

نگاهی بر ابزارهای کلان داده: معرفی ZooKeeper و Flume



دانشگاه فردوسی مشهد

دانشکده: علوم ریاضی

رشته: ریاضی کاربردی

گرایش: علوم داده

درس: مدل سازی و پردازش مه داده ها

استاد: دکتر جلال الدین نصیری

نویسنده: پرهام پیشرو

شماره دانشجویی: ۴۰۱۱۳۰۹۰۱۲

چکیده

در این فایل مستند که برای ارائه پاورپوینت آماده شده است، اطلاعات و توضیحات لازم برای فردی که قصد توضیح این ارائه را برای دیگران دارد ولی اطلاع کافی از ارائه موجود ندارد، آورده شده است. مطالب این مستند در ۴ فصل به همراه یک مقدمه و یک پیوست تنظیم شده است. فصل‌های مورد نظر به ترتیب عبارتند از: آپاچی فلوم، آپاچی زوکیپر، یک نمای کلی از اکوسیستم هدوپ و مراجع.

تعداد اسلایدهای فایل ارائه پاورپوینت برابر با ۲۰ اسلاید و تعداد اسلایدهای فایل پی دی اف منطبق بر پاورپوینت به علت تعداد زیاد و مهم بودن انیمیشن‌ها افزایش یافته و برابر با ۳۰ اسلاید می‌باشد. هم‌چنین تعداد صفحات این فایل مستند نیز برابر با ۲۱ صفحه است.

اگر بخواهیم به یکی از مهم‌ترین و پر کاربردترین مسائل حال حاضر جهان اشاره کنیم، باید از کلان داده یاد کرد. کلان داده یا همان Big Data مفهومی است که به مقدار عظیمی از داده (چه ساختاریافته باشد و چه بدون ساختار) اشاره می‌کند. امروزه بیش‌تر صنایع و شرکت‌های غول با این مفهوم درگیر هستند و درباره آن صحبت می‌کنند. در این مفهوم چیزی که مهم است، کارهایی است که با این کلان داده می‌توان انجام داد. از این کلان داده می‌توان برای استخراج اطلاعات با ارزش برای تصمیم‌گیری‌های مهم استفاده کرد. با آنالیز درست این داده‌ها می‌توان استراتژی‌های حساس و دقیقی را تنظیم کرد. استراتژی‌هایی که واقعاً بدون وجود این کلان داده‌ها احتمالاً تنظیمشان غیرممکن بود. [۱]

حال جهت فهمیدن تفاوت بین داده‌های معمولی و کلان داده‌ها باید به ویژگی‌های کلان داده اشاره کرد. طبق یک سری از مؤلفه‌ها و ویژگی‌ها می‌توان کلان داده را تعریف کرد. در سال‌های اخیر دانشمندان و محققین ویژگی‌های بسیاری را برای کلان داده در نظر گرفته اند که همگی آن‌ها با حرف “V” شرع می‌شوند. شایان ذکر است که اگر در صنایع مختلف صرفاً چند مورد از موارد زیر وجود داشتند، باز هم می‌توان آن‌ها را در حوزه کلان داده شمرد:

- **حجم^۱:** به اندازه مجموعه داده‌های مورد نیاز برای تجزیه و تحلیل و پردازش اشاره دارد که در حال حاضر اغلب از ترابایت و پتابایت بزرگتر هستند. حجم عظیم داده‌ها به فناوری‌های پردازش گر متمایز و متفاوتی نسبت به قابلیت‌های ذخیره‌سازی و پردازش سنتی نیاز دارد. به عبارت دیگر، این بدان معناست که مجموعه داده‌ها در کلان داده‌ها بسیار بزرگتر از آن هستند که بتوان با یک لپ‌تاپ یا پردازنده رومیزی معمولی پردازش کرد. نمونه‌ای از مجموعه داده‌های با حجم بالا، تمامی تراکنش‌های کارت‌های اعتباری در یک روز است.
- **سرعت^۲:** به سرعت تولید داده‌ها اشاره دارد. داده‌ها با چنان سرعتی تولید می‌شوند که به تکنیک‌های پردازش مجزا (توزیع شده) نیاز دارند. نمونه‌ای از داده‌هایی که با سرعت بالا تولید می‌شوند پیام‌های توییتر یا پست‌های فیسبوک هستند.
- **تنوع^۳:** تنوع داده‌ها، کلان داده را واقعاً بزرگ می‌کند. داده‌های بزرگ از منابع بسیار متنوعی به دست می‌آید و به طور کلی یکی از سه نوع است: داده‌های ساختار یافته، نیمه ساختار یافته و بدون ساختار. تنوع در انواع داده‌ها اغلب به قابلیت پردازش متمایز و الگوریتم‌های تخصصی نیاز دارد. نمونه‌ای از مجموعه داده‌های متنوع می‌تواند فایل‌های صوتی و تصویری دوربین مدار بسته باشد که در مکان‌های مختلف یک شهر تولید می‌شوند.

موارد فوق ۳ تا از مهم‌ترین ویژگی‌های کلان داده بودند. [۲] ابزارهای متعددی برای حوزه کلان داده در اختیار کاربران قرار دارد. از جمله آپاچی هدوپ (Apache Hadoop)، الگوریتم نگاشت کاهش (Map Reduce)، سیستم فایل توزیع شده هدوپ (Hadoop Distributed File System | HDFS)، آپاچی اسپارک (Apache Spark)، آپاچی هایو (Apache Hive)، آپاچی پیگ (Apache Pig)، آپاچی ماهوت (Apache Mahout)، کافکا (Kafka)، الاستیک سرچ (Elastic Search)، و ... در ادامه به دو مورد از ابزارهای کلان داده یعنی آپاچی فلوام (Apache Flume) و آپاچی زوکپیر (Apache Zookeeper) اشاره خواهد شد. [۳]

^۱ Volume

^۲ Variety

^۳ Velocity

قبل از بررسی آپاچی فلووم و آپاچی زوکیپر باید نگاهی کلی به روند کلی یک پروژه کلان داده داشته باشیم. در ۵ مرحله مختلف پروژه‌های کلان داده انجام می‌شوند:

(۱) **جمع آوری**: شامل جمع آوری داده‌ها از برنامه‌ها، نرم‌افزارهای موبایل، سرویس‌های کوچک، دستگاه‌های مجهز به اینترنت اشیا، وبسایت‌ها و انواع دیگر منابع داده است.

(۲) **یکپارچه سازی و انتقال**: داده‌ها با انواع مختلف (مثلاً صوت، تصویر، متن، جدول، داده‌های جریانی و ...) در این گام دریافت شده و پس از یکپارچه سازی آن‌ها به مرحله بعد ارسال می‌شوند. از جمله ابزارهای کاربردی در این مرحله می‌توان به کافکا و فلووم اشاره کرد.

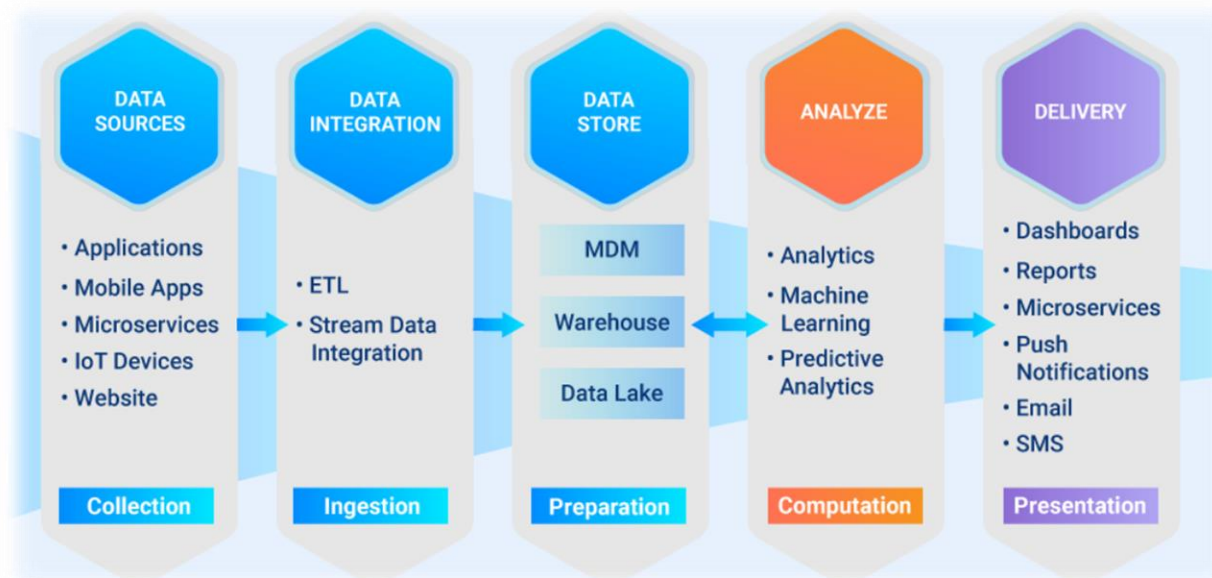
(۳) **ذخیره سازی**: داده‌ها با توجه به ماهیت و کاربردها و همچنین هدف صورت مسئله در حالت‌های مختلفی از جمله انبار داده و دریاچه داده ذخیره می‌شوند.

