Performance Evaluation of Database Models in Social Network Systems

Mojan Safaeinezhad
M.S. in Computer Engineering;
Department of Computer Engineering; Golestan University;
Gorgan, Iran Email: mojansafaei@gmail.com

Soheila Karbasi*
PhD in Computer Engineering; Assistant Professor;
Department of Computer Engineering; Golestan University;
Gorgan, Iran Email: s.karbasi@gu.ac.ir

Iranian Journal of Information Processing and Management

Received: 01, Feb. 2023 **Accepted: 16, Aug. 2023**

Iranian Research Institute for Information Science and Technology (IranDoc)
ISSN 2251-8223
eISSN 2251-8231
Indexed by SCOPUS, ISC, & LISTA
Vol. 39 | No. 2 | pp. 627-656
Winter 2024
https://doi.org/10.22034/jipm.2023.707244

Abstract: The usage of social networks has grown significantly in the past few years, and the result of this growth is the production of a large amount of data. On the other hand, it is necessary to effectively store and analyze social network data for all kinds of businesses today. In this research, the data models of widely used social networks, their advantages and disadvantages, as well as the things that need to be taken into account for storing and retrieving social network data have been reviewed and presented. The types of common storage methods based on the relational data model are not scalable for large amounts of information and therefore are not optimal and efficient. As a result, the use of storage methods based on non-relational models along with relational models have become very common. In this research, a large dataset of Twitter social network data, which includes 1,581,468 tweets from posts sent by 300,000 Persian users of this social network, are stored in three databases: MySQL, MongoDB, and Redis, and the performance of each of them with several different queries has been examined and compared. In addition to measuring the time spent to perform operations on the data, the amount of space occupied by the databases was also examined. Considering the advantages and limitations associated with the structure of all three databases, from the obtained values, it can be concluded that if the data has a specific structure, a relational database like MySQL is a good choice. If the data is unstructured or structured with the potential for rapid growth, NoSQL models will perform better. In addition, storing data temporarily and at high speed in key-value databases such as Redis, which store data in memory, are more suitable. Also, in terms of execution time, MongoDB executes queries very fast compared to MySQL and Redis, which is a proof that NoSQL databases show better performance and scalability for

* Corresponding Author

most operations in large datasets.

Keywords: Social Networks, Database, Big Data model, Scalability

ارزیابی عملکرد مدل‌های پایگاه داده در سیستم‌های شبکه‌های اجتماعی

موژان صفائی‌نژاد
کارشناسی ارشد کامپیوتر؛ دانشکده فنی و مهندسی؛ دانشگاه گلستان؛ گرگان، ایران؛
mojansafaei@gmail.com

سهیلا کرباسی
دکتری مهندسی کامپیوتر؛ استادیار؛ گروه کامپیوتر؛ دانشکده فنی و مهندسی؛ دانشگاه گلستان؛ گرگان، ایران؛
پدیدآور رابط s.karbasi@gu.ac.ir

پژوهش‌نامهٔ پردازش و مدیریت اطلاعات

دریافت: ۱۴۰۱/۱۱/۱۲ | **پذیرش:** ۱۴۰۲/۰۵/۲۵ | مقاله برای اصلاح به مدت ۳۹ روز نزد پدیدآوران بوده است.

نشریهٔ علمی | رتبهٔ بین‌المللی
پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)
شاپا (چاپی) ۸۲۲۳-۲۲۵۱
شاپا (الکترونیکی) ۸۲۳۱-۲۲۵۱
نمایه در LISTA، ISC، و SCOPUS
jipm.irandoc.ac.ir
دورهٔ ۳۹ | شمارهٔ ۲ | ص‌ص ۶۲۷-۶۵۶
زمستان ۱۴۰۲
https://doi.org/10.22034/jipm.2023.707244

چکیده: در این پژوهش مدل‌های داده محبوب‌ترین شبکه‌های اجتماعی به همراه مزایا و معایب آن‌ها و همچنین مواردی که برای ذخیره‌سازی و بازیابی داده‌های شبکه‌های اجتماعی حائز اهمیت هستند، بررسی و ارائه شده است. کاربردهای شبکه‌های اجتماعی در چند سال گذشته رشد چشمگیری داشته‌اند. نتیجه این رشد، تولید حجم زیادی داده است. از سوی دیگر، ذخیره‌سازی و تجزیه‌وتحلیل داده‌های شبکه‌های اجتماعی برای انواع کسب‌وکارهای امروزی به‌طوری مؤثر ضروری هستند. انواع روش‌های ذخیره‌سازی رایج بر اساس مدل داده رابطه‌ای برای حجم بالای اطلاعات مقیاس‌پذیر نبوده و بنابراین، بهینه و کارآمد نیستند. در نتیجه، استفاده از روش‌های ذخیره‌سازی مبتنی بر مدل‌های غیررابطه‌ای به همراه مدل رابطه‌ای بسیار مرسوم شده‌اند. در این تحقیق، مجموعه داده حجیمی از داده‌های شبکه اجتماعی «توییتر» که شامل ۱۵۸۱۱۴۶۸ توییت از پست‌های ارسالی توسط ۳۰۰۰۰۰ کاربر فارسی‌زبان این شبکه اجتماعی است، در سه پایگاه داده MongoDB، MySQL و Redis ذخیره شده و عملکرد هر یک از آن‌ها با اجرای پرس‌وجوهای مختلف بررسی و مقایسه شده است. افزون بر اندازه‌گیری زمان صرف‌شده برای اجرای عملیات بر روی داده‌ها، میزان فضای اشغال‌شده توسط پایگاه‌های داده نیز بررسی شد. با توجه به مزایا و محدودیت‌های مرتبط با ساختار هر سه پایگاه داده، از مقادیر به‌دست‌آمده می‌توان نتیجه گرفت که اگر داده‌ها دارای ساختار مشخصی باشند، پایگاه داده رابطه‌ای مانند MySQL انتخاب خوبی است. اگر داده‌ها بدون ساختار و یا ساختاریافته با پتانسیل رشد سریع باشند، مدل‌های NoSQL کارایی بهتری خواهند داشت. این در حالی است که نگهداری

داده‌ها به‌صورت موقت و با سرعت زیاد در پایگاه‌های داده کلید-مقدار مانند Redis که داده‌ها را در حافظه نگهداری می‌کنند، مناسب‌تر هستند. همچنین از نظر زمان اجرا، MongoDB با صرف زمان کمتری در مقایسه با MySQL و Redis پرس‌وجوها را اجرا می‌کند و این دلیلی است بر اینکه پایگاه‌های داده NoSQL عملکرد و مقیاس‌پذیری بهتری برای بیشتر عملیات در مجموعه داده‌های بزرگ نشان می‌دهند.

کلیدواژه‌ها: شبکه‌های اجتماعی، پایگاه داده، مدل داده‌های بزرگ، مقیاس‌پذیری

۱. مقدمه

کاربرد و محبوبیت شبکه‌های اجتماعی مانند «فیس‌بوک» و «توییتر» در چند سال گذشته رشد چشمگیری داشته‌اند. نتیجه این رشد، تولید حجم انبوهی از داده است. از سوی دیگر، روش‌های مرسوم ذخیره‌سازی مبتنی بر مدل رابطه‌ای[1] برای مدیریت حجم زیاد اطلاعات، مقیاس‌پذیر و کارآمد نیستند. در نتیجه، روش‌های ذخیره‌سازی مبتنی بر مدل‌های غیررابطه‌ای در کنار مدل‌های رابطه‌ای مورد توجه زیادی واقع شده‌اند (Roein Fard, Bahrampour & Jahanshahi Nokandeh 2018).

به‌طور کلی، امروزه محبوب‌ترین شبکه‌های اجتماعی از جمله «اینستاگرام»، «فیسبوک» و «توییتر» با اهدافی چون سرگرمی، رونق مشاغل و کسب‌وکارهای مختلف و یا ارائه سرویس میکروبلاگ[2]، که اهدافشان ارتباط سریع و ساده افراد به‌وسیله پیام‌ها و آشنایی با تفکرات یکدیگر است، مورد استفاده عموم قرار گرفته‌اند. در نتیجه، روزانه میلیون‌ها پیام از طریق این شبکه‌های اجتماعی ایجاد می‌شوند که جمع‌آوری، ذخیره‌سازی و تحلیل این داده‌ها مسئله بسیار مهمی است (Liu et al. 2017).

در برنامه‌نویسی سنتی، پایگاه‌های داده به‌طور معمول، از نوع مدل رابطه‌ای هستند که برای سازماندهی، ذخیره اطلاعاتِ پایه و تراکنشی گزینه مناسبی هستند. MySQL[3] یک سامانه مدیریت پایگاه داده متن‌باز و رابطه‌ای است که توسط شرکت «اوراکل»[4] توسعه، توزیع، و پشتیبانی می‌شود. پایگاه داده MySQL مجموعه‌ای ساختاریافته از داده‌هاست و این داده‌ها ممکن است یک لیست خرید ساده، گالری تصاویر یا حجم وسیعی از اطلاعات در یک شبکه بزرگ باشد. یکی از مزایای این پایگاه داده منبع باز بودن و امکان تغییر

1. relational database management system (RDBMS)
2. microblogging service
3. MySQL Support Manual, MySQL Developers
4. Oracle

و شخصی‌سازی نمودن آن است. موتور ذخیره‌سازی MySQL به‌طور پیش‌فرض InnoDB است که از ویژگی‌های آن می‌توان به پشتیبانی کامل از تراکنش‌ها، امکان بازیابی در صورت آسیب‌دیدن جداول، قفل‌شدن در سطح رکورد و نیز قفل‌نشدن کل جداول در عملیات ثبت و ویرایش داده‌ها که باعث کارایی بالای این موتور می‌شود، اشاره کرد. این موتور همچنین، به‌منظور ذخیره‌سازی برای سیستم‌های چندکاربره ایدآل است.

مدل رابطه‌ای ذخیره‌سازی داده به مرور و با گذشت زمان و افزایش حجم داده‌ها دیگر پاسخگوی نیاز غول‌های نرم‌افزاری مثل «گوگل»، «آمازون» و «فیسبوک» نبود. پایگاه‌های داده رابطه‌ای به‌دلیل نوع ساختار خود، برای تحلیل داده‌های بزرگ، ناکارآمد و کُند هستند. مشکلات دیگری مثل عدم حمایت از همروندی خواندن و نوشتن با تأخیر اندک، محدودیت قابلیت توسعه‌پذیری و دسترسی بالا، افزایش هزینه مالی و عملیاتی با افزایش حجم داده‌ها و عدم توانایی کارآمد ذخیره‌سازی داده‌های بزرگ، این شرکت‌ها را به سمت پایگاه داده‌های غیررابطه‌ای یا NoSQL[1] سوق داد (Han, Haihong Le & Du 2011).

افزون بر این، به‌وسیله «اُآرام»[2]، پایگاه داده‌های NoSQL راحت‌تر می‌توانند با زبان‌های برنامه‌نویسی ارتباط برقرار کنند. متداول‌ترین تمایز بین پایگاه‌های داده NoSQL، نحوه ذخیره‌سازی و اجازه دسترسی به داده‌هاست. انواع مدل‌های ذخیره‌سازی پایگاه‌های داده NoSQL به‌صورت کلید-مقدار[3]، مبتنی بر اسناد[4]، مبتنی بر ستون[5] و مبتنی بر گراف[6] معرفی و طبقه‌بندی می‌شوند. در ادامه، خصوصیات انواع مدل‌های پایگاه داده NoSQL با توجه به اصول طراحی سیستم‌های پایگاه داده توزیع‌شده[7] و تئوری «سازگاری، دسترس‌پذیری و تحمل بخش‌پذیری»[8] به شکل خلاصه بیان شده است.

