بهبود کارایی کارگزار پیام جهت تحلیل مقیاس‌پذیر داده

نیلوفر خادمی[۱] ، رضا عزمی[۲]

[۱]دانشجوی ارشد مهندسی نرم‌افزار، گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه الزهرا،
Khademi1375niloofar@gmail.com

[۲] دانشیار گروه مهندسی کامپیوتر، دانشکده فنی و مهندسی، دانشگاه الزهرا،
azmi@alzahra.ac.ir

چکیده‌ـ پردازش بی‌درنگ داده‌ها و ارسال پیام‌ها به صورت توزیع‌شده، مشکلاتی هستند که مقالات متعددی برای حل آنها، منتشر شده است. با افزایش حجم داده‌های مکانی، نیاز به پلتفرم‌هایی وجود دارد که امکان تحلیل داده‌های مکانی را فراهم کند. در این پژوهش، یک سیستم پیام‌رسانی توزیع‌شده با کارایی بالا بر روی داده‌های مکانی پیشنهاد شده است و روی یک مطالعه موردی در مورد تحلیل مکانی مجموعه داده توییتر با استفاده از روش پرس‌وجوی خط افق[۱]، پیاده‌سازی شده است. در این پژوهش، مقایسه‌ای بین دو روش استفاده از کارگزار پیام پالسار و استفاده از کارگزار پیام کافکا و اسپارک جهت تحلیل داده‌های مکانی، صورت گرفته است. از منظر میزان بار پردازنده، مصرف پردازنده و مصرف حافظه به هنگام تحلیل داده‌های مکانی، دو معماری مذکور مورد ارزیابی قرار گرفته‌اند. نتایج نشان‌دهنده بهبود کارایی هنگام استفاده از معماری پیشنهادی می‌باشد.

کلمات کلیدی- پالسار ـ تحلیل داده مکانی‌ـ کارگزار پیام ـ کافکا ـ سیستم پیام‌رسانی توزیع‌شده

۱. مقدمه

در سال‌های اخیر داده‌ها به سرعت در حال رشد هستند. منابع متعددی مانند رایانه‌ها، رسانه‌های اجتماعی و تلفن‌های همراه حجم زیادی از داده‌ها را با فرمت‌های مختلف (ساختار یافته، نیمه ساختاریافته و بدون ساختار) تولید می‌کنند. مدل‌ها، سخت‌افزارها و فن‌آوری‌های مختلفی برای کلان داده‌ها[۲] ایجاد شده‌اند تا نتایج قابل اعتمادتر و دقیق‌تری ارائه کنند.

داده‌های بزرگ نیاز به جذب[۳]، پاکسازی، پردازش و استخراج اطلاعات مهم از داده‌ها دارند. فرآیند جذب داده یک مرحله مهم در ساخت هر پروژه کلان داده است. فرآیند جذب داده‌ها، باید با حجم، سرعت ، عدم قطعیت[۴] و تنوع متفاوت داده‌ها انجام شود. جذب داده‌ها می‌تواند به صورت دسته‌ای[۵] یا جریانی[۶] باشد.[1] هدف برنامه‌های مدرن، ارائه مدلی برای پردازش و تصمیم‌گیری بلادرنگ است. در این مدل، زمانی که داده‌های جدید وارد می‌شوند، بلافاصله برای پردازش منتقل می‌شود. [2]هدف از این پژوهش ارائه معماری مناسب جهت پردازش بلادرنگ داده‌ها می‌باشد. ابزارهای متفاوتی جهت جذب داده‌ها به صورت جریانی موجود می‌باشد. در این پژوهش، از دو ابزار شناخته‌شده در این حوزه استفاده شده است. در چرخه کلان داده‌ها، پس از مرحله جذب داده‌ها به مرحله آماده‌سازی داده‌ها[۷] می‌رسیم که هدف آن پاکسازی، اعتبارسنجی و کاهش داده‌های جذب شده است. ابزارهای متعددی نیز جهت آماده‌سازی داده‌ها وجود دارند مانند Hive، Impala، Storm و Spark که در این پژوهش، از ابزار اسپارک استفاده شده است.