(۴) **تجزیه و تحلیل**: در این مرحله داده‌ها با ابزارهای مختلفی تجزیه و تحلیل شده و نتایج آن‌ها در گام بعدی نشان داده خواهد شد.

(۵) **ارائه نتایج**: پس از انجام تجزیه و تحلیل بر روی داده‌ها، باید نتایج حاصل را نشان داد که این کار با ابزارهای مختلفی از جمله داشبوردها، گزارش‌ها، اعلان‌ها، ایمیل، پیامک و ... انجام خواهد شد.

همه این ۵ مرحله گفته شده در شکل ۱ که در زیر آمده است، به نمایش درآمده است. به ترتیب از چپ به راست نشان‌دهنده مراحل مختلف می‌باشد.

شایان ذکر است که دو ابزاری که در این ارائه قرار است بررسی شوند، یعنی آپاچی فلووم و آپاچی زوکیپر در مراحل مختلفی قرار می‌گیرند. آپاچی فلووم در مرحله دوم یعنی گام انتقال داده‌ها قرار می‌گیرد. آپاچی زوکیپر نیز با توجه به وظیفه اش که مدیریت اجزای مختلف اکوسیستم هدوپ است، می‌تواند در گام‌های مختلفی قرار گرفته باشد.



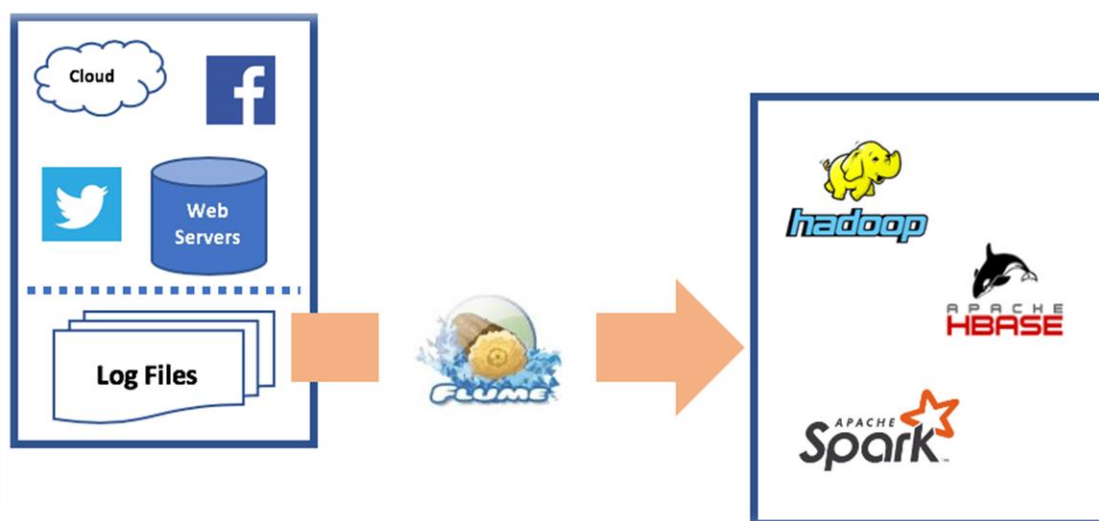
۱ آپاچی فلوم

مطالب این فصل از [۴]، [۵] و [۶] گردآوری شده است. قبل از پرداختن به بخش‌های مختلف آپاچی فلوم می‌توان خط زمانی به‌روز رسانی‌های این نرم‌افزار را مشاهده کرد. این خط زمانی شامل نسخه نرم‌افزار و تاریخ ارائه آن است. خط زمانی این نرم‌افزار به صورت زیر می‌باشد:

۲۶ جولای ۲۰۱۲	۱,۲,۰ •
۲ ژانویه ۲۰۱۳	۱,۳,۱ •
۲ جولای ۲۰۱۳	۱,۴,۰ •
۱۶ جولای ۲۰۱۴	۱,۵,۰,۱ •
۲۰ می ۲۰۱۵	۱,۶,۰ •
۱۷ اکتبر ۲۰۱۶	۱,۷,۰ •
۴ اکتبر ۲۰۱۷	۱,۸,۰ •
۸ ژانویه ۲۰۱۹	۱,۹,۰ •
۱۶ آگوست ۲۰۲۲	۱,۱۰,۱ •
۲۴ اکتبر ۲۰۲۲	۱,۱۱,۰ •

۱,۱ مقدمه‌ای بر آپاچی فلوم

همان طور که در شکل ۲ می‌توان مشاهده کرد، آپاچی فلوم در واقع داده‌ها را از منابع داده می‌گیرد (که این داده‌ها می‌توانند انواع مختلفی از جمله داده‌های فضای ابری، داده‌های سرورهای وب، داده‌های فیسبوک، داده‌های توییتر و داده‌های لاگ داشته باشند) و آن‌ها را به مرحله بعدی برای پردازش (مثل Hadoop، Apache HBASE و Apache Spark) ارسال کند. قدرت آپاچی فلوم زمانی معلوم خواهد شد که از داده‌های لاگ استفاده شود، زیرا فلوم بیش‌تر با هدف انتقال داده‌های لاگ ساخته شده است.



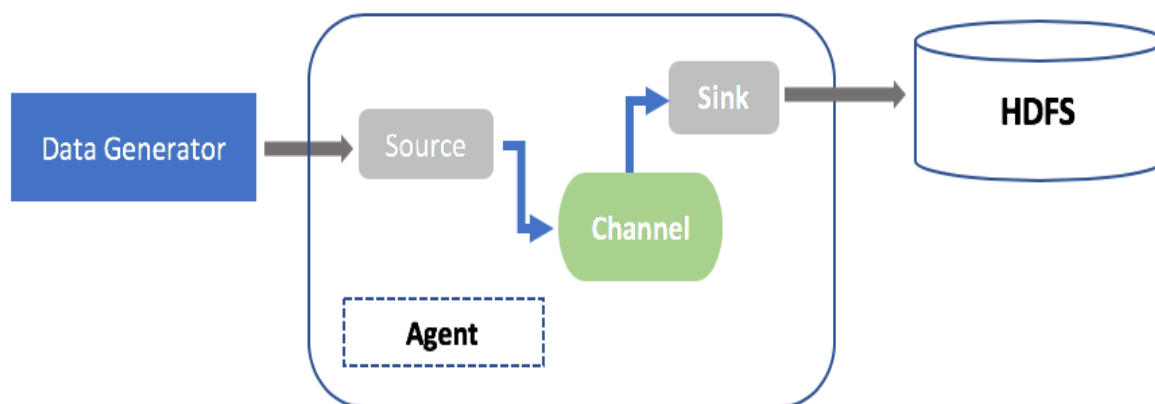
طبق تعریف رسمی خود سایت آپاچی، فلوم یک سرویس توزیع شده، قابل اعتماد و در دسترس برای جمع‌آوری کارآمد، تجمیع و انتقال حجم زیادی از داده‌های جریانی است. پس از فهمیدن مفهوم آپاچی فلوم، بیاید روی این مورد زوم کرده و نگاه دقیق‌تری به آن داشته باشیم.

۲.۱ معماری آپاچی فلوم

همان‌طور که در شکل ۳ می‌توان مشاهده کرد، فلوم از ۳ بخش مختلف تشکیل شده است. در سمت چپ شکل تولید کننده داده قرار دارد و سمت راست نیز منابع ذخیره‌سازی داده. در مستطیل وسط که نشان دهنده آپاچی فلوم است، می‌توان ۳ بخش مختلف آپاچی فلوم را مشاهده کرد:

- بخش اول یعنی Source وظیفه جمع‌آوری داده را بر عهده دارد.
- بخش دوم یعنی Channel وظیفه تجمیع داده را بر عهده دارد.
- بخش آخر یعنی Sink نیز وظیفه انتقال داده بر مرحله بعد را دارد.