مدل ذخیره‌سازی کلید-مقدار، از ساده‌ترین و محبوب‌ترین انواع پایگاه داده NoSQL است. در این مدل، داده‌ها به‌صورت مجموعه‌ای از کلیدها و مقادیر، مدیریت و به شکل کارآمد و مقیاس‌پذیر مانند جداول هش توزیع‌شده[9] ذخیره می‌شوند (Davoudian, Chen & Liu 2018).

بر اساس تداوم دسترسی به داده‌ها، مدل ذخیره‌سازی کلید-مقدار را می‌توان به سه نوع زیر طبقه‌بندی کرد:

---

1. Not Only SQL (NSQL)
2. object-relational mapping (ORM)
3. key-value
4. document-store
5. wide-column
6. graph-store
7. sharding
8. consistency, availability, partition tolerance (CAP)
9. distributed hash tables

- درون-حافظه[1] مانند Memcache که با نگهداری داده‌ها در حافظه، دسترسی بسیار سریع به آن‌ها را فراهم می‌کند. این مدل برای استفاده از داده‌های موقت مثل ذخیره‌سازی نشست کاربر[2] بسیار مورد استفاده قرار می‌گیرد. همچنین به‌عنوان لایه حافظه پنهان[3] در سیستم‌های ابری که در انتظار پاسخ از واسط برنامه‌نویسی نرم‌افزار کاربردی[4] و یا پرسش از پایگاه داده هستند، استفاده می‌شود. برای مثال، سرورهای MySQL «فیسبوک» از Memcache برای لایه حافظه پنهان خود استفاده می‌کنند (Borthakur et al. 2011). همچنین Voldemort در لینکدین[5] به‌عنوان لایه حافظه پنهان بر روی پایگاه داده اصلی استفاده می‌شود (Auradkar et al. 2012)؛
- ذخیره‌سازی با کلید-مقدار ثابت، مانند Riak KV و Oracle NoSQL، با ذخیره‌سازی داده‌ها در HDD/SSD، دسترسی بسیار بالایی به اطلاعات پیش‌بینی‌نشده ارائه می‌دهند؛
- ذخیره‌سازی با کلید-مقدار ترکیبی، مانند Redis[6] و Aerospike ابتدا داده‌ها در حافظه نگهداری می‌شوند و سپس در صورت برآورده‌شدن برخی شرایط، برای ذخیره‌سازی دائمی به دیسک منتقل می‌شوند.

در ذخیره‌سازی به شکل سند، مقدار به‌عنوان یک سند رمزگذاری‌شده در قالب‌های استاندارد نیمه‌ساختاریافته مانند XML[7]، JSON[8] یا BSON[9] نشان داده می‌شود. این اسناد به‌طور کامل انعطاف‌پذیر هستند، زیرا امکان افزودن و یا حذف ویژگی‌های داده‌ها در زمان اجرا نیز وجود دارد (Davoudian, Chen & Liu 2018). این نوع ذخیره‌سازی برای داده‌هایی که می‌توان آن‌ها را در قالب یک سند مانند سیستم‌های مدیریت محتوا[10] و بسترهای وبلاگ‌نویسی نشان داد، مناسب است. به‌عنوان مثال، یک پست وبلاگ شامل ویژگی‌های مختلف مانند برچسب‌ها، نظرات، تصاویر و فیلم‌هاست که می‌توانند به‌راحتی در قالب یک سند نمایش داده شود. سیستم‌های نمونه در این دسته شامل MongoDB، Amazon DynamoDB، Couchbase، Apache CouchDB و ArangoDB هستند. تمرکز مدل‌های داده فوق در مورد موجودیت‌ها به‌صورت مقادیر دودویی و یا سطری در جداول چندبعدی یا اسناد است.

1. in-memory
2. session
3. cache
4. application programming interface (API)
5. Linkedin
6. Remote Dictionary Server
7. extensible markup language (XML)
8. JavaScript Object Notation (JSON)
9. Binary JSON (BSON)
10. content management system (CMS)

در پایگاه داده‌های مدل رابطه‌ای مانند SQL Server داده‌ها به‌صورت پیش‌فرض به‌صورت سطری ذخیره می‌شوند. اما در ذخیره‌سازی مبتنی بر ستون، داده‌ها ستونی ذخیره می‌گردند. در مقایسه با پایگاه داده‌های رابطه‌ای، این پایگاه داده‌ها برای تجمیع و جست‌وجو در یک ستون خاص، سرعت به مراتب بالاتری دارند، زیرا داده‌های شبیه به‌هم به‌ترتیب و کنار یکدیگر ذخیره می‌شوند و هنگام بازیابی پرس‌وجوها، فقط همان ستون بررسی می‌شود.

مقیاس‌پذیری و انعطاف‌پذیری بالا همراه با پشتیبانی از MapReduce برای پردازش موازی مجموعه داده‌های بزرگ، این نوع ذخیره‌سازی را برای برنامه‌های تحلیلی مناسب می‌کند (Davoudian, Chen & Liu 2018). سیستم‌های نمونه در این دسته شامل Apache Cassandra ، Apache Hbase، Hypertable و Google Cloud Bigtable هستند.

با این حال، افزایش روزافزون مجموعه داده‌های مبتنی بر گراف، نیاز به پیمایش کارآمد روابط را ایجاد کرده است. بنابراین، پایگاه داده‌های گراف را می‌توان به‌سادگی به‌عنوان راهی برای نمایش و ذخیره داده‌ها توصیف کرد. به‌طور خلاصه، سه مزیت اصلی این مدل داده کارایی بالا، انعطاف‌پذیری و چابکی است. AllegroGraph، ArangoDB، InfiniteGraph، Neo4J و OrientDB نمونه‌هایی از پایگاه داده‌های مبتنی بر گراف هستند (Fernandes & Bernardino 2018). «فیسبوک» یک ساختار ذخیره‌سازی داده توزیع‌شده بر پایه گراف با نام تائو[1] را معرفی نموده که در آن برای هر گره، لیستی از انجمن‌ها به‌ترتیب نزولی بر اساس زمان ذخیره می‌شوند (Bronson et al. 2013).

در مجموع، با توجه به کاربردهای هر نوع از پایگاه‌های داده NoSQL می‌توان به مقیاس‌پذیری، کارایی و دسترسی بالای آن‌ها اشاره نمود. معماری استفاده‌شده برای شبکه‌های اجتماعی به‌دلیل تنوع کاربری، میکروسرویس[2] است. این معماری به‌گونه‌ای است که هر یک از سرویس‌ها با توجه به عملکردی که دارند، از پایگاه داده خاصی استفاده می‌کنند. استفاده روزافزون از سیستم‌های توزیع‌شده و تنوع انتخاب، چالش‌های جدیدی برای ذخیره‌سازی و پردازش داده‌ها به‌وجود می‌آورد که انتخاب مدل داده مناسب را دشوار می‌کند. به همین خاطر، مفهومی تحت عنوان تئوری CAP مطرح شد که شامل سه پارامتر اصلی در مدیریت داده‌ها و استفاده از سیستم‌های توزیع‌شده است.

1. the associations and objects (TAO)
2. microservice

مطابق این تئوری نمی‌توان سیستم توزیع‌شده‌ای طراحی کرد که همزمان سه پارامتر سازگاری، دسترس‌پذیری و تحمل بخش‌پذیری در آن پیاده‌سازی شده باشد. پس انتخاب مناسب‌ترین مدل داده باید بر اساس نوع داده، اهداف سازمان و نحوه مدیریت داده‌ها صورت پذیرد.

MongoDB یکی از پایگاه داده‌های معروف NoSQL است که داده‌ها را با حفظ سازگاری و در عین حال، در دسترس بودن مدیریت می‌کند. Apache Cassandra نیز یک پایگاه داده منبع باز NoSQL است که مبتنی بر ستون است و داده‌ها را در یک شبکه توزیع‌شده ذخیره می‌کند. Cassandra در دسترس بودن و تحمل بخش‌پذیری را ارائه می‌دهد، اما نمی‌تواند سازگاری پیوسته داده‌ها را ارائه دهد. با این حال، Cassandra اجازه نوشتن در هر گره و در هر زمان را به مشتریان خود می‌دهد و این نوشتن در سریع‌ترین زمان ممکن، نهایت سازگاری[1] را فراهم می‌کند.

مفاهیم دیگری که در انتخاب و دسته‌بندی پایگاه‌های داده حائز اهمیت است، خصوصیات اسید ACID[2] و بیس BASE[3] هستند. ACID چهار ویژگی اصلی در پایگاه داده است که کمک می‌کند دستورات تراکنش به‌طور کامل، جامع و بدون تأثیر مخرب بر روی تراکنش‌های دیگر انجام و حفظ شوند. در بین پایگاه داده‌های موجود، معمولاً پایگاه داده‌های رابطه‌ای مثل MySQL، Oracle، SQL Server به‌طور کامل از ویژگی ACID پشتیبانی می‌کنند. در واقع، خاصیت ACID این امکان را می‌دهد که تراکنش‌های پایگاه داده، به‌صورت جامع و با حفظ سازگاری به‌صورت مطمئن اجرا شوند. به‌طور معمول، این پایگاه داده‌ها، خرابی‌های موجود در سیستم را ترمیم می‌کنند و یا هنگامی که چندین کاربر به‌صورت همزمان قصد تغییر یک رکورد خاص را داشته باشند، به‌صورت ۱۰۰ درصد انجام عملیات صحیح را تضمین می‌کنند. اما از آنجا که پشتیبانی از ACID برای پایگاه‌های داده بسیار پرهزینه است، بسیاری از پایگاه داده‌های غیررابطه‌ای به‌طور کامل از این ویژگی پشتیبانی نمی‌کنند ولی قابلیت توزیع‌شدگی و تحمل سربار زیادی را ارائه می‌دهند. در حقیقت، پایگاه داده‌های غیررابطه‌ای خصوصیت BASE را دارا هستند؛ به این معنا که همه ویژگی‌های ACID را ندارد، ولی در عوض قابلیت توزیع‌شدگی و تحمل سربار زیاد را ارائه می‌دهند (Han, Haihong Le & Du 2011).

1. eventual consistency
2. atomicity, consistency, isolation, durability (ACID)
3. basically available, soft state, eventual consistency (BASE)

در پژوهش حاضر دو مدل پایگاه داده کلید-مقدار و مبتنی بر سند از انواع پایگاه داده‌های غیررابطه‌ای و پایگاه داده MySQL، از نظر پارامترهای مختلف عملکردی توسط مجموعه داده حجیمی از توییت‌های برگرفته از پست‌های ارسالی کاربران فارسی‌زبان شبکه اجتماعی «توییتر» مورد ارزیابی واقع شده‌اند. در این راستا، موضوع پژوهش حاضر در قالب دو پرسش اصلی زیر مطرح است:

۱. کارایی مدل‌های پایگاه داده غیررابطه‌ای در عملیات مختلف بر روی داده‌های شبکه‌های اجتماعی به‌چه صورت است؟
۲. کدام مدل‌ها برای انتخاب پایگاه داده در شرایط مختلف کاربرد داده‌های شبکه‌های اجتماعی مناسب‌تر هستند؟

۲. پیشینه پژوهش

در سال‌های اخیر، پژوهش‌های زیادی در زمینه مدل‌های ذخیره‌سازی داده‌های حجیم انجام شده است. هر یک از این پژوهش‌ها با در نظر گرفتن برخی ویژگی‌های پایگاه‌های داده، انتخاب مدل مناسب پایگاه داده در ذخیره‌سازی داده‌های شبکه‌های اجتماعی را پیشنهاد کرده‌اند که در ادامه مهم‌ترین و جدیدترین آن‌ها به‌صورت خلاصه بیان می‌شود.

