بسمه تعالی

راهنمای صریح هادوپ

اثر : تام وایت

مترجمان :

امین مرادبیکی – افروز مرادبیکی – وحید خطیبی بردسیری

انتشارات ارسطو
(چاپ و نشر ایران)
۱۳۹۶

سرشناسه : وایت، تام
White, Tom (.Tom E)
عنوان و نام پدیدآور : راهنمای صریح هادوپ/ اثر تام وایت؛ مترجمان امین مرادبیکی، افروز مرادبیکی، وحید خطیبی‌بردسیری.
مشخصات نشر : مشهد: ارسطو، ۱۳۹۶.
مشخصات ظاهری : ۲۱۰ص.: مصور.
شابک : ۸-۱۵۱-۴۳۲-۶۰۰-۹۷۸
وضعیت فهرست نویسی : فیپا
یادداشت : عنوان اصلی: Hadoop : the definitive guide ,Fourth edition. ,c2015
موضوع : آپاچی هادوپ
موضوع : Apache Hadoop
موضوع : سازماندهی فایل‌ها (کامپیوتر)
موضوع : File organization (Computer science)
شناسه افزوده: مرادبیکی، افروز، ۱۳۶۸ - ، مترجم
شناسه افزوده: مرادبیکی، امین، ۱۳۶۴ - ، مترجم
شناسه افزوده: خطیبی بردسیری، وحید، ۱۳۶۰ -، مترجم
رده بندی کنگره : ۱۳۹۶/QA۷۶ ۲و۲۳س/ ۹
رده بندی دیویی : ۰۰۵/۷۴
شماره کتابشناسی ملی : ۴۶۹۵۱۴۵

نام کتاب : راهنمای صریح هادوپ
نویسنده : تام وایت
مترجمان : امین مرادبیکی - افروز مرادبیکی - وحید خطیبی بردسیری
ناشر : ارسطو (چاپ و نشر ایران)
صفحه آرایی، تنظیم و طرح جلد : پروانه مهاجر
تیراژ : ۱۰۰۰
نوبت چاپ : اول - ۱۳۹۶
چاپ : مدیران
قیمت : ۱۷۰۰۰ تومان
شابک : ۸-۱۵۱-۴۳۲-۶۰۰-۹۷۸
تلفن های مرکز پخش : ۳۵۰۹۶۱۴۵ - ۳۵۰۹۶۱۴۶ - ۰۵۱
www.chaponashr.ir

چاپ و نشر ایران
انتشارات ارسطو

فهرست مطالب

عنوان صفحه

عنوان | صفحه

عنوان | صفحه

عنوان | صفحه

عنوان | صفحه

عنوان | صفحه

عنوان | صفحه

عنوان | صفحه

فصل اول

مقدمه

ما در دنیایی از اطلاعات زندگی می‌کنیم. بر اساس یک تخمین که در سال ۲۰۱۳ ارائه شـــد حجم داده‌های موجود در جهان ۴.۴ زتابایت اســـت و این مقدار تا سال ۲۰۲۰ به رقم باورنکردنی ۴۴ زتابایت خواهد رسید. یک زتابایت برابر با ۱۰۲۱ بایت یا معادل با ۱ بیلیون ترابایت اســـت. این مقدار بیشـــتر از آن است که شـــما به هر شخص در جهان یک دیسک حافظه بدهید.

این سیل بزرگ داده‌ها از منابع مختلفی می‌آیند که شامل موارد زیر است:

- بورس اوراق بهادار نیویورک روزانه حدود ۴ تا ۵ ترابایت داده تولید می‌کند.
- میزبان‌های فیسبوک بیش از ۲۴۰ بیلیون عکس دارند که این مقدار ماهیانه ۷ پتا بایت افزایش می‌یابد.
- وب سایت معرفی خانواده‌ها مانند Ancestry.com حدود ۱۰ پتا بایت داده را ذخیره کرده اند.
- آرشیو اینترنت حدود ۱۸,۵ پتابایت داده را ذخیره کرده است.
- برخورد دهنده بزرگ هادرون که در نزدیکی جنوا سوئیس قرار دارد سالیانه حدود ۳۰ پتابایت داده تولید می‌کند.
- ...