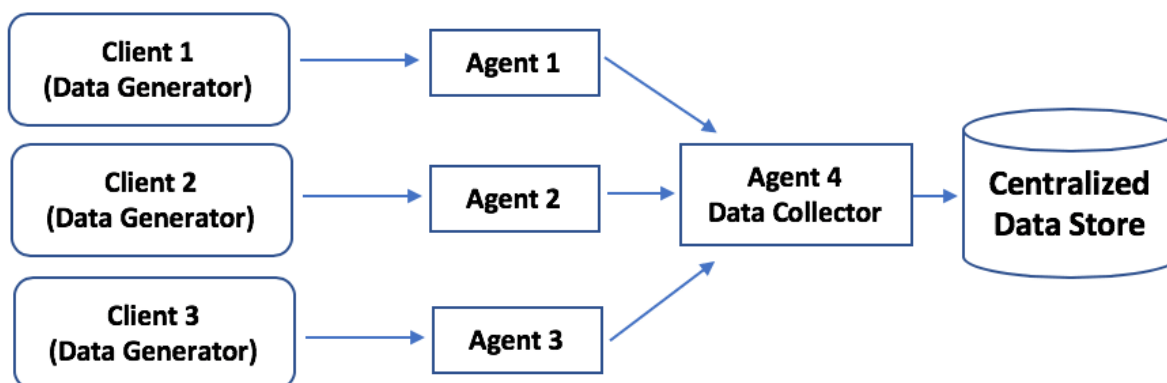
این ۳ بخش، اجزای اصلی آپاچی فلوم می‌باشند. معماری آپاچی فلوم که شامل Source، Channel و Sink می‌باشد در ادامه نمایش داده شده است.



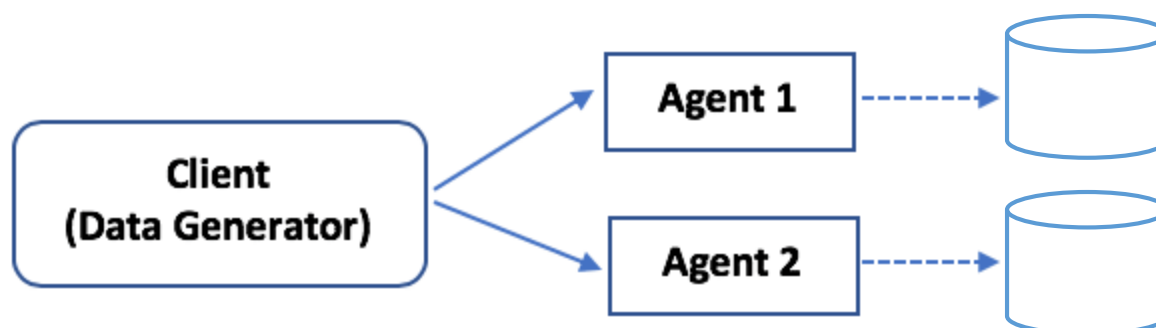
۳.۱ دو معماری مختلف

با توجه به معماری آپاچی فلوم که در بخش قبل توضیح داده شد، به دو صورت مختلف می‌تواند این معماری شکل بگیرد. البته می‌توان حالتی را در نظر گرفت که ترکیب شده از دو حالت زیر باشد (یا به عبارت دیگر حالات زیر بسط داده شده باشند): ولی ساخت چنین معماری که ترکیب شده از دو حالت گذشته بوده و حالت جامع‌تری را شامل می‌شود می‌تواند بسیار دشوار و چالش برانگیز باشد.

اولین حالت همان‌طور که در شکل ۴ قابل مشاهده است، حالتی است که چندین تولید کننده‌ی داده (تحت عنوان مشتری) موجود باشد و قصد داشته باشیم، داده‌های حاصل از این منابع تولید کننده داده را به یک منبع ذخیره سازی داده انتقال دهیم.



حالت دوم، حالتی است که در آن یک منبع تولیدکننده داده موجود باشد و قصد داشته باشیم داده‌های تولید شده را به چندین منبع ذخیره‌سازی انتقال دهیم. این حالت در شکل ۵ به نمایش درآمده است.



۵.۱ چند مثال از آپاچی فلوم

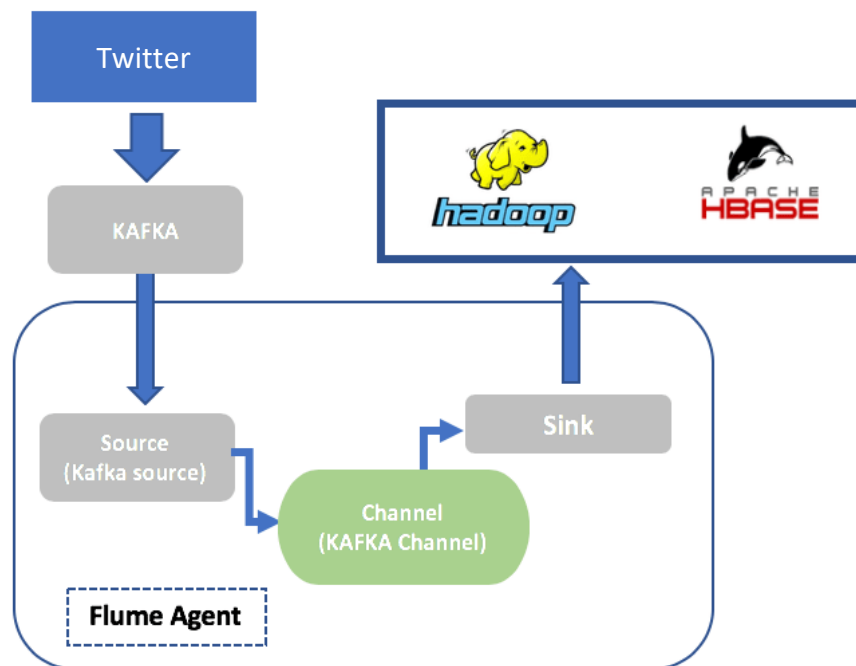
در ادامه چندین مثال از آپاچی فلوم نمایش داده شده تا بهتر بتوان طرز ساز و کار معماری این نرم افزار را درک کرد. پس از مثال‌ها نیز به محدودیت‌های آپاچی فلوم پرداخته خواهد شد.

۱.۵.۱ انتقال داده‌های جریانی لاگ از توییت‌ر به HDFS

نحوه دریافت، جمع‌آوری و انتقال داده‌ها از شبکه توییت‌ر^۴ به HDFS در شکل ۶ قابل مشاهده است. در واقع توییت‌ر در ابتدا داده‌های خود را با کمک Kafka به Flume انتقال می‌دهد. Kafka ابزاری است که در آن، هر کدام از داده‌ها به ترتیب همانند صف وارد مرحله بعد خواهند شد. پس از آن داده‌ها از طریق Source جمع‌آوری شده، در Channel تجمع شده و با Sink به مرحله بعد یعنی Hadoop و یا Apache HBASE منتقل می‌شوند.

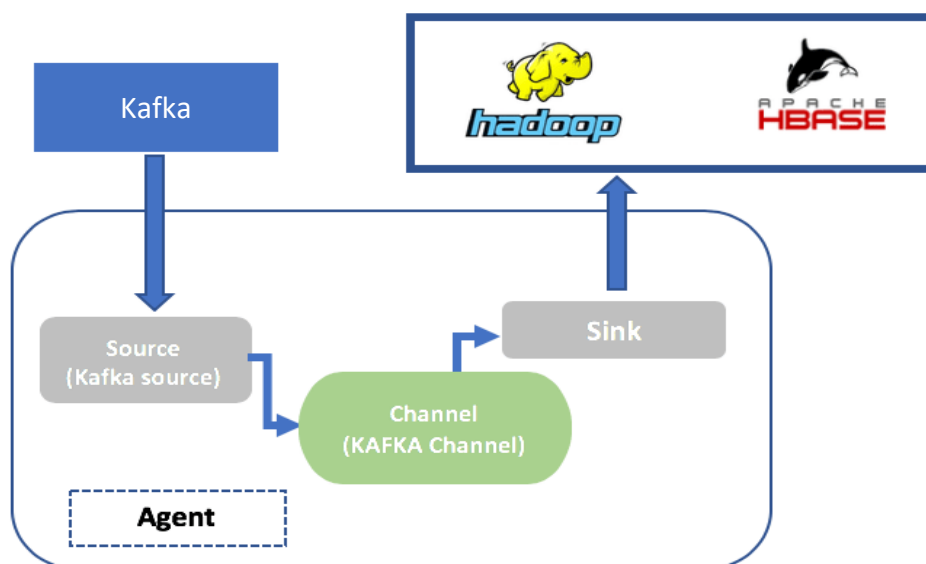
در شکل ۶ مستطیل وسط نشان‌دهنده Flume و اجزای مختلف آن می‌باشد. نحوه حرکت داده‌ها از توییت‌ر به منابع هدوپ نیز نشان داده شده است.

