستگاه، داده­های رسانه اجتماعی است.

**4. زمانبندی کار**

زمانبندی هادوپ از انواع مختلف زمانبندی الگوریتم­ها استفاده می­کند. انواع مختلف هادوپ به طور پیش­فرض برای زمانبندی FIFO مورد استفاده قرار می­گیرد. سپس فیس­بوک و یاهو پس از در نظر گرفتن سعی می­کند در این منطقه به ترتیب ظرفیت زمانبندی و خطای زمانبندی را بیاورد. سپس آنها انواع مختلف هادوپ را بعدا اضافه می­کنند.

**A. زمانبندی پیش فرض هادوپ**

نسخه­های هادوپ به طور سرراست رویکردهایی را به همراه وظایف کاربران مورد استفاده قرار می­دهد. آنها به طور مرتب اجراها را زمانبندی می­کنند و به اصول FIFO ارسال می­کنند (به ترتیب ورودی) ]12[.

در بعضی از زمان­ها، اولویت تخصیص وظایف به منظور فراهم کردن وظایف جدید انسانی است. اولویت زمانبندی کار با انتخاب وظیفه بعدی که بالاترین اولویت را دارد است. این انتخاب به طور پیش فرض روشن نمی­شود بلکه می­تواند در صورت نیاز استفاده کند. اما با زمانبندی پیشگزینی FIFO در طول اولویت پشتیبانی نمی­شود. بنابراین به طور تصادفی یک اجرا وجود دارد که در آن اولویت با آخرین بلوک­بندی با زمانبندی بالا است. اولویت­ها در کوئری FIFO اصلاح کار است که به طور مطمئن کار را انجام می­دهند، با این حال دارای نیازمندی­های مونیتورینگ و مدیریت کار نیز هست. اولویت مسئله با زمانبندی FIFO است و در آن هادوپ خوشه­های ورودی برای هر برنامه اجرا شده­ای تخصیص داده می­شوند. هادوپ دو وظیفه اضافی زمانبندی را که شامل رویکرد مختلف و اشتراک خوشه بین چندین وظیفه به صورت یکجا است را ارائه می­دهد. ظرفیت و شایستگی زمانبندی یک راه پیچیده را برای مدیریت منابع خوشه از میان چندین وظیفه به طور همزمان ارائه می­دهند. این زمانبندی­ها در بخش زیر مورد بحث قرار می­گیرند.

**B. شایستگی زمانبندی**

شایستگی زمانبندی ]13[ یک روش تخصیص منابع کاری است که تمام وظایف موجود در آن به طور یکسان میانگین اشتراک را به دست می­آورند. شایستگی زمانبندی به ارائه اشتراک ظرفیت خوشه بسیار اهمیت می­دهد. اگر فقط یک وظیفه در حال اجرا باشد، در اینصورت یک مزیت برای دستیابی به تمام ظرفیت خوشه وجود دارد. کاربران وظایف بیشتر را ارسال می­کنند، و شیارهایی را در میان هر کاربر که شایستگی زمانبندی خوشه را ارائه می­دهد به اشتراک می­گذارند. زمانبندی شایستگی دارای شایستگی کافی به همراه وظایف کوتاه و بلند در روش است و این امکان را می­دهد که وظایف کوچکتر در زمان مناسب به پایان برسند در حالی که وظایف بزرگ­تر دچار گرسنگی می­شوند. کاربران این امکان را فراهم می­کنند که خوشه­ها بتوانند به راحتی به اشتراک گذاشته شوند. همچنین اشتراک شایستگی می­تواند با وظایف اولویت­ها را تخصیص دهد.

شایستگی زمانبندی وظایف و منابع را به طور عادلانه تخصیص می­دهند. هر کاربر به طور پیش فرض دارای استخر است و این امکان را می­دهد که هر خوشه به اشتراک گذاشته شود. در داخل هر استخر FIFO یا زمابندی به صورت عادلانه مورد استفاده قرار می­گیرد. در هر یک از استخرها، مدل­ها به صورت پیش فرض تمام وظایف را به استخر ارسال می­کنند. بنابراین، اگر خوشه با استخر شود، دو کاربر A و B می­گویند که به هر کدام سه وظیفه اختصاص دهید تا خوشه­ها تمام 6 وظیفه را به طور موازی اجرا کنند. فرض کنید که یک استخر سهم خود را به صورت یک دوره زمانی دریافت نکرده است، زمانبندی به طور اختیاری از وظایف در سایر استخرها پشتیبانی می­کند. زمانبندی مجاز خواهد بود استخرهایی که در حال اجرا هستند از بین ببرد بنابراین شیارها می­توانند به استخرهایی که در حال اجرا هستند ارائه شوند.