---

[۱] skyline query
[۲] Big data
[۳] ingest
[۴] uncertainty
[۵] batching
[۶] streaming
[۷] Data preparation

حجم داده‌های مکانی[۸] در دسترس به شدت افزایش یافته است. چنین داده‌هایی شامل نقشه‌های آب‌وهوا، داده‌های اجتماعی‌ـ اقتصادی، شاخص‌های پوشش گیاهی، داده‌های برچسب‌گذاری‌شده جغرافیایی رسانه‌های اجتماعی[۹] و موارد دیگر می‌شوند، اما به این موارد محدود نمی‌شود. درک داده‌های مکانی برای چندین کاربرد مفید خواهد بود که ممکن است علم و جامعه را متحول کند ـ به عنوان مثال: تحلیل تغییرات آب و هوا، مطالعه جنگل زدایی ، مهاجرت جمعیت ،کمک به دولت در برنامه‌ریزی شهری/ منطقه‌ای، طراحی شبکه راه و ترافیک. این برنامه‌ها به یک پلتفرم مدیریت داده قدرتمند برای مدیریت داده‌های مکانی نیاز دارند.[3] در این پژوهش، از داده برچسب‌گذاری شده جغرافیایی توییتر استفاده شده است تا راهکاری مناسب، جهت تحلیل داده‌های مکانی در سیستم‌های پیام‌رسانی توزیع شده، ارائه شود.

در این پژوهش، در ابتدا در بخش دوم با مقالات پیشینی که در این خصوص پیاده‌سازی گردیده‌اند، آشنا می‌شویم. سپس در خصوص داده مورد استفاده در این پژوهش و معماری پیشنهادی در بخش سوم صحبت می‌کنیم و در آخر نیز در بخش چهارم، نتایج و ارزیابی‌ها را بررسی می‌کنیم. بخش پنجم نیز به نتیجه‌گیری می‌پردازیم.

۲. پیشینه پژوهش

از جمله مقالاتی که در این حوزه کار شده است، مقاله [4] می‌باشد که هدف اصلی مقاله، پیشنهاد یک معماری پیشرفته و گردش کار بر اساس پلتفرم‌های کلان داده Apache Hadoop و Apache Spark است تا امکان جمع‌آوری، ذخیره، پردازش و تحلیل داده‌ها از جریان‌های رسانه‌های اجتماعی را داشته باشد. این مقاله روی مطالعه موردی تشخیص هشدار سیل در مناطق خاص با استفاده از تحلیل توییت‌های توییتر اجرا شده است. تکنیک‌های تجزیه و تحلیل متن ثابت کرد که پیام‌های توییتر حاوی اطلاعات مکانی ارزشمند مرتبط با سیل می‌باشند.

در مقاله [5] ، یک چارچوب عمومی برای تکمیل چرخه داده برای سیستم آب ارائه شده است. ابزار تحلیل داده Spark جهت یکپارچه سازی داده‌های اندازه‌گیری شده با فرکانس بالا با مدل توزیع آب، استفاده شده است. با به روزرسانی مدل در زمان واقعی، تحلیل دقیق‌تر است و می‌تواند تفسیرهای نادرست جدی را آشکار کند. مقاله [6]، معماری کلان داده و فرایند مورد استفاده جهت دانلود توییت‌های برچسب‌گذاری شده جغرافیایی، ذخیره آنها در پایگاه داده MongoDB، تجزیه و تحلیل آنها با استفاده از Apache Spark و نمایش نتایج را ارائه می‌دهد. این مقاله نشان می‌دهد که چگونه تحلیل توییت‌ها می‌تواند برای به تصویر کشیدن رویدادهای شهر و کشف ویژگی‌های مکانی و زمانی آنها استفاده شود. در نهایت، به این نتیجه می‌رسد که شهر والنسیا دارای زیرساخت‌های شهری مناسب، جهت پشتیبانی از رویدادها می‌باشد.