^۴ Twitter



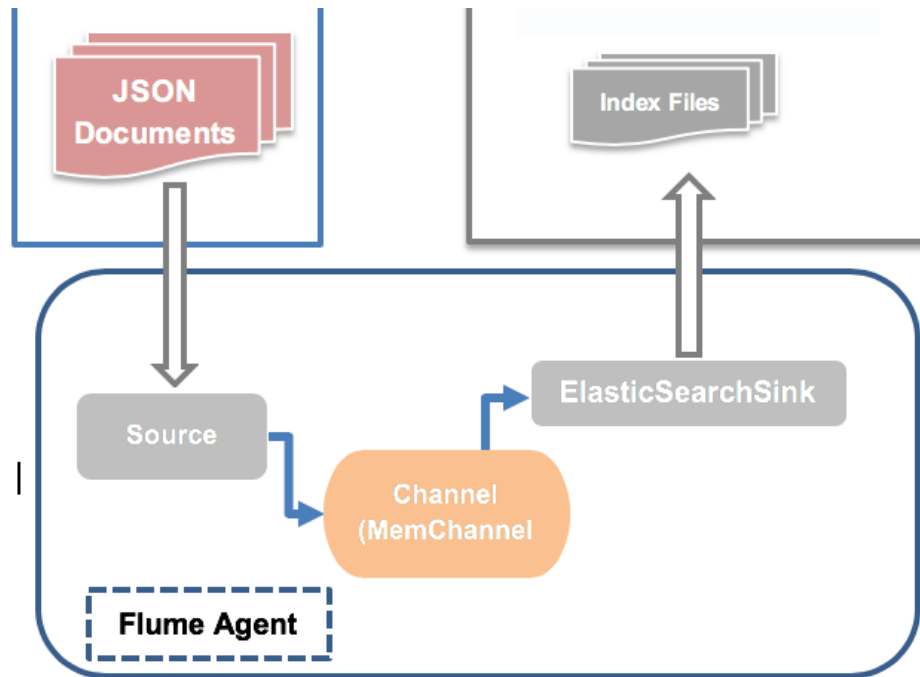
۲,۵,۱ انتقال داده‌های جریانی لاگ از کافکا به HDFS

این مثال نیز همانند مثال قبل می‌باشد با این تفاوت که داده‌ها از کافکا گرفته شده‌اند و منبع می‌تواند هر چیزی باشد. شکل ۷ نشان دهنده این مسئله است.



۳,۵,۱ انتقال داده‌های جریانی لاگ به الاستیک سرچ

در مثال بعد که در شکل ۸ به نمایش در آمده است، داده‌ها که به صورت فایل‌های JSON می‌باشند از منابع دریافت شده و پس از قرار گرفتن در Flume که آخرین مرحله آن (یعنی مرحله انتقال داده‌ها) توسط الاستیک سرچ انجام می‌شود، داده‌ها به صورت فایل‌های ایندکس شده به مرحله بعد منتقل می‌شوند.



۶,۱ محدودیت‌های آپاچی فلووم

اگر چه که فلووم دارای مزایای بسیاری است ولی دارای محدودیت‌ها و معایبی نیز می‌باشد. از جمله این محدودیت‌ها می‌توان به موارد زیر اشاره کرد:

- اگر معماری را به حالتی تغییر داد که داده‌ها از چندین مبدا (منابع تولید کننده داده) به چندین مقصد (منابع ذخیره سازی و پردازش داده) منتقل شوند، این معماری برای مدیریت و نگهداری داده‌ها بسیار پیچیده و سخت خواهد بود.
- داده‌های جریانی به صورت ۱۰۰٪ بلادرنگ نیستند.
- آپاچی فلووم برای تعریف و از هم تشخیص دادن داده‌های یکسان (داده‌هایی که در مجموعه داده تکرار شده اند) ضعف دارد و در این مورد ممکن است که داده‌ها از بین بروند.

۲ آپاچی زوکیپر

مطالب این فصل از [۷]، [۸]، [۹]، [۱۰] و [۱۱] گردآوری شده است. قبل از پرداختن به بخش‌های مختلف آپاچی فلووم می‌توان خط زمانی به‌روز رسانی‌های این نرم‌افزار را مشاهده کرد. این خط زمانی شامل نسخه نرم‌افزار و تاریخ ارائه آن است. خط زمانی این نرم‌افزار به صورت زیر می‌باشد:

۲۷ اکتبر ۲۰۰۸	• ۳,۰,۰
۱۴ دسامبر ۲۰۰۹	• ۳,۱,۲
۲ آگوست ۲۰۱۲	• ۳,۳,۶
۲ آپریل ۲۰۱۹	• ۳,۴,۱۴
۱۵ ژانویه ۲۰۲۱	• ۳,۵,۹
۱۳ آپریل ۲۰۲۱	• ۳,۶,۳
۱۲ می ۲۰۲۲	• ۳,۷,۱
۳۰ ژانویه ۲۰۲۳	• ۳,۸,۱

۱,۲ چرا به آپاچی زوکیپر نیاز داریم؟

علت اصلی نیاز ما به زوکیپر هماهنگی بین سرویس‌های مختلف در اکوسیستم هدوپ می‌باشد (یا به عبارت دیگر پیکربندی سیستم‌ها همزمان با همگام‌سازی داده‌ها). این بدین معنی است که در صورت استفاده از اکوسیستم هدوپ، با توجه به تعدد سرویس‌های موجود در آن و همچنین تزریق داده‌های جدید به برنامه، هماهنگ کردن بین این سرویس‌های مختلف دشوار است و پیکربندی این سرویس‌ها در حالتی که داده‌ها نیز دچار مشکل نشود، سخت می‌باشد. به همین منظور، آپاچی زوکیپر، ۴ سرویس در اختیار ما قرار می‌دهد که در ادامه هر کدام نامبرده شده و در شکل ۹ نیز قابل مشاهده است:

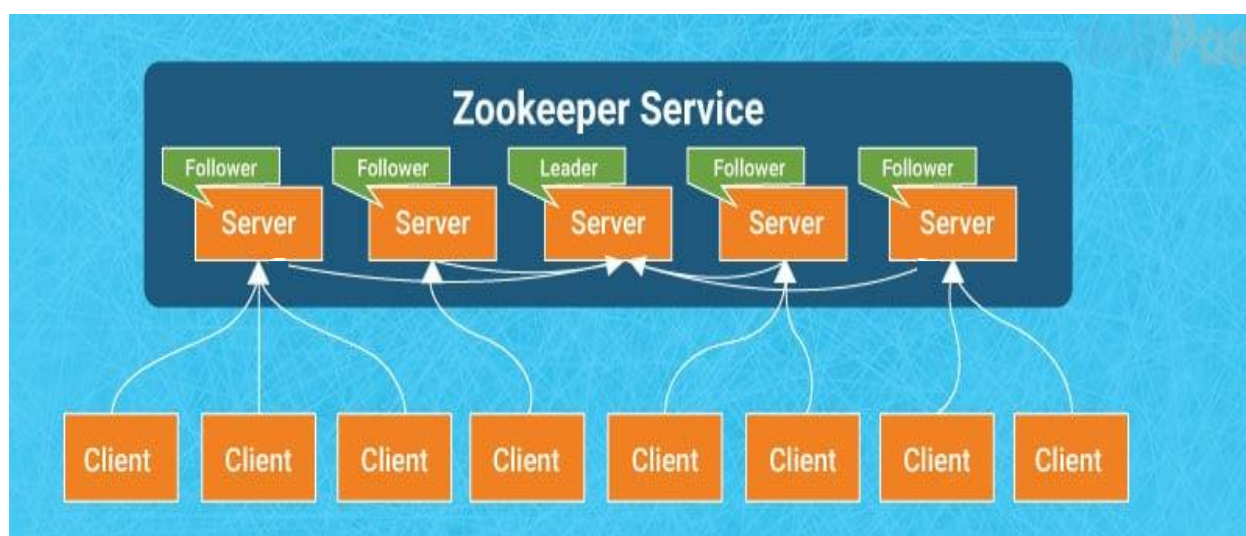
- (۱) سرویس نام‌گذاری
- (۲) سرویس مدیریت پیکربندی
- (۳) سرویس همگام‌سازی
- (۴) سرویس گروه‌ها



۲,۲ معماری آپاچی زوکیپر

همان طور که در شکل ۱۰ می توان مشاهده کرد، این معماری متشکل از ۴ عنوان مختلف است، به نام های سرور^۵، رهبر^۶، دنبال کننده^۷ و مشتری^۸.

- **سرور:** نشان دهنده همان سرورهای ما است که در اختیار داریم. هر سرور باید یک نقش بگیرد یا باید رهبر باشد یا باید دنبال کننده باشد.
- **رهبر:** سروری که رهبر است وظیفه هماهنگی بین سرورهای مختلف را بر عهده دارد و با مشتری ها به صورت مستقیم ارتباطی ندارد.
- **دنبال کننده:** سروری که دنبال کننده است در واقع همان درخواست های مشتری را نشان می دهد و هر سرور توسط یک رهبر هدایت می شود.
- **مشتری:** نشان دهنده همان دستگاه افرادی است که قصد دارند از سرورها استفاده کنند.



۳,۲ نحوه کار کردن آپاچی زوکیپر

حال این سرویس آپاچی زوکیپر چگونه می تواند با کمک سرورهایش که به دو دسته رهبر و دنبال کننده تقسیم شده اند، هماهنگی بین سرورها را انجام داده و همه درخواست های مشتریان را بدون قطع شدن انجام دهد؟ نحوه عملکرد آپاچی زوکیپر در چندین مرحله متوالی نشان داده شده است.