ذخیره و تحلیل داده

مسئله بسیار ساده است: اگرچه در سال‌های اخیر ظرفیت ذخیره دیسک‌های سخت افزایش قابل توجهی داشته است اما زمان دستیابی (زمان مورد نیاز که داده از روی دیسک خوانده شود) چندان کاهش نیافته است. به عنوان نمونه درسال ۱۹۹۰ درایوهای اولیه می‌توانستند ۱،۳۷۰ مگابایت داده را ذخیره کنند و سرعت خواندن داده در آن زمان ۴,۴ مگابایت در ثانیه بود یعنی برای خواندن همه داده‌های آن درایو ما به زمان حدود ۵ دقیقه نیاز داشتیم. ۲۰ سال بعد دیسک‌های ۱ ترابایتی بسیار فراگیر و پرکاربرد شد. اما سرعت خواندن داده‌های آن حدود ۱۰۰ مگابایت در ثانیه است. یعنی برای خواندن کل اطلاعات دیسک ما به زمان حدود نیم ساعت احتیاج داریم.

این زمان برای خواندن کل داده‌ها زمان بسیار زیادی است و زمان نوشتن داده در دیسک از این هم کندتر است. راه حل بدیهی برای کاهش زمان خواندن، خواندن همزمان از دیسک‌های چندگانه است. فرض کنید اگر ما ۱۰۰ درایو داشته باشیم، هرکدام ۱۰۰ داده را نگهداری میکنند. چنانچه این سیستم بصورت موازی کار کند، ما می‌توانیم داده‌ها را در کمتر از ۲ دقیقه بخوانیم.

استفاده همزمان از ۱۰۰ دیسک ممکن است قدری اسراف کارانه به نظر برسد. اما ما می‌توانیم ۱۰۰ دیتاست را ذخیره کنیم که هر یک از آنها ۱ ترابایت داده را ذخیره می‌کند، و امکان دسترسی اشتراکی را فراهم می‌کند. می‌توان تصور کرد که کاربران چنین سیستم‌هایی تمایل دارند تا دسترسی خود را در اختیار دیگران قرار دهند و در عین حال زمان آنالیزهای آنها کمتر شود و در عین حال با این روش در کار یکدیگر نیز اختلال ایجاد نکنند.

اولین مشکل که باید برای مسئله همزمان سازی حل شود مسئله مشکلات و خطاهای سخت افزاری است: به محض اینکه شما کار را با تعداد زیادی قطعات سخت افزار شروع می‌کنید، احتمال وقوع مشکلات سخت افزاری افزایش می‌یابد. راه حل عمومی برای جلوگیری از آسیب دیدن داده ایجاد کپی‌های متعدد از داده است. به عبارت دیگر هر سیستم یک کپی از داده را نگهداری می‌کند تا در صورت بروز خطا کپی از داده‌ها موجود

باشد. سیستم-فایل توزیع شده هادوپ (HDFS) کمی شبیه RAID است که بعدا در مورد آن صحبت خواهیم کرد.

مشـــکل دوم این است که برخی از تحلیل‌ها که روی داده انجام می‌شود نیاز به ترکیب کـــردن داده دارد، بنابراین داده ای که از یک دیســـک خوانده می‌شـــود باید با داده ای که از ۹۹ دیســـک دیگر خوانده می‌شود ترکیب شود.سیســـتم‌های توزیع شده گوناگون اجازه میدهند تا داده خوانده از منابع مختلف با یکدیگر ترکیب شـــوند، اما اجرای این مســـئله به صورت صحیح، خود یک چالش اســـت. نگاشـــت کاهـــش (MapReduce) یک مدل برنامه نویسی فراهم می‌کند که مسئله خوانده و نوشن دیسک و همچنین جابجایی داده‌ها هنگام محاســـبات را خلاصه می‌کنـــد. ما جزئیات این مدل را در فصل بعد بررســـی خواهیم کرد. همانند HDFS ، نگاشت کاهش برای قابلیت اطمینان ایجاد شده است.

به طور خلاصه می‌توان گفت آنچه هادوپ فرهم می‌کند عبارتســـت از: یک ســـکو قابل اطمینان و قابل توسعه برای ذخیره و تحلیل داده.

پرس و جوی همه داده‌های شما

راه حل ارائه شده بوسیله نگاشت کاهش به نظر می‌رسد که شبیه راه حل brute-force باشـــد. فرض کنید یک دیتاســـت وارد شده است – یا قسمت مناسبی از آن – که می‌تواند برای یک پرس و جو پردازش شـــود. اما این مطلب قدرت آن اســـت. نگاشـــت کاهش یک پردازنده دسته ای پرس و جو است. در این روش پرسش و جوهایی که بیش از حد بزرگ است می‌توان به پرس و جوهای کوچک تر تقسیم کرد و به آن‌ها پاسخ داد.

به عنوان مثال تعدادی از سرویس دهنده‌های پست الکترونیک از هادوپ برای پردازش گزارشات عملکرد کاربران استفاده می‌کنند.