«ماتیو و کومار» در پژوهش خود تمرکز اصلی را بر روی چهار نوع پایگاه داده غیررابطه‌ای مختلف متعلق به شبکه‌های اجتماعی «فیسبوک»، «لینکدین»، «توییتر»، «مای‌اسپیس»، «فوراسکور»، «فلیکر» و «فرندفید» قرار داده‌اند که در آن به بررسی و مقایسه ویژگی‌هایی مانند مقیاس‌پذیری، کنترل همزمانی، ثبات در ذخیره‌سازی، در دسترس بودن هنگام تقسیم‌بندی، دوام تراکنش‌ها، زبان پیاده‌سازی، امکانات پشتیبانی پرس‌وجو و ویژگی‌های برنامه‌نویسی پرداخته شده است. در تجزیه‌وتحلیلی دقیق، ویژگی‌هایی مانند ذخیره‌سازی داده‌ها و بازیابی سریع پرس‌وجوهای پردازش‌شده و همچنین مقایسه‌ای در مورد زمان صرف‌شده در هنگام درج و خواندن اطلاعات شبکه اجتماعی «فیسبوک» در بخش‌های روابط دوستان، دوستان صمیمی، خانواده و مفاهیمی مانند زادگاه و محل کار ارائه شده است. نتایج حاکی از نقش اساسی پایگاه داده گراف در مقایسه با سایر پایگاه داده‌های غیررابطه‌ای است. همچنین مشخص شده که برای عملیات درج و خواندن، پایگاه داده Neo4J در مقایسه با سایر پایگاه داده‌های گراف عملکرد بهتری دارد (Mathew and Kumar 2015).

«شارما» و همکاران به بررسی داده‌های جغرافیایی که تلفن‌های هوشمند در توسعه خدمات مختلف شهروندمحور برای توسعه جامعه پایدار تولید می‌کنند، پرداخته‌اند. این داده‌ها توسط کاربران در وب‌سایت‌های مختلف رسانه‌های اجتماعی مانند «فیسبوک»، «توییتر» و دارای برچسب جغرافیایی تولید می‌شوند و به‌دلیل سه ویژگی حجم، تنوع و سرعت داده‌ها می‌توانند به‌عنوان داده‌های بزرگ در نظر گرفته شوند و نمی‌توان آن‌ها را در یک ساختار از پیش تعریف‌شده محدود جای داد. بنابراین برای مدیریت این داده‌های بزرگ جغرافیایی از سیستم مدیریت غیررابطه‌ای استفاده می‌شود. هدف این پژوهش انتخاب پایگاه داده غیررابطه‌ای مناسب برای نرم‌افزار GIS و مقایسه عملکرد دو پایگاه داده MongoDB به‌عنوان پایگاه داده مبتنی بر سند و Neo4J به‌عنوان پایگاه داده گراف برای پرس‌وجوهای ساده است. نتایج نشان دادند که پایگاه داده MongoDB عملکرد بهتری نسبت به پایگاه داده رابطه‌ای PostgresSQL و پایگاه داده غیررابطه‌ای Neo4J دارد که دلیل این امر را عدم استفاده از زبان پرس‌وجو Cypher دانستند (Sharma, Sharma & Bundele 2018).

«رامش و کومار» به بررسی پایگاه داده غیررابطه‌ای Cassandra که توانایی مدیریت حجم زیادی از داده‌ها به همراه دسترسی بالا، تحمل بالای خطا و سازگاری داده‌ها را دارد، پرداخته‌اند. این پایگاه داده، بر خلاف SQL و MongoDB از درخواست‌های موردی و خاص پشتیبانی نمی‌کند و از این‌رو، به یک مدل داده کارآمد نیاز دارد که عمدتاً توسط نوع پرس‌وجو مشخص می‌شود. در این پژوهش یک مدل داده مبتنی بر پرس‌وجو در Cassandra برای ذخیره توییت‌ها و حفظ جدول زمانی پیشنهاد شده که عملیات خواندن و نوشتن را به روشی کارآمد انجام می‌دهد (Ramesh & Kumar 2018).

«اونگو و کوسوما» به بررسی پایگاه داده‌های ترکیبی متشکل از MySQL و MongoDB در توسعه برنامه‌های تحت وب پرداختند و کارایی MySQL را با ترکیب MongoDB بررسی کردند. آن‌ها یک وب‌سایت شبکه اجتماعی ساده برای ارزیابی ایجاد کردند که شامل پنل چت و نمایش پروفایل بود و کاربران به شکل تصادفی تولید می‌شدند. مدل داده در این وب‌سایت‌ها شامل داده‌های با ساختار و بدون ساختار بود. نتایج نشان داد که با افزایش اندازه پایگاه داده، مدل ترکیبی کارایی بهتری ارائه می‌کند. روش ترکیبی، زمان نوشتنِ به نسبت بهتری نسبت به MySQL دارد، اما نوشتن در MySQL سازگارتر است. از نظر فضای ذخیره‌سازی، استفاده از CPU و RAM، پایگاه داده مدل ترکیبی MySQL و MongoDB از فضای دیسک کمتری نسبت به MySQL استفاده می‌کند (Ongo & Kusuma 2018).

«داس» و همکاران، پایگاه داده‌های مبتنی بر گراف Infinite Graph، Allegro Graph، OrientDB، Neo4J، MongoDB، StarDog، FlockDB، Bitsy، ArangoDB را مقایسه و تحلیل کردند. ویژگی‌های مورد بررسی شامل انعطاف‌پذیری، تقسیم‌پذیری، مقیاس‌پذیری، امکان پشتیبان‌گیری آنلاین، مدل‌سازی ترکیبی، پشتیبانی از پرس‌وجوهای خاص، پشتیبانی از معماری‌های چندسکویی و خصوصیات ACID هستند. طبق مطالعه انجام‌شده بین این پایگاه‌های داده، نتیجه نشان داد که Neo4J و ArangoDB بهترین عملکرد را ارائه می‌دهند. اما Neo4J به‌دلیل زبان جست‌وجوی قدرتمند Cypher، سادگی و انعطاف‌پذیری از مقبولیت بیشتری برخوردار است (Das et al. 2020).

«مدی و باروا» به مقایسه دو پایگاه داده رابطه‌ای MySQL و غیررابطه‌ای Neo4J پرداختند. زمان لازم برای درخواست‌های مختلف در MySQL و Neo4J برای ۱۰۰، ۳۰۰ و ۴۰۰ شیء داده محاسبه شد و نتایج نشان داد که هر دو پایگاه داده رابطه‌ای و پایگاه داده گراف، برای نشان‌دادن داده‌ها کارا هستند؛ اما برای ذخیره و بازیابی داده‌هایی که ارتباط زیادی با یکدیگر دارند، پایگاه داده‌های گراف نتایج بهتری دارند. به گفته دیگر، هنگام درج یک داده جدید در پایگاه داده گراف نیازی به بازسازی مجدد پایگاه داده نیست. همچنین پایگاه داده گراف برای استفاده در اهداف تجاری مانند توسعه شبکه اجتماعی، بازار سهام، موتورهای پیشنهاددهنده و مدیریت شبکه بسیار مناسب است (Medhi & Baruah 2017).

پژوهش «داوودی» و همکاران بر روی نقطه ضعف سرعت عملکرد MySQL متمرکز است و نحوه برخورد با این مسئله و چگونگی سرعت‌بخشیدن در مواجهه با داده‌های بیش از یک میلیون رکورد را توصیف می‌کند. افزون بر این، در این پژوهش نحوه بازیابی داده‌های بزرگ «فیسبوک» از پایگاه داده MySQL با سرعت و عملکرد بالا و دلیل آن ارائه شده است. زمان بارگذاری صفحه یکی از مهم‌ترین ویژگی‌ها برای هر توسعه‌دهنده وب و پایگاه داده است. به‌طور پیش‌فرض، MySQL از موتور ذخیره‌سازی InnoDB استفاده می‌کند که برای مدیریت تعداد زیاد رکورد بسیار کند است. این تحقیق روشی را برای کاهش زمان بارگذاری صفحه با استفاده از موتور ذخیره‌سازی MyISAM پیشنهاد نموده است (Dawodi et al. 2019).

پژوهش «فیلیپ و چگان» عملکرد پایگاه‌های داده MySQL و MongoDB را هنگام اجرای انواع پرس‌وجوهای درج، به‌روزرسانی و حذف داده بررسی و مقایسه نموده و نشان می‌دهد که MongoDB در اکثر پرس‌وجوها عملکرد بهتری دارد، ولی MySQL هنگام اجرای پرس‌وجوهای تراکنشی نتیجه بهتری نسبت به پرس‌وجوهای غیرتراکنشی ارائه می‌کند.

همچنین این پژوهش به اهمیت نمایه‌سازی[1] صحیح و تأثیر آن هنگام جست‌وجوی درخواست‌ها اشاره می‌کند. اما باید توجه کرد که نمایه‌سازی هنگام اصلاح یا تغییر داده‌ها در پایگاه داده MySQL عملکرد را کاهش می‌دهد (Filip & Čegan 2020).

در پژوهش «تانگ و فن» سرعت اجرای پنج پایگاه داده غیررابطه‌ای Redis، MongoDB، Couchbase، Cassandra، HBase با استفاده از ابزار ارزیابی به نام YCSB (Yahoo! Cloud Serving Benchmark) بررسی شده است. مطابق با فرایند استاندارد YCSB، ابتدا ۱۰۰۰۰۰ رکورد در این پایگاه‌های داده به‌طور خودکار بارگذاری شده و سپس درخواست‌های مختلف مانند ۱) ۵۰ درصد خواندن و ۵۰ درصد به‌روزرسانی، ۲) ۱۰۰ درصد خواندن، ۳) ۱۰۰ درصد به‌روزرسانی مورد ارزیابی قرار گرفتند. از آنجا که تمرکز بر جنبه‌های مختلف عملکرد NoSQL دشوار است، در این آزمایش فقط کل زمان بارگذاری داده‌ها و اجرای بارهای کاری مختلف برای مقایسه سرعت و کارایی پایگاه‌های داده بررسی شده است. سرانجام، Redis بهترین کارایی را نشان می‌دهد، ضمن اینکه در مواجهه با داده‌های بسیار بزرگ محدودیت‌هایی نیز دارد. همچنین پایگاه‌های داده مبتنی بر سند و مبتنی بر ستون، کارایی و مقیاس‌پذیری مطلوبی ارائه می‌کنند (Tang & Fan 2016).

«جوزه و آبراهام» با مقایسه عملکرد پرس‌وجوها و دستورات مختلف در دو پایگاه داده MySQL و MongoDB با استفاده از دو مجموعه داده مختلف با اندازه‌های متفاوت، بر نقاط مثبت پایگاه‌های داده NoSQL نسبت به پایگاه‌های داده رابطه‌ای در بررسی کلان داده‌ها پرداخته‌اند. این پژوهش با در نظر گرفتن شش مقیاس مختلف ۱۰۰۰۰، ۲۵۰۰۰، ۵۰۰۰۰، ۷۵۰۰۰، ۱۰۰۰۰۰ و ۱۲۵۰۰۰ از دو مجموعه داده D1 و D2 پرس‌وجوهایی را اجرا و بررسی و زمان اجرای این دستورات را مقایسه کرده است. در این آزمایش از محیط کاربری[2] MySQL و MongoDB studio3T استفاده شده است. نتایج نشان می‌دهند که در همه موارد پایگاه داده NoSQL نسبت به پایگاه داده‌های رابطه‌ای عملکرد بهتری دارد (Jose & Abraham 2020).