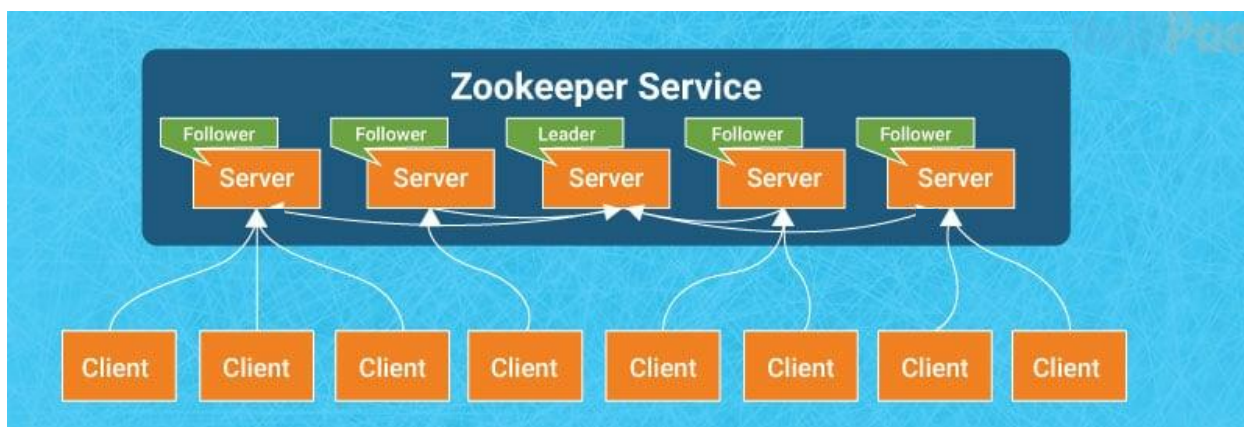
قبل از شروع سرویس زوکیپر وضعیت سرورها همانند شکل ۱۱ می باشد. همان طور که در شکل نیز پیداست هیچ ارتباطی بین سرورها و همچنین مشتری ها وجود ندارد. هم چنین نقش هیچ یک از سرورها نیز مشخص نشده است. یعنی مشخص نشده که کدام یک از سرورها رهبر است و کدام یک دنبال کننده.

^۵ Server
^۶ Leader

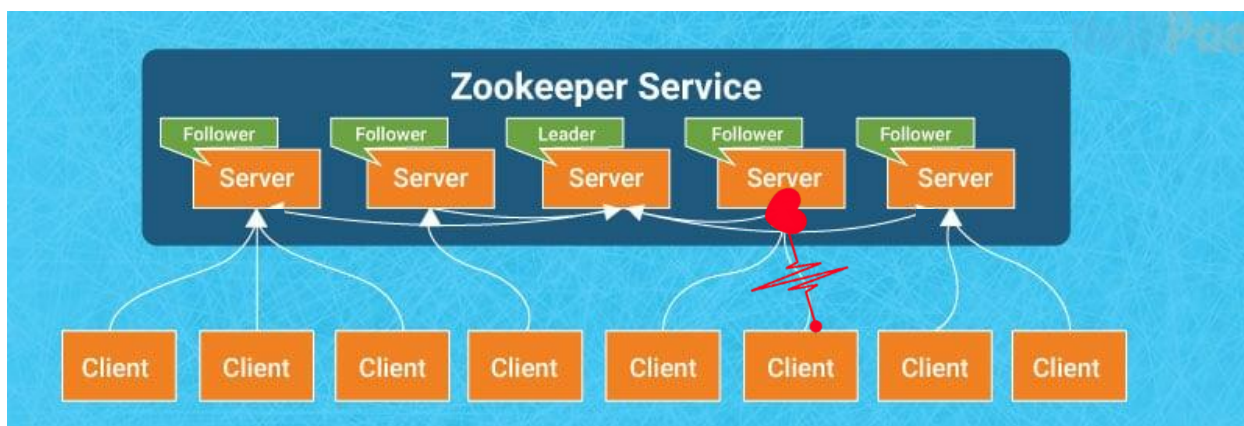
^۷ Follower
^۸ Client



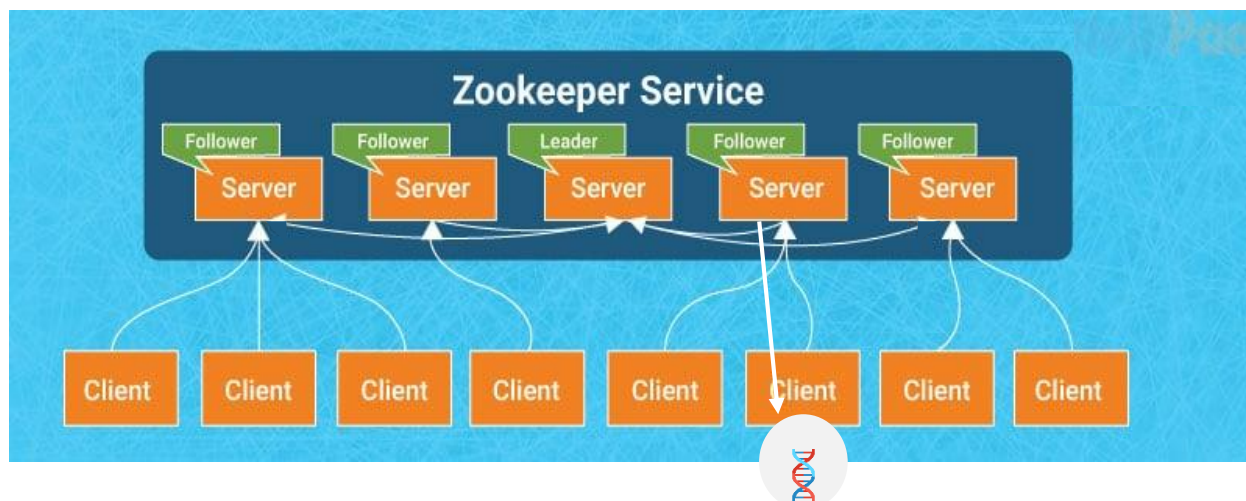
در مرحله بعد به صورت تصادفی هر کدام از مشتری‌ها به یکی از سرورها وصل می‌شوند. هر سروری که به آن مشتری متصل شود، سرور دنبال‌کننده نام خواهد گرفت. چندین سرور نیز وظیفه هماهنگی و هدایت این سرورهای دنبال‌کننده را بر عهده خواهند گرفت که آن‌ها نقش رهبر را دارند. در شکل ۱۲ این مسئله به نمایش درآمده است.



در مرحله بعدی، مشتری یک درخواست (که در اصطلاح به آن ضربه قلب می‌گویند) به سرور دنبال‌کننده ارسال می‌کند تا از وضعیت فعال بودن یا نبودن سرور اطمینان حاصل پیدا کند. این وضعیت در شکل ۱۳ نشان داده شده است.



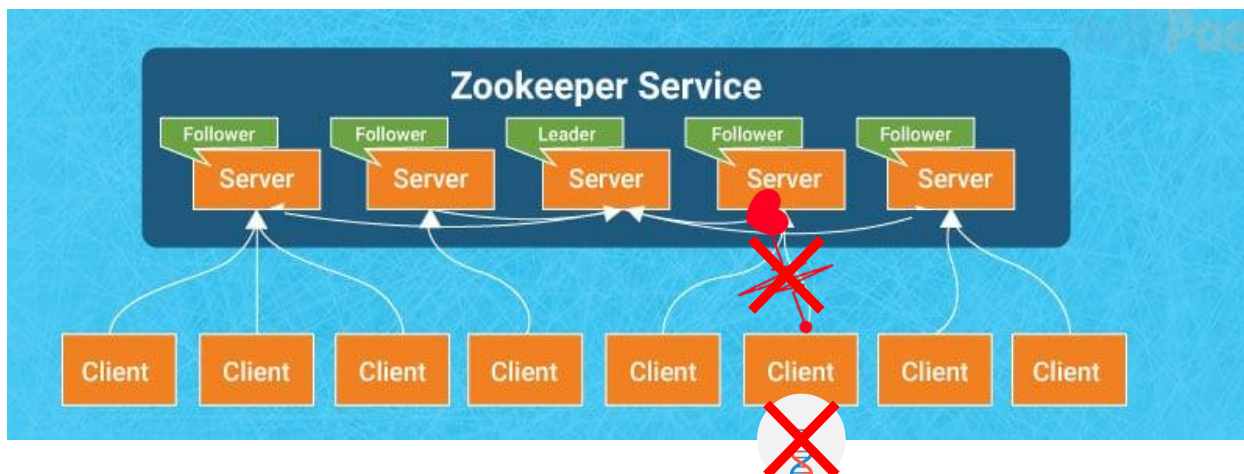
سرور نیز با اختصاص دادن یک آیدی منحصر بفرد به آن مشتری، به ضربان قلبی که از سوی مشتری ارسال شده پاسخ می‌دهد. هدف از این کار نشان دادن وضعیت فعال سرور است. تخصیص آیدی از سوی سرور به مشتری در شکل ۱۴ نشان داده شده است.