مقاله [7]، رویکرد جدیدی را برای شناسایی اخبار جعلی در توییتر در طول دوره Covid-19 ارائه کرده است. روش پیشنهادی شامل رویکرد دسته‌بندی است که از ویژگی‌های توییت‌های جدید استفاده می‌کند و مبتنی بر پردازش زبان طبیعی[۱۰]، یادگیری ماشین و یادگیری عمیق است. این روش به صورت موازی با اسپارک اجرا می‌شود. نتایج تجربی نشان می‌دهد که این رویکرد، با استفاده از الگوریتم جنگل تصادفی با دقت ۷۹ درصد نتایج بسیار ارزشمندی به دست می‌آورد. در مقاله دیگری [8] ، یک چارچوب ذخیره‌سازی برای مدیریت هر دو پایگاه داده SQL و NOSQL به نام (COVID-QF) برای مجموعه داده‌های COVID-19 به منظور درمان و رسیدگی به مشکلات ناشی از انتشار ویروس در سراسر جهان با کاهش زمان درمان، پیشنهاد می‌کند. در مورد پایگاه داده NoSQL، COVID-QF از Hadoop HDFS/Map Reduce و Apache Spark استفاده می‌کند.این معماری از لایه‌های جمع‌آوری داده، ذخیره‌سازی و پردازش پرس‌وجو تشکیل شده است. در مقاله [9]، سیستمی را برای شناسایی افراد در جریان‌های ویدئویی در زمان واقعی، محاسبه فاصله اجتماعی آنها و گزارش نتایج با استفاده از ابزارهای مختلف آپاچی از جمله کافکا، اسپارک، hive، Impala و ... پیشنهاد کردند. گزارش مناطقی که عدم رعایت فاصله اجتماعی در آن زیاد است به صورت بلادرنگ،ارائه می‌شود. مقاله دیگری [10]، پیشنهاد سیستم ذخیره‌سازی جامع برای داده‌های COVID-19 با استفاده از Apache Spark (CSS-COVID) که شامل سه مرحله است، مرحله درج و شاخص‌گذاری[۱۱]، مرحله ذخیره‌سازی و مرحله پرس‌وجو را ارائه می‌دهد. این معماری، امکان مدیریت و تجزیه و تحلیل موارد مختلف از جمله موارد مشکوک را فراهم می‌کند. استفاده از آپاچی اسپارک در CSS-COVID عملکرد مقابله با داده‌های بزرگ بیماری کرونا را که هر روز افزایش می‌یابند، بهبود می‌دهد. در مقاله [11]، یک میان‌افزار پیام‌رسانی انتشار-اشتراک مبتنی بر موضوع[۱۲] پیشنهاد می‌شود. این ابزار به کاربران

---

[۸] Spatial data

[۹] geo-tagged social media data

[۱۰] natural language processing

[۱۱] indexing

[۱۲] topic-based pub/sub messaging middleware

اجازه می‌دهد تا با استفاده از شبکه‌های کپسولی[۱۳] و ابرداده[۱۴]های آنها مانند جنسیت و سن، تصاویر پزشکی از قفسه سینه افراد را به عنوان COVID-19 یا غیر COVID-19 فیلتر کنند. بنابراین، میان‌افزار پیشنهادی فضای جستجوی کوچک‌تری را برای دستیابی به نتایج جستجو فراهم می‌کند.