فراتر از دسته‌ای

اساسا نگاشت کاهش یک سیستم پردازش دسته ای است و برای تحلیل‌های تعاملی گزینه مناسبی نیست. شما نمی‌توانید یک پرس و جو را مطرح کنید و پاسخ آن را در عرض چند ثانیه دریافت کنید. پرس و جوها نوعا چندین دقیقه یا حتی بیشتر زمان می‌گیرد بنابراین برای کاربردهای برخط که یک شخص منتظر دریافت پاسخ است مناسب نیست. اما در حقیقت هادوپ فراتر از یک سیستم دسته ای تکامل یافته است. درواقع کلمه هادوپ اغلب برای اشاره به مجموعه ای از پروژه‌های بزرگ استفاده می‌شود، نه فقط برای HDFS و نگاشت کاهش، بلکه همانند یک چتر فراساختار گونه برای محاسبات توزیع شده و پردازش‌های با ابعاد بزرگ مورد استفاده قرار می‌گیرد.

اولین قسمت برای فراهم کردن دسترسی برخط Hbase است. Hbase هر دو دسترسی برخط برای خواندن و نوشتن در سطرهای مجزا را فراهم می‌کند.

توانمند ساز حقیقی برای مدل پردازش جدید در هادوپ تحت عنوان یارن (Yet Another Resource Negotiator) در هادوپ ۲ معرفی شد. یارن یک سیستم مدیریت منابع خوشه ای است که به هر برنامه توزیع شده اجازه اجرا شدن روی داده در خوشه هادوپ را می‌دهد.

در سال‌های اخیر الگوهای پردازشی مختلفی که با هادوپ کار می‌کنند معرفی شده است.

مقایسه با دیگر سیستم‌ها

هادوپ اولین سیستم توزیع شده برای ذخیره و تحلیل داده‌ها نیست. اما برخی از ویژگی‌های منحصر بفرد آن ممکن است شبیه سیستم‌های مشابه به نظر برسد. در اینجا ما به برخی از این موارد اشاره می‌کنیم.

سیستم مدیریت بانک اطلاعات رابطه ای

چرا ما نمی توانیم از یک بانک اطلاعات با تعداد زیادی دیسک برای تحلیل داده با ابعاد بسیار بزرگ استفاده کنیم؟ چرا ما به هادوپ نیاز مندیم؟

پاسخ به این سوال از یکی از روندهای بکار رفته در هارد دیسک بدست می‌آید. بهبود زمان جستجو در دیسک‌ها آهسته تر از سرعت انتقال است. زمان جستجو زمانی است که هد دیسک در قسمتی که باید داده نوشته یا خوانده شود قرار می‌گیرد. این تاخیر یک مشخصه مربوط به عملیات دیسک است در حالی که سرعت انتقال مربوط به پهنای باند دیسک است.

البته مسئله سرعت زمانی که الگوی ما وابسته به سرعت جستجو باشد، به اندازه قطعه داده ای که قصد کار کردن با آن را داریم وابسته است. به عبارت دیگر برای به‌روزرسانی قسمت کوچکی از یک بانک اطلاعات، روش جستجوی سنتی B-Tree به خوبی عمل می‌کند. در مقابل برای بروز رسانی بخش بزرگی از بانک اطلاعات روش جستجو B-Tree نسبت به روش نگاشت کاهش، چندان موثر نیست.

در موارد زیادی روش نگاشت کاهش می‌تواند به عنوان مکمل برای سیستم مدیریت بانک اطلاعات رابطه ای (RDBMS) در نظر گرفته شود. روش نگاشت کاهش هنگامی که دیتاست‌ها به صورت دسته ای پردازش می‌شوند به ویژه برای تحلیل به صورت ad hoc بسیار مناسب است. نگاشت کاهش هنگامی که داده یکبار نوشته می‌شود و چندین بار خوانده می‌شود بسیار مناسب است در حالیکه بانک اطلاعات رابطه ای برای حالتی که دیتاست‌ها بطور مداوم بروزرسانی می‌شوند بسیار مناسب است.

جدول ۱-۱ مقایسه نگاشت کاهش با RDBMS

نگاشت کاهش	RDBMS سنتی	
پتابایت	Gigabytes	سایز داده
دسته ای	تعاملی و دسته ای	دسترسی
یکبار نوشتن و چندین بار خواندن	خواندن و نوشتن متعدد	بروزرسانی
---	ACID	نوع رابطه
Schema-on-read	Schema-on-write	ساختار
پایین	بالا	جامعیت
خطی	غیر خطی	توسعه پذیری

اگرچه تفاوت بین سیستم هادوپ و بانک اطلاعات رابطه‌ای کم‌رنگ است. بانک اطلاعات رابطه‌ای از ترکیب برخی ایده‌های هادوپ با ایده‌های دیگر بوجود آمده است. در حال حاضر سیستم هادوپ بسیار تعاملی‌تر شده است و ویژگی‌های جدیدی مانند اینکس‌ها و ... به آن اضافه شده که آن را شبیه RDBMS سنتی کرده است.