مرور پژوهش‌های مرتبط در این بخش نشان می‌دهد که بیشتر آن‌ها توسط پژوهشگران خارج از کشور صورت گرفته و مجموعه داده‌های مورد بررسی توسط آن‌ها به زبان فارسی نیستند. همان‌طور که گفته شد، امروزه با افزایش روزافزون حجم داده‌های ایجادشده در شبکه‌های اجتماعی و اهمیت زمان دستیابی به آن‌ها برای کاربران،

1. indexing
2. workbench

اهمیت کارایی مدل‌های ذخیره‌سازی داده‌ها و تکنیک‌های مورد استفاده در ذخیره‌سازی، مدیریت و بازیابی داده‌ها برای کاهش زمان دسترسی ضروری‌تر می‌شود. در راستای این هدف، از آنجا که تاکنون در حوزه ذخیره‌سازی داده‌های زبان فارسی، مقایسه‌ای در زمینه بررسی مدل‌های ذخیره‌سازی داده‌ها صورت نگرفته، در پژوهش حاضر از یک مجموعه داده واقعی و حجیم فارسی در بستر پیام‌رسان‌ها جهت بررسی تأثیر خصوصیات مدل‌های ذخیره‌سازی داده‌ها بر معیارهای ارزیابی کیفیت عملکرد پایگاه‌های داده استفاده شده است. همچنین به‌دلیل اینکه در حال حاضر، رویکردهای بررسی خصوصیات، مزایا و معایب مدل‌های مختلف پایگاه داده، هم از لحاظ سرعت دسترسی به داده‌های مورد نظر در انواع پرس‌وجوها و همچنین حجم مورد نیاز برای ذخیره‌سازی در مواجهه با روند روبه‌رشد تصاعدی داده‌های شبکه‌های اجتماعی بسیار متنوع و انتخابی است، بنابراین پژوهش حاضر به‌عنوان شروعی در این حوزه می‌تواند راهنمای خوبی جهت رفع چالش‌های مرتبط با انتخاب مدل مناسب‌تر در سیستم‌های شبکه‌های اجتماعی بومی محسوب گردد.

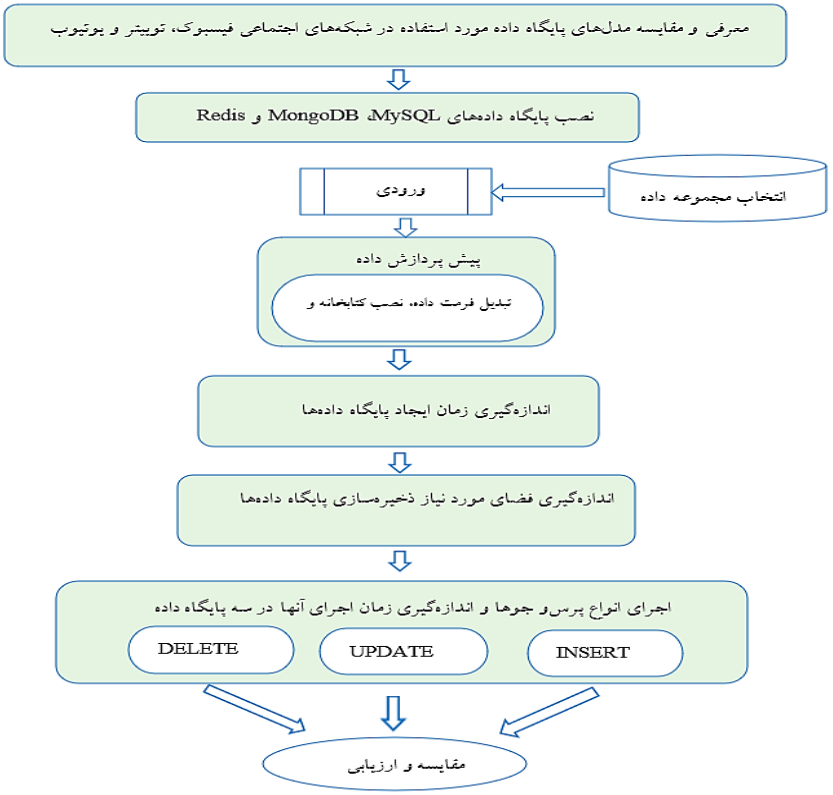
۳. روش پژوهش

پژوهش حاضر از نوع کاربردی است که به‌صورت تجربی و با رویکرد توصیفی انجام شده است. ابتدا جهت انتخاب مدل‌های داده و ارزیابی آن‌ها، مروری بر مدل‌های داده رایج در شبکه‌های اجتماعی که جزو دارندگان بیشترین تعداد کاربر محسوب می‌شوند، خواهیم داشت. «فیسبوک»، «توییتر»، «اینستاگرام» و «لینکدین» نمونه‌هایی هستند که با حجم زیادی از داده‌های در حال افزایش و مشکلاتی چون ذخیره‌سازی، بازیابی، بررسی و تجزیه‌وتحلیل آن‌ها روبه‌رو هستند (Bhogal & Choksi 2015). به‌طور مثال، شبکه اجتماعی «فیسبوک» و اعلان‌های آن به‌صورت بلادرنگ اجرا می‌شوند. اگر در نمایه یک کاربر تأیید «دوست داشتن»[1] رخ دهد، آن شخص در همان لحظه اعلان را دریافت می‌کند. بنابراین، «فیسبوک» به‌عنوان یکی از بزرگ‌ترین شبکه‌های اجتماعی، همواره تلاش می‌کند تا به نیاز کاربران خود با حداقل تأخیر پاسخ دهد.

گام‌های اجرایی پژوهش شامل ۸ مرحله اصلی است که در ادامه بیان شده و چارچوب آن نیز در شکل ۱، نشان داده شده است.

1. like

۱. معرفی و مقایسه مدل‌های پایگاه داده مورد استفاده در شبکه‌های اجتماعی بزرگ و پرطرفدار (فیسبوک، توییتر و یوتیوب)؛
۲. انتخاب پایگاه داده‌های مورد بررسی و نصب آن‌ها (خصوصیات آن‌ها در بخش ۳-۱ بیان شده‌اند)؛
۳. انتخاب مجموعه‌داده جهت آزمایشات؛
۴. پیش‌پردازش داده‌ها؛
۵. اندازه‌گیری زمان ایجاد پایگاه داده‌ها و عوامل مؤثر بر آن‌ها؛
۶. استخراج نتایج اجرای پرس‌وجوهای منتخب و اندازه‌گیری زمان اجرای آن‌ها در سه پایگاه داده؛
۷. اندازه‌گیری ظرفیت فضای مورد نیاز ذخیره‌سازی در سه پایگاه داده؛
۸. بحث، مقایسه و ارزیابی نتایج.



شکل ۱. چارچوب گام‌های اجرایی در پژوهش

فیسبوک

شبکه اجتماعی «فیسبوک» در اوایل سال ۲۰۰۴، توسط «مارک زاکربگ»[1] راه‌اندازی شد. معماری «فیسبوک» مبتنی بر سرویس[2] است که امکان استفاده از چندین پایگاه داده مختلف با ویژگی‌ها و مدل‌های داده مخصوص را ارائه می‌دهد. پایگاه داده اولیه مورد استفاده «فیسبوک» MySQL بود. InnoDB موتور ذخیره‌سازی پیش‌فرض MySQL است که در ابتدا توسط «فیسبوک» استفاده می‌شد. «فیسبوک» در سال ۲۰۱۶، موتور ذخیره‌سازی جدیدش، MyRocks را با هدف بهبود استفاده از فضای ذخیره‌سازی و کارایی نوشتن، فراتر از آنچه با InnoDB امکان‌پذیر بود، معرفی کرد. در شبکه «فیسبوک» تعداد خواندن داده از پایگاه داده نسبت به نوشتن به مراتب بالاتر است. اطلاعات کاربران و پست‌ها، ابتدا در پایگاه داده اصلی نوشته می‌شوند و سپس یک نمونه از آن در پایگاه داده Memcache، که از نوع کلید-مقدار در حافظه است، ذخیره می‌شود که موجب افزایش سرعت و جلوگیری از مراجعات زیاد به پایگاه داده اصلی و بهبود کارایی می‌شود (Gašpar and Mabić 2017). در سال ۲۰۱۳، «فیسبوک» با ارائه مسئله‌ای مبنی بر حجم بار سنگینی که با هر مراجعه بیش از یک میلیارد کاربر فعال در لحظه، از طریق مرورگر «دسکتاپ»[3] یا دستگاه تلفن همراه ایجاد می‌شد، به بررسی نقاط ضعف مدل ترکیبی پایگاه داده رابطه‌ای MySQL و لایه Memcache پرداخت که در نهایت، با ذخیره بخشی از داده‌ها در قالب API که با نیازهای «فیسبوک» مطابقت داشته باشد، موافقت شد. API اشیا و روابط آن‌ها (TAO) بر اساس مدل داده‌های گراف بود. این API، دسترس‌پذیری و کارایی سیستم را بر سازگاری قوی ترجیح داده و نقطه قوت آن مقیاس‌پذیری است که می‌توانست یک بیلیون درخواست در ثانیه را بر روی مجموعه داده‌های پویا با حجم پتابایت، با کارایی بالا و سرعت تأخیر کم پشتیبانی کند (Bronson et al. 2013).

پردازش و تجزیه‌وتحلیل داده‌ها برای «فیسبوک» که حدود ۲,۹۱ میلیارد کاربر فعال ماهانه دارد، امری ضروری است. Gorilla پایگاه داده سری زمانی دیگر «فیسبوک» است که عمدتاً در زیرساخت‌های نظارت و تجزیه‌وتحلیل استفاده می‌شود. این پایگاه داده هوشمند می‌تواند به انواع خرابی‌ها، با سربار کم و بدون هیچ‌گونه هزینه عملیاتی رسیدگی کند (Caers et al. 2013). برای ذخیره و مدیریت گزارش‌ها، «فیسبوک» از یک پایگاه داده

1. Mark Zuckerberg 2. polyglot persistence 3. desktop

توزیع‌شده و مقاوم در برابر خطا به‌نام LogDevice استفاده می‌کند. در جدول ۱، خلاصه‌ای از نام پایگاه‌های داده مورد استفاده «فیسبوک» و کاربردهای آن‌ها نشان داده شده‌اند.