در مقاله [12]، یک سیستم توزیع شده پیام‌رسانی سبک جهت مقابله با اپیدمی‌ها روی مجموعه داده برچسب‌گذاری شده توییتر پیشنهاد شده است. در این سیستم، سه مؤلفه اصلی عبارتند از ترجمه توییت‌های منطبق با محدوده تعریف‌شده توسط کاربر، شناسایی موجودیت نام[۱۵] در توییت‌ها و پرس‌وجوهای skyline. Apache Pulsarدر این پژوهش به عنوان کارگزار پیام[۱۶] استفاده شده است.

۳. ابزارها و روش‌ها

در این بخش در رابطه با نحوه جمع‌آوری داده ، معماری پیشنهادی و مقایسه این معماری با معماری پیشنهادی در مقاله [12] صحبت می‌کنیم.

۳-۱- *نحوه جمع‌آوری داده*

حجم داده‌هایی که در پروژه‌های کلان داده استفاده می‌شود بسیار زیاد است، همچنین منبع و فرمت داده‌ها به سرعت در حال تغییر هستند. رسانه‌های اجتماعی مهم‌ترین منبع داده‌ها می‌باشد، توییتر و فیسبوک مقدار بسیار زیادی از داده‌هایی مانند توییت‌ها، پروفایل‌ها و لایک‌ها را تولید می‌کند، این داده‌ها را می‌توان تحلیل کرد و اطلاعات ارزشمندی را ارائه کرد. فایل‌های گزارش[۱۷]منبع دیگری از این نوع داده‌ها می‌باشد، به‌عنوان مثال، کلیک‌های روی وب‌سایت خاصی را می‌توان به عنوان گزارش برای درک رفتار کاربر، تحلیل کرد. حسگرها و ماشین‌هایی مانند دستگاه‌های پزشکی، ابزارهای هوشمند و دوربین‌های جاده‌ای حجم زیادی از داده‌ها را تولید می‌کنند. داده‌های مکانی که توسط تلفن‌های همراه تولید می‌شوند منبع دیگری از داده‌ها هستند که می‌توانند توسط برنامه‌های کاربردی دیگر استفاده شوند.[2]

در این پژوهش، از داده شبکه اجتماعی توییتر استفاده شده است. جهت استفاده از داده‌های توییتر نیاز به API توییتر می‌باشد که توییتر محدودیت‌هایی در تعداد توییت‌های قابل استخراج دارد. برای ارزیابی عملکرد معماری پیشنهادی در پردازش جریانی، حجم زیادی از داده‌ها نیاز است. به همین علت از مجموعه داده GeoCOV19Tweets استفاده شده است که مقاله [13] آن را ارائه کرده است. برای این مطالعه، API جریان[۱۸] استفاده شده است. از ۲۰ مارس روزانه اکسلی از توییت‌های انگلیسی زبان که روی بیش از ۹۰ کلمه کلیدی در رابطه با کرونا فیلتر شده است و شامل توییت‌های با برچسب‌گذاری جغرافیایی است استخراج می‌شود و در وب سایت منتشر می‌شود.[۱۹]

برچسب‌گذاری جغرافیایی فرآیند قرار دادن اطلاعات موقعیت مکانی در یک توییت است. هنگامی که یک کاربر به برنامه توییتر اجازه می‌دهد تا از طریق یک سیستم موقعیت یابی جهانی (GPS) به موقعیت مکانی کاربر دسترسی داشته باشد، داده‌های مختصات جغرافیایی به ابرداده مکان توییت اضافه می‌شود.[14]

مجموعه داده هر روز با افزودن شناسه‌های توییت‌های جدید جمع‌آوری شده به روز می‌شود و روزانه فایل اکسل که شامل دو ستون شناسه توییت و امتیازات احساسات مربوط به توییت می‌باشد که با استفاده از TextBlob محاسبه شده است، منتشر می‌شود. نقشه‌ای از این امتیازات به صورت آنلاین منتشر می‌شود.[۲۰] مطابق با خط‌مشی توییتر، فقط شناسه‌های توییت به اشتراک گذاشته شدند. سایر اطلاعات توییت را می‌توان با hydrating دوباره ساخت.[13]

در این پژوهش، تمام اکسل‌ها از مارس ۲۰۲۰ الی سپتامبر ۲۰۲۲ دانلود و ترکیب شدند و سایر اطلاعات توییت‌ها از طریق API توییتر و ابزار Hydrator استخراج شدند. تعداد ۴۰۶،۶۸۸ داده استخراج شده است.