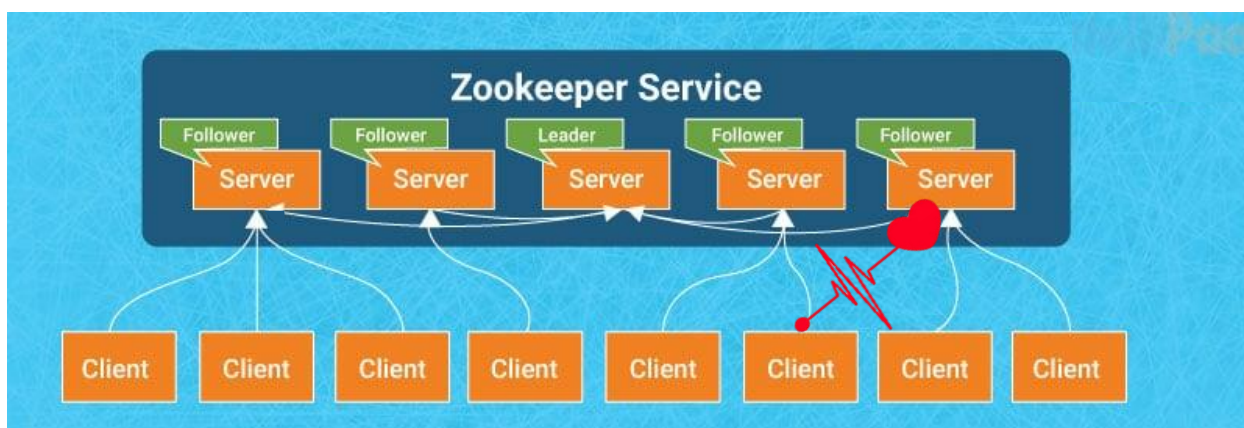
پس از ایجاد ارتباط بین سرور و مشتری، سرور شروع به کار خواهد کرد. مشتری در هر لحظه درخواستی (یا همان ضربان قلب) به سرور ارسال می‌کند و سرور نیز این درخواست را به مشتری جواب می‌دهد. این عملیات رفت و برگشت که به صورت مرتب تکرار خواهد شد، برای مطمئن شدن از ارتباط و پاسخ‌دهی سرور می‌باشد. در شکل ۱۵ این ارتباط بین سرور و مشتری به وضوح نشان داده شده است.



حال اگر سرور دچار اشکال شود و یا خاموش شود و یا به هر دلیلی این ارتباط قطع شود، چه اتفاقی خواهد افتاد؟ اینجاست که قدرت آپاچی زوکیپر به نمایش در آمده و فایده استفاده از این سرویس را می‌توان متوجه شد. در این حالت مشتری طبق معمول یک درخواست تحت عنوان ضربان قلب به سرور می‌فرستد ولی سرور دیگر این ضربان قلب را پاسخ نخواهد داد و در نتیجه مشتری نیز دیگر درخواستی را به آن سرور ارسال نخواهد کرد؛ همانند شکل ۱۶. این ارتباط قطع شده و در تصویر پس از آن نشان داده می‌شود که چگونه آپاچی زوکیپر این مسئله را مدیریت خواهد کرد.



حال که ارتباط بین مشتری و سرور دنبال کننده قطع شده است، سرور رهبر این ارتباط را با یک سرور دنبال کننده دیگر از سر خواهد گرفت. یعنی مشتری درخواست خود را به یک سرور دیگر ارسال خواهد کرد تا وقفه‌ای در عملیات ایجاد نشود. این عمل در شکل ۱۷ به نمایش درآمده است.



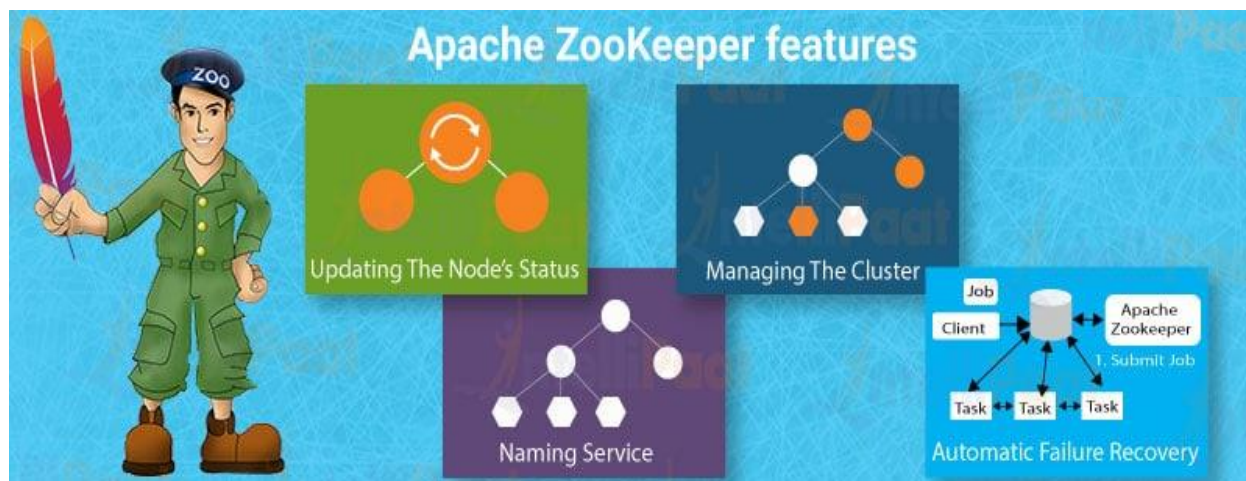
بدین صورت آپاچی زوکپیرو وظیفه خود را انجام می‌دهد و باعث مدیریت بین اجزای مختلف اکوسیستم هادوپ و هم‌چنین پیکربندی آن‌ها می‌شود.

۴.۲ ویژگی‌های آپاچی زوکپیرو

تا اینجا کار با برخی از ویژگی‌های آپاچی زوکپیرو آشنا شدیم. در این بخش، همان‌طور که در شکل ۱۸ نیز نشان داده شده است، صرفاً آن‌ها را جمع‌بندی کرده و توضیح مختصری درباره هر کدام از آن‌ها داده خواهد شد:

- **به‌روز رسانی وضعیت گره‌ها:** پس از نام‌گذاری مشتری‌ها و سرورها تحت عنوان گره، وضعیت هر کدام در هر لحظه بررسی شده و این وضعیت با توجه به فعال بودن یا عدم فعالیت آن گره، نمایش داده خواهد شد.
- **سرویس نام‌گذاری:** به هر گره (مشتری و سرورهای دنبال‌کننده و رهبر) یک نام خاص اختصاص داده خواهد شد که این نام خاص در واقع به عنوان شناسه همان گره در نظر گرفته خواهد شد. در ادامه به طور مفصل به آن پرداخته خواهد شد.

- مدیریت خوشه: پس از نام‌گذاری هر گره، که شکلی شبیه به یک درخت خواهد گرفت، هر کدام از خوشه‌ها به طور خاص مدیریت خواهند شد.
- بهبود خطا به صورت اتوماتیک: همان‌طور که در بخش قبل نیز نشان داده شد، در صورت بروز هر گونه خطا در سیستم این خطا به صورت خودکار رفع خواهد شد تا وظایف تکمیل شوند.



۵,۲ فواید آپاچی زوکیپر

همان‌طور که در شکل ۱۹ نشان داده شده است، در این بخش به ۵ تا از مهم‌ترین فواید آپاچی زوکیپر اشاره خواهد شد. این ۵ فایده شامل سادگی^۱، قابلیت اطمینان^۲، ترتیب^۳، سرعت^۴ و مقیاس پذیری^۵ می‌باشد.



۱,۵,۲ سادگی

همان‌گونه که بین اجزای مختلف به سادگی و با کمک فضای نام سلسله مراتبی انجام خواهد شد. این فضای نام‌گذاری سلسله مراتبی در بخش ۶,۲ توضیح داده خواهد شد.