جدول ۱. خلاصه‌ای از پایگاه داده‌های مورد استفاده فیسبوک

پایگاه داده	LogDevice	Apache Hadoop	TAO	Memcached	MySQL (MyRocks)
کاربرد	ذخیره‌سازی و مدیریت گزارش‌ها	تجزیه‌وتحلیل داده‌ها	ذخیره‌سازی گراف اجتماعی	ذخیره‌سازی cache	ذخیره‌سازی داده‌های اجتماعی

توییتر

«توییتر» یکی از شبکه‌های اجتماعی محبوب سراسر جهان است که کاربران آن متونی را به همراه عکس و یا ویدئو منتشر می‌کنند. معماری مدل داده «توییتر» نیز از نوع میکروسرویس است. در «توییتر» از Hadoop برای محاسبات و سیستم فایل توزیع‌شده[1]، پایگاه داده کلید-مقدار Manhattan برای ذخیره داده‌ها (مثل توییت‌ها، پیام‌های مستقیم، حساب‌های توییتر) و Gizzard/MySQL برای ذخیره گراف اجتماعی استفاده می‌شود. داده‌های حجیم مثل فیلم، عکس و فایل‌های باینری در فرمت Blob ذخیره می‌شوند. در این شبکه، پایگاه داده‌های رابطه‌ای MySQL، PostgreSQL و Vertica جهت سازگاری قوی، مدیریت کمپین‌های تبلیغاتی، تبادل تبلیغات در کنار ابزارهای داخلی مورد استفاده قرار می‌گیرند. همچنین معیارهای «توییتر» در پایگاه داده MetricsDB ذخیره می‌گردند. از آنجا که همواره حجم داده‌ها در این شبکه به‌طور تصاعدی در حال افزایش است، در سال ۲۰۱۴، Manhattan که یک پایگاه داده کلید-مقدار توزیع‌شده با قابلیت چند مستأجری است، برای ذخیره‌سازی توییت‌ها، اطلاعات کاربران و پیام‌ها مورد استفاده قرار گرفت. در سال ۲۰۱۷، این شبکه اجتماعی موتور جست‌وجوی خود را به RocksDB تغییر داد که موجب تأخیر کمتر، استفاده بهینه از فضا و پردازنده شد. «توییتر» از چندین فناوری مبتنی بر cache مانند Redis و Twemcache در مقیاس بسیار زیاد استفاده می‌کند. همچنین به‌تقریب تمام cacheهای اصلی برای کم‌کردن هزینه عملیاتی به Mesos منتقل شدند. Apache Mesos یک پروژه منبع باز است که حجم کار را در محیط توزیع‌شده از طریق به‌اشتراک‌گذاری منابع مدیریت می‌کند. در جدول ۲، اسامی پایگاه‌های داده مورد استفاده «توییتر» و کاربردهای آن‌ها نشان داده شده‌اند.

1. Hadoop distributed file system (HDFS)

جدول ۲. خلاصه‌ای از پایگاه داده‌های مورد استفاده «توییتر»

پایگاه داده	MetricsDB	MySQL/ PostgreSQL/ Vertica	Apache Mesos	Manhattan
کاربرد	ذخیره‌سازی معیارهای توییتر	سازگاری قوی، مدیریت کمپین‌های تبلیغاتی	ذخیره‌سازی cache	ذخیره‌سازی داده‌ها با تأخیر کم

یوتیوب

استریم ویدئو «یوتیوب»[1] با ۲,۲۹ میلیارد کاربر فعال بعد از «فیسبوک» دومین وب‌سایت محبوب جهان است[2]. «یوتیوب» در سال ۲۰۰۵، معرفی و در سال ۲۰۰۶، توسط «گوگل» خریداری شد. معماری «یوتیوب» از دو بخش اصلی شبکه تحویل محتوا[3] و پایگاه داده تشکیل شده است. ویدئوها یا محتواهایی که از نظر کاربران محبوب‌تر باشند، به CDN منتقل می‌شوند تا در مکان‌های مختلف تکرار و ذخیره شوند و ویدئوهای دیگر در سرورهای «یوتیوب» ذخیره می‌گردند. MySQL اولین پایگاه داده مورد استفاده «یوتیوب» است که به همراه Memcache برای ذخیره‌سازی cache و Zookeeper برای هماهنگی گره‌ها استفاده می‌شد. تصاویر کوچک ویدئوها که به بینندگان این امکان را می‌دهد که هنگام مرور «یوتیوب»، پیش‌نمایشی[4] از ویدئوها را ببینند، از معماری BigTable «گوگل» برای ذخیره‌سازی استفاده می‌کنند. در ابتدا، متاتگ‌ها[5]، نام کاربری، کلمه عبور، توضیحات ویدئو و عنوان در MySQL ذخیره می‌شدند. با توجه به افزایش درخواست‌ها، برای پاسخگویی به حجم زیادی از پرس‌وجوها در ثانیه، توسعه‌دهندگان تصمیم گرفتند که بعد از بهینه‌سازی پایگاه داده مثل استفاده از Memcache و بهینه‌کردن پرس‌وجوها، پایگاه داده رابطه‌ای MySQL را به‌صورت افقی مقیاس‌پذیر کنند. راه‌حل اول، ایجاد تکرارهایی از پایگاه MySQL به روش سرویس‌دهنده/ مشتری بود. بنابراین درخواست‌های خواندن از پایگاه داده به‌صورت موازی به سرویس‌دهنده‌ها ارسال می‌شدند تا بار اصلی را کاهش دهند. وجود چندین مرکز داده در سرتاسر جهان تأخیر سیستم را کاهش می‌داد. توسعه‌دهندگان به‌منظور کاهش پیچیدگی‌های سیستم و مقابله با چالش‌های مقیاس‌پذیری و کاهش هزینه، پایگاه داده Vitess[6] را توسعه دادند. Vitess بسیاری از ویژگی‌های مهم MySQL را با مقیاس‌پذیری یک پایگاه داده NoSQL ترکیب می‌کند. این پایگاه داده به‌طور خودکار از

1. YouTube stream video
2. https://www.dreamgrow.com/top-15-most-popular-social-networking-sites
3. content delivery network (CDN)
4. thumbnail
5. meta tags
6. https://vitess.io/

cache برای جلوگیری از رسیدن همزمان درخواست‌های تکراری به پایگاه داده استفاده می‌کند. هم‌اکنون، «یوتیوب» از PlanetScale که یک پایگاه داده NewSQL-as-a-service (DbaaS) در فضای ابری است و بر روی MySQL و Vitess ساخته شده، استفاده می‌کند. در جدول ۳، نام پایگاه داده‌های مورد استفاده «یوتیوب» و کاربردهای آن‌ها نشان داده شده‌اند.

جدول ۳. خلاصه‌ای از پایگاه داده‌های مورد استفاده «یوتیوب»

پایگاه داده	PlanetScale	Bigtable	Vitess	Memcached	MySQL
کاربرد	پایگاه داده آزمایشی اصلی	ذخیره‌سازی thumbnails	پایگاه داده اصلی برمبنای MySQL	ذخیره‌سازی cache	ذخیره‌سازی داده‌های اصلی

۳-۱. ویژگی مدل‌های داده مورد ارزیابی

با توجه به گوناگونی مدل‌های داده استفاده‌شده در شبکه‌های اجتماعی، برای ارزیابی و تست عملکرد با نمونه حجیمی از داده‌های واقعی، سه پایگاه داده MySQL، MongoDB و Redis از انواع رابطه‌ای و غیررابطه‌ای انتخاب شدند. همان‌طور که بیان شد، MySQL در دسته پایگاه‌های داده رابطه‌ای، MongoDB در دسته پایگاه داده‌های مبتنی بر سند و Redis در دسته پایگاه داده‌های کلید-مقدار غیررابطه‌ای و از نظر کارایی در لیست بهترین‌ها و از جنبه کاربرد، جزو پرکاربردترین‌ها قرار دارند. در ادامه، ویژگی‌های هر یک از این مدل‌های داده با جزئیات بیشتر، هم از لحاظ ساختار و هم از لحاظ عملکرد بیان شده است.

۳-۱-۱. پایگاه داده رابطه‌ای MySQL

MySQL یک سیستم مدیریت پایگاه داده رابطه‌ای منبع باز محبوب است که توسط شرکت Oracle توسعه، توزیع و پشتیبانی می‌شود[1]. پایگاه داده‌های رابطه‌ای MySQL داده‌ها را به شکل جدولی از سطرها و ستون‌ها ذخیره می‌کند و از زبان پرس‌وجوی ساخت‌یافته SQL برای دسترسی به داده‌ها استفاده می‌نمایند. در این پایگاه داده تعداد ستون‌ها برای تمام سطرهای جدول یکسان است و هر سطر جداول را می‌توان با یک شناسه یکتا به نام کلید اصلی علامت‌گذاری کرد که می‌تواند یک ویژگی یا ترکیبی از ویژگی‌ها باشد. در MySQL، ارتباطات جداول باید بر اساس الزامات و قوانین برای کنترل روابط بین ویژگی‌ها، از قبل تعریف‌شده باشد. اطلاعات مرتبط که در جداول مختلف ذخیره

1. https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

شده‌اند، توسط ارتباطات[1] به هم مرتبط می‌شوند. بنابراین، تکرار داده‌ها در این پایگاه داده به حداقل می‌رسد (Dipina Damodaran, Salim & Vargese 2016).

۳-۱-۲. پایگاه داده مبتنی بر سند MongoDB

MongoDB[2] یکی از شناخته‌شده‌ترین و پرکاربردترین پایگاه داده‌های مبتنی بر سند است که مقادیر مختلف به‌صورت مجموعه در آن ذخیره می‌گردند. این مقادیر به‌صورت سندهایی هستند که با اندازه‌های مختلف، انواع مختلفی از داده‌ها را در خود جای می‌دهند. این مسئله باعث شده که MongoDB بتواند داده‌هایی با ساختار پیچیده مانند داده‌های سلسله‌مراتبی را در خود ذخیره کند. در MongoDB از اسناد JSON و CSV به‌جای سطرها و ستون‌ها استفاده می‌شود. MongoDB به‌دلیل ساختار ذخیره‌سازی و با تکه‌کردن خودکار[3] داده‌ها، مقیاس‌پذیری افقی را به‌راحتی فراهم می‌نماید. در پایگاه داده MongoDB هر فیلد از اسناد با شاخص‌های اولیه و ثانویه شاخص‌گذاری می‌شود که این امر جست‌وجوی داده را آسان‌تر و سریع‌تر می‌کند. شاخص‌های MongoDB از ساختار داده B-tree جهت ذخیره داده‌ها استفاده می‌کنند. MongoDB از شاخص‌گذاری مناسب برای محدود کردن تعداد اسنادی که باید جست‌وجو کند، استفاده می‌کند و به‌صورت پیش‌فرض در حین ایجاد یک مجموعه، یک شاخص یکتا در فیلد id_ ایجاد می‌کند. همچنین مقایسه مجموعه‌های مختلف با استفاده از فناوری‌هایی مانند چارچوب تجمیع Map Reduce و سیستم‌های Hadoop انجام می‌شود. عملیات تجمیع[4] چندین سند را پردازش کرده و نتایج محاسبه‌شده را برمی‌گرداند (Jose & Abraham 2017).

روش پردازش پرس‌وجوی MongoDB به‌طور کامل از MySQL متفاوت است. در MongoDB، تمام پرس‌وجوهای نوشتن پس از دریافت توسط سرور MongoDB در صف قرار می‌گیرند و سپس پرس‌وجوها یک‌به‌یک توسط سرور پردازش می‌شوند. این مفهوم به‌عنوان قفل گسترده لحظه‌ای[5] شناخته می‌شود که در هر زمان تنها یک پرس‌وجوی مربوط به نوشتن پردازش می‌شود. اما امکان پردازش پرس‌وجوهای نوشتن از چندین پایگاه داده به‌صورت موازی وجود دارد. بنابراین، MongoDB پرس‌وجوها را بسیار سریع‌تر از MySQL پردازش می‌کند (Palanisamy & SuvithaVani 2020).

