

## 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 فیلم-های 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 فایل 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.

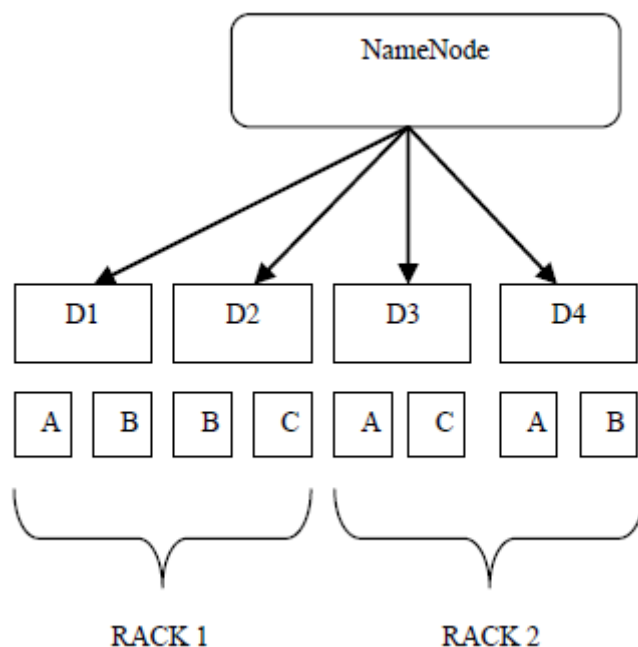
## 2- هادوپ

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 عمدتاً 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

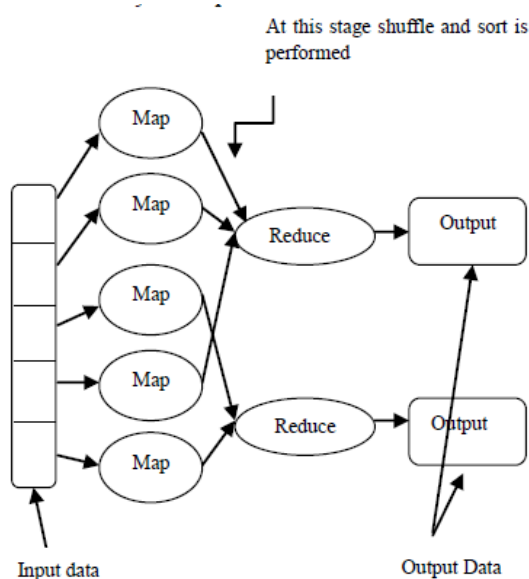
HDFS دارای معماری ارباب/برده است. معماری HDFS در شکل 1 نشان داده شده است. در شکل الفبای A,B,C بلوک داده‌ها را نشان می‌دهد و D عدد مربوط به گره داده را نشان می‌دهد. HDFS اکوسیستم با قابلیت تحمل بالا را ارائه می‌دهد. تک گره نام همراه با گره داده در یک خوشه معمولی HDFS وجود دارد. گره نام، مسئولیت سرور اصلی 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 گره-های داده نیست.

## B. نگاهت کاهش هادوپ

نگاشت کاهش [5] [6] یک تکنولوژی بسیار مهم است که توسط گوگل ارائه شده است. نگاهت 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 نگاهت کاهش هادوپ در شکل 2 نشان داده شده است. دو تابع پردازش داده مهم وجود دارند که عبارتند از نگاهت کاهش برنامه‌نویسی که شامل نگاهت و کاهش است.

داده‌های اصلی فاز نگاهت را به عنوان ورودی معرفی می‌کنند و طبق زمانبندی ای که توسط برنامه‌نویسان انجام شده نتایج را پردازش و تولید می‌کنند. برنامه‌های نگاهت موازی در یک زمان اجرا می‌شوند. اولاً، داده‌های ورودی به بلوک‌های با اندازه ثابت تقسیم می‌شوند و برنامه‌های نگاهت موازی را اجرا می‌کنند. خروجی نگاهت 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 در خواهد آمد.



شکل 2: نگاهت 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: ستون‌شی‌گرا، ذخیره داده‌های کلیدی توزیع شده غیررابطی 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 داده-ای داده‌ها را به صورت روش موازی پردازش می‌کند.
- آپاچی Zookeeper: واسط یک برنامه کاربردی است (API) که این امکان را فراهم می‌کند که پردازش داده-های 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 فلووم: 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 جغرافیایی 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 بعداً اضافه می‌کنند.

#### A. زمانبندی پیش‌فرض هادوپ

نسخه‌های هادوپ به طور سراسر 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 پیش‌فرض 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 زمانبندی 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 ببرند 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 نسبتاً 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 باقی مانده‌اند 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 راه‌اندازی شوند.