^۱ Simplicity
^۲ Reliability

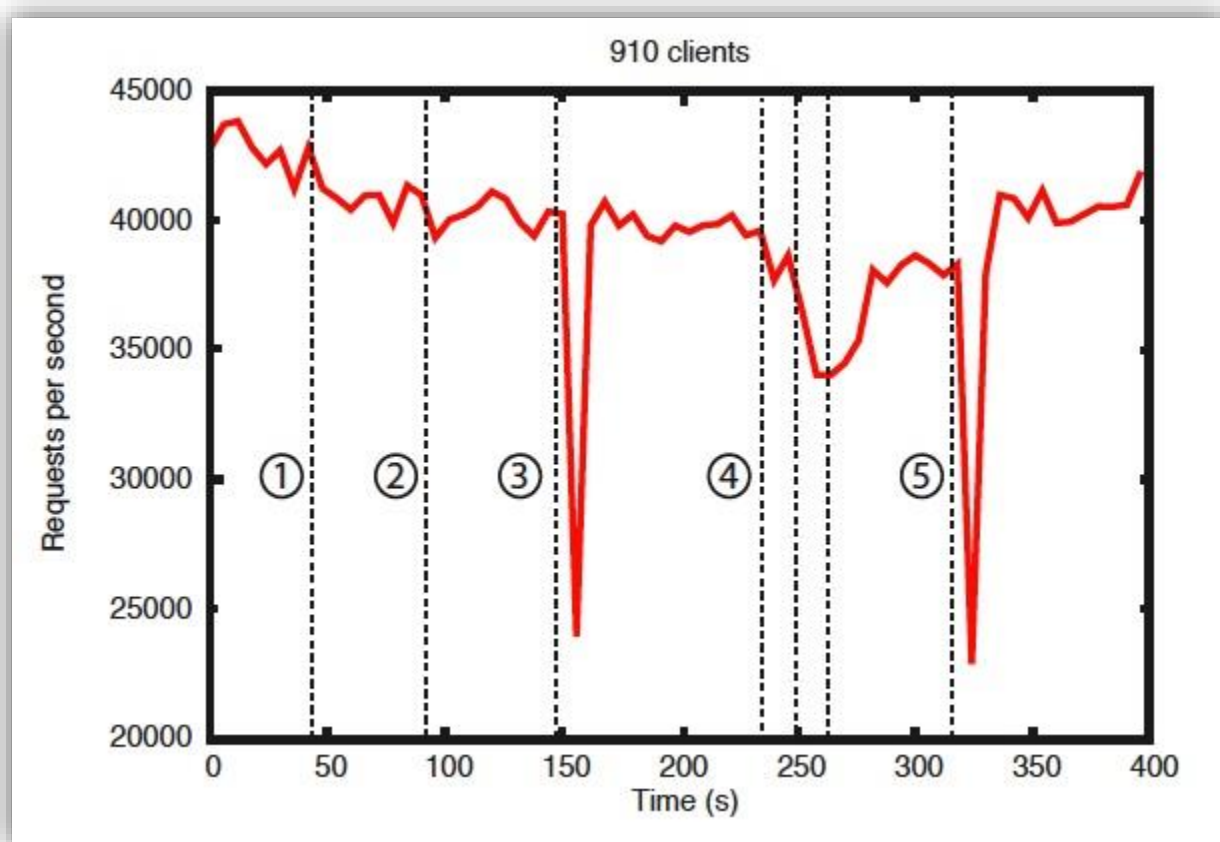
^۳ Order
^۴ Speed
^۵ Scalability

۲,۵,۲ قابلیت اطمینان

سیستم همچنان به کار خودش ادامه خواهد داد حتی اگر نصف گره‌ها از کار افتاده باشند. در واقع فقط کافیست که تعداد $1 + \left\lceil \frac{n}{2} \right\rceil$ از گره‌ها فعال باشند. در رابطه با بررسی صحت این ادعا آزمایشی انجام شده است که در آن ۹۱۰ مشتری به همراه ۷ سرور وجود دارند. از این ۷ سرور ۲ سرور رهبر و ۵ سرور دنبال‌کننده می‌باشند. آزمایش انجام شده به این صورت بود که در زمان‌های خاصی یک سری از این سرورها به صورت عامدانه از کار خواهند افتاد. با انجام این عمل می‌خواهیم نحوه کار آپاچی زوکیپر را متوجه شویم و ببینیم که آیا می‌توان به این سرویس اعتماد کرد یا خیر؟

طبق شکل ۲۰ می‌توان زمان‌های ایجاد مشکل در سرورها را متوجه شد. در ۵ زمان متفاوت اغتشاش‌هایی به مدل وارد شده و نتایج حاصل از این اغتشاش در زیر آمده است:

- (۱) خراب کردن و بهبود بخشیدن یک سرور دنبال‌کننده که مشکل خاصی را ایجاد نکرده است.
- (۲) خراب کردن و بهبود بخشیدن یک سرور دنبال‌کننده دیگر که مشکل خاصی را ایجاد نکرده است.
- (۳) خراب کردن و بهبود بخشیدن یک سرور رهبر که سیستم را با مشکل روبه‌رو کرده است.
- (۴) خراب کردن و بهبود بخشیدن دو سرور دنبال‌کننده که سیستم دچار مشکل جزئی شده است ولی همچنان به کار خود ادامه می‌دهد.
- (۵) خراب کردن و بهبود بخشیدن یک سرور رهبر دیگر که سیستم را با مشکل روبه‌رو کرده است.



۳,۵,۲ ترتیب

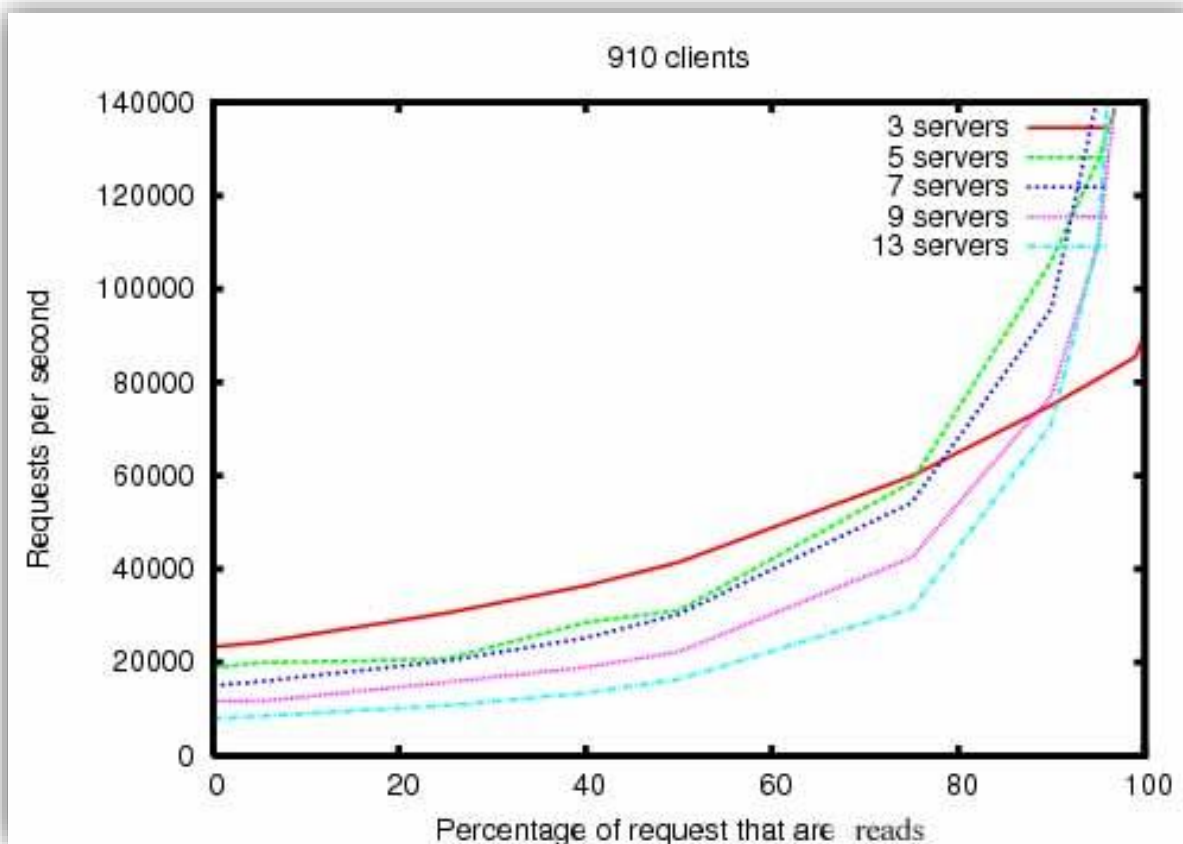
با مهر زدن به هر کدام از بهروز رسانی‌ها با شماره‌ای که نشان‌دهنده ترتیب آن است، این عمل انجام می‌شود. هدف از این کار ایجاد تمایز بین گره‌های مختلف و کارهایی است که هر کدام به ترتیب انجام می‌دهند.

۴,۵,۲ سرعت

در مواردی که عمل "خواندن" بیش‌تر از "نوشتن" و دیگر اعمال رایج باشد، این سیستم با یک نسبت ۱۰ به ۱ اجرا خواهد شد که این نشان‌دهنده سرعت فوق‌العاده این سرویس است.