1. relations
2. https://www.mongodb.com/docs/manual/introduction/
3. auto sharding
4. aggregation
5. instance-wide-locking

۳-۱-۳. پایگاه داده کلید-مقدار Redis

پایگاه داده Redis، یک پایگاه داده غیررابطه‌ای است که داده‌ها را با سیستم کلید-مقدار در حافظه اصلی نگهداری می‌کند. با توجه به این ویژگی از آنجا که ارتباط پیچیده‌ای میان داده‌ها ایجاد نمی‌شود، دسترسی و بازیابی این اطلاعات بسیار ساده‌تر خواهد شد. این پایگاه داده، به‌طور معمول، به‌عنوان حافظه پنهان و واسط پیام[1] استفاده می‌شود. از آنجا که داده‌های موجود در حافظه‌های اصلی پایدار نیستند، ممکن است داده‌ها در هنگام خرابی غیرمنتظره سیستم از بین بروند و سیستم را غیرقابل اعتماد کنند. برای غلبه بر این مشکل، چندین تکنیک وجود دارد که می‌توان از آنها استفاده کرد. به‌عنوان مثال، این پایگاه داده به‌صورت دوره‌ای از داده‌ها پشتیبان می‌گیرد و یا گزارش به‌روزرسانی را بر روی دیسک‌ها ذخیره می‌کند تا از داده‌ها محافظت کند. عملیات ذخیره‌سازی بر روی دیسک ممکن است به‌طور قابل توجهی به عملکرد کلی آسیب برساند. با این حال، به‌دلیل ظرفیت نسبتاً کمتر حافظه اصلی در مقایسه با دیسک‌ها، سیستم‌های ذخیره‌سازی حافظه تک‌گره[2] ظرفیت کافی برای داده‌های حجیم امروزی را ندارند. اگرچه مدل ذخیره‌سازی کلید-مقدار در پایگاه‌داده غیررابطه‌ای، مقیاس‌پذیری سیستم‌های ذخیره‌سازی را آسان کرده، اما همچنان با چالش‌هایی مانند شکست تک‌گره[3]، ناسازگاری داده‌ها و غیره مواجه است (Chen et al. 2016).

۳-۲. مجموعه داده مورد ارزیابی

به‌منظور ارزیابی عملکرد سه پایگاه داده MySQL، MongoDB و Redis، مجموعه داده‌ای از توییت‌های منتشرشده توسط کاربران شبکه اجتماعی «توییتر» از آدرس اینترنتی https://sakoo.ai/dataset/6 دانلود شد. این وب‌سایت توسط شرکت دانش‌بنیان و مبتنی بر تکنولوژی دیتاک[4] با تکیه بر دانش و تجربه چندین ساله در حوزه کلان‌داده و هوش مصنوعی، خدمات و محصولات متنوعی را برای تبدیل داده به دانش ارائه می‌نماید. این مجموعه دادگان متشکل از ۱۵۸۱۱۴۶۸ پیام توییتری منتشرشده در شبکه اجتماعی «توییتر» است که این مقدار توییت از پست‌های ارسالی توسط ۳۰۰۰۰۰ کاربر فارسی‌زبان این شبکه اجتماعی جمع‌آوری شده است. در این مجموعه به‌ترتیب، افزون بر تاریخ انتشار

1. message broker
2. single node
3. single point of failure
4. https://dataak.com/

هـر توییـت، متـن، تعـداد بازنشـرها[1]، تعـداد فیـو (لایـک)[2]، شناسـه یکتـای کاربـر منتشـرکننده، تعـداد دنبال‌کننـدگان کاربـر منتشـرکننده، تعـداد دنبال‌شـوندگان کاربـر منتشـرکننده و تعـداد کل توییت‌هـای منتشرشـده توسـط آن کاربـر نیـز بـه ازای هـر توییـت وجـود دارد.

۳-۳. پرس‌وجوهای مورد ارزیابی

پرس‌وجوهایـی کـه بـرای ایـن تحقیق در نظر گرفته و انجام شـده، شـامل درج[3]، به‌روزرسـانی[4] و حـذف[5] هسـتند کـه نتایـج عملکـرد اجـرای این دسـتورات بـر روی مجموعـه داده مـورد آزمایش بـر اسـاس جـداول ارائه‌شـده، ارزیابـی و تحلیـل می‌گردنـد. سـنجش‌ها بـا اسـتفاده از نرم‌افزارهـای MongoDB-compass-1.32.2، Redis-x64-5.0.14.1 و PhpMyAdmin 5.0.1 انجـام شـده اسـت. مشخصات سیستم استفاده‌شده برای ارزیابی، پردازنده Intel(R) Core (TM) i5–4300U @ 1.90GHz و سیسـتم عامـل Windows10، ۶۴ بیتی اسـت. این سیسـتم دارای ۸ گیگابایت حافظـه فیزیکی و ۲۵۶ گیگابایـت فضـای هارددیسـک SSD اسـت. به‌دلیـل حجیـم بـودن داده‌ها امکان اسـتفاده مسـتقیم از رابط‌هـای نرم‌افـزاری تحـت ویـندوز ایـن پایگاه‌هـای داده وجـود نداشـت. همچنیـن بـرای بعضـی از پرس‌وجوهـا نیـاز بـه کدنویسـی و ساده‌سـازی عملیـات بـود کـه بـا توجـه بـه نیاز هـر پایـگاه داده کتابخانه‌هـای مرتبـط نصـب گردیـد. بـرای اندازه‌گیـری اجـرای قطعه‌کدهـای مربـوط از کتابخانـه timeit اسـتفاده شـده اسـت. به‌دلیل سـاختار مشـخص جداول برای ذخیره‌سـازی داده در پایگاه داده MySQL نیاز به یکسان‌سـازی سـاختار، نام سـطر و سـتون‌های فایل دیتاسـت با سـاختار جدول ایجاد شـده بـود. بعـد از سـاخت جـدول، دسـتور SQL درج داده‌ها اجـرا و زمـان بارگـذاری کل مجموعه داده اندازه‌گیـری گردیـد. بـرای پرس‌وجوهـای درج، به‌روزرسـانی و حـذف، بـا اسـتفاده از توابـع مربـوط بـه زبـان برنامه‌نویسـی، ۵۰ رکـورد به‌صورت تصادفی انتخاب شـدند. سـپس دسـتورات SQL مربـوط بـر روی آن‌هـا اعمـال گردیـد و زمـان اجـرای هـر یـک اندازه‌گیـری شـد. بعـد از سـاخت مجموعـه، بـرای بارگـذاری دیتاسـت در MongoDB به‌دلیـل سـاختار پایـگاه داده غیررابطـه‌ای نیازی بـه تعریـف سـاختار اولیـه وجود نداشـت و از همان فایل CSV دیتاسـت اسـتفاده گردید. بـرای انجام پرس‌وجوهـا، به‌طـور مجـدد ۵۰ رکـورد به‌صورت تصادفی انتخاب شـد و سـپس مدت‌زمـان اجرای دسـتورات مربـوط MongoDB بـر روی آن‌هـا اندازه‌گیـری گردید.

بـرای آزمایـش پایـگاه داده Redis در سیسـتم‌عامل ویـنـدوز، فایـل نصبـی ایـن پایـگاه داده

1. retweet
2. favorite
3. insert
4. update
5. delete

و همچنین رابط کاربری مناسبی به‌صورت منبع باز و معتبر در دسترس نبود. در سایت اصلی این پایگاه داده نیز این نکته ذکر شده که بهتر است پایگاه داده Redis با استفاده از سیستم‌عامل «لینوکس» اجرا گردد و یا در صورت استفاده از سیستم‌عامل «ویندوز»، با استفاده از ماشین مجازی سیستم‌عامل «لینوکس» را اجرا کرده و سپس Redis بر روی آن نصب شود. با توجه به محدودیت مشخصات سخت‌افزاری سیستم مورد استفاده، امکان نصب ماشین مجازی و نصب Redis و اجرای عملیات سنگین پرس‌وجوها وجود نداشت. توسعه‌دهندگان متخصص در این زمینه در وب‌سایت گیت‌هاب[1]، نسخه قابل استفاده Redis را برای «ویندوز» منتشر کرده‌اند که در این آزمایش از نسخه Redis-x64-5.0.14.1 استفاده شده است. البته به‌منظور آزمایش برروی سیستم‌عامل «لینوکس»، که پیشنهاد اصلی وب‌سایت این پایگاه داده نیز بود، سرور ابری[2] از وب‌سایت پارس‌پک[3]، که ارائه‌دهنده سرورهای قدرتمند مجازی است، با مشخصات جدول ۴، خریداری شد، تا با نصب نسخه اصلی Redis و اجرای عملیات، نتایج دقیق‌تر مقایسه شوند.

جدول ٤. مشخصات سرور محاسباتی استفاده‌شده برای آزمایش با سیستم عامل «لینوکس»

موقعیت سرور	سیستم عامل	حافظه اصلی	هارددیسک	تعداد CPU
ایران	Ubuntu20-x64	4096 MB	25 GB	2

٤. تجزیه‌وتحلیل یافته‌ها

٤-١. یافته‌های سنجش

مقادیر اندازه‌گیری‌شده برای درج، به‌روزرسانی و حذف داده‌ها در سه پایگاه داده به‌ترتیب، در جداول ۵، ۶ و ۷ نشان داده شده، و جدول ۸ نیز فضای ذخیره‌سازی مجموعه داده مورد بررسی را نشان می‌دهد.

جدول ۵. مشخصات اجرای درج داده

	INSERT		
نام پایگاه داده	زمان ساخت جدول و درج کامل دیتاست (m:s:ms)	زمان درج ۵۰ رکورد در دیتاست کامل (m:s:ms)	زمان درج ۵۰ رکورد در ۵ میلیون رکورد (m:s:ms)
MySQL	۷۸:۰۶:۶۶	۰:۰۴:۰۹	۰:۰۶:۰۷

1. GitHub
2. public cloud
3. https://parspack.com/

INSERT			
نام پایگاه داده	زمان ساخت جدول و درج کامل دیتاست (m:s:ms)	زمان درج ۵۰ رکورد در دیتاست کامل (m:s:ms)	زمان درج ۵۰ رکورد در ۵ میلیون رکورد (m:s:ms)
MongoDB	۴۱:۳۱:۹۸	۰:۰۱:۲۴	۰:۰۱:۱۹
Redis	۱۹۸:۳۱:۱۹	۰:۰,۵۴:۱۲	۰:۰,۶:۰۲

جدول ۶. مشخصات اجرای به‌روزرسانی داده

UPDATE		
نام پایگاه داده	زمان به‌روزرسانی ۵۰ رکورد داده در دیتاست کامل (m:s:ms)	زمان به‌روزرسانی ۵۰ رکورد داده در ۵ میلیون رکورد (m:s:ms)
MySQL	۰:۰۵:۳۹	۰:۰۴:۰۸
MongoDB	۰:۰,۱۳:۰	۰:۰,۱۱:۰
Redis	۰:۰۲:۳۹	۰:۰۲:۷۸

جدول ۷. مشخصات اجرای حذف داده

DELETE		
نام پایگاه داده	زمان حذف ۵۰ رکورد داده در دیتاست کامل (m:s:ms)	زمان حذف ۵۰ رکورد داده در ۵ میلیون رکورد (m:s:ms)
MySQL	۰:۰۴:۰۱	۰:۰۳:۲۷
MongoDB	۰:۰,۴۱:۰	۰:۰,۱۲:۰
Redis	۰:۰۲:۱۱	۰:۰۲:۵۹