3) سایر روش‌های زمانبندی: علاوه بر این زمانبندی‌ها، انواع زمانبندی‌های دیگری [1] مانند زمانبندی 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 نداشت 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 نداشت 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. نداشت 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 نداشت 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 نداشت 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 نداشت 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 (اریاب) برنامه کاربردی در هر کدام از برنامه‌ها معرفی می‌شود. مستر (اریاب) 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 مجدداً 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.

## REFERENCES

- [1] B.Thirumala Rao, Dr.L.S.S.Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments", International Journal of Computer Applications (0975 – 8887) Volume 34– No.9, November 2011.
- [2] Kala Karun. A, Chitharanjan.K, "A Review on Hadoop – HDFS Infrastructure Extensions", Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).
- [3] Lizhe Wang, Jie Tao, Rajiv Ranjan , Holger Marten, AchimStreit , Jingying Chen, DanChen, "G-Hadoop: MapReduce across distributed data centers for data-intensive computing", 2013.
- [4] WeijiaXu, Wei Luo, Nicholas Woodward, "Analysis and Optimization of Data Import with Hadoop", 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum.
- [5] HUANG Lan, WANG Xiao-wei, ZHAI Yan-dong, YANG Bin, "Extraction of User Profile Based on the Hadoop Framework".
- [6] Jeffy Dean,Sanjay Ghemawat. MapReduce, "Simplified Data Processing on Large Clusters", OSDI04: Sixth Symposium on Operating System Design and Implementation, Ssn Francisco, CA,December, 2004.
- [7] What is Apache Hadoop? <https://hadoop.apache.org>
- [8] Introduction to the Hadoop Software Ecosystem. <http://www.revelytix.com/?q=content/hadoop-ecosystem>
- [9] Other applications. [https://en.wikipedia.org/wiki/Apache\\_Hadoop](https://en.wikipedia.org/wiki/Apache_Hadoop)
- [10] 10 ways companies are using Hadoop (for more than ads). <https://gigaom.com/2012/06/05/10-ways-companies-are-using-hadoop-to-do-more-than-serve-ads/>
- [11] Business Applications of Hadoop. <http://www.edureka.in/blog/business-applications-of-hadoop/>
- [12] Job Scheduling. Hadoop: The Definitive Guide, Third Edition, Textbook.
- [13] Hadoop's Fair Scheduler. [https://hadoop.apache.org/docs/r1.2.1/fair\\_scheduler](https://hadoop.apache.org/docs/r1.2.1/fair_scheduler).
- [14] Jagmohan Chauhan, Dwight Makaroff and Winfried Grassmann, "The Impact of Capacity Scheduler Configuration Settings on MapReduce Jobs".
- [15] Hadoop's Capacity Scheduler. [http://hadoop.apache.org/docs/r1.2.1/capacity\\_scheduler.html](http://hadoop.apache.org/docs/r1.2.1/capacity_scheduler.html).
- [16] M. Zaharia, A. Kowinski, A. Joseph, R. Katz and I. Stoica, Improving mapreduce performance in heterogeneous environments. USENIX OSDI, 2008
- [17] DongjinYoo, Kwang Mong Sim, "A comparative review of job scheduling for mapreduce," Multi-Agent and Cloud Computing Systems Laboratory, Proceedings of IEEE CCIS2011.
- [18] Preemptive Hadoop Jobs Scheduling under a Deadline, "Li Liu, Yuan Zhou, Ming Liu, Guandong Xu, Xiwei Chen, Dangping Fan, Qianru Wang", IEEE 2012 Eighth International Conference on Semantics, Knowledge and Grids.
- [19] Hadoop Administrative Guide. <http://caen.github.io/hadoop/administration-hadoop.html>.
- [20] Hadoop Cluster Configuration Files. <http://www.edureka.in/blog/hadoop-cluster-configuration-files>.
- [21] Arinto Murdopo, Jim Dowling, "Next Generation Hadoop: High Availability for YARN".
- [22] Vinod Kumar Vavilapalli, Arun C Murthy, Chris Douglas, SharadAgarwal, MahadevKonar, Robert Evans, Thomas Graves, Jason Lowe, Hitesh Shah, Siddharth Seth, BikasSaha, Carlo Curino, Owen O'Malley, Sanjay Radia, Benjamin, Reed, Eric Baldeschwieler, "Apache Hadoop YARN: Yet Another Resource Negotiator".
- [23] Hortonworks Hadoop YARN. <http://hortonworks.com/hadoop/yarn/>.