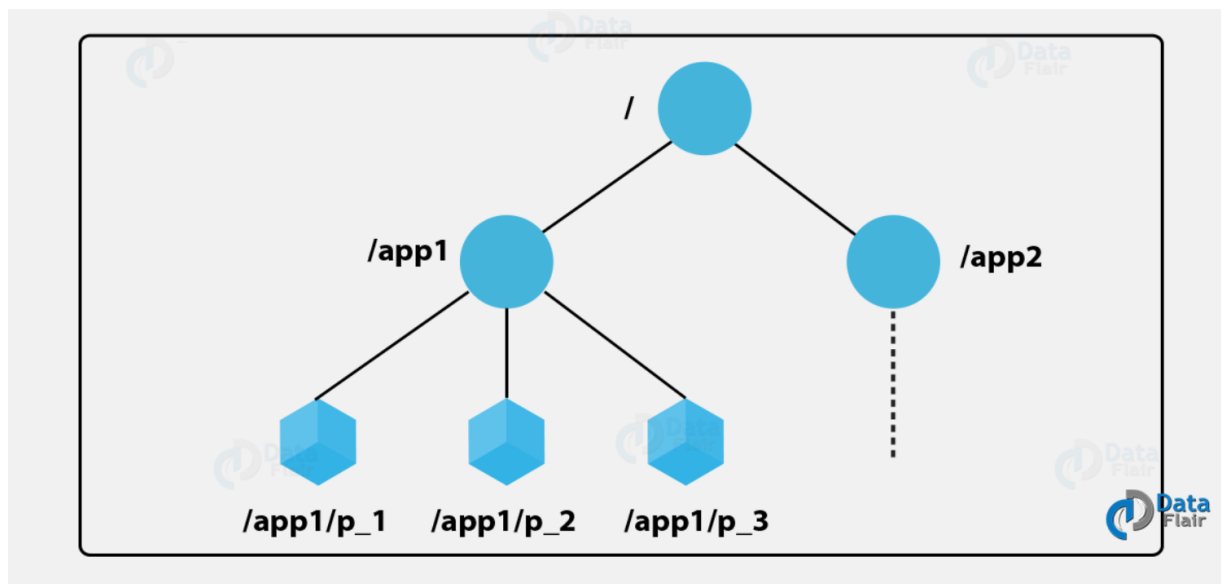
۵,۵,۲ مقیاس پذیری

عملکرد این سیستم را می‌توان با استقرار ماشین‌های بیش‌تر افزایش داد. صحت این ادعا نیز طبق یک آزمایش دیگر به اثبات رسیده است. همان‌طور که در شکل ۲۱ می‌توان دید در هر مرحله از آزمایش به ترتیب ۳، ۵، ۷، ۹ و ۱۳ سرور به کار گرفته شدند و وضعیت پاسخ‌دهی آن‌ها به وظایف یکسان، سنجیده شده است. نتایج حاصل از این آزمایش در شکل زیر به نمایش در آمده است.



۶،۲ مدل داده و فضای سلسله مراتبی نام

از ابتدا تا به اینجا کار، دفعات زیادی از سرویس نام‌گذاری و فضای سلسله مراتبی آن صحبت شد؛ ولی در حقیقت منظور چیست؟ در واقع در سیستم آپاچی زوکیپر هر گره یک نام مخصوص به خود را دارد که این نام در واقع یک دنباله است به صورت مسیری از عنصرهای مختلف که با یک / از یک‌دیگر جدا شده اند. در واقع می‌توان این‌طور گفت که هر گره یک دنباله شامل اسامی سرورهای بالای سر خود و خود آن گره دارد. یا به عبارت دیگر هر گره با یک مسیر تعریف خواهد شد. به طور مثال نحوه تعریف این مسیر در شکل ۲۲ نشان داده شده است.

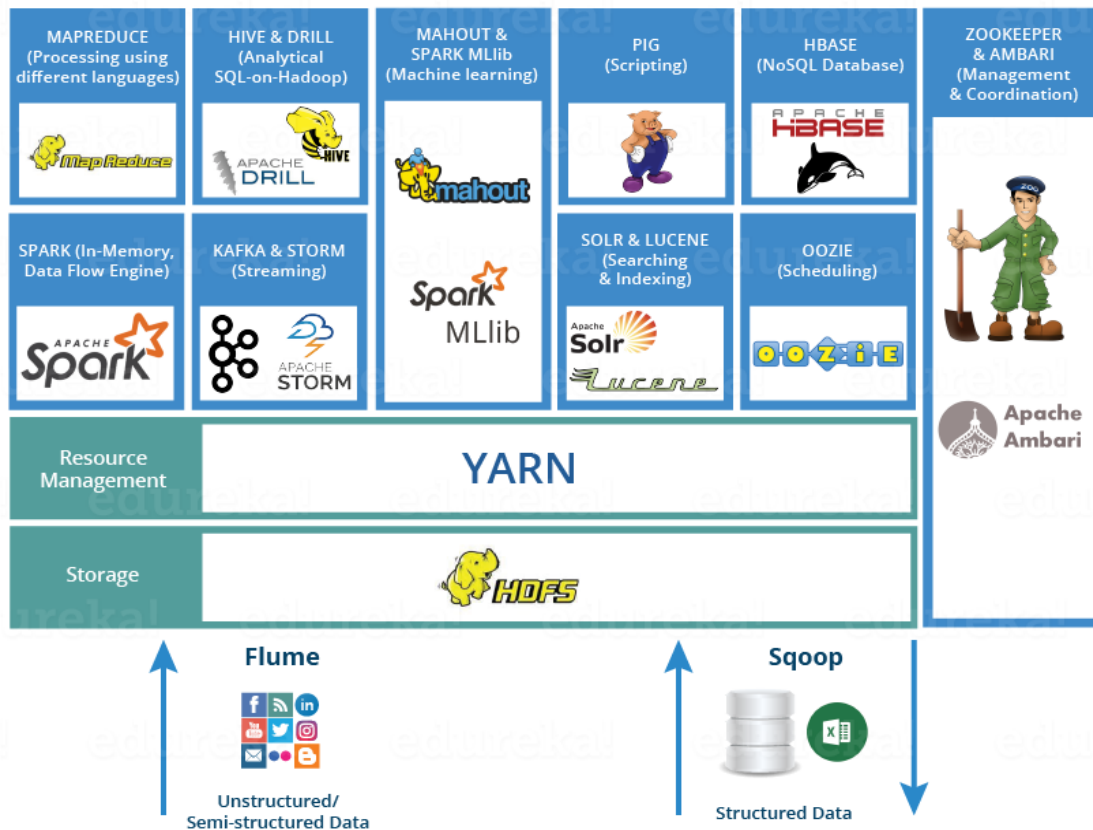


۳ یک نمای کلی از اکوسیستم هدوپ

در شکل ۲۳ می‌توان یک تصویر کلی از اکوسیستم هدوپ و اجزای مختلف آن داشت. همان‌طور که می‌توان دید آپاچی زوکیپر از ابتدا تا انتهای این اکوسیستم قرار دارد و آپاچی فلووم نیز وظیفه انتقال داده‌های بدون ساختار یا نیمه ساختار یافته را به این اکوسیستم بر عهده دارد.

اکوسیستم هدوپ بسیار گسترده می‌باشد به طوری که همه اجزای آن را نمی‌توان در یک یا حتی دو جلسه‌ی ۱ ساعت و ۳۰ دقیقه‌ای بررسی کرد. لذا در این ارائه صرفاً به معرفی دو مورد از اجزای مختلف این اکوسیستم پرداخته شد و در ارائه‌های دیگر نیز بقیه اجزا بررسی خواهند شد.

از جمله دیگر اجزای این اکوسیستم می‌توان به یارن (YARN)، آپاچی اسپارک (Apache Spark)، الگوریتم نگاشت-کاهش (Map Reduce)، کافکا (Kafka)، آپاچی استورم (Apache Storm)، آپاچی دریل (Apache Drill)، آپاچی هابو (Apache Hive)، ماشین لرنینگ در کلان‌داده یا همان اسپارک‌ایم‌لیب (Spark MLlib)، ماهوت (Mahout)، آپاچی پیگ (Apache Pig) و دیگر موارد اشاره کرد. [۱۲]



۴ مراجع

- [۱] <https://comco.computer/مقاله/کلان-داده/>
- [۲] <https://mta.co.ir/fa/uptodateslider/uptodatedetail/2325>
- [۳] <https://blog.faradars.org/big-data-tools/>
- [۴] <https://www.flume.apache.org>
- [۵] <https://logz.io/blog/apache-flume-and-data-pipelines/>
- [۶] https://www.tutorialspoint.com/apache_flume/apache_flume_environment.htm
- [۷] <https://www.zookeeper.apache.org>
- [۸] <https://hopetutors.com/blog/big-data/what-is-apache-zookeeper/>
- [۹] <https://cloudxlab.com/blog/introduction-to-apache-zookeeper/>
- [۱۰] <https://intellipaat.com/blog/what-is-apache-zookeeper/?US>
- [۱۱] <https://data-flair.training/blogs/zookeeper-architecture/>
- [۱۲] <https://www.edureka.co/community/1106/what-zookeeper-what-the-purpose-zookeeper-hadoop-ecosystem>

پیوست

کتابخانه لازم برای استفاده از ابزار آپاچی فلووم در پایتون.....

pyflume

کتابخانه لازم برای استفاده از ابزار آپاچی زوکیپر در پایتون.....

kazoo