۳-۲- *معماری پیشنهادی*

معماری پیشنهادی، استفاده از ابزار آپاچی کافکا جهت جذب داده‌ها و استفاده از ابزار آپاچی اسپارک جهت تحلیل داده‌های مکانی می‌باشد.(شکل ۱) به جهت مقایسه عملکرد معماری

---

[۱۳] capsule network
[۱۴] metadata
[۱۵] Name entity recognition
[۱۶] Message broker
[۱۷] log file
[۱۸] Streaming API
[۱۹] https://ieee-dataport.org/open-access/coronavirus-covid-19-geo-tagged-tweets-dataset
[۲۰] https://live.rlamsal.com.np

پیشنهادی ، معماری پیشنهادی در مقاله [12] نیز پیاده سازی شده و مورد مقایسه قرار گرفته است.

شکل ۱ ـ معماری پیشنهادی

مطابق تصویر ۳ داده‌های استخراج شده از مجموعه داده توییتر (مطابق با توضیحات در بخش ۱ـ۳) به عنوان ورودی به تولید کننده[۲۱] کافکا داده می‌شوند و به صورت جریانی به موضوع ورودی[۲۲]فرستاده می‌شوند و سپس در اسپارک، داده‌ها به صورت جریانی از موضوع ورودی خوانده می‌شوند، تحلیل داده‌های مکانی به روش skyline query اجرا می‌شود و خروجی آن در موضوع خروجی در کافکا قرارگذاشته می‌شود و مصرف کنندگان[۲۳] می‌توانند به این موضوع مشترک شوند.

۱-۲-۳- *مقایسه آپاچی کافکا و آپاچی پالسار*

آپاچی کافکا پرکاربردترین فریم‌ورک پردازش جریانی است و با توجه به ماهیت انعطاف‌پذیر، تأخیرکم و دسترسی‌پذیری بالا، درک دلیل آن دشوار نیست. اما کافکا تنها ابزار پردازش جریانی نیست. آپاچی پالسار،جایگزین منبع باز دیگری برای کافکا است.[15]

برای توسعه سیستم، کافکا و پالسار مزایا و معایب مختلف دارند. پالسار کمتر از کافکا از منابع سیستم استفاده می‌کند. با توجه به این مسئله می‌توانیم پالسار را به عنوان یک سرور سبک وزن[۲۴] در نظر بگیریم. در حالی که پالسار یک چارچوب پردازش جریانی مانند Apache Storm یا Spark Streaming نیست اما برخی از ویژگی‌های پردازش جریانی سبک را با استفاده از تابع پالسار[25] ارائه می‌کند. کافکا یک کارگزار پیام می‌باشد و برای پردازش جریانی داده‌ها باید از آپاچی اسپارک استفاده کرد که با کافکا یکپارچه شده است.[16] در این پژوهش نیز به علت نیاز به طراحی یک سیستم پیام‌رسانی توزیع شده با کارایی بالا ، کافکا به عنوان کارگزار پیام پیشنهاد می‌شود و برای تحلیل داده‌های مکانی نیز آپاچی اسپارک که با کافکا قابل یکپارچه سازی است و کارایی بالایی دارد، پیشنهاد می‌شود.