به منظور تضمین وظایفی که به تولید گرسنگی نرسیده­اند، پیشگزینی می­تواند مورد استفاده قرار گیرد در حالی که با اجازه، خوشه هادوپ می­تواند از تحقیقات و وظایف آزمایشی استفاده کند.

**C. زمانبندی ظرفیت**

زمانبندی ظرفیت ]14[ ]15[ به طور خاص برای محیط­هایی طراحی شده است که نیاز به اشتراک منابع محاسباتی میان تعدادی از کاربران در آن وجود دارد. این یک رویکرد نسبتا متفاوتی برای زمانبندی چند کاربره است. یک خوشه از چندین صف تشکیل شده و ممکن است سلسله مراتب متفاوتی داشته باشد و هر صف دارای ظرفیت تخصیص یافته است. در هر صف وظایف با زمانبندی FIFO اولویت­بندی شده­اند. در واقع، زمانبندی ظرفیت به کاربران یا سازمان­ها این امکان را می­دهد که بتوانند یک خوشه نگاشت کاهش را به صورت جداگانه با زمانبندی FIFO برای هر کاربر یا سازمانی شبیه­سازی کنند.

**D. ارتقاء زمانبندی براساس اشکال اجرایی وظایف**

گاهی اوقات ممکن است اتفاق بیفتد که چندین وظیفه در مجموعه­ای از وظایف ادامه و زمان اجرای آنها به آهستگی ادامه پیدا کند. با توجه به وظایف در حال انجام یک وظیفه می­تواند تمام کارها را انجام دهد و می­تواند وقت بیشتری را صرف اتمام کند و فقط در یک زمان کوتاهی به پایان می­رسد. ممکن است دلایل مختلفی مانند بارگذاری بالا یک گره در CPU، آهسته شدن فرآیندهای پس­زمینه، پیکربندی نرم افزار یا تخریب سخت افزار را داشته باشد. هادوپ تلاش می­کند تا یک نسخه پشتیبان از یک کار را که زمان کندتری دارد اجرا کند. این به عنوان یک اجرای وظیفه نظری نامگذاری می­شود. اگر وظایف اصلی قبل از کار به پایان برسند، سپس وظیفه­ی نظری از بین می­رود، اگر وظایف نظری به پایان برسند، وظایف اصلی نیز از بین خواهند رفت. اجرای نظری ]16[ قابلیت اطمینان شغلی را تضمین نمی­کند. اگر اشکال­ها دارای دو وظیفه اصلی نیز باشند گاهی اوقات کار را متوقف می­کنند سپس اشکالات مشابه در کارهای احتمالی ظاهر خواهند شد. بنابراین این نوع وضعیت برای کارهای احتمالی غیرممکن خواهد بود. بنابراین یک نیاز برای تعمیر اشکال­ها وجود دارد طوری که نحوه عملکرد وظایف کند نشود.

1) زمانبندی LATE: طولانی­ترین زمان تقریبی تا پایان (LATE) بهبود قابل توجهی را نسبت به اعداد به طور پیش فرض در نظر می­گیرند. پیاده­سازی زمانبندی احتمالی LATE به طور ضمنی روی فرض­های خاص متمرکز است: a) پیشرفت یکنواخت وظایف در گره b) محاسبات یکنواخت در تمام گره­ها. اما این فرضیات به راحتی در خوشه­های ناهمگن متوقف می­شوند.

با در نظر گرفتن یک نسخه اصلاح شده از اجرای احتمالی به جای کارهایی که تاکنون انجام شده، زمان محاسباتی که باقی­مانده­اند برآورد می­شوند و ارزیابی واضح­تری از وظایف استراتژی شغلی را به ما می­دهد.

2) الگوریتم زمانبندی تاخیر: هدف شایستگی زمانبندی طراحی اشتراک عادلانه ظرفیت در میان تمام کاربران است. این الگوریتم از دو مسئله رنج می­برد. اولین مسئله زمانبندی خطی است که در وظایف کوچک رخ می­دهد. دومین مسئله محل سکونت شیارها است ]17[. برای حل مسائل خطی، زمانبندی یک وظیفه را از یک کار دریافت می­کند و به گره بدون نگهداری از شایستگی داده­های محلی ارسال می­کند. زمانی که کاربر در یک گره داده­ای ندارد، انجام نمی­شود و بهتر است که وظیفه را در همان گره اجرا کند. در زمانبندی تاخیر، زمانی که یک گره درخواست وظیفه را انجام می­دهد، اگر وظیفه به صورت head-of-line باشد یک کار محلی را نمی­تواند راه­اندازی کند، و اجرای آن متوقف می­شود و به وظیفه بعدی می­رود. با این حال، اگر وظیفه به اندازه کافی طول بکشد، وظایف غیرمحلی می­توانند برای اجتناب از گرسنگی راه اندازی شوند.