جدول ۸. فضای ذخیره‌سازی مجموعه داده

نام پایگاه داده	فضای اشغال‌شده
MySQL	GB ۳/۵۰
MongoDB	GB ۳/۱۴
Redis	GB ۳/۷۵

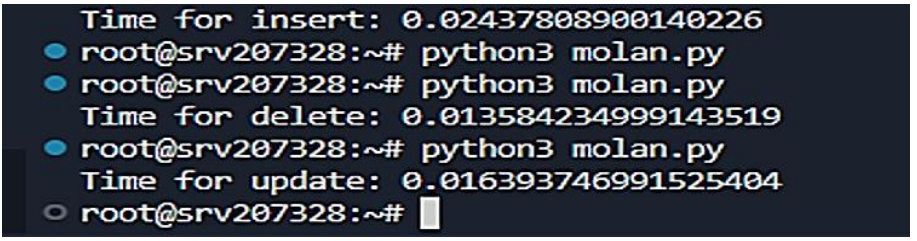
نتایج به‌دست‌آمده از درج داده‌ها در جدول ۵، ثبت شده است. ستون اول، بیانگر مدت‌زمانی است که درج کامل دیتاست در پایگاه داده غیررابطه‌ای MongoDB و Redis و پایگاه داده رابطه‌ای MySQL انجام شده است. این زمان در Redis مستقرشده در سیستم‌عامل «ویندوز»، نزدیک به ۱۹۹ دقیقه به طول انجامید. MongoDB با زمان بارگذاری نزدیک

به ۴۱ دقیقه و ۱/۹ برابر سریع‌تر از MySQL، این عملیات را اجرا کرد و بهترین عملکرد را در عملیات درج کامل دیتاست نشان می‌دهد. ستون دوم در جدول ۵، زمان درج ۵۰ رکورد داده به‌صورت تصادفی را در کل مجموعه داده به‌تفکیک سه پایگاه داده نشان می‌دهد. این مدت‌زمان در MySQL سه برابر MongoDB به‌طول انجامید. همچنین Redis با زمان درج نیم‌ثانیه بهترین عملکرد را داشت و بعد از آن MongoDB در رتبه دوم قرار گرفت. برای تست درج ۵۰ رکورد داده به‌صورت تصادفی با مقدار داده محدودتر از همان مجموعه داده، ابتدا ۵ میلیون رکورد وارد پایگاه‌های داده گردید. این امر برای آزمایش اینکه آیا اجرای پرس‌وجوهای یکسان در حجم کم داده در این سه پایگاه داده عملکرد زمانی مشابه با حجم زیاد را دارند یا خیر، انجام شده است. ستون سوم جدول ۵، نتایج به‌دست‌آمده را نشان می‌دهد. پایگاه داده Redis تقریباً با ۰/۶ ثانیه بهترین نتیجه را ارائه کرد. بعد از آن به‌ترتیب MongoDB و بعد MySQL قرار گرفتند. نتایج جداول ۶ و ۷ نشان می‌دهند که عملیات به‌روزرسانی و حذف در MongoDB در مدت‌زمان بسیار کمتری انجام شد. حذف از پایگاه داده MySQL، بدین صورت است که ابتدا عنصر مورد نظر جست‌وجو شده و سپس مورد بررسی قرار می‌گیرد که آیا به‌عنوان کلید خارجی توسط جدول دیگری استفاده شده است یا خیر، و سپس حذف می‌شود. اما در MongoDB این جست‌وجو انجام نمی‌شود. چنانکه اشاره شد، الگوریتم جست‌وجو مورد استفاده پایگاه داده MongoDB ساختار داده درختی با مرتبه زمانی O (log n) است. در واقع، بدون استفاده از این نوع ساختار داده و بدون شاخص‌گذاری، به تعداد عناصری که داخل جدول وجود دارد، مقایسه انجام می‌گیرد که مرتبه زمانی آن برابر O (n) می‌شود. این حالت در مقایسه با حالت شاخص‌گذاری‌شده، زمان زیادی است. جدول ۸، نشان می‌دهد که فضای اشغال‌شده توسط مجموعه داده در MongoDB به‌تقریب ۳۵ مگابایت کمتر از MySQL است. همچنین Redis با حجمی معادل ۳/۷۵ گیگابایت بیشترین فضا را اشغال کرده است.

برای بررسی عملکرد Redis در سیستم‌عامل «لینوکس»، ابتدا از طریق Visual Studio Code و پروتکل[1] SSH به سرور متصل شده و یک‌بار دیتاست به‌صورت کامل در پایگاه داده بارگذاری گردید. مدت‌زمانی که این عملیات به‌طور کامل انجام گرفت، به‌تقریب معادل ۸۰ دقیقه بود. اما سه پرس‌وجوهای درج، حذف و به‌روزرسانی ۵۰ رکورد به‌صورت تصادفی در کل دیتاست با عملکرد زمانی بسیار عالی انجام شد. در شکل ۲،

1. secure shell (برای ایجاد ارتباطی امن بین کاربر و سرور استفاده می‌شود)

نتایج به‌دست‌آمده از اجرای این پرس‌وجوها نشان داده شده است. از آنجا که نتایج حاکی از سرعت بالای Redis در اجرای این سه پرس‌وجو بود و بنا بر دانستن اینکه این عملیات در حافظه اصلی انجام شده و سرعت آن نسبت به دیسک بیشتر است، می‌توان به نتیجه اعتماد کرد. در جدول ۹، گرامر پرس‌وجوها در سه پایگاه داده نشان داده شده است. سادگی دستورات در پایگاه داده Redis و بعد از آن MongoDB مشهود است.



شکل ۲. عملکرد Redis در اجرای پرس‌وجوهای INSERT، UPDATE و DELETE سیستم‌عامل لینوکس (به ثانیه)

جدول ۹. گرامر دستورات INSERT، DELETE و UPDATE در پایگاه داده‌های مختلف

پرس‌وجو	MySQL	MongoDB	Redis
دستور INSERT	INSERT INTO table_name (col_name1,col_ name2) VALUES(value1,value2)	db.collection_name.insert (name1:value1,name2:value2)	db.set (index)
دستور UPDATE	UPDATE table_name SET column1 = value1 WHERE (Condition)	db.collection_name.update()	db.set (index)
دستور DELETE	DELETE FROM table_name WHERE (Condition)	db.collection_name.deleteOne (condition)	db.delete (key)

٤-٢. بحث و بررسی

نتایج مقایسه نشان می‌دهد که MongoDB با صرف زمان کمتری در مقایسه با MySQL و Redis هر سه پرس‌وجو را اجرا می‌کند. همچنین با توجه به مقادیر نشان داده‌شده در جدول ۸، از فضای ذخیره‌سازی هر پایگاه داده بر روی دیسک، MongoDB کمترین فضا را استفاده نموده است. بنابراین می‌توان گفت که این نوع پایگاه داده NoSQL، عملکرد و مقیاس‌پذیری خوبی را برای اکثر عملیات در مجموعه داده‌های حجیم ارائه می‌دهد. این یافته با یافته‌های تحقیق Ongo & Kusuma (2018) و Filip & Čegan (2020) همخوانی و مطابقت دارد.

در شکل ۲، زمان اجرای سه پرس‌وجوی INSERT، UPDATE و DELETE بر روی ۵۰ رکورد به‌صورت تصادفی در کل دیتاست پایگاه داده Redis نصب‌شده بر روی سیستم‌عامل «لینوکس» نشان داده شده است. عملکرد فوق‌العاده Redis در سیستم‌عامل «لینوکس» را می‌توان به حالت نیمه‌پایدار آن نسبت داد، یعنی ابتدا همه داده‌ها در حافظه ذخیره می‌شوند و سپس به‌صورت غیرهمزمان و به‌طور منظم برای ذخیره‌سازی دائمی روی دیسک ذخیره می‌شوند که پژوهش Tang & Fan (2016) نیز نتایج مشابهی را به‌دست آورده و به این موضوع اشاره نموده است. در این آزمایش، Redis به‌طور پیش‌فرض، فایل‌های پشتیبان از مجموعه داده را در یک فایل باینری به نام dump روی دیسک ذخیره می‌کند که منجر به ذخیره دائمی داده‌ها می‌شود. همچنین Redis را می‌توان با توجه به نیاز، روی حالت‌های دیگر مانند حالت In-Memory و یا Fully Persistent تنظیم کرد. بدیهی است که حالت‌های مختلف ویژگی‌های متفاوتی دارند.

به‌طور کلی، نتایج به‌دست‌آمده از یافته‌های پژوهش را می‌توان به این صورت بیان کرد که هر سه پایگاه داده MySQL و MongoDB و Redis مزایا و محدودیت‌های مرتبط با ساختار خود را دارند. به‌طور خلاصه، می‌توان نتیجه گرفت که اگر داده‌ها دارای ساختار مشخصی هستند، پایگاه داده رابطه‌ای مانند MySQL می‌تواند انتخاب خوبی باشد. اگر داده‌ها بدون ساختار و یا ساختاریافته با پتانسیل رشد سریع هستند، مدل‌های NoSQL انتخاب بهتری هستند. از طرفی، MongoDB با صرف زمان کمتری در مقایسه با MySQL و Redis پرس‌وجوها را اجرا می‌کند و این اثباتی است بر اینکه پایگاه‌های داده NoSQL عملکرد و مقیاس‌پذیری خوبی را برای اکثر عملیات در مجموعه داده‌های بزرگ نشان می‌دهند؛ ضمن اینکه ارزیابی‌های پژوهش Jose & Abraham (2020) نیز بر این موضوع تأکید نموده است.

همچنین، اگرچه هنگام ذخیره‌سازی حجم بالای داده در پایگاه داده Redis در سیستم‌عامل «ویندوز» به‌دلیل محدودیت منابع اجرایی، زمان بارگذاری کامل دیتاست بسیار طولانی‌تر از انتظار شد، چنانچه نگهداری داده به‌صورت موقت و با سرعت زیاد مد نظر باشد، پایگاه داده‌های کلید-مقدار مثل Redis که داده‌ها را در حافظه نگهداری می‌کنند، مناسب هستند. در ضمن، باید توجه داشت که به‌رغم سرعت بالای اجرای پرس‌وجوهای INSERT، UPDATE و DELETE در پایگاه داده Redis، نمی‌توان از Redis به‌عنوان پایگاه داده قابل اعتماد و دائمی یاد کرد؛ زیرا این نوع پایگاه داده برروی حافظه

موقت مستقر است و صرفاً برای خواندن‌های زیاد مورد استفاده قرار می‌گیرد. به همین علت، برای caching یا اطلاعاتی که به‌طور موقت مورد نیاز هستند، مناسب است. از طرفی، معمولاً درخواست خواندن داده‌ها در شبکه‌های اجتماعی بیشتر از نوشتن داده‌هاست که در نتیجه استفاده از یک پایگاه داده مانند Redis که در حافظه سیستم موقت مستقر می‌شود، موجب افزایش سرعت بازیابی اطلاعات، جلوگیری از مراجعات زیاد به پایگاه داده اصلی، و بهبود کارایی سیستم می‌شود.

۵. نتیجه‌گیری

امروزه، با توجه به گسترش شبکه‌های اجتماعی، نیاز به ذخیره‌سازی و پردازش سریع داده‌ها به‌شدت افزایش یافته است. شبکه‌های اجتماعی به‌دلیل ارائه سرویس‌های متعددی که دارند، از معماری مبتنی بر سرویس استفاده می‌کنند که در این معماری ابتدا نحوه مدیریت کردن داده‌ها مشخص شده و سپس متناسب با نیازمندی‌های موجود، بهترین پایگاه داده انتخاب می‌شود. به همین علت، یکی از مهم‌ترین فاکتورها برای انتخاب پایگاه داده مناسب، شناخت درست نیازمندی‌های سیستم و سپس شناخت انواع پایگاه داده، مزایا و معایب آن‌هاست. مدل‌های رابطه‌ای همواره یکی از پرکاربردترین رویکردها برای مدیریت داده‌ها بوده‌اند و در حال حاضر هم در شبکه‌های اجتماعی برای داده‌های ساختاریافته و مرتبط به هم استفاده می‌شوند. با این حال، مدل‌های رابطه‌ای دارای محدودیت‌های متعددی هستند که می‌توانند در موارد استفاده خاص مشکل‌ساز باشند. مسئله اصلی این است که پایگاه داده‌های رابطه‌ای برای مدیریت حجم زیادی از داده‌ها نمی‌توانند مؤثر واقع شوند. در این راستا، پایگاه‌های داده NoSQL قابلیت انعطاف‌پذیری بیشتری برای ذخیره و پردازش داده‌ها به‌وجود می‌آورند؛ به این معنا که نیازی به مشخص‌کردن طرح و یا نوع داده از پیش تعیین‌شده‌ای نیست. در حال حاضر، تعداد زیادی پایگاه داده NoSQL وجود دارند که ویژگی‌ها و امکانات متفاوتی ارائه می‌دهند.