۲-۲-۳- *ترکیب آپاچی کافکا و اسپارک*

Apache Spark یک چارچوب محاسباتی منبع باز، توزیع شده، سریع و خوشه‌ای برای حجم عظیمی از داده است. Apache Spark تا ۱۰۰ برابر سریعتر از Hadoop است زیرا یک موتور پردازش حافظه برای پردازش موازی در مقیاس بزرگ دارد. کتابخانه‌های Apache Spark عبارتند از Spark Core برای پردازش موازی و توزیع شده، Spark Streaming برای پردازش جریانی، Spark MLlib برای یادگیری ماشین، Spark SQL و GraphX برای پردازش گرافیکی که در این پژوهش از Spark Streaming استفاده شده است. Spark Streaming بعد از شکست گره‌ها[۲۶] بدون از دست دادن[۲۷] داده بازیابی می‌شود و به نسبت سایر چارچوب‌های پردازش جریانی مانند Storm، Flink و MapReduce بهتر عمل می‌کند. مزایای اصلی Spark Streaming استفاده آسان، توان عملیاتی بالا، تعادل بار، تاخیر کم، مقیاس‌پذیری بالا و تحمل خطا است.داده‌ها را می‌توان از چندین منبع مانند Apache Kafka، Apache Flume، Amazon Kinesis و غیره گرفت. در نهایت، جریان‌های داده‌های پردازش شده را می‌توان در پایگاه داده‌ها ذخیره کرد و توسط داشبوردهای آنلاین نمایش داد.[17]

۳-۲-۳- *تحلیل مکانی با استفاده از skyline query*

پردازش skyline query برای بسیاری از سناریوها که نیاز به تصمیم‌گیری چند معیاره برای تعریف قابل اعتمادترین نتایج دارند، استفاده می‌شود. روش skyline می‌تواند مجموعه داده بزرگی از نقاط را بگیرد و آن را فیلتر کند تا بر اساس مجموعه‌ای از معیارهای ارزیابی، تنها ارجح ترین آنها را استخراج کند. زیرمجموعه کوچکی که شامل آیتم‌های برگزیده می‌شود، مجموعه skyline یا مجموعه بهینه پَرِتو[۲۸] نام دارد. [18] در این پژوهش دو معیار تعداد موردعلاقه‌ها[۲۹] و تعداد دنبال‌کنندگان[۳۰] استفاده شده است که نیاز است تعداد مورد علاقه‌ها ماکزیمم و تعداد دنبال‌کنندگان مینیمم باشد.

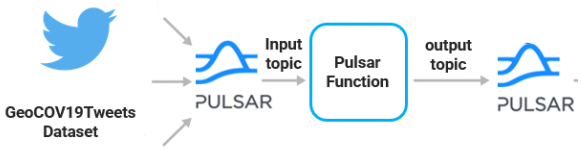
---

[۲۱] producer
[۲۲] Input topic
[۲۳] consumers
[۲۴] lightweight
[25] Pulsar Function

[۲۶] node failure
[۲۷] loss
[۲۸] pareto optimal set
[۲۹] favorite
[۳۰] follower

۳-۳- *معماری پیشنهادی در مقاله[12]*

همانطور که ذکر شد، مقاله [12] از پالسار استفاده کرده است. معماری پیشنهادی مقاله به صورت شکل زیر می‌باشد.



شکل ۲ـ معماری مقاله [12]

مطابق تصویر بالا، از مجموعه داده استخراج شده از توییتر به عنوان ورودی در تولیدکننده پالسار استفاده شده است. پیام‌ها به صورت جریانی در موضوع ورودی تولید می‌شوند سپس توسط تابع پالسار به جهت پردازش داده‌ها و استخراج داده‌های بهینه با روش skyline query دریافت می‌شوند و خروجی آن در موضوع خروجی قرار داده می‌شود تا مصرف‌کنندگان پالسار به آن مشترک شوند.