تفاوت دیگر بین هادوپ و RDBMS در اندازه ساختار دیتاست‌ها برای هریک از عملیات است. داده ساخت‌یافته به صورت خوش تعریف سازماندهی شده است، به عنوان مثال مانند فایل XML یا جداول بانک اطلاعات. این موارد در حوزه RDBMS است. داده نیمه ساخت‌یافته، به عبارت دیگر در این حالت ساختار سست تر است یعنی اگرچه ممکن است ساختاری وجود داشته باشد، اما اغلب نادیده گرفته می‌شود و ممکن است تنها به عنوان راهنمای برای ساختار داده مورد استفاده قرار گیرد. هادوپ به خوبی روی داده غیر ساخت‌یافته یا نیمه ساختیافته عمل می‌کند زیرا برای تفسیر داده در زمان پردازش طراحی شده است. از آنجایی که یک کپی از فایل را دارد این وضعیت موجب انعطاف‌پذیری و

اجتناب از هزینه بارگذاری داده در مقایسه با RDBMS می‌شود.

داده رابطه ای اغلب برای حفظ جامعیت و حذف افزونگی نرمالسازی می‌شود. نرمالسازی برای پردازش هادوپ یک مشکل به حساب می‌آید زیرا اطلاعات توسط عملیات غیر محلی خوانده می‌شود.

یک وب سرور گزارش عملکرد مثال خوبی از مجموعه ای از رکوردهای نرمال‌سازی نشده است و این یکی از دلایلی است که فایل گزارش عملکرد از هر نوعی که باشد برای پردازش توسط هادوپ بسیار مناسب است. توجه کنید که هادوپ قادر به ترکیب کردن منابع با یکدیگر نیز است اما این مبحث چندان که در نوع رابطه‌ای بکار برده می‌شود استفاده نمی‌شود.

نگاشت کاهش و سایر مدل‌های پردازش موجود در هادوپ بصورت خطی با افزایش ابعاد داده توسعه می‌یابد. در اینجا داده قطعه بندی می‌شود و عملکردهای اولیه می‌توانند بصورت موازی در قطعات جداگانه عمل کنند. این به آن معنی است که اگر شما اندازه داده ورودی را ۲ برابر کنید، کار به اندازه ۲ برابر کندتر خواهد شد.. اما اگر شما اندازه خوشه را ۲ برابر کنید، یک کار سریعتر از حالت اولیه خواهد بود. این مسئله به صورت عمومی برای پرس و جوهای SQL درست نیست.

محاسبات شبکه‌ای

مجموعه محاسبات با کارایی بالا(HPC) و محاسبات شبکه ای، سالهاست که برای پردازش داده با ابعاد بزرگ استفاده می‌شود. روش HPC به طور گسترده برای توزیع یک کار بین یک خوشه از ماشین‌ها است که به یک سیستم-فایلی بصورت اشتراکی دسترسی دارند و بوسیله شبکه منطقه ای ذخیره سازی (SAN) میزبانی می‌شوند. این کار برای مشاغلی که به محاسبات جامع نیازمندند مناسب است اما یک مشکل وجود دارد و آن هنگامی است که گره‌ها نیاز به دسترسی به حجم بالای داده دارند زیرا پهنای بند شبکه کم است و گره‌ها بیکار خواهند ماند.

هــادوپ تلاش می‌کند تا داده‌های گره‌هایی که در حال محاســبه هســتند را در کنار یکدیگر قرار دهد، بنابراین دسترسی به داده‌ها سریع خواهد بود زیرا داده محلی است. این ویژگی که به نام محلی‌ســازی داده شــناخته می‌شود قلب پردازش داده در هادوپ و علت کارایی بالای آن است. شناسایی پهنای باند شبکه با ارزش‌ترین مسئله در محیط یک مرکز داده است.

واســط ارســال پیام (MPI) یک کنترل بســیار مهم برای برنامه نویســان است، اما نیاز صریحی به کنترل جریان داده دارد. پردازش‌ها در هادوپ فقط در سطح بالا عمل می‌کنند.

