بررسی آپاچی هدوپ و آپاچی اسپارک

امروزه راه حل های زیادی برای پردازش کلان داده ها داریم. بسیاری از شرکت ها نیز هستند که خدمات سازمانی تخصصی ای برای تکمیل پلتفرم های open-source ارائه می دهند.

این روند از سال ۱۹۹۹ شروع شد، زمانی که آپاچی لوسن (Lucene) به وجود آمد. این پلتفرم خیلی زود تر از آنچه تصور میشد توسعه پیدا کرد و منجر به ایجاد Hadoop در سال ۲۰۰۸ شد. این روند مدتها ادامه یافت و به پیدایش آپاچی اسپارک در سال ۲۰۱۲ منتهی شد. حال این دو فریم ورک بزرگترین پلتفرم های open-source برای پردازش کلان داده ها هستند اما همیشه سوال اینجاست که کدام پلتفرم بهتر است، هدوپ یا اسپارک؟

پس بهتر است که کمی به تفاوت ها وشباهت های این دو فریم ورک بپردازیم.

هدوپ (Hadoop) چیست؟

آپاچی هدوپ پلتفرمی است که مجموعه داده های بزرگ را به صورت توزیع شده مدیریت می کند. این فریم ورک از MapReduce برای تقسیم داده‌ها به صورت چندین بلوک و تخصیص چانک ها به نود ها در یک کلاستر استفاده می‌کند. در نهایت MapReduce داده ها را به صورت موازی در هر گره پردازش می کند تا یک خروجی منحصر به فرد تولید کند.

هدوپ در برابر خطا و ارور بسیار مقاوم است و برای دسترسی بالا (High Availability) به سخت افزار متکی و وابسته نیست. همچنین هسته ی آن به منظور خطایابی در لایه ی اپلیکیشن به وجود آمده است. با تکثیر داده‌ها در یک کلاستر، زمانی که یک قطعه سخت‌افزار از کار می‌افتد، می‌تواند قسمت‌های گمشده را از مکان دیگری بسازد.

ساختار هدوپ از چهار مورد زیر است:

۱- HDFS (Hadoop Distributed File System): فایل سیستمی است برای مدیریت انباره ی دیتاست های کلان در کلاستر هدوپ.

۲- MapReduce : در اصل کامپوننت پردازشی هدوپ است.

۳- YARN (Yet Another Resource Negotiator) : مسئول مدیریت منابع محاسباتی و زمان بندی کار (job scheduling) .

۴- Hadoop Common : مجموعه ای از کتابخانه ها و ابزارهای کمکی رایج که سایر ماژول ها به آنها وابسته هستند.

اکنون پیش زمینه ای از اسپارک را برایتان تشریح کنم.

اسپارک (Spark) چیست؟

آپاچی اسپارک یک ابزار open-source است. این فریم ورک می تواند در حالت مستقل (standalone) یا روی یک فضای ابری یا سیستم مدیریت کلاستر مانند Apache Mesos و سایر پلتفرم ها اجرا شود. برای عملکرد سریع طراحی شده است و از RAM برای ذخیره و پردازش داده ها استفاده می کند.

Spark برای بهبود کارایی MapReduce و حفظ مزایای آن ایجاد شده است. حتی اگر Spark سیستم فایل خود را نداشته باشد ، می تواند به داده ها در بسیاری از راه حل های ذخیره سازی مختلف دسترسی داشته باشد. ساختار داده ای که اسپارک استفاده میکند RDD نام دارد.(Resilient Distributed Dataset)

آپاچی اسپارک دارای پنج جزء اصلی بوده که عبارتند از:

۱- Apache Spark Core : اساس کل پروژه است . هسته اسپارک وظایف ضروری مانند زمان بندی، ارسال وظایف، عملیات ورودی و خروجی، بازیابی خطا و غیره را بر عهده دارد.

۲- Spark Streaming : این مؤلفه پردازش جریان های Live Data را امکان پذیر می کند.

۳- Spark SQL : Spark از این مؤلفه برای جمع آوری اطلاعات در مورد داده های ساختار یافته و نحوه پردازش داده ها استفاده می کند.

۴- Machine Learning Library (MLlib) : این کتابخانه از بسیاری از الگوریتم های یادگیری ماشین تشکیل شده است.

۵- GraphX : مجموعه ای از API ها که برای تسهیل وظایف تجزیه و تحلیل گراف استفاده می شود.

تفاوت های کلیدی میان اسپارک و هدوپ

بخش‌های زیر به تشریح تفاوت ها و شباهت های اصلی بین این دو فریم ورک می‌پردازند. اکنون نگاهی به این دو پلتفرم از زوایای مختلف خواهیم داشت.