۴. ارزیابی

مشخصات سرور و ابزارها به شرح ذیل میباشد:

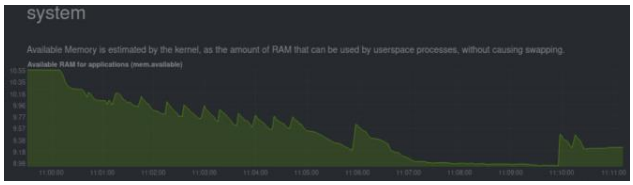
جدول ۱: **مشخصات سرور و ابزارها**

CPU	2
Memory	16GB
Ubuntu	20.04
Disk	102GB
Apache Pulsar	2.10.1
Apache Kafka	3.3.1
Apache Spark	3.3.1

به جهت ارزیابی عملکرد معماری‌ها، ۳ معیار مصرف پردازنده[۳۱]، بار پردازنده[۳۲] و مصرف حافظه[۳۳] در نظر گرفته شده‌اند. تمام ارزیابی‌ها با استفاده از ابزار ابری NetData[34] صورت گرفته است.

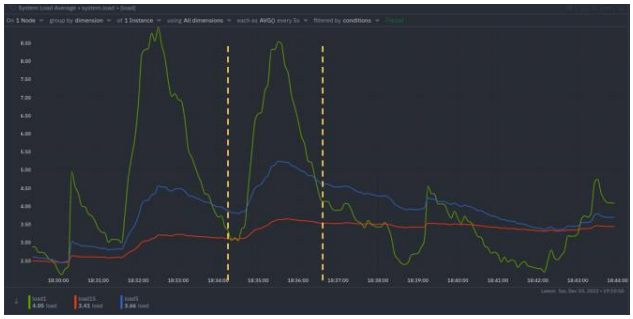
شکل ۳ـ بار پردازنده پالسار

شکل ۴ـ مصرف پردازنده پالسار



شکل ۵ـ مصرف حافظه پالسار

زمان اجرای استفاده از ابزار پالسار حدود ۱۱ دقیقه می‌باشد(از ساعت ۱۱:۰۰ الی ۱۱:۱۱). میزان مصرف پردازنده، بار پردازنده و مصرف حافظه نیز در این بازه زمانی در تصاویر بالا نشان داده شده است. زمان اجرای معماری پیشنهادی این پژوهش با استفاده از ابزارهای کافکا و پالسار حدود ۲ دقیقه می‌باشد(از ساعت ۱۸:۳۴ الی ۱۸:۳۶)که در تصاویر زیر با نقطه چین زرد رنگ مشخص شده است.



شکل ۶ـ بار پردازنده معماری پیشنهادی

[۳۱] CPU usage
[۳۲] CPU load
[۳۳] Memory usage
[34] https://learn.netdata.cloud

شکل ۷ـ مصرف پردازنده معماری پیشنهادی

شکل ۸ـ مصرف حافظه معماری پیشنهادی

نتایج مقایسه تصاویر ارزیابی دو معماری در جدول ۲ مشاهده می‌شود. همانطور که انتظار میرفت معماری روش اول برای روش‌هایی با مصرف کم منابع مناسب می‌باشد در حالیکه معماری پیشنهادی این پژوهش، عملکرد بالاتری دارد.

جدول ۲: **نتایج ارزیابی دو معماری**

معماری پیشنهادی	معماری مقاله [12]	معماری / معیار ارزیابی
۸/۵	۱۵	حداکثر بار مصرفی
نهایتا ۹۰	نهایتا ۱۰۰	مصرف پردازنده
ابتدا روی ۴ و سپس به سرعت تا ۳/۳ کاهش می‌یابد.	ابتدا روی ۱۰/۵و سپس به تدریج کاهش می‌یابد.	مصرف حافظه

۵. نتیجه‌گیری

با توجه به نتایج بدست آمده، انتخاب کافکا یا پالسار به هدف استفاده از کارگزار پیام، بستگی دارد. اگر کاربر نیاز به سیستمی با کارایی بالا و راحتی در استفاده دارد، استفاده از کافکا به کاربران توصیه می‌شود. از طرفی پالسار برای سیستم‌های سبک، توصیه می‌شود. ترکیب آپاچی کافکا و اسپارک به جهت جذب و پردازش داده‌های مکانی نتایج بهتری از منظر کارایی به نسبت استفاده از پالسار و تابع پالسار دارد. البته مصرف منابع در معماری پیشنهادی بیشتر از استفاده از پالسار می‌باشد. بنابراین در صورتی که نیاز به سیستمی سبک می‌باشد، استفاده از پالسار پیشنهاد می‌شود.