3) سایر روش­های زمانبندی: علاوه بر این زمانبندی­ها، انواع زمانبندی­های دیگری ]1[ مانند زمابندی پویا وجود دارد. همانطور که از نام آن پیداست، این زمانبندی از توزیع ظرفیت به طور پویا از میان کاربران به صورت همزمان کاربرانی که اولویت بالایی دارند را پشتیبانی می­کند. زمانبندی محدودیت­هایی که متوقف شده است طراحی می­کند و آنها را به منظور افزایش بهره­وری سیستم آدرس­دهی می­کند. زمانی که شغلی ارائه می­شود، آن را تحت آزمایش زمانبندی قرار می­دهند و می­توانند شغلی را تعیین کنند در غیر اینصورت نمی­توانند تعیین کنند. دسترسی به شیارهایی که در زمان محاسبه شده­اند در آینده بدون در نظر گرفتن تمام وظایف سیستم اجرا می­شوند. یک بار وظایف تعیین می­شوند و در آن کار مشخص را می­توان در مدت محدود زمانی که زمانبندی شغلی در نظر گرفته می­شود انجام داد. محدودیت­های متوقف شدن زمانبندی که در مورد کارهای غیرپیشگیرانه است مورد بحث قرار می­گیرند. اما در عوض وظایف می­توانند در زمانبندی پیشگیرانه اجرا شوند. رویکرد زمانبندی پیشگیرانه ]18[، دارای مزایای متعددی است از جمله در کار تولیدی جلوی تاخیر را می­گیرد و در عین حال اجازه می­دهد که سیستم با دیگر وظایف غیرتولیدی خود به اشتراک گذاشته شود. نوع دیگری از زمانبندی که تاکنون با در دسترس بودن منابع در مقیاس به حساب نمی­آمد مورد بحث قرار می­گیرد. منابع یادگیرنده زمانبندی به عنوان نام سعی می­کند بر استفاده از منابع به طور موثر دلالتی داشته باشد. این هنوز هم یک چالش پژوهشی است.

**5. نگاهی کلی به جنبه­های فنی مسائل**

هادوپ دارای فایل­های پیکربندی مهمی است ]19[ ]20[ و در هنگام پیکربندی هادوپ باید در نظر گرفته شوند. چند فایل پیکربندی به طور قابل توجهی در زیر ارائه شده است:

* Hadoop-env.sh: متغیرهای محیطی در اسکریپت­ها به منظور اجرای هادوپ مورد استفاده قرار می­گیرند.
* Core-site.xml: این گزینه شامل تنظیمات پیکربندی هسته هادوپ است، و شامل تنظیمات ورودی/خروجی است که به طور رایج به HDFS و نگاشت کاهش دسترسی پیدا می­کند.
* Hdfs-site.xml: تنظیمات پیکربندی دمون HDFS: شامل گره نام، دومین گره نام و گره­های داده است.
* Mapred-site.xml: تنظیمات پیکربندی دمون نگاشت کاهش: شامل jobtracker و tasktrackers است.
* ارباب: فهرستی از ماشین­هایی است که به طور متوسط هر گره نام را اجرا می­کنند.
* برده: فهرستی از ماشین­هایی است که هر کدام گره داده و tasktracker را اجرا می­کنند.

**6. یکی دیگر از منابع انتقال دهنده (YARN)**

هادوپ یکی از چارچوب محاسباتی خوشه­ای برای پردازش کلان داده­ها است. اگرچه مبحث هادوپ قابل استدلال است با این حال به یک راه حل استاندارد به منظور مدیریت کلان داده­ها تبدیل شده است، این از محدودیت­های آزاد نیست. نگاشت کاهش دارای مقیاس­پذیری محدودی است و شامل 4000 گره می­باشد ]21[. یکی دیگر از محدودیت­ها عدم توانایی هادوپ در اشتراک گذاشتن منابع بین چندین چارچوب محاسباتی است. برای حل این محدودیت­ها، دسته­ای از منابع باز در نسل بعدی نگاشت کاهش که YARN نام دارد ارائه می­شوند (یکی دیگر از منابع انتقال دهنده) ]21[ ]22[. دانشمندان کامپیوتر و مهندسان تلاش می­کنند تا محدودیت­ها را از بین ببرند و عملکرد هادوپ را بهبود ببخشند. YARN محدودیت مقیاس­پذیری نسل اول نگاشت کاهش را حذف می­کند.

نسخه قبلی هادوپ YARN را نداشت اما در نسخه 2.0 هادوپ قابلیت افزایش اضافه شد ]23[. سیستم عامل پردازش به طور کلی توسط معماری مبتنی بر YARN از نسخه 2.0 هادوپ ارائه شده است و به نگاشت کاهش محدود نمی­شود. ایده اصلی YARN این است که دو ویژگی jobtracker، مدیریت منابع و زمانبندی شغلی را به دلایلی به صورت جداگانه تقسیم کند. این ایده دارای یک مدیریت منابع سراسری و برنامه­های مستر است. مدیریت منابع، منابع بین تمام برنامه­های کاربردی در سیستم را داوری می­کند و دارای دو مولفه است: زمانبندی و مدیریت منابع.