MongoDB یکی از انواع پایگاه داده NoSQL مبتنی بر سند، مقیاس‌پذیر، متن باز و دارای عملکرد بالاست. این پایگاه داده با داشتن مزایای مختلف در حال حاضر عمدتاً برای شبکه‌های اجتماعی برنامه‌هایی با داده‌های حجیم و یا سایر کارهای پردازشی دیگر که شامل داده‌هایی هستند که در یک پایگاه داده مدل رابطه‌ای پشتیبانی نمی‌شود، مورد استفاده قرار می‌گیرد. این مدل پایگاه داده امکان استفاده از انواع متغیرها را بدون هیچ

مشکلی ارائه می‌کند. افزون بر این، ویژگی ذخیره‌سازی بدون الگو و استفاده از فرمت داده‌هایی مانند JSON، تغییر داده‌ها، به‌روزرسانی و حذف را به سهولت در هر زمانی ممکن می‌کند. از مزایای دیگر این پایگاه داده در مقایسه با پایگاه داده‌های رابطه‌ای، می‌توان به تنظیمات محیطی بسیار ساده و پیاده‌سازی و راه‌اندازی سریع آن اشاره نمود.

Redis نیز یک پایگاه داده غیررابطه‌ای سریع است که بر اساس معماری کلید-مقدار کار می‌کند و ویژگی‌هایی همچون انتقال داده از حافظه ناپایدار به حافظه پایدار را داراست. از Redis، می‌توان به‌عنوان یک پایگاه داده موقت و برای ذخیره‌سازی اطلاعاتی که بسیار پرکاربرد هستند، در لایه cache استفاده کرد. استفاده از این پایگاه داده به‌عنوان پایگاه داده اصلی سیستم‌های نرم‌افزاری پیشنهاد نمی‌شود، زیرا کنترل داده‌هایی که دارای ارتباطات زیادی با هم هستند، با ساختار کلید-مقدار بسیار سخت و طاقت‌فرساست. همچنین با توجه به اینکه Redis داده‌ها را در حافظه موقت به‌عنوان یک حافظه ناپایدار نگهداری می‌کند، در صورت بروز مشکل امکان از دست رفتن داده‌ها وجود دارد. البته برای رفع این مسئله امکان انتقال داده‌ها به حافظه پایدار مانند دیسک وجود دارد که از طریق تنظیمات فایل config صورت می‌گیرد. طبق گزارش وب‌سایت رسمی Redis[1]، این پایگاه داده توسط شرکت‌های بزرگی چون Twitter، GitHub، Snapchat، Craigslist، StackOverflow استفاده می‌شود. برای مثال، «توییتر» در خوشه‌های Redis داده‌هایی چون درجه تأثیر[2]، نرخ تعامل[3] و پیام‌های مستقیم[4] را ذخیره می‌کند.

از نظر فضای ذخیره‌سازی، MongoDB فضای کمتری را نسبت به MySQL و Redis اشغال می‌کند و استفاده از نرم‌افزار MongoDB Compass و اتصال آن به «پایتون»، بسیار کاربرپسند است. از آنجا که پایگاه داده غیررابطه‌ای MongoDB مبتنی بر سند است و از ساختار داده JSON پشتیبانی کرده و نیازی به ایجاد ساختار خاصی ندارد، کار با آن بسیار راحت‌تر است.

به‌عنوان آخرین نکته و تجربه مشاهده‌شده، به‌دلیل حجم بالای جامعه توسعه‌دهندگانی که از پایگاه داده‌های NoSQL استفاده می‌کنند، مستندسازی و مشارکت برای پایگاه داده MongoDB در اینترنت به خوبی مشاهده می‌شود. با استناد بر موارد بالا، می‌توان گفت

1. https://redis.io/docs/about/users/
2. impression
3. engagement data
4. direct messages

MongoDB بـرای پروژه‌هـای تجـاری و کوچـک بـا حجـم زیـاد داده، حداقـل تـا زمانـی کـه نیازهـای پـروژه بـه شـکل کامـل تحلیـل نشـده، گزینـه مناسـبی اسـت. همچنیـن تأکیـد می‌شـود کـه ابتـدا پایـگاه داده‌هـای NoSQL را لازم اسـت از جنبه‌هـای دیگـری ماننـد محاسـبه تأخیـر، کارایـی بعـد از نمایه‌سـازی صحیـح و غیـره بررسـی و سـپس تصمیـم مناسـبی در جهـت انتخـاب آن‌هـا اتخـاذ نمـود.

References

Auradkar, A., C. Botev, S. Das, D. De Maagd, A. Feinberg, P. Ganti, L. Gao, B. Ghosh, K. Gopalakrishna, B. Harris, et al. Data infrastructure at LinkedIn. 2012. IEEE 28th International Conference on Data Engineering. pp. 1370-1381: IEEE. Arlington, VA, USA USA. ISBN: 978-1-4673-1640-8.

Bhogal, J. & I. Choksi. 2015. Handling big data using NoSQL. In: IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2015. South Korea.

Borthakur, D., J. Gray, J. S. Sarma, K. Muthukkaruppan, N. Spiegelberg, H. Kuang, K. Ranganathan, D. Molkov, A. Menon, S. Rash et al. 2011. Apache hadoop goes realtime at facebook. Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. ACM. pp. 1071–1080. Athens, Greece.

Bronson, N., Z. Amsden, G. Cabrera, P. Chakka, P. Dimov, H. Ding, J. Ferris, A. Giardullo, S. Kulkarni, H. Li, et al. 2013. {TAO}: Facebook's distributed data store for the social graph. In 2013 {USENIX} Annual Technical Conference ({USENIX} {ATC} 13), pages 49–60. San Jose CA.

Caers, R., T. D. Feyter, M. D. Couck, T. Stough, C. Vigna, C. D. Bois. 2013. Facebook: A literature review. *New Media & Society*. 15 (6): 982 - 1002.

Chen, S., X. Tang, H. Wang, H. Zhao & M. Guo. 2016. Towards scalable and reliable in-memory storage system: A case study with Redis. IEEE Trustcom/BigDataSE/ ISPA, 2016, pp. 1660-1667. doi: 10.1109/TrustCom.2016.0255.

Das, A., A. Mitra, S. N. Bhagat & S. Paul. 2020. Issues and Concepts of Graph Database and a Comparative Analysis on list of Graph Database tools. International Conference on Computer Communication and Informatics (ICCCI), pp. 1-6: IEEE. Coimbatore, India.

Davoudian, A., L. Chen & M. Liu. 2018. A survey on NoSQL stores. *ACM Computing Surveys (CSUR)* 51 (2): 1-43.

Dawodi, M., M. H. Hedayati, J. A. Baktash & A. L. Erfan. 2019. Facebook MySQL Performance vs MySQL Performance. IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2019, pp. 0103-0109: IEEE. Vancouver, British Columbia, Canada.

Dipina Damodaran, B., S. Salim & S. M. Vargese. 2016. Performance evaluation of MySQL and MongoDB databases. *International Journal on Cybernetics & Informatics* (IJCI). vol. 5: ?

Fernandes, D. & J. Bernardino. 2018. Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. Proceedings of the 7th International Conference on Data Science, Technology and Applications - Volume 1: DATA, INSTICC. SciTePress. doi: 10.5220/0006910203730380. ISBN 978-989-758-318-6 pp. 373–380.

Filip, P. & L. Čegan. 2020. Comparison of mysql and mongodb with focus on performance. International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS). pp. 184-187: IEEE. Jakarta, Indonesia.

Gašpar, D. & M. Mabić. 2017. NoSQL databases as social networks storage systems. Proceedings of the ENTRENOVA - ENTerprise REsearch InNOVAtion Conference, Dubrovnik, Croatia. pages 251-257.

Han, J., E. Haihong, G. Le & J. Du. 2011. Survey on NoSQL database. 6th international conference on pervasive computing and applications. pp. 363-366: IEEE. Port Elizabeth, South Africa.

Jose, B. & S. Abraham. 2017. Exploring the merits of nosql: A study based on mongodb. International Conference on Networks & Advances in Computational Technologies (NetACT). pp. 266-271: IEEE. DOI: 10.1109/NetACT41440.2017

Jose, B. & S. Abraham. 2020. Performance analysis of NoSQL and relational databases with MongoDB and MySQL. *Materials today: PROCEEDINGS*. 24: 2036-2043.

Liu, W., A. Sidhu, A. M. Beacom & T. W. Valente. 2017. Social network theory. The international encyclopedia of media effects. Hoboken: Wiley. pp. 1-12.

Mathew, A. B. & S. M. Kumar. 2015. Analysis of data management and query handling in social networks using NoSQL databases, International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp. 800-806: IEEE. Kochi, India.

Medhi, S. & H. K. Baruah. 2017. Relational database and graph database: A comparative analysis. *Journal of Process Management- New Technologies* 5 (2): 1-9.

Ongo, G. and G. P. Kusuma. 2018. Hybrid database system of MySQL and MongoDB in web application development. International Conference on Information Management and Technology (ICIMTech). pp. 256-260: IEEE. Jakarta, Indonesia.

Palanisamy, S. & P. SuvithaVani. 2020. A survey on RDBMS and NoSQL Databases MySQL vs MongoDB. International Conference on Computer Communication and Informatics (ICCCI). pp. 1-7: IEEE. Coimbatore, India.

Ramesh, D. & A. Kumar. 2018. Query Driven implementation of Twitter base using Cassandra. International Conference on Current Trends towards Converging Technologies (ICCTCT). pp. 1-4: IEEE. Coimbatore, India.

Roein Fard, F., F. Bahrampour & H. Jahanshahi Nokandeh. 2018. A Study of NoSQL and MongoDB Application, 6th National Conference on Applied Research in Computer Engineering and Information Technology. Tehran https://civilica.com/doc/1011692.

Sharma, M., V. D. Sharma & M. M. Bundele. 2018. Performance Analysis of RDBMS and No SQL Databases: PostgreSQL, MongoDB and Neo4j. 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE). pp. 1-5: IEEE. Jaipur, India.

Tang, E. & Y. Fan. 2016. Performance comparison between five NoSQL databases. 7th International Conference on Cloud Computing and Big Data (CCBD). pp. 105-109: IEEE. Macau, China.

موژان صفائی‌نژاد

دارای مدرک کارشناسی ارشد مهندسی کامپیوتر از دانشگاه گلستان و فارغ‌التحصیل سال ۱۴۰۱ است.

مدیریت اطلاعات، پایگاه داده و شبکه‌های اجتماعی از جمله علایق پژوهشی وی است.

سهیلا کرباسی

استادیار گروه کامپیوتر دانشگاه گلستان است.

سیستم‌های اطلاعاتی، فرایندهای بازیابی اطلاعات و داده‌کاوی از جمله علایق پژوهشی وی است.

پژوهش‌نامه

پردازش و مدیریت اطلاعات