مراجع

[1] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, Samir Belfkih,, “Big Data technologies: A survey”, Journal of King Saud University - Computer and Information Sciences, Vol 30, Issue 4, pp 431-448, 2018
[2] Alwidian, Jaber & Rahman, Sana & Gnaim, Maram & Al-Taharwah, Fatima. ,”Big Data Ingestion and Preparation Tools”. Modern Applied Science, 2020
[3] Yu, J., Zhang, Z. & Sarwat, M. “Spatial data management in apache spark: the GeoSpark perspective and beyond”. Geoinformatica 23, 37–78 ,2019
[4] Podhoranyi, M. “A comprehensive social media data processing and analytics architecture by using big data platforms: a case study of twitter flood-risk messages”. Earth Sci Inform 14, 913–929 ,2021
[5] Shafiee ME, Barker Z, Rasekh A , “Enhancing water system models by integrating big data”. Sustain Cities Soc 37:485–491, 2018
[6] Martin A, Julian ABA, Cos-Gayon F ,” Analysis of twitter messages using big data tools to evaluate and locate the activity in the city of Valencia (Spain)”. Cities 86:37–50,2019
[7] Youness Madani, Mohammed Erritali, Belaid Bouikhalene,, “Using artificial intelligence techniques for detecting Covid-19 epidemic fake news in Moroccan tweets”,Results in Physics,Volume 25,2021
[8] Khashan EA, Eldesouky AI, Fadel M, Elghamrawy SM ,” A big data based framework for executing complex query over covid-19 datasets (covid-qf)”, 2020
[9] S. Melenli and A. Topkaya, "Real-Time Maintaining of Social Distance in Covid-19 Environment using Image Processing and Big Data," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020, pp. 1-5
[10] Elmeiligy MA, Desouky AIE, Elghamrawy SM ,” A multi-dimensional big data storing system for generated covid-19 large-scale data using apache spark”, 2020
[11] Eken, S. “A topic-based hierarchical publish/subscribe messaging middleware for COVID-19 detection in X-ray image and its metadata”,Soft Comput ,2020
[12] Özgüven, Y.M., Eken, S. “Distributed messaging and light streaming system for combating pandemics”. J Ambient Intell Human Comput, 2021
[13] Lamsal, R. “Design and analysis of a large-scale COVID-19 tweets dataset”. Appl Intell 51, 2790–2804 ,2021
[14] Bennett NC, Millard DE, Martin D, “Assessing twitter geocoding resolution”. In: Proceedings of the 10th ACM Conference on Web Science, pp 239–243,2018
[15] Haines, S, Apache Kafka and Spark Structured Streaming. In: Modern Data Engineering with Apache Spark. Apress, Berkeley, CA, pp 365–404,2022
[16] S. Intorruk and T. Numnonda, "A Comparative Study on Performance and Resource Utilization of Real-time Distributed Messaging Systems for Big Data," 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2019, pp. 102-107
[17] M. T. Tun, D. E. Nyaung and M. P. Phyu, "Performance Evaluation of Intrusion Detection Streaming Transactions Using Apache Kafka and Spark Streaming," 2019 International Conference on Advanced Information Technologies (ICAIT), 2019
[18] Y. Gulzar, A. A. Alwan and S. Turaev, "Optimizing Skyline Query Processing in Incomplete Data," in IEEE Access, vol. 7, pp. 178121-178138, 2019