همکاری پردازش‌ها در محاسبات توزیع شده با ابعاد بالا یک چالش مهم است. بزرگترین مشــکل در این بحث، بوجود آمدن خرابی در بخشــی از یک سیستم توزیع شده است، در حالی که شــما نمی‌دانید یک پردازش راه دور دچار خطا شده است یا خیر. برنامه‌نویسانی که در چهارچوب پردازش‌های توزیع شــده کار می‌کنند باید به شــکلی عمل کنند که در صورت تشــخیص خطا بتوانند پردازش‌های خود را بر روی یک ماشــین سالم ادامه دهند. نگاشــت کاهش قادر به انجام این مهم است زیرا از معماری shared-nothing بهره می‌برد، و این معماری به معنای آن اســت که وظایف هیچ وابســتگی به یکدیگر ندارند. بنابراین از نقطه نظر برنامه نویس نظمی در انجام وظایف موجود نیست. در مقابل برنامه‌های MPI باید خودشان مسائل مربوط به نقاط کنترل و بازیابی خود را مدیریت کنند، این مسئله کنترل بیشتری به برنامه‌نویس می‌دهد اما کمی کار را مشکل می‌کند.

محاسبه داوطلبانه

هنگامی که شــخصی برای اولین بار درباره هادوپ و نگاشــت کاهش می‌شــنود اغلب می‌پرســد: چه فرقی با SETI@home دارد؟ (SETI، جســتجوی هوش فرازمینی، در این پروژه افراد به صورت داوطلبانه اقدام به پردازش سیگنال‌های رادیویی توسط CPU سیستم شــان برای جســتجوی هوش فرازمینی می‌کنند). پروژه SETI@home معروفترین پروژه محاسبه داوطلبانه است.

پروژه محاسبه داوطلبانه بر اساس شکستن یک مسئله به تکه‌هایی به نام واحد کار، عمل می‌کند، این تکه‌ها به کامپیوترهایی در سطح جهان ارسال می‌شود تا مورد تجزیه تحلیل قرار گیرد و حل شود. هنگامی که تجزیه و تحلیل داده به پایان رسید، نتایج به سرور ارسال می‌شود و یک واحد کار دیگر برای محاسبه ارسال می‌شود. برای جلوگیری از تقلب‌های احتمالی، هر محاسبه به ۳ ماشین ارسال می‌شود و پاسخی که بیشتر تکرار شده است مورد پذیرش قرار می‌گیرد.

اگرچه ممکن است قدری شباهت بین نگاشت کاهش و SETI@home وجود داشته باشد، اما تفاوت‌های قابل توجهی وجود دارد. مسئله SETI@home تمرکز بالایی روی CPU دارد بنابراین برای اجرا کردن صدها هزار کامپیوتر در سطح جهان مناسب است بنابراین زمان انجام یک فرایند به مراتب بیشتر از زمان اجرای آن خواهد بود. بنابراین داوطلبانه بودن در این جا تنها در خصوص سیکل‌های CPU است و ارتباطی با پهنای باند ندارد.

فصل دوم

نگاشت کاهش

نگاشت کاهش

نگاشـــت کاهشـــی (Map Reduce) یک مدل برنامه‌نویسی برای پردازش داده است. این مدل برنامه‌نویســـی باوجود بهبودهای انجام‌شـــده تاکنون جهت ارائه‌ی برنامه‌های کاربردی چندان ساده نیست. هادوپ (Hadoop) برنامه‌های نگاشت کاهشی انجام شده در زبان‌های مختلف را می‌تواند اجرا کند. در این فصل ما برنامه‌های یکســـان بیان شـــده به زبان جاوا را بررســـی می‌کنیم. مهم‌تر از همه با توجه به اینکه ذات موازی‌ســـازی در برنامه‌های نگاشت کاهشـــی وجود دارد، بنابراین قرار دادن تجزیه و تحلیل داده در مقیاس بسیار بزرگ به هر کدام از زبان‌های برنامه‌نویسی که دارای ماشین پردازش کافی باشد، به انتخاب شما وابسته است.

مجموعه داده آب‌وهوا

به‌عنوان‌مثـــال مـــا یک برنامه کـــه داده‌های آب‌وهوا را اســـتخراج می‌کند می‌نویســـیم. حس‌گرهـــای آب‌وهوا داده را در هر ســـاعت از مناطق در سراســـر جهـــان جمع‌آوری کرده و حجـــم زیادی از اطلاعات ثبت شـــده را فراهم می‌کنند که نمونه‌ای مناســـب برای آنالیز به‌وســـیله نگاشت کاهشی اســـت چون ما قصد پردازش تمام داده‌ها را داشته و این داده‌ها نیمه ساختاریافته و رکورد گرا هستند.