زمانبندی مسئول تخصیص منابع بین اجرای برنامه­های کاربردی است. مدیر برنامه­های کاربردی وظیفه انتقال دهنده را مورد پذیرش قرار می­دهد، و سرویس­هایی را به منظور شروع مجدد برنامه کاربردی مستر روی خطاها ارائه می­دهد. در حال حاضر قابلیت­های مدیریت منابع در نگاشت کاهش وجود دارد همچنین منابع توسط YARN شدت نگاشت کاهش را در پردازش داده­ها به طور موثر به دست می­آورند. با برنامه­های کاربردی چندگانه YARN می­توان در هادوپ همه اشتراک­ها را به طور رایج در مدیریت منابع اجرا کرد.

**A. مقایسه YARN و نگاشت کاهش**

با جداسازی مدل برنامه­نویسی توابع مدیریتی منابع، YARN مسئول زمانبندی مربوط به توابع مولفه­های per-job است. در متن جدید، نگاشت کاهش فقط برنامه­های کاربردی مربوط به YARN را اجرا می­­کند. این جداسازی مقادیر بزرگی را به صورت انعطاف­پذیر در چارچوب برنامه­نویسی انتخاب و سپس ارائه می­دهد. چارچوب­های برنامه­نویسی مختصات برنامه­های ارتباطی را روی YARN اجرا می­­کنند، و جریان اجرا، و بهینه­سازی پویا را به عنوان شایستگی، بهبود عملکرد به آنها نشان می­دهند. نمونه­ای از معماری­های YARN و هادوپ در زمانبندی­های مختلف استفاده می­شود. نمونه­ای دیگر از هادوپ در زمانبندی JobTracker استفاده می­شود، در حالیکه YARN به طور پیش فرض از ظرفیت زمانبندی استفاده می­کند.

انواع مختلفی از نگاشت کاهش به طور قابل توجهی در زیر ارائه شده است.

* مدیر منابع مسئول مدیریت و تخصیص منابع خوشه­ای سراسری است.
* مستر (ارباب) برنامه کاربردی در هر کدام از برنامه­ها معرفی می­شود. مستر (ارباب) برنامه کاربردی با مدیر منابع در برابر درخواست منابع محاسباتی به صورت متقابل کار می­کنند.
* مدیر گره مسئول مدیریت فرآیندهای کاربران در هر گره است.

در نسخه قبلی هادوپ JobTracker بدقت چارچوب نگاشت کاهش را به هم متصل می­کرد. هر دو مسئول مدیریت منابع و مدیریت برنامه­ها نیز بودند. JobTracker این امکان را پیدا خواهد کرد که وظایف نگاشت کاهش هادوپ را اجرا کند. مدیر منابع جدید اجازه می­دهد که بقیه سرویس­ها مانند MPI در داخل همان خوشه از طریق برنامه کاربردی مستر (ارباب) اجرا شوند. در نگاشت هادوپ و کاهش، شیارها نمی­توانند به صورت متناوب مورد استفاده قرار گیرند. این بدان معنی است که خوشه در طول فاز نگاشت یا کاهش کم­تر استفاده می­شود. در آواتار جدید هادوپ شیارها می­توانند مجددا به عنوان بهترین منابع مورد استفاده قرار گیرند.

**7. نتیجه­گیری**

مقاله با یک مقدمه مختصر در مورد کلان داده­ها آغاز شد. کلان داده­ها می­توانند مزایای قابل قبولی را برای کسب و کار ارائه دهند.

سپس مقاله در مورد تکنولوژی­های کلان داده­ها، و هادوپ بحث خواهد کرد. مقاله در مورد HDFS و مدل برنامه­نویسی نگاشت کاهش صحبت می­کند. ما در مورد برخی از برنامه­های کاربردی هادوپ صحبت خواهیم کرد. سپس انواع زمانبندی در هادوپ که به طور مختصر مورد بحث قرار گرفته­اند را مورد استفاده قرار می­دهیم. ما برخی از جنبه­های فنی هادوپ را که فایل­های مهمی در پیکربندی را دارند مورد بحث و بررسی قرار دادیم. نگاشت کاهش برخی از محدودیت­ها را به طور قابل توجهی در صورتی که تعداد گره­ها افزایش پیدا کرد به دست می­آورد، و تکنولوژی محدودیت­های YARN را برطرف می­کند و به طور مختصر آنها را مورد بحث قرار می­دهد.