Spark	hadoop	معیار مقایسه
عملکرد سریع در حافظه با کاهش عملیات خواندن و نوشتن دیسک.	عملکرد کندتر، از دیسک برای ذخیره سازی استفاده میکند و سرعتش وابسته به سرعت خواندن و نوشتن دیسک است.	**کارایی**
یک پلتفرم متن باز، اما برای محاسبات به حافظه متکی است، که همین امر به طور قابل توجهی هزینه های اجرا را افزایش می دهد.	یک پلتفرم متن باز، هزینه کمتر برای اجرا. از سخت افزار مقرون به صرفه استفاده می کند. یافتن متخصصان آموزش دیده هدوپ آسان تر است.	**هزینه**
مناسب برای داده های تکراری و live data . برای اجرای عملیات با RDD و DAG کار می کند. (DAG: Directed acyclic graph)	بهترین برای پردازش batch ها. از MapReduce برای تقسیم یک مجموعه داده بزرگ در یک کلاستر برای تجزیه و تحلیل موازی استفاده می کند.	**پردازش دیتا**
فرآیند ایجاد بلوک RDD را ردیابی می کند، و سپس می تواند یک مجموعه داده را در صورت خرابی یک پارتیشن، بازسازی کند. Spark همچنین می تواند از DAG برای بازسازی داده ها در سراسر گره ها استفاده کند.	یک سیستم بسیار مقاوم در برابر خطا. داده ها را در سراسر گره ها تکرار می کند و در صورت بروز مشکل از آنها استفاده می کند.	**تحمل خطا**
مقیاس پذیری آن کمی چالش برانگیزتر است زیرا برای محاسبات به RAM متکی است. از هزاران گره در یک کلاستر پشتیبانی می کند.	به راحتی با افزودن گره ها و دیسک ها برای ذخیره سازی مقیاس پذیر است. از ده ها هزار گره بدون محدودیت پشتیبانی می کند.	**مقیاس پذیری**

امن نیست. به طور پیش فرض، امنیت خاموش است. برای دستیابی به سطح امنیتی لازم ، نیاز به ادغام با Hadoop دارد.	بسیار امن	**امنیت**
کاربر پسندتر. API ها را می توان در Java، Scala، R، Python، Spark SQL نوشت.	استفاده از زبان های کمتر. از Java یا Python برای برنامه های MapReduce استفاده می کند.	**سهولت در استفاده و زبان های برنامه نویسی پشتیبانی شده**
با پردازش در حافظه بسیار سریعتر. از MLlib برای محاسبات استفاده می کند.	کندتر از اسپارک. قطعات داده می توانند خیلی بزرگ باشند و گلوگاه (bottleneck) ایجاد کنند. کتابخانه اصلی آن Mahout نام دارد.	**یادگیری ماشین**

هدوپ یا اسپارک؟

هدوپ و اسپارک فناوری هایی برای مدیریت کلان داده ها هستند. به غیر از این، آنها در نحوه مدیریت و پردازش داده ها، فریم ورک های بسیار متفاوتی هستند.

با توجه به بخش‌های مقایسه ای قبلی ، به نظر می‌رسد که اسپارک فریم ورک قدرتمندتری است. در حالی که ممکن است این تفکرتا حدی درست باشد. در واقعیت آنها برای رقابت با یکدیگر ایجاد نشده اند، بلکه مکمل هستند. البته در برخی موارد هست که انتخاب یکی از آن ها به تنهایی باید صورت بگیرد تا بهترین عملکرد را داشته باشیم (بعنوان مثال استفاده از اسپارک در موارد مربوط به یادگیری ماشین ارجح تر است) اما در اکثر مواقع هدوپ و اسپارک در کنار هم بهترین کارکرد را دارند.

هر دو فریم ورک نقش مهمی در برنامه های کاربردی کلان داده ایفا می کنند . در حالی که به نظر می رسد Spark با سرعت و حالت کاربر پسند خود پلتفرم مورد استفاده تری است، برخی پروژه ها نیاز به اجرای Hadoop دارند. این امر به ویژه زمانی مهم است که حجم زیادی از داده ها نیاز به آنالیز داشته باشند.

Spark به بودجه بیشتری برای تعمیر و نگهداری نیاز دارد، اما همچنین نیاز به سخت افزار کمتری برای انجام کارهای مشابه که Hadoop نیز انجام می دهد ، دارد. باید در نظر داشت که این دو فریم ورک مزایای خود را دارند و بهترین حالت را زمانی دارند که با هم کار کنند.